

ARQUITETURA DE COMUNICAÇÃO ENTRE RASPBERRY PI E UMA SPA UTILIZANDO PYTHON E REACTJS

CUNHA, Ana Paula S.¹; FERREIRA, Bruno²

¹Estudante do curso de Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG) - *Campus* Formiga. E-mail: heeypaulaa@gmail.com

²Professor orientador do IFMG - *Campus* Formiga. E-mail: bruno.ferreira@ifmg.edu.br

Resumo: Este trabalho introduz um protótipo de arquitetura de comunicação e apresentação de dados entre um dispositivo *Raspberry PI* e uma aplicação SPA (Single Page Application). O objetivo da arquitetura é facilitar a criação de futuras aplicações para a Internet das Coisas. Nesse caso, os testes foram feitos em uma aplicação para automatizar funcionalidades cotidianas de uma biblioteca, assim, foi criado um serviço de autoatendimento para as tarefas de empréstimo e devolução de livros, os quais possuem tags de RFID para identificação.

Palavras-chave: ReactJS. Internet das coisas. Raspberry PI.

1 INTRODUÇÃO

Internet das coisas (IoT – *Internet of Things*) é um conceito que se refere à interconexão digital de objetos cotidianos através da Internet (TANENBAUM, 2011). Contudo, não é trivial desenvolver aplicações que consigam apresentar informações referentes aos dados gerados pelos mais diversos dispositivos; isso acontece porque esse tipo de aplicação requer um profissional que detenha conhecimento em diversas áreas (GSTI, 2016). Assim, com o objetivo de facilitar o desenvolvimento de aplicações desse perfil, esse projeto apresenta uma arquitetura para conectar dispositivos *Raspberry PI* com páginas Web modernas, conhecidas como *Single Page Application* (SPA). Sites sob essa tecnologia pré-carregam toda a estrutura da aplicação na requisição inicial e, a cada novo pedido, somente os dados necessários são transferidos, economizando assim largura de banda (JADHAV *et al.*, 2015). Aplicações Web neste formato podem ser criadas através de bibliotecas *JavaScript*, como o *ReactJS*. Essa tecnologia foi criada pela equipe do *Facebook* com código aberto e uso gratuito (KOPPALA, 2018).

Já o *Raspberry PI* é um minicomputador criado sob a arquitetura Von Neumann. Ele executa um sistema operacional *Linux* e atualmente é muito usado em IoT para automação residencial (VUJOVIĆ; MAKSIMOVIĆ, 2015). Ele interage com diversos dispositivos, um deles é o leitor de tags de RFID ou, traduzindo do inglês, etiqueta de identificação por rádio frequência.

2 MATERIAIS E MÉTODOS

A metodologia seguida neste trabalho foi inspirada no processo unificado (PU), a qual prega o desenvolvimento de produtos de software através de ciclos de vida incrementais e iterativos, em que cada ciclo é formado por quatro fases: Concepção, Elaboração, Construção e Transição (MAKINO, 2009). Para a criação da arquitetura, essas quatro fases se resumem, respectivamente, nas tarefas: (1) conhecer melhor os requisitos para desenvolver aplicações voltadas à internet das coisas, (2) estudar e documentar as tecnologias e ferramentas utilizadas pelo mercado de trabalho, (3) implementar os códigos de integração entre as tecnologias e ferramentas escolhidas e (4) testar a arquitetura implementando em uma aplicação piloto.

Algumas tecnologias utilizadas foram: 1) Redux, o qual é um contêiner de estado para aplicativos *JavaScript*. Ele tem a função de conectar os componentes com o Servidor sem grande acoplamento (RIBEIRO, 2019); 2) Express, *framework* para desenvolvimento de servidores *Web* (RUBENS, 2017); 3) AXIOS, cliente HTTP para Node.js baseado na API Promise (AXIOS, 2019); e 4) Python, uma das linguagens de programação suportada pelo *Raspberry PI* para criação de aplicações.

3 RESULTADOS E DISCUSSÃO

Referente a arquitetura, assim que o servidor da aplicação *Web* criada em *NodeJS* é iniciada, ele aguarda por uma conexão do cliente Python, ou seja, espera até que o usuário aproxime uma *tag* para a leitura no *Raspberry*. Do mesmo modo, a aplicação SPA se conecta ao servidor para receber a informação da *tag*. Esta comunicação é representada pela Figura 1, nessa arquitetura tanto *back-end* quanto o *front-end* da aplicação se encontram na mesma máquina, na qual o primeiro se refere a todas as regras de negócio, acesso a banco de dados e segurança, já o *front-end*, é todo o código da aplicação responsável pela apresentação do software (*client-side*). Se tratando de aplicações *Web*, é exatamente o código do sistema que roda no navegador (ANDRADE, 2018).

