

Revista e-Tec de Ciência e Tecnologia

A photograph of an industrial facility at night, featuring large cylindrical tanks, a central metal structure, and walkways illuminated by bright lights. The scene is viewed from an elevated position, possibly a walkway or platform, with a yellow safety railing in the foreground.

Faculdade da Indústria SENAI Londrina

Edição V2 – ano 2022 – Londrina Paraná

Corpo Editorial

Editor Chefe – Edição V2-2022
Prof. MS.c. Anderson Ávila dos Santos

Comitê Executivo
Prof.^a Adriana G. Carvalho
Prof. MS.c. Anderson Ávila dos Santos
Prof.^a Dra. Camila F. de Oliveira
Prof. Fábio R. Milanez
Prof. MSc. Renato Kazuo Myiamoto
Prof. Dr. Rodolfo A. Hildebrandt
Prof. Dr. Vicente de Lima Gongora
Prof. Wesley Candido da Silva

Editora: Faculdade de Tecnologia Senai Londrina
ISSN: 2358-5528

Direitos reservados
Faculdade de Tecnologia SENAI Londrina
Rua Belém, 844 – Londrina PR.
43 -3294-5100 – faculdade.londrina@sistemafiep.org.br

Partes desta publicação, ou todo, pode ser utilizada para fins didáticos somente, desde que, citadas suas fontes.

Este periódico publica nomes individuais, comerciais, marcas registradas e produtos pertencentes a diversas companhias. O Editor utiliza-se destes nomes somente para fins editoriais e em benefício dos proprietários

dos nomes e marcas, sem intenção de atingir seus direitos.

Observa-se ainda que os dados contidos nos artigos são de responsabilidade dos próprios autores dos artigos.

e-TEC

Revista de
Tecnologia e
Ciência

Prefácio

Nesta época de necessárias e rápidas transformações estão as instituições preocupadas em acompanhar as tendências no exterior e no Brasil?

Acredita-se que sim, pois aqui na Faculdade da Indústria SENAI Londrina estas transformações já se iniciaram; nossos docentes estão utilizando as melhores práticas de educação nos cursos de engenharia e de tecnologia.

Participando junto com a comunidade acadêmica em um maior diálogo com a sociedade industrial, ao longo de todo o processo da formação do perfil profissional do egresso.

Esses diálogos, apresentam como resultado o desenvolvimento de projetos e soluções que fazem todo o sentido aos cursos ofertados, além de, estabelecerem duradouras e harmoniosas relações.

Mesmo que em etapas e com diferentes abrangências, esses movimentos são, por nós, muito celebrados e devem ser disseminados; pois sinalizam as boas práticas da mudança positiva na cultura das instituições.

Por esse motivo, merecem todo o apoio em sua divulgação, é o que este periódico tem por objetivo principal e efetivamente se propõe.

Salientando que, esse periódico apresenta em diversas formas a “inspiração”, para a comunidade em geral; oportunizando a experiência de através da compreensão e discussão dos trabalhos de outros autores, ou mesmo submeter o seu próprio trabalho para a compreensão e discussão pela comunidade.

Utilizando-se desta experiência para uma futura e melhor produção, uma vez que a partir de boas práticas de discussão os aperfeiçoamentos possam ser estabelecidos e desta forma um maior impacto científico poderá ser alcançado.

Incentiva-se de diversas formas estas publicações, com a execução e desenvolvimento de resultados experimentais que possuem aplicações reais na indústria em nossa metodologia ativa - jornadas de aprendizagem.

Evidências da nossa exitosa metodologia ativa – Jornadas de Aprendizagem podem ser verificadas durante o “Tempo de Falar”. A capa desta edição é uma foto tirada em visita ao Moinho M.Dias Branco, Cidade de Rolândia Pr.

