

TESTES AUTOMATIZADOS UTILIZANDO APRENDIZAGEM DE MÁQUINA

Flávio Scartezini Junior 1
Isaak Iedo de Assunção 2
Roberto Fabiano Fernandes 3
Sérgio Murilo Schutz 4
Ibsem Agrello Dias 5

INTRODUÇÃO

Este trabalho faz parte do Projeto Multidisciplinar Integrador I, o qual tem por objetivo interar os alunos no mundo da pesquisa para desenvolver o seu projeto de desenvolvimento de software, como trabalho final para o curso de Análise e Desenvolvimento de Sistemas. Com este entendimento objetivou-se, neste momento, conceber uma ideia inovadora que fosse amparada pela pesquisa acadêmica. Quando um software entra em produção (termo utilizado na engenharia de software para informar que já está sendo utilizado pelo cliente) cada defeito encontrado pode gerar transtorno na operação, dependendo da complexidade do defeito, pode parar a operação até que o mesmo seja corrigido, gerando prejuízos e falta de confiança na qualidade do produto (BARBOSA, 2000; MONARD e BARANAUSKAS, 2003).

OBJETIVO

Propor o desenvolvimento de um protótipo de sistema de testes automatizado utilizando aprendizagem de máquina que possa encontrar os defeitos antes dos usuários.

METODOLOGIA

As etapas metodológicas deste trabalho se iniciam fazendo o levantamento bibliográfico, abordando as áreas de testes de software e aprendizagem de máquina. Para isso, realizou-se pesquisas na base de dados computadorizada, o Google Acadêmico®. O Google Acadêmico® para Creswell (2010) é uma base de dados gratuita que proporciona amplitude na busca na literatura de várias fontes, como teses, resumos e artigos, com a vantagem de poder obtê-los de forma integral. Ele também pode ser utilizado como auxílio às buscas complementares minimizando a possibilidade de que algum artigo reconhecido cientificamente e alinhado com o tema desta pesquisa não fosse incorporado ao portfólio bibliográfico. Para apoiar a elaboração do mesmo, a segunda etapa é proposta em desenvolver uma arquitetura que consiste em capturar, armazenar e analisar cenários de uso dos usuários no sistema e depois criar e executar scripts de testes com base na informações levantadas. A partir da arquitetura, entra a prototipagem para atestar a viabilidade da arquitetura, e, em seguida, um estudo de caso, aplicando o protótipo, e, ao final, validar o protótipo através da aplicação de um questionário, podendo então formular as conclusões e trabalhos futuros.

