

Ensinando Programação em Lógica por meio do Desenvolvimento de Aplicações Comerciais para a Internet

Carlos R. Lopes¹, Alexsandro S. Soares¹, Paula C. Souza¹, Sérgio C. Peres¹

¹Faculdade de Computação – Universidade Federal de Uberlândia (UFU)
CEP 38400-902 – Uberlândia – MG – Brasil

{crlopes, alex}@facom.ufu.br, {paulacamargo, aranzel}@comp.ufu.br

Abstract. *Many students find it difficult to grasp the fundamental concepts of Logic Programming, which results in a poor performance and consequently in a failure. There is a misconception that Logic Programming paradigm is not appropriate for the software industry, which also contributes to lower student motivation. Aiming to increase student motivation, this article describes a proposal and also the resources developed in recent years for teaching Logic Programming via the implementation of commercial internet-based applications in Prolog. This approach has become a highly motivating factor for students. Preliminary results show an increased interest in learning concepts as well as an increase in the pass rate observed in the discipline.*

Resumo. *Muitos estudantes encontram dificuldades ao serem expostos aos conceitos fundamentais da Programação em Lógica, levando a um desempenho insuficiente nas avaliações o que resulta conseqüentemente em reprovação. Existe uma concepção errônea sobre o paradigma lógico não se adequar à indústria de software, diminuindo a motivação dos alunos. Este artigo descreve um proposta e também os recursos desenvolvidos para o ensino de Programação em Lógica via a implementação em Prolog de aplicações comerciais baseadas na internet. Essa abordagem tornou-se um fator altamente motivador para os estudantes, refletido no aumento do interesse na aprendizagem de conceitos, assim como no incremento na taxa de aprovação observada na disciplina.*

1. Introdução

Programação em Lógica é um paradigma de programação importante em Ciência da Computação. Cada paradigma fornece uma forma alternativa de se resolver problemas e não raro alguns problemas de fácil solução em determinado paradigma podem se tornar muito mais complexos se abordados em outro. Por essa razão, muitos programas de Engenharia de Computação ou de Ciência da Computação incluem a disciplina de Programação em Lógica em seus currículos.

O paradigma de Programação em Lógica apresenta alguns conceitos raramente encontrados na programação procedimental, predominante na maioria dos cursos. Muitos estudantes encontram dificuldades ao serem expostos aos conceitos fundamentais da programação em lógica, levando a um desempenho insuficiente nas avaliações e que resulta conseqüentemente em reprovação. Esse não é o único problema.

Existe uma concepção errônea entre os estudantes, e também entre alguns professores, que Programação em Lógica é apenas uma curiosidade acadêmica, levando muitos

a pensar que o paradigma todo não é adequado à indústria de software. Isto por sua vez induz o aluno a não reservar tempo suficiente para o estudo adequado desse paradigma, forçando o professor a dedicar tempo e esforço extras para motivar os estudantes.

Os autores deste artigo acreditam que um método inovador e recursos motivadores deveriam ser desenvolvidos para auxiliar os estudantes na aprendizagem dos conceitos de Programação em Lógica.

Este artigo descreve um proposta e também os recursos desenvolvidos nos últimos anos para o ensino de Programação em Lógica via a implementação de aplicações comerciais em Prolog. Essa tarefa é dificultada em nossa universidade pois os estudantes que cursam a disciplina de Programação em Lógica estão no primeiro ano do curso e o desenvolvimento de aplicações comerciais requer muitos conceitos disponíveis apenas em disciplinas de outros períodos do curso.

Para que a abordagem proposta aqui funcionasse, foi observado que é possível obter na internet monografias de Trabalhos de Conclusão de Curso, escritos por estudantes do último ano de seus cursos. De fato, para a obtenção do diploma em muitos cursos de Engenharia da Computação, Sistemas de Informação ou Ciência da Computação a escrita da monografia e posterior defesa do Trabalho de Conclusão de Curso é obrigatória e, muitas vezes, o texto completo da monografia é disponibilizado na internet.

Nós investigamos algumas dessas monografias e notamos que muitas delas descrevem soluções para parte de projetos reais feitos para empresas. Em geral esses projetos consistem no desenvolvimento de aplicações para a internet e requerem ferramentas para o desenvolvimento para a *web* e/ou para o gerenciamento de bancos de dados.

