

PuzzlEdu: uma Ferramenta para Prover Educação como Serviço

Eric Rommel Galvão Dantas,
Ryan Ribeiro de Azevedo
CIn-UFPE / DSI-UFPI

{ergd, rra2}@cin.ufpe.br

Vinicius Cardoso Garcia, Silas
Cardoso de Almeida
CIn-UFPE

vcg@cin.ufpe.br,
silas.sca@gmail.com

Cleyton M. de O. Rodrigues,
Fred Freitas
CIn-UFPE

{cmor,fred}@cin.ufpe.br

ABSTRACT

This paper presents PuzzlEdu, a tool designed to provide education as a service by making use of cloud computing and thus aiding students and teachers in the teaching and learning of the discipline of programming language, particularly the concepts of object-oriented paradigm. We describe in detail the objectives of the tool, its architecture and infrastructure development, as well as its main features.

Keywords

Programação Orientada a Objetos, Educação como Serviço, Computação em Nuvem.

1. INTRODUÇÃO

No contexto educacional, um *software* para tal fim facilita o processo de ensino/aprendizagem, além de propiciar o desenvolvimento de diversos ambientes virtuais de aprendizagem (AVAs), favorecendo aos envolvidos nesse processo (alunos e professores) um melhor aproveitamento do conteúdo apresentado em sala de aula. Esses recursos são normalmente disponibilizados em *hardwares* próprios, executando *software* diversos, de forma a manter o ambiente sempre disponível. Utilizando-se dos conceitos de *Hardware* como Serviço (*Hardware as a Service* – HaaS) e *Software* como Serviço (*Software as a Service* – SaaS) [1], e a isso a integração de recursos educacionais disponíveis, é possível vislumbrar um novo conceito denominado: Educação como Serviço (*Education as a Service* – EdaaS) [2].

Na EdaaS, tudo passa a ser disponibilizado na nuvem computacional [1], oferecendo educação como um serviço ou uma prestação de serviços para a comunidade. Por estarem na nuvem, esses serviços não necessitam ser monitorado o tempo todo, têm nível maior de abrangência, são multiplataforma e podem ser acessados por qualquer dispositivo computacional tais como PC, *notebook*, celular etc., e ainda podem reduzir os custos com pessoal e equipamento.

A EdaaS, portanto, facilita e agrega mais conhecimento aos alunos de forma rápida e simples, aumentando o nível de discussão e interação entre alunos e professores em sala de aula. Além disso, a EdaaS pode diminuir a necessidade de novos recursos computacionais em instituições menos favorecidas financeiramente, podendo diminuir a distância educacional para instituições financeiramente mais estáveis.

Como forma de demonstrar as potencialidades da EdaaS, propomos o desenvolvimento de uma ferramenta denominada PuzzlEdu, disponibilizando seu uso como serviço. O PuzzlEdu é um *software* educacional que tem por objetivo auxiliar alunos e professores no processo de ensino e aprendizagem de POO.

Este artigo está estruturado da seguinte forma. Na Seção 2 são apresentados os principais objetivos da ferramenta. Na Seção 3 é apresentada a arquitetura e a infraestrutura de desenvolvimento do PuzzlEdu. As principais funcionalidades são apresentadas na Seção 4 e, por fim, na Seção 5, apresentam-se as conclusões.

2. OBJETIVOS DA FERRAMENTA

O PuzzlEdu, na essência, tem por objetivo principal prover educação como um serviço disponibilizado à comunidade. Para isso, utilizou-se de *cloud computing*, integrando às vantagens desse ambiente, requisitos como usabilidade, flexibilidade e extensibilidade. O PuzzlEdu está hospedado na infraestrutura da Google, no *link*: <http://testpuzzledu.appspot.com/>. Além disso, pode-se visualizar o projeto no Google Code, no endereço de acesso público: <http://code.google.com/p/puzzledu/>.

A escolha de qual ferramenta construir com essa finalidade levou em consideração a área de informática, escolhendo algum desafio intrínseco a ela. Na maioria dos cursos de graduação em informática, em suas matrizes curriculares, há disciplinas consideradas base para o ensino da programação, como Algoritmos e Lógica de Programação, disciplinas que exploram os princípios básicos da lógica de programação objetivando desenvolver a capacidade de análise e resolução de problemas na forma de algoritmos [3]. Infelizmente essas mesmas disciplinas possuem um alto índice de reprovação e evasão [4], [5], [6], [7], e seu ensino é considerado como um dos sete maiores desafios do século [8].

