

Cosmo: Um ambiente virtual de aprendizado com foco no Ensino de Algoritmos

Dilson José Lins Rabêlo Júnior¹, Carlos de Salles Soares Neto¹, Antonio Carlos Raposo¹, Luis Alves dos Santos Neto¹

¹ Av. dos Portugueses, 1966 - Vila Bacanga, São Luís - MA, 65065-545

{dilsonjlrjr, csallesneto}@gmail.com

{antoniocarlostraposo93, alvesgamadossantos}@gmail.com

Resumo. *Um dos principais problemas enfrentados pelos cursos de graduação na área de informática é a alta taxa de desistência prematura. Deve-se então considerar o papel crítico da disciplina de algoritmos, já que tal disciplina é o primeiro contato que o discente possui com programação de computadores. Tornar o ensino de programação mais prático é um desafio e o processo de ensino-aprendizagem tem contado com o auxílio da tecnologia para construções de ferramentas. O Cosmo é um ambiente virtual de aprendizado, multitarefa e extensível a plugins, focado em atividades voltada ao estudo da disciplina de algoritmos. A proposta deste trabalho se insere no contexto do projeto do Cosmo, apresentando sua arquitetura e funcionalidades, como elas foram concebidas e estruturadas.*

1. Introdução

Nos cursos de graduação na área de informática, a disciplina de algoritmos tem papel fundamental. Isso porque, é através desta que a maioria dos alunos tem o primeiro contato com a programação de computadores. A disciplina de algoritmos ensina a aplicação do raciocínio lógico para resolução de problemas e sua descrição na forma de algoritmos. Apesar da sua importância, essa disciplina apresenta altos índices de reprovação e desistência. Cursos como Sistema de Informação e Ciência da Computação apresentam as maiores taxas de evasão dentre os cursos de educação superior no Brasil [Carvalho and Tafner 2006]. Segundo [Palmeira and Santos 2014], cursos de Ciência da Computação possuem uma taxa de ingressantes e concluintes de 14,3%. [dos Santos and Costa 2006] afirma que um dos motivos para desistência nos cursos é a dificuldade em aprender os conceitos básicos de programação.

Alguns fatores que causam a desmotivação no aprendizado de algoritmos é a dificuldade dos alunos em desenvolver raciocínio lógico, em conjunto da dificuldade em compreender as abstrações envolvidas no processo de programação e também a abordagem pedagógica [Rapkiewicz et al. 2007]. Diante disso, faz-se necessário buscar meios que tornem o ensino de algoritmos mais prático e atrativo para esse público.

A utilização de ambientes virtuais de aprendizagem no suporte ao ensino vem ganhando espaço no contexto educacional como ferramenta de apoio, devido a sua eficiência junto aos diferentes perfis de estudantes existentes. No ambiente virtual de aprendizagem (AVA), o processo de ensino-aprendizado pode ser mais dinâmico e personalizado.

Assim, em um mesmo AVA, podem ser oferecidos aos alunos diferentes mídias e atividades para auxiliá-los na assimilação do conteúdo. Essa possibilidade do aluno aprender sob diferentes formas é bem vista levando em consideração a existência de diferentes estilos de aprendizagem. Potencializando a oferta de um ensino cada vez mais eficaz, dado que o conteúdo da disciplina passa a ser adaptado ao estilo de aprendizado de cada discente.

Diante disso, uma forma de auxiliar o aprendizado dos alunos, é construir um ambiente que o permita conseguir estudar de acordo com suas características e preferências. Assim, é apresentado o Cosmo um ambiente virtual de aprendizado, focado em atividades voltada ao estudo da disciplina de algoritmos. O ambiente é uma plataforma multitarefa extensível a *plugins* que disponibiliza ao aluno uma experiência de aprendizado diversificada.

O artigo está organizado como segue. A Seção 2 descreve os trabalhos relacionados. A Seção 3 descreve o ambiente Cosmo. A Seção 4 descreve os testes de uso e resultados obtidos por uso do ambiente. A Seção 5 descreve os trabalhos futuros. Por fim, na Seção 6 descreve as considerações finais.