Infelizmente, não temos livros que cubram tais assuntos em Prolog, uma linguagem de programação representativa do paradigma de programação em lógica. Tome-se como exemplo o livro *Prolog Programming for Artificial Intelligence* [Bratko 2011], um livro conhecido no ensino de Prolog e Inteligência Artificial, que não possui referência alguma a aplicações para a internet. Em virtude desse fato, foram desenvolvidos materiais pedagógicos nos quais o desenvolvimento para a *web* e o gerenciamento de bancos de dados puderam ser realizados diretamente em Prolog.

Durante nossos cursos foram propostos projetos baseados nas monografias mencionadas anteriormente, algumas vezes com simplificações para torná-los factíveis de conclusão no tempo de um semestre letivo.

Para comparar a abordagem atual com a adotada anteriormente, os autores formularam um questionário que foi submetido aos estudantes. A compilação das respostas nos levou a concluir que foram benéficos, em termos de aprendizagem e motivação, o uso do método e dos recursos desenvolvidos pelos autores. Nós acreditamos que a mesma abordagem poderá ser utilizada com sucesso em qualquer curso cujo tema seja a Programação em Lógica.

2. Trabalhos relacionados

Muitos projetos foram desenvolvidos com o objetivo de motivar estudantes para o aprendizado de programação lógica. Muitos deles tem por finalidade o desenvolvimento de jogos e a resolução de quebra-cabeças baseados na utilização de técnicas de Inteligência Artificial (IA). Exemplos incluem o Robocup, Sudoku e Ataxx.

RoboCup [Kitano et al. 1997] é uma iniciativa científica internacional com o objetivo de avançar o estado da arte em robôs móveis inteligentes. Desde o seu estabelecimento em 1997, a missão original era criar uma equipe de robôs capazes de vencer a seleção campeã do mundo de futebol no ano de 2050. Robôs equipados com estratégias baseadas em IA podem ser usados para motivar os estudantes.

Uma outra motivação para estudantes aprenderem programação lógica são os jogos lógicos. Existem muitos quebra-cabeças interessantes que podem ser resolvidos, por exemplo, com Prolog. De fato, o esforço necessário para resolver muitos desses problemas em Prolog é baixo [Linck and Schubert 2011].

[Ribeiro et al. 2009] desenvolveram um sofisticado *framework* de competição baseado no Ataxx, que inclui uma interface gráfica. Ataxx é jogado em tabuleiro por dois jogadores que revezam a vez de jogar, usando peças de cores diferentes. Apesar de possuir regras muito simples, o Ataxx oferece um ambiente rico e interessante para jogos. Os autores relataram que a resposta obtida dos estudantes foi impressionante, ultrapassando suas expectativas mais otimistas.

[Linck and Schubert 2011] entendem que estudantes de computação deveriam aprender mais que um paradigma de programação. Cada paradigma é único e influencia a forma que uma pessoa raciocina para resolver problemas. Os autores argumentam que Programação em Lógica deve ser um dos paradigmas ensinados. Com intenção de motivar os estudantes, sugerem o uso de aplicações com finalidades práticas. Além disto, os autores apresentam um plano didático para a aprendizagem de programação lógica.

[Stamatis and Kefalas 2007] descrevem os erros mais comuns cometidos por estudantes quando estão programando em Prolog. Seus resultados vem de testes nos quais os estudantes tinham que marcar quais trechos de códigos eram os corretos. O artigo cita as confusões e explica o porquê dos estudantes cometerem tais erros. Os erros estão relacionados aos seguintes tópicos: recursão, programação descendente (*top down*), programação incremental e não determinismo inteligente.

Diferentemente dos projetos anteriores, nossa abordagem recai sobre o desenvolvimento de aplicações interessantes para a indústria de software. Nos dias atuais existe uma demanda crescente por aplicações baseadas na internet. Além disto, muitos dos alunos que terminaram suas graduações estão empregados em empresas que desenvolvem aplicações para a internet. Daí segue a ideia de propor projetos que consistam no desenvolvimento de sistemas para a *web* usando Prolog, criando assim uma motivação maior para os estudantes aprenderem programação lógica.

