

REVISTA DO

FORUM GERENCIAL

SISTEMAS DE
INFORMAÇÃO

VOLUME 1, NÚMERO 2



UNIPAM
Educação que transforma

REVISTA DO FÓRUM GERENCIAL

Revista do Centro Universitário de Patos de Minas

Vol. 1, n. 2, out. 2021

Patos de Minas: Revista do Fórum Gerencial, UNIPAM, v. 1, n. 2, out. 2021: 1-139



Centro Universitário de Patos de Minas



Núcleo de Editoria e Publicações

UNIPAM | Centro Universitário de Patos de Minas

Reitor

Milton Roberto de Castro Teixeira

Pró-reitor de Ensino, Pesquisa e Extensão

Henrique Carivaldo de Miranda Neto

Pró-reitor de Planejamento, Administração e Finanças

Renato Borges Fernandes

Coordenadora de Extensão

Adriana de Lanna Malta Tredezini

Diretora de Graduação

Maria Marta do Couto Pereira Rodrigues

Coordenador do Núcleo de Editoria e Publicações

Geovane Fernandes Caixeta

A *Revista do Fórum Gerencial* é uma publicação anual dos cursos de Administração, Ciências Contábeis, Gestão Comercial, Gestão Financeira, Gestão do Agronegócio, Gestão de Marketing, Gestão de Recursos Humanos, Gestão Pública, Sistemas de Informações e Comunicação Social do Centro Universitário de Patos de Minas (UNIPAM).

R454 Revista do Fórum Gerencial [recurso eletrônico] / Centro Universitário de Patos de Minas. – Dados eletrônicos. – v. 1, n. 2 (2021)- . – Patos de Minas : UNIPAM, 2021-

Irregular

Disponível em: <https://revistas.unipam.edu.br>

1. Gerência – periódicos. I. Centro Universitário de Patos Minas. II. Título.

CDD 650.130 5

Os textos publicados na Revista do Fórum Gerencial são de exclusiva responsabilidade de seus autores e não expressam necessariamente a opinião da Comissão Editorial e do Conselho Científico.

Editor responsável

Dra. Adriene Stéfane Silva – UNIPAM
Me. Gustavo de Magalhães Vieira – UNIPAM

Conselho Editorial Interno

Dra. Adriene Stéfane Silva, Centro Universitário de Patos de Minas – UNIPAM
Me. Ana Paula Lara de Vasconcelos Ramos, Centro Universitário de Patos de Minas – UNIPAM
Me. Carlen Fonseca Gonçalves, Centro Universitário de Patos de Minas – UNIPAM
Me. Claudio Roberto Vaz Teixeira, Centro Universitário de Patos de Minas – UNIPAM
Me. Heitor Cunha Barros, Centro Universitário de Patos de Minas – UNIPAM
Dr. Henrique Carivaldo de Miranda Neto, Centro Universitário de Patos de Minas – UNIPAM
Me. Guilherme Caixeta Borges, Centro Universitário de Patos de Minas – UNIPAM
Esp. João Wander Silva, Centro Universitário de Patos de Minas – UNIPAM
Me. Milton Roberto de Castro Teixeira, Centro Universitário de Patos de Minas – UNIPAM
Me. Mislene Dalila Silva, Centro Universitário de Patos de Minas – UNIPAM
Me. Pablo Fonseca da Cunha, Centro Universitário de Patos de Minas – UNIPAM
Me. Sandro de Paula Matias, Centro Universitário de Patos de Minas – UNIPAM

Conselho Editorial Externo

Esp. Marcos Geraldo Alves da Silva – Sebrae Minas
Dr. Nidelson Falcão – Fecoagro

Diagramação

Núcleo de Editoria e Publicações

SUMÁRIO

A CULTURA DEVOPS.....	07
Raphael Custodio Soares Henaldo Barros Moraes	
APLICAÇÃO MOBILE PARA GESTÃO DE AULAS EAD: UTILIZANDO AS TECNOLOGIAS OFFLINE FIRST E OPTIMISTIC USER INTERFACE.....	18
Edson Junio Malaquias Juliana Lilis da Silva	
APLICATIVO DE DELIVERY PARA EMPRESAS DO RAMO DA ALIMENTAÇÃO.....	29
Kevem Pedro Pereira Mesquita Martins Lima Juliana Lilis da Silva	
DESENVOLVIMENTO DE PULSEIRA INTELIGENTE PARA DETECÇÃO E PREVENÇÃO DE ATAQUES EPILÉPTICOS.....	43
Higor Férrer Henaldo Barros	
FRILA: APLICATIVO PARA PRESTADORES DE SERVIÇO.....	54
Adryel Klever Vieira de Almeida Mislene Dalila da Silva	
PLATAFORMA GERENCIAL PARA MARATONAS HACKATHONS.....	65
Lucas Bruno Ferreira Henaldo Barros Moraes	
QUICKCLASS: APLICATIVO PARA GERENCIAR FREQUÊNCIA UNIVERSITÁRIA VIA GPS.....	76
Gian Michel Nunes Fernandes Henaldo Barros de Moraes	
SISTEMA DE ANÁLISE DE SENTIMENTO EM COMENTÁRIOS RELACIONADOS A UNIVERSIDADES DE ENSINO SUPERIOR EM REDES SOCIAIS (VENERA).....	86
Wender Lucas Souza José Corrêa Viana Fernando Corrêa de Mello Júnior Henaldo Barros Moraes	
SISTEMA PARA GERENCIAMENTO DE ESTOQUE E FINANÇAS.....	100
Jefferson Nogueira Rocha José dos Reis Mota	

**UTILIZAÇÃO DE CHATBOT NO AUXÍLIO AO PROCESSO DE INCLUSÃO
DIGITAL DE PCDS..... 112**

Wellington Coêlho de Araújo

Henaldo Barros Moraes

**UTILIZAÇÃO E ORQUESTRAÇÃO DE CONTAINERS EM APLICAÇÕES
WEB..... 126**

João Antônio Caetano Rosa

José dos Reis Mota

A CULTURA DEVOPS

Raphael Custodio Soares¹
Henaldo Barros Moraes²

RESUMO: Este artigo tem intuito de apresentar a cultura *DevOps*, abordando as três maneiras que são seus pilares fundamentais como também exemplificar as principais ferramentas que esse novo movimento oferece

PALAVRAS-CHAVE: *DevOps*; três maneiras.

ABSTRACT: This article aims to present the DevOps culture, addressing the three ways its fundamental pillars are as well as exemplifying the main tools that this new movement offers.

KEYWORDS: DevOps; three ways.

1 INTRODUÇÃO

Apesar do movimento *DevOps* está crescendo aos poucos, ele já é uma cultura bastante distribuída. *DevOps* surgiu a partir de uma necessidade: simplificar as regras de negócios através dos esforços coordenados e colaborativos. Em outras palavras, *DevOps* é responder com rapidez as mudanças do mercado de trabalho com seus esforços. Esse movimento é projetado para garantir alta qualidade de software para que chegue aos seus consumidores finais com excelência na qualidade.

A mudança cultural traz consigo confiança, aprendizado contínuo, colaboração entre equipes e alta sinergia. Para que tudo isso ocorra em um intervalo pequeno de tempo e com sucesso, foi criado os pilares do *DevOps* que são chamadas as três maneiras.

De acordo com Antonio Muniz (2020): “A Primeira Maneira possui princípios e práticas que potencializam o fluxo rápido de desenvolvimento para operações, seu foco é executar ações para acelerar o fluxo da esquerda para a direita, visando reduzir o tempo da implantação”.

Segue abaixo os benefícios da primeira maneira.

- Tornar trabalho visível;
- Reduzir tamanho dos lotes e intervalos;
- Remover desperdícios e foco no cliente;
- Incorporar qualidade na origem.

DevOps é cinquenta por cento automação e cinquenta por cento colaboração. Neste primeiro pilar como mencionado os benefícios acima pode resumir todos esses atributos como em qualidade na origem que um termo essencial nesse movimento, sendo responsável pela entrega da experiência ao cliente.

De acordo com Gim Kane (2016): “A Segunda Maneira permite fluxo de *feedback* rápido e constante, do cliente para o desenvolvimento, em todos os estágios do fluxo de valor”.

¹ Aluno de Sistemas de Informações do UNIPAM. E-mail: raphaelcustodio94@gmail.com.

² Mestre em Redes de Computadores pela UFU. E-mail: henaldobarros@gmail.com.

Exige que amplifique o feedback para evitar que problemas ocorram novamente. “Para garantir a satisfação e um ótimo feedback, segue alguns princípios básicos da segunda maneira”.

- Telemetria;
- Desenvolvimento por Hipóteses e Testes A/B;
- Programação em pares;
- Programação sobre os ombros;
- Qualidade próxima da fonte.

O Teste A/B é um dos principais conceitos que se utiliza no *DevOps* referente à segunda maneira. O mesmo vem da área de marketing. Steve Krug (2020) relata que os testes de usabilidade há muito tempo e a ideia básica: se deseja saber se o software ou website é, suficientemente, fácil de usar, assista algumas pessoas enquanto navegam pelo software observando suas dificuldades.

Um dos casos de sucesso de Steven Krug sobre o teste A/B foi usado na Amazon. O teste foi baseado se o usuário compraria no site da Amazon com apenas um clique ou no modo convencional com três cliques. A partir dos resultados do teste, a Amazon adotou, em seu site, a compra em “um clique”, tornando mais fácil a compra pelo site, pois não precisa digitar informações pessoais. Na Amazon o teste A/B é de extrema importância quando se quer lançar um produto novo ou requer alterações de software sem ter um feedback assertivo.

A Terceira maneira de acordo com Antonio Muniz (2020) é mencionada como a criação de uma cultura de alta confiança, que permite correr riscos e potencializar o aprendizado contínuo, possibilitando a adoção de uma cultura de experimentação.

Seguem algumas práticas adotadas na terceira maneira:

- Cultura organizacional: confiança, desempenho e gestão sem culpa ou medo;
- Institucionalizar a melhoria do trabalho diário;
- Transformar descobertas locais em melhorias globais;
- Criar padrões de resiliência no trabalho diário;
- Cultura de aprendizado.

2 REFERENCIAL TEÓRICO

Nesta seção são apresentados conceitos e estudos realizados sobre a cultura *DevOps* e algumas técnicas desse movimento.

2.1 ENTREGA CONTÍNUA

Brian Docker (2020) relata que a entrega contínua permitirá que as equipes de desenvolvimento de software se movam mais rapidamente e se adaptem às necessidades dos usuários com mais rapidez, reduzindo o atrito inerente associado ao lançamento de alterações de software.

2.1.1 Integração Contínua

A integração Contínua é uma das principais rotinas da cultura *DevOps*. Aécio Pires (2019) demonstra que

Integração Contínua é uma prática que consiste em juntar o código de vários desenvolvedores de uma mesma aplicação com maior frequência, identificar problemas e corrigi-los mais rapidamente, resultando em menos bugs, automatização de testes, feedbacks mais frequentes e entrega mais rápida (PIRES, 2019).

2.3 DOCKER

Docker é um conceito que vem ganhando espaço no mercado de trabalho. Essa ferramenta tem o intuito de compartilhar recursos do kernel ao invés de duplicar como acontece nas máquinas virtuais, levando assim a um problema de performance.

Por ser uma ferramenta *open source*, qualquer pessoa pode visualizar o código e contribuir com melhorias para o *Docker*. “Isso traz maior transparência e faz com que correções de bugs e melhorias aconteçam bem mais rápido do que seria em um software proprietário com uma equipe bem menor e poucos cenários de testes” (FERNANDO; ANDRÉ, 2018).

2.4 JENKINS

Como o *DevOps* aborda bastante a parte de automação, deve-se usar o conceito de *Pipeline* e também *CI/CD* para que tudo ocorra de forma automatizada desde submeter o código até levar para produção.

Pipeline de acordo com Aécio Pires (2019) é uma execução em sequência de vários jobs, que podem ter dependência entre si. Um job que faz o *build* de uma aplicação web dependa da execução com sucesso do job anterior ou seja para passar para próxima etapa só será possível se estes jobs forem aprovados. Apenas o *pipeline* não é suficiente para consolidar a implantação e entrega contínuas. Para que tudo ocorra como previsto, a ferramenta Jenkins é peça fundamental para que haja sucesso.

O Jenkins é uma ferramenta de código-fonte aberto e desenvolvida em Java. Aécio Pires (2019) afirma que Jenkins pode ser usado para realizar o *Continuous Integration*, *Continuous Delivery* e *Continuous Deployment* em projetos de diversos tamanhos e com linguagens e tecnologias variadas, tais como: .NET, Ruby, Groovy, Grails, PHP, Python, Java e outros.

2.5 BALANCEADOR DE CARGA

Outro grande conceito que este estudo tem a oferecer é o de *open-source* ou simplesmente código livre que se baseia em uma licença, *General Public License* (GPL), que garante benefícios de software como.

- A liberdade de executar o programa, para qualquer propósito;

- A liberdade de estudar como o programa funciona;
- A liberdade de redistribuir cópias;
- A liberdade de aperfeiçoar o programa e liberar os seus aperfeiçoamentos.

Como se trata de um código livre, qualquer pessoa pode contribuir para que o software cresça ainda mais, além de personalizar seu próprio software. Segundo o autor Christian Leoanardo (2020), o simples fato de ser disponibilizado sob a licença GPL permitiu que outros programadores adotassem o projeto, passando a contribuir com melhorias e correções.

A partir desse ponto de vista, o projeto desenvolvido adotou a ferramenta *HAPROXY* que é disponibilizado pela licença GLP, de acordo com o site logz.io “The 5 Best Open Source Load” o *haproxy* se encontra na lista de melhores ferramentas para realizar o balanceamento de carga entre servidores.

O balanceador distribui cargas de trabalho em vários servidores virtuais aumentando a disponibilidade e a tolerância a falhas de seus aplicativos (AMAZON, 2020).

2.6 GIT

Um dos grandes requisitos para *DevOps* ou até mesmo para os desenvolvedores é saber como controlar a versão do código, com isso o *Git* se torna peça fundamental.

De acordo com o autor Christian Leoanardo (2020), o *Git* foi criado por Linus Torvalds em 2005 para desenvolver o kernel Linux. Também é usado como uma ferramenta de controle de versão distribuída importante para o *DevOps*.

Ryan Hodson (2020) afirma que o *GIT* é um Sistema de Controle de Versão (VCS) criado para uma única tarefa: gerenciar alterações em seus arquivos. Permite que você acompanhe todas as alterações pelas quais um projeto de software passa, bem como de onde essas alterações vieram”.

Como mencionado acima, o *Git* é uma ferramenta essencial para gerenciar grandes projetos, mas também pode abrir uma vasta gama de possibilidades para controlar o fluxo de trabalho pessoal como senhas salvas a partir de um bloco de notas.

2.7 AZURE

DevOps está cada vez mais presente no mercado, este artigo usou as ferramentas mais atualizadas até o momento, como por exemplo o conceito de nuvem ou popularmente conhecido como *cloud*.

Cloud surgiu como a transformação mais importante na computação em décadas o autor Kirshc (2020) retrata que “A nuvem está mudando a forma como as organizações planejam fornecer serviços a seus clientes para aumentar o ritmo dos negócios e oferecer serviços novos e inovadores”.

Todo o projeto será baseado em nuvem. Atualmente, existem inúmeros fornecedores que oferecem este serviço. Devido ao custo e benefício e, também, à alta disponibilidade, a plataforma escolhida foi Azure.

Azure é uma plataforma de computação em nuvem pública da Microsoft que oferece muitos serviços combinados com uma estimativa, análise, armazenamento e rede. George Prestonship (2020), menciona que os usuários podem escolher entre esses serviços para ampliar e dimensionar novos programas ou executar pacotes atuais, dentro da nuvem pública.

2.8 DYNATRACE

Para conseguir medir a experiência do usuário e receber um feedback coerente, o projeto utilizou o conceito de telemetria.

De acordo com Abel Rubio (2019), que se faz uma analogia com fórmula 1, a telemetria considera todo o sistema e permite que os dados coletados pelos sensores do carro sejam enviados ao computador dos engenheiros. É um sistema imediato. Conforme as coisas acontecem no carro, a equipe vai recebendo feedback para poder analisá-las e tomar decisões posteriormente.

Esse contexto se encaixa perfeitamente no ambiente em que o carro é o software. Na medida que o cliente vai encontrando erros no software, o programa *Dynatrace*, que utiliza o conceito de telemetria, coleta essas possíveis falhas de desenvolvimento e retorna para os programadores, para que se realize uma ação corretiva de bugs solucionando assim as falhas.

O *Dynatrace* tem por objetivo monitorar a aplicação em um ambiente de produção, homologação como também detalhar a infraestrutura como servidores, discos rígidos, cargas da unidade de processamentos tudo em tempo real.

2.9 TERRAFORM

O software para automação da infraestrutura usado em todo projeto para criar recursos na nuvem Azure de forma automática foi escolhido o *Terraform*, a partir dessa ferramenta é possível integrar várias nuvens ao mesmo tempo com apenas um código fonte, criando várias máquinas virtuais ou recursos individuais facilitando o desenvolvimento.

3 METODOLOGIA

O presente estudo foi desenvolvido a respeito da cultura *DevOps* voltado para o conceito das três maneiras e como se aplicam no ambiente de produção.

Foi construído um ambiente de produção abordando as três maneiras aplicando, também, as ferramentas do movimento *DevOps* para consolidar a ideia dos pilares para uma melhor análise.

Neste sentido, foram consultados livros, artigos e projetos desenvolvidos para que o usuário tenha uma melhor disseminação do conteúdo.

Em uma última etapa, observa o estudo como um todo desde eliminar funções manuais até usar a telemetria para ter um rápido *feedback* o que acelerar a integração contínua e entrega contínua, toda aplicação foi utilizado na plataforma Azure uma das mais conceituadas no meio de arquitetura em nuvem.

4 DESENVOLVIMENTO E RESULTADOS

Nessa primeira fase, o acesso ao portal Azure é fundamental devido ser a base de todo o projeto. Para inscrever na plataforma é bastante fácil, apenas ter em mãos um cartão de crédito sendo possível desfrutar de todo conteúdo que Azure oferece.

Em casos de uma conta estudantil, ganhe um saldo de até R\$200 mensais para analisar todos os serviços que a plataforma disponibiliza. A Figura 1 demonstra a página inicial como também seus principais recursos.

Figura 1: Apresentação do portal Azure



Fonte: Dados do trabalho.

Na Figura 1, é possível perceber alguns serviços em destaque na plataforma do Azure como grupos de recursos, máquinas virtuais e redes virtuais.

A criação do grupo de recursos é como se fosse a base das máquinas virtuais, sendo possível adotar um *IP Address* estático ou dinâmico, escolha da localização do grupo de recursos o que pode impactar diretamente o custo em torno do projeto e por último ajuste de regras de firewall.

Para que todo esse conteúdo do grupo de recursos seja criado rapidamente e com alta qualidade usou a ferramenta *Terraform* que por padrão tem função de *MultiCloud*, onde a ferramenta faz integração com vários fornecedores de nuvem com apenas um código.

O *Terraform* tem o objetivo de simplificar a infraestrutura como código, ao invés de criar vários grupos de recursos para cada serviço de nuvem, o que demanda tempo e trabalho. O *Terraform* usa um método de código universal, ganhando qualidade na origem, parte essencial do *DevOps*.

A Figura 2 tem o objetivo de apresentar o código usado para criação de um grupo de recursos usando *Terraform*.

Figura 2: Código para implantação do grupo de recursos

```

provider "azurerm" {
  version = "~>2.20.0"
  features {}
}

# Create a resource group
resource "azurerm_resource_group" "tccgroup" {
  name     = "tccgroup"
  location = "West Europe"
}

#Create Virtual Network
resource "azurerm_virtual_network" "mynetwork"{
  name                = "NetworkTCC"
  address_space       = ["10.0.0.0/16"]
  location             = "West Europe"
  resource_group_name = azurerm_resource_group.tccgroup.name
}

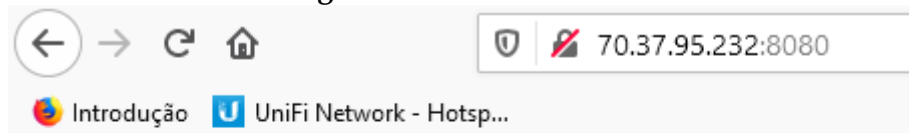
#Create Firewall Police
resource "azurerm_network_security_group" "tccnsgs"{
  name                = "tccsecuritygroup"
  location             = azurerm_resource_group.tccgroup.location
  resource_group_name = azurerm_resource_group.tccgroup.name
  security_rule {
    name                = "http"
    priority             = 100
    direction           = "Inbound"
    access               = "Allow"
    protocol             = "TCP"
    source_port_range    = "*"
    destination_port_ranges = [22,80,443]
    source_address_prefix = "*"
    destination_address_prefix = "*"
  }
}
    
```

Fonte: Dados do trabalho.

O próximo objetivo é criar máquina virtual utilizando o sistema operacional *Ubuntu* na versão 18.0. Após definir o host e a hora de criar contêineres, um para hospedar o website com intenção de *deploy contínuo* e um segundo contêiner para testar o balanceador de carga *haproxy*.

Para instalar o site que tem como propósito geral testar CI/CD deve-se primeiro aplicar as configurações do apache. Para isso baixe a imagem httpd com o comando "Docker pull httpd". Ao executar o comando, o Docker vai baixar a imagem para a máquina virtual e em seguida execute com "Docker run -name test -p 8080:80 httpd". Caso tudo ocorra perfeitamente, vai aparecer a imagem como na Figura 3.

Figura 3: Container Online



It works!

Fonte: Dados do trabalho.

O *GitHub* é responsável pelo controle de versão do projeto, servido como um repositório para armazenar um site e também por controlar as alterações feitas pela equipe de desenvolvimento, o projeto usou o *GitHub* para controlar a estrutura de código e também para aplicar o conceito de integração contínua e entrega contínua junto com o *Jenkins*. A Figura 4 abaixo demonstra a estrutura básica de um *commit*.

Figura 4: Commit no Github

Showing 1 changed file with 1 addition and 1 deletion.

```

  2  index.html
  ↑  @@ -3,7 +3,7 @@
  3  3
  4  4      <p> Digite o cupom de desconto </p>
  5  5      <p> Valor: </p>
  6  - 6      <p> Item: </p>
  7  + 6
  8  7      <head>
  9  8          <meta charset="utf-8">
  9  9          <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  ↓
  
```

Fonte: Dados do trabalho.

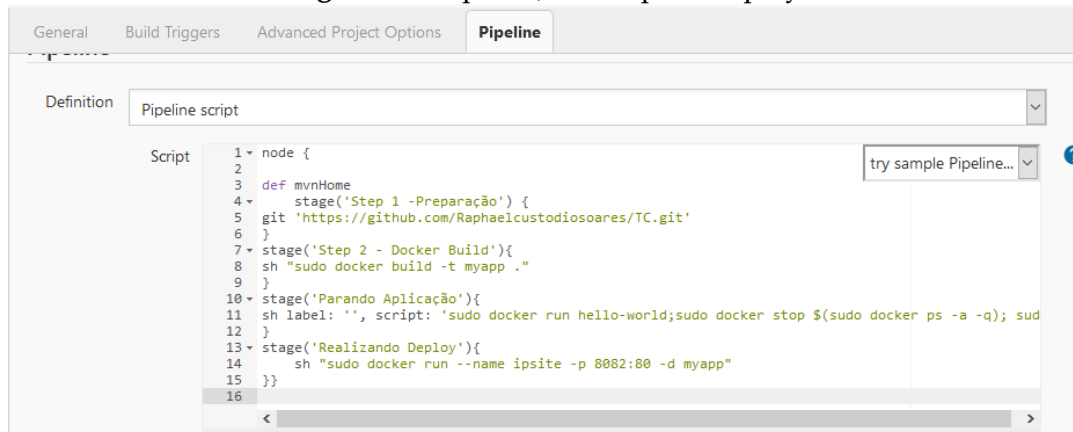
A Figura 4 representa um *commit* simples, porém deve destacar o item em vermelho e a linha em verde sendo também representadas pelos símbolos de menos ou mais.

A linha em vermelho significa que o texto foi retirado e em seguida acrescentado uma nova linha destacada com a cor verde. Pode dizer também que houve uma inclusão de código e uma exclusão. Esse *commit* demonstra apenas o básico. Hoje, os desenvolvedores utilizam para analisar cuidadosamente toda estrutura do projeto quando se trabalha em equipe, sabendo quem alterou, em qual linha e quando, em caso de bug, o que deixa o sistema inoperante é possível encontrar a falha rapidamente.

O *Jenkins* junto com o *GitHub* promove o *CI/CD*. Imagine que uma grande empresa de varejo onde suas vendas são realizadas através de um e-commerce está prestes a entrar na *Black Friday*. Antigamente para ajustar todos os preços do site, promoções, cupons de descontos, layout da página, menu personalizado entre outros até o horário da *Black Friday*, o site deveria ficar fora do ar para fazer todos os ajustes. Sendo assim, a perda da receita é extremamente alta por ser a promoção mais esperada do ano. Com *CI/CD* já é possível ajustar em tempo real as alterações, basta ter uma ferramenta que faça a integração entre os ajustes de homologação até o ambiente de produção.

A Figura 5 representa o código do *Jenkins* que busca no repositório *GitHub* e em seguida realizar o *Deploy* no *Docker*.

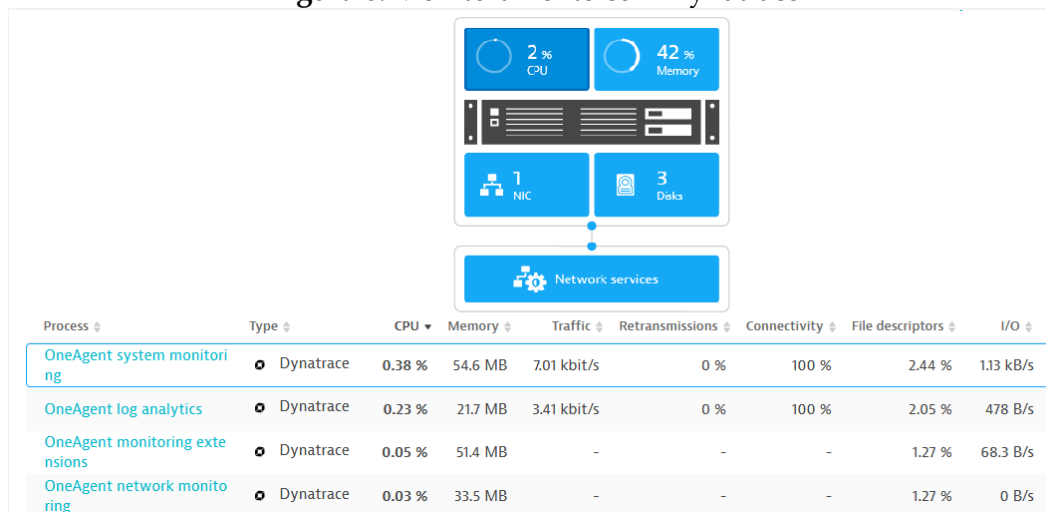
Figura 5: Script do Jenkins para Deploy



Fonte: Dados do trabalho.

Um grande fator para o *DevOps* é sempre receber feedbacks rápidos desde de clientes até mesmo da infraestrutura. *Dynatrace* foi usado para monitorar o ambiente de produção como por exemplo percentual de memória RAM, unidade de processamento e discos físicos. A Figura 6 apresenta todos os dados monitorados pela ferramenta *Dynatrace* em tempo real.

Figura 6: Monitoramento com Dynatrace



Fonte: Dados do trabalho.

5 CONCLUSÃO

O artigo demonstrado objetivou descrever a cultura *DevOps* como também a introdução das três maneiras, um passo fundamental para quem deseja seguir a profissão *DevOps*. Ao entender como funciona essa metodologia é possível estruturar um fluxo de projeto com mais rapidez e menos suscetível a falhas.

Com intuito de consolidar ainda mais o aprendizado, foi desenvolvido exemplos práticos em cada pilar do *DevOps*. As ferramentas utilizadas neste artigo são

as mais atualizadas no mercado de trabalho, o que proporciona um melhor aprendizado e uma melhor inclusão para quem deseja ingressar na carreira de *DevOps*.

Todas ferramentas mencionadas neste artigo podem se encontrar no *Gartner* que hoje é referência mundial em destaque de melhores ferramentas do mundo. A partir do Azure, foram criadas máquinas virtuais para consolidar o mecanismo de balanceamento de carga. As máquinas também foram utilizadas para demonstrar a integração da automação entre o Jenkins e *Github*, onde ao alterar no *Github* o site escrito em *Html* é totalmente modificado em segundos. Com o auxílio do balanceador de cargas, pode-se obter atualizações constantemente no site para atrair novos clientes e, também, proporcionar uma plataforma sem interrupção.

REFERÊNCIAS

AMAZON. **Elastic Load Balancing: User Guide**. Seattle 2020.

BROWN, Mike. **VMware vSphere 6.7 Data Center Design Cookbook: over 100 practical recipes to help you design a powerful virtual infrastructure based on vSphere**. 3. ed. Birmingham, 2019.

DOCKER, Brian. **KUBERNETES: a simple guide to master kubernetes for beginners and advanced users**. 2020.

FERNANDO, Jeferson; ANDRÉ, MARCUS. **Descomplicando Docker**. 2 ed. Rio De Janeiro, 2018.

LEVANTAMENTO do PageGroup mostra 38 profissões em alta em 2020. **G1**. 2020. Disponível em: <https://g1.globo.com/economia/concursos-e-emprego/noticia/2020/01/14/levantamento-do-pagegroup-mostra-38-profissoes-em-alta-em-2020.ghtml>. Acesso em: 17 ago. 2020.

HODSON, Ryan. **Ry's Git Tutorial**. Sugar Land. 2020.

HUMBLE, Jez; FARLEY, David. **Entrega contínua: como entregar software de forma rápida e confiável**. Poro Alegre: Bookman, 2014.

HURWITZ, Judith; KIRSCH, Daniel. **Cloud computing for dummies**. 2. ed. New Jersey. 2020.

JEE, Charlotte; MACAULAY, Thomas. **Falhas e invasões prejudicam empresas de diferentes setores**. 2018. Disponível em: <https://computerworld.com.br/2018/07/12/10-grandes-falhas-da-tecnologia-nos-ultimos-anos>. Acesso em: 28 set. 2020.

KIM, Gene; DEBOIS, Patrick; WILLIS, John; HUMBLE, Jez. **The DevOps Handbook: how to create world-class agility, reliability, and security in technology organizations**. IT Revolution Press. Portland, 2016.

KRUG, Steve; CROCE, DANIEL. **Não me faça pensar**. São Paulo: Alta Books, 2020.

LEONARDO, Christian. **Git**: a fast and easy guide to version control. 2020

MORIMOTO, Carlos. **LINUX, Guia Prático**, 2020.

MUNIZ, Antonio; SANTOS, Rodrigo; IRIGOYEN, ANALIA; MOUTINHO, RODRIGO **Jornada DevOps**: unindo cultura ágil, lean e tecnologia para entrega de software com qualidade. 2. ed. Rio De Janeiro: Brasport, 2020.

PIRES, Aécio; MILITÃO, Janaina. **Integração contínua com Jenkins**: automatize o ciclo de desenvolvimento, testes e implantação de aplicações. São Paulo: Novatec Editora. 2019.

PRESTONSHIP, George. **Azure**: essential guide to learn microsoft azure fundamentals, cloud, security, machine learning and devops. 2020.

RANJAN, Ankit. **Using the devops three ways to do laundry**. *Online*. Disponível em: <https://freshservice.com/itsm/phoenix-project-three-ways-devops-blog/>. Acesso em: 15 jul. 2020.

REBACK, Gedalyah. **The 5 best open source load balancers**. Disponível em: <https://logz.io/blog/best-open-source-load-balancers/>. Acesso em: set. 2020.

RUBIO Abel. **Telemetry in Formula 1**. 2019.

APLICAÇÃO MOBILE PARA GESTÃO DE AULAS EAD: UTILIZANDO AS TECNOLOGIAS *OFFLINE FIRST* E *OPTIMISTIC USER INTERFACE*¹

Edson Junio Malaquias²
Juliana Lilis da Silva³

RESUMO: Neste artigo, trata-se sobre o desenvolvimento de uma aplicação voltada para a área da *EdTech*, essa área realiza a junção da educação com a tecnologia. A aplicação tem como objetivo auxiliar alunos de instituições que estudam através da modalidade de educação a distância (EAD) progredirem com o aprendizado, realizando o estudo de diversos tipos de conteúdos e acompanhando o progresso acadêmico. Ela também permite que o aluno faça o download do conteúdo para o dispositivo móvel, fazendo com que ele tenha acesso mesmo sem ter uma conexão estável com a internet. Para a realização do desenvolvimento, foram utilizados alguns conceitos como o *offline first* e *optimistic user interface* juntamente com a linguagem de programação *React Native* e o banco de dados local *RealmDB*.

PALAVRAS-CHAVE: *Edtech; Offline First; React Native; RealmDB*.

ABSTRACT: This article is about the development of an application focused on the EdTech area, that combines education with technology. The application aims to help students from institutions that study through distance education progress with learning, carrying out the study of various types of content and monitoring academic progress. It also allows students to download the content to the mobile device giving them access even without having a stable internet connection. To carry out the development, some concepts were used, such as *offline first* and *optimistic user interface* together with the *React Native* programming language and the local database *RealmDB*.

KEYWORDS: *Edtech; Offline-First; React-Native; RealmDB*.

1 INTRODUÇÃO

O avanço da tecnologia nas últimas décadas vem revolucionando o modo como a educação é distribuída entre professores e alunos. Esse avanço permitiu a popularização da internet e, com isso, diversas áreas foram beneficiadas, sendo uma delas a educação a distância. A educação a distância (EAD) vem crescendo muito nos últimos anos o que gera um grande desafio para os fornecedores de conteúdo como a criação e disponibilização de conteúdos em larga escala.

De acordo com o censo realizado pela ABED (Associação Brasileira de Educação a Distância), no ano de 2018 foram matriculados um total de 2.358.934 alunos em cursos totalmente a distância, com um crescimento de aproximadamente 56%. E nos cursos semipresenciais, foram matriculados 2.109.951 alunos, com um crescimento

¹ Artigo apresentado com a finalidade de obter o título de bacharel de Sistemas de Informação.

² Estudante de Graduação do Curso de Sistemas de Informação do UNIPAM. E-mail: edsonmalaquias@unipam.edu.br.

³ Orientadora, Mestre em Ciência da Computação (UFU) e docente do UNIPAM. E-mail: juliana@unipam.edu.br.

aproximado de 53% comparado ao ano de 2017 (ABED, 2018). Assim, pode-se constatar que a quantidade de alunos matriculados em cursos EAD vem crescendo consideravelmente nos últimos anos e, também, a necessidade de boas conexões com a internet para acessar as aulas e os conteúdos dos cursos.

No entanto, percebe-se que no mundo, a conexão à internet faz com que a maioria das aplicações fiquem dependentes desta tecnologia praticamente o tempo todo. A dependência de conexão gera grandes problemas aos usuários, como o custo elevado para manter aplicações que precisam de uma conexão com a internet a todo momento. Outra situação que pode ser relatada é que pessoas de comunidades mais remotas geralmente têm um problema ainda maior, porque em muitos momentos elas não têm um bom sinal de internet e raramente conseguem se conectar a uma rede. As pessoas que passam por esses problemas evitam utilizar aplicações que consomem todos os seus dados móveis, como por exemplo, assistir uma aula remotamente.

Diante do contexto apresentado, vem surgindo diversas propostas para tentar solucionar alguns desses problemas com relação à conexão com a internet, como utilizar as aplicações em modo *offline* a maior parte do tempo, fazendo com que a única dependência com a internet seja por um breve período, somente para enviar os dados mais críticos da aplicação para um banco de dados.

Em meio a este cenário surgiu o conceito *offline first*, o qual visa resolver problemas de aplicações que dependem de internet a todo momento. A proposta desta tecnologia é que toda aplicação deve manter pelo menos suas funções essenciais funcionando independente da conexão de internet. Ao utilizar esta tecnologia, as pessoas que utilizam a aplicação não percebem que por um breve instante perderam a conexão ou, se perceberem, não ficam totalmente de mãos atadas sem poder utilizá-la. Neste modelo, no momento que essa pessoa se conectar novamente à internet as informações serão atualizadas.

Para atender a esta perspectiva, o objetivo geral deste estudo foi desenvolver uma aplicação mobile utilizando o *React Native* juntamente com o conceito *offline first*. A aplicação é destinada a alunos de instituições que trabalham com a modalidade EAD, possibilitando aos mesmos acessar as aulas por meio do celular, assistir às videoaulas e fazer os exercícios propostos. A aplicação também permite que o aluno faça o download de aulas para assistir offline. Assim, mesmo sem internet, o aluno pode utilizar a aplicação e acompanhar seu desenvolvimento acadêmico.

2 REVISÃO DE LITERATURA

Nesta seção são tratados temas relacionados ao projeto desenvolvido e que são de grande importância para entendimento do mesmo.

2.1 EDUCAÇÃO A DISTÂNCIA

De acordo com o Ministério da Educação (MEC):

Educação a distância é a modalidade educacional na qual alunos e professores estão separados, física ou temporalmente e, por isso, faz-

se necessária a utilização de meios e tecnologias de informação e comunicação. Essa modalidade é regulada por uma legislação específica e pode ser implantada na educação básica (educação de jovens e adultos, educação profissional técnica de nível médio) e na educação superior (MEC, 2009).

O aumento da educação a distância vem crescendo principalmente por oferecer maior flexibilidade, comodidade e um valor mais acessível. O aluno tem a chance de escolher o horário e o local em que vai realizar seus estudos, o que chama a atenção de diversas pessoas que querem atualizar seu currículo, porém têm o tempo muito corrido. As aulas a distância, por não precisarem disponibilizar um espaço físico e o deslocamento dos professores, faz com que o seu valor seja muito mais acessível do que os cursos presenciais e semipresenciais (SANTOS, 2018).

2.2 AMBIENTES VIRTUAIS DE APRENDIZAGEM

Hoje, na grande maioria dos casos, a educação a distância ou EAD é desenvolvida em ambientes virtuais de aprendizagem (AVA). Por meio destes ambientes ocorrem encontros entre alunos e professores algumas vezes por semestre. Normalmente os professores gravam suas aulas e disponibilizam no AVA, mas também é comum acontecerem aulas ao vivo. Já os cursos híbridos ou semipresenciais se dão por uma combinação dos cursos EAD e os cursos presenciais. O curso é considerado semipresencial se a carga horária presencial for maior que 20% (E+B EDUCAÇÃO; SILVA, 2018).

Os conteúdos de ensino são disponibilizados no ambiente virtual de aprendizagem, o qual é composto por um conjunto de ferramentas disponíveis na internet, sendo permitido também a interação entre os alunos e o acompanhamento dos alunos pelos professores. O objetivo do AVA é de simular uma sala de aula, assim, é possível realizar as atividades programadas, debater assuntos, visualizar o conteúdo das aulas e acompanhar todo o progresso através de relatórios (SILVA, 2020).

Devido ao cenário atual de crise gerado pela pandemia do Covid-19, os ambientes virtuais de aprendizagem tiveram um aumento significativo de acessos. Alguns países adotaram medidas de isolamento para evitar o espalhamento da doença, o que gerou a paralisação das aulas presenciais, com isso muitas instituições de ensino tiveram que se adaptar para conseguir fornecer seus conteúdos a distância se dirigindo para os AVA (SOUZA, 2020).

2.3 OFFLINE FIRST

Offline First é um conceito de arquitetura para aplicações mobile. Esse conceito propõe que a aplicação mantenha o funcionamento de suas principais funções independente da conexão com a internet. Sua implementação é de suma importância, pois desenvolver uma aplicação pensando que o usuário sempre terá conexão com a internet é um erro, pois muitos usuários sofrem com perda de sinal de internet ou até mesmo gastos exagerados em sua conta de telefone (FENANDES, 2018).

Optimistic User Interface é outro padrão muito utilizado com o *offline first* pois ele simula os resultados de uma ação em uma aplicação antes mesmo de receber a resposta do servidor. Esse padrão foi criado fornecendo uma maneira mais simples de fazer com que a interface do usuário responda mais rapidamente, e garantindo que os dados se tornem consistentes com a resposta real quando ela chegar do servidor (MICHEL, 2020).

3 METODOLOGIA

Durante o processo de desenvolvimento deste projeto foi utilizada a metodologia ágil Kanban. Normalmente, propõe a divisão de tarefas em etapas, as quais são colocadas em um quadro facilitando a visualização e compreensão para aqueles envolvidos no processo. As etapas normalmente são divididas em *to do* (a fazer), *doing* (fazendo) e *done* (feito), porém o quadro é bem flexível, permitindo que sejam realizadas adaptações de acordo com cada desenvolvimento. Desde o planejamento até a entrega de cada uma das tarefas, o quadro tem que ser mantido atualizado, realocando os *post-its* de acordo com a descrição de cada etapa.

Outros processos que foram utilizados são as técnicas comentadas na revisão de literatura, o *Offline first* juntamente com o *Optimistic User Interface*. Essas técnicas foram utilizadas para melhorar a *user interface* (UX), fazendo com que o usuário tenha uma melhor usabilidade da aplicação.

Durante o processo de desenvolvimento da aplicação foi necessário a utilização de algumas ferramentas para auxiliar e acelerar o processo de criação. A lista das ferramentas juntamente com a descrição de cada uma está apresentada no Quadro 1.

Quadro 1: Ferramentas e descrições

Ferramentas	Descrição
Trello	É uma ferramenta que possibilita organizar tarefas em quadros, listas e cartões. É muito utilizado em metodologias ágeis e foi utilizado para organização e acompanhamento do quadro Kanban.
Visual Studio Code (VS Code)	O VS Code é uma ferramenta para edição de códigos e foi utilizado para escrever os códigos em <i>React Native</i> .
<i>React Native</i>	<i>React Native</i> é uma linguagem de programação utilizada para o desenvolvimento de aplicativos móveis, foi utilizada para a criação das telas e lógica da aplicação.
<i>RealmDB</i>	<i>RealmDB</i> é um banco de dados criado especificamente para aplicações móveis, e foi utilizado para armazenar os dados offline da aplicação (DRUMOND, 2016).
Android Studio – Emulador	O Android Studio é uma plataforma para desenvolvimento de aplicações mobile android, porém foi utilizado somente o emulador para fins de acompanhamento de como o aplicativo estava sendo desenvolvido.
Lucid Chart	Lucidchart é um website que permite a criação de fluxogramas e modelagens em geral foi utilizado para criação da arquitetura do sistema.

Moqups	Moqups é um website que permite a criação de protótipos de interfaces e foi utilizado para a prototipação das interfaces da aplicação.
Git Hub	O Github é "um website que fornece serviço de hospedagem online e gerenciamento de código fonte, bem como controle de versão distribuído aos usuários" (BATISTA, 2017, p. 224). Este recurso foi utilizado para armazenamento e controle de versão do código fonte.

Fonte: Elaborado pelo autor, 2020.

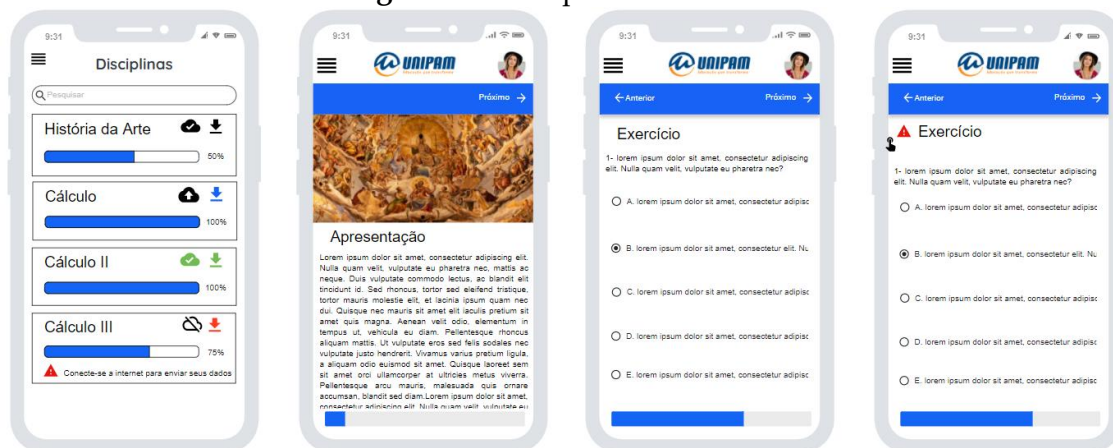
As pesquisas sobre as partes teóricas e práticas necessárias para o desenvolvimento do projeto foram feitas através de consultas e estudos de artigos acadêmicos, fóruns, sites e outros meios. Essas pesquisas proporcionaram o conhecimento necessário para desenvolver o projeto.