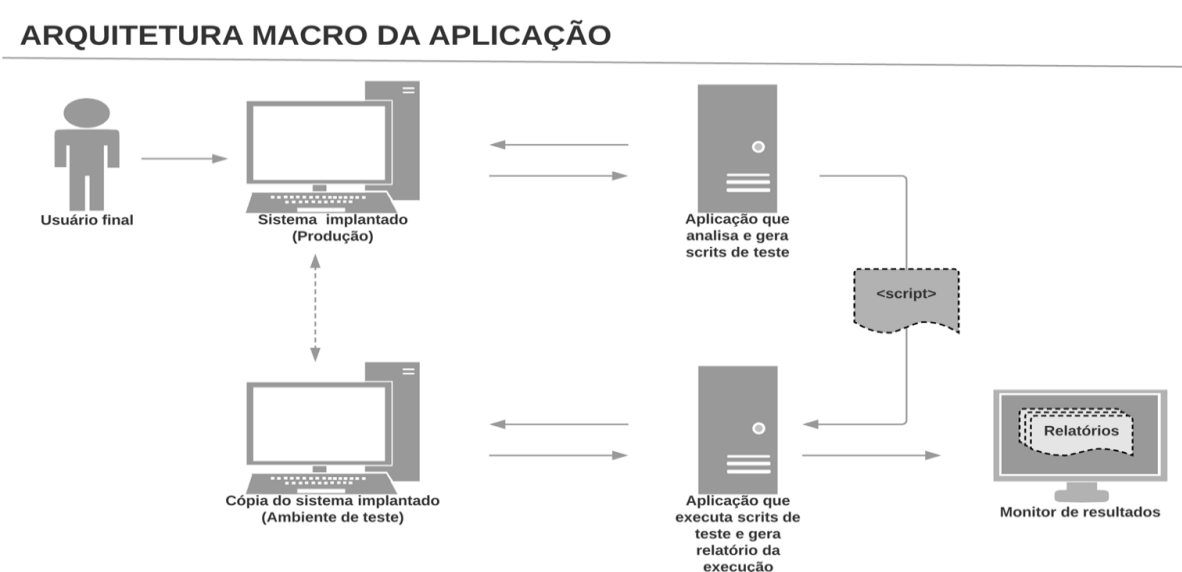
DESENVOLVIMENTO

O projeto se encontra em fase de idealização, sendo assim, ainda não foram estipuladas métricas e/ ou metodologias para a concepção do sistema, bem como linguagem de programação ou ambiente.

Como a premissa do sistema consistem em mapear, capturar, e, a partir do armazenamento destas informações realizar a análise dos cenários de uso dos usuários no sistema, obtendo mapas de calor com base no uso mais recorrente por parte dos usuários, e depois criar e executar scripts de testes automatizados com base na informações levantadas, já é possível determinar que a linguagem usada deverá atender ao paradigma orientado a eventos (Python, Delphi, JavaScript), pois deverá funcionar de forma transparente para o usuário.

O processo de captura será executado no momento que o usuário acionar sensores (que são chamados de gatilhos ou eventos) de acordo com as iterações com o sistema. Estes eventos podem ser click, duplo click, ou o pressionar de uma tecla, capturando a posição do cursor e como o usuário interagiu com a interface, conforme é representado na figura 1.

Figura 1 – Arquitetura da proposta



Fonte: Autores (2018)

CONSIDERAÇÕES FINAIS

Levando em consideração a crescente demanda por software de qualidade e a exigência por parte das empresas que pagam por serviços de software, o sistema aqui idealizado tem como principal objetivo diminuir a incidência de problemas em fase de produção, que é quando o sistema já está sendo utilizado pelo usuário final. A principal visão é, além de diminuir a ocorrência de problemas no sistema, um aumento considerável na agilidade do processo de construção do software, visto que o tempo dedicado a corrigir falhas encontrados já na fase de produção será diminuído. Considerando também, que, o tempo das entregas será beneficiado, aumentando a produtividade, diminuição dos erros encontrados pelo usuário final, aumentando a credibilidade no sistema, e assim também na empresa.

1 Graduando em Análise e Desenvolvimento de Sistemas. Faculdade Cesusc / flavio.scartezini@gmail.com
2 Graduando em Análise e Desenvolvimento de Sistemas. Faculdade Cesusc / isaakassuncao@gmail.com
3 Doutor em Engenharia do Conhecimento. Faculdade Cesusc / robertofabiano.fernandes@gmail.com
4 Mestre em Ciência da Informação. Faculdade Cesusc / sergio.schutz@cesusc.edu.br
5 Mestre em Administração. Faculdade Cesusc / ibsem.dias@gmail.com

REFERÊNCIAS BIBLIOGRÁFICAS

BARBOSA, Ellen Francine et al. Introdução ao teste de software. **Minicurso apresentado no XIV Simpósio Brasileiro de Engenharia de Software (SBES 2000)**, 2000.
CRESWELL, John W. Projeto de pesquisa métodos qualitativo, quantitativo e misto. In: **Projeto de pesquisa métodos qualitativo, quantitativo e misto**. 2010.
MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Conceitos sobre aprendizado de máquina. **Sistemas inteligentes-Fundamentos e aplicações**. 1 ed. Barueri-SP: Manole Ltda, v. 1, n. 1, p. 32, 2003.