Forte abraço e boa leitura, prof. Dr. Vicente Gongora

Sumário

Capítulo I – Gestor de Gistema: Gecurity	4
Capítulo II – NOBLEWARE: Estimativas de projeto de software ..	21
Capítulo III – Uma solução <i>web</i> e <i>mobile</i> para predição e análise de sentimentos sobre o avanço da Covid-19	34
Capítulo IV – Melhorando a comunicação com a imobiliária por meio de um sistema web	45
Capítulo V – SENAI METRICS: IMPULSIONANDO SEU NEGÓCIO COM DADOS.....	56

Capítulo I – Gestor de Sistema: Gecurity

Isabella Aparecida Fraga
Michell Sales Leonne
Pedro Henrique Moraes
Rogério Dagnon Candeira
Fábio Takeshi Matsunaga¹
Lucas Busatta Galhardi²
Anderson Paulo Avila Santos³

RESUMO

O principal objetivo deste trabalho é o desenvolvimento de um sistema que integra as equipes comerciais e técnicas com o gerente da empresa, facilitando assim a análise de um serviço que está para iniciar ou que está em andamento. Na prática o sistema proposto tem total controle das equipes, serviços, produtos, agendamentos, metas, análises, relatórios de serviços e histórico. O desenvolvimento da solução passou por 2 etapas, onde na segunda foi-se melhorado e realizado um re-design no sistema todo que foi além do visual, como a logo, os padrões de cores, tipografia e ícones (identidade visual). Além disso, houve também mudanças direcionadas para a adaptação e a inclusão da acessibilidade, dando às pessoas com deficiência mais oportunidades dentro das empresas. O sistema até o momento está remodelado para pessoas que possuem deficiência visual – baixa visão e daltonismo (ícones/cores e fontes ampliadas) e auditiva (libras). Agora o sistema foi integrado com um aplicativo mobile para agilizar ainda mais os serviços, para que o técnico na rua não perca tempo escrevendo relatórios ou aguardando recebimento de ordens para conclusão do serviço.

Palavras-chave: Gestão, SI, Segurança, Patrimonial, Acessibilidade, Libras, Mobile.

SYSTEM MANAGER: SECURITY

ABSTRACT

The main objective of this work is the development of a system that integrates the commercial and technical teams with the company manager, thus facilitating the

¹ Mestre em Ciência da Computação. E-mail: anderson.avila@sistemafiep.org.br

² Mestre em Ciência da Computação. E-mail: lucas.galhardi@sistemafiep.org.br

³ Mestre em Ciência da Computação. E-mail: fabio.matsunaga@sistemafiep.org.br

analysis of a service that is about to start or that is in progress. In practice, the proposed system has full control of teams, services, products, schedules, goals, analyses, service reports and history. Development of the solution went through 2 stages, in which the second stage was improved, and a re-design was carried out on the entire system that went beyond the visual, such as the logo, color patterns, typography and icons (visual identity). In addition, there have also been changes aimed at adapting and including accessibility, giving people with disabilities more opportunities within companies. The system so far is remodeled for people who have visual impairments – low vision and color blindness (icons/colors and enlarged fonts) and hearing impairment (libras). Now the system has been integrated with a mobile application to further streamline the services, so that the technician on the street does not waste time writing reports or waiting to receive orders to complete the service.

Keywords: Management, IS, Security, Property, Accessibility, Libras, Mobile.

1. INTRODUÇÃO

A crescente violência urbana vem aumentando a busca por sistemas de vigilância mais desenvolvidos, que consigam realizar um monitoramento de forma autônoma, garantindo um melhor controle e segurança.

Diversas empresas de segurança patrimonial contam com grandes equipes técnicas e comerciais, mas não é o suficiente para ter qualidade nos atendimentos e serviços prestados, a falta de estrutura básica de um software para facilitar o dia a dia e a comunicação entre os colaboradores e clientes, pode gerar problemas como atrasos, equívocos e erros gerando assim transtornos. Controle é fundamental para melhorar a agilidade e eficiência no atendimento ao cliente.

Independentemente do tamanho da empresa e da quantidade de sistemas envolvidos na rotina do negócio, controlar a abertura de chamados e finalização de serviços é sinal de organização e eficácia. Pensando nisso, desenvolvemos o gestor de sistema Security, para ajudar as grandes, médias e pequenas empresas de segurança a melhorarem os seus serviços.

O sistema facilita a comunicação entre usuários e responsáveis técnicos, agiliza os processos de solicitação de atendimento e centraliza as informações em um único local para conferência da empresa e de clientes. Assim, nenhum detalhe é perdido, sem conversas cruzadas e falta de interpretação das duas partes.