4 DESENVOLVIMENTO E DISCUSSÃO DOS RESULTADOS

Nesta seção são apresentados os pontos mais relevantes da construção do sistema, descrevendo desde a fase inicial com a prototipação até os resultados das principais funções do sistema.

Na fase inicial do desenvolvimento foi criada a prototipação das telas da aplicação com o Moqups, conforme demonstrado na Figura 1. Essa prototipação foi uma forma rápida de destacar as principais funcionalidades e validar algumas hipóteses antes de começar o desenvolvimento das funcionalidades e também funcionou como modelo para as telas da aplicação.

Figura 1: Protótipo das interfaces

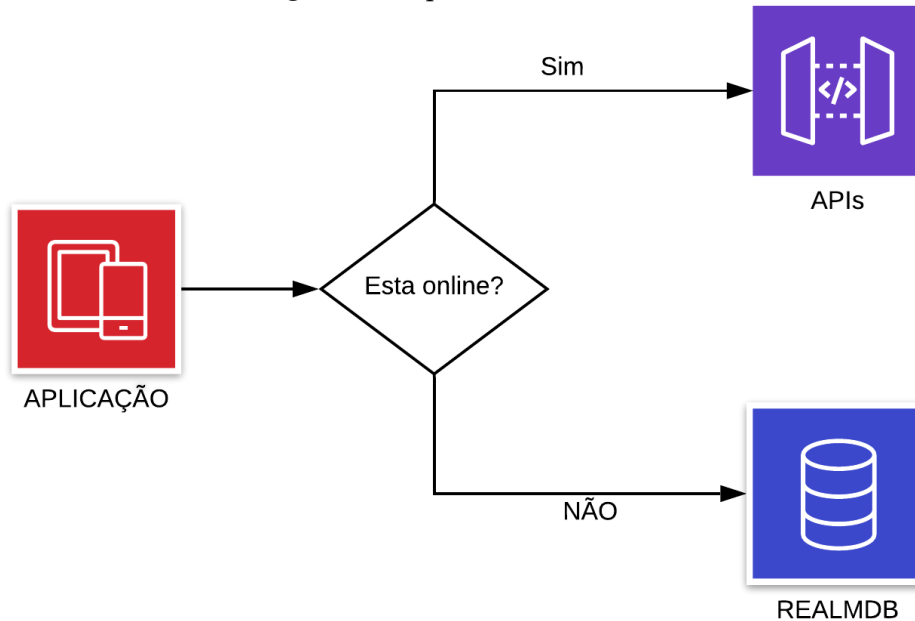


Fonte: Elaborado pelo autor, 2020.

Após a prototipação foi realizada uma análise de arquitetura para definir como a aplicação se comportaria quando estivesse offline. Essa análise foi baseada no conceito de arquitetura *offline first*, na qual o foco era a aplicação manter as suas principais funções, independente da conexão com a internet. O desenho dessa arquitetura foi criado utilizando o Lucid Chart e pode ser observado na Figura 2. Nesta Figura foi demonstrado os dois acessos aos conteúdos que são via API e RealmDB.

Assim, pode-se observar que a aplicação acessa o conteúdo por padrão na API, mesmo que o conteúdo esteja salvo no *RealmDB*, pois dessa forma sempre que o usuário estiver online é garantido que a fonte de dados que ele está utilizando é a mais atualizada e ao perder a conexão com a internet ele pode continuar o estudo do conteúdo acessando as unidades de aprendizagem que estão salvas no *RealmDB*.

Figura 2: Arquitetura do sistema



Fonte: Elaborado pelo autor, 2020.

Após ter os levantamentos de telas e arquitetura definidos, foram detectadas diversas tarefas a serem realizadas. Essas foram listadas no Trello para facilitar o controle do fluxo e especificação de quais tarefas eram prioritárias para as entregas do projeto. Também, foi levantada a demanda do aluno poder acessar o conteúdo de aprendizagem mesmo sem internet, pois vários alunos não têm uma conexão estável nos dispositivos móveis. Devido a essas tarefas e demandas foi realizado um estudo para selecionar algumas bibliotecas para ajudar a iniciar o desenvolvimento da aplicação, sendo elas:

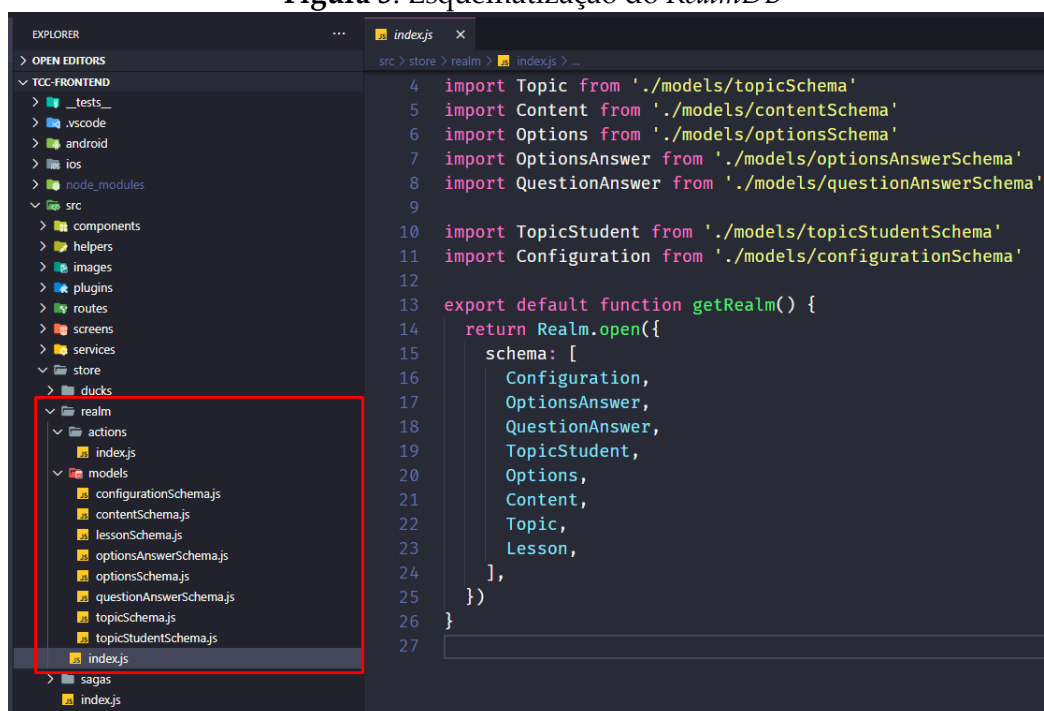
- *rn-fetch-blob*: proporcionou uma forma fácil de realizar o download dos arquivos contidos dentro das unidades de aprendizagem.
- *Cheerio*: foi utilizado para manipulação e validação dos conteúdos *HTML* (*Hypertext Markup Language*).
- *Redux*: para realizar o controle de estado, facilitando a comunicação de componentes dentro do *React Native*.
- *react-native-render-html*: foi muito utilizado na tela de conteúdo para ler os diversos tipos de conteúdo *HTML* e os transformar em elementos que o *React Native* conseguisse interpretar para realizar a renderização da interface.

Durante o desenvolvimento foi realizada a criação das telas em *React Native* seguindo os modelos de interface e de arquitetura comentados anteriormente. Para o

versionamento e armazenamento do código fonte foi utilizado o *GitHub*, pois fornece segurança no armazenamento, acesso rápido ao código e possibilita a criação de várias versões do código fonte. Nessa fase do desenvolvimento, foram criados os esquemas do *RealmDB* que definiram como o conteúdo seria salvo dentro do banco de dados local.

Como comentado, o banco de dados local da aplicação é o *RealmDB*. Este foi estruturado em um arquivo de controle, *models* e *actions*. Essa estrutura pode ser observada na Figura 3, onde a direita é mostrada a função *getRealm*, que retorna a instancia da base de dados, e à esquerda é demonstrada a divisão de pastas das *actions* e *models*.

Figura 3: Esquematização do *RealmDB*



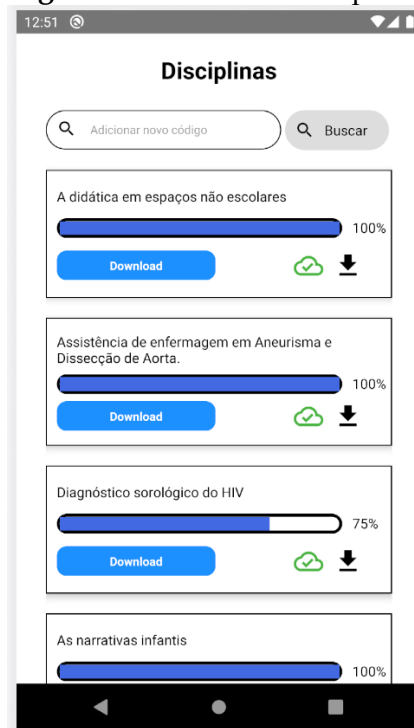
Fonte: Elaborado pelo autor, 2020.

O arquivo de controle mostrado na Figura 3 é responsável por criar uma nova instância ou retornar a instância já existente do banco de dados. Essa instância é esquematizada pelas configurações contidas nos esquemas. Os esquemas são salvos dentro da pasta *models* e são responsáveis por definir como os dados são salvos no *realmDB* e como se relacionam com os outros esquemas. As *actions* são responsáveis por intermediar toda a comunicação do banco de dados com o resto da aplicação, nelas ficam as lógicas de controle das operações do *RealmDB* como inserção, atualização, exclusão e busca.

A Figura 4 mostra a tela da central das disciplinas onde o aluno tem acesso a suas unidades de aprendizagem. Nessa tela, ele pode visualizar o progresso da unidade, adicionar novas unidades e também baixar grande parte do conteúdo através da opção de download.

Para realizar a função de download, todo o conteúdo *HTML* (*Hypertext Markup Language*) é manipulado utilizando a biblioteca *cheerio* seguindo a esquematização do banco de dados da aplicação e sendo salvo no *RealmDB*. Os arquivos são salvos no armazenamento interno do celular com o auxílio da biblioteca *rn-fetch-blob*. Essa função permite ao aluno acessar o conteúdo em qualquer lugar, independente da conexão com a internet. Lembrando que todo o conteúdo será sincronizado com a API quando o usuário tiver novamente uma conexão com a internet.

Figura 4: Central das disciplinas

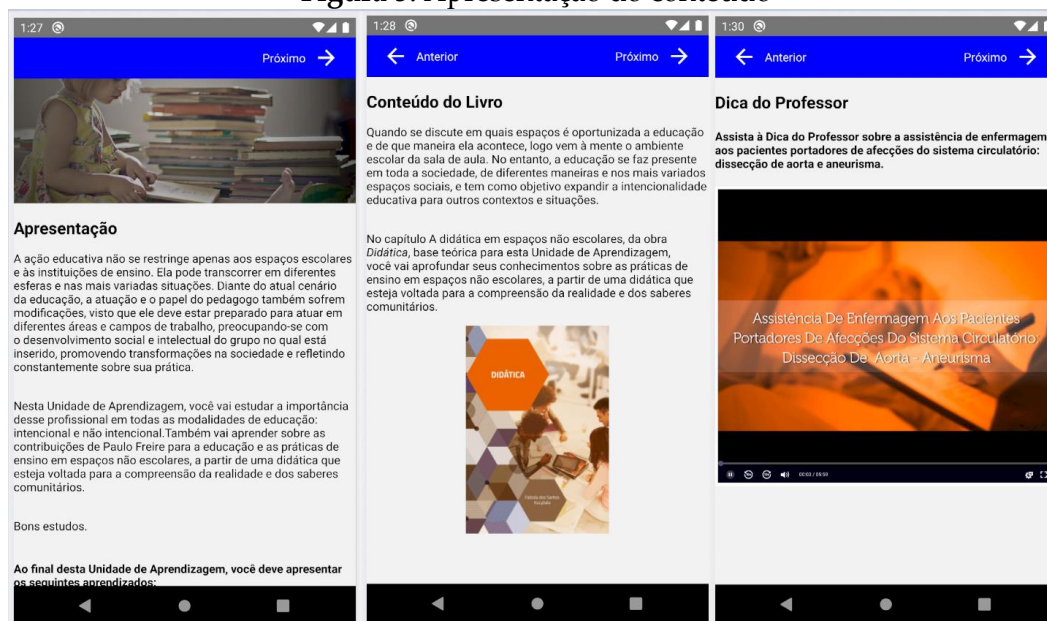


Fonte: Elaborado pelo autor, 2020.

Ao acessar uma unidade de aprendizagem o aluno entra na tela de conteúdo, conforme exemplificado na Figura 5. Nesta tela, o conteúdo dispõe de vários elementos para dinamizar a aprendizagem e tornar mais agradável a experiência do aluno no contexto de estudo de um curso EAD.

Pode-se observar que os conteúdos são abordados em vários formatos, como textos, imagens, vídeos, livros em *pdf* e conteúdos interativos que são *iframes*. Alguns dos elementos citados são representados por uma *tag HTML* (*Hypertext Markup Language*) customizada.

Figura 5: Apresentação do conteúdo



Fonte: Elaborado pelo autor, 2020.

Para realizar a renderização dos elementos que compõem as unidades de aprendizagem, como no exemplo mostrado na Figura 5, houve um desafio que era conseguir renderizar *tags* padrões do *html* e *tags* customizadas que vem da *API* e do *RealmDB*.

Para solucionar esse problema foi utilizada a biblioteca *react-native-render-html* que realiza a leitura desse conteúdo transformando-o em elementos reconhecidos pelo *React Native*. E para as *tags* customizadas, essa biblioteca permite que o desenvolvedor crie elementos específicos do *React Native* para que ela consiga realizar a transformação, assim fazendo com que o *React Native* consiga compilar o conteúdo e realizar a renderização da unidade de aprendizado.

5 CONSIDERAÇÕES FINAIS

Na seção anterior foi demonstrado o desenvolvimento do aplicativo e como pode-se observar ele atingiu o seu objetivo principal que era a criação de um aplicativo mobile desenvolvido em *React Native*. O aplicativo teria que permitir ao aluno realizar o estudo de sua unidade de aprendizagem com a possibilidade de acessar o conteúdo mesmo sem internet. Essa função de acesso offline foi disponibilizada por um botão de download de fácil acesso na tela das disciplinas.

Como melhorias futuras, pretende-se desenvolver alguns elementos específicos que, por terem uma demanda menor, não foram priorizados, que são os componentes de áudio e fórmulas. Outra melhoria seria a conclusão do elemento de vídeo para ser salvo durante o download do conteúdo e também ser disponibilizado offline. Um fluxo bastante interessante de ser desenvolvido no futuro é a criação de filas para as chamadas de APIs, pois quando a aplicação ficar offline ela armazenaria essas chamadas para serem realizadas em ordem quando a aplicação retomar sua

conexão. Assim, a universidade poderia acompanhar o progresso do aluno mesmo que a internet do aluno caia por alguns minutos.

O aplicativo acabou sendo uma boa alternativa de ambiente virtual de aprendizagem por ter a função de acesso offline, a qual muitos desses ambientes não fornecem. E para os alunos funciona como mais uma forma de acesso às unidades de aprendizagem, assim podendo concluir sua formação no ensino a distância com maior comodidade.

REFERÊNCIAS

ABED. **Censo EAD.BR**: relatório analítico da aprendizagem a distância no Brasil. 2018. Disponível em: http://abed.org.br/arquivos/CENSO_DIGITAL_EAD_2018_PORTUGUES.pdf. Acesso em: 16 mar. 2020.

BATISTA, Natércia A. *et al.* **GitSED**: um conjunto de dados com informações sociais baseado no GitHub. *In*: SBBB-Dataset Showcase Workshop. 2017. p. 224-233.

DRUMOND, DOUGLAS. **Iniciação ao Realm**: o que é o Realm?. 2016. Disponível em: <https://medium.com/android-dev-br/iniciacao-ao-realm-b7624e380194/>. Acesso em: 19 out. 2020.

EaD, PRESENCIAL E SEMIPRESENCIAL: qual é a diferença? 2018. Disponível em: <https://www.educamaisbrasil.com.br/educacao/noticias/ead-presencial-e-semipresencial-qual-e-a-diferenca>. Acesso em: 29 mar. 2020.

FERNANDES, Diego. **Offline first**: como funciona e como aplicar no React Native?. 2018. Disponível em: <https://blog.rocketseat.com.br/offline-first-react-native/>. Acesso em: 29 mar. 2020.

MEC. **O que é educação a distância?**. 2009. Disponível em: <http://portal.mec.gov.br/escola-de-gestores-da-educacao-basica/355-perguntas-frequentes-911936531/educacao-a-distancia-1651636927/12823-o-que-e-educacao-a-distancia>. Acesso em: 31 mar. 2020.

MICHEL, MAYKON. **Optimistic UI**. 2020. Disponível em: <https://ezdevs.com.br/offline-first-com-react-native/>. Acesso em: 29 mar. 2020.

SANTOS, Barbara. **Dados e tendências da educação a distância no Brasil**. 2018. Disponível em: <https://blog.hotmart.com/pt-br/educacao-a-distancia-no-brasil/>. Acesso em: 29 mar. 2020.

SILVA, Rafael Silvério. **O que é o ambiente virtual de aprendizagem?**. 2020. Disponível em: <https://eadbox.com/o-que-e-ambiente-virtual-de-aprendizagem/>. Acesso em: 29 mar. 2020.

SOUZA, Lays Serpa. **Os dilemas na implementação do EAD em universidades frente à Covid-19**. 2020. Disponível em: <https://www.jota.info/opiniao-e-analise/artigos/os-dilemas-na-implementacao-do-ead-em-universidades-frente-a-covid-19-10042020/>. Acesso em: 21 abr. 2020.

APLICATIVO DE DELIVERY PARA EMPRESAS DO RAMO DA ALIMENTAÇÃO¹

Kevem Pedro Pereira Mesquita Martins Lima²

Juliana Lilis da Silva³

RESUMO: O presente estudo trata de uma aplicação *web* e outra *mobile* desenvolvidas para a empresa de *delivery* Me Petisque situada em Patos de Minas. A empresa utiliza o aplicativo WhatsApp e o *marketplace* iFood para atender aos seus clientes, o que dificulta o atendimento e o aumento de lucros. Nesse sentido, o objetivo do desenvolvimento das aplicações *web* e *mobile* foi entregar à empresa Me Petisque uma plataforma que permitisse um melhor atendimento aos clientes, aumentando o número de atendimentos e, conseqüentemente, aumentando os lucros da empresa. No desenvolvimento foi utilizado o framework NestJS na parte do *backend* da aplicação, a biblioteca React Native na aplicação *mobile* e a biblioteca ReactJS na aplicação *web*. A linguagem utilizada foi o JavaScript e o banco de dados escolhido para armazenamento dos dados foi o *PostgreSQL*. Para facilitar e agilizar o desenvolvimento, a metodologia utilizada seguiu as premissas do *Scrum*, sendo utilizados alguns artefatos e eventos do mesmo. Como resultado, a empresa de *delivery* em questão tem em mãos uma ferramenta que pode impulsionar seu negócio.

PALAVRAS-CHAVE: *Delivery*; Aplicações Híbridas; JavaScript.

ABSTRACT: The present study it's a *web* and a *mobile* application developed to the *delivery* company Me Petisque located in Patos de Minas. The company use the WhatsApp application and the iFood marketplace to serve its customers, which makes it difficult to service and increase profits. In this sense, the objective of developing *web* and *mobile* applications was to provide the company Me Petisque a platform that allows better service to customers, increasing the number of services and, consequently, increasing the company's profits. In development was used the framework, NestJS in the *backend* part of the application, library React Native in the *mobile* application and library ReactJS in the *web* application, the language used was JavaScript and the database chosen for data storage was *PostgreSQL*. To facilitate and speed up development, the methodology used followed the premises of *Scrum*, using some artifacts and events from it. As a result, the *delivery* company in question has a tool in hand that can boost your business.

KEYWORDS: *Delivery*; Hybrid Applications; JavaScript.

1 INTRODUÇÃO

Em decorrência da mudança no estilo de vida da população nas últimas décadas, ocorreu um aumento significativo da alimentação fora do domicílio. Alguns

¹ Artigo apresentado como avaliação parcial da disciplina de Estágio Supervisionado II, para obtenção de título de bacharel em Sistemas de Informação, pelo Centro Universitário de Patos de Minas – UNIPAM.

² Estudante de Graduação do Curso de Sistemas de Informação do UNIPAM. E-mail: kevempedro@unipam.edu.br.

³ Orientadora, Mestre em Ciência da Computação pela UFU e professora do UNIPAM. E-mail: juliana@unipam.edu.br.

fatores são responsáveis, como serviços fornecidos pelos *fast foods*, *self services* e *deliveries* (LEAL, 2010).

No mercado de *delivery* é de suma importância que os donos de estabelecimentos apostem na tecnologia para aumentar suas receitas e fidelizar seus clientes. Como pode ser observado, o mercado americano absorveu uma grande quantidade de downloads de aplicativos próprios dos estabelecimentos e ultrapassou em muito a dos *marketplaces*. No futuro, é provável que todo estabelecimento que optar por *delivery* tenha seu próprio aplicativo para se aproximar mais do seu cliente e fidelizá-lo melhor (OLD, 2017).

Ao fazer uma busca no Google pelo tema “mercado de *delivery* no Brasil”, a maioria das notícias são sobre o crescimento do setor nos últimos anos. De acordo com dados do Sebrae, em 2017 o *delivery* movimentou mais de R\$9 bilhões e cresce na casa dos 12% ao ano. A projeção do mercado de *delivery* no mundo também é bastante positiva. O *delivery* online, por exemplo, movimentou US\$ 82,2 milhões em 2018 e a taxa média de crescimento para o setor é de 9,3% ao ano, resultando em um volume de US\$ 134,5 milhões até 2023 (FRANÇA, 2019).

No cenário brasileiro, atualmente, existem grandes *marketplaces* atuando no mercado de *delivery* online, como UberEats, Rappi e um dos principais, o iFood, que lidera o mercado brasileiro. Percebe-se que este mercado vem ganhando força ano a ano e revolucionando o mercado de alimentos em todo mundo. Porém, *marketplaces* como iFood, UberEats e Rappi nos últimos tempos vem desagradando os donos de estabelecimentos que fazem uso dessas plataformas.

As principais reclamações das empresas que trabalham com *delivery* e utilizam dos *marketplaces* atuais é a falta de uma plataforma que permita uma melhor fidelização dos clientes, pois neste segmento geralmente não existe a identidade visual da empresa. Portanto, as dificuldades que as empresas que estão no mercado de *delivery* online encontram são: enviar notificações personalizadas na hora que elas quiserem para clientes específicos, lançar promoções de forma imediata e sem perda de tempo, baixa personalização da identidade digital e o alto custo cobrado pelos grandes *marketplaces*. Este conjunto de fatos impulsiona as empresas a criarem suas próprias plataformas, com o objetivo de atender suas necessidades e garantir um melhor atendimento para seus clientes.

Observando esse cenário, este artigo tem como objetivo apresentar o desenvolvimento de um aplicativo para as plataformas Android e IOS, junto com um painel *web*, para a empresa Me Petisque localizada em Patos de Minas. Esta empresa atualmente atende seus clientes pelo WhatsApp e iFood. No aplicativo proposto, o cliente do Me Petisque pode fazer seu pedido, ganhar cupons descontos e escolher uma forma de pagamento. Já pelo painel, o Me Petisque pode gerenciar seus produtos no aplicativo, receber e acompanhar os pedidos e visualizar gráficos que os ajudem nas tomadas de decisões.

Para que o objetivo geral fosse atingido, foram atendidos os seguintes objetivos específicos:

- *Cadastrar Clientes*: o aplicativo realiza o cadastro do cliente, cadastro esse que é utilizado para a realização dos pedidos.
- *Autenticar*: tanto o aplicativo quanto o painel realizam a autenticação.

- *Cadastrar Cupons*: Pelo painel é possível cadastrar cupons de desconto para serem utilizados pelos clientes em seus pedidos.
- *Visualizar Gráficos*: através do painel é possível visualizar alguns gráficos que ajudam o Me Petisque a tomar decisões na hora de lançar uma promoção ou desconto.

O sistema tem por objetivo também proporcionar o crescimento na disputa pelo mercado de *delivery* online para empresas como o Me Petisque, que buscam cada dia mais sua sobrevivência nesse ramo e não podem ficar à mercê de grandes *marketplaces*, os quais podem levar as mesmas à falência (MADUREIRA, 2020) e/ou diminuir seus lucros. Além dos fatos apresentados, não oferecerem uma plataforma que atenda às necessidades como lançar notificações de uma promoção ou desconto em um momento de necessidade da empresa em tempo oportuno e sem oneração, enviar um cupom de desconto para o cliente aniversariante do mês ou que seja um cliente consolidado e fazer um marketing direcionado para um grupo de cliente específico.

Garantir a fidelização dos clientes é tão importante quanto conquistar novos clientes. Tanto a fidelização quanto a busca por novos clientes são fundamentais para que a mesma se torne referência e continue atuando no mercado por muito tempo (MORAES, 2018), para isso a empresa precisa apostar na tecnologia e investir em ferramentas que proporcionem uma melhor tomada de decisão.

Todavia, a criação do aplicativo e do painel proporciona para o Me Petisque uma melhor gerência de seu negócio. O aplicativo permite aos clientes da empresa uma maneira mais fácil e ágil de realizar seus pedidos e o painel entrega a empresa uma melhor forma de gerenciar seu negócio, possibilitando assim que a mesma se mantenha no mercado de *delivery* online.

2 REVISÃO DE LITERATURA

Nessa seção são apresentados conceitos referentes à Aplicações Híbridas, Gerenciador de estado Redux e Sistemas de Informações Gerenciais - SIG.

2.1 APLICAÇÕES HÍBRIDAS

Com a demanda cada vez maior de usuários por dispositivos móveis, surgiram novas empresas. Cada uma com sua proposta de aparelho e com isso surgem também novas plataformas e sistemas operacionais, o que impacta no desenvolvimento de aplicações, para atender a todos os dispositivos, pois é necessário ter conhecimentos específicos de cada plataforma, para que seja desenvolvida uma aplicação diferente para cada uma, implicando também em custos e tempo mais elevados para o desenvolvimento (PREZOTTO, 2014).

Neste cenário surgem as aplicações híbridas, onde se desenvolve apenas uma aplicação, e esta pode ser utilizada em vários dispositivos com diferentes sistemas operacionais. Isso é possível devido aos frameworks direcionados para o desenvolvimento de aplicações híbridas, que empacotam o código-fonte para as diferentes plataformas, permitindo que elas possam ser instaladas no dispositivo e sejam capazes de acessar diferentes recursos, como contatos, câmera e GPS.

Aplicações híbridas possuem como finalidade funcionar em qualquer dispositivo, em que independentemente da plataforma, será utilizado o mesmo código-fonte. As mesmas possuem acesso a todos os recursos do dispositivo, o que permite aos desenvolvedores criarem aplicações com os mesmos recursos disponíveis para a criação de aplicações nativas. Diferente das aplicações nativas, que utilizam das linguagens específicas de cada plataforma, como no Android em que se faz uso do *Java* ou *Kotlin*, e no IOS *Objective-C* ou *Swift*, as aplicações híbridas utilizam do *Hypertext Markup Language - HTML*, *Cascading Style Sheets - CSS* e *JavaScript*, independentemente da plataforma.

Aplicações desenvolvidas nos frameworks que oferecem um desenvolvimento híbrido são instaladas nos dispositivos e podem ser disponibilizadas juntamente com as aplicações nativas nas lojas de aplicativos, no caso de Android (Play Store) e para o IOS (Apple Store). Apesar das aplicações híbridas conseguirem entregar bons resultados, um dos principais motivos para sua escolha é o baixo custo de tempo de desenvolvimento, já que o código-fonte é o mesmo, impactando inclusive na manutenção da aplicação.

Atualmente existem diversos frameworks híbridos, como o Ionic, React Native e Flutter por exemplo, que são alguns dos principais frameworks do mercado (SILVA, 2018). Cada um oferece recursos, tanto compartilhados quanto específicos, a escolha do framework para o desenvolvimento da aplicação varia de acordo com os requisitos de cada projeto ou preferência da equipe de desenvolvimento.

2.2 GERENCIADOR DE ESTADO REDUX

Quando se fala em desenvolvimento utilizando bibliotecas como ReactJS ou React Native, é imprescindível não utilizar componentes para facilitar a reutilização e manutenção de código. Contudo ao se utilizar componentes em uma aplicação, independente se a biblioteca escolhida for ReactJS ou React Native, é preciso se ter a preocupação de como esses componentes vão compartilhar os diferentes estados da aplicação. Para facilitar o compartilhamento desses estados entre os componentes pode-se utilizar o Redux.

Redux é um contêiner de estado para aplicações JavaScript, o estado é mantido em uma (*store*) loja e os componentes que estão interessados nos dados da loja ficam observando-a e caso haja alguma alteração nesses dados o componente reage de acordo com a mudança. O Redux implementa o padrão Flux que gerencia o fluxo de dados da aplicação (WECK, 2017).

Esse fluxo é dividido em três etapas: *store*, *reducers* e *actions*. A *store* é onde ficam armazenadas as informações e possui disponibilidade para receber e entregar para o componente o que ele requisita. Quando algum componente precisa alterar algum dado o *reducer* é encarregado de lidar com isso. Já as *actions* são responsáveis por requerer ao *reducer* uma ação de acordo com o tipo da mesma.

Com isso se houver uma alteração no valor de um produto do sistema por exemplo, é disparado uma *action* para o *reducer* que através do tipo dessa *action*, nesse caso é uma alteração no produto, altera o valor do produto na *store* e a mesma é responsável por enviar esse dado alterado para as *views* ou componentes que observam

essa *store*. Portanto, com a utilização do Redux torna-se muito mais fácil o compartilhamento dos diferentes estados da aplicação entre os diferentes componentes da mesma.

2.3 SISTEMAS DE INFORMAÇÕES GERENCIAIS - SIG

Existem diferentes definições sobre a expressão sistemas de informações gerenciais dentre as várias, pode-se definir a mesma como um método organizado que fornece informações passadas, presentes e futuras, relacionadas com as operações internas e serviços de inteligência externa. Dão suporte ao planejamento, controle e operação de uma empresa, a partir do fornecimento de informações, no tempo apropriado, para auxiliar o tomador de decisão (OLIVEIRA, 2012).

Sistemas de Informações Gerenciais para auxiliar o gerenciamento das atividades de um negócio é algo de extrema importância nos dias de hoje, bem como para auxiliar no planejamento estratégico do mesmo. O sistema de informação, combinado com a pesquisa de mercado, fornece informações valiosas para que a empresa tome decisões importantes e precisas diante do mercado em que se encontra, atuando em todos os níveis de uma organização: estratégico, tático e operacional. É de uma grande importância que os sistemas de informação forneçam informações precisas para uma adequada tomada de decisão.

Antigamente sistemas de informação apoiados por computadores eram opções caras e acessíveis para poucos, além de terem limitações graves e serem muito simples. Atualmente, a realidade é outra, pois com o avanço tecnológico e com o surgimento de equipamentos mais poderosos os sistemas agora são complexos e conseguem fornecer informações precisas para o apoio às decisões gerenciais.

A Informação nos dias atuais está cada vez mais disponível para todos e em grande abundância, podendo ser acessada de forma instantânea e ser consumida a qualquer hora. Portanto, é cada vez maior a importância de que, gestores ou responsáveis por uma empresa tenha em mãos ferramentas que os auxiliem em uma tomada de decisão gerencial mais precisa e correta, proporcionando uma vantagem estratégica em relação aos seus concorrentes (DOS SANTOS CLARO, 2013).

Para se desenvolver um sistema que seja eficaz é preciso diferenciar conceitos como: dados, informação e conhecimento. Esses conceitos muitas vezes são confundidos ou tratados como a mesma coisa, entretanto, cada um tem o seu significado e importância. Dados são registros soltos, ainda não tratados e sem sentido, representa a matéria-prima da informação, ou seja, a informação sem tratamento que ainda não tem relevância, com isso o dado por si só não é capaz de transmitir nenhum conhecimento. Informação representa a estrutura ou organização dos dados, ela é um registro derivado dos mesmos e também é insumo para o conhecimento. Já o conhecimento é a informação processada, que pode ser utilizada para criar, inovar, progredir, ou seja, o conhecimento é uma ferramenta de grande valor que permite pessoas evoluírem, assim como as empresas (REZENDE, 2015).

Portanto, um software, aplicativo ou afins, apesar de serem importantes para automatizar os serviços e processos de uma empresa, se os mesmos não forem capazes de gerar dados com qualidade, para serem processados e gerar informações que

possam ser transformadas em conhecimento para que a empresa possa evoluir com seus negócios, então talvez esse software ou aplicativo não tenha tanta relevância para a mesma.

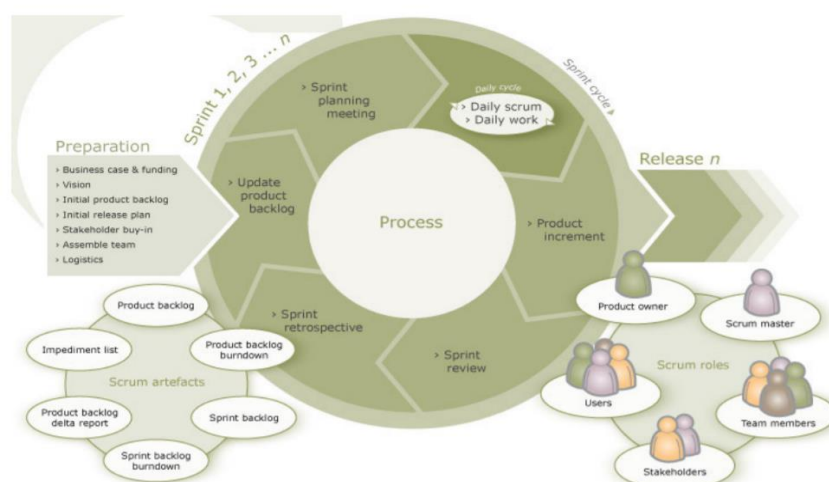
3 METODOLOGIA

A metodologia utilizada no desenvolvimento do artigo segue as premissas do Scrum. O Scrum é um framework para desenvolver e manter produtos. Desenvolvido por Ken Schwaber e Jeff Sutherland, o Scrum não é um processo ou uma técnica para construir produtos, mas sim um framework dentro do qual se emprega vários processos ou técnicas (SCHWABER E SUTHERLAND, 2013).

O framework Scrum consiste nos times do Scrum, cada um associado aos seus papéis, artefatos, eventos e regras. Cada um desses componentes dentro do framework serve a um propósito, sendo fundamentais para o sucesso do Scrum. Na Figura 1 pode-se observar que o Scrum é constituído de vários artefatos e eventos que facilitam o andamento de um dado processo ou projeto. Para o desenvolvimento desse projeto foram consideradas *sprints* de 4 semanas. Dos artefatos e eventos que o Scrum oferece, foram utilizados:

- *Product Backlog*: consiste em uma lista ordenada das funcionalidades necessárias no projeto, normalmente organizada pelo valor que cada funcionalidade entrega.
- *Sprint Backlog*: um conjunto de algumas das funcionalidades do *Product Backlog* que são selecionadas para serem desenvolvidas durante uma *Sprint*.
- *Sprint*: é um time-boxe de duração de um mês ou menos, no qual os itens da *Sprint Backlog* são desenvolvidos, gerando uma versão incremental do produto.
- *Increment*: é onde todos os itens do *Product Backlog* desenvolvidos durante a *Sprint* serão somados aos itens das outras *Sprints*, incrementando mais valor ao já desenvolvido.

Figura 1: Fluxo Scrum



Fonte: BISSI, 2007.

O Quadro 1 apresenta as sprints definidas e executadas durante o desenvolvimento do projeto.

Quadro 1: Estruturação do desenvolvimento do projeto seguindo o framework Scrum

Sprints	Atividades
Sprint 1	<ul style="list-style-type: none"> · Desenvolver a estrutura das aplicações <i>backend</i>, <i>frontend</i> e <i>mobile</i>. · Desenvolver o CRUD de usuários e administradores. · Autenticar no painel. · Cadastrar usuários pelo aplicativo.
Sprint 2	<ul style="list-style-type: none"> · CRUD de produtos. · Apresentar produtos no painel e aplicativo.
Sprint 3	<ul style="list-style-type: none"> · Autenticar no aplicativo. · Realizar os pedidos pelo aplicativo. · Gerenciar pedidos pelo painel.
Sprint 4	<ul style="list-style-type: none"> · Desenvolver o serviço de cupons do sistema. · Desenvolver a tela de informações gerenciais no painel.
Sprint 5	<ul style="list-style-type: none"> · Escrever o artigo final. · Estruturar apresentação do artigo.

Fonte: Dados do trabalho, 2020.

Para a implementação do sistema foram utilizadas as subsequentes ferramentas:

- *Visual Studio Code*: trata-se de um editor de código leve, multiplataforma, gratuito e *open source*. Mantido pela Microsoft, essa ferramenta conta com suporte a várias linguagens, extensões, integração com *Git*, *debug*, terminal integrado, entre outros recursos.
- *NestJS*: é um framework *Node.js* progressivo para criar aplicativos eficientes, confiáveis e escalonáveis do lado do servidor.
- *ReactJS*: Biblioteca JavaScript para criar interfaces de usuário.
- *React Native*: é uma biblioteca JavaScript criada pelo Facebook. É utilizada para desenvolver aplicativos para os sistemas Android e IOS de forma nativa.
- *PostgreSQL*: sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (Structured Query Language) para a persistência de dados.
- *Git*: sistema de controle de versão que gerencia arquivos e diretórios e as modificações feitas neles ao longo do tempo.

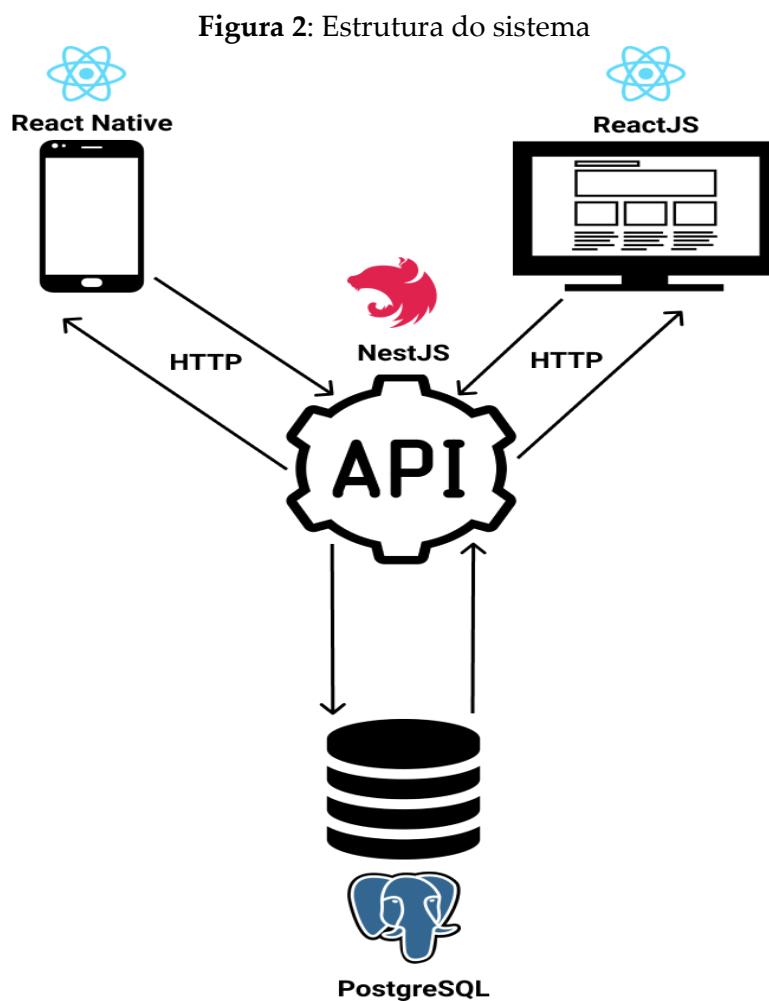
O desenvolvimento do sistema foi dividido em três aplicações, sendo elas: *backend*, *frontend* e *mobile*. Durante o desenvolvimento do mesmo foi utilizado o fluxo de trabalho do Gitflow. O Gitflow é um design de fluxo de trabalho Git, que define um modelo de ramificação projetado com base no lançamento do projeto, oferecendo uma estrutura robusta para gerenciar projetos.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Na fase inicial do projeto foi definido a linguagem de programação, banco de dados, framework e bibliotecas. A linguagem utilizada no desenvolvimento foi o

JavaScript, o framework escolhido e as bibliotecas utilizadas para o desenvolvimento do *backend* da aplicação, painel e aplicativo foram respectivamente: NestJS, ReactJS e React Native. Já para o armazenamento dos dados foi utilizado o banco de dados PostgreSQL.

Na Figura 2 é apresentada a estrutura do sistema que consiste em um padrão *Model-View-Controller* - MVC, mostrando as tecnologias que foram citadas anteriormente e ilustrando como funciona o fluxo do sistema.



Fonte: Dados do trabalho, 2020.

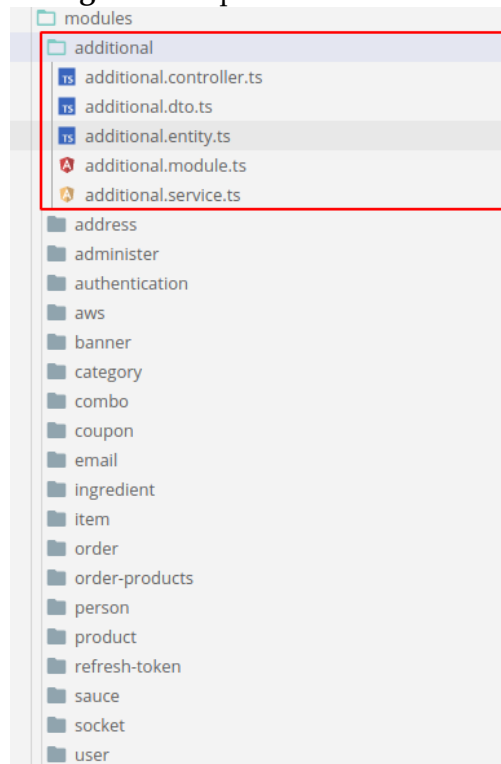
Falando um pouco mais sobre o backend, o mesmo foi desenvolvido em NestJS como já foi dito. O NestJS é um framework Node.js progressivo para criar aplicativos eficientes, confiáveis e escalonáveis do lado do servidor. Permite trabalhar com *Object-relational mapping* - ORM, utiliza do *TypeScript* o que torna o desenvolvimento mais seguro e menos propenso a erros. O NestJS tem sua arquitetura inspirada no *Angular*, possuindo um baixo acoplamento e uma fácil manutenção.

A arquitetura da aplicação *backend* está dividida de forma que cada funcionalidade está separada, deixando a mesma coesa, menos acoplada e de fácil manutenção. Cada funcionalidade da aplicação possui seu módulo específico, nesse módulo encontra-se a *entity*, *controller*, *dto*, *service* e *module* da funcionalidade

Entity é o modelo da entidade que é utilizado para gerar as tabelas no banco de dados, a *controller* é responsável por fazer a comunicação entre as requisições e as regras de negócio da aplicação, ou seja, ela é uma fronteira. O *dto* é uma classe responsável por validar os atributos vindos na requisição antes de chegar na *controller*. A *service* possui toda a regra de negócio, validações e é responsável por persistir os dados no banco de dados caso necessário. E a *module* é onde é configurado e exportado o módulo para todo o projeto caso uma outra funcionalidade precise do mesmo.

Na Figura 3 pode-se observar o que foi explicado anteriormente, utilizando como exemplo o módulo de adicionais.