A abordagem adotada neste artigo está em conformidade com uma das quatro diferentes abordagens para o ensino de programação descrito por Selby [Selby 2011]. Segundo essa autora a abordagem por sistema completo (*full system*) é aquela no qual os aprendizes projetam uma solução para um problema não trivial e representativo dos sistemas comerciais reais. Aos estudantes é apresentado o uso de ferramentas e de conceitos essenciais de uma linguagem de programação somente quando o problema requisitar essas informações para a sua solução.

3. Panorama da Programação em Lógica em nossa universidade

O curso de Bacharelado em Ciência da Computação oferecido pela nossa faculdade possui um currículo no qual a disciplina de Programação Lógica (PL) é incluída no segundo período. Em termos de habilidade de programação, um curso em programação procedimental serve como pré-requisito informal para a PL. Entretanto, os conhecimentos necessários para a construção de sistemas baseados na internet ou para o gerenciamento de bancos de dados somente serão oferecidos em disciplinas ofertadas nos demais períodos do curso.

Em nossa universidade dá-se especial atenção ao uso de sistemas baseados em código livre. Devido a isto, nossa faculdade decidiu pelo uso do SWI-Prolog [Wielemaker et al. 2012].

O curso de Programação Lógica está incluído em nossos currículos por mais de vinte anos. Infelizmente, um número razoavelmente grande de estudantes enfrentam dificuldades em adquirir as competências mínimas exigidas para as suas aprovações. A percepção de que a programação lógica somente serve para finalidades acadêmicas e não para a construção de sistemas comerciais, em nada contribui para motivar os estudantes a se empenharem mais.

4. Motivando estudantes via o desenvolvimento de aplicações para a internet

Muitos cursos de graduação nas várias áreas da computação possuem um Trabalho de Conclusão de Curso (TCC) que culmina na escrita de uma monografia que depois de defendida e aprovada é disponibilizada publicamente na internet. Por exemplo, o curso de Sistemas de Informação da Fundação Universidade Regional de Blumenau (FURB) disponibiliza todas as monografias dos TCC em sua página web, com livre acesso a todos.

Os autores examinaram algumas das monografias disponíveis e em muitas delas, encontrou-se evidências de que estavam relacionadas a projetos desenvolvidos para a web. Por exemplo, [Sorroche 2010] apresenta o desenvolvimento e a implementação de um sistema para controle de uma frota de veículos. Neste trabalho além dos requisitos principais, há também a especificação, por meio de diagramas da UML, de todo o sistema, incluindo o modelo entidade e relacionamento do banco de dados criado. Mesmo quando o conteúdo não estava diretamente relacionado à web, verificou-se a possibilidade de recriar o sistema, por exemplo um sistema de gerenciamento para locadora de vídeos, usando uma arquitetura cliente-servidor, com o lado do cliente funcionando como uma interface gráfica usando os recursos disponíveis na maioria dos navegadores modernos.

Para completar a implementação dos sistemas, os estudantes foram treinados em ferramentas facilitadoras do desenvolvimento de aplicações para a internet, tais como:

- editores para HTML, PHP, JavaScript e CSS. Dreamweaver, EasyPHP e Netbeans são ferramentas utilizadas para essas tarefas.
- Sistemas de gerenciamento de dados. Em muitas das monografias analisadas foram encontradas referências para a InterBase, MySQL, PostGreSQL e DBDesigner.

A análise das monografias ajudou-nos a confirmar que o desenvolvimento de aplicações para a internet é ainda um tema importante nos dias atuais. Para corroborar, informalmente em conversas com estudantes, os autores foram informados que programação para a web é um tópico em que a maioria estava interessada em trabalhar.

5. Detalhamento dos projetos desenvolvidos pelos estudantes

No início da disciplina os estudantes foram organizados em grupos para o desenvolvimento do projeto. Tais projetos são baseados nas monografias realizadas em outras instituições e disponíveis na internet. Como mencionado anteriormente, não é difícil encontrar estas monografias.