Assim, outro objetivo do PuzzlEdu é auxiliar no processo de ensino e aprendizagem da disciplina de Linguagem de Programação, mais especificamente os conceitos do paradigma orientado a objetos. Essa especificidade se deve pelo fato da Programação Orientada a Objetos (POO) ser muito difundida na academia e na indústria, tendo sido evidenciada pela Sociedade Brasileira de Computação (SBC) como um dos principais tópicos dentre as disciplinas de linguagens de programação [9].

WebMedia'11: Proceedings of the 17th Brazilian Symposium on Multimedia and the Web. X Workshop on Tools and Applications.

October 3 -6, 2011, Florianópolis, SC, Brazil.

ISSN 2175-9650.

SBC - Brazilian Computer Society

3. ARQUITETURA E IMPLEMENTAÇÃO

O PuzzlEdu foi construído utilizando a arquitetura em 3 (três) camadas (apresentação, negócio e persistência), proporcionando maior flexibilidade ao sistema e baixo acoplamento. A representação gráfica da arquitetura do PuzzlEdu é apresentada na Figura 1.

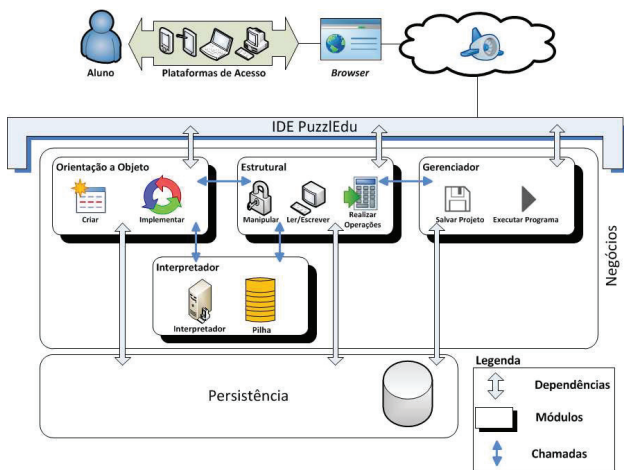


Figura 1: Representação gráfica da arquitetura do PuzzlEdu.

O PuzzlEdu possui na sua camada de apresentação a infraestrutura necessária para que os usuários executem a construção de seus programas, ou seja, é a IDE PuzzlEdu de desenvolvimento. Essa camada possui dependências com os módulos da camada de negócios (ou implementação), no caso, módulos de orientação a objeto, estrutural, gerenciador e interpretador que, interligados, realizam toda a lógica de negócios do projeto (ação dos métodos, operações, manipulações e execução). Na camada de negócios os módulos comunicam-se para a integração das ações e atividades a serem descritas pelos programas montados na camada anterior. Através de um interpretador e uma pilha de comandos, o projeto pode então ser executado da forma como foi programado.

Por ser um serviço, inicialmente não há a necessidade de armazenamento dos dados, sendo tudo processado na nuvem computacional. Entretanto, o usuário da ferramenta pode optar por salvar seus projetos para implementação ou melhoramento futuro e, por isso, a camada de persistência para guardar informações como usuário conectado (com uma conta da Google) e os componentes utilizados durante o projeto. Toda essa infraestrutura utiliza-se de diversas tecnologias, todas tratadas na próxima seção.

3.1 Infraestrutura

A construção de todo o projeto do PuzzlEdu envolveu diversas tecnologias, desde a parte de modelagem até a parte de desenvolvimento e controle de versão. Por ser concebido com fortes perspectivas de crescimento e continuidade, todas as soluções de tecnologias utilizadas são as mais atuais possíveis. Ainda, por buscar uma maior abrangência de utilização e distribuição, o PuzzlEdu é uma ferramenta de código aberto, ou seja, disponível para toda a comunidade, através de licença GPL (*General Public License*), utilizando-se também de tecnologias com licenças semelhantes, de maneira a não proibir ou coibir sua utilização. Espera-se para o futuro contribuir ainda mais para comunidade, passando a disponibilizá-lo como um *software*

público brasileiro. As ferramentas utilizadas no desenvolvimento do PuzzlEdu são brevemente explicitadas a seguir.