2. Trabalhos Relacionados

Várias pesquisadores estão dedicando esforços para desenvolver métodos e softwares educacionais para apoiar o aprendizado do ensino de algoritmos. Durante a pesquisa pode-se perceber que existe uma forte tendência no uso de jogos para como apoio ao ensino e aprendizado de algoritmos.

Professores e alunos do CSET (Ciência da Computação, Engenharia e Tecnologia) ressaltaram em seu trabalho [de Oliveira Brandão et al. 2012], a mesma problemática destacada na introdução deste artigo, sobre o processo de ensino e aprendizagem de algoritmos a alunos iniciantes nos cursos de computação. Como solução, eles apresentam o iVProg, uma aplicação derivada de um outro projeto chamado Alice [Conway and Pausch 1997], este faz uso da programação visual, no qual possibilita o aluno criar funções e métodos para construção de um programa com um simples arrastar do mouse. A ferramenta já traz consigo, elementos pré-definidos como comparação de variáveis, funções matemáticas e elementos básicos da programação principal como *if-then-else* e *loops*.

O trabalho desenvolvido por [Topalli and Cagiltay 2018], apresenta o uso da ferramenta *Scratch* sendo utilizada nas disciplinas de programação no curso de engenharia. Os estudantes foram acompanhados durante todo o curso. Nesse tempo, diversos projetos eram entregues aos alunos e o seu desenvolvimento deveria ser feito com auxílio do *Scratch*. Todo o progresso dos estudantes foi devidamente registrado e ao final da pesquisa os dados tabulados mostraram que o uso da ferramenta, apoiadas a outras metodologias, podem ajudar potencialmente no andamento do curso.

No artigo [Tillmann et al. 2014] apresenta o Code Hunt, um jogo educacional estilo caça código. O jogador, ou caçador de código, busca fragmentos que faltam para completar, ou desenvolver uma solução mais elegante para um código desafio. O jogador tem possibilidade de escrever os códigos em C# ou Java, o que exige dele um conhecimento básico ou mediano em uma das linguagens. O fluxo de execução do jogo, se inicia

quando o jogador se inscreve no site, logo em seguida é levado para o processo de seleção da linguagem e por último uma lista de fases são apresentadas, sendo inicialmente disponível somente a fase de treinamento. Cada fase aborda tópicos específicos sobre lógica de programação, como exemplo: estruturas de repetição, vetores, ordenação e etc.

Um jogo sério amplamente utilizado no ensino introdutório de programação de computadores pode ser visto no artigo do [Kazimoglu et al. 2012]. O Lightbot é um jogo com temática lúdica, ensina lógica de programação através comandos estruturados em blocos. Foi concebido com o intuito de capacitar alunos iniciantes e a lógica de programação. O seu ambiente aborda tópicos introdutórios de lógica, tais como a construção de algoritmo, depuração e simulação. O objetivo do jogo é fazer com o robzinho acendas as luzes durante um percurso e escape de obstáculos que possa encontrar no caminho. Sua mecânica está organizada em organizar os blocos de movimentos para que o robzinho possa se movimentar e acender as luzes durante o caminho.

A proposta do Cosmo se diferencia das demais, pois espera-se integrar os diversos tipos de atividades em um único ambiente. O modelo de agregar essa diversidade de atividades se objetiva em melhorar a experiência do usuário no uso da aplicação. Com modelo extensível a *plugins* é admissível o uso de qualquer modelo de atividade dentro do ambiente.

3. O Cosmo

Cosmo é um ambiente virtual de aprendizado, multitarefa e extensível a plugins, focado em atividades voltada ao estudo da disciplina de algoritmos.

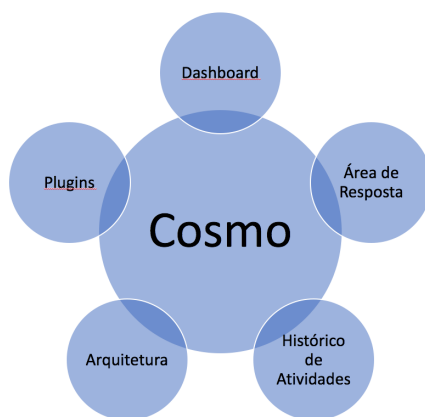


Figura 1. Apresentação do Cosmo

Assim como listado na Figura 1, o Cosmo pode ser apresentado descrevendo os cinco itens na figura. Cada um dos itens será detalhado nas sub-seções a seguir.