Os professores fizeram uma seleção prévia de um conjunto de monografias que se mostraram mais adequadas à finalidade do curso. Em geral, foi observado o nível de detalhamento da arquitetura proposta na monografia. A presença de diagramas da UML juntamente com a descrição textual clara dos requisitos do sistema foram considerados um fator decisivo para a inclusão no conjunto de monografias aceitáveis. Quase todas as monografias selecionadas também continham imagens mostrando as telas dos sistemas.

Após a determinação das monografias adequadas, os grupos de estudantes foram convidados a escolher uma delas que melhor representasse os seus interesses. A escolha era mutuamente exclusiva, no sentido de uma vez escolhida por um determinado grupo, a monografia não estaria mais disponível para os demais grupos. No caso de mais de um grupo se interessar pela mesma monografia foi sugerido que entrassem em acordo para decidir quem, de fato, usaria a monografia. Se o impasse persistisse então o professor realizaria um sorteio.

O projeto todo foi organizado em três fases:

1. Desenvolvimento da interface com o usuário baseado em páginas HTML5.
2. Implementação da arquitetura descrita na monografia usando um servidor web feito em Prolog e o gerenciamento de arquivos do banco de dados, usando predicados do Prolog.
3. Integração dos dois itens anteriores em uma aplicação cliente-servidor completa.

Uma das dificuldades notadas foi o baixo número de referências bibliográficas para auxiliar os estudantes no desenvolvimento de aplicações para a internet usando SWI-Prolog. De fato, muitas vezes somente o manual do usuário estava disponível e todas as informações deveriam ser obtidas a partir daí. Esse problema se complica ao tentar encontrar materiais em um nível de descrição adequado a estudantes do primeiro ano. A solução encontrada foi planejar e desenvolver os recursos didáticos necessários.

Em primeiro lugar, foram desenvolvidos textos tutoriais para a criação de páginas HTML5 estáticas e dinâmicas usando Prolog. Não foi um objetivo do projeto que os estudantes se tornassem especialistas em HTML5. Basicamente, o material desenvolvido contém o seguinte conteúdo:

- Elementos de HTML5: menus e formulários. A razão para enfatizar esses dois componentes é explicado pelo formato das interfaces descritas nas monografias. Essencialmente, pode-se desenvolver uma versão simplificada de uma aplicação usando esses dois elementos. Também foi incluída a informação básica que permitia a inclusão de folhas de estilo (CSS3) em páginas HTML5.
- Ferramentas para a criação e visualização de páginas HTML5. Foi descrito como visualizar páginas HTML5 por meio do uso de navegadores da web. Também foi enfatizado que o material continha uma introdução básica ao HTML5. Foi deixado para o estudante a busca e posterior leitura de documentos que aprofundassem as

suas compreensões sobre a criação de páginas HTML5. Entretanto, informou-se a existência de ferramentas que permitiriam o desenvolvimento de páginas HTML5 mais rapidamente.

Para a segunda parte do projeto, foram apresentadas informações relativas a:

- Conceitos básicos sobre redes de computadores e internet. Foi descrito o processo que permite a comunicação entre computadores. Mostrou-se como uma página web armazenada em um computador pode ser vista em outro e como os diferentes tipos de conteúdo podem ser visualizados na internet. Focou-se sobre texto e figuras via a confecção de código Prolog que permite a criação dinâmica de respostas baseadas nesses dois tipos de conteúdos.
- Configuração de um servidor web em Prolog. Foi mostrado como criar um servidor web em Prolog. Também foram criados exemplos para servir textos puros e textos formatados em HTML5. Mostrou-se adicionalmente neste ponto como criar uma página HTML5 dinamicamente, usando as bibliotecas disponíveis no SWI-Prolog.
- Interação com o servidor Prolog por meio de páginas HTML5 com posterior armazenamento de informações em arquivos. Por meio de menus e formulários é possível estabelecer uma interação com o usuário do sistema. Essa interação pode envolver a leitura de dados de um formulário e o armazenamento dessas informações em um arquivo usando predicados do Prolog. Foi desenvolvido um exemplo para mostrar como implementar tal interação. Esse exemplo funcionou bem para ilustrar o funcionamento dos predicados de entrada ou saída do Prolog.

