

MANUTENÇÃO E EVOLUÇÃO DO SOFTWARE PBL-VSII

Rodrigo Fraga Oliveira¹; Gabriela Ribeiro Peixoto Rezende Pinto²

1. Bolsista PIBIC/FAPESB, Graduando em Engenharia de Computação, Universidade Estadual de Feira de Santana, e-mail: rodrigofraga6@gmail.com

2. Orientador, Departamento de Ciências Exatas, Universidade Estadual de Feira de Santana, e-mail: gabrielarprp@gmail.com

PALAVRAS-CHAVE: Informática; PBL; PBL-VSII.

INTRODUÇÃO

O dinamismo do conhecimento na sociedade contemporânea é notavelmente acelerado e para acompanhar tal evolução novas estratégias pedagógicas, como o método *Problem Based Learning* (PBL), também oferecem suporte ao processo de ensino-aprendizagem. Tanto a utilização de recursos tecnológicos quanto o método PBL, sugere uma alteração da forma de ensino-aprendizagem convencional, na qual o professor é o centro transmissão de conhecimentos, nessa nova visão o estudante juntamente com o docente passa a ser um participante ativo do processo educacional, fazendo com que ambos produzam o conhecimento (ALMEIDA, 2003; SANTOS, 2012).

A partir da primeira experiência realizada no curso de Medicina da McMaster University, várias outras instituições de ensino seguiram princípios similares, não apenas em programas de Medicina, mas também de Veterinária, Direito, Arquitetura, Engenharia, entre outros, dando mais dinamismo e autodidatismo para os estudantes. No Brasil, a primeira a implementar o método foi a faculdade de Medicina de Marília, em 1997. Na Bahia, verifica-se a sua implantação no curso de Medicina da Universidade Estadual de Santa Cruz (UESC) e da Universidade Federal da Bahia (UFBA).

A Universidade Estadual de Feira de Santana (UEFS), em 2003, o implantou no curso de Medicina e no curso de Engenharia de Computação (EComp) (PINTO, et al. 2012). Seguindo as novas estratégias pedagógicas, desde a implantação do curso de Engenharia de Computação, da UEFS, que adotou o método PBL no currículo, para mediar o processo de ensino-aprendizagem, o software Problem Based Learning – Virtual System II (PBL-VS II) vem sendo desenvolvido. O objetivo do desenvolvimento do PBL-VS II é possibilitar a realização da dinâmica prevista no método PBL tanto de forma presencial como a distância.

Desde sua primeira versão, denominada Problem Based Learnig – Virtual Enviroment (PBL-VE), o PBL-VS II continua em implementação. Atualmente vem sendo desenvolvido, pelo autor da pesquisa, a partir da linguagem de programação *Personal Home Page* (PHP) e da base de dados PostgreSQL. Este vem realizando atualizações no software, tais como: articulação do código, baseando-se no modelo de arquitetura *Model View Controler* (MVC) no lado do servidor (Back-end) e utilizando um dos modelos de desenvolvimento mais atuais, *Single Page Application* (SPA) no lado do cliente (Front-end), redefinição do banco de dados (criou chaves primárias, tabelas, campos que estavam pendentes e atualizou os relacionamentos) e refatoração do código para liberar a versão do PBL-VS II para a comunidade (PINTO, 2004; COSTA, 2011; SANTOS, 2012 e OLIVEIRA, 2013).

Fraga (2014), autor desta pesquisa, vem trabalhando na equipe de desenvolvimento do PBL-VSII, juntamente com Oliveira(2014), a partir da realização das atividades previstas em seu primeiro plano de trabalho de iniciação científica. Desenvolveu mecanismos para filtrar e neutralizar ataques ao sistema e após o desenvolvimento foram executados teste como de penetração, de injeção de SQL, testes de *cross-site scripting*, para garantir essas características. Inicialmente o principal objetivo do trabalho de Fraga (2014) foi encontrar métodos que garantam ao PBL-VS II ser uma aplicação que ofereça disponibilidade, confiabilidade, segurança e proteção aos seus usuários.

