UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA
Autorizada pelo Decreto Federal nº 77.496 de 27/04/76
Recredenciamento pelo Decreto n°17.228 de 25/11/2016
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
COORDENAÇÃO DE INICIAÇÃO CIENTÍFICA

XXIV SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UEFS
SEMANA NACIONAL DE CIÊNCIA E TECNOLOGIA - 2020

Elaboração de Arquitetura de Software para Desenvolvimento de Aplicações Multiplataforma para Simulação de Circuitos Elétricos e Eletrônicos

Marcos de Jesus Ramos[1] e Ana Cláudia Fiorin Pianesso [2]

1. Bolsista PROBIC/CNPq, Graduando em Engenharia da Computação, Universidade Estadual de Feira de Santana, e-mail: mjramosfsa@gmail.com

2. Ana Cláudia Fiorin Pianesso, Departamento de Tecnologia (DTEC), Universidade Estadual de Feira de Santana, e-mail: acfpianesso@uefs.br

PALAVRAS-CHAVE: Arquitetura de software, Protocolo de comunicação, Circuitos elétricos e eletrônicos

INTRODUÇÃO

A engenharia elétrica possui um histórico de problemas na realização de testes destinados a comprovação do comportamento de circuitos elétricos e eletrônicos. Para isso, utilizava-se componentes físicos, o que consequentemente acarretava em uma elevação nos custos do projeto, além da ausência de segurança e eficiência na análise desses sistemas (MANZONI, 2005). Surgem então, os softwares de simulação, que permitem a criação e manipulação virtual de cenários reais, possibilitando assim a realização de testes sem que haja a necessidade de manipulação de componentes físicos. São ferramentas atualmente utilizadas nos estudos de sistemas elétricos e eletrônicos.

Existem variações disponíveis para uso no mercado, entretanto, mesmo com a evolução destas ferramentas, as pesquisas desenvolvidas na área de engenharia elétrica ainda não estão totalmente amparadas. Os problemas encontrados são os mais diversos, desde funcionalidade limitada a circuitos simples, uso de linguagens defasadas, interface pouco intuitiva, alto custo das licenças de uso, ausência de manutenção em software livre, falta de documentação até a inexistência e flexibilidade para incorporação de um modelo matemático elaborado. Segundo Souza (2011), esses problemas acabam impactando de forma negativa as pesquisas acadêmicas na área.

A motivação do projeto de pesquisa é o desenvolvimento de um simulador de sistemas elétricos que possa ser capaz de possibilitar a inclusão de modelos já desenvolvidos, sejam em simuladores ou em resultados de pesquisas. Para isso, técnicas e métodos necessitam ser aplicados para garantir a usabilidade e facilidade de manutenção. Esse conjunto de técnicas estão descritas na Engenharia de Software com os aspectos fundamentais para a idealização, estudo, elaboração e concepção de um software e/ou aplicação (SOMMERVILLE, 2007). E, sendo o propósito do projeto a elaboração de um Simulador de Sistemas elétricos e eletrônicos faz-se necessário projetar uma arquitetura de software, objetivando definir e desenvolver as estruturas e características que o sistema possuirá. Neste contexto deve-se considerar as implicações relacionadas à comunicação das partes da aplicação a ser desenvolvida (GUEDES, 2008). Surge assim

a necessidade de um protocolo de comunicação próprio, que propicie recursos capazes de assegurar a integridade, a disponibilidade e a proteção dos dados que serão gerenciados em aplicações multiplataformas (ESCARPINATI, 2009).

Diante do cenário apresentado, situa-se este trabalho, para o desenvolvimento de uma arquitetura de software que forneça recursos para a criação de aplicações multiplataforma, com suporte para elaboração de interfaces gráficas amigáveis e intuitivas voltados à montagem de circuitos, além do desenvolvimento de protocolos de comunicação que garantam a interação entre as aplicações desenvolvidas.

MATERIAL E MÉTODOS

O desenvolvimento deste trabalho utilizou-se de análises e mapeamento das principais ferramentas relacionadas ao tema abordado, já existentes no mercado e suas funcionalidades. Identificados os aspectos positivos e negativos presentes em cada uma destas aplicações relacionando-as com os objetivos do projeto e contribuindo para o levantamento de requisitos.