Finalmente, para a realização da última fase do projeto foram criados mais materiais didáticos com instruções sobre como integrar o Prolog a um sistema de gerenciamento de bancos de dados.

Com base nos requisitos encontrados nas monografias foram criadas informações detalhadas para:

- Instalar um sistema de gerenciamento de bancos de dados. No caso, o sistema escolhido foi o SQLite por esse ser um sistema de gerenciamento de banco de dados pequeno, robusto e já com uma interface para o SWI-Prolog. Foram fornecidas instruções de instalação tanto para as plataformas baseadas em Windows quanto para aquelas baseadas em Linux.
- Um gabarito para criar um banco de dados. A partir de um exemplo especificado como um diagrama entidade-relacionamento foi explicado como criar um banco de dados juntamente com as tabelas descritas no modelo.
- Modelos de inserção, atualização e recuperação de informações contidas no banco de dados. Usando o banco criado no item anterior foram ilustradas as operações mais usuais sobre o banco de dados. Todos os exemplos criados mostraram as mesmas operações realizadas diretamente na interface presente no SQLite e também internamente via a biblioteca disponibilizada pelo SWI-Prolog.

Muitos detalhes do projeto e gerenciamento de banco de dados foram omitidos e concentrou-se na informação básica essencial para a execução dos requisitos existentes nas monografias. Aos estudantes foi informado que mais informações seriam dadas em cursos subsequentes.

Com todas as informações dadas nesta seção os estudantes estavam aptos a desenvolverem um projeto.

É necessário observar que a execução do projeto foi realizada paralelamente à introdução dos conceitos de programação em lógica em aulas teóricas e práticas, com maior ênfase nos aspectos distintivos do paradigma lógico. Usou-se algumas aulas práticas em laboratório para auxiliar o estudante a adquirir os conhecimentos necessários para a implementação do projeto.

6. Avaliação

A abordagem anteriormente usada na disciplina Programação Lógica consistia em desenvolver projetos baseados no XPCE. O XPCE é uma biblioteca para a criação de interfaces gráficas com o usuário, disponibilizada para o Prolog e para outras linguagens de programação interativas e dinamicamente tipadas. Esse sistema está integrado ao SWI-Prolog já há algum tempo.

Com o intuito de comparar diferentes abordagens, uma usando o XPCE e outra usando HTML5, foram desenvolvidos questionários. Esses questionários foram enviados a todos os estudantes que tiveram a oportunidade de trabalhar com HTML5 e com XPCE.

A página inicial do questionário claramente descrevia o seu propósito. Essa página também continha um conjunto de instruções sobre o preenchimento e ao completá-lo, para onde deveria ser submetido. Ele foi assinado pelo coordenador do curso. Também foi enfatizado o caráter anônimo do preenchimento para deixar os estudantes mais tranquilos.

O questionário era composto de treze questões onde os perguntados poderiam expressar seus níveis de concordância com respeito a cada uma delas em uma escala de Likert. Em algumas questões foram deixados espaços adicionais para que os perguntados pudessem acrescentar alguma outra informação que julgassem necessária.

Uma atenção especial foi necessária para formular um número reduzido de questões curtas, em uma linguagem clara e simples. As questões estavam relacionadas aos seguintes temas:

- Quão fácil é integrar HTML5 e Prolog?
- Qual o nível de preferência entre HTML5 e XPCE?
- Qual é o nível de preferência entre armazenar e recuperar os dados usando o gerenciamento de arquivos em Prolog ou usando um sistema de gerenciamento de bancos de dados.

7. Discussão

Em relação ao material disponibilizado para o trabalho com HTML5, a maioria dos estudantes compreendeu o objetivo de dar uma breve introdução ao tópico. Também, eles sabiam da existência de muitos outros materiais para a aprendizagem mais aprofundada do tópico.

Os perguntados também preferiram desenvolver a interface gráfica usando HTML5 ao invés de usar o XPCE. Entretanto, os perguntados mostraram preferência pela manipulação direta dos arquivos em Prolog ao invés de utilizar um sistema de gerenciamento de banco de dados. Fornecer informações a estudantes do primeiro ano sobre

esse assunto não é uma tarefa fácil e um esforço extra deverá ser realizado para aperfeiçoar o material.