3.1. Área de Dashboard

Ainda em sua versão *beta*, o Cosmo tem sua interface subdividida em duas áreas. A primeira corresponde a chamada de área *Dashboard*, nela o aluno poderá observar todas as atividades disponíveis para resolução. A segunda corresponde ao **Histórico de atividades**, nela são listadas todas as atividades já respondidas pelo aluno. A Figura 2 apresenta a área de *Dashboard* da plataforma. No item 1 refere-se aos caixas de atividades, o item

2 a Ação de pular, o item 3 o Perfil do usuário, item 4 a Ação de responder e o item 5 o Menu do Usuário.

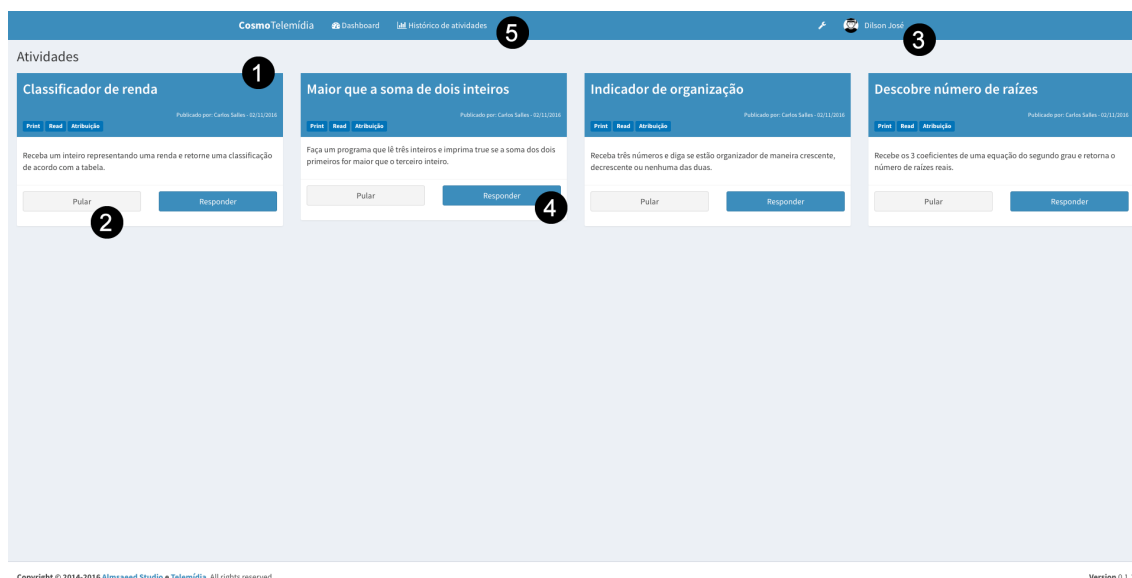


Figura 2. Dashboard do Cosmo

A composição das atividades no *Dashboard*, se dá de forma randômica. Cada caixa de atividade possui um título, um breve conceito para expor previamente o tipo de atividade, quem a criou e *tags*, elas indicam ao usuário de qual assunto que a questão está abordando.

3.2. Área de Resposta

Ao clicar no botão "Responder", em uma caixa de atividade, o usuário é levado a uma tela chamada de Área de resposta, conforme apresentado na Figura 3. O conteúdo que o usuário visualiza nessa tela é dinâmico, ou seja, deriva do *plugin* que está sendo instanciado no momento. Na Figura 3, vemos o item 1 que corresponde o enunciado da questão e em seguida o item 2 que apresenta a forma como o *plugin* aceita a entrada de resposta para a atividade. Neste exemplo pode ser observado o plugin "Resolver Problemas".