Técnicas da Engenharia de *Software* foram aplicadas visando minimizar os custos e os possíveis problemas na elaboração de um software. Para isso, utilizou-se a metodologia ágil *Scrum* (GOMES, 2014), coleta de requisitos através de entrevistas e questionários, prototipagem e documentação, por meio de especificações técnicas do usuário do sistema.

A proposta de arquitetura de software implementada utiliza o modelo conceitual de microsserviços (REDHAT, 2013) e o modelo estruturado de comunicação distribuída cliente servidor (TANENBAUM, 2003). O protocolo de comunicação, que permite integração multiplataforma utiliza-se do padrão de estruturação de dados *JSON* (JavaScript Object Notation). Na elaboração da *API* (*Application Programming Interface)* de comunicação e interconexão utilizou-se a linguagem *Javascript* em conjunto com a *runtime node js*, o *framework express js* e a *lib socket.io*, além de técnicas de encapsulamento de dados (MDW, 2010) e de cacheamento utilizando o banco de dados *firebase.*

RESULTADOS E DISCUSSÃO

A arquitetura de software proposta baseou-se nas definições das estruturas e características desejáveis para o Simulador Digital de sistemas elétricos e eletrônicos. Neste contexto, estão englobadas as implicações relacionadas à comunicação das partes que compõem a aplicação em questão (GUEDES, 2008). Surgindo assim, a necessidade de um protocolo de comunicação próprio, que propicie recursos capazes de assegurar a integridade, a disponibilidade e a proteção dos dados que serão gerenciados. Na sua concepção, foram adotados critérios obtidos através de características e funcionalidades desejáveis, identificadas nas principais ferramentas de simulação disponíveis no mercado, além das necessidades acadêmicas e de pesquisa.

Utiliza o conceito de arquitetura de microsserviços (REDHAT, 2013) visando propor uma maior escalabilidade das aplicações que venham a ser desenvolvidas a partir do seu uso. Aliado a este conceito, foi adotado o modelo estruturado de comunicação entre aplicações distribuídas cliente/servidor (TANENBAUM, 2003). Com isso, fornecesse maior compatibilidade com os sistemas operacionais das principais

plataformas hoje presentes no mercado. É uma arquitetura híbrida (REDHAT, 2013), permitindo abstrair a complexidade no desenvolvimento de aplicações, por fornecer regras de consumo e produção dos *endpoints* que serão processados (AMAZON, 2013). As partes que compõem a arquitetura são apresentadas na Figura 1.