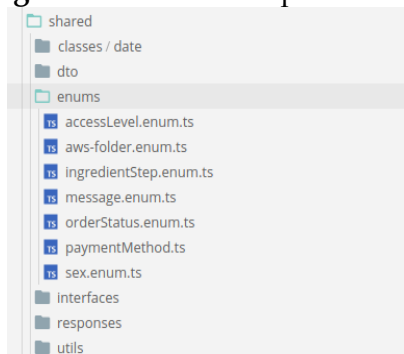
Figura 3: Arquitetura do *backend*



Fonte: Dados do trabalho, 2020.

A aplicação *backend* também conta com uma pasta *shared*, que contém todas as classes, interfaces e funções que são comuns para mais de um módulo da aplicação, centralizando tudo o que é compartilhado em apenas um lugar, permitindo uma melhor reusabilidade e manutenção. Na figura 4 é possível observar melhor a utilização da pasta *shared*.

Figura 4: Estrutura da pasta *shared*



Fonte: Dados do trabalho, 2020.

O painel do sistema foi desenvolvido utilizando o ReactJS, uma biblioteca JavaScript de código aberto usada para construir interfaces de usuário eficientes e que permite a reusabilidade de componentes que tenham sido desenvolvidos em outras aplicações. Esses componentes são fáceis de escrever por usarem JSX, uma extensão de sintaxe opcional para JavaScript, a qual permite que se combine Hypertext Markup Language - HTML, com o JavaScript. Na Figura 4 pode-se observar um exemplo de uma sintaxe JSX.

Figura 5: Sintaxe JSX

```

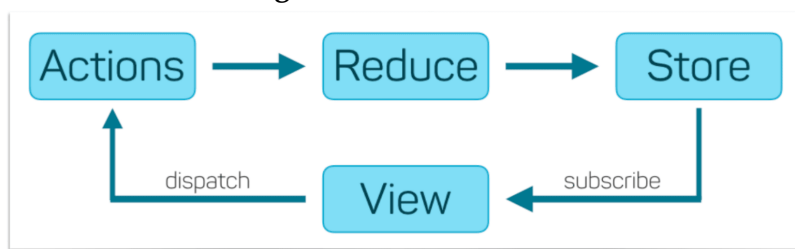
1
2  const element = <h1>Hello, world!</h1>;
3
    
```

Fonte: Dados do trabalho, 2020.

O conceito da arquitetura utilizada no painel é a mesma do *backend*, com apenas algumas diferenças para adaptar-se à aplicação que agora é *web*, mas ainda procurando manter a coesão, a fácil manutenção e uma boa reusabilidade. Para facilitar o compartilhamento de estados entre vários componentes, foi utilizado o Redux tanto no painel quanto no aplicativo.

Teoricamente o Redux é um controlador de estados geral para uma aplicação. Compartilhar estados entre os diferentes componentes torna-se muito mais fácil quando se utiliza o Redux, o qual é basicamente dividido em 3 partes: *store*, *reducers* e *actions*, como pode ser observado na Figura 6.

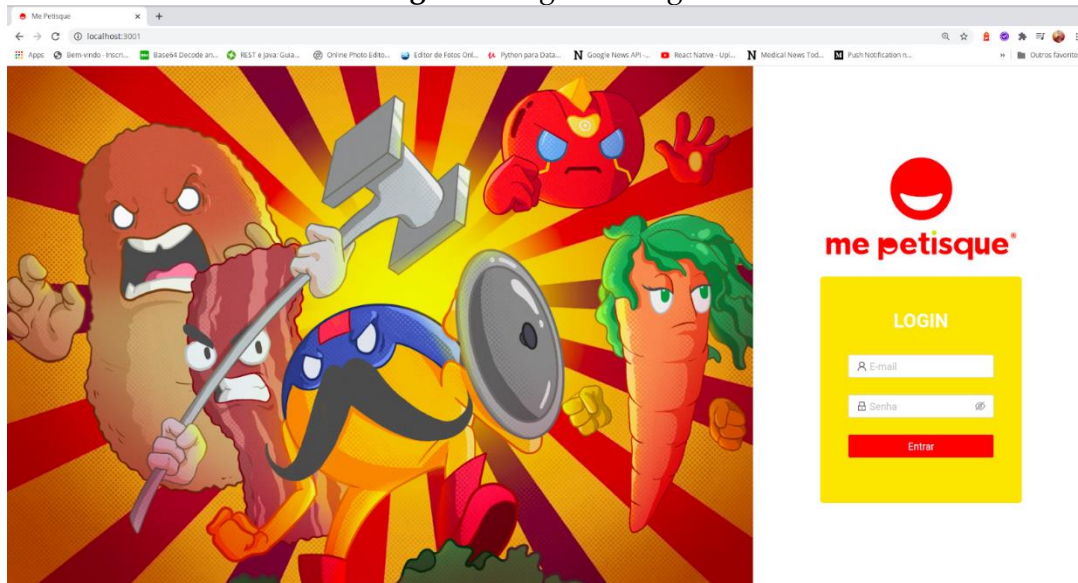
Figura 6: Fluxo do Redux



Fonte: WECK, 2017.

A seguir é apresentado, na Figura 7, a página de login do painel um dos resultados obtidos durante o desenvolvimento da aplicação *web*.

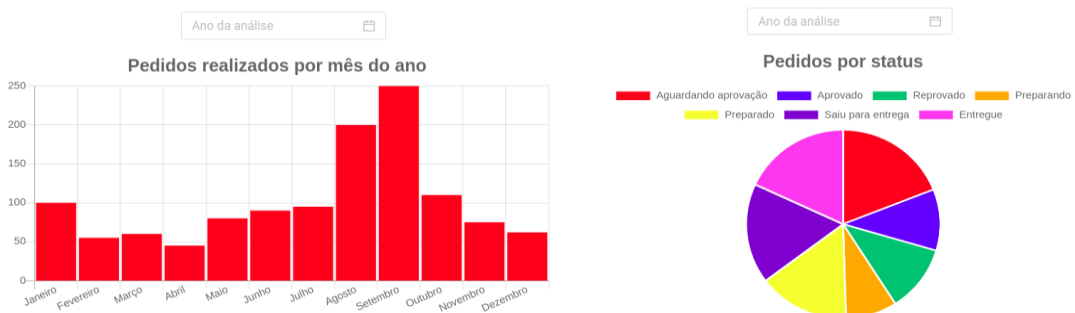
Figura 7: Página de login



Fonte: Dados do trabalho, 2020.

Outro resultado obtido durante o desenvolvimento da aplicação *web*, foi a parte de análise gráfica do sistema, como pode-se observar na Figura 8, tem-se dois gráficos, no primeiro é mostrado uma relação da quantidade de pedidos realizados por cada mês do ano e o segundo traz informações da quantidade de pedidos que se encontra em cada status proposto, tanto o primeiro quanto o segundo gráfico pode ter suas informações filtradas pelo ano, possibilitando a análise de dados também de anos anteriores ou posteriores.

Figura 8: Página de análise



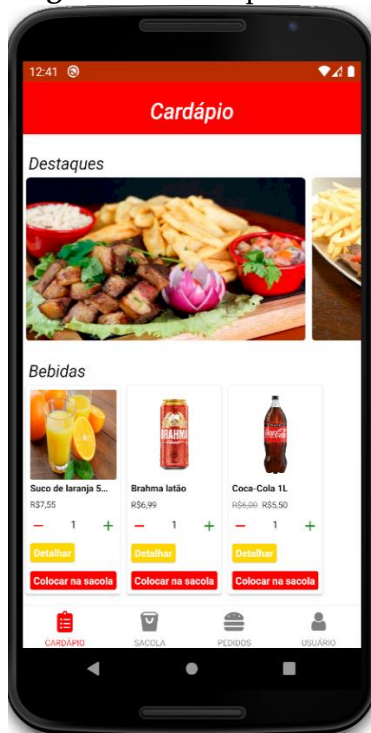
Fonte: Dados do trabalho, 2020.

Para finalizar a parte do aplicativo, este foi desenvolvido utilizando a biblioteca React Native, sendo essa uma biblioteca JavaScript criada pelo Facebook e utilizada para desenvolver aplicativos para os sistemas Android e IOS de forma nativa. Com o React Native é possível gerar aplicativos tanto para a plataforma Android

quanto IOS utilizando apenas uma linguagem de programação, agilizando muito o desenvolvimento *mobile*.

Praticamente não existe diferença entre as bibliotecas ReactJS e React Native, o que agiliza mais ainda o desenvolvimento de um projeto que possui uma versão *web* e *mobile*. Com isso, a arquitetura, estrutura e forma de programação tanto do painel quanto do aplicativo não possuem diferenças. Na Figura 9 é apresentada a tela de produtos do aplicativo, nela é possível observar como os produtos estão dispostos na tela. Já na figura 10 é apresentado a tela de detalhes do produto do aplicativo, nela é possível visualizar de forma mais detalhada as informações do produto.

Figura 9: Tela de produtos



Fonte: Dados do trabalho, 2020.

Figura 10: Detalhes do produto



Fonte: Dados do trabalho, 2020.

De acordo com a proposta do projeto e os objetivos propostos, foi apresentado nesta seção alguns dos resultados obtidos durante o desenvolvimento do sistema, mostrando algumas páginas, telas, padrões de projeto e arquiteturas.

5 CONSIDERAÇÕES FINAIS

O painel desenvolvido tem como proposta entregar para o Me Petisque uma aplicação *web* por meio da qual pode-se gerenciar os produtos que serão vendidos no aplicativo, visualizar os clientes cadastrados pelo aplicativo, cadastrar cupons de descontos para serem disponibilizados para os clientes, gerenciar os pedidos do dia e também visualizar algumas análises. Com isso a empresa tem em mãos um painel para poder administrar seu negócio.

Pelo aplicativo desenvolvido o cliente do Me Petisque pode se cadastrar, efetuar login, visualizar os produtos disponíveis para venda, realizar seus pedidos e se necessário fazer a atualização dos seus dados cadastrados no aplicativo. Entregando para o cliente uma aplicação *mobile* que permite a utilização dos serviços prestados pela empresa.

Mesmo com todos os objetivos propostos para este trabalho terem sido alcançados, salienta-se que o sistema desenvolvido não passou por uma etapa de testes, para garantir que todas as suas funcionalidades estão funcionando como o planejado e o mesmo também não foi colocado em produção. Além disso, para projetos futuros podem ser desenvolvidas novas funcionalidades, como *push notifications* para a empresa poder enviar notificações de promoções ou algum cupom de desconto para seus clientes, ou ainda alguma forma de autenticação diferente como, pela conta Google ou Facebook.

Com a mudança no estilo de vida da população, a mesma está cada vez mais optando por fazer suas refeições fora do domicílio ou pedir sua comida no conforto de sua casa, com isso o mercado de *delivery* vem ganhando força a cada ano. Portanto, é imprescindível que as empresas desse ramo apostem em tecnologia para conseguirem se manter na disputa do mercado de *delivery*. Neste sentido, o aplicativo desenvolvido para a empresa Me Petisque pode ser adaptado e atender a diversas outras empresas do ramo de alimentação que trabalha com *delivery*.

REFERÊNCIAS

BISSI, Wilson. Metodologia de desenvolvimento ágil. **Campo Digital**, v. 2, n. 1, 2007. Disponível em: <https://bit.ly/2SMTjnm>. Acesso em: 5 maio. 2020.

DOS SANTOS CLARO, José Alberto Carvalho. **Sistemas de Informações Gerenciais**, São Paulo. 2013. Disponível em: <https://bit.ly/2yd0NIE>. Acesso em: abr. 2020.

FRANÇA, Ivanir. **Investimentos: tendências para o mercado de delivery no Brasil e no mundo em 2019**. 2019. Disponível em: <https://bit.ly/33TSopd>. Acesso em: 8 fev. 2020.

LEAL, Daniele. Crescimento da alimentação fora do domicílio. **Segurança Alimentar e Nutricional**, v. 17, n. 1, p. 123-132, 2010. Disponível em: <https://bit.ly/2Ueqm4U>. Acesso em: 18 março 2020.

MADUREIRA, Daniele. **Como apps de entrega estão levando pequenos restaurantes à falência**, 2020. Disponível em: <https://bit.ly/2UotevY>. Acesso em: 18 março 2020.

MORAES, Daniel. **A arte de fidelização do cliente: entenda o que você precisa para ter um consumidor que propague sua marca**, 2018. Disponível em: <https://bit.ly/3ao0JUG>. Acesso em: 16 março 2020.

OLIVEIRA, Djalma. **Sistemas de Informações Gerenciais**. 15. ed. São Paulo: Atlas, 2012.

OLD, Dino. **Startup brasileira que faz apps de delivery online vai para o Vale do Silício pelo Google**, 2017. Disponível em: <https://bit.ly/2UotevY>. Acesso em: 23 fev. 2020.

PREZOTTO, Ezequiel Douglas; BONIATI, Bruno Batista. **Estudo de frameworks multiplataforma para desenvolvimento de aplicações mobile híbridas**. Trabalho de Conclusão de Curso. Universidade Federal de Santa Maria. 2014. Disponível em: <https://bit.ly/2Jq0g8y>. Acesso em: mar. 2020.

REZENDE, Eliana. **Dados, informação e conhecimento: o que são?**. 2015. Disponível em: <https://bit.ly/2WZ5U9A>. Acesso em: 27 março 2020.

SILVA, Mateus. **Quais são os principais frameworks para desenvolver aplicações móveis híbridas?**. 2018. Disponível em: <https://bit.ly/2yiyAjP>. Acesso em: 27 março 2020.

SCHWABER, Ken; SUTHERLAND, Jeff. **Um guia definitivo para o Scrum: as regras do jogo**. Processo de Desenvolvimento de Software, 2013. Disponível em: <https://bit.ly/2yurxoo>. Acesso em: 21 fev. 2020.

WECK, Sandy. **Developing modern offline apps with ReactJS, Redux and Electron - Part 3 ReactJS + Redux**. 2017. Disponível em: <https://bit.ly/346KIY1>. Acesso em: 18 out. 2020.

DESENVOLVIMENTO DE PULSEIRA INTELIGENTE PARA DETECÇÃO E PREVENÇÃO DE ATAQUES EPILÉPTICOS

Higor Férrer¹
Henaldo Barros²

RESUMO: Esse artigo aborda o estudo e a aplicação da Tecnologia da Informação aplicada à saúde, visando auxiliar e melhorar a vida de pacientes com casos de epilepsia e seus familiares, focando no desenvolvimento de um protótipo de pulseira e um aplicativo para a comunicação ser realizada. Assim, será apresentado o processo de criação de uma pulseira inteligente que monitora a temperatura, umidade da pele e caso tenha uma aceleração brusca será possível enviar um sinal da pulseira para os celulares cadastrados de familiares do paciente. Para o desenvolvimento do protótipo foi utilizada a linguagem C++ e o aplicativo para comunicação entre a pulseira e o celular, desenvolvido em Java.

PALAVRAS-CHAVE: arduino; wearable; epilepsia.

ABSTRACT: This article will address the study and application of Information Technology applied to health, aiming at helping and improving the lives of patients with epilepsy cases and their families, through the development of a bracelet prototype and an application to communicate with it. Thus, it will be presented the process of creating a smart bracelet that will monitor the temperature, skin humidity, and if a sudden acceleration happens it will be possible to send a signal from the bracelet to the registered cell phones of the patient's family members. For the development of the prototype, the C ++ language was used, and the application for communication between the bracelet and the cell phone, developed in java.

KEYWORDS: arduino; wearable; epilepsy.

1 INTRODUÇÃO

A epilepsia é um transtorno neurológico sério e comum no mundo e atinge aproximadamente 50 milhões de pessoas, grande parte delas em países subdesenvolvidos (MARCHETTI *et. al.*, 2005).

Em geral, pode-se afirmar que a epilepsia é uma das condições que mais afeta o comportamento e a qualidade de vida, não só da pessoa que sofre dessa condição, mas também de toda sua família e grupo social, especialmente pelo estigma existente (FERNANDES, 2006).

As crises podem se manifestar por vários sinais e sintomas, como alterações sensitivas (visuais, auditivas, gustativas.), autonômicas, motoras, cognitivas e do nível de consciência (VALENÇA *et. al.*, 2000). De acordo com o Ministério da Saúde (2013) existe um risco para pessoas com ataques epiléticos, pois, quando não tratadas imediatamente, podem levar à morte, já que durante uma crise, é possível ter asfixia ou traumas devido à queda. Além disso, pode-se ocorrer a morte classificada como SUDEP (Morte Súbita Por Epilepsia).

¹ Aluno de Sistemas de Informação, UNIPAM. E-mail: higorferrer@gmail.com.

² Mestre em Redes de Computadores pela UFU, docente do UNIPAM. E-mail: henaldo@unipam.edu.br.

De forma geral, a SUDEP é responsável por aproximadamente 17% de todas as mortes em epilepsia e apresenta uma taxa anual de incidência entre adultos de 1:1.000 pacientes, enquanto em crianças varia em média de 0,2/1.000 por ano (SCORZA *et. al.*, 2017).

Nesse cenário, é necessário pontuar que, de acordo com SCORZA *et. al.* (2017) não há método eficaz e definitivo para evitar os ataques epiléticos ou a SUDEP causada por eles, mesmo que noites de sono melhores, hábitos de vida mais saudáveis e exercícios possam ajudar. Portanto, o acompanhamento contínuo dos pacientes é fundamental.

De acordo com Amarilian (2009), há uma necessidade contínua de acessibilidade e inclusão por parte das sociedades para com os indivíduos que tenham suas particularidades, diferenças e limitações. Para isso, em tempos atuais, podemos contar com a ajuda da tecnologia para auxiliar pessoas no crescimento na demanda por acessibilidade.

Com métodos eficientes de prevenção, auxílio dos pacientes em momentos de ataque e monitoramento seguido de análise, existe uma maior chance de auxiliar os enfermos, e trazer mais segurança e conforto a todas as partes envolvidas.

Pensando nisso, neste artigo são realizados:

- O estudo sobre a Tecnologia da informação aplicada à saúde;
- O estudo sobre arduino, dispositivo *wearable*, compreendendo seus conceitos básicos e identificar dispositivos que foram desenvolvidos com essas ferramentas;
- Um dispositivo capaz de monitorar um paciente e o acontecimento de um ataque epilético através de detecção de movimento brusco ou fora do padrão, aumento de temperatura do pulso fora do padrão ou queda;
- Um aplicativo capaz de enviar sinais de alerta para um responsável, assim sendo capaz de socorrer a pessoa em momento de ataque epilético.

Atualmente, um grande número de pessoas e seus familiares sofrem problemas e desconfortos causados por essa doença, e boa parte da população não sabe como proceder em situações de ataque epilético, o que gera uma preocupação maior ainda, portanto com o estudo da epilepsia e a Tecnologia da Informação aplicada à saúde, foi possível desenvolver um protótipo capaz de oferecer uma melhoria na qualidade de vida e conforto para pacientes e seus familiares.

2 REFERENCIAL TEÓRICO

Nesta seção são apresentados conceitos e estudos realizados sobre a Tecnologia da Informação e também sua aplicação à área da saúde.

2.1 TECNOLOGIA DA INFORMAÇÃO APLICADA À SAÚDE

Entre o século XVIII e o século XIX, houve a primeira revolução técnico-científica, onde houve mudanças na produção, como a substituição do homem pelas máquinas, primeiramente pelas máquinas à vapor e em seguida pelas máquinas elétricas. Então a tecnologia passa a ser entendida como o estudo do emprego de

teorias, processos e métodos científicos para solução de problemas técnicos (NIETSCHE, 1999).

Essas mudanças da tecnologia também ocorreram na área da saúde, com a introdução da informática e de aparelhos modernos, que trouxeram muitos benefícios e facilidades na batalha contra as doenças nos últimos anos. O aumento em grande escala do desenvolvimento de técnicas e aparelhos que facilitam e melhoram as condições de atendimento ao cliente e também de funcionários na área da saúde continuam a ser implementadas a cada momento.

2.2 ARDUINOS

Originalmente, o arduino foi criado como uma ferramenta para auxiliar os estudantes, mas no ano de 2005 ele foi comercialmente lançado por Massimo Banzi e David Cuartielles, e devido a fácil utilização e durabilidade que ele proporciona, tornou-se um produto de sucesso entre fabricantes e estudantes (MONK, 2013).

O arduino é um dispositivo formado principalmente por dois componentes básicos, sendo: a placa arduino, que é o hardware utilizado para construir seus objetos, e a IDE (Integrated Development Environment) do arduino, que é um software que é executado pelo computador onde desenvolvemos o código e enviará para a placa (BANZI, 2010).

Em resumo, é uma placa simples com entradas e saídas, que propõe uma tecnologia de baixo custo e bom desempenho, podendo ser usada na construção de projetos baseados em microcontrolador (MONK, 2013).

2.3 PULSEIRAS INTELIGENTES

Pulseiras inteligentes, um tipo de dispositivo *wearable*, são aparelhos que possuem diversas funções, incluindo facilitar a rotina com o envio de mensagens ou funções como alarme e despertador. Recentemente, vem surgindo inovações na confecção destes dispositivos, capacitando-os para funções mais complexas e úteis, como monitoramento de funções vitais e motoras (PELLANDA, 2016).

Um dispositivo *wearable* é por norma um dispositivo pequeno, portátil e interativo que está sempre pronto para ser utilizado. Uma das principais características que diferencia um dispositivo *wearable* de um simples computador, ou equipamento informático portátil, é a possibilidade de este dispositivo permanecer anexado ao corpo do utilizador durante um longo período de tempo sem que se torne incômodo ou condicionador de movimentos (GODINHO, 2013).

Para o desenvolvimento de uma pulseira para monitorar a ocorrência de ataques epiléticos, seria necessário utilizar uma placa de arduino com três sensores, sendo eles: um de movimento, para detectar movimentos bruscos ou fora do padrão, um de calor para detectar aumento de temperatura do pulso fora do padrão, indicando febre; e um de umidade, para identificar suor no pulso. Também é necessário que a pulseira se conecte via *bluetooth* a um aplicativo no celular do portador e imediatamente envie uma mensagem de alerta para contatos pré-definidos.

2.4 INTERNET DAS COISAS

A Internet das Coisas (IoT) é uma extensão da Internet atual, que possibilita aos objetos de sua rotina com capacidade de comunicação se conectarem à internet. Surgiram do progresso de várias áreas, como sistemas embarcados, microeletrônica, comunicação e sensoriamento.

De acordo com (YAN *et. al.*, 2008); (GIUSTO *et. al.*, 2010) e (ATZORI *et. al.*, 2010) definir especificamente o que é a Internet das Coisas parece ainda mais complexo quando é colocado a inúmera quantidade de assuntos que a temática pode envolver. Na publicação existem artigos sobre coleta de dados, segurança de rede, *middleware*, e o mais importante, sensores de baixo consumo de energia.

Com certeza, a Internet das Coisas vem recebendo bastante atenção, quanto à área da indústria, mas como também da área acadêmica, devido ao seu alto potencial de uso nas mais diversas áreas da atividade humana, mas a falta de compreensão sobre o assunto é tema de publicações e aparecem nos trabalhos de Atzori *et. al.* (2011) e Gigli e Koo (2011), que já apontavam que não seria apenas mais um número de IP para cada objeto, já que os objetos podem se comunicar por radiofrequência, assim sendo mais do que uma tecnologia de comunicação, uma vez que sistemas usam de várias tecnologias.

3 METODOLOGIA

O atual artigo é elaborado com base no estudo da Tecnologia da Informação aplicada à saúde. É realizada, portanto, uma revisão da literatura, com destaque na dificuldade encontrada em pessoas com problemas epiléticos, sendo consultados artigos, livros e trabalhos acadêmicos que são elaborados a partir dessas dificuldades.

Após o estudo é desenvolvido um protótipo de uma pulseira para auxiliar pessoas com problemas de ataques epiléticos e seus familiares, onde é utilizada uma placa básica de arduino conectados a sensores, sendo programado em seu próprio ambiente de desenvolvimento na linguagem C++. Depois de realizado o protótipo da pulseira, um aplicativo *mobile* é desenvolvido para o gerenciamento e controle dos dados gerados pela pulseira, onde a pulseira envia o sinal para o aplicativo via *bluetooth*, e caso os dados indiquem um possível ataque, um sinal via Wi-Fi ou 4G é enviado para os celulares cadastrados de familiares dos responsáveis pela pulseira.

4 DESENVOLVIMENTO E RESULTADOS

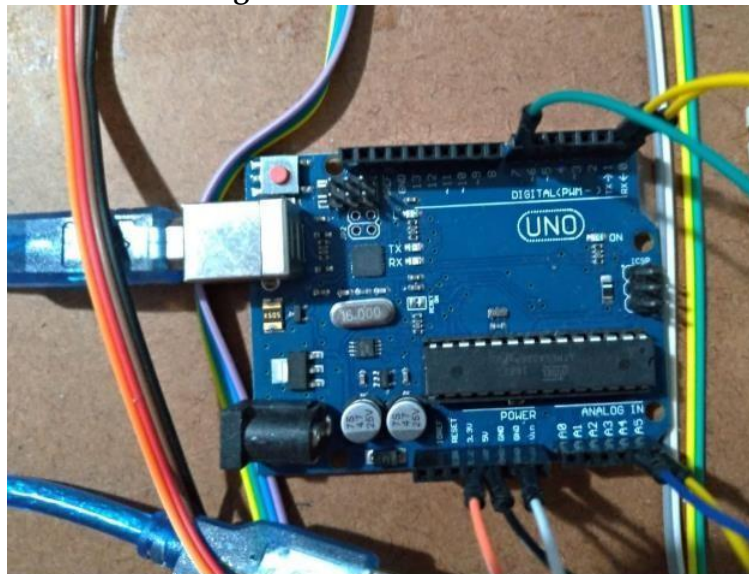
De acordo com MCROBERTS (2010), Arduino é um sistema embarcado, ou seja, que é livre para interagir com seu ambiente via hardware e software agregado a um dispositivo com um objetivo pré-definido. Essa plataforma tem o intuito de facilitar, a nível doméstico e comercial, a automação e controle nestes ambientes.

Na fase inicial do projeto foram realizadas pesquisas sobre o uso do Arduino e da Tecnologia aplicada à saúde.

Para MONK (2010), a placa existe somente para oferecer suporte estrutural para os microcontroladores (sensores), permitindo que seus sensores se comuniquem uns com os outros, ou seja, tudo que será realizado pelo Arduino, deverá ser

programado por meio da IDE (*Integrated Development Environment*), onde são estabelecidas orientações para cada execução. A placa escolhida para o projetado, é a placa Arduino Uno R3, apresentada na Figura 1.

Figura 1: Placa Arduino R3



Fonte: Arquivo pessoal.

Três microcontroladores são utilizados para o desenvolvimento da pulseira, o DHT11, MPU-6050 e o HC-05, todos microcontroladores conectados à placa Arduino Uno.

Na Figura 2, é apresentado o MPU-6050, microcontrolador com dois chips sensoriais, sendo que, em um único chip possui acelerômetro e giroscópio, responsáveis pelo monitoramento da aceleração do paciente, e o outro chip responsável pelo monitoramento da temperatura, permitindo medições entre -40° e 85°C .

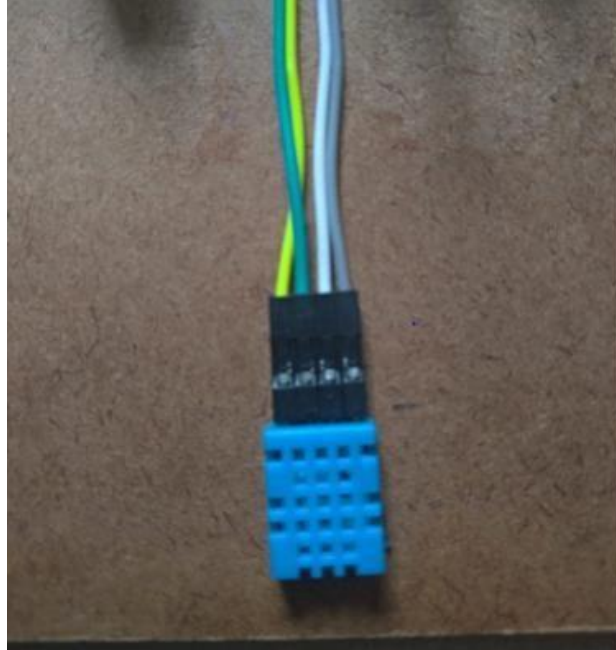
Figura 2: Microcontrolador MPU-6050



Fonte: Arquivo pessoal.

Na Figura 3, é mostrado o DHT11, microcontrolador com chip sensorial capaz de fazer leituras de umidade entre 20 a 90%, responsável pelo controle da umidade da pele do paciente.

Figura 3: Microcontrolador DHT11



Fonte: Arquivo pessoal.

Na Figura 4, é apresentado o HC-05, microcontrolador que possibilita transmitir e receber dados através da comunicação sem fio (*bluetooth*), responsável por fazer a conexão entre o celular do paciente e a pulseira.

Figura 4: Microcontrolador HC-05



Fonte: Arquivo pessoal.

No desenvolvimento e programação do código foi utilizado a linguagem C++, programada na própria IDE do Arduino, onde é possível projetar, alterar e compilar ações para o Arduino ou seus sensores executarem.

Na Figura 5, temos o trecho do código para comunicação com o microcontrolador MPU-6050, onde a todo momento estará monitorando a temperatura do paciente, caso a temperatura monitorada seja maior que 36°C (temperatura que indica que o paciente está febril), uma mensagem será enviada ao dispositivo conectado, já em relação a aceleração, caso ultrapasse 8 m/s, uma mensagem também será enviada.

Figura 5: Código do microcontrolador MPU-6050

```
void loop()
{
  Wire.beginTransmission(MPU);
  Wire.write(0x3B);
  Wire.endTransmission(false);

  Wire.requestFrom(MPU, 14, true);

  //Armazena o valor dos sensores nas variaveis correspondentes
  acelX=Wire.read()<<8|Wire.read();
  acelY=Wire.read()<<8|Wire.read();
  acelZ=Wire.read()<<8|Wire.read();
  temperatura=Wire.read()<<8|Wire.read();
  giroX=Wire.read()<<8|Wire.read();
  giroY=Wire.read()<<8|Wire.read();
  giroZ=Wire.read()<<8|Wire.read();

  Serial.print("Acel:");
  Serial.print(" X:");Serial.print(acelX);
  Serial.print("\tY:");Serial.print(acelY);
  Serial.print("\tZ:");Serial.print(acelZ);

  //valores do giroscpio
  Serial.print("\tGiro:");
  Serial.print(" X:");Serial.print(giroX);
  Serial.print("\tY:");Serial.print(giroY);
  Serial.print("\tZ:");Serial.print(giroZ);

  Serial.print("\tTemperatura: ");   Serial.println(temperatura/340.00+36.53);

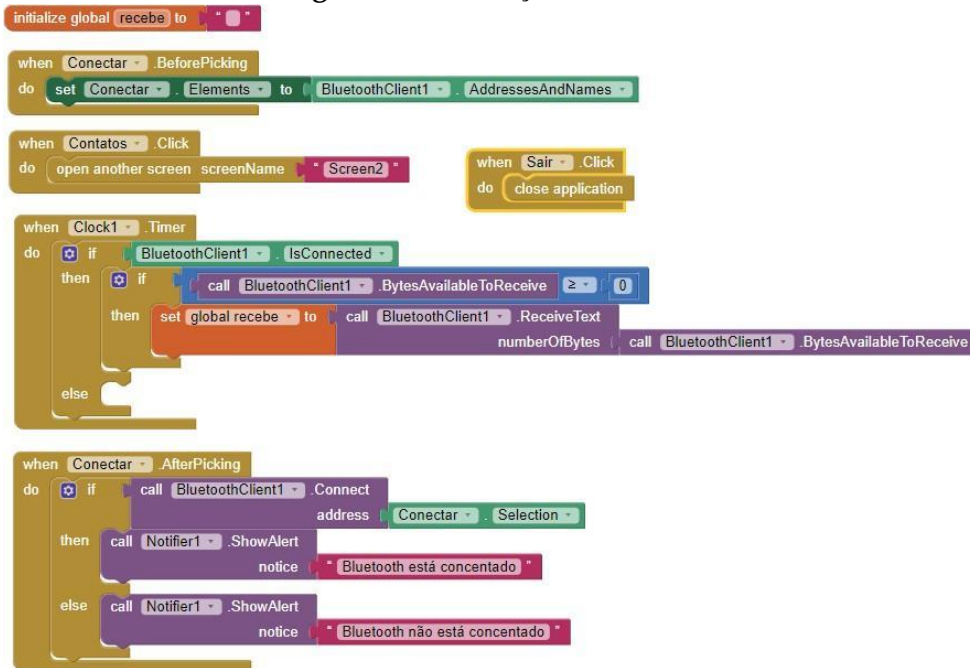
  delay(500);
}
```

Fonte: Dados do trabalho.

O aplicativo foi desenvolvido através do site e aplicativo MIT *app inventor*, através da codificação em blocos, com a codificação de blocos da tela de blocos representada na Figura 6, e sua tela no aplicativo na Figura 7.

DESENVOLVIMENTO DE PULSEIRA INTELIGENTE PARA DETECÇÃO E PREVENÇÃO DE ATAQUES EPILÉPTICOS

Figura 6: Codificação em blocos



Fonte: Dados do trabalho.

Figura 7: Tela inicial do aplicativo



Fonte: Dados do trabalho.

Na segunda tela, representada pela Figura 8, temos a parte de cadastro e controle de contatos de emergência, onde é possível adicionar, remover ou consultar

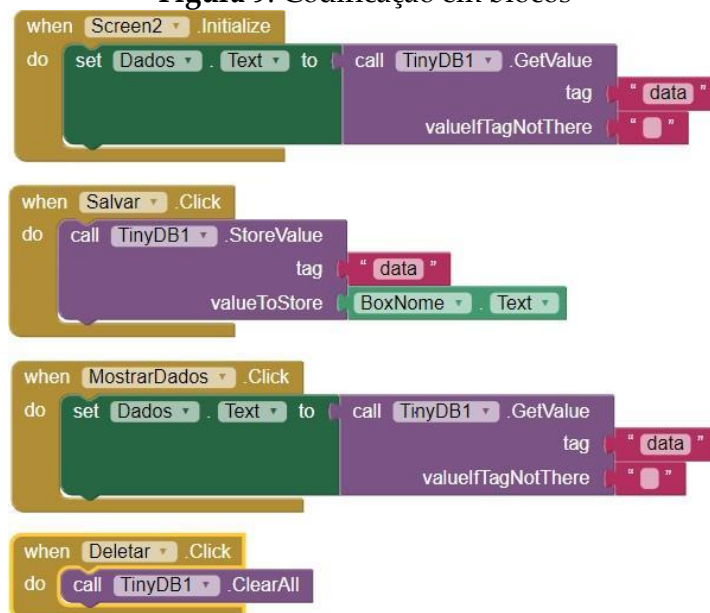
contatos salvos. Na Figura 9, temos um trecho do desenvolvimento da tela de contatos, onde no primeiro bloco temos a inicialização do banco, depois o trecho responsável para armazenar os dados no banco, seguido do trecho responsável por deletar um contato e por último o responsável por exibir a lista de contatos salvos.

Figura 8: Tela de contatos



Fonte: Dados do trabalho

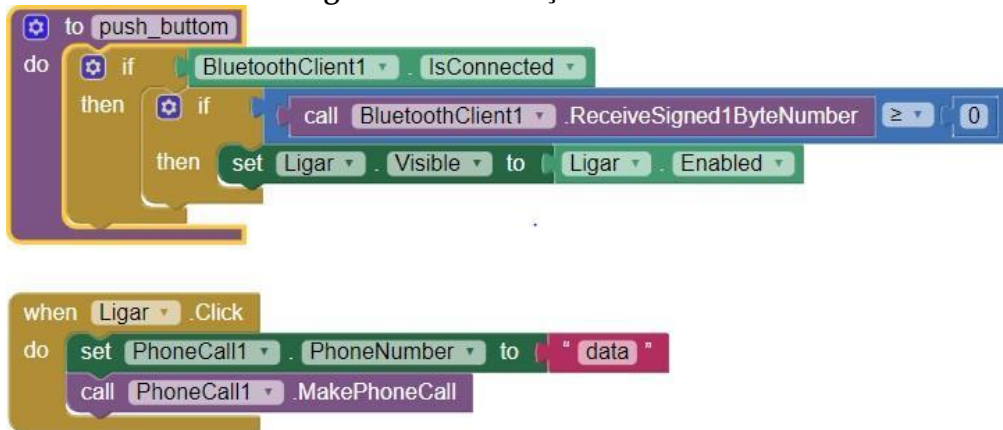
Figura 9: Codificação em blocos



Fonte: Dados do trabalho.

Na última etapa, representada pela Figura 10, temos a parte responsável por ligar para o contato salvo. Uma procedure foi criada para que quando o aplicativo receber uma mensagem via *bluetooth* (mensagens são enviadas apenas caso o sensor aponte um dos sistemas do ataque) o botão de ligação estará disponível, e quando esse botão passa para o estado “enabled”, a ligação é feita automaticamente.

Figura 10: Codificação em blocos



Fonte: Dados do trabalho

5 CONCLUSÃO

O atual projeto desenvolvido sugere possível através do estudo parcial sobre crises epiléticas, a possível detecção num curto espaço de tempo pelo dispositivo desenvolvido, com preço acessível, que tem capacidade de monitorar possíveis reações normais do corpo humano e que são indícios de uma crise epilética.

O protótipo e estudo foi desenvolvido para o controle de pacientes com crise epiléticas, mas pode também ser utilizado para o auxílio de cuidado de pessoas idosas, caso ela tenha uma aceleração brusca, assim com uma margem segura de tempo, é possível alertar familiares ou responsáveis que convivem com o paciente, que previamente já conhecem os procedimentos para o auxílio da crise.

A IoT existe há muitos anos e encontra-se em ascendência desde seu surgimento, sendo claro o quanto o uso de *wearables* estão acessíveis e em tendência para o uso das pessoas. Diante de tal possibilidade os indivíduos e familiares podem ter uma melhor qualidade de vida, possibilitando não controlar, mas que se preparem para a chegada da crise.

REFERÊNCIAS

AMARILIAN, M. L. T. Comunicação e participação ativa: a inclusão de pessoas com deficiência visual. In: AMARILIAN, M. L. T. (org.). **Deficiência visual: perspectivas na contemporaneidade**. São Paulo: Vetor, 2009.

- ATZORI, Luigi; IERA, Antonio; MORABITO, Giacomo. The Internet of Things: a survey. **Computer Networks**, 2010.
- BANZI, M. **Primeiros passos com o arduino**. São Paulo: Novatec Editora Ltda., 2010.
- GIGLI, M.; KOO, S. Advances in Internet of Things. *In: Scientific Research*, v. 1, p. 27-31, 2011. Disponível em: <https://www.scirp.org/journal/ait/> Acesso em: 30 jun. 2020.
- GODINHO, P. M. A. S. **Pulseira inteligente para monitorização de funções vitais**. Porto, Portugal, 2013. Disponível em: https://recipp.ipp.pt/bitstream/10400.22/6324/1/DM_PedroGodinho_2013_MEEC.pdf. Acesso em: 24 maio 2020.
- MARCHETTI, Renato Luiz *et al.* Transtornos mentais associados à epilepsia. **Archives of Clinical Psychiatry**, São Paulo, v. 32, n. 3, p. 170-182, 2005.
- MARQUES, I. R.; SOUZA, R. Tecnologia e humanização em ambientes intensivos. **Revista brasileira de Enfermagem**, v. 63, n. 1, p. 141-144, 2010.
- MINISTÉRIO da Saúde. **Protocolo Clínico e Diretrizes Terapêuticas – Epilepsia**. 2003. Disponível em: <http://portalarquivos.saude.gov.br> Acesso em: 15 março 2020.
- MONK, S. **Programação com Arduino >>começando com sketches**. Porto Alegre: Bookman Editora Ltda., 2013.
- NIETSCHE, E. A; DIAS L. P. M; LEOPARDI, M. T. Tecnologias em enfermagem: um saber prático? *In: Anais do Seminário Nacional de Pesquisa em Enfermagem*. 9, 1999, maio, Gramado (RS), Brasil. ABEn-RS; 1999.
- PELLANDA, E. C. A prevenção primordial e a "Saúde de Vestir": os wearables na cardiologia. **Arq. Bras. Cardiol.**, São Paulo, v.106, n. 6., 2016.
- SCORZA, F. A. *et al.* SUDEP: a steep increase in publication since its definition. **Epilepsy Behav**, v. 72, p.195-197, 2017.
- VALENÇA, Marcelo Moraes; VALENÇA, Luciana Patrícia A. Andrade. Etiologia das crises epiléticas na cidade do Recife, Brasil: estudo de 249 pacientes. **Arquivos de Neuro-Psiquiatria**, v. 58, n. 4, p. 1064-1072, 2000.
- YAN, Lu; ZHANG, Yan; YANG, Laurence T.; NING, Huansheng. **The Internet of Things: from RFID to the Next-Generation Pervasive Networked Systems**. Auerbach Publications: Nova Iorque e Londres, 2008.
- MCROBERTS, Michael. **Beggining Arduino**. Apress. Nova Iorque: 2010.

FRILA: APLICATIVO PARA PRESTADORES DE SERVIÇO¹

Adryel Klever Vieira de Almeida²
Mislene Dalila da Silva³

RESUMO: Esse estudo teve como objetivo o desenvolvimento de uma ferramenta de divulgação de serviços para auxiliar os trabalhadores autônomos, como eletricitas, pedreiros etc. Na aplicação da ferramenta, ela se demonstrou eficiente por simplificar os processos de divulgação dos serviços e de comunicação entre trabalhadores e clientes. A comunicação entre os envolvidos foi facilitada pela utilização de uma plataforma mobile com serviços do Parse Server.

PALAVRAS-CHAVE: *Dart(Flutter), Parse Server, Aplicativo Mobile.*

ABSTRACT: This study aimed to develop a service dissemination tool to assist self-employed workers, such as electricians, bricklayers, etc. In the application of the tool, it proved to be efficient by simplifying the processes of disclosing services and communicating between workers and customers. Communication between those involved was facilitated by the use of a mobile platform with Parse Server services.

KEYWORDS: *Dart(Flutter), Parse Server, Mobile App.*

1 INTRODUÇÃO

Com o crescimento do mercado de *smartphones* e conseqüentemente dos aplicativos, as pessoas passaram, cada vez mais, a usar o celular no dia a dia, seja para diversão ou seja para trabalho. O número de trabalhadores que prestam serviços de maneira autônoma aumentou, com isso, o mercado se tornou mais competitivo. Diante dessa competitividade, surge, então, a necessidade de se ter um diferencial, algo que possa tornar o trabalho mais atraente, prático e eficiente, para atingir as expectativas dos clientes.

Assim, o desenvolvimento dessa ferramenta tem o intuito de agregar valores aos serviços prestados pelos trabalhadores autônomos, bem como de facilitar o acesso dos clientes aos serviços por eles prestados. Em uma parte considerável dos casos analisados, os tipos de serviços citados anteriormente, são divulgados no *Marketplace*, grupos e páginas do *Facebook*, ou no site de compras e vendas da OLX. Tais plataformas têm o foco na venda e na compra de produtos, não em prestação de serviços.

O objetivo deste estudo foi criar uma ferramenta para os trabalhadores autônomos com o propósito de auxiliar o trabalho e a divulgação dos serviços prestados. Para isso foi considerado os dados básicos de divulgação dos serviços como a descrição das atividades desempenhadas, os valores dos serviços a serem prestados e a experiência dos trabalhadores. Com as informações tratadas pela aplicação, será

¹ Artigo apresentado por Adryel Klever Vieira de Almeida.

² Estudante de Graduação do Curso de Sistemas de Informação do UNIPAM. E-mail: adryelklever@unipam.edu.br.

³ Professor orientador. E-mail: mislene@unipam.edu.br.

possível listar todos os serviços cadastrados, filtrando por categoria e região. Com o serviço escolhido, o cliente poderá ver todas as informações como, valores, descrição dos serviços, nome do trabalhador e avaliação que outros clientes deram para o serviço. Para se atingir o objetivo geral, apresentam-se os seguintes objetivos deste estudo:

- Auxiliar a cadastrar, modificar e gerenciar as informações dos serviços;
- Aumentar a quantidade de informação disponível de cada trabalhador e seus serviços;
- Aumentar a comunicação entre o trabalhador autônomo e seus clientes;
- Melhorar a forma de pagamento para seus clientes.