Alguns professores de Prolog poderiam gostar do último resultado pois trabalhar com um sistema gerenciador de banco de dados requer menos esforço no aprendizado da linguagem Prolog com respeito à entrada ou saída de dados. Todavia, isso não é bom para os objetivos desta proposta que almeja desenvolver aplicações que usem muitas das tecnologias comumente empregadas pela indústria do software em seus produtos, aliada à efetiva utilização da programação em lógica.

No gráfico mostrado na Figura 1 é possível observar que a taxa da aprovação aumentou com a aplicação de nossa metodologia. Esse resultado preliminar mostra um aumento do desempenho dos alunos na disciplina, o que nos leva a crer que houve um aumento também na motivação para se aprender as técnicas da programação em lógica. Entretanto, é necessário continuar monitorando o progresso dos estudantes que usam o método abordado neste artigo com o objetivo de verificar a consistência das observações já encontradas.

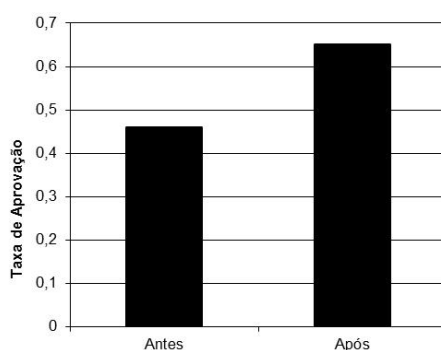


Figura 1. Taxa de aprovação antes e após a aplicação do método proposto neste trabalho.

8. Considerações finais

A programação em lógica é uma paradigma de programação importante. Assim, maiores esforços devem ser realizados para torná-la mais fácil de ser compreendida.

A compreensão dos métodos de resolução de problemas dentro do paradigma lógico ultrapassa as próprias implementações de linguagens de programação que reforçam o paradigma. Não são raras as situações, como parte de uma aplicação maior, em que se necessita de noções tais como a descrição de sistemas baseados em regras ou de algum tipo de inferência lógica. Para esses casos, mesmo se a aplicação como um todo não usar uma das linguagens da programação em lógica, o simples fato de entender que uma solução poderia ser realizada dentro do paradigma pode servir como inspiração para a efetiva solução do problema em questão.

O desenvolvimento de aplicações reais pode ser uma forma de motivar estudantes, profissionais e organizações a se interessarem mais pela abordagem lógica. Neste artigo fizemos uma contribuição nessa direção.

9. Agradecimentos

Os autores gostariam de agradecer à Pró-Reitoria de graduação da Universidade Federal de Uberlândia pelas bolsas concedidas e que tornaram possível a execução deste projeto. Também desejamos agradecer aos revisores anônimos por suas sugestões.

Referências

- Bratko, I. (2011). *Prolog Programming for Artificial Intelligence*. Pearson Education Canada, 4th edition.
- Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E., and Matsubara, H. (1997). Robocup: A challenge problem for AI and robotics. In Kitano, H., editor, *RoboCup*, volume 1395 of *Lecture Notes in Computer Science*, pages 1–19. Springer.
- Linck, B. and Schubert, S. (2011). Logic programming in informatic secondary education. In *Proceedings of 5th International Conference ISSEP 2011*, Bratislava, Slovak Republic.
- Ribeiro, P., Simões, H., and Ferreira, M. (2009). Teaching artificial intelligence and logic programming in a competitive environment. *Informatics in Education*, 8(1):85 – 100.
- Selby, C. C. (2011). Four approaches to teaching programming. In *Learning, Media and Technology: a doctoral research conference*, London, UK.
- Sorroche, R. (2010). Sistema de informações para controle da frota de veículos. Trabalho de conclusão de curso, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau – Santa Catarina.
- Stamatis, D. and Kefalas, P. (2007). Logic programming didactics. In *Proceedings of the Informatics Education Europe II Conference (IEEII)*, pages 136 – 144.
- Wielemaker, J., Schrijvers, T., Triska, M., and Lager, T. (2012). SWI-Prolog. *Theory and Practice of Logic Programming*, 12(1-2):67–96.