No entanto, apesar das contribuições promovidas pelos trabalhos de Oliveira (2014) e Fraga (2014), o software, ainda em fase de desenvolvimento, apresentava algumas pendências de implementação nas visões Administrador, Tutor e Aluno; também precisou ser submetido a novos testes, para se chegar em uma versão mais estável. Assim, foi fundamental a realização de algumas atividades de implementação e teste, a partir das visões do sistema, para que as pendências pudessem ser resolvidas e possíveis falhas corrigidas.

MATERIAL E MÉTODOS OU METODOLOGIA (ou equivalente)

Esta pesquisa contou com o apoio dos integrantes do GICE; e tem como principal referência os resultados obtidos pelo projeto de pesquisa intitulado “Estudo da aplicação do método de aprendizagem baseada em problemas em cursos de graduação e pós-graduação de computação”. Especialmente o software PBL-VS II (Problem Based Learning – Virtual System II), que é o software alvo para os experimentos da pesquisa.

Para a realização do plano de trabalho, foram realizadas pesquisas bibliográficas e documentais obtendo uma melhor compreensão das tecnologias utilizadas no processo de desenvolvimento do PBL-VS II, tendo como exemplo Laravel e AngularJS ambos *frameworks* para o desenvolvimento do *Back-end* e *Front-end*, respectivamente. Também foi importante, durante todo o período de pesquisa, buscar referências sobre o método PBL e as tecnologias envolvidas no processo de produção.

A partir da documentação e bibliografia consultada, explorou-se o código do software e o banco de dados associado. Então, foi possível o desenvolvimento das habilidades em utilizá-lo e implementá-lo. Uma das ferramentas de suporte, o LARAVEL *framework*, disponibiliza em sua documentação toda a estrutura do seu código baseada na arquitetura *Model-View-Controller* (MVC), estrutura esta que é a base do código do PBL-VS II, juntamente com *AngularJS*, responsável pelo *front-end* da aplicação.

Após o procedimento de exploração do código, com o objetivo de identificar as falhas e pendências de implementação no PBL-VSII, foram levantadas as falhas encontradas no software bem como os requisitos que se encontravam pendentes, e, a cada falha ou requisito encontrado, foi realizada a implementação dos requisitos e falhas. Nesta etapa, o desenvolvimento dos requisitos que se encontravam pendentes foi levado com prioridade em relação aos novos, pois havia a necessidade de atender os princípios básicos do método PBL. Além disso, foram corrigidas as falhas que foram encontradas.

Durante a exploração e refatoração do código foi identificada a possibilidade de utilizar o método de desenvolvimento *Single Page Application*, que possibilitou vantagens na aplicação como balanceamento da responsabilidade da execução entre cliente e servidor removendo a sobrecarga do servidor; menos código do servidor, e mais responsabilidade no cliente, menor consumo de banda, pois as cargas de dados são feitas por demanda e por AJAX.

Visto que, a velocidade do tráfego de informações entre os usuários, principalmente no chat onde ocorre a dinâmica do método PBL, é fundamental, já que o ambiente virtual deve realizar uma conversa em tempo real.

Para validar os requisitos do software que foram implementados, foi elaborado um problema-teste, e, no laboratório em que o GICE funciona, foram testadas as funcionalidades já disponibilizadas pelo PBL-VSII. A orientadora e o pesquisador deste plano de trabalho acessaram o software online, a partir de cinco máquinas, simulando os seguintes usuários fictícios: Estudante01, Estudante02, Estudante03, Admin01 e Tutor04. Após a realização dos testes, foi feita a análise dos resultados obtidos, destacando as funcionalidades que já estão disponíveis e aquelas que ainda precisam ser implementadas (FRAGA e PINTO, 2016).