O sistema registra a data e horário da abertura de chamados e orçamentos, o responsável e demais informações para contato. Além de arquivar orçamentos já realizados e serviços que já foram concluídos para possíveis consultas futuras.

O sistema foi baseado sobre estudos e análises de empresas do ramo de segurança patrimonial da cidade de Londrina, com uma estimativa de 20 empresas na área.

2. FUNDAMENTAÇÃO TEÓRICA

Inovação é um conceito amplo, mas na essência é a exploração com sucesso de novas ideias, conceitos ou práticas, quando disruptiva, transforma um mercado ou até um setor inteiro e geralmente faz isso através da introdução de simplicidade, conveniência e acessibilidade em grande escala.

No mundo dos negócios essas inovações, que podem ou não estarem ligadas a tecnologia, precisam gerar valor ao negócio, melhorando a performance, a qualidade ou a satisfação do cliente. Nesse ponto a organização é fundamental para que todas novas ações melhorem a experiência do cliente com os serviços prestados.

Segurança Patrimonial: Soluções Completas, Customizadas e Eficientes. Para cada cliente. Além de controles dos mais diversos e uma gestão eficiente, todas essas características garantem soluções completas em segurança e outros serviços que fortalecem a parceria de clientes e empresas. O resultado é a melhoria constante das operações que agregam valor ao negócio.

O poder transformador que os avanços tecnológicos vêm provocando em nossas áreas de atuação, principalmente no segmento de segurança nos diz onde deve estar o foco tanto de clientes como para empresas. Contudo, a tecnologia também possui formas criativas e inovadoras de realizar aplicações da tecnologia, e tem proporcionado melhorias em diversas soluções que vão além de mão de obra, com performance superior e redução de custos.

3. METODOLOGIA

3.1 Linguagens Utilizadas

O desenvolvimento web foi feito em Django, um framework para desenvolvimento rápido para web, escrito em Python, onde ele utiliza o padrão model-template-view. Foi-se utilizado o banco de dados padrão do Django: o SQLite.

O concorrente mais direto do Python é o Perl, vamos enfatizar os motivos por ter escolhido o Python como a Linguagem padrão do projeto. Python e Perl vêm de uma base semelhante e possuem também características parecidas, mas no fim têm uma filosofia diferente. Perl enfatiza o suporte para tarefas comuns orientadas a aplicativos. Agora o Python enfatiza o suporte para metodologias de programação comuns, como

desenho de estrutura de dados e programação orientada a objetos, e encoraja os programadores a escrever um código legível e sustentável. Uma semelhança delas que as fazem concorrentes tão diretas são porque suportam a programação procedural e orientada a objetos, e são linguagens multiparadigmáticas, interpretadas e de alto nível. Ambas podem se integrar com banco de dados que utilizam a linguagem SQL como interface. Podem ser usadas para desenvolver interfaces gráficas de usuário e distinguir entre letras maiúsculas e minúsculas em sua programação.

Para estilização utilizamos o CSS, que é uma linguagem para adicionar estilo ao documento web, também utilizamos o Bootstrap, que é um framework web utilizado para desenvolver componentes de interface e front-end para sistemas e aplicações web utilizando o HTML.

Normalmente o Javascript é utilizado junto com linguagens de marcação e regras, como o HTML e o CSS. Assim, é possível ter muito mais do que textos e tabelas coloridas, mas sim um conteúdo que realmente vai prender a atenção dos usuários da web.

3.2 Redesign

O redesign feito no sistema foi além do visual, como a logo, os padrões de cores, tipografia e ícones. Ele foi totalmente baseado nas necessidades reais do consumidor. Por tanto o foco e o objetivo desse semestre foram direcionados para adaptação e inclusão de acessibilidade, dando as pessoas com deficiência mais oportunidades dentro das empresas. Em cima do redesign também foi realizado a solução da avaliação heurística e teste de usabilidade com base nos resultados obtidos.

O sistema até o momento está remodelado para pessoas que possuem deficiência visual como baixa visão e daltonismo, para elas foram alterados os botões para ícones e adicionados pop-up com a descrição da ação. As fontes foram ampliadas para melhor visualização. E para a auditiva foi implementado libras.