- **IDE Eclipse:** projeto de código aberto e gratuito, escrito em Java e desenvolvido pela IBM, com rápida aceitação entre a comunidade de desenvolvedores de sistemas, sendo um dos IDEs mais utilizados para desenvolvimento de *software* para as mais variadas linguagens, haja vista empregar *plug-ins* com tal suporte e ainda fornecer um forte apoio para o desenvolvimento baseado em Java [10]. Sua escolha foi influenciada pelos *plug-ins* disponíveis da Google para essa IDE, de forma a facilitar o desenvolvimento de aplicações para uso na sua nuvem computacional;

- **Linguagem Java:** linguagem de programação orientada a objetos desenvolvida pela Sun Microsystems e bastante difundida ao redor do mundo. Sua escolha foi influenciada principalmente pela Plataforma como Serviço (PaaS) escolhida para implantação do PuzzlEdu, neste caso o Google App Engine;

- **Google Web Toolkit (GWT):** é um conjunto completo de ferramentas (*framework*) para desenvolvimento de complexas aplicações Web baseadas em AJAX usando Java [10]. É gratuito e possui *plug-in* para uso dentro do Eclipse. A utilização do GWT facilita o desenvolvimento de aplicações para Web, plataforma base do PuzzlEdu;

- **SmartGWT:** é um *framework* baseado em GWT com diversas bibliotecas de *widgets* disponíveis para desenvolvimento das interfaces com o usuário, facilmente integrável aos IDEs de desenvolvimento, principalmente que utilizam o GWT, como o Eclipse;

- **Google App Engine (GAE):** é uma plataforma e um SDK (*Software Development Kit*) para desenvolver e hospedar aplicações Web usando toda a infraestrutura e servidores da Google, suportando os ambientes de execução e SDKs das linguagens Java e Python [10]. Sua escolha levou em consideração a oferta de cotas gratuitas generosas de espaço e processamento dentro da nuvem computacional da Google, além da facilidade de integração a partir de um *plug-in* próprio para o Eclipse, neste caso, com o GWT;

- **Bigtable:** desenvolvido pela Google, é um sistema de armazenamento distribuído para gerenciamento de dados estruturados de larga escala e alto desempenho [11];

- **MagicDraw:** para a modelagem dos casos de uso foi utilizado o MagicDraw, uma ferramenta para modelagem visual UML. Apesar de não ser gratuito, possui licença disponível somente para uso acadêmico e não comercial;

- **Microsoft Visio:** utilizado para construção da arquitetura do PuzzlEdu, o Visio é um *software* para construção de diagramas técnicos e profissionais, de fácil manipulação, com diversos recursos para desenho. A versão utilizada foi de avaliação, ou seja, sua licença não é gratuita, porém, assim como o MagicDraw, seu uso não é impeditivo para liberação do *software*, tendo sido utilizado apenas para desenho da arquitetura;

- **Subversion:** tecnologia para controle de versão, gratuito e de código aberto, feito para gerenciar arquivos e diretórios, permitindo recuperar e verificar todas as versões ao longo do desenvolvimento [12].

4. PRINCIPAIS FUNCIONALIDADES

Ao acessar o endereço onde o PuzzlEdu está hospedado, o usuário depara-se com a tela inicial exibida na Figura 2. O PuzzlEdu

possui um conjunto de janelas, áreas e botões específicos, com objetivos distintos. Para melhor entendimento, esse conjunto foi numerado e descrito a seguir:

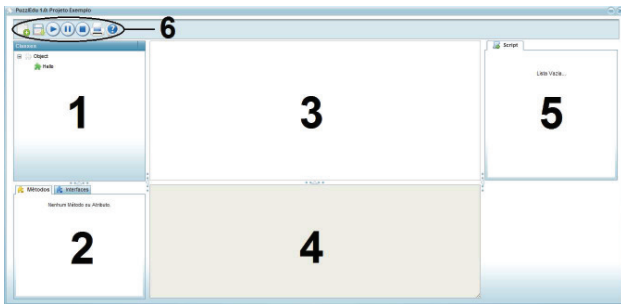


Figura 2: Tela inicial do PuzzlEdu.

1. Classes: janela onde as classes são criadas e manipuladas. Na execução inicial do PuzzlEdu (ou quando a página é atualizada), verifica-se a existência de uma classe nomeada como *HelloWorld*. O motivo disso é proporcionar ao usuário a familiarização com a ferramenta e inicializá-lo nos conceitos propostos. Nota-se também uma classe em um nível superior nomeada como *Object*. A classe *Object* é, em muitas linguagens, a classe no topo da hierarquia de classes. Todas as classes herdam direta ou indiretamente dela. Isso possibilita ao professor introduzir os primeiros conceitos de herança, um dos fundamentos básicos da orientação a objeto;

2. Métodos/Interface: guia com as opções *Métodos* e *Interfaces*. Nela ficam os atributos e métodos da classe selecionada na janela *Classes*. Nesta área só é possível remover ou alterar os atributos e métodos. Na guia *Interfaces* encontram-se as interfaces que a classe selecionada está implementando, possibilitando ao professor introduzir conceitos de Interfaces aos alunos. Nesta área há possibilidade apenas de remover a interface associada à classe selecionada;

