AVAA: Ambiente Virtual de Aprendizagem de Algoritmos

Bruno Gonzaga de Mattos Vogel¹; Rafael da Silva Macêdo² e Claudia Pinto Pereira³

1. Bolsista PROBIC/UEFS, Graduando em Engenharia de Computação, Universidade Estadual de Feira de Santana, e-mail: bugvogel@gmail.com
2. Participante do projeto, Graduando em Engenharia de Computação, Universidade Estadual de Feira de Santana, e-mail: 3011rafael@gmail.com
3. Orientador, Departamento de Ciências Exatas, Universidade Estadual de Feira de Santana, e-mail: claudiap@uefs.br

PALAVRAS-CHAVE: Objetos de aprendizagem, Algoritmos, Educação.

INTRODUÇÃO

Nos estudos realizados por Silva e Sena (2013), observou-se que a maioria dos estudantes de engenharia de computação, sistema de informação ou ciência da computação, que se depara com a necessidade de aprender algoritmos, acaba por ter conhecimento das estruturas de funcionamento (sintaxe), mas apresentam alguma dificuldade na lógica de relacionamento entre elas (semântica). Muitas vezes, tal circunstância é dada porque, para se adquirir a habilidade de programação em computadores, é necessário um significativo esforço em atividades práticas de laboratório (PRIOR, 2003). De tal maneira, um ambiente virtual de aprendizagem de algoritmos (AVAA) poderia acrescentar funcionalidades associadas ao aprendizado de algoritmos, assim como exercícios práticos e resolução dos mesmos (ROMERO, VENTURA e GARCIA, 2008).

Diante deste contexto, este trabalho teve como objetivo desenvolver uma ferramenta voltada ao processo de ensino e aprendizagem de algoritmo e programação, levando em conta, portanto, os aspectos citados anteriormente. Em meio a esta problemática encontrada, o AVAA possibilita a garantia do acompanhamento dos estudantes das turmas pelo professor, para assegurar que as necessidades cognitivas para construir um algoritmo sejam internalizadas.

As funcionalidades levantadas como requisitos e desenvolvidas durante o período de um ano, portanto, estão voltadas a sanar essas dificuldades, com um sistema autônomo na correção de exercícios de algoritmo, dividido em três níveis de dificuldade, com correções instantâneas nos dois primeiros.

A ferramenta desenvolvida busca reduzir os altos índices de reprovação em disciplinas de programação e algoritmo de cursos superiores. Como se trata de um sistema *web*, apenas com acesso à internet o usuário pode se conectar ao seu painel, consultar seu desempenho nos exercícios, avisos do professor e exercícios a serem resolvidos. O AVAA em questão não substitui de fato o aprendizado presencial de uma disciplina comum, porém, apesar de muito utilizados no ensino a distância, estes mostram sua eficácia quando aplicadas ao próprio ensino presencial (FRANCO et al., 2003).

MATERIAL E MÉTODOS OU METODOLOGIA (ou equivalente)

Foram utilizadas linguagens de programação *web* e um *framework* para auxiliar o desenvolvimento. O levantamento e a escolha do *framework*, da linguagem e outras tecnologias também foram parte deste trabalho de pesquisa.

A execução do trabalho seguiu as seguintes sistemática e etapas:

- **Revisão bibliográfica**

Aproximação com a temática do trabalho, ou seja, realização de pesquisas e leituras sobre tecnologias para desenvolvimento *web* e ensino-aprendizagem de algoritmos e programação. Além disso, foram realizadas análises funcionais de softwares semelhantes já existentes, além do proposto por Silva e Sena (2013), e uma avaliação da relevância de tais no processo educacional de algoritmos.

- **Preparação do ambiente de programação**

Esta etapa compreendeu a instalação de softwares necessários para o desenvolvimento do sistema. Além disso, foi utilizado um servidor local *web* para a realização de testes durante o desenvolvimento do sistema.

- **Implementação e realização de testes**

Nesta etapa, foram implementadas as funcionalidades do sistema, tais como: sistema de administração pelo professor, cadastro de turma, cadastro de alunos, cadastro de atividades, realização de atividades, avaliação por turma, avaliação por aluno, dentre outras. Ao final de cada implementação ou tela instituída, foram realizados testes funcionais, a fim de atestar o correto funcionamento da ferramenta construída.

RESULTADOS E/OU DISCUSSÃO (ou Análise e discussão dos resultados)

O projeto foi desenvolvido ao longo de um ano na Universidade Estadual de Feira de Santana por dois bolsistas de iniciação científica, devido à necessidade de substituição durante o período da bolsa de estudos. A construção da ferramenta foi dividida em três partes, sendo a primeira delas a pesquisa teórica a respeito das tecnologias necessárias para uma aplicação *web*, e pesquisa de ambientes virtuais de aprendizagem para um melhor embasamento do seu funcionamento (SILVA e SENA 2013; MOODLE, 2017).

Na segunda etapa, foi realizada a modelação do software com a construção de diagramas de atividades dos módulos de estudante e professor e modelação do banco de dados, para que, em seguida, fossem escolhidas as tecnologias que seriam utilizadas na terceira etapa, com a efetiva implementação do que foi anteriormente estudado e desenvolvido. Na terceira etapa, utilizou-se a linguagem de programação PHP *Hypertext Preprocessor* (PHP, 2017) como a principal tecnologia. Além desta, também foram utilizadas a linguagem de programação *JavaScript* (ECMA, 2017), linguagem de marcação HTML (*HyperText Markup Language*), CSS (*Cascading Style Sheet*) (W3C, 2017), como mecanismo de estilização dos componentes visuais, e *MySQL* (MYSQL, 2017), como sistema de gerenciamento do banco de dados. Também se utilizou o *Bootstrap* (GETBOOTSTRAP, 2017), como *framework front-end* do projeto.