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

Um dos primeiros passos dessa pesquisa foi a compreensão do código que já havia sido construído por outros integrantes do grupo de pesquisa. Após o domínio do código que possuía sua estrutura baseada no MVC, linguagem de programação PHP utilizando o *Framework* Laravel e o banco de dados *postgreSQL*, deu-se início a refatoração do código. A arquitetura do Laravel possibilitou a evolução do PBL-VS, uma vez que, a versão anterior do sistema utilizava a versão 4 do *framework*, uma versão estável, porém com algumas limitações que inviabilizava a utilização de novas tecnologias, que dão suporte ao sistema.

Um dos grandes impactos causados na implementação da versão anterior do sistema foi o atraso de informações na tela da sessão tutorial. Essa tela faz requisições a todo instante ao controlador para enviar/receber as novas mensagens da sessão, simulando assim um ambiente em tempo real, porém devido a quantidade de requisições ao servidor e o número de dados retornados, as informações chegavam em cerca de 0,9 à 1,5 segundos, sendo que as informações eram requisitadas a cada 2 segundos.

Para solução desse problema foi adotado um novo modelo de desenvolvimento de aplicações Web e mobile que vem ganhando destaque em grandes empresas como *Microsoft*, *Google*, *Twitter*, *Facebook*, etc. Esse novo modelo *Single Page Application* (SPA) é praticamente uma aplicação *Desktop* rodando sob o navegador. Tendo como principais vantagens melhorar a experiência ao usuário criando interface com usabilidade moderna e de fácil entendimento do usuário; menor consumo de banda, pois as cargas de dados são feitas por demanda e por AJAX.

Basicamente o usuário (cliente) recebe uma página html do servidor com o javascript e estilo da aplicação, e a partir da interação do usuário, o outros fragmentos da aplicação serão carregados todos sob demanda e apenas uma vez. Com isso, o problema do tráfego de informações, consumo de banda e velocidade de transmissão foi

reduzido consideravelmente, de modo geral podemos representar a arquitetura da camada superior do PBL-VSII da seguinte forma:

1. Cliente requisita a página inicial do sistema,
2. O servidor retorna a página inicial, que contém, além do html, o estilo e o *javascript*.
3. Nesse ponto, inicia-se um ciclo em que o cliente faz as requisições dos dados e o servidor se encarrega de retornar apenas o essencial para aquela requisição.

A aplicação contém três papéis para os usuários, administrador, estudante e tutor. O administrador é pré-cadastrado pelo sistema e, no momento da configuração para disponibilizar o software, ele é criado e armazenado no banco de dados.

O administrador é o responsável por cadastrar os tutores, no entanto, para segurança do sistema e integridades dos usuários (tutor) foi utilizado os mecanismos de envio de e-mail oferecidos pelo *Laravel*, que disponibiliza uma API simples utilizando a biblioteca *SwiftMailer*, que oferece uma abordagem orientada a objetos flexível e elegante para enviar e-mails e também muito popular na comunidade PHP. Foi desenvolvido um mecanismo para que, após o cadastro de um tutor, seja enviado um e-mail para o mesmo, com a confirmação do cadastro e uma senha de acesso, para que possa ser efetuado a autenticação no sistema. Após autenticado, o usuário poderá alterar a qualquer momento sua senha.

Embora as telas do sistema sejam similares, existem regras para cada papel, como exemplo, durante uma sessão tutorial o tutor pode encerrar uma sessão, funcionalidade que não deve estar disponível para os estudantes. Como existem restrições de funcionalidades para os diversos papéis foram construídas telas similares para cada tipo de usuário levando em consideração seu papel.

A versão do sistema anterior ao início dessa pesquisa disponibilizava parte da implementação da tela do quadro do tutor, tomando-a como base, foi implementado a tela do quadro do estudante e refatorada a tela do tutor. As atividades de implementação levou em consideração os seguintes requisitos: Lista de quadros, os quadros de Ideia, Questões, Fatos e Metas; Exibir postagens nos quadros com nome do remetente e ordenado pela última postagem; Chat, onde ocorre as discussões sobre o problema e envio das ideias, questões, fatos e metas. Com essa implementação foi possível realizar a dinâmica prevista no método PBL.