O recurso utilizado para a tradução em libras no sistema é o VLibras que é o resultado de uma parceria entre o Ministério do Planejamento, Desenvolvimento e Gestão (MP), por meio da Secretaria de Tecnologia da Informação (STI) e a Universidade Federal da Paraíba (UFPB), a Suíte VLibras consiste em um conjunto de ferramentas computacionais de código aberto, responsável por traduzir conteúdos digitais (texto, áudio e vídeo) para a Língua Brasileira de Sinais - LIBRAS, tornando computadores, dispositivos móveis e plataformas Web acessíveis para pessoas com deficiência auditiva.

3.3 Avaliação Heurística

A capacidade heurística é uma característica humana para descobrir ou resolver problemas a partir da experiência prática, da observação e da criatividade. Por isso, para realizar uma boa avaliação heurística, é necessário um pequeno grupo de avaliadores com experiências sólidas em usabilidade, para examinar a interface e avaliar a sua conformidade com a lista de princípios de usabilidade escolhida previamente. Existem diversas listas de heurísticas definidas por diferentes especialistas, e em geral elas descrevem regras de propriedades comuns em interfaces com boa usabilidade.

Heurísticas: diretrizes de usabilidade que descrevem características desejáveis da interação e da interface. Avaliadores inspecionam sistematicamente a interface em busca de problemas que prejudiquem a usabilidade. Alternativa rápida e de baixo custo comparada a métodos com usuários. Método criado para encontrar problemas de usabilidade durante um processo de design iterativo.

O projeto contou com o planejamento e a realização da Avaliação Heurística e o Teste de Usabilidade.

4. Engenharia de Software do Sistema

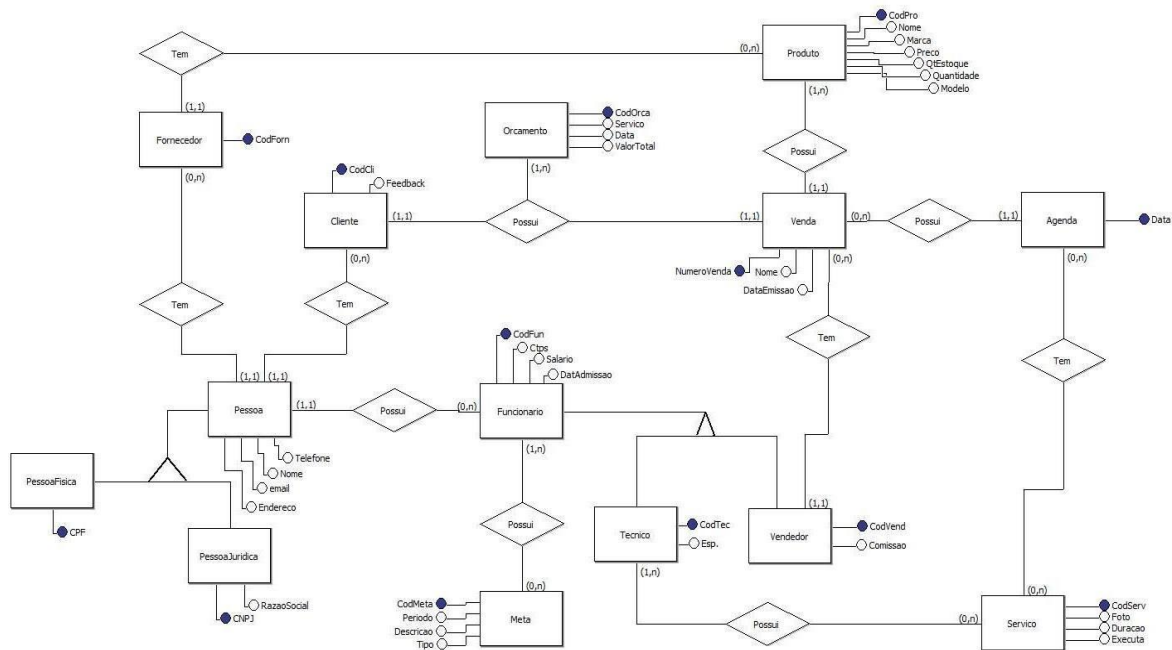
4.1 Requisitos

Após a aplicação da metodologia proposta, foram desenvolvidos os primeiros artefatos do projeto, os quais foram os requisitos funcionais e não-funcionais, agora fizemos um novo levantamento com requisitos funcionais adicionais e não funcionais (apresentados como duas tabelas), além das dependências entre eles (apresentado como um diagrama).