O processo de implementação das funcionalidades se iniciou com a construção das telas e, em seguida, foram elaboradas as seções de códigos referentes à efetiva lógica da funcionalidade, inclusive com as instruções de manipulação do banco de dados. As primeiras funcionalidades montadas foram os requisitos funcionais de cadastro de professor, cadastro de estudante e realização de *login*. Já os requisitos de cadastro de turma e inserção de estudantes na mesma foram desenvolvidos em sequência, devido a sua correlação e dependência. Como o professor pode fazer parte de várias instituições diferentes, o cadastro de turmas está sempre vinculado à instituição e ao nome da turma para evitar confusões no sistema.

Os requisitos seguintes podem ser visualizados nas Figuras 1,2 e 3, que mostram o comportamento do sistema no acompanhamento do professor e aluno, e

consequentemente a UX (*User experience*) causada aos usuários e UI (*User Interface*) da ferramenta.

Figura 1: Cadastro de avisos do professor

Na Figura 1, o professor tem a capacidade de cadastrar avisos para a turma e receber a garantia de envio com sucesso, por meio da exposição dos avisos já cadastrados. Esta figura engloba o funcionamento dos requisitos de adição de aviso e visualização de avisos. Para a amostragem de resolução e cadastro de exercícios, pode ser utilizada a Figura 2 que ilustra a tela de resolução de um exercício do nível 1 para um aluno. Nesta tela, o aluno deve ordenar os blocos de entrada, processamento e saída destes dados, com resposta automática do sistema. Esta característica se aplica também ao segundo nível de resolução de exercícios, contudo, neste é necessária a ordenação de blocos condicionais e de repetição do algoritmo cadastrado.

Figura 2: Resolução de exercício do nível 1

No nível três de resolução de exercícios, não ocorre a ordenação de blocos, apenas a correção semântica e de sintaxe, pelo estudante, de um algoritmo cadastrado. Este se trata do único nível de exercício no qual é necessária a correção do docente da turma, devido à ausência de um interpretador léxico, semântico e de sintaxe para o sistema de uma linguagem específica.

Figura 3: Correção de exercícios do nível 3

Os testes no sistema foram realizados pela equipe de pesquisa durante toda a etapa de desenvolvimento. As funcionalidades eram testadas logo após a sua implementação. Seguindo esta metodologia, foi possível realizar testes e assegurar o correto funcionamento da aplicação. Pretende-se, assim que a versão 1.0 esteja completa, realizar testes com o público-alvo. Deseja-se selecionar um grupo de professores e estudantes da UEFS para utilizar a ferramenta na prática e assegurar, desta forma, além do correto funcionamento, a aceitação do AVAA pelo público, além de coletar possíveis sugestões e melhorias para a ferramenta.

CONSIDERAÇÕES FINAIS (ou Conclusão)

Por encontrar-se na primeira versão, a ferramenta ainda precisa de melhorias, principalmente no que tange a reação do público-alvo usuário, que se torna diferente da visão do programador, que, por estar no papel de autor, prevê os resultados que podem ser encontrados. Dessa forma, é possível que sejam encontradas falhas na verificação de uma navegação intuitiva por parte de um usuário real.

Funcionalidades a mais também podem ser inseridas, como fóruns, cadastro de videoaulas, armazenamento de material de apoio entre outras, todas estas podendo transformar a ferramenta em um instrumento de educação poderoso em um curso de algoritmo e programação. Contudo, mesmo com apenas as funcionalidades já implementadas, é possível obter projeções de melhoras na educação, pelos próprios resultados já obtidos ao longo deste programa de iniciação científica.

REFERÊNCIAS

ECMA. Disponível em: <http://www.ecma-international.org/>. Acesso realizado em 21 de Julho de 2017.

FRANCO, Marcelo A.; CORDEIRO, Luciana M.; CASTILLO, RENATA, A. F.; O ambiente virtual de aprendizagem e sua incorporação na Unicamp. São Paulo, 2003.

GETBOOTSTRAP. Disponível em: <http://getbootstrap.com>. Acesso realizado em 21 de Julho de 2017.

MOODLE. Disponível em: <https://moodle.org/>. 2017. Acesso realizado em 21 de Julho de 2017.

MYSQL. Disponível em: <https://www.mysql.com/>. Acesso realizado em 21 de Julho de 2017.

PHP. Disponível em: <http://www.php.net/>. Acesso realizado em 21 de Julho de 2017.

PRIOR, J. C. Online assessment of SQL query formulation skills. In Proceedings of the Fifth Australasian Conference on Computing Education. Adelaide, Australia. 2003.

ROMERO, C., VENTURA, S., and GARCIA, E. Data mining in course management systems: Moodle case study and tutorial. **Computers & Education**, 51(1):368–384, 2008.

SILVA, Jefté Goes S.; SENA, Claudia P. P. Ambiente Virtual de Aprendizagem de Algoritmos (AVAA): a Construção de uma Ferramenta no Processo Ensino-Aprendizagem. In: **XII SEPA – Seminário Estudantil de Produção Acadêmica** – UNIFACS, 2013.

W3C. Disponível em: <http://www.w3c.br/Home/WebHome>. Acesso realizado em 21 de Julho de 2017.