

# Desenvolvimento de uma plataforma de testes automatizados como forma de apoiar a aprendizagem em testes

Rafaela Melo  
Universidade Federal do Amazonas  
Manaus, Amazonas, Brasil  
rmelo@icomp.ufam.edu.br

Fernanda Pires  
Ramayana Júnior  
Rodrigo Choji  
Universidade do Estado do Amazonas  
Manaus, Amazonas, Brasil  
{fpires,rajunior,rcfreitas}@uea.edu.br

Em empresas de tecnologia, sejam de software ou hardware, o processo de testes é necessário para manter a qualidade dos produtos e garantir que o cliente final tenha uma boa experiência e não encontre problemas. Para tanto, verificações, validações e testes devem ser realizados durante o processo de desenvolvimento [2]. Apesar disso, pesquisas apontam que grande parte dos estudantes da área de Computação só aprendem sobre testes quando precisam aplicá-los em algum projeto [3]. Isso pode se dar pelo fato de que conteúdos relacionados a testes são geralmente trabalhados por meio de aulas expositivas e não práticas [1]. Este trabalho trata da criação de uma plataforma de testes automatizados de dispositivos de hardware, intitulada *Smart Test*, como ferramenta para o desenvolvimento de habilidades e competências em testes. O processo considerou a aplicação de metodologias de aprendizagem ativas, pois o time de estudantes dos cursos de Licenciatura em Computação, Sistemas de Informação e Engenharia da Computação e professores foram responsáveis pelo projeto desde a concepção até o desenvolvimento da ferramenta.

Visto que se tratava de um ambiente de testes, foi necessário entender como funcionavam as rotinas. Assim, o passo inicial foi entender a visão da equipe sobre testes, tanto de software quanto de hardware. Posteriormente, iniciou-se o processo de idealização da plataforma, partindo da arquitetura e levantamento de requisitos. Dessa forma, os estudantes puderam também ter a experiência de como funciona parte do mercado de desenvolvimento de software.

Testes de hardware geralmente são feitos de forma manual, onde são gerados scripts de comandos referentes a funcionalidades existentes em dispositivos físicos. Logo, o propósito de desenvolver a ferramenta foi automatizar esse processo, de forma a oferecer uma ferramenta onde os responsáveis pelos testes pudessem ter em sua disposição as funções necessárias e precisassem apenas montar os "planos de testes". Nesse sentido, pensou-se em facilitar a montagem dos planos por meio de recursos de programação em blocos, visto que nem todas as pessoas envolvidas nos testes de hardware possuem conhecimento e facilidade em linguagens de programação. Um ambiente de programação baseado em blocos permite ao usuário criar programas apenas com o uso do mouse e

a mecânica de arrastar e soltar (*drag and drop*), semelhante ao ato de montar um quebra-cabeça [4].

Diante disso, iniciaram-se pesquisas sobre tecnologias para a implementação da plataforma. Depois de testar algumas ferramentas como HTML, CSS e JavaScript, foi decidido que, para iniciar o desenvolvimento do *Smart Test*, seria utilizado o Google Blockly<sup>1</sup>, uma biblioteca de JavaScript que possibilita a criação de ambientes de programação baseado em blocos. A Figura 1 apresenta o ambiente desenvolvido, que dispõe de uma aba de categorias de blocos separadas por cores do lado esquerdo, semelhante ao Scratch<sup>2</sup>, e, ao centro existe um plano de teste montado a partir dos blocos disponíveis nas abas.

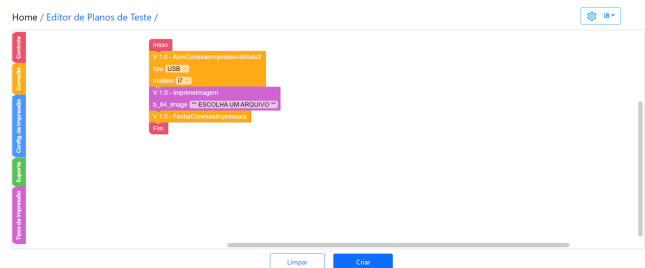


Figura 1: Ambiente de testes desenvolvido com o apoio da biblioteca Google Blockly.

Durante e após o desenvolvimento do ambiente foram executados testes na interface e no *back-end* integrado para garantir que o sistema funcionava como foi idealizado. Essa rotina de testes iniciou-se com a criação de um documento intitulado "Roteiro de Testes", que possuía todas as especificações sobre as funcionalidades da plataforma e como executar cada uma delas, os caminhos felizes e infelizes. Os estudantes também participaram ativamente dessa etapa, tanto na criação do roteiro quanto na execução, entendendo de fato como funcionava o processo de testes.

O desenvolvimento do *Smart Test* possibilitou que estudantes de três áreas distintas de Computação pudessem participar ativamente do processo de desenvolvimento de uma plataforma automatizada de testes, colocando em prática os conhecimentos abordados nas disciplinas e conhecendo a rotina de testes que é necessária para a entrega de um bom produto.

Fica permitido ao(s) autor(es) ou a terceiros a reprodução ou distribuição, em parte ou no todo, do material extraído dessa obra, de forma verbatim, adaptada ou remixada, bem como a criação ou produção a partir do conteúdo dessa obra, para fins não comerciais, desde que sejam atribuídos os devidos créditos à criação original, sob os termos da licença CC BY-NC 4.0.

EduComp'23, Abril 24-29, 2023, Recife, Pernambuco, Brasil (On-line)

© 2023 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

<sup>1</sup><https://developers.google.com/blockly>

<sup>2</sup><https://scratch.mit.edu/>

## AGRADECIMENTOS

Os resultados apresentados nesta publicação foram obtidos por meio de atividades de Pesquisa e Desenvolvimento executadas no projeto Elgin Smart Test, com recursos oriundos da Lei nº 8.387/1991, devendo qualquer publicidade fazer referência a citada lei conforme Art. 48 do Decreto nº 6.008/2006, tendo a empresa Elgin Industrial da Amazônia LTDA financiado esse projeto nos termos da mencionada lei.

## REFERÊNCIAS

- [1] Tarcila Gesteira da Silva, Felipe Martins Müller, and Giliane Bernardi. 2011. Panorama do ensino de engenharia de software em cursos de graduação focado em teste de software: uma proposta de aprendizagem baseada em jogos. *RENOTE* 9, 2.
- [2] Marcio Delamaro, Mario Jino, and Jose Maldonado. 2013. *Introdução ao teste de software*. Elsevier Brasil.
- [3] Christiane Gresse Von Wangenheim and Djoni Antonio Silva. 2009. Qual conhecimento de engenharia de software é importante para um profissional de software. *Proceedings of the Fórum de Educação em Engenharia de Software* 2, 1–8.
- [4] David Weintrop and Uri Wilensky. 2015. To block or not to block, that is the question: students' perceptions of blocks-based programming. In *Proceedings of the 14th international conference on interaction design and children*. 199–208.