3. Execução/Manipulação: essa é a área onde as classes serão manipuladas, ou seja, é o corpo do programa ou projeto a ser desenvolvido. Dependendo da manipulação realizada, essa é também a área de execução, como por exemplo, fazer um objeto mover-se de um lado a outro;

4. Console: área na qual alguns resultados são exibidos durante a execução dos programas construídos na área Execução/Manipulação;

5. Script: guia onde será exibida a pilha de comandos de manipulação realizada pelo usuário. O PuzzlEdu utiliza a estratégia de execução de comandos através de *scripts*, ou seja, o usuário faz a manipulação desejada sobre um objeto (ou instância) em particular, e cada manipulação realizada vai sendo empilhada nessa guia, sendo executada posteriormente após clique no botão executar;

6. Botões de comandos: são sete botões com ações distintas. Criar novos e salvar projetos, executar a pilha de comandos dos *scripts*, pausar ou parar (cancelar) uma execução em andamento, imprimir o código gerado e acessar a documentação.

As principais funcionalidades do sistema estão ligadas diretamente ao processo de criação de um programa orientado a objetos, ou seja, a criação de classes, métodos, atributos e instâncias (objetos) da classe.

5. CONCLUSÕES

Neste artigo apresentamos a ferramenta PuzzlEdu, que cumpre o papel para o qual foi desenvolvida e utiliza os principais conceitos

e infraestrutura das nuvens computacionais, tendo sido disponibilizada como serviço à comunidade, sendo prática, desenvolvida sob a licença GPL (*General Public License*), sem limitações de uso ou limitações tecnológicas, haja vista estar utilizando-se de uma infraestrutura elástica e global. PuzzlEdu promove a EdaaS como um insumo benéfico, eficiente e construtivo.

6. REFERÊNCIAS

- [1] Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. 2009. *Above the Clouds: A Berkeley View of Cloud Computing*. Technical Report. University of California at Berkeley;
- [2] Robert Fogel. 2010. *The Education Cloud: Delivering Education as a Service*. Whitepaper. 2010. In: <http://www.intel.com/assets/pdf/whitepaper/delivering-education-as-a-service.pdf>. Access: 12/2010.
- [3] Rapkiewicz, C. E., Falkembach, G., Seixas, L., Rosa, N. S., Cunha, V. V. and Klemann, M. 2006. *Estratégias Pedagógicas no Ensino de Algoritmos e Programação Associadas ao Uso de Jogos Educacionais*. Renote - Novas Tecnologias na Educação, v. 4, n. 2, p. 1-11.
- [4] Silva, H. V. D. R. C. 1991. *Representações Computacionais Auxiliares ao Entendimento de Conceitos de Programação*. Tese (Doutorado em Engenharia Elétrica). Universidade Estadual de Campinas (Unicamp). Campinas, p. 461.
- [5] Santos, R. P. D. and Costa, H. A. X. 2006. *Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos iniciantes em Computação e Informática*. INFOCOMP - Journal of Computer Science, Lavras, 5, n. 1, p. 41-50.
- [6] Hinterholz Júnior, O. 2009. Tepequém: uma nova Ferramenta para o Ensino de Algoritmos nos Cursos Superiores em Computação. In: *XXIX Congresso da Sociedade Brasileira de Computação (CSBC)*. Bento Gonçalves.
- [7] Rívollí, A., Guimarães, M. D. P. and Moreira, D. D. A. 2010. Uma Estratégia Colaborativa para Complementar o Ensino e Aprendizagem. In: *VII Simpósio Brasileiro de Sistemas Colaborativos (SBSC)*. Belo Horizonte.
- [8] McGettrick, A., Boyle, R., Ibbett, R., Lloyd, J., Lovegrove, G. and Mander, K. 2004. *Grand Challenges in Computing - Education*. The British Computer Society (BCS), v. 4.
- [9] Chaves, R. O., Tavares, E. M. C., Rodrigues, I. C., Oliveira, S. R. B. and Favero, E. L. 2010. A Contribuição da Filosofia de Platão e de Aristóteles para o Ensino do Paradigma de Orientação a Objetos. In: *XXI Simpósio Brasileiro de Informática na Educação (SBIE)*. João Pessoa.
- [10] Guerneur, D. and Unruh, A. 2010. *Google App Engine Java and GWT Application Development*. Packt Publishing.
- [11] Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A. and Gruber, R. E. 2008. *Bigtable: A Distributed Storage System for Structured Data*. ACM T COMPUT SYST (TOCS), New York, 26, n. 2. DOI = <http://dx.doi.org/10.1145/1365815.1365816>.
- [12] Collins-Sussman, B., Fitzpatrick, B. W. and Pilato, C. M. 2004. *Version Control with Subversion*. O'Reilly Media.