Para desenvolver o aplicativo foi utilizado a documentação oficial do *Flutter* e *Dart* na versão *mobile*. Para o banco de dados, envio de *e-mail* e notificações, monitoramento de usuários, *dashboard* para controle de tarefas e *upload* de imagens, foi utilizado a documentação oficial do *Back4App*. Desse modo, segue a listagem e ferramentas utilizadas para o desenvolvimento do aplicativo e suas respectivas descrições:

- Dart (Flutter): Como linguagem de desenvolvimento;
- Android Studio: desenvolvimento do aplicativo;
- Visual Studio Code: gerar apk;
- Parser Server: Banco de dados e notificações;
- Back4App: Banco de dados;

Portanto o aplicativo foi desenvolvido contemplando todos os objetivos propostos para solucionar as necessidades dos trabalhadores autônomos, começando pela gerência dos serviços, que será realizada pelos seus *smartphones* em qualquer lugar, aumentar a rede de divulgação de seus serviços para que mais pessoas sejam alcançadas e assim, aumentando seus clientes, melhorar e facilitar a comunicação entre o trabalhador e cliente, e propor novas condições de pagamentos.

2 REFERENCIAL TEÓRICO

2.1 APLICATIVOS

Aplicativos são programas de software presentes em celulares *Android*, *iPhone* (*iOS*) e outros diversos dispositivos inteligentes, como *smart TVs*. Os *apps* podem ser gratuitos ou pagos. Eles desempenham diversas funções: mensageiro *online*, *streaming*, gerenciadores, editores de fotos e vídeos. Alguns já vêm instalados de fábrica, enquanto outros podem ser obtidos na *Apple Store* ou na *Play Store*. Os aplicativos facilitam a vida do usuário e se tornaram indispensáveis no dia a dia.

Nem todos os aplicativos buscam facilitar tarefas ou proporcionar entretenimento. O app nocivo, chamado pelo *Google* de “aplicativo potencialmente nocivo” coleta dados do usuário, como contatos e localização em tempo real, exhibe anúncios intrusivos e ainda deixa o dispositivo vulnerável para outros programas maliciosos. Esse tipo de app ataca principalmente os dispositivos *Android*. Eles podem se infiltrar na loja de aplicativos do *Google* e ser instalados por vários usuários antes que a fiscalização da loja detecte e ele seja removido da plataforma. Aplicativos

intrusivos costumam levantar suspeitas com permissões excessivas e desnecessárias. Já o *iOS* é um sistema operacional mais seguro. Segundo a *Apple*, ele foi projetado “com segurança em sua essência”.

2.2 MERCADO DE APLICATIVOS

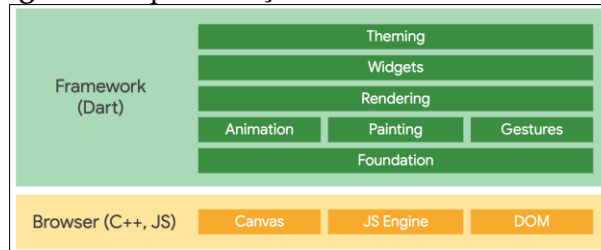
A indústria de aplicativos movimentou, de maneira virtual, o mundo todo. O Brasil aparece entre os países emergentes, trazendo grande potencial para desenvolvedores e investidores. O Brasil é o segundo mercado de aplicativo que mais cresce no mundo estando atrás apenas da Indonésia e à frente da Coreia do Sul. O uso de apps no Brasil é alto; os brasileiros passam cerca de três horas por dia utilizando aplicativos. Na média, de 70 a 80 softwares são instalados no smartphones, sendo que 30 a 40 são efetivamente utilizados. Enquanto o mercado global de smartphones apresentou uma leve queda durante 2019, o setor brasileiro foi pelo caminho oposto, registrando a venda de *smartphone* maior em relação ao ano de 2018. Os *smartphones* intermediários ajudaram nessa entrada.

Aplicativos de transporte, apps de relacionamento, *delivery* e robôs são os aplicativos mais utilizados no dia-a-dia. Enquanto há alguns anos era necessário ligar o computador ou até mesmo fazer ligações para realizar uma compra, hoje basta abrir o aplicativo no celular e ter todas as funções desejadas em poucos cliques. Essa é uma das principais razões pelas quais o mercado de aplicativos tem ganhado tanto espaço e está se expandindo rapidamente.

Com o isolamento causado pelo coronavírus, os serviços do apps tiveram um aumento significativo durante a quarentena. Aplicativos de *delivery* apresentaram um aumento de 15% em suas instalações entre os meses de fevereiro e março. Empresas apontam cada vez mais o aumento no volume de vendas online originadas de dispositivos móveis, principalmente em datas comemorativas. O mercado *mobile* está cada vez mais aquecido e não há previsões de quedas expressivas no setor. Investir no desenvolvimento de apps, na construção de plataformas digitais sólidas e robustas, é apostar em um mundo com possibilidades quase infinitas onde pode-se conectar mais rapidamente empresas, fornecedores e consumidores finais.

2.3 FLUTTER

O *Flutter* é dividido em duas camadas em sua *Engine*. A primeira é responsável por todo o Framework *Flutter* escrito em *Dart*. A segunda é algo que dificilmente terá que se preocupar, pois se trata do core do *Flutter*, onde o cérebro dele está onde estarão tratamentos específicos de cada sistema e a *engine* gráfica (*OpenGL*ES), que é o diferencial do *Flutter* com *React Native*.

Figura 1: Representação das 2 camadas do *Flutter*

Fonte: flutter.dev/imagens/Dart-framework-v-browser-framework.png, 2020.

Como o *Flutter* tem sua própria *engine* de renderização, ela não necessita de um *Bridge* específica para cada sistema executar o código. No *React Native*, por exemplo, sempre que você usa um componente de *View*, basicamente debaixo dos panos. Ele está chamando uma *View* em cada sistema específico, que de alguma forma ou outra, trará um gargalo para as *views* mais complexas.

2.4 PARSE SERVER

Em 2011, a empresa Parse Inc. foi fundada e disponibiliza o serviço com o mesmo nome: Parse. O Parse permite que os desenvolvedores mobile armazenem dados na nuvem, utilizando métodos de autenticação (e-mail e redes sociais) e enviem notificações por *push*. Com o serviço, desenvolvedores contam com um *backend* escalável, sendo capaz de lançar aplicações mobile em tempo recorde, sem a necessidade de se preocupar com a infraestrutura e gerenciamento de servidores.

Em 28/01/2016 o *Facebook* anuncia o fim do serviço *Parse*, com desligamento no prazo de 1 ano. Milhares de desenvolvedores ficariam preocupados e teriam que migrar todos os seus aplicativos para outros serviços. Com isso, o *Facebook* decide abrir o código do Parse, assim nenhum desenvolvedor teria que migrar seus aplicativos para outros serviços. Assim nasce o *Open Source Parse Serve*. Com isso, várias empresas ofereceram *hosting* para o Parse Server e suas características são:

- Projeto: *Open Source*;
- Arquitetura: *Node.JS, Express, MongoDB / PostgreSQL, API Rest / GraphQL*;
- Pode ser utilizado com outros bancos de dados através de Adaptadores;
- Possibilidade de desenvolver e testar localmente;
- Várias opções de hospedagem;
- Melhorou várias deficiências do Parse (limites de conexões, live queries, etc);
- Mantido com recursos financeiros da própria comunidade.

Disponibiliza *SKD's* e bibliotecas para várias tecnologias:

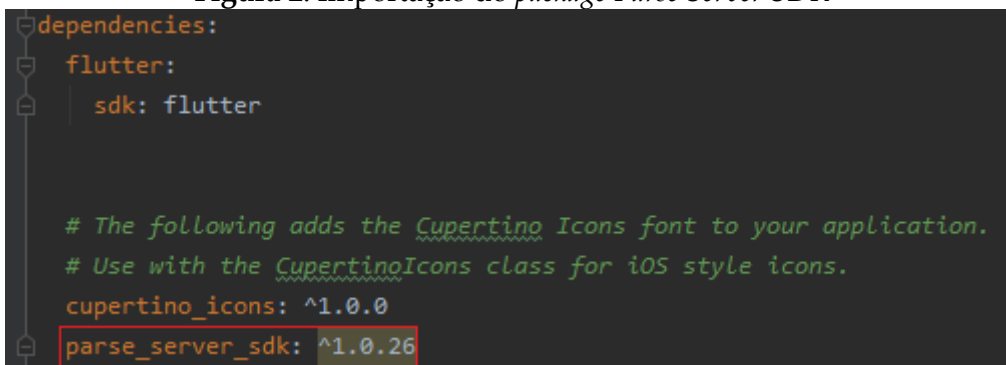
- *iOS + MacOS + tvOS*;
- *Android*;
- *JavaScript*
- *.NET + Xamarin*;
- *Unity*;
- *PHP*;
- *Arduino*;
- *Embedded C*.

- REST API;
 - GraphQL;
 - Go;
 - Python;
 - Ruby;
 - Flutter.
- É possível integrar o Parse Server com:
- Parse Server Hosting;
 - Back4app.com;
 - Sashido.io;
 - Servidor Próprio (VPS);
 - Hostgator;
 - AWS;
 - Azure;
 - Google.

3 DESENVOLVIMENTO E RESULTADOS

Entre os fatores importantes para a elaboração deste trabalho está a utilização do Parse Server. Para utilizar o *Parse Server* é necessário informar o nome do *package* no arquivo *pubspec.yaml* criado pelo próprio *Flutter*. O *package* deve ser informado em *dependencies > flutter > sdk*, como mostra a figura 1. Após informar o nome do *package*, é preciso clicar em Pub get para que o *package* seja instalado na aplicação e seja possível importá-lo em qualquer arquivo dentro do projeto. Todo *package* que será usado no *Flutter*, deve ser nesse caminho descrito anteriormente.

Figura 2: Importação do *package Parse Server SDK*



```
dependencies:
  flutter:
    sdk: flutter

  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.0
  parse_server_sdk: ^1.0.26
```

Fonte: Dados da Pesquisa, 2020.

Após criar um projeto no Back4App, é possível encontrar a opção *Server Setting* onde há informações importantes para realização da conexão do *Flutter* com o *Parse*. Essas informações são: *App Id*, *Parser API Address* e *Client Key*, como mostram as figuras, 2, 3 e 4. Essas informações devem ser mantidas em segredo; todas as informações mostradas nas imagens são de um projeto teste.

Figura 3: Server Settings**App Id**

KuyPutbQa40vcTTToN3tu9SwGPxKZggeLeX0CuNY

App Name

Sample Blog App

App Description**Parse API Address**<https://parseapi.back4app.com/>

Fonte: Dados da Pesquisa, 2020.

Figura 4: Server Settings**Client Key**

P0n30hY2BC3Jv5MBfSt9TsmX8DbbmMssAUsFqD48

Fonte: Dados da Pesquisa, 2020.

Figura 5. Server Settings**File Key**

6aa68d10-339a-484b-997f-7ad666f079a8

Fonte: Dados da Pesquisa, 2020.

Na figura 5, está sendo inicializado o *Parse*, informando o *App Id*, *Parse API Address* e o *Client Key*. O *autoSendSessionId* identifica quem está fazendo uma requisição no *Parse*, ou seja, não há necessidade de passar quem está fazendo uma requisição, todas as vezes que for fazê-la. Passando *true*, o *Parse* se encarrega de fazer todo esse trabalho. O *debug* é uma espécie de *print*, onde tudo que o *Parse* fizer, será mostrado no console do *Android Studio*.

Figura 6: Inicialização do Parse

```
await Parse().initialize(
    'KuyPutbQa40vcTTToN3tu9SwGPxKZggeLeX0CuNY',
    'https://parseapi.back4app.com/',
    clientKey: 'P0n30hY2BC3Jv5MBfSt9TsmX8DbbmMssAUsFqD48',
    autoSendSessionId: true,
    debug: true,
);
```

Fonte: Dados da Pesquisa, 2020.

Na figura 6, mostra um exemplo de como criar um objeto dentro do Back4App. No *response* tem *await* pois essa escrita não é síncrona e sim assíncrona, podendo gastar um tempo para fazer a inscrição no Back4App.

Figura 7: Criando objeto no Back4App

```
final category = ParseObject('Categoria')
    ..set('Title', 'Camisetas')
    ..set('Position', 2);

final response = await category.save();
```

Fonte: Dados da Pesquisa, 2020.

Na tela de login, o usuário tem a opção de *login* ou se cadastrar. A tela possui um slide onde o botão de ação muda o texto de acordo com a escolha do usuário. A opção “Ir para anúncios” dá liberdade para um cliente visualizar os serviços sem a necessidade de ter um cadastrado no aplicativo.

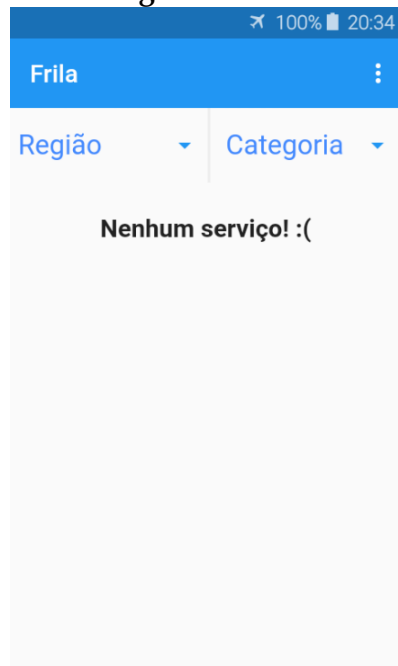
Figura 8: Tela de Login / Cadastrar



Fonte: Dados da Pesquisa, 2020.

Após acesso ao aplicativo, na tela *Home* o usuário pode visualizar todos os serviços cadastrados no *app*. Os serviços podem ser filtrados informando região e categoria.

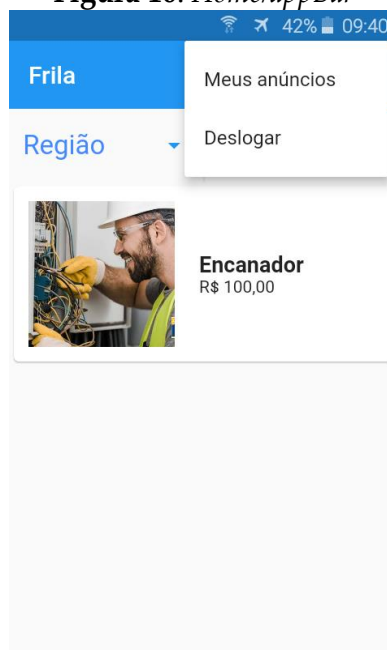
Figura 9: Home



Fonte: Dados da Pesquisa, 2020.

A *AppBar* que está localizada na tela Home, possui a funcionalidade “Entrar / Cadastrar” ou caso o usuário esteja logado, deslogar ou consultar seus serviços cadastrados.

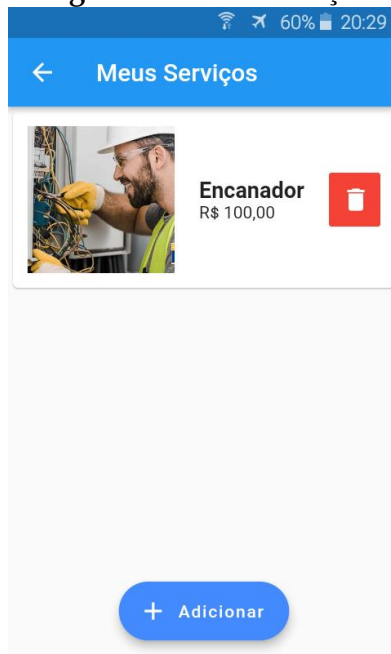
Figura 10: Home/appBar



Fonte: Dados da Pesquisa, 2020.

A tela “Meus Serviços” lista todos os serviços cadastrados pelo trabalhador autônomo, com a possibilidade de excluir quando quiser. Caso ele queira cadastrar um novo serviço, basta clicar no botão “Adicionar”.

Figura 11: Meus Serviços



Fonte: Dados da Pesquisa, 2020.

A tela “Novo Serviço”, deve ser informado até 5 imagens, a região, a categoria, título, preço, telefone e uma descrição com no máximo 200 caracteres.

Figura 12: Tela Novo Serviço.



Fonte: Dados da Pesquisa, 2020.

4 CONSIDERAÇÕES FINAIS

Esta pesquisa tem como intenção trazer praticidade e eficiência no trabalho dos prestadores de serviços (autônomos) e busca por serviços do dia-a-dia. O aplicativo foi desenvolvido para solucionar as necessidades do prestadores de serviço, a começar pela comodidade de cadastrar seus serviços prestados e poder gerenciá-los pelos seus smartphones em qualquer lugar com conexão à internet e também daqueles que precisam de um determinado serviço, podem procurar pelos prestadores em qualquer lugar, caso haja um prestador cadastrado na sua região.

Cada serviço cadastrado no aplicativo poderá ser avaliado por quem o contratou, ajudando outros usuários a encontrar os melhores prestadores de serviço dentro do aplicativo. Cada serviço contará com uma descrição, imagens, valor, número celular para entrar em contato, facilitando a comunicação e fechamento do negócio.

Dessa maneira, este estudo conseguiu atingir seus objetivos, alinhando a popularidade dos *smarthphones* com a necessidade de uma plataforma para unir prestadores de serviços autônomos e aqueles que necessitam de pessoas que realizam serviços do dia-a-dia. O aplicativo além de agregar valor ao trabalho dos prestadores de serviço, traz praticidade e uma maior objetividade para pessoas que precisam desses serviços. Melhorias na usabilidade, layout e novas funcionalidades como pagamento, chat, avaliação do serviço serão realizadas em trabalhos futuros com base nos *feedbacks* dos usuários, além de ajustes e funções disponíveis em outros aplicativos.

REFERÊNCIAS

APPLE. Apple, c2020. **App store**. Disponível em: apple.com/br/ios/app-store. Acesso em: 20 de fev. de 2020.

CIOLFI, Daniel. Udemy, c2020. **Criação de Apps Android e iOS com Flutter**. Acesso em: 02 de set. de 2020.

DAMASCENO, Jamilton. Udemy, c2020. **Desenvolvimento Android e iOS com Flutter**. Disponível em: udemy.com/course. Acesso em: 24 de ago. de 2020.

DÂMASO, Lívia. **Techtudo**, c2000. Notícias. Disponível em: techtudo.com.br. Acesso em: 14 de fev. de 2020.

DEMARTINI, Felipe. **CanalTech**, c2020. Apps. Disponível em: canaltech.com.br/apps. Acesso em: 28 de fev. de 2020.

FLUTTER. **Flutter**, c2020. Flutter Dev. Disponível em: flutter.dev. Acesso em: 02 de mar. de 2020.

LUCCA, Allysson. **O caminho dos Apps**, c2020. ed. Luccaco *be Digital, 2013.

SERVER, Parse, c2020. **Parse Server Documentation**. Disponível em: docs.parseplatform.org. Acesso em: 03 de mar. de 2020.

PLATAFORMA GERENCIAL PARA MARATONAS HACKATHONS¹

Lucas Bruno Ferreira²
Henaldo Barros Moraes³

RESUMO: O presente artigo apresenta a implementação de uma plataforma para gerenciamento de maratonas *hackathons*. Estas maratonas possuem três entidades fundamentais: a empresa organizadora, os jurados e os participantes imersos no evento. Através da plataforma em questão, todas as entidades envolvidas são beneficiadas, a empresa organizadora que, ao realizar estes eventos esperam diversificar seu banco de talentos de acordo com suas necessidades, pode de forma ágil e eficaz, gerenciar completamente a maratona e os participantes, reunindo informações que possam resultar em uma futura contratação. Aos jurados, é disponibilizado um painel para avaliação dos participantes, simplificando assim todo o processo, e aos participantes, a plataforma traz uma série de funcionalidades para auxiliar o desenvolvimento e aumentar sua visibilidade durante a maratona. Desenvolvida através de tecnologias atuais, como *ReactJS* e *React Native*, a plataforma pode ser acessada de qualquer dispositivo, seja ele um *desktop* ou *mobile* possibilitando ainda mais a interação evento e plataforma.

PALAVRAS-CHAVE: Sistemas gerenciais; *Hackathons*; *Software*.

ABSTRACT: The present study presents the implementation of a platform for hackathon marathon management. These events have three fundamental entities: the organizing company, the judges and the participants immersed in the event. Through the platform in question, all the entities involved are benefited, the organizing company that, when carrying out these events, hopes to diversify its talent bank according to its needs, can in an agile and efficient way, completely manage the marathon and the participants, gathering information that may result in future hiring. The judges are provided with a panel to evaluate the participants, thus simplifying the whole process, and to the participants, the platform brings a series of features to assist the development and increase their visibility during the marathon. Developed using current technologies, such as *ReactJS* and *React Native*, the system can be accessed from any device, be it a *desktop* or *mobile*, enabling even more the event and platform interaction.

KEYWORDS: Management systems; *Hackathons*; *Software*.

1 INTRODUÇÃO

O setor de TDIC (Tecnologias Digitais de Informação e Comunicação) vem crescendo mais a cada ano, segundo a BRASSCOM (Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação). Em 2019, este setor teve um crescimento de 8% e ocupou pelo terceiro ano consecutivo a sétima posição no ranking mundial.

¹ Artigo apresentado como avaliação parcial da disciplina de Estágio Supervisionado II, para obtenção de título de bacharel em Sistemas de Informação, pelo Centro Universitário de Patos de Minas – UNIPAM.

² Aluno Sistemas de Informação, UNIPAM. E-mail: lucasbrunoferreira@unipam.edu.br.

³ Mestre em redes de computadores, UFU. E-mail: henaldo@unipam.edu.br.

No entanto, o déficit de profissionais capacitados vem acompanhando este crescimento, o estudo “Achados e Recomendações para Formação Educacional e Empregabilidade em TDIC” apresentado em setembro de 2019 pela BRASSCOM, aponta que o Brasil tem cerca de 845 mil empregos e que a demanda projetada de mão de obra qualificada entre 2019 e 2024 está em torno de 70 mil profissionais, no entanto apenas 46 mil se formam por ano, para atender a esta demanda (GALLINDO, 2019).

Diante disto, as empresas deste setor necessitam se reinventar para atrair os melhores profissionais para contribuir com a evolução da empresa. Uma das formas de realizar a captura de talentos é através de maratonas de programação, conhecidas como *hackathons*, onde os candidatos são divididos em equipes e em um curto período de tempo são levados a solucionar algum problema, visando a inovação.

Através destas maratonas, os participantes são submetidos a várias experiências que agregaram em suas carreiras profissionais, tais como desenvolver o trabalho em equipe, liderança e até mesmo despertar o lado competitivo. “Os *Hackathons* podem ser vistos como um método direto para implementar a inovação orientada ao cliente e obter feedback externo sobre as práticas de negócios” (HERALA, 2019, p. 2).

A captura dos aspectos e habilidades individuais de cada participante é de grande importância para que as empresas organizadoras possam manter um banco de talentos eficiente diante suas necessidades. Todavia, estes eventos normalmente atraem uma grande quantidade de participantes, o que se torna um fator de complicação durante a avaliação dos projetos submetidos e seus participantes.

Diante deste problema, foi realizado o desenvolvimento de uma plataforma online para gerenciamento de maratonas *hackathons*, para tornar o processo como um todo, mais ágil e eficaz. Através da plataforma, os interessados podem realizar a inscrição no evento, informando sua área de atuação e habilidades que pretendem destacar durante o desenvolvimento do projeto. As informações dos candidatos podem ser acessadas através de relatórios pelos organizadores dos eventos, possibilitando a estruturação eficiente de um banco de talentos.

Este projeto tem como objetivo principal, auxiliar na organização de maratonas de inovação, principalmente aqueles voltados ao estilo de *hackathons*. Aos organizadores, a plataforma visa auxiliar na divulgação e inscrição dos participantes, e posteriormente realizar a captura de informações destes, possibilitando a seleção de profissionais com maior eficácia. Aos participantes, a plataforma tem como objetivo melhorar a interação dos mesmos durante a realização do evento, através da criação de equipe e gerenciamento do projeto no qual estão submetidos.

Para que o objetivo geral seja alcançado, os seguintes objetivos específicos serão desenvolvidos ao decorrer do projeto:

- *Registrar eventos*: Através da plataforma, os usuários que possuem acesso privilegiado podem criar e gerenciar eventos de suas instituições ou empresas.
- *Inscrição de participantes*: Permitir que pessoas interessadas no evento possam realizar a inscrição no mesmo e informar suas principais características e habilidades que pretendem agregar durante o desenvolvimento do projeto.

- *Gerenciar Projetos*: Os participantes cadastrados, podem descrever o projeto a ser trabalhado dentro da plataforma, informando por exemplo, nome e brasão para representar o mesmo.
- *Gerenciar Equipe*: Permitir aos participantes líderes de projetos, que convidem outros participantes para que possam atuar em equipe.
- *Avaliar Projetos*: Aos usuários denominados avaliadores, permitir a avaliação dos projetos submetidos de acordo com tópicos definidos pelos organizadores do evento.
- *Painel de resultados*: Disponibilizar o resultado da avaliação dos projetos em tempo real para que todas as equipes possam acompanhar.
- *Gerar relatórios*: Aos organizadores, disponibilização de relatórios contendo as informações dos candidatos e seus projetos.

2 REFERENCIAL TEÓRICO

Nesta seção são apresentados conceitos e estudos realizados referentes a maratonas *Hackathon*, Sistemas de informação gerencial e Análise de dados.

2.1 MARATONAS HACKATHON

A palavra *Hackathon* é uma junção dos termos: *hack* e *marathon*, A palavra *hack* no âmbito de tecnologia, significa programar com excelência e o termo *marathon*, em português significa maratona.

Este tipo de evento já está presente no cotidiano de grandes empresas referência em inovação, como *Google* e *Apple*, para estimular a criatividade de seus funcionários e poder converter novos talentos. “As pessoas disponibilizam seu tempo para resolver um problema e desvendar dados cuja lógica muitas vezes transcende a técnica e se entrelaça com a política. Poucos vencem as maratonas, mas todos deixam legados de grande valia para a sociedade nesse processo” (SVAB, 2014).

O *networking* que é possível realizar nestes eventos é um dos grandes fatores que levam a inscrição dos participantes. Em grande parte das maratonas, são convidados empresários e mentores para auxiliarem e avaliarem os projetos e seus integrantes, possibilitando ainda mais a integração dos participantes diante do mercado.

2.2 SISTEMAS DE INFORMAÇÃO GERENCIAIS

Cada vez mais, a tecnologia está presente nas empresas, auxiliando em tarefas simples do cotidiano e até mesmo tarefas, que manualmente poderiam ser exaustivas. Os sistemas gerenciais são um dos principais agentes desta transformação digital, estes que fornecem conceitos, metodologias, técnicas e ferramentas para os executivos das organizações tomarem decisões baseadas em informações estratégicas, precisas, atualizadas e em tempo hábil (CARMO, 1999).

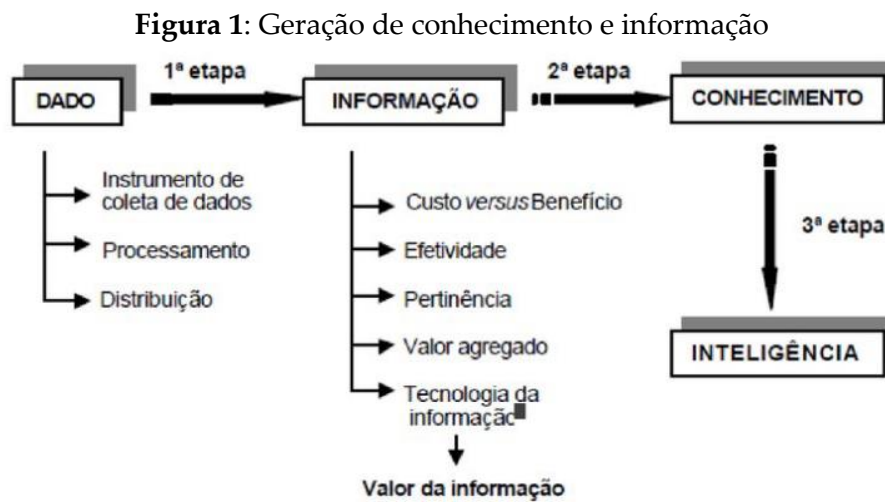
A implantação destes sistemas possui várias vantagens, independentemente do tamanho da empresa. A melhoria na produtividade e na tomada de decisão são os

principais pontos destacados, no entanto outras vantagens como acesso rápido e preciso de informações pode se tornar um fator interessante.

2.3 ANÁLISE DE DADOS

Diante de um mercado cada vez mais competitivo, buscar estratégias para amenizar essa pressão é inevitável. Através da ciência de análise de dados, pequenos e até mesmo grandes empreendedores conseguem alcançar vantagens importantes diante sua concorrência. Para que esses resultados positivos sejam alcançados, é importante reunir informações e transformá-las em índices de tomadas de decisão. As bases de dados surgem justamente da necessidade de se obter informações com mais rapidez, mais qualidade, mais eficácia e de forma selecionada (VALENTIM, 2001).

Antes da realização de qualquer manipulação de dados, é importante destacar a diferença entre dados, informação. Dados são registros sem dispersos, sem qualquer análise ou tratamento, podem ser definidos como a estrutura base da informação. Dado é necessariamente uma entidade matemática e, desta forma, é puramente sintático. Isto significa que os dados podem ser totalmente descritos através de representações formais, estruturais (SETZER. 2001). A definição de informação pode ser dada pela reunião de dados e estes possuem valor significativo ou lógico, ou seja, é o resultado do tratamento de dados. A Figura 1 apresenta uma perspectiva sobre a geração do conhecimento através de etapas, desde a coleta de dados até o tratamento dos mesmos.



Fonte: TJADEN, 1996, p. 91.

Portanto é fundamental para uma análise de dados eficaz, que a base de dados seja sólida. O tratamento destes dados pode ser realizado através de algoritmos e o resultado desse processo, normalmente, é apresentado de forma visual, através de gráficos e tabelas.

2.4 MODELO DE NEGÓCIOS

Conhecido também como *Business Canvas Model*, foi desenvolvido e defendido na tese de doutorado do teórico suíço Alex Osterwalder. Esta ferramenta é feita através de um diagrama de nove blocos que prestígiam as quatro áreas consideradas fundamentais de um negócio de sucesso, que são: clientes, oferta, viabilidade e infraestrutura. Para realizar o preenchimento do diagrama, é necessário se basear nas seguintes perguntas: “Como?” (infraestrutura), “O que?” (oferta), “Quem?” (cliente) e “Quanto?” (finanças). Esta ferramenta pode ser utilizada durante todo o ciclo de vida em uma empresa ou produto. Para empresas recém-criadas ou prestes a entrar no mercado, o *Canvas* auxilia por exemplo na segmentação de clientes. E para empresas que possuem sua base sólida no mercado, este modelo possibilita uma avaliação de desempenho do negócio.

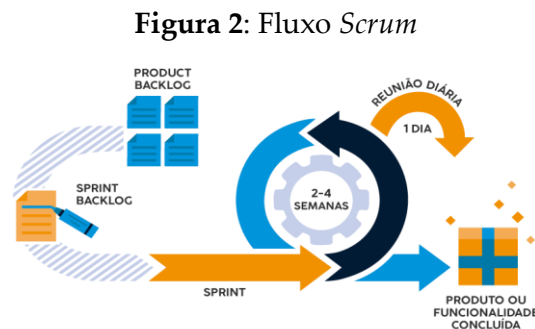
3 METODOLOGIA

Desenvolvido por Ken Schwaber e Jeff Sutherland, o *framework Scrum* será adotado como metodologia para o desenvolvimento deste trabalho. Nos últimos anos, o Scrum vem se tornando cada vez mais conhecido entre as empresas, principalmente em empresas do setor de tecnologia e desenvolvimento. As características como flexibilidade dos resultados, flexibilidade dos prazos, times pequenos, revisões frequentes e colaboração são definidas como premissas do Scrum. A metodologia reconhece que os processos de desenvolvimento subjacentes são incompletamente definidos e usa mecanismos de controle para melhorar a flexibilidade (SCHWABER, 1995).

Conhecido como uma metodologia ágil, o Scrum permite aumentar a produtividade através de ciclos de atividades planejados. Estes ciclos, são definidos como o principal evento do Scrum, normalmente possuem duração de 2 a 4 semanas e são chamados de *Sprints*.

Product Backlog são todos os requisitos e atividades a serem desenvolvidas até o fim do projeto. Estas tarefas são organizadas de forma a respeitar o tempo total de uma *Sprint*.

A Figura 2, demonstra o fluxo completo do Scrum, onde a cada iteração das *Sprints*, tem-se reuniões diárias, também conhecidas como *daily's*.



Fonte: Dados do projeto, 2020.

Seguindo a premissa da metodologia ágil apresentada, o projeto será realizado em 5 *Sprints* de 2 semanas cada, e terá a distribuição das atividades de acordo com a Tabela 1.

Tabela 1: Distribuição de atividades de acordo com a metodologia *Scrum*

Sprints	Atividades a serem desenvolvidas
Sprint 01	Prototipação das telas Arquitetura dos módulos do software
Sprint 02	Criação e autenticação de usuários. Gerenciamento de eventos.
Sprint 03	Gerenciamento de equipes Gerenciamento de projetos
Sprint 04	Avaliação de projetos e participantes Painel de resultados do evento
Sprint 05	Relatórios Dashboard

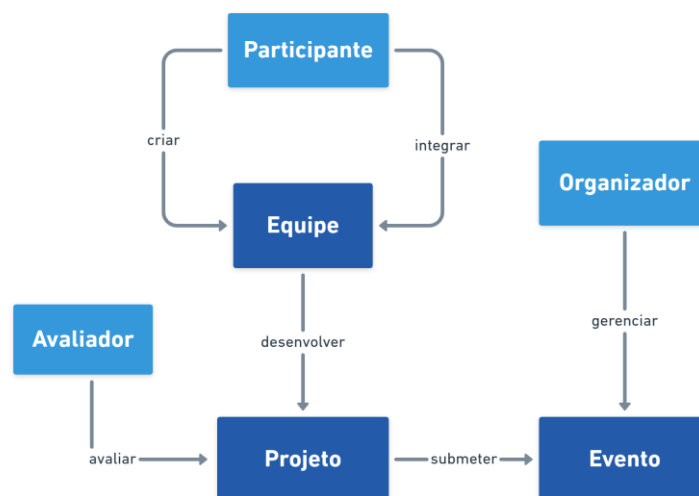
Fonte: Dados do projeto, 2020.

Para o desenvolvimento do sistema, as seguintes ferramentas serão utilizadas:

- *Visual Studio Code* - Desenvolvido pela empresa Microsoft, é um editor de código fonte, com suporte à depuração e diversos plugins para auxiliar na codificação.
- *Figma* - Ferramenta colaborativa para desenvolvimento de protótipos e designer de interfaces.
- *Github Projects* - Aplicação baseados em quadros, listas e cartões, para gerenciamento e organização de projetos.
- *Git* – Sistema de controle de versão distribuído.
- *Github* – Plataforma de hospedagem de código fonte.
- *ReactJS* - Biblioteca de código aberto, desenvolvido em JavaScript para criação de interfaces de usuário e aplicações *front-end*.
- *AdonisJS* - *Framework* para desenvolvimento *back-end* baseado em Node.JS.
- *PostgreSQL* - Banco de dados relacional de código aberto.

4 DESENVOLVIMENTO E RESULTADOS

A primeira etapa do desenvolvimento foi a definição do fluxo das funcionalidades, diante os objetivos gerais e específicos apresentados anteriormente. A Figura 3, apresenta todas as entidades e as respectivas ações que estas podem realizar diante os escopos do sistema.

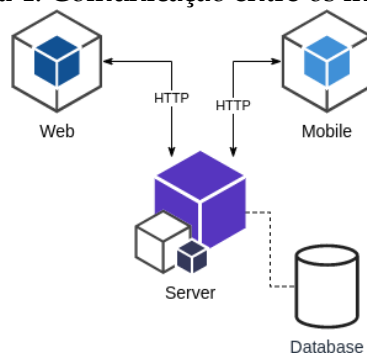
Figura 3: Fluxo de Funcionalidades

Fonte: Elaborado pelo Autor, 2020.

A segunda etapa foi composta pela definição da arquitetura do projeto, nesta etapa foram levantadas as ferramentas e tecnologias a serem utilizadas. O desenvolvimento foi realizado através de módulos, onde cada módulo possui sua própria responsabilidade. Foram desenvolvidos três módulos principais: *web*, *mobile* e *server*.

O módulo *web* é responsável pelo fluxo dos organizadores e participantes dos eventos, desenvolvido na linguagem *Javascript*, através da biblioteca de código aberto *React.JS* mantida pela empresa *Facebook*, este módulo pode ser acessado em qualquer navegador de internet. O módulo *mobile* desenvolvido em *React Native*, também mantida pelo *Facebook*, tem a responsabilidade de orquestrar todo o fluxo que os avaliadores estão presentes, visando a praticidade, este módulo está disponível para qualquer dispositivo móvel, seja ele um celular ou tablet, independentemente do sistema operacional. Podemos definir o terceiro módulo, *server*, como o coração da aplicação, desenvolvido também com *Javascript*, utilizando o *framework AdonisJS*, é responsável por todo o processamento e controle dos dados.

Como mostra a Figura 4, a comunicação entre os módulos é feita através do protocolo de comunicação *HTTP* (*Hypertext Transfer Protocol*).

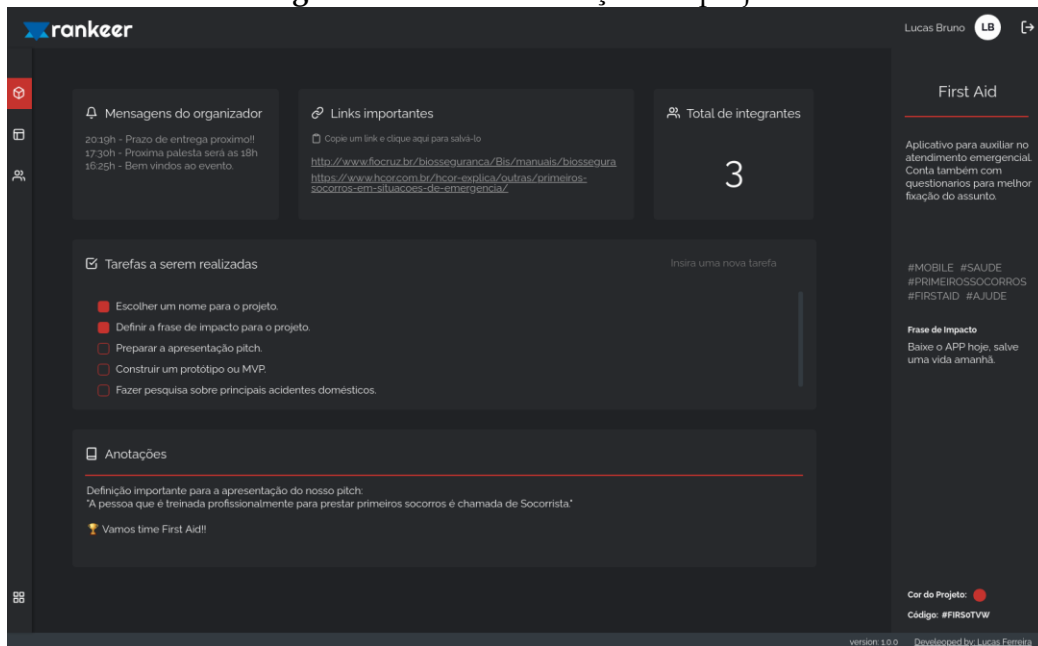
Figura 4: Comunicação entre os módulos

Fonte: Elaborado pelo autor, 2020.

Um dos fatores que levam empresas a criarem eventos *hackathons*, é o *networking* e a alta visibilidades que estes eventos trazem para a empresa organizadora. Todas as informações dos participantes são armazenadas em um banco de dados, possibilitando assim, posteriormente, que os organizadores possam traçar os perfis dos participantes e designá-los a possíveis vagas de empregos.

Como participante do evento, o usuário tem possibilidade de iniciar um novo projeto ou participar de um projeto criado por outros participantes. Ao ser integrante de um projeto, o sistema disponibiliza uma série de recursos para auxiliar a equipe durante o andamento do evento, como por exemplo armazenar informações, links e tarefas que a equipe visar importantes. Outro fator que o sistema possibilita para os participantes, é a possibilidade de customização da página do projeto representada na Figura 5, como cor do projeto, nome e frase de impacto, para tornar a imersão dos participantes mais agradável.

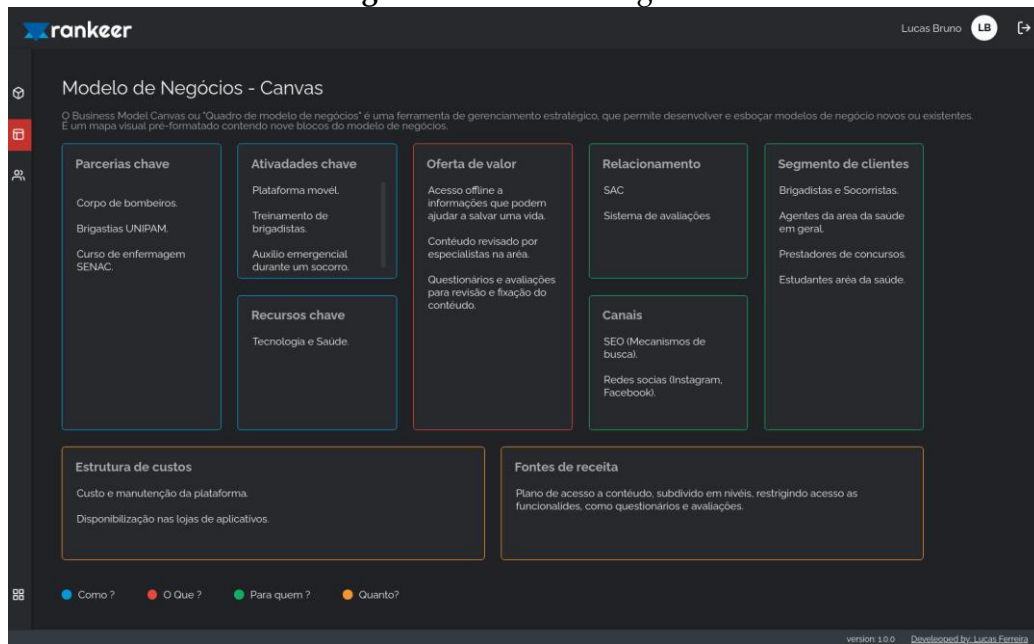
Figura 5: Tela de informações do projeto



Fonte: Dados do projeto, 2020.

Em grande parte das maratonas *hackathons*, os participantes são levados a discutir a viabilidade da ideia ou projeto que estão trabalhando. Uma das formas mais utilizadas, é a elaboração do modelo de negócios, ou normalmente chamado de *Canvas*. Considerada a função de maior importância dentro do sistema, ela possibilita que a equipe monte seu modelo, tendo as alterações realizadas nele, compartilhadas em tempo real entre os integrantes da equipe. Como podemos ver na Figura 6, o modelo é composto com 9 blocos editáveis, que podem ser preenchidos da forma que a equipe desejar.

Figura 6: Modelo de negócios



Fonte: Dados do projeto, 2020.

Durante os eventos *hackathons*, os organizadores também têm a possibilidade de exibir em tempo real o *ranking* dos projetos que estão sendo avaliados, como mostra a Figura 7. Esta é uma das funcionalidades que permite maior interação dos participantes no evento.

Figura 7: Ranking de pontuação

rankeer Avaliações em tempo real

Hack Summit III - Resultados

1	First Aid	Baixe o APP hoje, salve uma vida amanhã.	19 pontos
2	MobXdelivery	Praticidade para seu novo sistema de delivery.	16 pontos
3	Alpha Barber	Agende já seu horario sem sair de casa.	15 pontos
4	Help!	Prestação de serviços em uma unica plataforma.	10 pontos
5	iGirls	Tecnologia contra a violencia contra mulher.	9 pontos

Fonte: Dados do projeto, 2020.

A disponibilização do painel com o *ranking* possibilita uma maior interação no evento, aos participantes gera uma certa competição, pois os dados serão atualizados em tempo real, de acordo com as avaliações que estão sendo feitas pelos jurados do

evento. Para os organizadores, esta funcionalidade eleva a praticidade da administração do evento.