3.3. Arquitetura

A arquitetura do Cosmo segue o modelo Cliente-Servidor. A Figura 4 apresenta a arquitetura do Cosmo.

No seu *Front-End* foram utilizados as tecnologias HTML5, CSS3 e Javascript. Sua interface foi desenhada com auxílio do AdminLTE ¹ e suporte do JQuery ². Em seu *Back-End*, o Cosmo foi escrito com auxílio da linguagem de programação PHP ³.

3.4. Plugins

O Cosmo é um ambiente extensível a *plugins*, ele tem a capacidade de incorporar qualquer tipo de atividade que seja renderizada pelo HTML5. Isso possibilita que seja possível

¹<https://adminlte.io/themes/AdminLTE/index2.html>

²<https://jquery.com/>

³<http://www.php.net/>

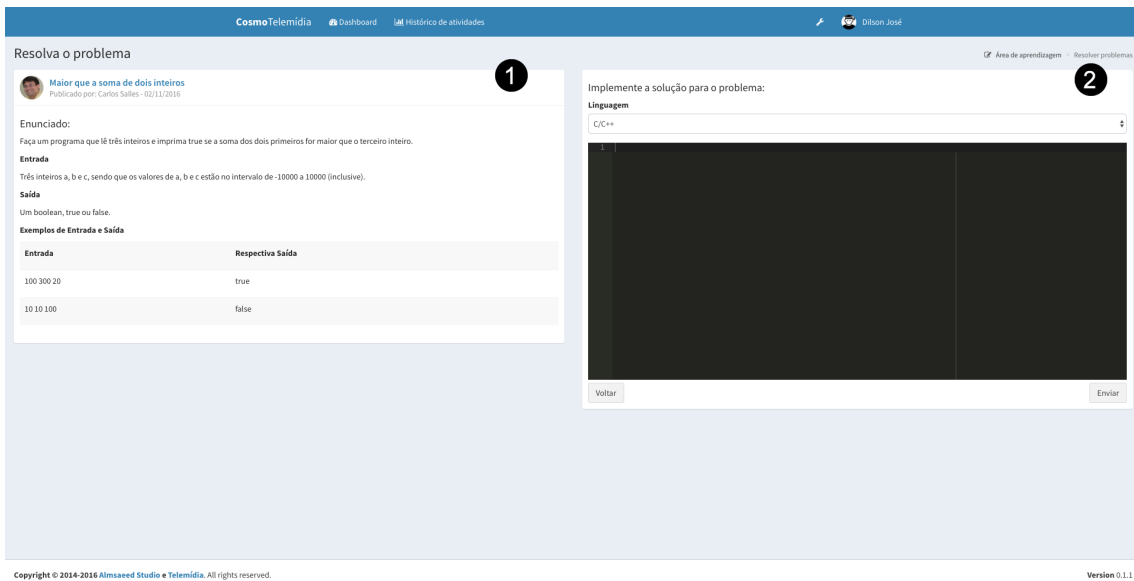


Figura 3. Área de Resposta

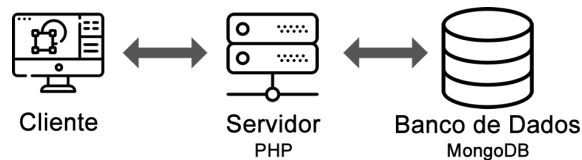


Figura 4. Arquitetura do Cosmo

utilizar um jogo como atividade, um vídeo direto de um provedor de *stream*, a leitura de um documento, as possibilidades são inúmeras.

A Figura 5 apresenta a estrutura de *plugin* do Cosmo.