4.2 Diagramas UML

Para demonstrar as classes dentro do banco de dados utilizados no sistema, foram desenvolvidos o diagrama de classes e o diagrama conceitual, onde este segundo pode ser observado no Diagrama 1.

Diagrama 1: Conceitual



Fonte: o autor.

5. Desenvolvimento com Django

O Django possui uma interface de administração embutida onde é possível as informações armazenadas pelo modelo, para sua utilização é necessário adicionar as pastas para sua administração – “security/app1/admin.py”.

O modelo é definido pelo comportamento e campos chamada classes, ele é necessário para armazenagem no banco de dados onde são mapeadas como uma tabela – “security/app1/models.py”.

O Settings é um arquivo que contém toda a configuração do projeto, timezones, registro de apps, configurações da base de dados e etc. –

“security/security/settings.py”

Serializers permitem que dados complexos, como querysets e model instances, sejam convertidos em tipos de dados Python nativo que são renderizados em JSON,

XML etc.– “security/security/serializers.py”

Uma função View, é uma função Python que recebe uma solicitação da Web e retorna uma resposta da Web. Essa resposta pode ser o conteúdo HTML de uma página da Web, um redirecionamento, um erro 404, um documento XML ou uma imagem. – “security/app1/views.py”

As URLs consistem em um mapeamento que cuida de direcionar cada requisição recebida pelo servidor HTTP para uma determinada visão dentro das aplicações do projeto. Elas podem ser configuradas dentro do arquivo “./app1/urls.py”

Os arquivos que não precisam ser interpretados pelo Django, ou seja, imagens, arquivos de estilo CSS, programas em JavaScript etc., ficam por padrão no diretório “static” - “./app1/static//static/app2”.

O Django tem um componente próprio para a criação de formulários que se encarrega da tarefa de montá-los, tratá-los e validá-los – “./app1/forms.py”.

Sendo um framework para aplicações web o Django necessita de um jeito fácil para criar dinamicamente arquivos HTML. E a forma mais prática de fazê-lo é através do uso de templates contendo código HTML.

Foi utilizado bibliotecas adicionais como:

“libgravar”: é específica para acessar e recuperar informações da API do Gravatar.

“django-widget-tweaks”: é uma extensão que permite alteração de atributos dos campos HTML, que são gerados automaticamente no HTML.

Foi utilizado framework adicionais como:

“djangorestframework”: O Django Rest Framework é uma toolkit poderosa e flexível para criar Web APIs ou Rest APIs. Com ele conseguimos implementar Rest APIs de forma muito rápida.

6. Apresentação e discussão dos resultados

6.1 TELA DE LOGIN

Ao iniciar o sistema é necessário realizar o login para que ocorra a autenticação sobre as funções que o usuário pode realizar, é necessário usuário e senha.

Figura 02: Tela de Login



Fonte: o autor.

Tabela 06: Usuários.

<i>TIPO DE USUÁRIO</i>	<i>PERMIÇÃO</i>	<i>SEM ACESSO</i>
Gerente	Total	---
Comercial	Parcial	Cadastrar, editar e remover os produtos, funcionários e fornecedores
Técnico	Parcial	Cadastrar, editar e remover os produtos, funcionários e fornecedores

Fonte: o autor.

6.2 Tela Cliente

Figura 03: Tela Cliente.

Na tela do cliente, podemos realizar o cadastro de novos clientes e verificar os clientes já cadastrados conforme a Figura 03. Foi retirada está tela no atual Security.

Fonte: o autor.