O sistema possui controle de permissões, onde cada usuário possui uma das respectivas atribuições dentro do evento: organizador, participante ou avaliador. Cada atribuição possui seu conjunto de ações permitidas, por exemplo, apenas o organizador pode iniciar e encerrar o evento, e apenas um avaliador tem acesso a página de avaliação de projetos. Estas dentre outras restrições permitem maior controle e organização dos eventos.

5 CONCLUSÃO

O desenvolvimento do presente trabalho possibilitou a análise mais profunda para a implementação de um sistema gerencial para eventos e maratonas *hackathons*. Colocando em prática aprendizados adquiridos durante o curso, tais como engenharia de software, banco de dados e redes de computadores e telecomunicações, os objetivos específicos propostos foram alcançados com sucesso.

Após finalizada a implementação do sistema, pode-se notar que algumas funcionalidades trouxeram grande diferencial ao sistema, como por exemplo a possibilidade dos participantes terem a possibilidade de preencher e apresentar o modelo de negócios, dentro do sistema em questão e também personalizar e gerenciar a página do projeto com informações relevantes para a avaliação do mesmo durante o evento.

O próximo passo para a evolução do sistema será colher feedbacks ao envolvê-lo em eventos que o Centro universitário de Patos de Minas (UNIPAM) realiza durante todo o ano, e que o sistema é totalmente compatível aos modelos propostos, tais como Prêmio de empreendedorismo e Startup Weekend.

REFERÊNCIAS

CARMO, Vadson Bastos do; PONTES, Cecília Carmen Cunha. Sistemas de informações gerenciais para programa de qualidade total em pequenas empresas da região de Campinas. **Ciência da Informação**, Brasília, v. 28, n. 1, p. 49-58, janeiro de 1999.

Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19651999000100007. Acesso em: 8 maio 2020.

GALLINDO, Sergio Paulo. **Formação educacional e empregabilidade em TIC**.

BRASSCOM, São Paulo, 9 de agosto de 2019. Disponível em:

<https://brasscom.org.br/estudo-brasscom-formacao-educacional-e-empregabilidade-em-tic-achados-e-recomendacoes/>. Acesso em: 8 maio 2020.

HERALA, Antti *et al.* Estratégia para dados: abrir ou hackear? *Journal of theoretical and applied electronic commerce research*, **Talca**, v. 14, n. 2, p. 33-46, 2019. Disponível em: https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-18762019000200104. Acesso em: 8 maio 2020.

SCHEWABER, K. SCRUM. **Development process**. 1995. Disponível em: <http://jeffsutherland.com/oopsla/schwapub.pdf>. Acesso em: 10 maio 2020.

SETZER, Valdemar. Meios eletrônicos e a educação: uma visão alternativa. **Datagrama**, São Paulo: Editora Escrituras, v. 10, n. 10, 2001.

SVAB, Haydée. Hackathon: o que é isso?. **Estadão**, São Paulo, 7 de fevereiro de 2014. Disponível em: <http://blogs.estadao.com.br/codigo-aberto/hackathon>. Acesso em: 11 maio 2020.

VALENTIM, Marta Lígia Pomim. Estrutura de dados: modelos de metadados e a qualidade de resposta. **Transinformação**, Campinas, v. 13, n. 1, p 67-68, junho 2001. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-37862001000100006. Acesso em: maio 2020.

QUICKCLASS: APLICATIVO PARA GERENCIAR FREQUÊNCIA UNIVERSITÁRIA VIA GPS¹

Gian Michel Nunes Fernandes²
Henaldo Barros de Moraes³

RESUMO: Este artigo aborda o estudo e a aplicação da tecnologia no processo de aprendizagem, visando identificar quais tecnologias se saem melhores neste processo e também visando o desenvolvimento de um aplicativo mobile e uma interface *web* que atenda a área da educação. Assim, será demonstrado o processo de criação deste aplicativo, que tem o propósito de auxiliar alunos e professores a registrar frequência de uma forma simples e rápida. Para o desenvolvimento deste trabalho foram utilizadas tecnologias como *React Native* para a criação do aplicativo mobile, *ReactJS* para a criação da interface WEB, e o banco de dados não-relacional *Firestore* para armazenamento dos dados na nuvem.

PALAVRAS-CHAVE: *React Native*, *Firestore*, Processo de Ensino.

ABSTRACT: This article will focus on the study and application of technology in the learning process, aiming to identify which technologies do better in this process and also aiming at the development of a mobile application and a web interface that serves the area of education. Thus, it will be demonstrated the process of creating this application, which has the purpose of helping students and teachers to register frequency in a simple and fast way. For the development of this work, technologies such as *React Native* were used to create the mobile application, *ReactJS* to create the WEB interface, and the non-relational database *Firestore* to store data in the cloud.

KEYWORDS: *React Native*, *Firestore*, Learning Process.

1 INTRODUÇÃO

Os avanços tecnológicos relacionados aos celulares têm possibilitado que os mesmos realizem diversas funções, com desempenhos cada vez melhores. No entanto, em relação ao uso no contexto educacional, os celulares dividem opiniões, principalmente quando utilizados em sala de aula. Se por um lado esse equipamento pode ser responsável por problemas, como distrações durante a aula, por outro pode apoiar tarefas pedagógicas (BATISTA; BARCELOS, 2013).

O uso de aparatos de comunicação para o ensino-aprendizagem não se trata de uma novidade, bem como o fascínio que exercem e o status de “solução” comumente atribuído a eles no contexto educacional. Com a diversificação e evolução tecnológica desses aparatos, reunidos na denominação Tecnologia Digital de Informação e Comunicação (TDIC), expectativas e argumentos vêm sendo gerados em

¹ Artigo apresentado como avaliação final da disciplina de Estágio Supervisionado II, para obtenção de título de bacharel em Sistemas de Informação, pelo Centro Universitário de Patos de Minas – UNIPAM.

² Estudante de Graduação do Curso de Sistemas de Informação do UNIPAM, email: gianmnf@unipam.edu.br

³ Mestre em redes de computadores, UFU, henaldo@unipam.edu.br

relação à apropriação dessas ferramentas para fins de ensino-aprendizagem (DA FONSECA, 2013).

Atualmente, os alunos possuem certa dificuldade para permanecer em sala de aula, pois estão sempre preocupados com a realização da frequência, para logo ir embora. Isso prejudica tanto o professor que deseja ministrar seu conteúdo sem interrupções como também o aluno, que acaba não tendo a devida atenção ao conteúdo, e posteriormente tem um rendimento bem menor comparado aos alunos que mantém a atenção e foco nas explicações do professor.

O avanço da internet e o aperfeiçoamento dos dispositivos móveis (celulares e tablets), aliados à diminuição dos preços destes dispositivos, permitiu ao Brasil atingir a marca de aproximadamente 154 milhões de smartphones (INFO ABRIL, 2015). Além disso, através de tecnologias relacionadas à geolocalização (serviços, dispositivos, sensores, redes de comunicação sem fio e respectivos conjuntos de dados), é possível haver uma integração dinâmica, dentro de um contexto, entre dispositivos digitais, lugares, informação e pessoas (LEMOS, 2009) (CHAVES NETO E BÔAVENTURA, 2015).

Com este avanço foi possível desenvolver novas tecnologias, como por exemplo, a computação em nuvem para facilitar o desenvolvimento de aplicativos cada vez mais robustos e inteligentes, capazes de executar tarefas de grande porte, como o envio de uma grande quantidade de dados, onde em meios antigos, levaria minutos, na nuvem podem levar apenas segundos, comparado com os bancos de dados locais que o Unipam possui, será de grande vantagem utilizar desta tecnologia para sempre ter essas informações acessíveis de forma segura e rápida. Este artigo tem como princípio desenvolver um aplicativo no qual através de seu uso, professores e alunos consigam fazer melhor controle de frequência utilizando de recursos como a geolocalização, além de oferecer funcionalidades simples e de fácil acesso para todos.

O propósito geral do projeto é tornar o processo de efetuar a frequência em algo descomplicado para que todos possam usar sem nenhum problema. Este aplicativo terá as seguintes funções: Consultar frequência, Consultar Aulas (Aluno), Cadastrar aulas (Professor), Registrar frequência, Obter lista de frequência (Professor).

No sentido de atingir o objetivo geral, os seguintes objetivos específicos serão implementados:

- *Consultar Frequência:* Mostrar ao aluno toda a sua frequência em todas as disciplinas de sua turma.
- *Consultar Aulas:* Permitir ao aluno fazer a consulta de todas as aulas que possui na semana.
- *Cadastrar Aulas:* Auxiliar o professor a gerenciar suas aulas de forma dinâmica, onde o mesmo poderá definir o local onde a aula será, e os alunos poderão efetuar a presença de forma instantânea.
- *Registrar frequência:* Permitir que o aluno marque que está presente em uma determinada aula, porém esta opção só é ativada quando o aluno estiver dentro do local determinado pelo professor para a aula ocorrer.
- *Obter lista de frequência:* Permitir que o professor possa obter a lista de todos os alunos presentes na aula de forma rápida e simples.

Este estudo visa o desenvolvimento de um aplicativo mobile que consiga auxiliar professores e alunos a terem uma melhor visão e uso da frequência nas aulas, evitando problemas que possam ser causados por descuidos ou falta de atenção, tanto pelo aluno quanto pelo professor.

Ao usar a chamada automática por aplicativo ao invés da chamada tradicional, o professor irá aplicar o conteúdo dentro de sala de aula de forma mais simples e objetiva, visto que o mesmo não precisará chamar aluno por aluno para registrar frequência. É a economia de tempo que todo professor precisa para ser mais produtivo em suas aulas.

O sistema que foi desenvolvido terá como foco principal evitar que os alunos faltem em aulas ou falsifiquem sua presença através de outros colegas de classe, usando um sistema totalmente online e realizado pelo smartphone, que empregará conceitos de geolocalização e computação em nuvem para certificar que não haja manipulação de informações.

O artigo segue a seguinte divisão: na seção 2 foram abordados alguns conceitos a respeito de *React Native*, Geolocalização, *Firebase* entre outros aspectos importantes para o desenvolvimento do mesmo, e na seção 3 será demonstrada a metodologia utilizada neste trabalho.

2 REFERENCIAL TEÓRICO

Nesta seção são abordados alguns conceitos a respeito de *React Native*, Geolocalização, *Firebase* entre outros aspectos importantes para o desenvolvimento deste artigo.

2.1 REACT NATIVE

Segundo Hudson (2016), atualmente é difícil trabalhar na web sem ter ouvido falar algo sobre *React*. Desenvolvido pelo *Facebook*, *Airbnb* está usando, assim como *Netflix*, *Uber* e outras empresas. Conforme Robbestad (2016), *React* não é um framework. *React* representa o V no padrão de design MVC (Modelo-Visão-Controlador) (SILVA, 2018).

Ele é uma biblioteca de *Javascript* para construção de interfaces de usuários que pode ser combinada com frameworks como *AngularJS*, *Ember* e *Meteor* ou em conjunto com outras bibliotecas *Javascript*, por exemplo o *Knockout*. *React* é baseado em componentes que podem ser encapsulados gerenciando seu próprio estado, uma vez que a lógica dos componentes está escrita em *Javascript* em vez de modelos, pode-se facilmente passar dados através do aplicativo e manter o estado fora do DOM (Modelo de Objeto de Documento) (SILVA, 2018).

2.2 GEOLOCALIZAÇÃO

Geolocalização é a arte de descobrir onde um usuário está localizado e, opcionalmente, compartilhar essa informação com outras pessoas e/ ou aplicativos. Diferentes métodos podem ser utilizados para se descobrir a localização de um

usuário: seu endereço *Internet Protocol* (IP), sua conexão de rede sem fio, a torre de celular que seu telefone está conectado, ou hardware GPS — dispositivo dedicado que calcula dados geográficos como latitude e longitude através de dados enviados por satélites (SANTOS, 2015).

Os primeiros métodos de descoberta da localização eram baseados no endereço IP, o que fornecia informações pouco confiáveis. Graças ao *HTML5* e sua API de Geolocalização, hoje é possível saber a posição do usuário de forma mais precisa, escrevendo aplicações que utilizam esse recurso (SANTOS, 2015).

2.3 FIREBASE

É uma plataforma poderosa do Google para armazenamento e sincronização de dados em tempo real. Provê uma variedade de soluções de desenvolvimento para acelerar a integração de recursos baseados em nuvem em aplicativos móveis e web (SMYTH, 2017). Além disso, provê infraestrutura necessária para construir grandes aplicativos, dando a possibilidade de crescimento e ganho (MORONEY, 2017) (SILVA, 2018).

2.4 CLOUD FIRESTORE

O *Cloud Firestore* é um banco de dados flexível e escalonável para desenvolvimento de dispositivos móveis, Web e servidores a partir do *Firebase* e do *Google Cloud Platform*. Como o *Firebase Realtime Database*, ele mantém seus dados em sincronia em aplicativos cliente por meio de *listeners* em tempo real (GOOGLE, 2020).

Seguindo o modelo de dados *NoSQL* do *Cloud Firestore*, você armazena dados em documentos que contêm mapeamentos de campos para valores. Esses documentos são armazenados em coleções, que são contêineres de documentos que você pode usar para organizar dados e criar consultas. Os documentos são compatíveis com muitos tipos de dados diferentes, desde *strings* e números simples a objetos complexos e aninhados. Também é possível criar subcoleções dentro dos documentos e criar estruturas de dados hierárquicas que podem ser escalonadas à medida que o banco de dados cresce (GOOGLE, 2020).

3 METODOLOGIA

O artigo foi desenvolvido tendo como princípio o estudo do uso de telefone celular aplicado aos sistemas voltados para a aprendizagem. Foi realizada uma revisão de literatura que irá auxiliar na identificação de tecnologias para o desenvolvimento do aplicativo. Foram fontes de consulta essenciais: livros, artigos, trabalhos acadêmicos que atendem a resolução deste problema.

Uma análise foi efetuada para identificar as aplicações e métodos que obtiveram maior sucesso no uso pelo aluno e pelo professor, a fim de promover um melhor ensino e uma melhor aprendizagem.

Na última etapa, foi desenvolvido um aplicativo protótipo para auxiliar na melhor aplicação da frequência em sala de aula, onde foi utilizado o serviço *Firebase*

para armazenamento de dados, em conjunto com a *API* de mapas do *Google Maps*, e desenvolvido um aplicativo usando a linguagem *React Native*, onde toda a sua estrutura está armazenada no *Google Cloud Platform*. Em um segundo momento, o app foi devidamente conectado a plataforma *Google Cloud*, e os devidos testes foram efetuados. Após todo o processo de revisão de literatura e as análises do aplicativo, foi possível identificar os métodos e funções que obtiveram maior sucesso e que serão implantadas no aplicativo proposto.

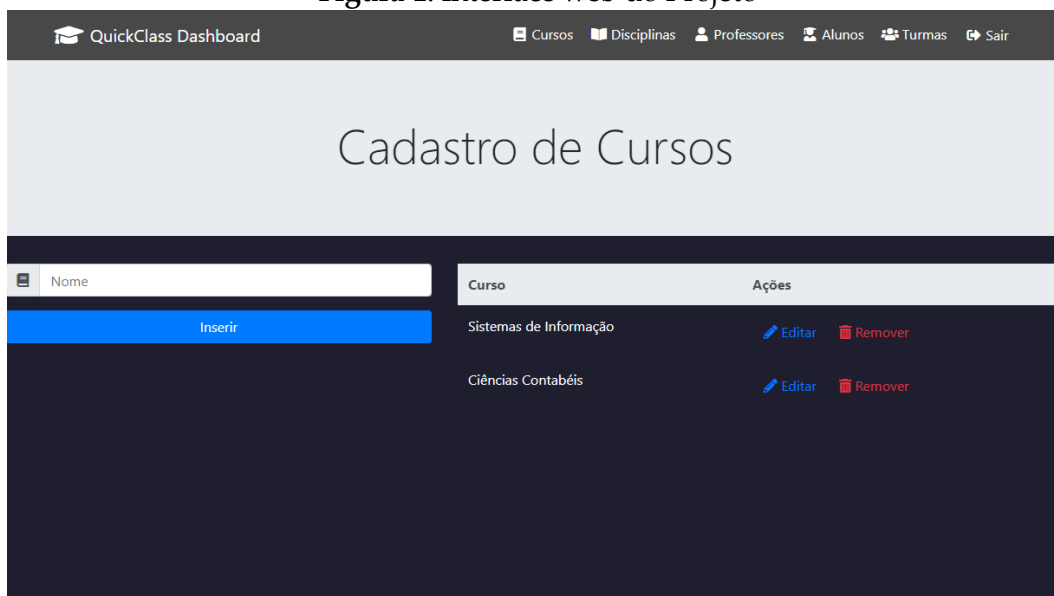
4 ANÁLISE E DISCUSSÃO DE RESULTADOS

Na fase inicial do projeto foram feitas pesquisas sobre linguagens para desenvolvimento mobile que fossem robustas o suficiente para atender os objetivos deste projeto. Também foram realizados testes com bancos de dados para saber qual teria melhor desempenho.

Foi utilizada a tecnologia *React* para o desenvolvimento de uma interface *web*, onde um administrador poderá efetuar a criação de turmas, cursos e fazer a pré-inscrição de professores e alunos no banco de dados.

Na Figura 1, é mostrada a interface *web* do projeto, sendo apresentados todos os menus possíveis de interação pelo administrador. Utilizando-se dessa interface, fica mais eficiente o uso do aplicativo pelos alunos e professores, pois tudo será cadastrado no banco de dados previamente, e com isso, não será necessário fazer um cadastro ao entrar no aplicativo, bastando entrar em uma conta do *Google* e utilizá-lo.

Figura 1: Interface web do Projeto



Fonte: Dados da Pesquisa, 2020.

Na tela apresentada pela Figura 1 é possível perceber todas as ações que o administrador pode efetuar:

Cursos: Permite que o administrador cadastre os possíveis cursos no banco de dados do projeto.

Disciplinas: Permite que o administrador efetue o cadastro das possíveis disciplinas no banco de dados do Projeto.

Professores: Permite efetuar um cadastro prévio de Professores, para que, ao entrar no aplicativo, o professor seja redirecionado para o local adequado.

Alunos: Permite efetuar um cadastro prévio de Alunos, para que, ao entrar no aplicativo, o aluno seja redirecionado para o local adequado.

Turmas: Permite que o administrador cadastre as possíveis turmas no banco de dados do projeto.

Para que o funcionamento da interface web e do aplicativo fosse rápido e fácil de usar, foi criada uma conexão com o *Firestore*, local onde todos os dados de ambas as partes estão armazenados.

Figura 2: Código de validação de dados

```
const checkExistence = obj=>{
  try {
    app.firestore().collection('turmas')
      .where('nome', '==', obj.nome)
      .get()
      .then(querySnapshot => {
        const result = [];
        querySnapshot.forEach((documentSnapshot) => {
          result.push({
            ...documentSnapshot.data(),
            key: documentSnapshot.id,
          });
        });

        if(result.length === 0) {
          addOrEdit(obj);
        } else {
          alerts.erro(`Já existe uma turma com este nome!`);
          setCurrentId('');
        }
      });
  } catch(err) {
    alerts.erro(`${err}`);
  }
}
```

Fonte: Dados da pesquisa, 2020.

Após realizar toda a configuração para efetuar o cadastro inicial de dados, foram criados métodos para validar se não existiria nenhuma duplicidade no banco de dados, como pode ser visto na Figura 2. Para que tudo isto ocorresse foram criados diversos alertas, onde a cada tentativa de inserção com duplicidade, o mesmo retornava a resposta de que já existia aquele mesmo dado no banco de dados e não permitia a reinserção.

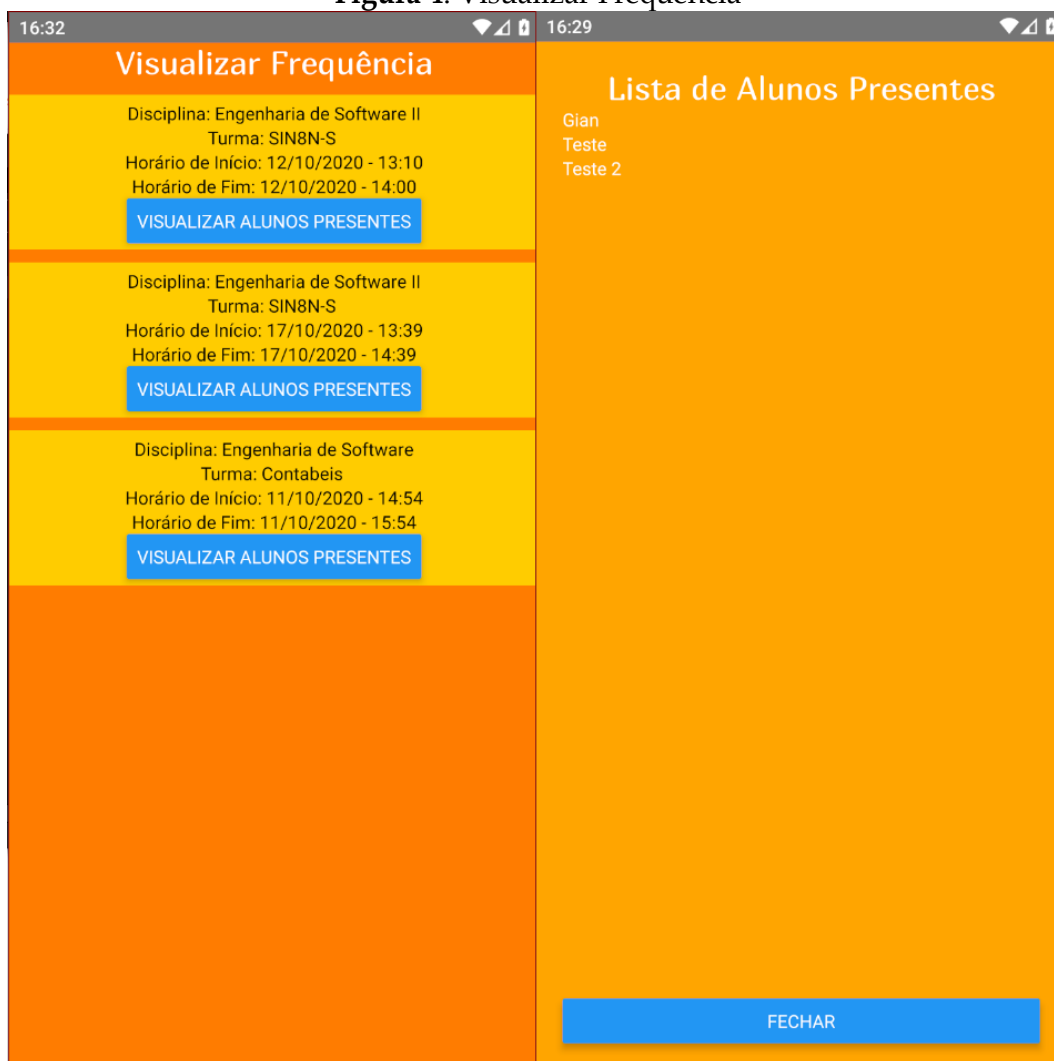
Para que todo o fluxo do projeto ocorresse normalmente foi criado também um aplicativo mobile na linguagem *React Native*, onde nele, professor e aluno pudessem gerenciar de forma bastante simples as suas aulas, disciplinas e afins. No escopo do professor, é possível cadastrar as aulas de acordo com cada curso e disciplina, como mostra a Figura 3. Ao efetuar este passo, o professor habilitará a função de Marcar Presença apenas para os alunos que estão próximos do professor, através das coordenadas do GPS.

Figura 3: Cadastro de Aulas pelo Professor

Fonte: Dados da pesquisa, 2020.

O professor também possui a função de visualizar toda a frequência dos alunos por aula ao clicar na opção visualizar frequência, onde ele será redirecionado para a tela apresentada na Figura 4. Nela basta selecionar a aula que deseja verificar a frequência, e então será apresentada uma lista de todos os alunos presentes naquela aula.

Figura 4: Visualizar Frequência

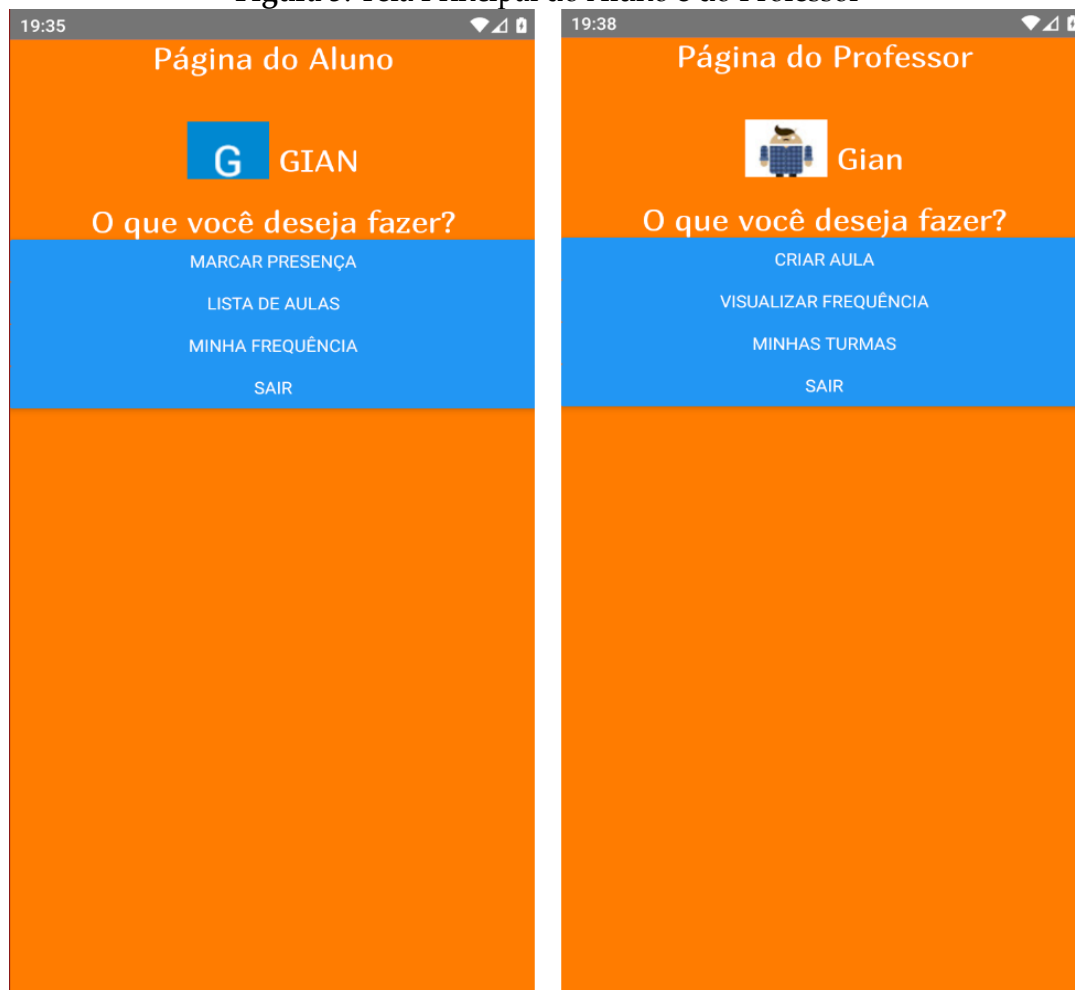


Fonte: Dados da pesquisa, 2020.

A última função que o professor possui é a visualização de turmas, onde ao clicar nesta opção ele pode obter uma lista completa de todas as turmas em que está vinculado no momento. Isto conclui a lista de funcionalidades referentes ao professor. A seguir serão apresentadas todas as funcionalidades disponíveis para uso pelo aluno.

A primeira funcionalidade disponível para o aluno é a de marcar presença, onde o mesmo ao tocar nela será redirecionado para uma tela onde é apresentada apenas a aula que está ocorrendo naquele momento, com um botão para que o aluno marque que está presente naquela aula. O aluno também conta com a funcionalidade de Lista de Aulas, onde ao tocar sobre o botão, ele será enviado a uma tela mostrando todas as aulas que o aluno tem ou terá naquele dia em específico. E por fim, o aluno também tem a funcionalidade Minha Frequência, onde poderá acompanhar sua frequência de acordo com cada aula dada em sua turma. Na figura 5 é possível ver a tela principal do professor e do aluno, mostrando todas as funções disponíveis para cada um.

Figura 5: Tela Principal do Aluno e do Professor



Fonte: Dados da pesquisa, 2020.

5 CONSIDERAÇÕES FINAIS

O estudo apresentado visou descrever técnicas para melhorar o processo de frequência em universidades, onde o mesmo tem o intuito de simplificar ao máximo a vida do aluno e do professor, para que com isso as aulas passem a ter um maior foco, e a frequência possa ser efetuada de forma mais rápida.

O aplicativo desenvolvido foi abastecido com dados para trabalhar com uma turma de Sistemas de Informação, porém esta proposta pode se expandir para alunos de outros cursos, necessitando apenas da inserção de dados dos mesmos através da interface *web* desenvolvida para este projeto. Além disto, podem ser desenvolvidas diferentes características, como a automatização do processo de presença, onde basta que o aluno esteja na sala para que a frequência dele seja marcada automaticamente, tornando esta solução extremamente importante no âmbito educacional.

Os objetivos deste artigo foram atingidos de forma parcial, sendo que mesmo com todas as etapas propostas sendo concluídas, os testes foram efetuados com apenas dois alunos de uma mesma turma. Para que o mesmo possa atingir seu potencial total, ele deve ser abastecido com diversos alunos de diferentes cursos. Por isso, acredita-se

que a sua solução desenvolvida poderá facilitar bastante a vida de professores e alunos, mostrando que a tecnologia pode ser fácil e atrativa de se lida.

REFERÊNCIAS

BATISTA, Sílvia Cristina Freitas; BARCELOS, Gilmara Teixeira. Análise do uso do celular no contexto educacional. **RENOTE-Revista Novas Tecnologias na Educação**, 2013.

SILVA, Werliton Carlos Sousa da. **Aplicações móveis nativas com react native e firebase: um estudo de caso**. Universidade Federal do Maranhão, 2018. Disponível em: <http://hdl.handle.net/123456789/3498>. Acesso em: 23 mar. 2020.

DA FONSECA, Ana Graciela Mendes Fernandes. Aprendizagem, mobilidade e convergência: mobile learning com celulares e smartphones. **Revista Mídia e Cotidiano**, 2013.

GOOGLE. **Cloud Firestore | Firebase**. CLOUD FIRESTORE, 2020. Disponível em: <https://firebase.google.com/docs/firestore?hl=pt-br>. Acesso em: 06 abr. 2020.

SANTOS JÚNIOR, Gesmar de Paula. **Desenvolvimento de sistema de geolocalização em realidade aumentada para multiplataforma móvel**. Universidade Federal de Uberlândia, 2015. Disponível em: <https://repositorio.ufu.br/handle/123456789/14599>. Acesso em: 06 abr. 2020.

CHAVES NETO, Gumercindo Rodrigues; BÔAVENTURA, Ricardo Soares. **Geolocalização mobile aplicado no sistema de estacionamento zona azul em Uberlândia: um estudo de caso**, 2015. Disponível em: <https://bit.ly/2PSxLEo>. Acesso em: 07 mar. 2020.

MORONEY, L. **The definitive guide to firebase: build android apps on Google's Mobile Platform**. 2017. Disponível em: <https://bit.ly/30RLgK8>. Acesso em: 23 mar. 2020.

SISTEMA DE ANÁLISE DE SENTIMENTO EM COMENTÁRIOS RELACIONADOS A UNIVERSIDADES DE ENSINO SUPERIOR EM REDES SOCIAIS (VENERA)¹

Wender Lucas Souza²

José Corrêa Viana³

Fernando Corrêa de Mello Júnior⁴

Henaldo Barros Moraes⁵

RESUMO: Este artigo aborda o estudo e a aplicação de análise de sentimentos em cima de comentários de usuários de redes sociais referentes a instituições de ensino superior a fim de gerar indicadores de satisfação e dados volumétricos visando o apoio e suporte à criação de estratégias de divulgação e marketing. Assim, será apresentado o processo de criação de agentes coletores desses comentários na internet, de módulos de análise de sentimento desses comentários coletados e da interface final do usuário para visualização e manipulação desses dados analisados. Para o desenvolvimento do trabalho foram utilizadas técnicas de *webscraping* e *webcrawling* utilizando a linguagem *Python*, serviços de análise de sentimentos do *Microsoft Azure Cognitive Services* com o uso de *NodeJS* e os *frameworks React* e *VueJS* para o desenvolvimento da interface final.

PALAVRAS-CHAVE: Marketing Digital e Social; Análise de Sentimento; Análise de Indicadores.

ABSTRACT: This article will focus on the study and application of sentiment analysis based on comments from users of social networks referring to higher education institutions in order to generate satisfaction indicators and volumetric data aiming to support the creation of dissemination strategies and marketing. Thus, the process of creating agents collecting these comments on the internet, of sentiment analysis modules of these collected comments and of the final user interface for viewing and manipulating these analyzed data will be presented. For the development of the work, webscraping and webcrawling techniques were used using the Python language, sentiment analysis services from Microsoft Azure Cognitive Services using NodeJS and the React and VueJS frameworks for the development of the final interface.

KEYWORDS: Social and Digital Marketing; Sentiment Analysis; Indicator Analysis.

1 INTRODUÇÃO

A mídia sempre foi um ótimo modo de exposição e divulgação de imagem, seja ela de forma pessoal, empresarial ou outros afins. A televisão e o rádio tiveram papéis significativos nessa empreitada, pois após a chegada da internet, tornaram-se

¹ Artigo apresentado como avaliação final da disciplina de Estágio Supervisionado II, para obtenção de título de bacharel em Sistemas de Informação, pelo Centro Universitário de Patos de Minas – UNIPAM.

² Estudante de Graduação do Curso de Sistemas de Informação do UNIPAM. E-mail: wenderls@unipam.edu.br.

³ Professor orientador. E-mail: jcorrea@unipam.edu.br.

⁴ Professor orientador. E-mail: fernandocmj@unipam.edu.br.

⁵ Professor orientador. E-mail: henaldo@unipam.edu.br.

veículos de menor intensidade de compartilhamento de informação, de acordo com o crescimento de tecnologias mais flexíveis como celulares, tablets e computadores e aplicativos que fazem com que as pessoas resolvam tudo de maneira prática e simples, como o *WhatsApp* (FERRARI, 2019), podendo causar um impacto significativo de acordo com as estratégias implementadas no seu uso. Atualmente, o ser humano encontra uma necessidade de ser socialmente visíveis para os outros, esforço o qual na maior parte é descartado pelos outros que simplesmente recebem uma quantidade absurda de informações no feed de notícias do *Facebook*, por exemplo. Em vista disso, é observado o quão exposto um usuário de rede social pode colocar suas informações pessoais ao alcance de qualquer um.

Como as mídias têm esse poder de dispersão de informação em frações de segundo, juntamente com a necessidade de consumo dessas informações pelos seus milhões de usuários diários, surgem oportunidades de redução de esforço na publicidade de certas organizações, como por exemplo, as Instituições de Ensino Superior. Essa proximidade permite a conversão de usuários para alunos agregados à Instituição de maneira rápida e eficiente. Uma promoção em um curso específico, mostras de estrutura e capacitação dos profissionais fazem com que a pessoa sinta vontade de realizar um curso superior. Outro fator decisivo que faz com que a procura por uma faculdade são as exigências cada vez maiores dos mercados de trabalho, que procuram sempre os melhores e mais preparados profissionais para atuar em determinada área. Para isso, as Instituições investem pesado em áreas comerciais voltadas à divulgação em massa de seu produto, como por exemplo, o Marketing Digital, Publicidade e Propaganda Digitais e Engenharia de Redes Sociais.

Da mesma forma que a divulgação acontece, a instituição também está exposta ao negativismo gerado na internet, pois um usuário mal-intencionado ou insatisfeito pode expor sua experiência ruim para outros usuários e esses podendo ser persuadidos e desistir de sua escolha. Como não existe um controle sobre o que está na internet, podem existir diversos comentários, bons ou ruins, sobre determinado assunto ou acontecimento que pode provocar sentimentos na sociedade. Em vista disso, é importante analisar a quantidade e o teor desses comentários sobre tal assunto para talvez assim reproduzir novamente decisões corretas e evitar o prejudicial, a fim de gerar valor tanto para as instituições quanto para as pessoas que estão relacionadas à mesma, seja pela prestação de serviços como pelo consumo do mesmo.

Neste contexto e com o propósito de contribuir com a efetividade dessas técnicas de divulgação e dar suporte na análise de dados das instituições de ensino superior, propôs-se, neste artigo, o estudo de técnicas de análise de sentimento em comentários voltados a essas instituições em redes sociais visando identificar indicadores de satisfação e possíveis estratégias de divulgação para auxiliar nas tomadas de decisões e na aplicação de marketing direcionado.

Considerando o cenário apresentado, foram realizados os seguintes objetivos específicos:

- Obter informações de maneira legal: conseguir os dados de forma anônima que não infringe leis de direitos morais, autorais e quaisquer fatores que envolvam exposição indevida de opinião ou argumento.

- Implementar uma análise de sentimento: realizar a análise de sentimento com base nos dados obtidos anteriormente por meio de raspagem de dados e consultas em *API*'s públicas.
- Definir estratégias de exibição dos indicadores: quais informações serão relevantes para as instituições e como exibi-las de maneira intuitiva e eficiente.
- Criar um ambiente de análise: tratar os dados de maneira eficaz e gerar indicadores de qualidade para as instituições.

Portanto, de acordo com os pontos apresentados acima, a análise desses indicadores e dados gerados pela aplicação servirá de material de apoio na elaboração de planos estratégicos por parte das instituições de ensino para a obtenção de novos alunos e demais interessados na área.

2 REFERENCIAL TEÓRICO

Nesta seção são apresentados conceitos e estudos realizados sobre raspagem de dados e *webcrawling* para obtenção de dados, análise de sentimentos e marketing digital voltado às Instituições de Ensino Superior.

2.1 RASPAGEM DE DADOS E WEBCRAWLING

Data scraping (do inglês, raspagem de dados, comumente chamado de *web scraping*) é uma técnica computacional na qual um programa extrai dados de saída legível somente para humanos, proveniente de um serviço ou aplicativo. Teoricamente, *web scraping* é a prática de coletar dados por qualquer meio que não seja um programa interagindo com uma *API* (ou, obviamente, por um ser humano usando um navegador web). (MITCHELL, 2018). Normalmente, a transferência de dados é feita utilizando-se estrutura de dados adequada para processos automatizados por computadores, e não por humanos. Tais comunicações de formato e protocolo são rigidamente estruturadas, documentadas, facilmente analisadas, mantendo a ambiguidade ao mínimo. Primeiro, é coletada a página alvo em forma de texto não estruturado, segundo, é determinado seus padrões recorrentes por trás da informação a qual se deseja e por último são aplicados esses padrões no texto não estruturado para se obter as informações desejadas (MUNZERT, 2014).

A raspagem de dados é mais frequentemente realizada em serviços ou aplicativos web legados ou aqueles que não oferecem uma interface de programação de aplicação. Neste último caso, geralmente os responsáveis pelo serviço ou aplicação consideram a raspagem de dados como indesejada, possivelmente pela sobrecarga do sistema, perda de receita por propagandas, ou pela perda do controle do conteúdo da informação.

A extração de dados é muitas vezes considerada *ad hoc*, uma técnica deselegante, frequentemente utilizada como último recurso quando não há outro mecanismo de intercâmbio de dados. Páginas *Web* são construídas utilizando-se linguagens de marcação baseadas em textos (*HTML* e *XHTML*) e frequentemente contêm uma riqueza de dados textuais úteis. No entanto, a maioria das páginas da web são projetadas para usuários finais humanos e não para uso automatizado. Devido a

isso, criou-se ferramentas que raspam conteúdo da web. Um *web scraping* é uma *API* para extrair dados de um web site. Empresas como a *Amazon AWS* e *Google* fornecem ferramentas de extração, serviços e dados públicos disponíveis sem custo para os usuários finais. Novas formas de *web scraping* envolvem capturar feeds de dados de servidores *web*. Por exemplo, *JSON* é comumente usado como um mecanismo de armazenamento de transporte entre o cliente e o servidor *Web*.

Recentemente, as empresas têm desenvolvido sistemas web que dependem de técnicas em análise do Modelo de Objeto de Documentos, visão computacional e processamento de linguagem natural para simular o processamento humano que ocorre ao visualizar uma página da web para extrair automaticamente informações úteis. Outro ponto bastante analisado é a legalidade da extração dessas informações, que quando essa informação coletada é utilizada para fins pessoais, não existe problema aparente, mas, se essa informação for republicada ou chegar a cunhos públicos sem os devidos reconhecimentos autorais ou permissões, isso pode acarretar problemas legais dependendo do tipo desse conteúdo (LAWSON, 2015).

Portanto, a aplicação se beneficiou dessa técnica para obtenção dos comentários em redes sociais para a realização da análise de sentimento em cima desses comentários obtidos, de forma legal e sem a exposição dos usuários autores, utilizando bibliotecas desenvolvidas pela comunidade com a linguagem de programação *python*.

2.2 ANÁLISE DE SENTIMENTO

Análise de sentimentos tem se tornado um importante tópico na *Web*, especialmente em redes sociais, com o desenvolvimento de aplicações para monitoramento de produtos e marcas, assim como a análise da repercussão de eventos importantes (BENEVENUTO, RIBEIRO e ARAÚJO, 2015). Nesse contexto, surgiram sistemas para tratar opiniões automaticamente, utilizando-se dos conceitos da área de Análise de Sentimentos (AS), também conhecida como Mineração de Opiniões (SILVA, LIMA e BARROS, 2012).

A análise de sentimentos refere-se ao uso de processamento de linguagem natural, análise de texto, linguística computacional e biometria para identificar sistematicamente, extrair, quantificar e estudar estados afetivos e informações subjetivas com a tarefa de identificar se a opinião que foi expressa em um determinado texto é positiva ou negativa. A ascensão das mídias sociais, como blogs e redes sociais tem despertado interesse na análise de sentimento.

Com a proliferação de opiniões, avaliações, recomendações e outras formas de expressão on-line, a opinião se transformou em uma espécie de moeda virtual para empresas que desejam comercializar os seus produtos, identificar novas oportunidades e gerenciar suas reputações. Como as empresas estão cada vez mais interessadas em automatizar o processo de pesquisa sobre o que está sendo falado sobre elas, a compreensão das conversas nas mídias sociais, a fim de identificar conteúdo relevante, que permita tomar ações específicas, tem levado as empresas a buscar soluções de análise de sentimentos.

Portanto, a aplicação se beneficiou dessa tecnologia, disponibilizada pela *Azure Cognitive Services* para criar indicadores de satisfação e levantar dados importantes para as instituições, sendo aplicada para analisar os comentários obtidos.

2.3 MARKETING DIGITAL

“Marketing é a atividade humana dirigida para satisfazer necessidades e desejos por meio de troca”. Essa definição contém em si alguns dos aspectos essenciais do marketing, que é dirigido para satisfazer necessidades e desejos humanos e, dessa forma, precisa levar em consideração o público alvo antes de tudo, conhecendo-o para poder satisfazê-lo (GABRIEL, 2010, p. 28). O Marketing digital seria uma modalidade de marketing onde ações de comunicação são realizadas de modo que as empresas possam utilizar por meio da internet, da telefonia celular e outros meios digitais para divulgar e comercializar seus produtos, conquistando novos clientes e melhorando a sua rede de relacionamentos. Ele engloba a prática de promover produtos ou serviços pela utilização de canais de distribuição eletrônicos, para então chegar aos consumidores rapidamente de forma relevante, personalizada e eficientemente.

Esse tipo de marketing traduz-se em ações adaptadas aos meios digitais, de forma a obter, nestes canais, a mesma eficiência e eficácia do marketing direto e, simultaneamente, potencializar os efeitos do marketing tradicional. Na sua operacionalização são, normalmente, utilizados canais, meios e ferramentas digitais. Segundo Martha Carrer Cruz Gabriel (2008, Marketing de otimização de buscas na web), o termo “marketing digital” não existe. O que determinará o trajeto do marketing em si são as estratégias de uso de plataformas digitais ou tradicionais, sendo que a escolha por uma é definida no modo como as pessoas as utilizam.