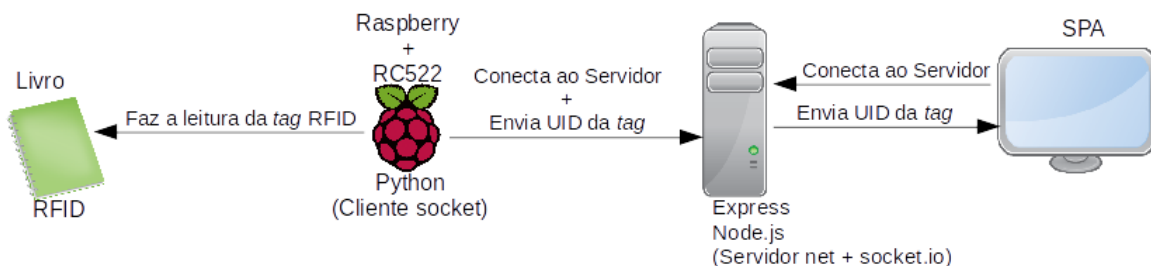


Figura 1 - Arquitetura de comunicação entre o servidor, ReactJs e Raspberry Pi

A Figura 2 mostra o fluxo de dados na máquina servidora quando a SPA recebe o identificador da *tag* vindo do leitor. Ao acontecer um evento de leitura da *tag*, a SPA comunica com o Redux, que passa os dados ao Axios, este faz requisições REST ao banco de dados, tendo o Mongoose como mediador na comunicação. Ao receber os dados do banco de dados pelo Axios, o Redux atualiza os *states* no Store, para que a SPA ReactJS possa usá-los na interface gráfica.

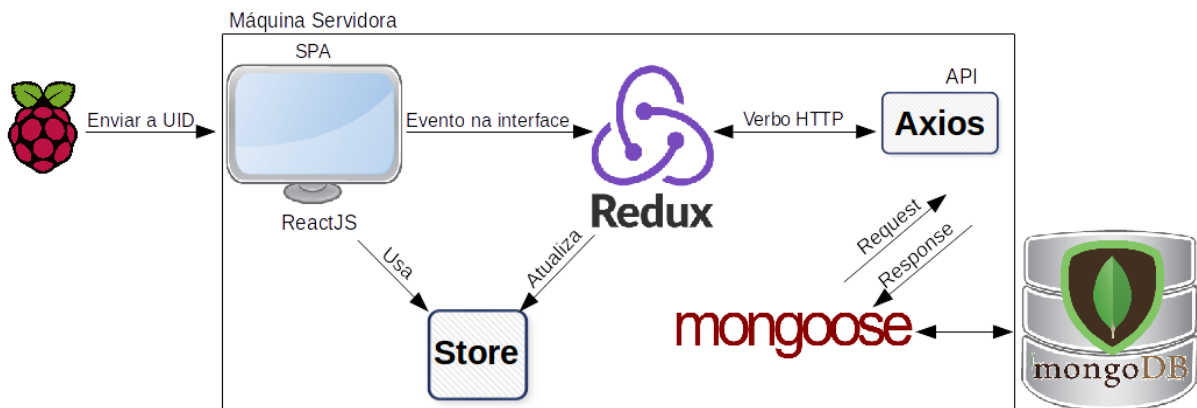


Figura 2 - Arquitetura de comunicação entre o servidor, *ReactJs* e *Raspberry PI*

Para leitura da *tag* RFID foram utilizados o *Raspberry* e o módulo RC522 (Figura 3). A comunicação entre eles é realizada com a tecnologia SPI, que tem como características ser serial e síncrona, essa interface periférica serial, que tem um protocolo com quatro linhas de sinais (SCLK, SSn, MOSI, MISO) (LEENS, 2009). O código e a montagem utilizados para fazer a leitura da *tag* RFID foram baseados no repositório do GITHUB de Ondryáš (2018). O código de leitura da *tag* RFID utiliza a biblioteca *pir522*. Para envio do UID (*Unique Identifier*) do RFID foi desenvolvido um algoritmo que utiliza comunicação via *socket*, responsável por enviar o identificador único da *tag* para o servidor localizado em outro computador.