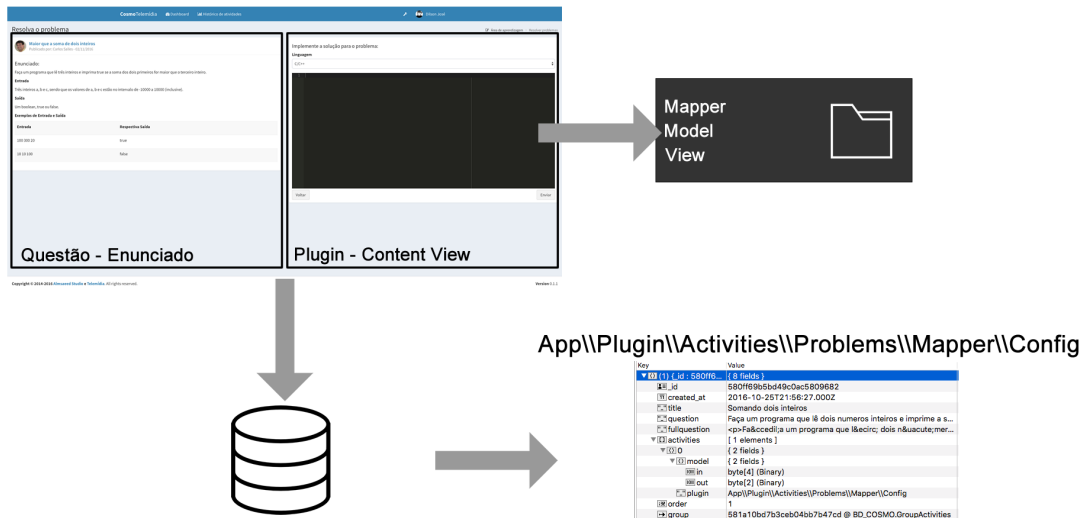


Figura 5. Estrutura do Plugin

A Área de Resposta, já apresentada anteriormente, é onde o *plugin* é instanciado e exibido. Ao ser instanciado o Cosmo monta a área de Enunciado e o *Content View*.

Cada *plugin* possui uma estrutura isolada do núcleo do ambiente e todo acesso ao núcleo é feito através de *interfaces*. Basicamente um *plugin* possui um estrutura de 3 pastas, a *Model*, a *View* e a *Mapper*.

O diretório *Model*, armazena todos os elementos que possam auxiliar a execução da tarefa. Nela pode ser aplicado elementos que não parte da estrutura do Cosmo.

O diretório *View*, armazena o HTML que será apresentado no *Content View*. Para auxiliar o desenvolvimento é utilizado um *template engine*, o Twig⁴.

A *Mapper*, é responsável por mapear as configurações do tipo de tarefa para a estrutura de dados do Cosmo, validar a atividade, integrar a tarefa ao banco de dados, especificar a *view* e definir o modelo da tarefa.

4. Testes e Resultados

O Cosmo atualmente se encontra na versão *beta* e com a um tipo de tarefa disponível. Com essa versão da aplicação, foi preparado um teste preliminar com uma turma do 2º período da disciplina de algoritmos. Os testes foram realizados uma semana antes da primeira prova da disciplina. Estava previsto a presença de 56 membros, sendo confirmados 34 presenças e 22 ausências. O período de monitoramento da ferramenta ocorreu no prazo de duas semanas, nesse período, o sistema estava armazenando dados como tempo de uso da ferramenta em horas e dias, número de questões respondidas e o tempo médio de uso da ferramenta.

A base de dados do ambiente continha um total de 22 tarefas do tipo "Resolver o Problema".

Esse tipo de tarefa o aluno recebe um enunciado o tipo de entrada de dados e sua respectiva saída, com base nesses dados o aluno precisa desenvolver o código que atenda a bateria de testes contida dentro do Cosmo. Esse tipo de atividade oferece o suporte as linguagens Lua, C/C++ e Python.

Todas as tarefas foram elaboradas por um professor especialista, seguindo a temática da prova e classificadas em uma escala de Fácil Médio e Difícil.

A análise dos dados de respostas dos alunos foram cruzados com os resultados da prova, a partir delas pode-se concluir que: 7,14% dos alunos que participaram do teste, são os alunos que tiveram o maior tempo de uso, maior número de questões respondidas (entre 13 e 17 questões) e obtiveram na prova notas acima de 9,5. Outra porção dos alunos, cerca de 12,5%, foram os alunos que passaram em média 124 minutos de uso da ferramenta, responderam entre 10 e 12 questões e obtiveram notas entre 4,5 e 10. Os demais alunos da amostra, especificamente 80,36% foram caracterizados dados de corte da amostra.