Toda a construção de um plano de ação se inicia com uma visão, e, junto a ela, um objetivo a ser alcançado. A partir deste traço são definidos os meios a serem utilizados, para enfim atingir os resultados. O tratamento singular a cada cliente é o segredo para que a escolha de um plano de ação seja de fato apropriada, somente o conhecer de um cliente ou projeto em questão fará com que os esforços de mídia potencializem o trabalho de marketing, o tornando assim digital. Dessa forma, não é mais possível pensar em ações isoladas na internet ou nas redes sociais. Seja qual for seu negócio, com toda a certeza uma parcela significativa de seus consumidores é representada por usuários conectados que acessam a mídia. Assim, incluir estratégias digitais como parte de seu planejamento estratégico e de marketing, criando ações integradas que permitam utilizar melhor os recursos disponíveis (TORRES, 2018).

De acordo com o apresentado, a aplicação levanta dados e indicadores com os comentários coletados e analisados conforme dito nos tópicos 2.1 e 2.2 para o uso estratégico das instituições de ensino em pontos específicos de sua divulgação a fim de gerar novos LEADS.

2.4 CAPTAÇÃO DE ALUNOS PELAS INSTITUIÇÕES DE ENSINO (LEAD)

A captação de alunos consiste em uma gestão de estratégias de marketing, seja digital ou social, para conseguir alunos para determinados cursos e atividades realizadas pelas Instituições de Ensino. Essa prática tem como foco o uso de meios de comunicação para divulgação de eventos e promoções realizadas para a conversão em matrículas correntes. No início dos anos 90, surge a era do marketing de relacionamento, com esforço e ampliação do foco concentrado no cliente. A partir dessa era, é reconhecido o valor e o potencial de lucro na conservação dos clientes. Com o mercado inovando todos os dias, com os clientes exigindo cada vez mais preço e devido ao maior uso da tecnologia da informação na gestão dos negócios surgiu a necessidade de uma forte relação entre a empresa e o consumidor, para que os clientes continuem comprando (DA SILVA NILSEN e HÜSKES, 2008).

Em um primeiro momento, para as instituições que se viram obrigadas a estabelecer estratégias para evitar a diminuição do número de seus alunos, a noção de marketing envolvia basicamente a noção do senso comum: o marketing se caracterizava como venda ou propaganda (COLOMBO, 2005). Mas, com o passar do tempo, essas mesmas instituições viram que seria possível realizar ações para que, ao invés de tentar manter seus clientes, identificar potenciais clientes para novos contratos, estratégia a qual é denominada "lead". O lead é um potencial cliente que, de alguma forma, demonstrou interesse no produto ou serviço que a instituição oferece, seja entrando em contato para obter mais informações, acessando perfis nas redes sociais ou nos portais de ensino, entre outros, despertando o interesse por diversos motivos: formação profissional, recolocação no mercado, aquisição de novos conhecimentos e excelência acadêmica. Essa prática permite que mais alunos interessados se matriculem nas instituições e assim manter uma taxa alta de clientes, mesmo perdendo alguns mais antigos.

Com isso, temos uma relação entre o Marketing Digital, citado anteriormente, que visa dar suporte à geração de leads da instituição por meio da publicidade e estratégias de obtenção, onde a aplicação tem um importante papel em apontar em quais pontos a comunidade tem mais atenção nas instituições de ensino e que se focados, conseguem um nível maior de captação com menor recurso.

3 METODOLOGIA

A aplicação foi desenvolvida tendo por base o estudo de técnicas de análise de sentimento e coleta de dados na *web* por meio de raspagem de dados, além de pesquisas relacionadas a estratégias de divulgação, mídias e redes sociais. Após a realização das pesquisas, primeiramente foi desenvolvida uma *API* para coleta de dados (comentários nas redes sociais) na linguagem *Python*, linguagem de programação de alto nível, interpretada, de script, imperativa, orientada a objetos, funcional. Os dados coletados nessa *API* foram armazenados no MongoDB, um banco de dados de alta performance orientado a documentos e não relacional.

Em sequência foi desenvolvida uma *API* para a gestão desses dados utilizando *NodeJS*, interpretador de JavaScript assíncrono com código aberto orientado a eventos,

focado em migrar a programação do *JavaScript* do cliente (*frontend*) para os servidores, criando aplicações de alta escalabilidade (como um servidor *web*), manipulando milhares de conexões/eventos simultâneas em tempo real numa única máquina física. Essa *API* é responsável por realizar chamadas à *API Text Analysis* provida pela *Azure Cognitive Services*, gerando indicadores e dados relacionados ao processo de análise de sentimentos dos comentários coletados. Após isso, foi desenvolvida a interface da aplicação em *VueJS*, biblioteca *JavaScript* de código aberto com foco em criar interfaces de usuário (*frontend*) em páginas *web*. Na conclusão da conexão entre os módulos desenvolvidos, a aplicação é hospedada no *Heroku*, plataforma em nuvem como um serviço que suporta várias linguagens de programação.

4 DESENVOLVIMENTO E RESULTADOS

A princípio foram realizadas pesquisas sobre as redes sociais, o impacto que elas trazem aos usuários e qual a frequência de utilização das pessoas sobre esses meios de comunicação e interação social, para determinar quais ambientes seriam o foco da coleta de comentários e materiais para a análise da aplicação. Também foram realizados estudos nas tecnologias selecionadas para a criação dos componentes visuais, dos módulos de análise e dos serviços que são consumidos e utilizados para a gestão dos dados, desde o processamento até o envio para exibição, além da análise da estrutura das páginas das redes sociais na internet para a realização da coleta dos dados.

Após as pesquisas e estudos, foram criados os módulos de coleta de comentários nas redes sociais, utilizando *Python* e técnicas de *webscraping* e *webcrawling*, que permitem a automatização da procura e da obtenção de resultados, análise de conteúdos de páginas web e assim a coleta de dados que são relevantes para a aplicação como um todo. Na Figura 1, é apresentado um trecho de código que é responsável pela coleta automática de comentários em uma rede social.

Figura 1: Código de coleta de comentários no Twitter

```
modules | helloworld | @ helloworld
12 def twitterSearch(query, maximum=10):
13     options = Options()
14     options.headless = True
15     options.add_argument('--window-size=1920,1200')
16     options.add_argument('--disable-dev-shm-usage')
17     options.add_argument('--no-sandbox')
18
19     if os.environ.get('ENV') == 'development':
20         driver = webdriver.Chrome()
21         ChromedriverManager().install(), options=options)
22     else:
23         options.binary_location = os.environ.get('GOOGLE_CHROME_BIN')
24         driver = webdriver.Chrome(executable_path=os.environ.get(
25             'CHROMEDRIVER_PATH'), chrome_options=options)
26
27     url = 'https://twitter.com/search?q=' + query + '&src=recent_search_click&live'
28     finalComments = []
29     # finalComments = []
30     # finalComments = []
31     try:
32         driver.get(url)
33         time.sleep(2)
34
35         while True:
36             javascript = "window.scrollTo(0, document.body.scrollHeight);"
37             driver.execute_script(javascript)
38             time.sleep(2)
39
40             element = driver.find_element_by_tag_name("body")
41             commentsDiv = element.get_attribute("innerHTML")
42
43             soup = BeautifulSoup(commentsDiv, 'html.parser')
44
45             comments = soup.find_all('div', attrs={
46                 'class': 'css-90laoa r-1kayrb r-1q8b0m r-ab22e0 r-16dbaf1 r-ad920e r-bc0qeo r-bwqia r-quotes'})
47
48             for i in comments:
49                 if i in finalComments:
50                     continue
51                 else:
52                     finalComments.append(html.escape(i.text))
53
54             if len(finalComments) >= maximum:
55                 break
56
57     print('twitter crawling done!')
```

Fonte: Dados da pesquisa, 2020.

Após finalizar a coleta dos dados, os mesmos são armazenados em um banco de dados não relacional *MongoDB*, que é capaz de armazenar estruturas de dados de maneira documental e que apresenta alta performance no acesso e na manipulação de seus documentos. Com os dados armazenados, foi criada uma *API* em *NodeJS*, seguindo o padrão *REST* para comunicação com o banco de dados e para o envio dos comentários para a análise de sentimento, de acordo com os padrões de uso da Microsoft e o Azure Cognitive Services. Na Figura 2, é apresentado o trecho de código responsável pela obtenção dos documentos no banco de dados, na organização desses nos padrões e a requisição à Azure para receber a análise de sentimento.

Figura 2: Código de requisição à Azure para análise dos dados

```
src > controllers > analysisData.js > (6) index
You, a few seconds ago | 1 author (You)
1  "use strict";
2
3  const Data = require("../schemas/dataSchema");
4  const azureUtil = require("../utils/azure.util.js");
5  const returnUtil = require("../utils/return.util.js");
6
7  const index = async (req, res) => {
8    try {
9      const id = req.query.id;
10
11     const result = await Data.findOne({ _id: id }, async function (err, result) {
12       if(err) { return; }
13
14       return result;
15     });
16
17     const documents = azureUtil.buildRequest(result.comments);
18     const analysedData = await azureUtil.makeRequest(documents);
19     const analysis = returnUtil.buildAnalysisObject(
20       result,
21       documents,
22       analysedData
23     );
24
25     result.analysis = analysis;
26
27     console.log(result.analysis);
28
29     await Data.updateOne({ _id: id }, { analysis: analysis });
30
31     res.status(200).json({ message: "OK!" });
32   } catch(error) {
33     res.status(500).json({ error: "Error: " + error });
34   }
35 };
36
37 module.exports = {
38   index,
39 };
40
```

Fonte: Dados da pesquisa, 2020.

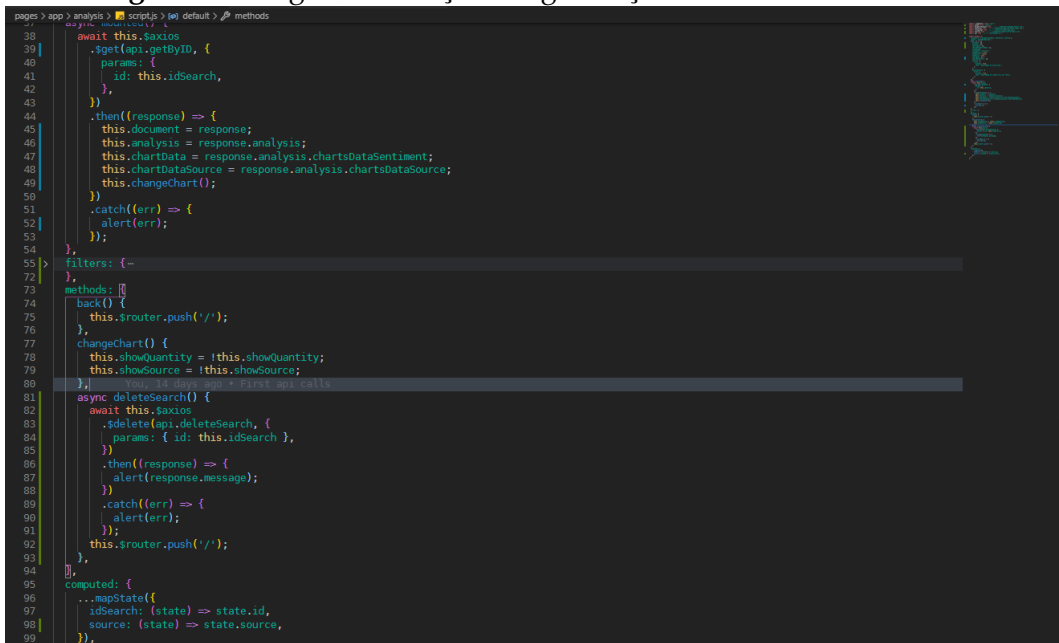
Assim que a análise é finalizada, os métodos de análise são responsáveis por gerar as estatísticas que serão consumidas pela interface da aplicação para a exibição dos indicadores, sendo realizados cálculos matemáticos simples, como médias, porcentagens e contagem dos dados, além de organizar dados relevantes, como sentenças que impactaram na análise de cada comentário e seus *scores* de avaliação.

Feito isso, o documento contendo os comentários é manipulado e assim é enviado para que a interface da aplicação consiga exibir os resultados da análise. Na interface de exibição, os dados são coletados por meio de requisições também no padrão *REST*, sendo manipulados pelos *scripts* em *JavaScript* interpretados pelo *VueJS*, organizados em componentes e propriedades que são persistidos no estado da aplicação para exibição e interação com o usuário. Na Figura 3 é apresentado um trecho do código de obtenção e organização desses dados.

Figura 3: Código de obtenção e organização dos dados da interface

```
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```



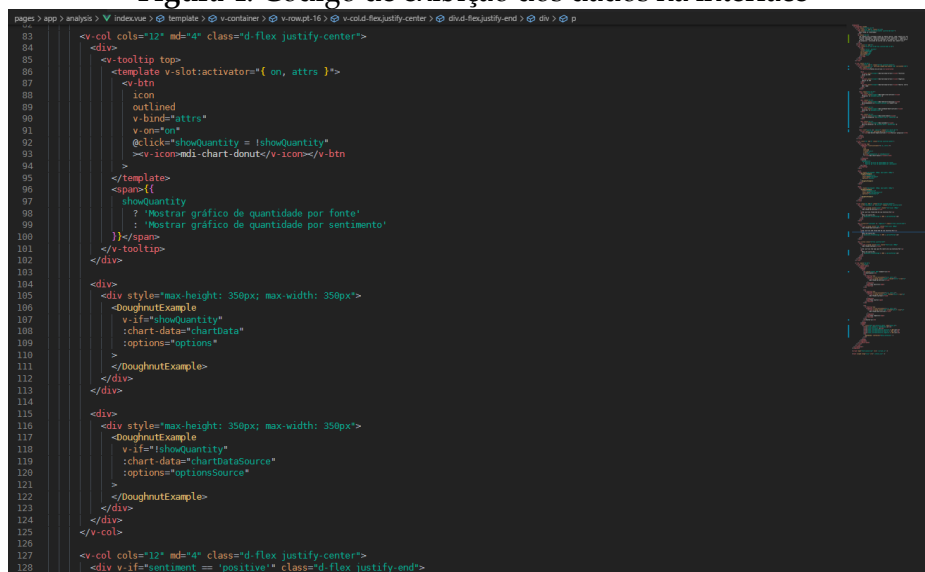
Fonte: Dados da pesquisa, 2020.

Com os dados organizados, o *VueJS* também é capaz de interpretar linguagem de marcação (*HTML*, *CSS*), com algumas propriedades nativas do *framework* de desenvolvimento *front-end*, permitindo a exibição dos dados de maneira limpa e organizada, juntamente com a possibilidade de configuração de visualização responsiva (dispositivos *mobile*) de maneira ágil e simplificada. Na Figura 4, uma representação dessa organização de componentes que serão visualizados em código *VueJS*.

Figura 4: Código de exibição dos dados na interface