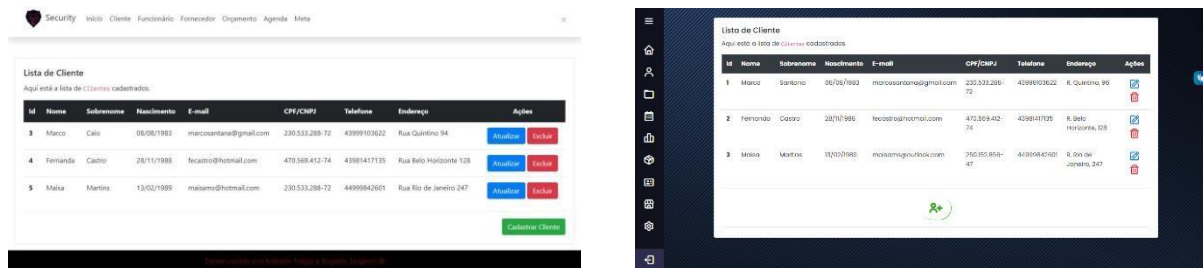
Ao realizar um novo cadastro de cliente temos os campos obrigatórios como na Figura 04: nome, sobrenome, nascimento, e-mail, CPF/CNPJ, telefone e endereço. Caso esteja algum campo incorreto ou faltando ocorre uma validação, se tiver algo errado irá aparecer uma notificação para que se possa ser corrigido antes de finalizar o cadastro.

Figura 04: Cadastro de cliente.

Fonte: o autor.

Na lista de clientes, podemos conferir todos os clientes e seus dados, podendo atualizar ou excluir qualquer cliente, conforme a Figura 05.

Figura 05: Lista de clientes.



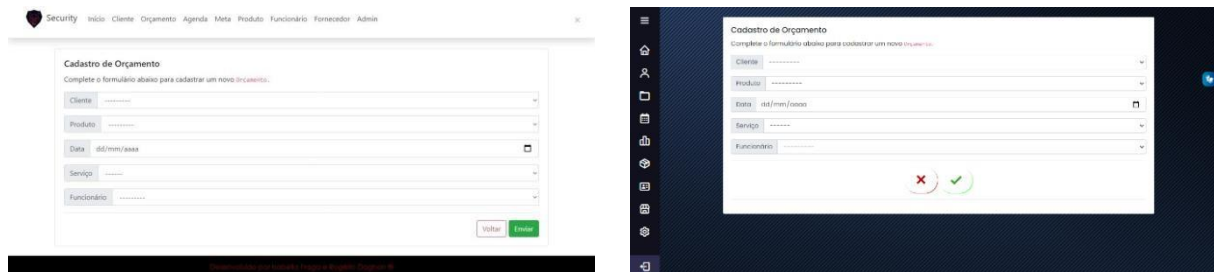
Fonte: o autor.

6.3 Tela Orçamento

Na tela do orçamento, podemos realizar o cadastro de novos orçamentos e verificar orçamentos cadastrados.

Ao realizar um novo cadastro de orçamento temos os campos obrigatórios para cadastro como, cliente, produto, data, serviço e funcionário. Caso esteja algum campo incorreto ou faltando ocorre uma validação, se tiver algo errado irá aparecer uma notificação para que possa ser corrigido antes de finalizar o cadastro. Conforme a Figura 07.

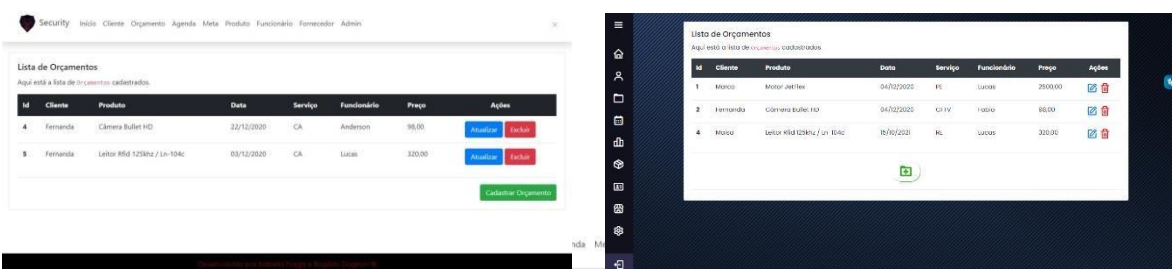
Figura 07: Cadastro de Orçamento.



Fonte: o autor.