Notou-se que, ao que se propõe, o Cosmo cumpriu com as expectativas, pois contribuiu no aprendizado dos alunos no período próximo a prova da disciplina de algoritmos. A assertiva foi coletada a partir de um questionário expectativa aplicado após a prova, ava-

⁴<https://twig.symfony.com/>

liando o uso do Cosmo para auxílio da disciplina de algoritmos. Mais de 55% dos alunos responderam que sim o ambiente pode auxiliá-los a fixar o conteúdo da disciplina de algoritmos.

5. Trabalhos Futuros

Os primeiros testes do Cosmo foram satisfatórios pelo que o ambiente propõe. Durante os testes pode-se perceber que algumas funcionalidades precisam ser revisadas, como exemplo o *Log* da aplicação que precisa ser refinado.

Novos requisitos foram levantados durante os testes, a partir dos questionários de expectativas. Novos modelos de atividades foram propostas, pedido de tutoriais, área para reportar *bugs*, gamificação (já previsto pela equipe), dentre outros requisitos.

Na próxima versão do Cosmo serão incorporadas novas atividades e uma versão beta de um algoritmo de tutoria esta sendo desenvolvido para recomendação de atividades ao aluno.

6. Conclusão

É evidente que um dos principais problemas enfrentados pelos cursos de graduação na área de informática é a alta taxa de desistência. Deve-se então considerar o papel crítico da disciplina de algoritmos. Tal disciplina representa para um grande grupo de estudantes seu primeiro contato com programação de computadores. Experiências negativas podem desencorajar os estudantes de um estudo mais aprofundado e afetar seriamente a sua motivação.

Buscar alternativas para reduzir as taxas de abandono é tornar o ensino de programação mais prático e concreto. Essa é a proposta apresentada pelo ambiente Cosmo, um ambiente virtual multitarefa focado para o ensino de algoritmos. Com a capacidade de incorporar novos *pluginse* permitir que seja possível diversificar a experiência do usuário no processo de ensino-aprendizagem.

Referências

- Carvalho, M. d. and Tafner, P. (2006). Ensino superior brasileiro: a evasão dos alunos e a relação entre formação e profissão. *Anais do 30º Encontro anual da ANPOCS*.
- Conway, M. J. and Pausch, R. (1997). Alice: easy to learn interactive 3d graphics. *ACM SIGGRAPH Computer Graphics*, 31(3):58–59.
- de Oliveira Brandão, L., da Silva Ribeiro, R., and Brandão, A. A. (2012). A system to help teaching and learning algorithms. In *Frontiers in Education Conference (FIE), 2012*, pages 1–6. IEEE.
- dos Santos, R. P. 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*, 5(1):41–50.
- Kazimoglu, C., Kiernan, M., Bacon, L., and Mackinnon, L. (2012). A serious game for developing computational thinking and learning introductory computer programming. *Procedia-Social and Behavioral Sciences*, 47:1991–1999.

- Palmeira, L. B. and Santos, M. P. (2014). Evasão no bacharelado em ciência da computação da universidade de Brasília: análise e mineração de dados. *Monografia do curso de Ciência da Computação da UnB—Universidade de Brasília. Brasília: UnB.*
- Rapkiewicz, C. E., Falkembach, G. A. M., Seixas, L. M. J. d., Santos, N. d. S. R. S. d., Cunha, V. V. d., Klemann, M., et al. (2007). Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais. *RENOTE: revista novas tecnologias na educação [recurso eletrônico]. Porto Alegre, RS.*
- Tillmann, N., Bishop, J., Horspool, N., Perelman, D., and Xie, T. (2014). Code hunt: searching for secret code for fun. In *Proceedings of the 7th International Workshop on Search-Based Software Testing*, pages 23–26. ACM.
- Topalli, D. and Cagiltay, N. E. (2018). Improving programming skills in engineering education through problem-based game projects with scratch. *Computers & Education*, 120:64–74.