```
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128

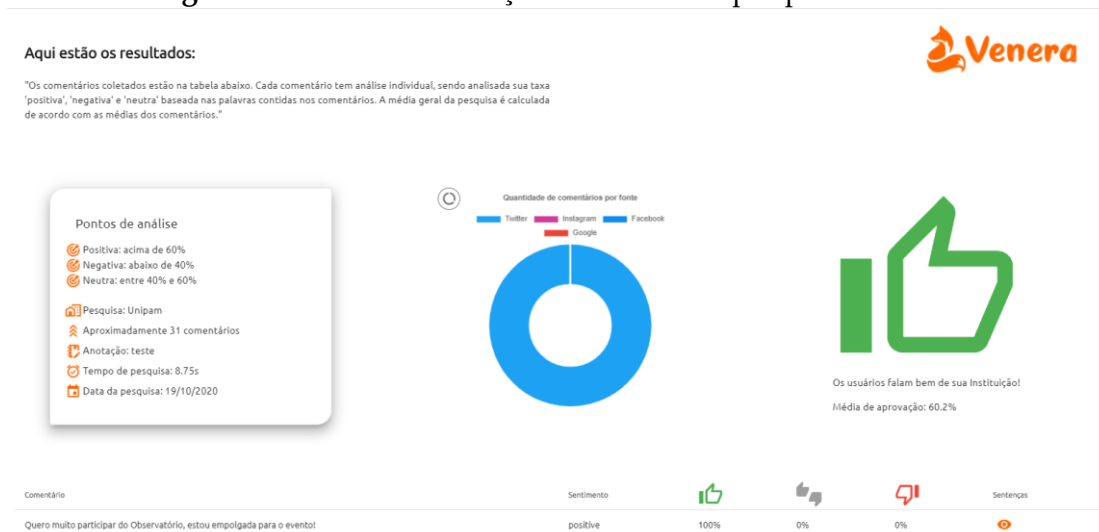
```



Fonte: Dados da pesquisa, 2020.

O resultado é a interface totalmente populada com os dados disponibilizados pela análise, processados pelos módulos de comunicação e conexão juntamente com a Azure, sendo apresentados de maneira intuitiva e simples, como apresentado na Figura 5.

Figura 5: Interface de exibição dos dados de pesquisa e análise



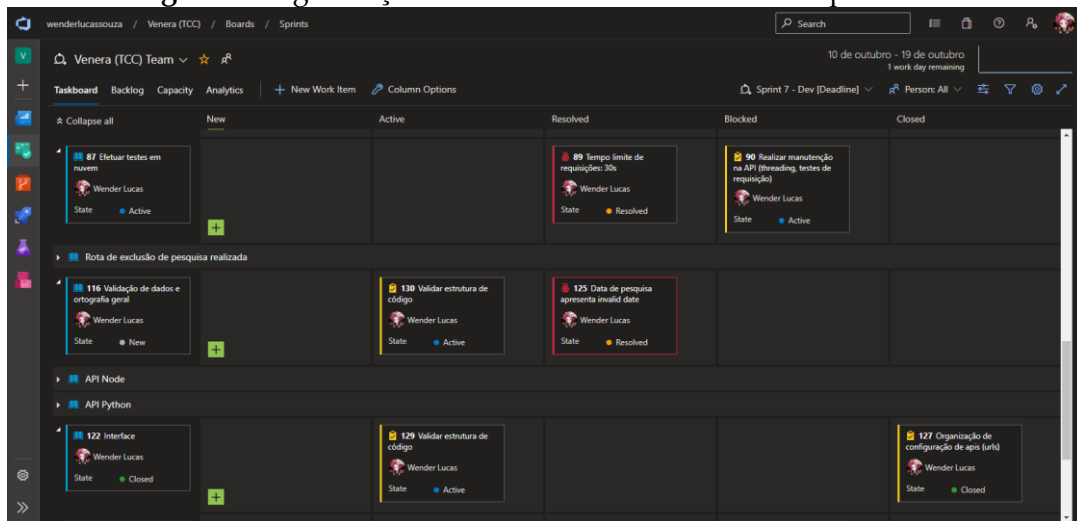
Fonte: Dados da pesquisa, 2020.

Na Figura 5, são apresentados os seguintes elementos:

- Pontos de análise e dados de pesquisa: São apresentados os pontos principais de análise da aplicação e os dados que foram realizados na pesquisa;
- Gráficos quantitativos: São apresentados os gráficos quantitativos por fonte de obtenção do comentário (rede social utilizada) e também por análise (positiva, negativa e neutra);
- Média geral de aprovação: É apresentada a média geral de aprovação de acordo com a análises dos comentários;
- Tabela de comentários: São apresentados todos os comentários utilizados na análise, seus *scores* individuais, as sentenças que mais impactaram na análise e o sentimento do comentário.

As tarefas realizadas durante o desenvolvimento da aplicação foram devidamente registradas e organizadas de acordo com o padrão de desenvolvimento ágil *SCRUM*, sendo utilizado o Azure DevOps para gerência das atividades, como mostrado na Figura 6.

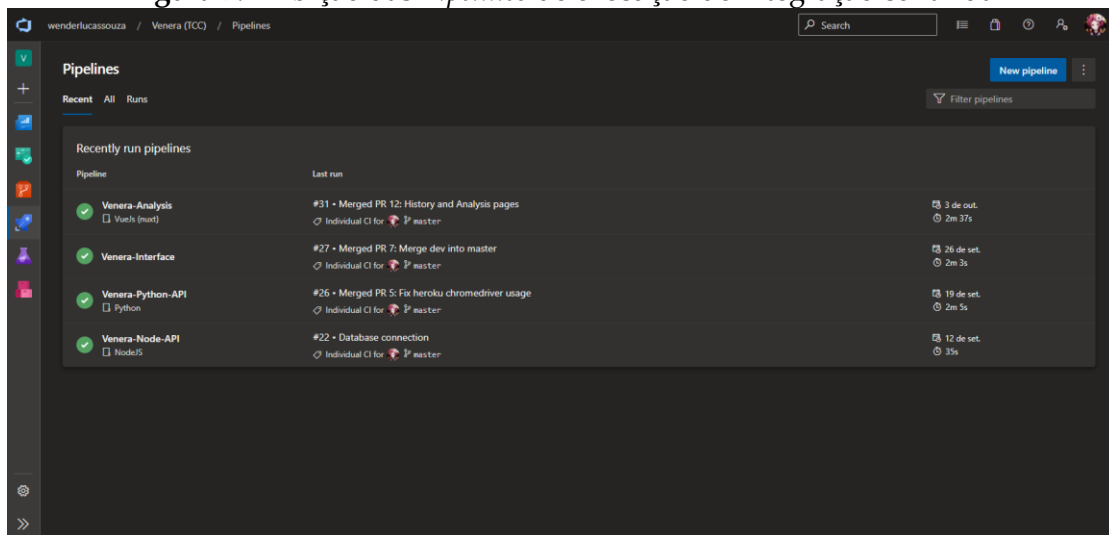
Figura 6: Organização das atividades no Azure DevOps Boards



Fonte: Dados da pesquisa, 2020;

Por fim, para que a aplicação fosse disponibilizada para uso, foram criadas *Pipelines* de execução de integração contínua com os serviços em nuvem para *deploy* e hospedagem dos repositórios contendo as aplicações. Para isso, foram configuradas essas *Pipelines* no Azure DevOps, integradas com o serviço de *cloud Heroku* e assim, basta acessar a aplicação pelo domínio disponibilizado na conclusão da integração. Nas Figuras 7 e 8 são exibidas as *Pipelines* e as execuções.

Figura 7: Exibição das *Pipelines* de execução de integração contínua

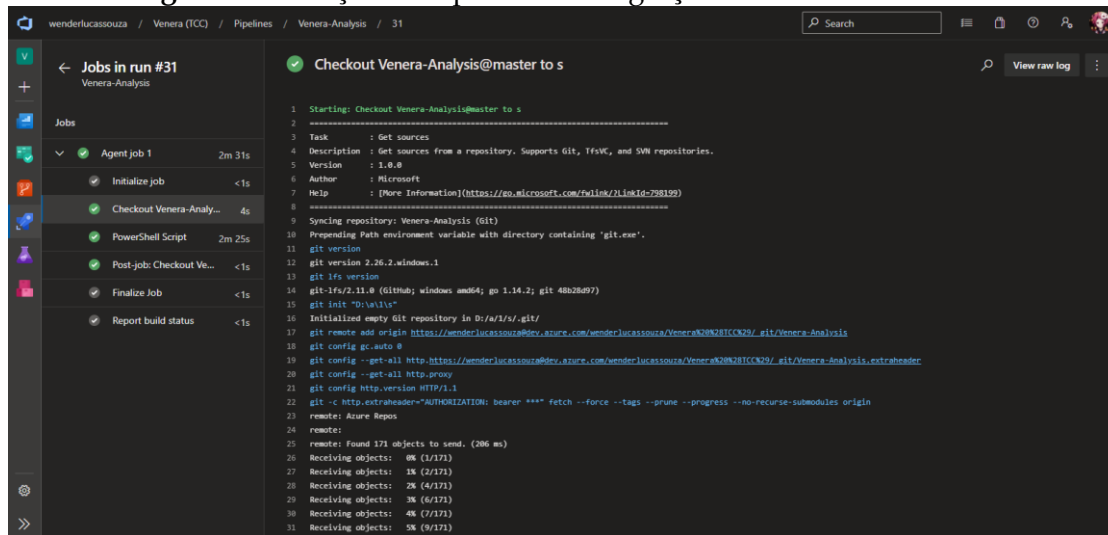


Fonte: Dados da pesquisa, 2020.

Como exibido na Figura 7, cada repositório contém suas *Pipelines* de execução independente, fazendo com que cada módulo do sistema consiga ser disponibilizado separadamente, porém se integrando internamente com os outros módulos para o funcionamento da aplicação como um todo. Na Figura 8, é apresentado um exemplo de execução de uma *Pipeline*, contendo informações de disponibilização, versão do

repositório e os comandos necessários para que a integração seja feita de maneira correta.

Figura 8: Execução da Pipeline de integração contínua da interface



Fonte: Dados da pesquisa, 2020.

De acordo com o sucesso das integrações e da implantação da aplicação em nuvem, a plataforma é disponibilizada para uso aberto.

5 CONCLUSÃO

O estudo demonstrado objetivou coletar comentários de usuários nas redes sociais, aplicar técnicas de Inteligência Artificial por meio de Análise de sentimentos em textos e converter os resultados das análises em indicadores de satisfação, dados estatísticos e expor a opinião desses usuários sobre determinados assuntos, principalmente voltado à instituições de ensino superior para que essas possam agir em cima desse feedback para a captação de novos alunos ou na melhoria de estratégias de marketing empresarial, social e geral.

Como dito anteriormente, o sistema desenvolvido foi projetado voltado para as instituições de ensino superior, porém essa proposta poderá se expandir para diferentes áreas, tanto no setor de educação quanto em outros setores gerais, os quais possam se beneficiar da análise de comentários nas redes sociais sobre determinada marca, assunto ou discussão.

Os objetivos deste artigo foram considerados alcançados parcialmente, pois com a execução de todas as etapas, foi conseguido o levantamento de dados estatísticos que comprovem o nível de satisfação/insatisfação relacionado a instituições de ensino, juntamente com opiniões sobre as mesmas, porém, com a visão de que o sistema possa atender a outras áreas, uma gama de possibilidades é criada a partir do conceito de que todo e qualquer comentário em uma rede social pode ter seu sentimento analisado e que esse ato pode gerar um indicador de satisfação relacionado a pesquisa realizada.

Tendo em mente o atual de mercado de trabalho, as exigências que o mesmo pede para que sempre sejam conseguidos os melhores profissionais nas mais diversas áreas existentes, o Ensino Superior e seu contexto social estão sempre em pauta, ao passo que as pessoas também procuram as melhores instituições para aprendizado e preparação acadêmica, e utilizando as ferramentas que a tecnologia nos propões nos dias de hoje, como por exemplo a ascendente Inteligência Artificial, para nos auxiliar no processo de unificação entre pessoas que buscam o conhecimento e a busca por profissionais de qualidade, o sistema poderá ser de grande utilidade nessa empreitada tanto para as Instituições de Ensino Superior quanto para seus possíveis clientes.

Para melhor aproveitamento e utilização do sistema em momentos futuros, a exploração de novas áreas além da educação superior é um diferencial que pode tornar a ferramenta ainda mais poderosa, a fim de gerar dados estatísticos a partir de qualquer problema ou assunto que se deseja realizar a pesquisa. Também, com o avanço dos estudos no campo da inteligência artificial ou do desenvolvimento voltado à web, a plataforma pode se tornar ainda mais eficiente e performática de acordo com a aplicação de outras técnicas de coleta e análise de sentimento, aumentando assim a precisão e a veracidade da análise, gerando dados ainda mais reais sobre a pesquisa e aumentando também a qualidade da tomada de decisões baseada no levantamento desses indicadores.

REFERÊNCIAS

AGUIAR, Sonia. **Redes sociais na internet: desafios à pesquisa**. In: XXX Congresso Brasileiro de Ciências da Comunicação. 2007.

BENEVENUTO, Fabrício; RIBEIRO, Filipe; ARAÚJO, Matheus. **Métodos para análise de sentimentos em mídias sociais**. In: Brazilian Symposium on Multimedia and the Web. (Webmedia), Manaus, Brazil. 2015. p. 30.

CAMACHO, Dóris. **Marketing digital: compra de mídia e inbound**. São Paulo: Editora Senac São Paulo, 2019.

COLOMBO, Sonia Simões. **Marketing Educacional em Ação: estratégias e ferramentas**. Porto Alegre: Bookman Editora, 2005.

DA SILVA NILSEN, Andréia; HÜSKES, Monika Ilse. Marketing de relacionamento para a fidelização e captação de alunos na Wizard Blumenau. **Revista Interdisciplinar Científica Aplicada**, v. 1, n. 2, p. 1-21, 2008.

FERRARI, Pollyana. **A força da mídia social: interface e linguagem jornalística no ambiente digital**. [S. l.]: Editora Estação das Letras e Cores, 2019.

GABRIEL, Martha Carrer Cruz. **Marketing de otimização de buscas na web**. São Paulo: Ed. Esfera, São Paulo, 2008.

GABRIEL, Martha Carrer Cruz. **Marketing na era digital**: conceitos, plataformas e estratégias. São Paulo: Novatec Editora, 2010.

LAWSON, Richard. **Web scraping with Python**. Packt Publishing Ltd, 2015.

MITCHELL, Ryan. **Web scraping with Python**: collecting more data from the modern web. "O'Reilly Media, Inc.", 2018.

MUNZERT, Simon et al. **Automated data collection with R: A practical guide to web scraping and text mining**. John Wiley & Sons, 2014.

SILVA, Nelson Rocha; LIMA, Diego; BARROS, Flávia. **Sapair**: um processo de análise de sentimento no nível de característica. *In*: 4nd International Workshop on Web and Text Intelligence (WTI'12), Curitiba. 2012. p. 2.

TORRES, Cláudio. **A bíblia do marketing digital**: tudo o que você queria saber sobre marketing e publicidade na internet e não tinha a quem perguntar. São Paulo: Novatec Editora, 2018.

ZIMMERMAN, Jan. **Marketing digital para leigos**. Tradução de Marcella de Melo Silva São Paulo: Alta Books Editora, 2018.

SISTEMA PARA GERENCIAMENTO DE ESTOQUE E FINANÇAS¹

Jefferson Nogueira Rocha¹

José dos Reis Mota²

RESUMO: Este artigo aborda sobre o desenvolvimento e análise de um sistema para gerenciamento de estoque e finanças para a empresa de vendas de roupas Dulci Modas da empreendedora Dulcineia Cristina. Assim, é apresentado o processo de estruturação e criação de um painel administrativo que possibilita gerenciar todas as compras e vendas dos produtos, bem como visualizar relatórios. Para o desenvolvimento do trabalho, foi utilizado a metodologia Programação Extrema, ou simplesmente XP, sendo uma metodologia ágil por se ajustar bem em desenvolvimentos com requisitos vagos e em constante mudança. Seus princípios básicos são *feedback* rápido, presumir simplicidade, mudanças incrementais, abraçar mudanças e trabalho de qualidade. As tecnologias escolhidas foram Angular para o painel, TypeORM para o backend, o serviço AWS S3 para armazenar imagens dos produtos, o serviço de banco de dados AWS RDS, o serviço de versionamento de código GitHub e o serviço de hospedagem Heroku implementando *deploy* contínuo com o GitHub. Ao final do estudo são apresentados a arquitetura escolhida e os resultados obtidos durante o seu desenvolvimento.

PALAVRAS-CHAVE: Empreendedorismo, Estoque, Finanças, Aplicação Web.

ABSTRACT: This article discusses the development and analysis of an inventory and finance management system for the clothing sales company Dulci Modas by entrepreneur Dulcineia Cristina. Thus, the process of structuring and creating an administrative panel that allows managing all purchases and sales of products, as well as viewing reports, is presented. For the development of the work, the Extreme Programming methodology, or simply XP, was used, being an agile methodology for adjusting well in developments with vague and constantly changing requirements. Its basic principles are quick feedback, assuming simplicity, incremental changes, embracing change and quality work. The technologies chosen were Angular for the panel, TypeORM for the backend, the AWS S3 service for storing product images, the AWS RDS database service, the GitHub code versioning service and the Heroku hosting service implementing continuous deployment with GitHub. At the end of the study, the chosen architecture and the results obtained during its development are capitalized.

KEYWORDS: Entrepreneurship, Inventory, Finance, Web Application.

1 INTRODUÇÃO

O comércio de roupas teve um crescimento acentuado nos últimos anos, apenas na última década o mercado brasileiro saiu da 14^a para a 8^a posição entre as maiores do mundo. Com esse crescimento acelerado, diversos eventos anuais de modas estão a ocorrer em grandes cidades brasileiras, como o São Paulo Fashion Week (São Paulo - SP), Dragon Fashion Brasil (Fortaleza - CE), Inspira+ (São Paulo - SP), Casa

¹ Aluno do curso de Sistemas de Informação do UNIPAM.

² Mestre em Ciência da Computação, UFU.

¹Aluno de Sistemas de Informações, UNIPAM, raphaelcustodio94@gmail.com

² Mestre em Redes de Computadores, UFU, henaldobarros@gmail.com

de Criadores (São Paulo - SP), Moda Rio Moda (Rio de Janeiro - RJ) e Denim Meeting que pode vir ocorrer em Balneário Camboriú (SC), São Paulo (SP), Maringá (PR), Goiânia (GO) e Caruaru (PE), é neste momento que reside uma grande oportunidade: o fomento de um tipo de economia com alto potencial de inclusão social e equilíbrio na relação entre indivíduos (HACO, 2019).

No Brasil, o comércio de vestuário movimentou cerca de R\$ 200 bilhões no ano de 2017 e existiam, ao fim do mesmo ano, 149,1 mil pontos de vendas, considerando lojas especializadas em vestuário, redes de pequenas lojas, lojas independentes e hipermercado, sendo que o que possui o volume mais representativo são as lojas independentes, detendo 36,5% do comércio (CLIENTESA, 2018).

Juntando esse crescimento com o fenômeno que está sendo a internet, o resultado é uma combinação muito forte que movimenta milhares de produtos entregues diariamente pelo país, onde o *e-commerce* ou comércio eletrônico, é um modelo de negócio que disponibiliza um catálogo de produtos ou serviços através de um *site*, lojas como Centauro, Netshoes, Dafiti, Zattini, são pioneiras no cenário de vestuário no Brasil.

Entretanto, para manter uma venda constante é preciso possuir um estoque com uma logística muito organizada para não se perder nas dezenas, centenas ou até milhares de pedidos, foi pensando nisto que a gigante Amazon conseguiu implementar uma automação em seus depósitos utilizando robôs Kiva, tendo um controle de estoque tão perfeito, é possível que toda a transação de mercadoria desde os *racks* até os funcionários que irão embalar e enviar os produtos para seus respectivos compradores. Esse controle é tão bem feito que o ciclo todo de procurar os itens que compõem o pedido até o operador de *picking* (separação e preparação de pedidos) teve uma redução de 30-45 minutos (ILOS, 2017).

Não apenas o controle de estoque, mas a administração financeira é igualmente importante para se manter um fluxo de entrada e saída de produtos, tendo não somente essas informações, mas também as de clientes e gastos que possam vir a ocorrer, resulta em um controle organizacional que nas mãos certa poderá fornecer uma base para a implementação de inteligência artificial que pode auxiliar em previsões do que comprar e manter em estoque com base nesse fluxo.

Tendo em vistas esses aspectos, a empreendedora Dulcineia Cristina, dona da marca Dulci Modas, necessitou de um modelo de administração de negócio que atendesse todas as demandas que chegavam até ela. Dulcineia faz até duas viagens mensais a cidades grandes como Goiânia e São Paulo, onde são compradas grandes quantidades de roupas, para fazer a revenda aos seus clientes, fazendo que o controle de venda e de estoque se transformassem em um grande desafio.

Antes da implementação do novo sistema, o controle das vendas era realizado fisicamente através de um caderno. Isto resultava em um local de armazenamento vulnerável a acesso aos dados e também efeitos de degradação ou até mesmo eventuais contingências de seu uso, como por exemplo, ser roubado, rasgado ou molhado, o que acarretaria em uma indisponibilidade dos dados.

O sistema de gerenciamento de estoque e finanças teve como objetivo de facilitar e automatizar tarefas de armazenar informações de compras e vendas de produtos na área de vestuário e acessórios, sendo uma parte para visualizar e manter

os dados dos produtos, bem como seu preço de compra e de venda, e uma outra para registrar as vendas. Com este novo sistema aplicado, se obteve uma visualização mais transparente e acessível aos administradores da empresa, mostrando quais produtos são mais vendidos e quais dão um retorno financeiro maior.

2 REFERENCIAL TEÓRICO

Essa seção visa apresentar alguns conceitos relacionados a estoque, finanças e empreendedorismo. Neste sentido para melhor entendimento foi dividido em 3 sendo, 2.1 Estoque tratará de quais são as responsabilidades do setor de estoque, 2.2 Finanças, que irá discorrer sobre a importância dela para o negócio e 2.3 Empreendedorismo e sua definição.

2.1 ESTOQUE

Para Pozo, a responsabilidade do setor de estoque é o controle das disponibilidades e necessidades do processo produtivo. Assim, a principal função do setor de estoque é a maximização do uso dos recursos envolvidos na área de logística da empresa (POZO, 2010).

Sendo assim, a principal razão para que empresas mantêm grandes quantias de estoques é a possibilidade que se abre para a firma comprar e produzir em lotes econômicos, que nada mais é um modelo de quantidade de produção econômica que determina a quantidade que a empresa ou varejista deve solicitar para minimizar os custos totais de estoque assim equilibrando esse custo e o custo médio fixo de pedidos (POZO, 2010).

De acordo com Ballou (2006) estoques aparecem em numerosos pontos por todos os canais logísticos e de produção da empresa e são um conjunto de matérias-primas, insumos, componentes, produtos em processo e produtos já acabados. Assim, estoque não é apenas produtos armazenados em depósitos e galpões, mas deve-se levar em consideração os produtos expostos em prateleiras e vitrines para clientes.

Gestão de estoque são os atos de controlar a quantidade de produtos armazenados, decidir quando fazer uma nova compra, distribuir em lotes, classificar, organizar, identificar, etc. Administradores devem controlar suprimentos, embalagem, transporte, comercialização, controle de produção, podendo auxiliar cada fase do sistema empregado, com máxima eficiência e com mínimo de capital investido (DIAS, 2010).

Gerenciar estoque nada mais é do que fazer um planejamento total de como controlar os materiais dentro da empresa, tendo em base exatamente o que é mais necessário dentro da organização e suas áreas de estocagem, sendo o objetivo manter equilíbrio entre estoque e consumo, obtendo um controle mais bem aplicado, utilizando sistemas integrados de gestão.

Atualmente há vários métodos de controle de estoque, utilizando-os corretamente terá um grande impacto no caixa da empresa e em seus custos operacionais. Pozo (2010) aponta que mesmo existindo diversos de tipos e nomes de estoques, usualmente empresas possuem cinco tipos: almoxarifado de matérias-primas,

almoxarifado de materiais auxiliares, almoxarifado de manutenção, almoxarifado intermediário e almoxarifado de acabados. Cada um tendo um objetivo específico para fornecer o maior auxílio para os demais departamentos da organização.

2.2 FINANÇAS

Finanças, para GropPELLI e Nikbakht (2010), são aplicações que visam maximizar a riqueza ou valor total de um negócio seguindo uma série de princípios econômicos e financeiros. Com a empresa investindo em projetos e adquirindo ativos que possuem lucros altos e com riscos baixos, é possível conseguir essa maximização da riqueza (GROPPELLI, NIKBAKHT, 2010).

O termo finanças pode ser compreendido como a arte e a ciência de administrar o dinheiro. Segundo Gitman (2002, pág. 3) “finanças diz respeito ao processo, às instituições, aos mercados e aos instrumentos envolvidos na transferência de dinheiro entre pessoas, empresas e órgãos governamentais”. A compreensão desse termo dará ao indivíduo a capacidade de tomar melhores decisões financeiras pessoais (GITMAN, 2002).

A gestão dos negócios financeiros de todos os tipos de organizações é de responsabilidade dos administradores financeiros, essa gestão possui diversas tarefas como planejamento, concessão de crédito a clientes, avaliação de propostas que envolvam grandes desembolsos e captação de fundos para financiar as operações da empresa. Com as mudanças nos ambientes econômico, competitivo e regulamentador, a importância e complexidade das tarefas desses profissionais aumentaram. Hoje, esses administradores estão mais envolvidos com o desenvolvimento e implementação de estratégias para o crescimento da empresa, por isso grandes executivos com cargos altos tendem a vir da área financeira (GITMAN, 2002).

Dentre as modalidades de organização de empresas, a mais comum é a firma individual, empresa pertencente a uma única pessoa, que a opera em busca de lucro próprio. As firmas individuais são geralmente pequenos empreendimentos, como uma oficina de bicicletas, um serviço de personal trainer ou uma empresa de encanamentos. A maioria das empresas dessa modalidade se encontra nos setores de atacado, varejo, serviços e construção civil (GITMAN, 2002).

2.3 EMPREENDEDORISMO

Um empreendedor não são apenas os fundadores de empresas e criadores de novos negócios, mas também membros da geração de uma empresa familiar e gerentes-proprietários que compraram empresas já existentes, ser empreendedor é ser uma pessoa que inicia e dinamiza um negócio com base em uma ideia tomando para si riscos e responsabilidades para que o projeto pessoal dê certo (CHIAVENATO, 2012).

Criar um novo negócio ou reinventar um já existente refletem o empreendedorismo, por isso essa prática é comumente associado a riscos e incertezas, principalmente quando é algo totalmente novo ou até mesmo não exista nada igual no mercado, com a chegada da internet e de tecnologias avançadas, que permitiram

pessoas ordinárias a possuírem um PC (personal computer), que até algumas décadas atrás era usado apenas por grandes empresas e instituições com um alto custo para mantê-lo, veio uma grande leva de oportunidades de empreender em cenários que até o momento estavam fora da imaginação de muitos (CHIAVENATO, 2012).

Tem exemplos de negócios como a Dell que começou montando e vendendo computadores utilizando peças em desuso por um preço baixo e acessível ou a Amazon que começou suas atividades em uma garagem. Mas uma das dúvidas é se com negócios em pleno funcionamento e sucesso, novos empreendedores podem ingressar nesse mundo e obter sucesso, esse é o caso do Facebook que mesmo tendo um forte concorrente que era o MySpace, conseguiu se consolidar e se tornar a maior rede social do mundo (MAIORES E MELHORES, 2020).

Ao longo dos séculos o conceito que envolve empreendedor foi sendo modificado conforme os negócios iam evoluindo, na idade média esse termo era designado a indivíduos que gerenciam projetos de produção, já no século XVII este mesmo termo correspondia a profissionais que realizavam os acordos contratuais e assumiam riscos. Devido ao processo de industrialização, durante o século XVIII houve uma diferença e separação entre os termos empreendedor e capitalista. E entre os dois séculos seguintes, XIX e XX, empreendedores foram confundidos com gerentes e administradores, o que ocorre até os dias atuais (DORNELAS, 2005).

A definição de empreendedorismo ainda varia muito para cada autor, devido às concepções ainda não consolidadas. Entende-se disto como um processo relacionado com inovação, descobertas que estimulam a geração de riqueza através de novos negócios. Segundo Dornelas (2005), empreendedorismo é a transformação de ideias em oportunidades através do envolvimento de pessoas e processos que gera a criação de negócios de sucesso.

Assim entende-se de empreendedorismo como o resultado de iniciativas que são impulsionadas por reconhecer oportunidades que podem vir a gerar negócios lucrativos, criando um valor para a sociedade através do envolvimento de pessoas, processos e recursos.

3 METODOLOGIA

A metodologia utilizada como embasamento no desenvolvimento do projeto foi a Programação Extrema, ou simplesmente XP, sendo uma metodologia ágil por se ajustar bem em desenvolvimentos com requisitos vagos e em constante mudança. Seus princípios básicos são feedback rápido, presumir simplicidade, mudanças incrementais, abraçar mudanças e trabalho de qualidade.

Dentre todas as variáveis de controle de projetos como custo, tempo, qualidade e escopo, há um foco muito maior nesta última, a priorização de novas funcionalidades deve ocorrer o máximo possível, mesmo que signifique que outros requisitos menos valiosos sejam adiados ou cancelados, para ter um retorno maior possível para o negócio.

As tecnologias escolhidas foram Angular com o template Fuse para o painel, *frameworks* NestJS e TypeORM para o *backend*, o serviço AWS S3 para armazenar as imagens dos produtos, o serviço de banco de dados AWS RDS, o serviço de

versionamento de código GitHub com o fluxo organizacional GitFlow e o serviço de hospedagem Heroku implementando *deploy* contínuo com o GitHub.

Com isso em mente, as demandas mesmo sendo incompletas, foram acordadas e organizadas de modo que a cada entrega era agregado mais ao sistema, conforme o Quadro 1.

Quadro 1: Entregas e Demandas

Entregas	Demandas
Entrega 1	Criar arquitetura da API e do painel administrativo.
	Login e token para autenticar requisições.
	Cadastro de usuários.
Entrega 2	Cadastro de categorias dos produtos.
	Cadastro de produtos.
Entrega 3	Cadastro de clientes.
	Cadastro de vendas.
Entrega 4	Cadastro de fotos e mudar ordem de exibição.
Entrega 5	Gerar relatório de clientes.
	Gerar relatório de compras.
	Gerar relatório de vendas/faturamento.
Entrega 6	Gerar sugestões de compras.
	Gerar sugestões de vendas com promoções.

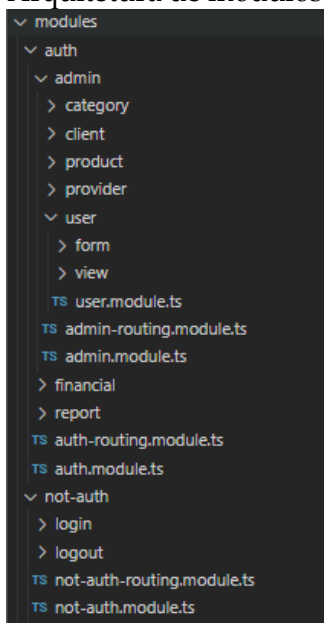
Fonte: Dados da pesquisa, 2020.

4 DESENVOLVIMENTO E RESULTADOS

Na fase inicial do estudo, foram levantadas as principais funcionalidades a serem desenvolvidas para que fosse mais amena a transição do uso do caderno para o novo sistema. Com isso em mente, foram levadas em conta as principais tecnologias e aplicações para que a implementação fosse o mais rápido possível, com uma arquitetura que possibilitasse a manutenibilidade.

Para o desenvolvimento do painel administrativo foi utilizado a plataforma Angular que, com uma arquitetura de módulos, possibilita um rápido desenvolvimento e manutenibilidade. Na Figura 1 é possível ver a disposição dos módulos na arquitetura que foi definida no painel, que são acessados quando o usuário estiver autenticado (módulo *auth*) e um outro que é usado quando ele não está autenticado (módulo *not-auth*).

Figura 1: Arquitetura de módulos do painel



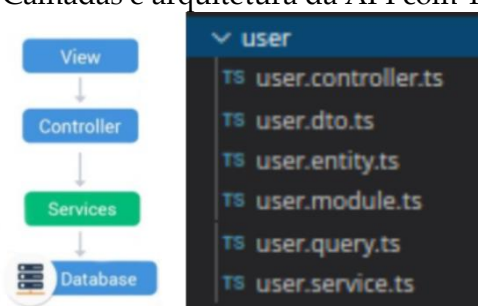
Fonte: Dados da pesquisa, 2020.

Na arquitetura apresentada na Figura 1, cada módulo é dependente de seu módulo pai, mas independente dos módulos “irmãos”, sendo assim é possível utilizar o *Lazy Loading*, um padrão de projeto que irá carregar os módulos solicitados somente se o usuário fizer a navegação até ele, logo, se o usuário navegar para a tela de usuários, o módulo de produtos não será carregado.

Seguindo a arquitetura do painel, foi desenvolvido uma API seguindo o padrão de arquitetura REST, que define um conjunto de características a serem usadas para a criação de serviços *web*.

Para realizar a construção da mesma foi utilizado o *framework* NestJS e o ORM TypeORM que possibilitam mapear a base de dados dentro dos arquivos com extensão “.entity”. Os *controllers* capturam a URI requisitada e direcionam para o *service*, realizando o seu método designado e aplicando as regras de negócio definidas utilizando o DTO para mapear as entidades e os arquivos “.query” para realizar uma melhor estrutura de filtros. É possível visualizar essa distribuição na Figura 2 a seguir.

Figura 2: Camadas e arquitetura da API com TypeORM

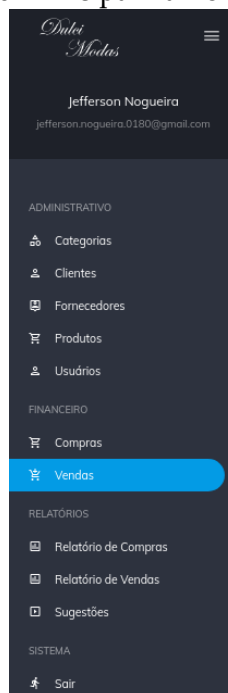


Fonte: Dados da pesquisa, 2020.

Após determinar e implementar a arquitetura inicial do painel e da API, deu-se início ao desenvolvimento dos métodos de *login* e autenticação para que todas as telas e rotas do sistema fossem seguras para que apenas requisições com *tokens* válidos, pudessem ser realizadas, caso contrário emite-se uma resposta de *status* de erro do cliente HTTP 401 *Unauthorized*.

Com a segurança do sistema implementada, todos os CRUD's (*Create, Read, Update e Delete*), acrônimo utilizado para definir as quatro operações básicas usadas em um banco de dados relacional, foram desenvolvidos. O menu com as principais funcionalidades é mostrado na Figura 3.

Figura 3: Menu Principal Painel Administrativo



Fonte: Dados da pesquisa, 2020.

No decorrer do projeto, houve mudanças no cadastro de compras e produtos, pois percebeu-se a necessidade de ter o cadastro dos fornecedores, já que em atacados de roupas, mais de um vendedor pode oferecer o mesmo produto. Assim, houve essa separação de cadastro de compra e produto, que no início tinham sido implementados em conjunto. Anteriormente a cada compra seria um novo registro de produto cadastrado, após a refatoração há um módulo voltado a todo o controle do produto e ao realizar a compra do mesmo, se informa a quantidade e quem o forneceu, como é mostrado na Figura 4.

Figura 4: Cadastro de Compra

Fonte: Dados da pesquisa, 2020.

Para o controle das vendas foi desenvolvido um CRUD que possibilita informar, além do produto e cliente que efetuou a compra, especificações como porcentagem de desconto, número de parcelas e se a venda foi dada como baixa, ou seja, se o cliente efetuou pagamento à vista. Isso permite que a venda de qualquer produto ocorra com mais flexibilidade, não se prendendo ao preço ou margem de lucro do produto em questão.

Além de todas as possibilidades acerca do valor da venda, foi desenvolvida uma funcionalidade que não foi especificada nas entregas, mas devido a sua importância para um completo entendimento, foi analisado e aprovado a implementação da mesma, o cadastro e exclusão de comentários sobre a venda, que é importante para o administrador controlar os recebimentos parciais, pois caso a venda seja efetuada em parcelas, poderá anotar todos os recebimentos dos valores até a sua baixa, conforme a Figura 5 apresenta. Posteriormente, um controle mais estruturado dos recebimentos parciais será implementado como trabalho futuro.

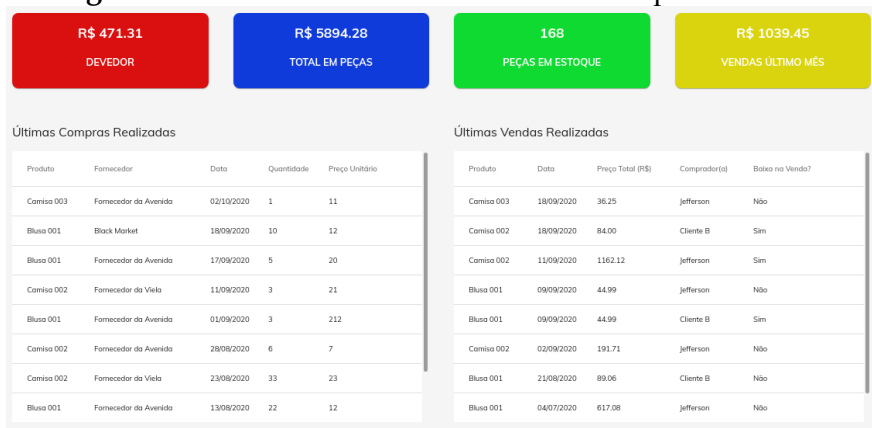
Figura 5: Cadastro de Venda

Fonte: Dados da pesquisa, 2020.

Além de todos os CRUD's desenvolvidos e mencionados, havia a necessidade de transformar os dados armazenados, tanto nas folhas quanto no sistema, em relatórios palpáveis e automáticos. Com isso em mente, foi solicitada a implementação de relatórios em formato de gráficos, para que fosse possível uma visualização mais simplificada do sistema como um todo, por exemplo a evolução das vendas com baixa ou não, o fluxo do número de peças em estoque, maiores devedores, últimas compras e vendas de produtos.

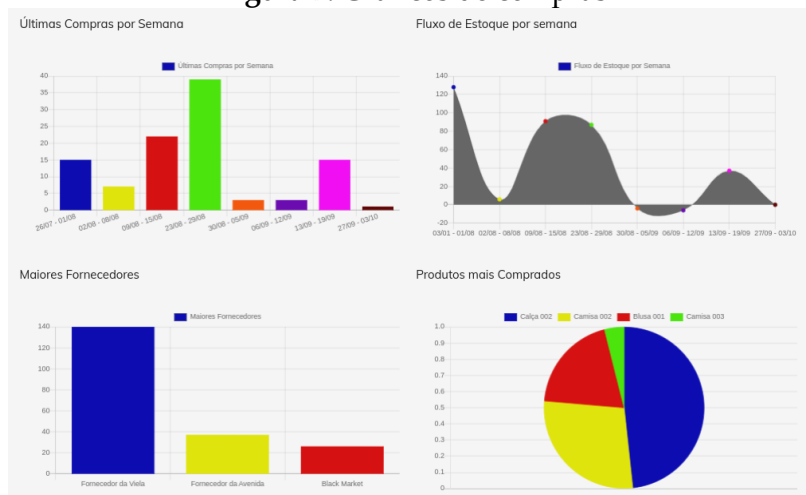
Para o desenvolvimento desses relatórios foi utilizado o componente de tabela comum do painel para a lista das últimas dez compras e últimas dez vendas. Foi criado também um componente de *card* que recebe como parâmetros um título, uma descrição e uma cor. Para os gráficos foi utilizada a biblioteca Chart.js que permite criar uma dinamicidade na apresentação dos dados, com tipos de gráficos diferentes, que permitem fazer uma componentização para reutilização de código. Os *cards* e as tabelas são apresentados na Figura 6 e os gráficos na Figura 7.

Figura 6: Tela inicial com *cards* e listas de compras e vendas



Fonte: Dados da pesquisa, 2020.

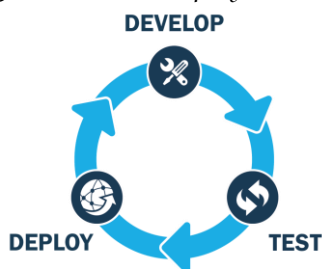
Figura 7: Gráficos de compras



Fonte: Dados da pesquisa, 2020.

Durante todo o desenvolvimento das funcionalidades, foi utilizado o repositório de código GitHub em conjunto com o padrão GitFlow, para realizar o versionamento das *branches* e o *deploy* contínuo integrando com o serviço Heroku, assim possibilitando ao cliente dar um *feedback* mais rápido e coeso, já que, mesmo o ambiente não sendo de homologação, os requisitos e prioridades foram se alterando durante o desenvolvimento, resultando com que o cliente testasse e reportasse com mais rapidez. Esse fluxo é mostrado na Figura 8.

Figura 8: Fluxo *deploy* contínuo



Fonte: <https://www.fellow-consulting.com/easy-continuous-integration/>.

5 CONCLUSÃO

O estudo e desenvolvimento apresentado descreveram a implementação de um sistema de estoque e finanças, que teve como principal objetivo suprir e substituir o antigo método de controle utilizado pela empresa Dulci Modas, baseado em cadernos e notinhas, que por sua vez eram facilmente perdidas, gerando um desgaste de tempo e esforço para localizar vendas e clientes.

Os objetivos propostos neste artigo foram alcançados, pois todas as entregas planejadas inicialmente foram feitas. Algumas funcionalidades importantes para que seja considerado um sistema de finanças e estoque completo não foram implementadas neste primeiro momento, mas já estão em *backlog* para serem desenvolvidas e dar uma maior abrangência para o negócio, como por exemplo o cadastro de despesas geradas durante viagens de compras, assim como o registro de pagamentos parcelados, e com isso ter total manutenção do dinheiro em caixa, podendo até lançar os gastos com modelos de roupas, que são utilizados para tirar fotos e postar nas redes sociais.

O mercado de moda e roupas cresce bastante todos os anos, mas mesmo assim a procura por marcas menos conhecidas é enorme, com isso pequenos empreendedores podem emergir tendo altos lucros fazendo essas revendas, mas é bastante comum que esses novos administradores de seu próprio negócio se percam por falta de controle e organização. Por isso, um sistema que os auxilie é tão importante, gerando informações confiáveis e possibilitando aos gestores focar em como fazer o negócio expandir.

REFERÊNCIAS

BALLOU, R. H. **Gerenciamento da cadeia de suprimentos**: planejamento, organização e logística empresarial. Tradução de Elias Pereira. 5. ed. Porto Alegre: Bookman, 2006.

CHIAVENATO, I. **Empreendedorismo dando asas ao espírito empreendedor**. 4. ed. São Paulo: Editora Manoele, 2012.

CLIENTESA. **A retomada do varejo de vestuário**. Disponível em: <https://www.clientesa.com.br/estatisticas/67713/a-retomada-do-varejo-de-vestuario>. Acesso em: 24 março 2020

DIAS, M. A. P. **Administração de materiais: uma abordagem logística**. 5. ed. São Paulo: Atlas, 2010.

DORNELAS, J. C. A. **Transformando idéias em negócios**. 2. ed. Rio de Janeiro: Elsevier, 2005.

GITMAN, Lawrence J. **Princípios de administração financeira**. 7. ed. São Paulo: Editora Harbra, 2002. 3p.

GROPPELLI, A. A.; NIKBAKHT, E. **Administração financeira**. 3. ed. São Paulo: Editora Saraiva, 2010.

HACO. **Conheça os eventos de moda que acontecem no Brasil**. Disponível em: <https://www.haco.com.br/blog/eventos-de-moda-que-acontecem-no-brasil/>. Acesso em: 10 março 2020.

MAIORES E MELHORES. **Essas são as 20 maiores redes sociais do mundo (2020)**. Disponível em: <https://www.maioresemelhores.com/maiores-redes-sociais-do-mundo/>. Acesso em: 03 maio 2020.

NOMI, Thatiana. **Como os robôs estão transformando o trabalho nos centros de distribuição do mundo**. Disponível em: <https://www.ilos.com.br/web/como-os-robos-estao-transformando-o-trabalho-nos-centros-de-distribuicao-do-mundo/>. Acesso em: 25 março 2020.

POZO, Hamilton. **Administração de recursos materiais e patrimoniais**. 6. ed. São Paulo: Atlas, 2010. 25p.

UTILIZAÇÃO DE *CHATBOT* NO AUXÍLIO AO PROCESSO DE INCLUSÃO DIGITAL DE PCDS

Wellington Coêlho de Araújo¹

Henaldo Barros Moraes²

RESUMO: O presente artigo tem por objetivo apresentar um *software* que, em condições de uso, poderá auxiliar as pessoas com deficiências físicas. Trata-se de um sistema inteligente capaz de acessar por meio de um comando de voz emitido pelo usuário, as várias funcionalidades de um computador. Através de palavras chaves, o *software* poderá ser usado para as mais diversas finalidades, desde ouvir as notícias do dia a reproduzir um vídeo no *YouTube*, pesquisar sobre algo ou alguém no *Wikipédia*, enviar um *tweet* para um amigo, um *e-mail*, dentre outras atividades.

PALAVRAS-CHAVE: Acessibilidade, deficiência física, aprendizado-de-máquina, inteligência-artificial, reconhecimento-de-voz.

ABSTRACT: In order to assist people with physical disabilities, a smart system was developed, capable of accessing the Internet and its various pages with a voice command issued by the user. Through key words, the software can be used for a wide range of purposes, from auscultating the weather forecast, searching for something or someone on Wikipedia, sending a tweet to a friend, and other activities.

KEYWORDS: Accessibility, physical disability, machine-learning, artificial intelligence, speech recognition.

1 INTRODUÇÃO

O presente artigo irá apresentar o GLADIS, um software desenvolvido para auxílio a pessoas que portam alguma deficiência a terem acesso às funcionalidades de um computador e a internet como um todo, sendo o mesmo podendo ser utilizado totalmente por comandos de voz, proporcionando uma maior inclusão destas pessoas ao mundo digital.

No século XXI, as pessoas querem estar conectadas a todo instante, compartilhando seus momentos nas redes sociais, conhecendo pessoas e realizando suas tarefas do dia-a-dia de forma rápida e prática. De acordo com Otoni (2015), autor da pesquisa Futuro Digital em Foco Brasil 2015, os brasileiros são líderes em tempo gasto nas redes sociais. A média do Brasil é 60% maior do que a do resto do planeta.

Passar o tempo conectado à internet pode não parecer algo tão grandioso para algumas pessoas. Para outras, o fato de acessar uma rede social, assistir um vídeo ou

¹ Estudante de Graduação do Curso de Sistemas de Informação do UNIPAM. E-mail: wellingtoncoelho@unipam.edu.br.

² Professor de Graduação do Curso de Sistemas de Informação do UNIPAM. E-mail: henaldobarros@unipam.edu.br.

¹Aluno de Sistemas de Informações, UNIPAM, raphaelcustodio94@gmail.com

² Mestre em Redes de Computadores, UFU, henaldobarros@gmail.com

até mesmo ler publicações e notícias é algo incrível de se fazer. Pessoas com determinadas limitações físicas têm restrição quanto ao acesso à *internet*, necessitando da ajuda de alguém ou de recursos para conseguirem se conectar. Em grande parte do tempo, estes recursos não são acessíveis a todos.

No Brasil, quase 24% da população brasileira é composta por pessoas que possuem algum tipo de deficiência. De acordo com o Censo do Instituto Brasileiro de Geografia e Estatística (IBGE) realizado em 2010, o Brasil possui em torno de 45 milhões de Pessoas com Deficiência (PCDs). Dependendo do grau da deficiência, estas são impossibilitadas de realizarem tarefas do cotidiano de forma independente.

Neste contexto, como objetivo, identificou-se a necessidade de um sistema de apoio para auxiliar deficientes físicos a realizarem tarefas das mais diversas complexidades, seja na *internet*, em suas redes sociais, assim como a utilização dos mais variados programas que um computador pode disponibilizar. Como por exemplo de tais atividades, podem ser citados redigir um texto no *Word*, executar algum arquivo de mídia, na internet, acessar a *Wikipedia*, acessar notícias, assistir ou ouvir algo no *Youtube*, dentre outras. Desta forma, essas pessoas se veem habilitadas para aproveitar todo o conteúdo, entretenimento e conhecimento oferecidos pela rede global de computadores. Com o GLADIS instalado na máquina do usuário, assim que inicializado, por meio do comando de voz, o usuário terá acesso ao mundo disponibilizado pela internet, adquirindo conhecimento pedagógico, fazendo novos amigos, novas experiências, pautando novas discussões e conhecendo novos lugares.

É de grande valia que se faça a integração de tecnologias no processo de acessibilidade a inclusão digital de todos os cidadãos, o avanço tecnológico exerce uma pressão cada vez maior sobre a sociedade, onde quem tem um maior domínio sobre elas acaba por se destacar mais, tanto no mercado de trabalho concorrido dos dias de hoje, quanto nas interações sociais em tempos de pandemia. Técnicas de IA e automatização de tarefas podem ser o grande trunfo nestes casos.

Considerando os objetivos específicos deste projeto, são os seguintes objetivos gerais:

- Prover uma interação legítima de uma pessoa portadora de deficiência física com o ambiente da *internet*.
- Facilitar a realização de tarefas corriqueiras do dia-a-dia (fazer pesquisas, visitar páginas *web*, ver notícias do mundo todo, enviar e ler mensagens, etc.).
- Promover a inclusão digital de pessoas com algum tipo de comprometimento na condição física, e que por isso, tais pessoas não conseguem acessar a rede sem ajuda de terceiros;
- Garantir que o usuário tenha total acesso a todas as ferramentas disponíveis em sua máquina.
- Garantir a automatização de tarefas por comandos de voz.

Atualmente, percebe-se que as pessoas estão cada vez mais conectadas entre si e observa-se que este número tende a crescer exponencialmente a cada ano.

Em 2020, com o atual cenário que se instaurou em decorrência da pandemia do novo Corona-Vírus, as empresas estão atentas, de olho no futuro do trabalho em estilo *HomeOffice*, quando a maioria das tarefas do dia-a-dia de um escritório, por exemplo, podem ser totalmente feitas do conforto da casa do funcionário, evitando o

contato direto com outras pessoas no mesmo ambiente. Ao se evitar sair de casa e aumentar o fluxo de pessoas transitando pelas cidades, estamos todos preservando a vida da população. O sistema GLADIS será inserido neste contexto como um auxiliar às pessoas que estão passando por esse momento, tornando o trabalho e o uso do computador menos monótonos e facilitando que tarefas simples possam ser mais inclusivas a todos. Não se deve pensar que esse tipo de acesso é exclusividade das pessoas com deficiência visto que todos têm o direito de desfrutar dessa ferramenta, tornando tudo mais perto é mais possível de ser alcançado.

Desta maneira, este sistema possibilitará que pessoas que adquiriram algum tipo de deficiência física em algum momento da vida, ou que já nasceram com tal condição, tenham a mesma experiência que as demais pessoas no acesso às mais diversas informações via *internet*, como também a possibilidade de ter acesso total a quaisquer aplicações presentes em sua máquina, como por exemplo, editores de texto, reprodutores de mídias, caixa de *e-mail*, etc.

2 REFERENCIAL TEÓRICO

Este item aborda os conceitos e temas que fazem parte do escopo de desenvolvimento do artigo.

2.1 INTELIGÊNCIA ARTIFICIAL

Inteligência artificial, conhecida pela sigla IA, é a inteligência similar à humana exibida por *software*. Também é um campo de estudo acadêmico. Os principais pesquisadores e livros didáticos definem o campo de IA como "o estudo e projeto de agentes inteligentes", onde um agente inteligente é um sistema que percebe seu ambiente e toma atitudes que maximizam suas chances de sucesso. John McCarthy, quem cunhou o termo em 1956 (numa conferência de especialistas celebrada em *Darmouth Colege Gubern, Román: O Eros Eletrónico*), a define como a ciência e engenharia de produzir máquinas inteligentes.

Segundo Elaine Rich (1988), é o estudo de como fazer os computadores realizarem tarefas de forma semelhante à maneira humana.

Um sistema IA não é capaz somente de armazenamento e manipulação de dados, mas também da aquisição, representação, e manipulação de conhecimento. Esta manipulação inclui a capacidade de deduzir ou inferir dois novos conhecimentos - novas relações sobre fatos e conceitos - a partir do conhecimento existente e utilizar métodos de representação e manipulação para resolver problemas complexos que são frequentemente não quantitativos por natureza. Uma das ideias mais úteis que emergiram das pesquisas em IA, é que os fatos e as regras - conhecimento declarativo - podem ser representados separadamente dos algoritmos de decisão - conhecimento procedimental (SCHUTZER, 1987). Atualmente, com a capacidade de processamento dos computadores aumentando, percebe-se que a busca de informações através dos sistemas computacionais vem se tornando cada vez mais eficaz e acredita-se que a IA é uma das tecnologias que abrirão portas para os *softwares* do futuro. Com todo esse

avanço da atualidade, existem muitos estudos de ferramentas que aplicam técnicas da inteligência artificial no processo de ensino/aprendizagem.

É uma área de pesquisa da computação dedicada a buscar métodos ou dispositivos computacionais que possuam ou multipliquem a capacidade racional do ser humano de resolver problemas, pensar ou, de forma ampla, ser inteligente. Também pode ser definida como o ramo da ciência da computação que se ocupa do comportamento inteligente, ou ainda, o estudo de como fazer os computadores realizarem coisas que, atualmente, os humanos fazem melhor.

2.2 RECONHECIMENTO DE FALA

Tecnologias de reconhecimento da fala, também conhecida como Reconhecimento de Voz, permitem que computadores interpretem a fala humana, por exemplo, para transcrição ou como método de comando por voz. Tais sistemas podem ser classificados por requererem, ou não, que o usuário treine o sistema a reconhecer seus padrões particulares de fala, por ter a habilidade de reconhecer fala contínua ou por requerer que o usuário fale pausadamente, e pelo tamanho do vocabulário que é capaz de reconhecer (pequeno, da ordem de dezenas a centenas de palavras, ou grande, com milhares de palavras).

2.3 MACHINE LEARNING

Segundo Coppin (2010), o aprendizado está diretamente ligado com a inteligência, pois realmente se um sistema é capaz de aprender a exercer determinada tarefa merece ser chamado de inteligente. Um processo de aprendizagem inclui a aquisição de novas formas de conhecimento: o desenvolvimento motor e a habilidade cognitiva (através de instruções ou prática), a organização do novo conhecimento (representações efetivas) e as descobertas de novos fatos e teorias através da observação e experimentação.

Desde o início da era dos computadores, têm sido realizadas pesquisas para implantar algumas destas capacidades em computadores. Resolver este problema tem sido o maior desafio para os pesquisadores de inteligência artificial (IA), a inteligência artificial é um ramo da ciência da computação, onde através de mecanismos e *softwares* pode-se obter resultados cognitivos similares ao humano. O estudo e a modelagem de processos de aprendizagem em computadores e suas múltiplas manifestações constituem o objetivo principal do estudo de aprendizado de máquinas (SANTOS, 2005, p 10).

Como sugere Coppin (2010), uma forma bastante avançada de aprendizado de máquina são as redes neurais que têm semelhança com o funcionamento do cérebro humano, sendo uma grande rede de neurônios.

Essa rede é organizada geralmente em duas camadas. A primeira recebe as informações a serem classificadas, usa aprendizado supervisionado por modificarem a forma das conexões de acordo com o que é informado e por último ativam os neurônios de saída. É uma forma bastante complexa, mas tem muita utilidade por ser bastante precisa e dificilmente acontecer erros, que em outros ambientes de

aprendizagem são comuns. Dentro das redes neurais existe a forma de aprendizado não supervisionado que não precisa de nenhum tipo de classificação. Isso acontece, por exemplo, ao ser feita uma pesquisa na *Internet*, que traz vários resultados ao interpretar a informação sem nenhum tipo de classificação definida pelo usuário e, a partir disso, tem-se várias usabilidades de sua técnica.

Machine learning é um método de análises de dados que é utilizado em diversos processos *on-line*, pois com ele é possível coletar, analisar e categorizar os dados gerados ou inseridos. No ramo educacional, também é possível utilizar dessa técnica, como por exemplo, em cursos *on-line* ou EAD, quando é possível medir a eficácia, a qualidade e os métodos propostos, além de ajudar os professores a expandir consideravelmente o conhecimento dos alunos.

2.4 WEB SCRAPING

Web Scraping é uma técnica de extração de dados utilizada para coletar dados de sites. Por meio de processos automatizados, implementados, usando um rastreador *bot*, esse tipo de “raspagem” de informações é uma forma de realizar cópias de dados em que informações específicas são coletadas e copiadas da *web*, tipicamente em um banco de dados ou planilha local central, para posterior recuperação ou análise”. (BLOG BRASIL WESTCON, 2020).

2.5 AUTOMAÇÃO DE GUI

O teste *Graphical User Interface* (GUI) é realizado para verificar os recursos visíveis a um usuário, como menus, botões, ícones, caixas de texto, listas, caixas de diálogo, etc. Também garante que os elementos de aparência, como fontes e imagens, estejam em conformidade com as especificações de design. O teste de GUI ocorre no nível de teste do sistema.

Utilizando-se da ferramenta *Pyautogui* que é um módulo de automação de GUI para *Python2* e *Python3* que fornece métodos para controlar mouse e teclado. Esse módulo pode ser usado para criar *bots* para automatizar tarefas repetitivas, como executar *cliques* em qualquer parte da tela, ter o controle do *keyboard* do usuário para digitar em campos de texto (inserindo nomes de usuário e/ou senhas, para acesso a páginas privadas, como redes sociais).

2.6 SISTEMAS ESPECIALISTAS

“Sistemas especialistas são programas que utilizam conhecimento e procedimentos inferenciais para resolver problemas que normalmente requerem muita perícia humana” (WEBBER *et. al.*, 2009).

Um sistema especialista manipula o conhecimento, faz inferências às informações fornecidas pelo usuário e busca soluções aplicando esse mesmo conhecimento. E do ponto de vista educacional, a maioria dos sistemas especialistas têm pouca utilidade direta, porque não foram projetados para ensinar. Entretanto, a estrutura do sistema especialista serve perfeitamente para ser adaptada para a

construção de sistemas tutoriais, proporcionando um grande potencial para a criação de ambientes educacionais. Portanto, um sistema tutorial não necessita somente do conhecimento de seu domínio, mas também da perspectiva sobre este conhecimento que permita transmiti-lo ao estudante adequadamente.

2.7 CHATBOT

Um *chatbot* é um programa de computador que tenta simular um ser humano na conversação com as pessoas. O objetivo é responder as perguntas de tal forma que as pessoas tenham a impressão de estar conversando com outra pessoa e não com um programa de computador (TEIXEIRA; MENEZES, 2003).

Os *bots* utilizam o Processamento de Linguagem Natural (PNL) e a técnica de *Machine Learning*, que consiste em chegar mais próximo de um ser humano e, a partir disso, atender ao usuário da maneira mais dinâmica e assertiva possível.

Quanto aos *bots*, existem diferentes tipos sendo eles o de motivação que têm como função animar e motivar os alunos; o de revisão, que auxilia os alunos a melhorar seus pontos fracos e compreender matérias, e o de avisos, que foca e prepara os alunos para os testes e provas. Existem outros como o social, que verifica as pessoas que gostariam de montar um grupo de estudo e monta os melhores horários, e também o que permite encontrar pessoas próximas que possam explicar determinada dúvida.

Já a utilização de robôs nas conversações educacionais pode ser de extrema vantagem, sendo que os mesmos são treinados para tirar dúvidas e direcionar o interlocutor para o caminho mais apropriado de acordo com suas necessidades. Existe também o fato de que o interlocutor pode se sentir mais à vontade ao realizar as indagações ao *chatbot*.

É perceptível o quanto e como os *bots* vêm crescendo, e como podem ser implantados tranquilamente na área educacional sendo que existem diversos atualmente em implantação, tornando essa realidade cada vez mais próxima das pessoas.

3 METODOLOGIA

A presente pesquisa foi desenvolvida tendo por base o estudo das técnicas de Inteligência Artificial aplicadas em sistemas voltados para o processo de automação e reconhecimento de fala. Foi realizada, assim, uma revisão da literatura, com ênfase na identificação de soluções computacionais de IA desenvolvidas para a área da acessibilidade. Neste sentido, foram consultados livros, artigos e trabalhos acadêmicos desenvolvidos dentro desta problemática.

Após o estudo, foi feita uma análise com o objetivo de identificar os métodos e aplicações computacionais que obtiveram sucesso no uso pelo usuário e que realmente contribuíram para promover uma melhor interação do usuário com o computador e as demais funcionalidades *web*.

E, em uma última etapa, foi desenvolvido o protótipo de uma aplicação para auxiliar no processo de utilização de todas as funcionalidades disponíveis pela rede, quando foi utilizado o serviço de inteligência disponibilizado pelo *Google Cloud*, e

desenvolvido um *chatbot* na linguagem *Python*. Em última instância, foi realizado o treinamento do *bot* desenvolvido.

Foram utilizadas para o desenvolvimento do projeto algumas ferramentas como *Python* para linguagem de programação orientada a objetos, *IDL Python* responsável pela plataforma de desenvolvimento, *wikipedia API* e *google search API* sendo os bancos de dados de pesquisa, *pyautogui* na automação para manipulação de periféricos de E/S, *machine learning* como aprendizado da máquina, *SQLite* que é a biblioteca que implementa o banco de dados embutido, *pyttsx3* para conversão da fala do software para PT-BR, *pyttsx3*: conversão da fala do *software* para PT-BR, *speechrecognition* para reconhecimento inteligente de fala, *chatterBot* para reconhecimento gramatical e o *beautifulSoup*, que é o pacote *Python* para analisar documentos HTML e XML.

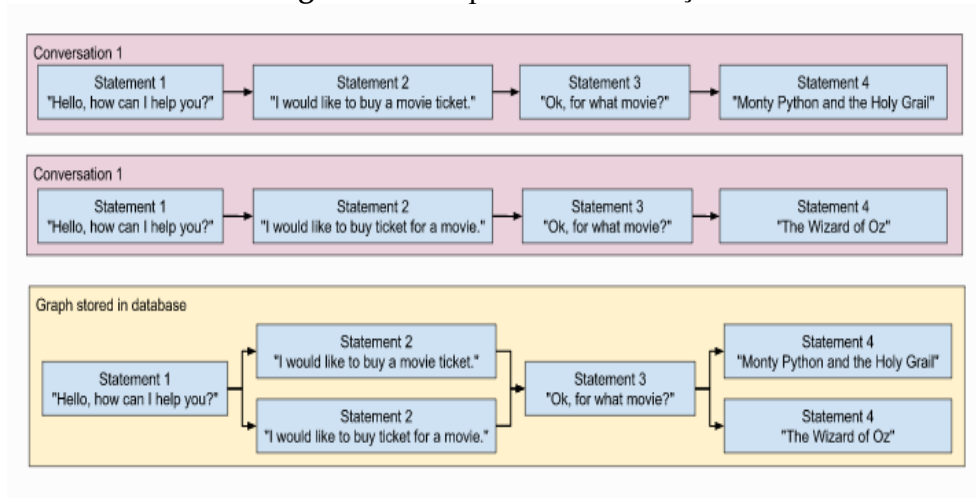
4 DESENVOLVIMENTO E RESULTADOS

Inicialmente, foi feito um levantamento de todas as funcionalidades do sistema e para ajudar a definir os requisitos foi necessário realizar um estudo coletivo sobre as práticas de *Machine Learning* a fim de colher todos os requisitos que seriam necessários para o desenvolvimento do sistema.

A linguagem de programação *Python* é uma excelente escolha para a realização de automação de tarefas, escolha essa direcionada pelos seguintes fatores: Simplicidade de sintaxe, Disponibilidade em várias plataformas, Disponibilidade de bibliotecas e Uso futuro, pois *Python* é uma linguagem de uso muito abrangente.

A *framework ChatterBot* inclui ferramentas que ajudam a deixar o processo de treinamento de uma instância de *bot* de bate-papo mais dinâmica e simples. Na Figura 1, pode se ver que o processo de treinamento envolve o carregamento de um diálogo de exemplo no banco de dados do *bot*. Isso cria uma estrutura de dados que representa os conjuntos de declarações e respostas conhecidas. Quando um *training* de *bot* recebe um conjunto de dados, ele cria as entradas necessárias no banco de dados de conhecimento do *bot* para que as entradas e respostas das declarações sejam representadas corretamente.

Figura 1: Exemplo de Conversação



Fonte: *chatterbot docs*, 2020.

Na Figura 2, é possível ver o processo de ativação do motor e “Chico Xavier” que será a fonte a ser pesquisada. Ao se observar, a título de exemplo, supondo que o usuário queira pesquisar sobre Chico Xavier, ele dirá “Quem foi Chico Xavier”, “Quem foi” é a palavra-chave.

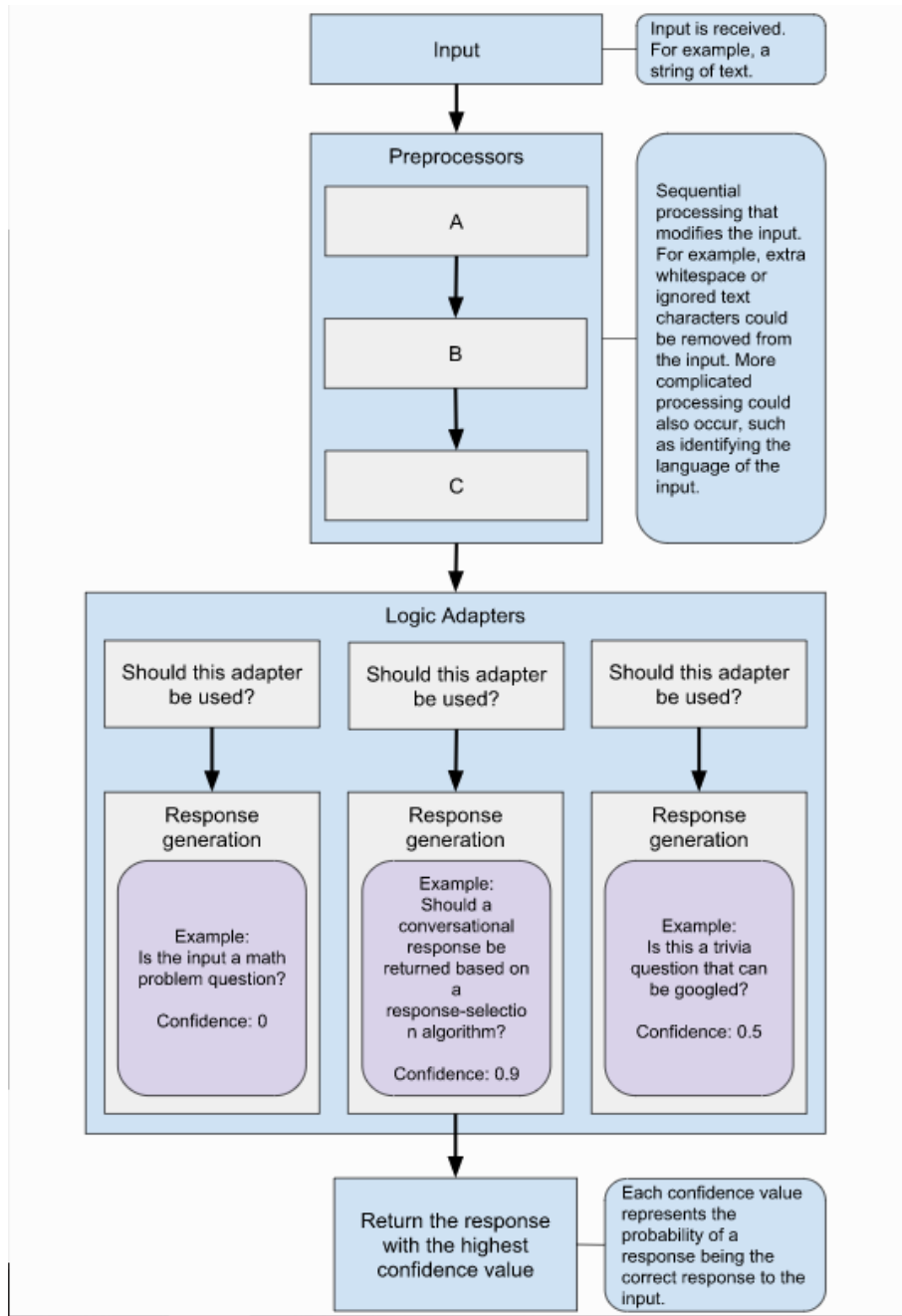
Figura 2: Exemplo de pesquisa

```
def wikipedia_search():
    print('Hawk: Qual tema deseja pesquisar?')
    pesquisa = (u"Qual tema deseja pesquisar?")
    tts.Speak(pesquisa)
    speech = recVoz(r)
    print('Você: ', speech)
    busca = speech.replace(" ", '_')
    url = "https://pt.wikipedia.org/wiki/"+busca
    page = rq.get(url=url)
    soup = bs(page.content, 'html.parser')
    conteudo = soup.find(id="mw-content-text")
    paragrafo = conteudo.find('p')
    texto = paragrafo.get_text()
    print('Hawk: '+texto)
    pesquisa = (u""+texto)
    tts.Speak(pesquisa)
def write(texto, busca, encoding='utf-8', errors='strict'):
    data = str(texto).encode(encoding, errors=errors)
    try:
        with open(busca, 'wb') as f:
            f.write(data)
    except IOError as e:
        if e.errno == 2:
            os.makedirs(os.path.dirname(busca), exist_ok=True)
            return write(texto, busca, encoding, errors)
        else:
            raise e
main()
```

Fonte: Dados da pesquisa, 2020.

Todas as respostas dadas pelo bot na parte que se diz respeito à conversação leva com base os adaptadores lógicos. Representado na Figura 3, o adaptador lógico que o bot usa pode ser especificado definindo o parâmetro `logic_adapters` para o caminho de importação do adaptador lógico que o usuário deseja usar. O bot retornará a resposta com o maior valor de confiança calculado. Os Preprocessors do ChatterBot são funções que modificam a instrução de entrada que um bot recebe antes que a instrução seja processada pelo adaptador lógico.

Figura 3: Adaptador lógico



Fonte: chatterbot docs, 2020.

Na Figura 4, é mostrado o trecho de código do GLADIS responsável pelo treinamento do bot e a garantia de 0.90 da resposta ser a mais assertiva.

Figura 4: Código de treinamento

```

logic_adapters=[
    {
        'import_path': 'chatterbot.logic.BestMatch',
        'default_response': 'Me desculpe, não compreendi!',
        'threshold': 0.90
    }
]

trainer = ChatterBotCorpusTrainer(bot)
trainer.train(["chatterbot.corpus.portuguese.greetings",
              'chatterbot.corpus.portuguese.compliment',
              'chatterbot.corpus.portuguese.conversations',
              'chatterbot.corpus.portuguese.games',
              'chatterbot.corpus.portuguese.linguistic_knowledge',
              'chatterbot.corpus.portuguese.money',
              'chatterbot.corpus.portuguese.proverbs',
              'chatterbot.corpus.portuguese.suggestions',
              'chatterbot.corpus.portuguese.trivia',
              'chatterbot.corpus.portuguese.unilab'
])

bot= ChatBot('Bot')
trainer = ChatterBotCorpusTrainer(bot)
corpus_path = 'C:/Users/Usuario/AppData/Local/Programs/Python/Python37/Lib/site-packages/chatterbot_corpus/data/portuguese/dic.json'

```

Fonte: Dados da pesquisa, 2020.

Trabalhando com a ferramenta *Chatterbot* e *Google API*, o GLADIS é capaz de identificar uma frase dita pelo usuário e encontrar a resposta mais assertiva através de seus algoritmos de comparação. Inicialmente o software será criado com o conhecimento de uma gama de palavras, dentre elas, frases compostas, cumprimentos, respostas a perguntas cotidianas, conhecimentos sobre games, dinheiro, provérbios e também todas as palavras do dicionário de língua portuguesa.

A partir disso, ele irá armazenar cada seção de conversas feitas com o usuário para aprendê-las e usar em futuras conversas, aumentando sempre seu aprendizado e suas respostas.

Trabalhando em conjunto com a ferramenta *BeautifulSoup*, que faz a raspagem de documentos HTML e XML em páginas *web*, para assim conseguir interpretar os textos contidos, retornando o mesmo audível ao usuário, seria o caso de uma busca, por exemplo, de notícias, previsões do tempo e etc.

Na Figura 5 é possível ver o funcionamento do *BeautifulSoup* (bs4). Neste trecho, o bs4 executa a extração de dados de arquivos HTML e XML de um site de piadas, ele é acionado toda vez que o usuário diz: eu gostaria de ouvir uma piada, por exemplo, aplicando o *random* para alternar entre as várias piadas que foram “parseadas” no site. Ela funciona com o interpretador (parser) de sua preferência a fim de deixar mais intuitivas as maneiras de navegar, modificar e buscar uma árvore de análise (*parse tree*).

Figura 5: Código utilizando o bs4