CONSIDERAÇÕES FINAIS (ou Conclusão)

Com base no relatório apresentado, pode-se perceber que o software encontra-se em uma versão em que atende os requisitos propostos pelo método *Problem Based Learning* (PBL), embora existam requisitos do sistema que ainda não estão implementados e podem ser inclusos no processo de evolução do software, tal processo que, segundo a tese sobre evolução de software publicado pela PUC-RIO (PUC-RIO, 1998), um projeto de software dificilmente chegara a um estágio final definitivo, uma vez que, mesmo quando o software atinge um estágio considerado satisfatório para quem o desenvolveu, este estágio pode não ser satisfatório para todos os usuários.

Gostar-se-ia de ter abrangido um leque maior implementação de requisitos e testes para que se possa chegar em uma versão mais próxima do estágio final, o que foi

restringido pelo tempo que foi estipulado para o projeto. Nota-se, no entanto, que a implementação de métodos como SPA, possibilita uma melhor usabilidade para os usuários, assim como o método de geração e senhas para os usuários com papel de tutor no sistema; além de utilizar padrões desenvolvimento claramente documentados que oferece maior agilidade e fácil compreensão do código do projeto e facilitando futuras evoluções. Conclui-se, desta forma, que os resultados obtidos foram satisfatórios.

REFERÊNCIAS

- ALMEIDA, M. E. B. (2003). “Educação à distância na internet: abordagens e contribuições dos ambientes digitais de aprendizagem”. São Paulo: PUC/SP; Microsoft Brasil, 2003.
- PINTO, G.R.P.R. AVPBL – um ambiente virtual para auxiliar sessões tutoriais do método de aprendizagem baseada em problemas. Dissertação de Mestrado apresentada à UNIFACS. Salvador, 2004.
- COSTA, R. A. ; PEREIRA, H. B. B. . PBL-ME: um ambiente de aprendizagem para dispositivos móveis voltado para o aprendizado baseada em problemas. In: X Escola Regional de Computação Bahia-Sergipe-Alagoas, 2010, Maceió. X ERBASE: Inovação, Tecnologia e Interdisciplinaridade, 2010. v. 1.
- SANTOS (2012). “Problem Based Learning Virtual System (PBL – VS): um software para apoiar a aplicação do método PBL presencialmente e a distância”. Trabalho de Conclusão de Curso. Universidade Estadual de Feira de Santana. Feira de Santana.
- PINTO, G.R.P.R et al. (2012). “PBL-VE: Um Ambiente Virtual para Apoiar a Aprendizagem Baseada em Problemas”. In: Congresso Brasileiro de Educação em Engenharia, 2011, Blumenau. Congresso Brasileiro de Educação em Engenharia, 2012.
- OLIVEIRA, J. C. S. (2014). Modelagem e desenvolvimento do aplicativo PBL-MD para integração ao sistema PBL-VS.
- FRAGA, R. O. (2014). Identificação de vulnerabilidades e implementação, verificação e validação de estratégia de segurança para o Software PBL-VS II.
- ESCOLAS MÉDICAS DO BRASIL. Metodologia de Ensino, disponível em: < <http://www.escolasmedicas.com.br/metodo.php> > Acesso em 30 de novembro de 2015.
- PUC-RIO- Tesses Abertas cap-3 Evolução de Software, disponível em: < http://www2.dbd.puc-rio.br/pergamum/tesesabertas/0210500_04_cap_03.pdf > Acesso em 30 de maio de 2016.
- FRAGA, R. O , PINTO, G.R.P.R. (2016). “Ambiente Virtual para Apoiar a Aplicação do Método de Aprendizagem Baseada em Problemas”. In: Congresso Brasileiro de Educação em Engenharia, 2016, Natal.