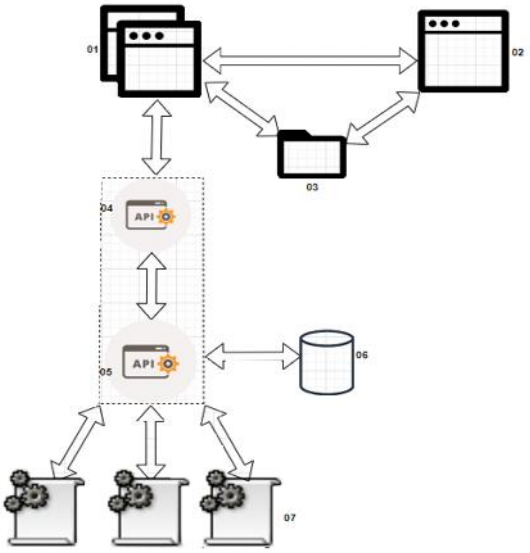

Figura 1: Arquitetura de *software* proposta

É composta por uma interface para o usuário, sendo esta principal do sistema (elemento 01 da Figura 1) que pode ser desenvolvida em qualquer linguagem. A interface desenvolvida utilizou-se do *framework ionic*. Nela são elaborados os circuitos, as simulações e as verificações dos resultados. O elemento 02 representa o módulo de elaboração de bibliotecas para a adição de componentes elétricos e eletrônicos. Possui liberdade para escolha de linguagem de implementação, está sendo implementado em linguagem Java. É composto por uma interface que propicia a elaboração de componentes (desenho ou representação dos componentes e seus parâmetros) e bibliotecas de componentes, que posteriormente podem ser importados para a interface principal do Simulador. O armazenamento de bibliotecas e componentes é realizado em um banco de dados *firebase*, utilizando os serviços de *real databases* (elemento 03). A API de comunicação é representada pelo elemento 04. Esta processa as informações da interface principal. O elemento 05 representa a API de seleção, outra parte que compõem a API de comunicação geral da arquitetura. Esta recebe os dados da API de comunicação e realiza a seleção de qual módulo de cálculo será usado na simulação, e a técnica de cacheamento com o banco de dados. O Banco de Dados representado pelo item 06 é responsável pelo arquivamento e organização da chamada dos módulos. Os módulos de análises matemáticas de circuitos elétricos e eletrônicos, representados pelo elemento 07, possui a modelagem do circuito. Este recebe as informações das APIs e, enviam as modelagens para as APIs, para serem analisados na interface.

Compondo a arquitetura, o protocolo de comunicação é capaz de trabalhar com uma quantidade de dados que atenda às exigências atuais do grupo de pesquisa que necessita validar modelos matemáticos elaborados, além de assegurar interações entre as aplicações e o servidor que realizará análise e os cálculos matemáticos. A abordagem

utilizada permite a criação de diversas ferramentas de análise e simulação para diversas áreas, e não somente para o escopo deste projeto. A garantia de que as simulações e análises matemáticas dos circuitos só sejam realizadas em estados seguros, utilizando dados consistentes, foi viabilizada. Assim como, o aspecto de interação em tempo real das simulações.

CONSIDERAÇÕES FINAIS

A Arquitetura implementada viabiliza a criação de aplicações para variadas plataformas e áreas de conhecimento de forma simples e padronizada. Com isso, colabora para o ensino-aprendizagem e pesquisa, no sentido de sanar lacunas das ferramentas de mercado disponíveis atualmente. Desta forma, considera-se que os objetivos traçados para este plano de trabalho foram alcançados, em que pese a falta de resultados, de outros planos de trabalho, para a disponibilização do Simulador Digital, objeto do projeto geral.

Como trabalhos futuros já identificados e em andamento, destacam-se a inserção de novas funcionalidade para API de comunicação e interconexão, a realização de um processo de refatoração na base arquitetural visando proporcionar ainda mais recursos para o desenvolvimento de aplicações, além de aplicação mais abrangente da abordagem de microsserviços na Arquitetura.

REFERÊNCIAS

AMAZON. 2013. Escolher um tipo de endpoint para configurar uma API do API Gateway. Homepage: https://docs.aws.amazon.com/pt_br/apigateway/latest /developerguide /api-gateway-api-endpoint-types.html.

GUEDES, G. T. 2008. UML: uma abordagem prática. [S.l.]. Novatec Editora.

GOMES, A. F. 2014. Agile: Desenvolvimento de software com entregas frequentes e foco no valor de negócio. 1ed. Casa do Código.

ESCARPINATI, M. C.; Paz, M. A.; SANTANA, F. C. B. 2009. Desenvolvimento de um Esquematizador Gráfico de Modelos de Linhas de Transmissão. In: WTICG-BASE - Workshop de Trabalhos de Iniciação Científica e Graduação Bahia- Sergipe. IX ERBASE. Ilhéus, BA.

MANZONI, A. 2005. Desenvolvimento de um sistema computacional orientado a objetos para sistemas elétricos de potência: aplicação a simulação rápida e análise da estabilidade de tensão. Rio de Janeiro.

MDW. 2010. Trabalhando com objetos. Homepage: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Trabalhando_com_Objetos.

REDHAT. 2013. O que são os microsserviços? Homepage: https://www.redhat.com/pt-br/topics/microservices/what-are-microservices.

SOMMERVILLE, I. 2007. Engenharia de Software, 8. ed. São Paulo: Pearson Addison Wesley.

SOUZA, D. M. 2011. Simulador de circuitos elétricos de pequeno porte utilizando modelagem orientada a objetos. Universidade Federal do Rio de Janeiro. Rio de Janeiro.

TANENBAUM, S. A. 2003. Redes de Computadores. 4. ed. Amsterdam-Holanda: Editora Campus.