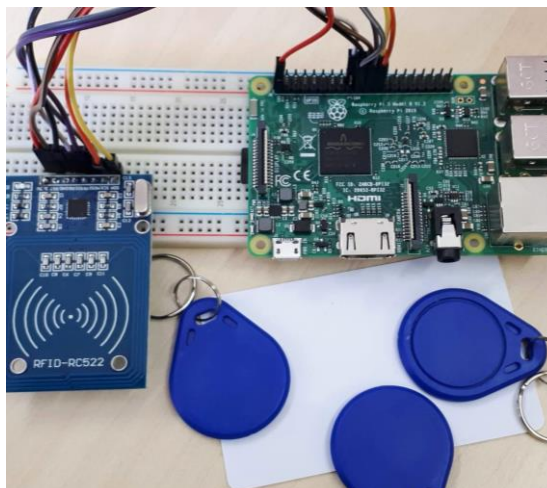


Figura 3 - Montagem do circuito em para comunicação entre leitor e *Raspberry PI*

4 CONCLUSÃO

Nesse trabalho, foi apresentada uma arquitetura de comunicação entre um SPA desenvolvida com o uso de *ReactJS* e um dispositivo *Raspberry PI*. A arquitetura foi validada através da criação de uma aplicação para autoatendimento bibliotecário. Contudo, pode-se concluir que os *softwares*, *frameworks*, dispositivos e tecnologias utilizados para construir a aplicação seguindo a arquitetura funcionaram corretamente, a interoperabilidade entre as tecnologias atendeu os requisitos de desenvolvimento referente a disponibilidade, integridade dos dados e desempenho. Como foi usado, a interface *socket* para fazer a comunicação entre *Raspberry PI* e SPA, para trabalhos futuros, o protocolo de MQTT pode substituir esse método, já que é adequado para aplicações IoT, com rede leve e flexível (YUAN, 2017).

REFERÊNCIAS

- ANDRADE, T. F. **Back-end vs front-end vs fullstack**: Escolha o seu futuro como programador!. Algaworks. 2018. Disponível em <https://bit.ly/2JCAeQX>. Acesso em: 08 set. 2019.
- AXIOS, M. **Promise based HTTP client for the browser and node.js**. 2019.. Disponível em: <https://github.com/axios/axios>. Acesso em: 22 abr. 2019.
- GSTI, P. **Os 05 principais desafios da internet das coisas**. Disponível em: <https://bit.ly/2mfwFXn>. Acesso em: 09 set. de 2019.
- JADHAV, M. A.; SAWANT, B. R.; DESHMUKH, A. Single page application using angularjs. **Journal of Computer Science and Information Technologies**, v. 6, n. 3, p. 2876–2879, 2015.
- KOPPALA, J. **Erp solution with reactjs**. Metropolia Ammattikorkeakoulu, 2018.
- LEENS, F. An introduction to i 2 c and spi protocols. **IEEE Instrumentation & Measurement Magazine**, v. 12, n. 1, p. 8-13, 2009.
- MAKINO, A. **Abordagem da metodologia Rup no desenvolvimento de um sistema de gestão comercial**. 2009.
- ONDRYÁŠ, O. **Raspberry Pi Python library for SPI RFID RC522 module**. 2018. Disponível em: <https://github.com/ondryaso/pi-rc522>. Acesso em: 17 jun. 2019.
- RIBEIRO, S. **Como criar uma aplicação full-stack com react**. 2019. Disponível em: <https://bit.ly/2WSe9S0>. Acesso em: 20 abr. 2019.
- RUBENS, J. **Primeiros passos com Node.js**. Casa do Código, 2017.
- TANENBAUM, A. S. **Redes de Computadores**. 5. ed. Pearson, 2011.

VUJOVIĆ, V.; MAKSIMOVIĆ, M. Raspberry pi as a sensor web node for home automation. **Computers & Electrical Engineering**, v. 44, p. 153–171, 2015.

YUAN, M. **Conhecendo o MQTT**. IBM Developer. 2017. Disponível em: <https://ibm.co/2qc8pr3>. Acesso em: 01 nov. 2019.

Como citar este trabalho:

CUNHA, A. P. S.; FERREIRA, B. Arquitetura de comunicação entre Raspberry PI e uma SPA utilizando Python e ReactJS. *In*: SEMINÁRIO DE PESQUISA E INOVAÇÃO (SemPI), III., 2019. Formiga. **Anais eletrônicos** [...]. Formiga: IFMG – *Campus* Formiga, 2019. ISSN – 2674-7111.