```

def piadas():
    aleatorio = random.randint(1, 7)
    url = 'https://www.piadas.com.br'
    page = rq.get(url=url, timeout=2)
    soup = bs(page.content, 'html.parser')
    if aleatorio == 1:
        conteudo = soup.find(class_="views-row views-row-2 views-row-even")
    elif aleatorio == 2:
        conteudo = soup.find(class_="views-row views-row-3 views-row-even")
    elif aleatorio == 3:
        conteudo = soup.find(class_="views-row views-row-4 views-row-even")
    elif aleatorio == 4:
        conteudo = soup.find(class_="views-row views-row-5 views-row-even")
    elif aleatorio == 5:
        conteudo = soup.find(class_="views-row views-row-6 views-row-even")
    elif aleatorio == 6:
        conteudo = soup.find(class_="views-row views-row-7 views-row-even")
    elif aleatorio == 7:
        conteudo = soup.find(class_="views-row views-row-8 views-row-even")
    piada = conteudo.get_text()
    print('Gladis: ...')
    texto = (u""+piada)
    tts.Speak(texto)
    main()

```

Fonte: Dados da pesquisa, 2020.

Com o *Pyautogui* o software é capaz de tornar o navegador uma espécie de marionete, podendo ser realizados *clicks* com o mouse e digitações em campos de texto, sendo útil para o preenchimento de campos de nome de usuário e senhas em redes sociais, por exemplo.

A ferramenta *SpeechRecognition* e *Pyttsx* possibilita ao software transformar fala em texto e texto em fala, em português, exibindo o que foi dito por ambos na tela. Representado na Figura 6 (Caso o usuário escolha por esta opção).

Figura 6: Reconhecimnto de voz

```

def recVoz(r):
    try:
        with sr.Microphone() as source:
            r.adjust_for_ambient_noise(source)
            audio = r.listen(source)
            speech = r.recognize_google_cloud(audio, language='pt-BR')
            return speech
    except sr.UnknownValueError:
        print('Erro de reconhecimento de fala')
        time.sleep(2)
    main()

```

Fonte: Dados da pesquisa, 2020.

4 CONCLUSÃO

As pesquisas realizadas para fundamentação tiveram um caráter muito importante e impactante para direcionar o projeto, possibilitando o acesso à realidade de muitas pessoas que para alguns pode passar despercebido. Sendo algo tão comum nos dias atuais a rede de internet é a principal responsável pela comunicação social, aprendizado e diversão, é o que liga toda a população mundial, sendo que alguém que não tenha essa possibilidade se sinta excluído desse universo.

Em meio a esse impedimento foi pensado em uma intervenção para incluir qualquer indivíduo ao mundo virtual por meio particular, permitindo a essas pessoas o mínimo de independência. O uso da tecnologia por parte do portador de necessidades especiais possibilita a subversão de obstáculos, pois a tecnologia associada a uma prática pedagógica pode ser uma excelente ferramenta no processo de aprendizagem, além de apoiar seu desenvolvimento social.

A contribuição deste estudo para a comunidade é relevante no sentido de cooperar para que pessoas com deficiência tenham acesso a qualquer informação de aprendizado, lazer virtual e que possam também direcionar possíveis progressos para a população em geral. O sistema GLADIS melhora consideravelmente a utilização do computador e da internet por pessoas portadoras de deficiência física, já que devido às suas limitações, as mesmas não possuem, em boa parte dos casos, nenhuma integração com o ambiente digital. Dessa forma, elas poderão interagir com mais pessoas através de redes sociais, acessar informações em sites de notícias, acessar vídeos do Youtube, escrever um texto, etc. Nunca foi tão fácil realizar essas tarefas visto que tudo é feito através de comandos de voz pré-programados. Desse modo, o projeto apresentado contribuirá no avanço da integração de questões de acessibilidade virtual.

REFERÊNCIAS

ANDERSON, David J.; CARMICHAEL, Andy. **Essential kanban condensed**. Seattle, Washington, 28 de jul. de 2016. Disponível em: <http://leankanban.com/wp-content/uploads/2016/06/Essential-Kanban-Condensed.pdf>. Acesso em: 01 julho 2018.

ARAJABAT. O bom e o mau da ferramenta de automação de teste GUI Ranorex. **Blog Agatetepe**. 2019. Disponível em: <https://www.agatetepe.com.br/o-bom-e-o-mau-da-ferramenta-de-automacao-de-teste-gui-ranorex>. Acesso em: 14 maio 2020.

CAMARGO, Robson. **Gerenciamento de Projetos: Scrum: Conheça regras e artefatos**. 2019. Disponível em: <https://robsoncamargo.com.br/blog/Scrum-regras-artefatos>. Acesso em: 28 maio 2020.

EQUIPE Runrun.it. **Metodologia ágil: um presente da indústria de software para todo o universo da gestão**. 2020. Disponível em: <https://blog.runrun.it/metodologia-agil/>. Acesso em: 26 maio 2020.

LITTLEFIELD, Andrew. **Blog.Trello**. Guia da metodologia ágil e scrum para iniciantes. 2016. Disponível em: <https://blog.trello.com/br/scrums-metodologia-agil>. Acesso em: 28 maio 2020.

OTONI, Ana. Brasileiros gastam 650 horas por mês em redes sociais. **Blog O Globo**. Rio de Janeiro, 2015. Disponível em: <https://blogs.oglobo.globo.com/nas-redes/post/brasileiros-gastam-650-horas-por-mes-em-redes-sociais-567026.html>. Acesso em: 04 julho 2018.

SCHWABER, Ken; SUTHERLAND, Jeff. **The definitive guide to scrum: the rules of the game**. Mountain View, California, 2017. Disponível em: <http://www.scrumguides.org/docs/scrumguide/v2017/2017ScrumGuideUS.pdf#zoom=100>. Acesso em: 02 jul. 2018.

SPRINT Review Meeting. [Desenvolvimentoagil.com.br](http://www.desenvolvimentoagil.com.br). 2014. Disponível em: https://www.desenvolvimentoagil.com.br/scrums/sprint_review_meeting. Acesso em: 26 maio 2020.

VILLELA, Flávia. IBGE: 6,2% da população têm algum tipo de deficiência. **Portal EBC**. Rio de Janeiro, 2015. Disponível em: <http://www.ebc.com.br/noticias/2015/08/ibge-62-da-populacao-tem-algum-tipo-de-deficiencia>. Acesso em: 22 fev. 2018.

UTILIZAÇÃO E ORQUESTRAÇÃO DE CONTAINERS EM APLICAÇÕES WEB

João Antônio Caetano Rosa¹

José dos Reis Mota²

RESUMO: O presente artigo tem por objetivo principal abordar a utilização e orquestração de *containers* em aplicações *web*, assim como utilizar boas práticas e demonstrar técnicas de observabilidade aplicada nessa estrutura, visto que isso pode ser benéfico para empresas de soluções de TI no gerenciamento de suas aplicações. Neste estudo buscou-se analisar as principais técnicas de orquestração, garantia de boas práticas, escalabilidade e a utilização da observabilidade no ambiente da aplicação. Utilizou-se, no decorrer deste trabalho, o método organizacional *kanban* que visa aumentar a produtividade e otimizar a gestão do trabalho. Ressalta-se a importância da observabilidade e da maneira como os serviços interagem uns com os outros durante a execução, sendo preciso esse processo ser monitorado, gerenciado e controlado, isto começa com a capacidade de observação e a capacidade de entender o comportamento de arquiteturas de serviços distribuídos. Num âmbito geral buscou-se demonstrar que orquestração de *containers* em aplicações *web*, baseadas em serviços distribuídos e utilizada de maneira correta pode trazer vários benefícios para a organização.

PALAVRAS-CHAVE: Container. Orquestração. Observabilidade. Serviços Distribuídos.

ABSTRACT: The main objective of this article is to address the use and orchestration of containers in web applications, as well as to use good practices and demonstrate techniques of observability applied in this structure, since this can be beneficial for IT solution companies in the management of their applications. This study sought to analyze the main orchestration techniques, guarantee of good practices, scalability and the use of observability in the application environment. Use, in the course of this work, the *kanban* organizational method that aims to increase productivity and optimize work management. The importance of observability and the way services interact with each other during execution is emphasized. This process needs to be monitored, managed and controlled. This starts with the observation capacity and the ability to understand the behavior of architectures. distributed services. In a general scope, we tried to demonstrate that container orchestration in web applications, based on diversified services and used correctly, can bring several benefits to an organization.

KEYWORDS: Container. Orchestration. Observability. Distributed Services.

1 INTRODUÇÃO

A evolução das TICS (tecnologia de informação e comunicação) possibilitou à sociedade ter acesso a milhares de informações e complexidade de contextos, próximos ou distantes de sua realidade. Sendo assim, além da criação de padrões de projetos que auxiliam na utilização de padrões comuns para resolver problemas parecidos, surge a necessidade de criar ferramentas que auxiliam o processo de desenvolvimento com a

¹ Aluno de Sistemas de Informação, UNIPAM. E-mail: joaoantonio@unipam.edu.com.br.

² Mestre em Ciência da Computação, UFU. E-mail: josereis@unipam.edu.br.

¹Aluno de Sistemas de Informações, UNIPAM, raphaelcustodio94@gmail.com

² Mestre em Redes de Computadores, UFU, henaldobarros@gmail.com

finalidade de acompanhar essa evolução onde, desde a linha zero até o *deploy*, existem vários processos que podem impactar na qualidade do *software* que está sendo criado.

Quando se fala em aplicações *web*, normalmente a primeira coisa que se pensa é em um site, mas o que há por trás dessa aplicação, como ela funciona e como ela está ali disponível para o usuário, na maioria das vezes é algo bastante abstrato quando não se conhece o contexto do que se está sendo visualizado.

O processo de desenvolvimento de uma aplicação *web* é algo que vem gradativamente evoluindo ao longo dos anos e, como o processo evolui, as ferramentas também evoluem e o contexto torna-se cada vez mais abstrato, uma vez que “recriar a roda” é desnecessário, mesmo que muitas vezes seja preciso adaptar pequenas coisas para atender a distintas necessidades.

Durante muito tempo vem se discutindo a melhor maneira de se manter aplicações em funcionamento, várias teses são defendidas e algumas delas se destacam, com objetivo de atender ao máximo os requisitos que garantam a menor taxa de falha e maior excelência possível em todo o ciclo de vida da aplicação, uma vez que, várias maneiras e padrões são aceitos, tornando isso em algo simples ou extremamente complexo dependendo de como é aplicado.

Partindo de um ponto de vista cronológico, a execução do compilador ou interpretador na máquina *host*, até a criação de *containers*, mostra como a escala evolutiva de processos tecnológicos tornou toda a abstração do trabalho de manter a aplicação disponível mais simples, ou até mesmo complexo, ao mesmo tempo que facilita muita a vida de quem trabalha com isso no dia a dia. É claro que requisitos técnicos são necessários para se manter toda essa estrutura, porém a facilidade que essas tecnologias trouxeram e a maneira como agregam valor, tanto para a empresa quanto para os colaboradores de forma mais ágil, se torna inestimável e muitas vezes incalculável.

Quando se refere ao conceito “custo” entende-se como sendo despesas ou contas por exemplo. Entretanto, o custo pode ser visto de várias outras formas, como tempo, desgaste mental, e até mesmo recurso humano, uma vez que a criação e manutenção de processos malfeitos aumentam a carga de material humano para se garantir a vivacidade do produto de *software* da empresa.

Princípios como estabilidade, agilidade, manutenção, escalabilidade, dentre outros vários requisitos não funcionais, vem buscando cada vez mais facilitar todo o processo e ciclo de vida do projeto, tornando-o mais estável e padronizado.

Nessa perspectiva, o presente trabalho tem por objetivo principal abordar a utilização e orquestração de *containers* em aplicação *web*. Considerando o objetivo geral, os objetivos específicos são: estruturar aplicações *web* em *containers*, aplicar boas práticas de provisionamento de infraestrutura como código, buscando escalabilidade e a observabilidade em serviços distribuídos que utilizam *containers* orquestrados e gerenciados por plataformas de *cloud computing*.

A confecção deste estudo justifica-se por serem temas atuais e com alta demanda de mercado, já que grande parte das empresas de TI buscam aplicar essas características em suas aplicações.

A tecnologia de *containers* para a virtualização do ambiente de aplicações *web* pode ser algo muito positivo, pois gera uma economia do ponto de vista

computacional, uma vez que não é necessário a abstração de todo um ambiente de sistema operacional para a execução da aplicação e sim só os requisitos operacionais básicos, como o compilador ou interpretador e as dependências aplicadas para que aquele serviço funcione, de forma que tendo-se uma estrutura criada da maneira correta e seguindo boas práticas, é possível uma evolução gradual num ponto de vista de análise de desempenho e observabilidade de métricas da aplicação, que auxiliam tanto na tomada de decisão em nível técnico como também num ponto de vista de negócio.

2 REFERENCIAL TEÓRICO

Nesta seção são apresentadas as principais ideias e resultados de outros autores que pesquisaram tópicos diretamente associados ao tema do artigo, conceitos relacionados a aplicações *web*, estruturas de serviços distribuídos, aplicações monolíticas e observabilidade de sistemas. Para melhor organização do estudo, a abordagem será estruturada em subseções. 2.1 refere-se à estrutura e utilização de *containers* para ambos os casos; 2.2. refere-se a uma discussão sobre a utilização e orquestração de *containers* em aplicações *web*; 2.3. trata do tema de orquestradores e *cloud computing* para garantia de vários requisitos não funcionais que podem impactar na qualidade do projeto; 2.4 aborda sobre o conceito de observabilidade e seus três pilares buscando explicar sua importância em arquiteturas de serviços distribuídos.

2.1 ESTRUTURAS DE APLICAÇÕES WEB E CONTAINERS

Uma aplicação *web* pode ser vista muitas vezes como um *software* voltado para a utilização de usuários que tenham acesso a internet, porém o conceito vai muito além disso. Com a abstração de aplicações monolíticas ou a utilização de arquiteturas baseadas em serviços distribuídos, a complexidade pode aumentar bastante, entretanto toda evolução tem por objetivo uma melhoria, seja ela em processos ou até mesmo qualidade do produto. Se tratando de aplicações monolíticas a primeira coisa que se percebe é que, por mais que sua divisão lógica e o desenvolvimento de seu código visem por algo com o menor acoplamento, é possível notar que durante o processo de implantação ou *deploy* desse projeto, o simples fato de ser um monólito já impacta o funcionamento do serviço, uma vez que qualquer alteração de código influencia todo o processo de entrega e gera uma parada geral em qualquer ambiente o qual ele esteja implantada, seja em desenvolvimento, homologação ou produção, sendo a produção o ambiente mais crítico na maior parte dos casos.

Com o passar dos anos e com algumas lições aprendidas, tornou-se cada vez mais comum a utilização de arquiteturas de software baseadas em serviços distribuídos, buscando, dentre outros atributos, a redução do chamado *down time* durante o *deploy* da aplicação. Os serviços distribuídos baseiam-se em estruturas de *softwares* desejavelmente interdependentes e específicas para cada função, muitas vezes subdivididos em estruturas menores chamadas micro serviços, que geralmente comunicam entre si, visando assim o funcionamento modularizado da aplicação, e reduzindo o acoplamento em nível estrutural não somente de código, mas podendo

abstrair-se também na infraestrutura, tornando-se ideais para serem utilizados em *containers*.

Os chamados *containers* podem ser definidos como a virtualização da aplicação, não sendo necessário a abstração de todo um sistema operacional para que o *software* funcione, o *container* já trata de manter somente o necessário para o seu funcionamento como, a estrutura básica do sistema operacional, o compilador ou interpretador da linguagem que está sendo usada, e as dependências necessárias como bibliotecas de código já utilizadas anteriormente que facilitem o desenvolvimento do código dentre outras coisas, mas tudo sendo o mínimo possível para que o *software* não tenha seu desempenho afetado devido a utilização de estruturas e recursos desnecessários (VITALINO, CASTRO 2016).

2.2 UTILIZAÇÃO E ORQUESTRAÇÃO DE CONTAINERS EM APLICAÇÕES WEB

O contexto de aplicações web atuais se mostra um cenário bastante propício para a utilização de *containers* para virtualização e *deploy* das aplicações, uma vez que cenários onde se exige cada vez mais escalabilidade, resiliência e outros requisitos não funcionais tornam o “terreno” propício para que *containers* tenham cada vez mais espaço e sua utilização torna-se cada vez mais comum em qualquer âmbito de aplicações web, sejam elas aplicações monolíticas ou micro serviços.

O porquê disso acaba sendo óbvio a cada momento, pois grandes gerências de migrações e mudanças na forma de manter um serviço no ar, mostram que por mais que se tenha um esforço significativo no início, isso acaba sendo bem proveitoso a longo prazo. O mesmo paradigma aplica-se na forma de orquestrar pois, quando uma arquitetura já é construída se pensando não somente no seu modelo de aplicação, sendo ela monolítica ou de microsserviços, mas também em sua infraestrutura, isso pode acabar eliminando vários problemas e reduzindo riscos, ao longo da vida do *software* que está sendo trabalhado (CAMISSO, 2020).

O ciclo de vida de uma aplicação *web*, dentro deste contexto, depende de vários fatores como, requisitos de produto ou até mesmo método de desenvolvimento bem ou mal implementados, que podem acarretar numa futura refatoração do código, o que acaba sendo custoso quando o processo é mal planejado. Partindo do pressuposto de que recriar a roda é algo completamente desnecessário, a utilização da tecnologia de *containers*, mostra que o que já foi aprendido até hoje pode se tornar cada vez mais útil e aprimorável, visto que apenas utilizar tecnologia não é garantia que se está fazendo inovação da forma correta pois a principal meta é entregar valor no que se está sendo proposto.

A decisão de utilizar um orquestrador passa sempre por um processo avaliativo, sendo que o contexto em que o mesmo será utilizado pode ser o fator decisivo para a sua implementação, pois tudo que será feito deve ter um propósito e um significado específico para atingir as metas desejadas e entregar o máximo de valor possível, com o melhor custo benefício, e é por isso que o *docker-swarm* acabou sendo deixado de lado e a utilização do *kubernetes* se sobressai ao mesmo (SANTOS, 2020).

2.3 ORQUESTRADORES DE CONTAINERS E CLOUD COMPUTING

Não dá para se falar em orquestração de *container* sem se falar de *frameworks* como *Kubernetes* ou *Docker-Swarm*, que são as principais e mais conhecidas plataformas de orquestração de *containers* da atualidade, uma vez que se tratando não somente de *Cloud Computing* a utilização dessas poderosas ferramentas podem também ser utilizados em ambientes de servidores *On Promise* sendo sempre uma ótima opção quando se fala em escalabilidade e resiliência, uma vez que as mesmas foram pensadas para resolver problemas como esses e outros mais.

Partindo do princípio de *Cloud Computing*, é possível observar que desde que grandes empresas como *Google*, *Amazon* e *Microsoft*, começaram a oferecer este tipo de opção, a disputa por quem oferece o melhor serviço se tornou cada vez mais acirrada e o *Kubernetes*, que foi uma iniciativa do *Google* e hoje pertence a *Cloud Native Foundation*, se tornou o que é chamado de a nuvem dentro da nuvem. Isso por que esta poderosa ferramenta é capaz de não somente realizar a orquestração de *containers*, como também a gerência de servidores, agindo quase que como uma camada abaixo do sistema operacional gerenciando todo o *host* da aplicação (SANTOS, 2020).

Por mais que o *Docker-Swarm* seja um bom orquestrador, o *Kubernetes* está a anos luz na sua frente, como pode-se perceber ao analisar estatísticas do site “*dzone*” onde o *kubernetes* supera o *Docker-Swarm* não só em comparações de desempenho como também em pesquisas e utilização por maior parte das empresas de tecnologia, e é por isso que ele será o orquestrador mais citado e utilizado para a experimentação e desenvolvimento deste trabalho (DZONE, 2020).

2.4 OBSERVABILIDADE

Para se compreender melhor sobre o que é observabilidade, primeiro é preciso deixar bem claro o que não é observabilidade.

A observabilidade não é uma caixa pronta, não é um produto que a empresa compra ou desenvolve para “monitorar” o ciclo de trabalho de suas aplicações, a observabilidade é algo que deve ser construído, a aplicação precisa atender a uma série de requisitos para se tornar observável da maneira correta, e isso é atingido através de uma estrutura arquitetural feita de forma inteligente, que permita que a mesma atinja esses requisitos e venha a se tornar observável da forma desejada (GOMES, 2020).

De forma clara e objetiva a observabilidade pode ser descrita como a capacidade de observar os estados anteriores e o estado atual da aplicação, monitorar e detectar falhas eventuais em qualquer ponto do sistema, tendo como objetivo principal não só a manutenção corretiva, como também, a análise da causa raiz da falha, com o objetivo de impedir que esse mesmo erro ocorra novamente não só naquele ponto, mas em qualquer outro dentro do ciclo de trabalho de aplicação (SRIDHARAN, 2020).

Dentro da observabilidade existem três pilares muito bem definidos que constituem todo o conceito por trás da mesma, são eles, *logs*, *metrics* e *tracing*.

2.4.1. *Logs*

Logs são definidos como um conjunto de registros que informam os atuais eventos da aplicação, de modo que os desenvolvedores e até mesmo o pessoal de operação, possam detectar possíveis falhas ou avisos indesejados que estejam ocorrendo no momento atual da visualização e leitura dos mesmos.

Os *logs* são importantes pois na maioria dos casos eles são cruciais para a se saber o que pode ter ocasionado o erro que fez o sistema parar ou não funcionar da maneira desejada naquele momento. Os logs são a consulta direta à informação que o serviço expõe, mostrando o que aconteceu no exato momento da falha e qual foi a tratativa de exceção lançada pelo sistema durante o evento (PRONSCHINSKE, 2020).

2.4.2. *Metrics*

Metrics (métricas) são a coleta de informações de qualquer estado anterior ou atual da aplicação no nível não só de consumo de recursos computacionais, como também de negócio, sendo peças chave e essenciais para a tomada de decisão dentro da organização, servindo como painel de observação de como se dá o comportamento cronológico de todos os estados do serviço.

É necessário saber como utilizar as métricas de forma correta, nem em falta nem em excesso, pois as mesmas são uma das ferramentas que possibilitam à empresa tomar as melhores decisões não só técnicas como de negócio, sendo possível decidir para qual rumo a organização deve seguir a partir de cada análise (PRONSCHINSKE, 2020).

2.4.3 *Tracing*

Dentre os três pilares, o *Tracing* talvez seja o mais “difícil” de se aplicar, pois, o mesmo veio se tornando cada vez mais necessário em arquiteturas de serviços distribuídos, uma vez que é através dele que é possível rastrear toda a rota que a requisição de falha tomou, sendo possível saber qual foi o ponto exato que disparou o alerta dentro do fluxo da malha de serviços.

A abstração de malha se dá pelo fato de que o *tracing*, utiliza a mesma dentro da camada de serviços, permitindo um rastreamento único e identificável dentro do próprio ambiente da aplicação, adicionando um índice no cabeçalho da requisição como um identificador único para cada uma delas, ou até mesmo um rastreamento de ponta a ponta dentro de toda a malha de serviços (PRONSCHINSKE, 2020).

Saber o ponto exato da falha é essencial, e garante não só a rápida correção, como também uma análise bem mais precisa para a prevenção e previsão de possíveis incidentes que venham a ocorrer devido ao mesmo motivo, ou por fluxos de trabalho parecidos.

3 METODOLOGIA

O presente trabalho foi baseado na análise do uso de boas práticas, para a criação de ambientes preparados para serem utilizados por orquestradores de *containers* e aplicando técnicas de observabilidade de sistemas. O KANBAN, que faz parte das metodologias ágeis, foi utilizado para auxiliar na execução de tarefas e gerência do processo de desenvolvimento deste trabalho.

O KANBAN baseia-se na criação de tarefas com datas pré-dispostas e entrega de resultados após a execução de cada tarefa, agregando assim continuamente valor e agilidade em todo o processo de desenvolvimento (ABRANTES, 2020). As principais tecnologias e ferramentas utilizadas para a execução do projeto foram as seguintes:

- **Kubernetes:** “Kubernetes (K8s) é um produto *Open Source* utilizado para automatizar a implantação, o dimensionamento e o gerenciamento de aplicativos em *container*” (KUBERNETES, 2020).
- **Docker:** Docker é uma tecnologia de *containers open source* que visa virtualizar o ambiente de aplicação e isolamento de recursos, evitando assim a sobrecarga e utilização de máquinas virtuais na abstração de todo um sistema operacional para execução do *software* (DOCKER, 2020).
- **Git:** “Git é um sistema de controle de versão distribuído gratuito e de código aberto projetado para lidar com tudo, desde projetos pequenos a muito grandes com velocidade e eficiência” (GIT, 2020).
- **Gpc:** O *Google Cloud Platform* ou GPC é a plataforma do *Google* que oferece uma gama de serviços baseada em computação em nuvem (G-CLOUD, 2020).
- **Gke:** “Serviço seguro e gerenciado do *Kubernetes* com suporte de escalonamento automático de quatro direções e *multicluster*” (GKE, 2020).
- **Terraform:** “Terraform é uma ferramenta para criar, alterar e criar versões de infraestrutura com segurança e eficiência. Terraform pode gerenciar provedores de serviços existentes e populares, bem como soluções internas personalizadas” (TERRAFORM, 2020).
- **Istio:** “O Istio é uma plataforma *open source* de criação e gerenciamento de *service mesh*, permitindo conectar, proteger, controlar e observar toda a malha de serviços e comunicação dos mesmos dentro de um *cluster*” (ISTIO, 2020).
- **Jaeger:** “Jaeger é um sistema de rastreamento distribuído lançado como código aberto pela *Uber Technologies*. Ele é usado para monitorar e solucionar problemas de sistemas distribuídos baseados em microsserviços” (JAEGER, 2020).
- **Kiali:** “Kiali é um console de gerenciamento para malha de serviço baseada no Istio. Ele fornece painéis, capacidade de observação e permite operar sua malha com recursos robustos de configuração e validação. Ele mostra a estrutura de sua malha de serviço inferindo a topologia de tráfego e exibe a integridade de sua malha. Kiali fornece métricas detalhadas, validação poderosa, acesso Grafana e forte integração para rastreamento distribuído com Jaeger” (KIALI, 2020).

- **Grafana:** “O Grafana permite que você consulte, visualize, alerte e entenda suas métricas, independentemente de onde estejam armazenadas. Crie, explore e compartilhe painéis com sua equipe e promova uma cultura baseada em dados” (GRAFANA, 2020).
- **Prometheus:** “O Prometheus é um kit de ferramentas de monitoramento e alerta de sistemas de código aberto” (PROMETHEUS, 2020).
- **Vegeta:** “Vegeta é uma ferramenta versátil de teste de carga *HTTP* criada a partir da necessidade de perfurar serviços *HTTP* com uma taxa de solicitação constante. Ele pode ser usado como um utilitário de linha de comando e uma biblioteca” (VEGETA, 2020).

4 DESENVOLVIMENTO E RESULTADOS

Seguindo um fluxo de pesquisa e desenvolvimento contínuo, o KANBAN se mostrou extremamente eficiente, uma vez que as *tasks* e o fluxo de trabalho foram se tornando cada vez mais fluidos à medida do tempo. A execução de cada tarefa foi cronologicamente primordial para o desenvolvimento das seguintes, uma vez que a flexibilidade da metodologia permitiu que mesmo que não fosse em ordem cronológica, o desenvolvimento pudesse ser quebrado em partes importantes que constituíram uma pesquisa que proporcionou a cada etapa uma nova descoberta e comprovação das teses, o que pode ser satisfatória para o propósito final do estudo realizado a respeito da tecnologia de *containers* e uso de orquestradores, mostrando-se uma prática fortemente difundida em todo mundo devido aos seus benefícios para todo o âmbito, não só técnico como financeiro, num ponto de vista organizacional e de negócio.

O primeiro passo foi a coleta de dados a respeito da tecnologia de *containers*, permitindo uma melhor compreensão do funcionamento da mesma, sendo possível uma análise do que estaria por vir, à medida que foi se aprofundando o estudo e testes manuais em cima do *docker*, que se mostrou flexível para a entrada de novas possibilidades e ponto de vistas mais interessantes para o funcionamento de aplicações, sendo de fácil gerenciamento para software baseados em arquitetura distribuída e de infraestrutura como código.

Contudo, mesmo o *docker* sendo uma peça chave para o estudo, foi possível observar que ele fazia parte de toda uma escala evolutiva para se garantir os níveis necessários de infraestrutura inteligente e observável, que foram as principais propostas descritas para o desenvolvimento da solução.

Uma tarefa importante e crucial foi a pesquisa e validação de algumas soluções em *cloud* para se ter ideia de qual seria a mais viável para a criação do protótipo, e o *gpc* se mostrou à frente das demais soluções, não só financeiramente, mas também estruturalmente, além de possuir uma versão *trial*. Por possuir os atributos necessários para o desenvolvimento do trabalho, optou-se por utilizar essa plataforma. Vale a pena lembrar que a solução de orquestração de *container* independe da plataforma de *cloud*, uma vez que a abstração de sua arquitetura é flexível e bastante compatível com a maioria delas, o que facilita a criação de *clusters kubernetes* gerenciados pelo próprio serviço provedor de *cloud*.

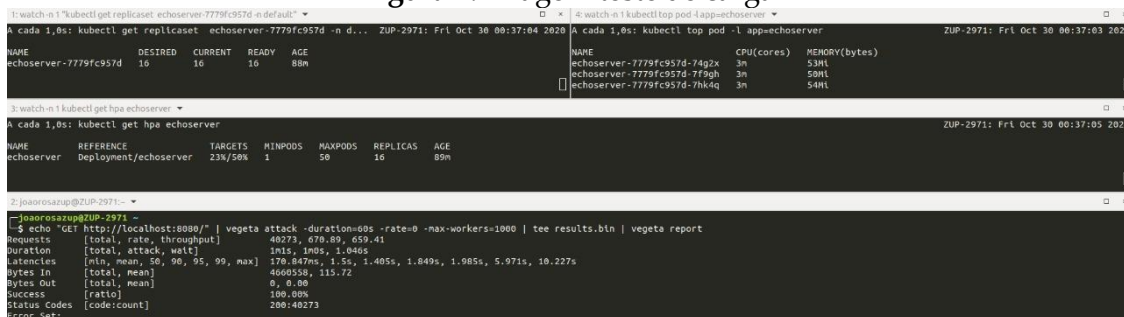
O *kubernetes* mostrou-se mais avançado que qualquer outro serviço de orquestração de *containers*. À medida em que foi sendo aprofundado o estudo desta ferramenta, percebeu-se que suas soluções de automação e gerência são eficientes e fáceis de implementar. Esta etapa foi muito tranquila de ser administrada, pois o *kubernetes* possui uma ampla documentação e comunidade ativa, sendo possível achar milhares de exemplos para cada solução desejada, de acordo com o propósito do estudo.

Após o domínio essencial deste orquestrador, foi a vez de pôr em prática a criação do ambiente, nesse caso do *cluster* de *kubernetes*. O *cluster* foi criado através do *terraform*, sendo usado nesse estudo como a principal ferramenta de provisionamento de infraestrutura como código.

Para que fosse possível testar as soluções propostas por este estudo, foi feito o *deploy* de uma aplicação base conhecida como *echo server*, disponível em “<https://github.com/orian/go-echoserver>” sendo muitas vezes utilizada para testes de implantação em ambientes de desenvolvimento, o que constitui uma boa prática dentro do universo *Devops*.

Após o *deploy* foi feita a análise do comportamento da aplicação em situações de estresse, através de testes de carga com a ferramenta *vegeta* para a verificação de escalabilidade comportamental que o software teria no momento que o fluxo de requisições fosse aumentando e exigindo cada vez mais performance e resiliência do mesmo, conforme mostra a Figura 1.

Figura 1: Imagem teste de carga



Fonte: Dados do trabalho, 2020.

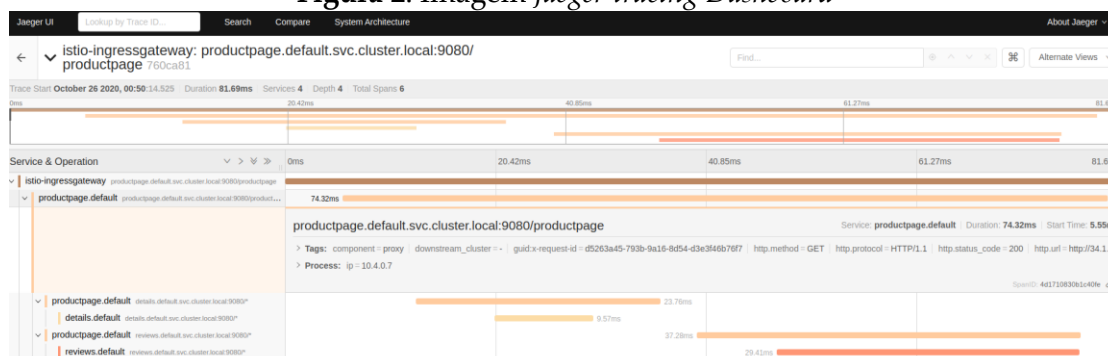
Foi possível perceber através da Figura 1 que os testes foram bastante satisfatórios, em relação à performance em um ponto de vista de escalabilidade, pois quando o número de requisições aumentou em um curto espaço de tempo, a sua performance não deixou a desejar, já que através da solução de *HPA(horizontal pod autoscaling)* fornecida pelo *kubernetes*, que é feita através de arquivos de configuração da própria plataforma, a escalabilidade horizontal automática da aplicação se torna uma poderosa aliada para a garantia de resiliência da mesma.

Após a execução desta tarefa foi a vez de criar uma malha de serviços e prover uma rede *mesh* através do *istio*. Seguindo a documentação foi possível verificar que o mesmo possui um binário próprio de instalação e gerência de seus recursos dentro do *cluster*. O *kiali* também foi implementado no *cluster*, sendo uma das *features* que o *istio* traz para a visualização da malha de serviços do ambiente.

Com o intuito de simplificar o processo, todos os recursos presentes no *istio* foram instalados de forma padrão e seguindo por base o passo a passo exposto em sua documentação. Foi realizado também o *deploy* de um conjunto de micro serviços base denominado *Bookinfo*, que o próprio *istio* fornece, disponível em “<https://istio.io/latest/docs/examples/bookinfo/>”. Este serviço foi utilizado para a exemplificação da utilização dos recursos que visam mostrar como funciona a observabilidade dentro do ambiente da aplicação.

O passo seguinte foi a implantação de um recurso chamado *Jaeger* que é o principal responsável por todo o processo de *tracing* das requisições que a aplicação venha a receber. Com ele é possível visualizar todo o caminho que a requisição fez até o exato ponto da falha ou da conclusão, o que torna o processo de depuração em caso de incidentes bem mais fácil de ser realizado, como é possível observar na Figura 2.

Figura 2: Imagem *Jaeger tracing Dashboard*

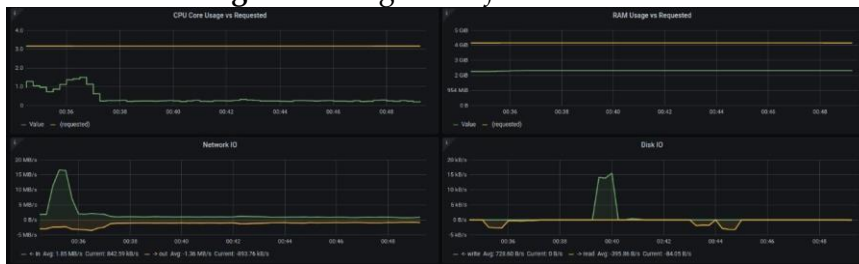


Fonte: Dados do trabalho, 2020.

Com o *Jaeger* em funcionamento foi a vez de utilizar o *grafana* como centralizador de *dashboards*, para a visualização das métricas expostas pelos serviços do cluster. O *grafana* foi utilizado localmente, uma vez que para o propósito deste trabalho não foi necessário a criação de um ambiente separado para o mesmo. No entanto, é uma boa prática utilizá-lo fora do ambiente da aplicação, para que a mesma não interfira em seu funcionamento em caso de incidentes.

Para que o *grafana* funcione da maneira correta, é necessário também um serviço de exposição de métricas, pois ele é utilizado somente para que sejam consultados e criados os *dashboards* de monitoramento e possíveis alertas com base nas métricas consumidas. Partindo desse pressuposto, foi a vez de expor o serviço do *prometheus*, que por padrão já vem implantado no cluster de *kubernetes* gerenciado do *gcloud*, onde seguindo as devidas configurações, o *prometheus* passou a exportar para o *grafana* todas as métricas encontradas no ambiente. Logo após conexão entre os dois serviços ser realizada com sucesso, foram criados os respectivos *dashboards* de visualização de métricas, concluindo assim parte da *stack* de observabilidade, como mostra a Figura 3.

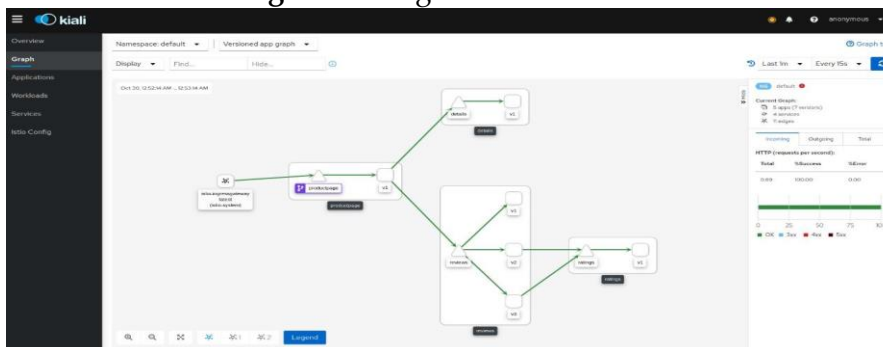
Figura 3: Imagem *Grafana Dashboard*



Fonte: Dados do trabalho, 2020.

Em seguida, foi realizado o teste de funcionamento de toda a *stack* de observabilidade, através de requisições feitas no serviço de Bookinfo, onde foi possível observar no *dashboard* do *kiali*, representado na Figura 4, o fluxo que as requisições seguiram até chegar em seu destino final.

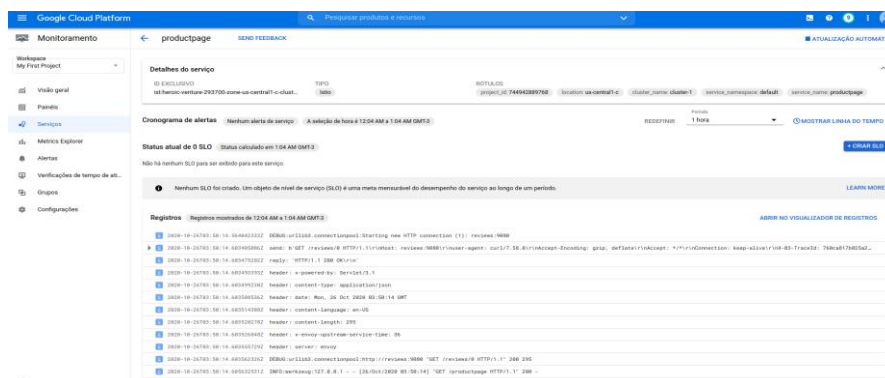
Figura 4: Imagem *kiali Dashboard*



Fonte: Dados do trabalho, 2020.

Depois disso, foi a vez de verificar os *logs* da aplicação, uma boa prática é manter isso centralizado de forma que fique fácil para a equipe de operações agir da forma mais rápida e precisa possível, pois nesses casos é necessário fazer a junção de *logs* de todas as aplicações envolvidas no processo. Na Figura 5 está representada a utilização do serviço de *StackDriver* do *Google* para a centralização e visualização de *logs* dos serviços.

Figura 5: Imagem *logs Stackdriver*



Fonte: Dados do trabalho, 2020.

5 CONCLUSÃO

O presente trabalho teve por objetivo abordar as principais técnicas e boas práticas relacionadas à utilização e orquestração de *containers* em aplicações *web*, com foco voltado para arquitetura de serviços distribuídos, buscando demonstrar a resiliência dos serviços através de testes de carga e abordar a importância da observabilidade aplicada de forma correta, sendo usada para garantir o monitoramento e tomada de decisão em nível técnico e de negócio.

Com base nos estudos apresentados, é possível se ter uma ideia de qual direção tomar para se implementar da melhor forma possível a tecnologia de *containers* orquestrados na infraestrutura de aplicações. Espera-se que esse estudo possa contribuir com desenvolvedores e empresas que visam uma melhoria contínua em seus processos de implementação de infraestruturas inteligentes e auto gerenciadas, onde a aplicação da observabilidade pode garantir vários benefícios antes não explorados.

O trabalho teve grande parte de seus objetivos alcançados, uma vez que o uso de *softwares* bases para testes, muitas vezes não abrangem todos os casos de uso que uma aplicação empresarial e complexa venha a exigir para se garantir a excelência específica de todos os seus pontos. Porém, os mais importantes, ou grande parte deles, puderam ser testados. Comprovou-se que, as técnicas utilizadas garantem que a maioria dos requisitos não funcionais de aplicações baseadas em *containers* e serviços distribuídos venham a necessitar puderam ser alcançados.

Ao longo dos anos as pesquisas sobre utilização e orquestração de *containers* vem evoluindo cada vez mais. Uma enorme gama de ferramentas e técnicas vem surgindo a cada dia, o que torna essa área de estudo bastante atrativa para vários estudiosos e entusiastas de tecnologia. Tendo isso como base, o estudo pode ser uma boa referência para quem desejar empregar essas técnicas em seus processos de desenvolvimento de software, na criação de aplicações baseadas em *containers* com arquitetura de serviços distribuídos e infraestruturas provisionadas como código.

REFERÊNCIAS

ABRANTES, Letícia. **Kanban**: entenda o que é e como funciona o método Kanban. 2018. Disponível em: <https://rockcontent.com/blog/kanban/>. Acesso em: 20 maio 2020.

CAMISSO, Jamon; JETHA, Hanif; JUELL, Kathleen. *Kubernetes For Full-Stack Developers*. **DigitalOcean**, New York City, New York, USA. 2020.

DOCKER. **About Docker**. Disponível em: <https://www.docker.com/company/>. Acesso em: 28 junho 2020.

DZONE. **“Docker Swarm ou Kubernetes?”: é a pergunta certa a se fazer?**. Disponível em: <https://dzone.com/articles/quotdocker-swarm-or-kubernetesquot-is-it-the-righ/>. Acesso em: 28 jun. 2020.

GCLOUD. **Ofereça mais soluções com o Google Cloud**. Disponível em: <https://cloud.google.com/>. Acesso em: 22 maio 2020.

GIT. **Sistema de controle de versão distribuído**. Disponível em: <https://git-scm.com/>. Acesso em: 25 maio 2020.

GKE. **Google Kubernetes Engine**. Disponível em: <https://cloud.google.com/kubernetes-engine?hl=pt-br/>. Acesso em: 25 maio 2020.

GOMES, Pedro César Tebaldi. **O que é observabilidade e qual a diferença para a monitoração?**. Disponível em: <https://www.opservices.com.br/observabilidade/>. Acesso em: 30 junho 2020.

GRAFANA. **A plataforma de análise para todas as suas métricas**. Disponível em: <https://grafana.com/grafana/>. Acesso em: 26 junho 2020.

ISTIO. **What is Istio?**. Disponível em: <https://istio.io/latest/docs/concepts/what-is-istio/>. Acesso em: 25 junho 2020.

JAEGER. **Jaeger**: open source, end-to-end distributed tracing. Disponível em: <https://www.jaegertracing.io/>. Acesso em: 30 junho 2020.

KIALI. **Service mesh management for Istio**. Disponível em: <https://kiali.io/>. Acesso em: 30 junho 2020.

KUBERNETES. **Orquestração de containers prontos para produção**. Disponível em: <https://kubernetes.io/pt/>. Acesso em: 20 maio 2020.

PROMETHEUS. **What is Prometheus?**. Disponível em: <https://prometheus.io/docs/introduction/overview/#what-is-prometheus/>. Acesso em: 30 junho 2020.

PRONSCHINSKE, Mitch. **Monitoring demystified**: a guide for logging, tracing, metrics. Disponível em: <https://techbeacon.com/enterprise-it/monitoring-demystified-guide-logging-tracing-metrics/>. Acesso em: 25 ago. 2020.

SANTOS, Lucas. **Kubernetes**: tudo sobre orquestração de containers. Casa do Código, Rua Vergueiro, 3185, 8º andar, São Paulo, SP, Brasil. 2020.

SRIDHARAN, Cindy. **Distributed systems observability**: a guiding to building robusted systems. O'reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. maio 2018.

TERRAFORM. **O que é Terraform?**. Disponível em: <https://www.terraform.io/intro/index.html/>. Acesso em: 30 jun. 2020.

VEGETA. **Vegeta**. Disponível em: <https://github.com/tsenart/vegeta/>. Acesso em: 30 jun. 2020.

VITALINO, J. F. N.; CASTRO, M. A. N. **Descomplicando o Docker**. Brasport Livros e Multimídia LTDA., Rua Pardal Mallet, 23, Tijuca, Rio de Janeiro, Brasil. 2016.