Na lista de orçamento, também podemos conferir todos os orçamentos cadastrados e seus dados, podendo atualizar e excluir.

Figura 08: Lista de Orçamento.



Cadastrar Cliente
Cadastre aqui um novo cliente.

[Novo Cliente](#)

Lista de Cliente
Veja aqui a lista de clientes cadastrados.

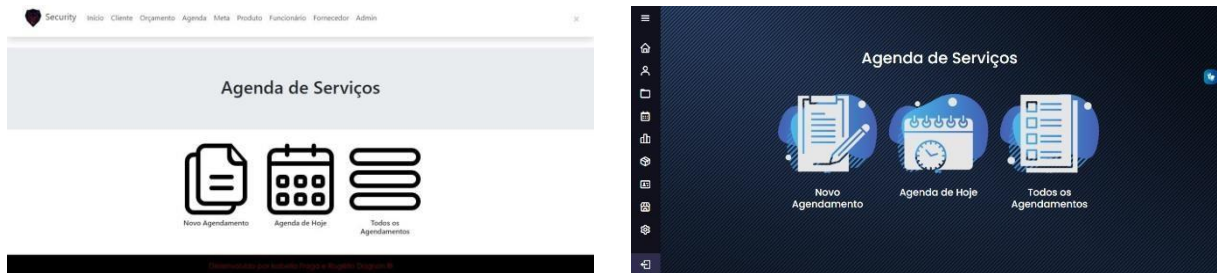
[Vá para Lista](#)

Fonte: o autor.

6.4 Tela Agenda

Na tela da agenda, podemos realizar o cadastro de novos agendamentos, verificar agendamento do dia e verificar todos os agendamentos, conforme a Figura 09.

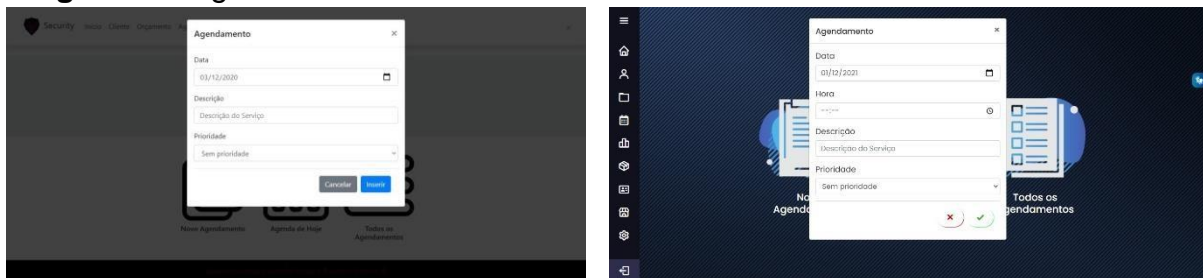
Figura 09: Tela Agenda.



Fonte: o autor.

Para realizar um novo agendamento, será necessário inserir os dados, data, descrição e prioridade, conforme a Figura 10.

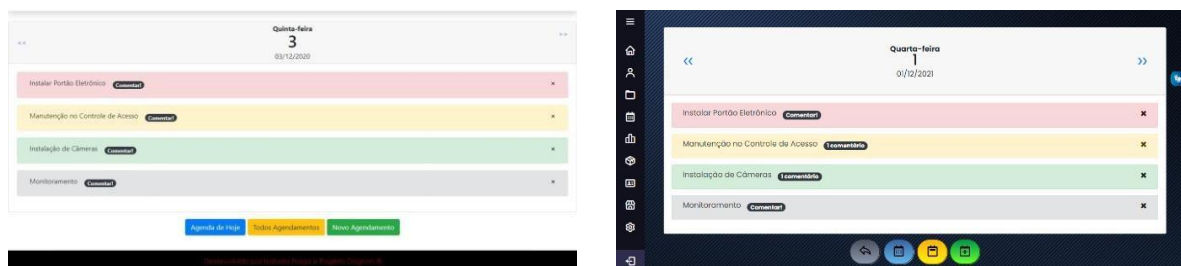
Figura 10: Agendamento.



Fonte: o autor.

Na agenda de hoje pode ser verificado os agendamentos para o dia específico, conforme a Figura 11.

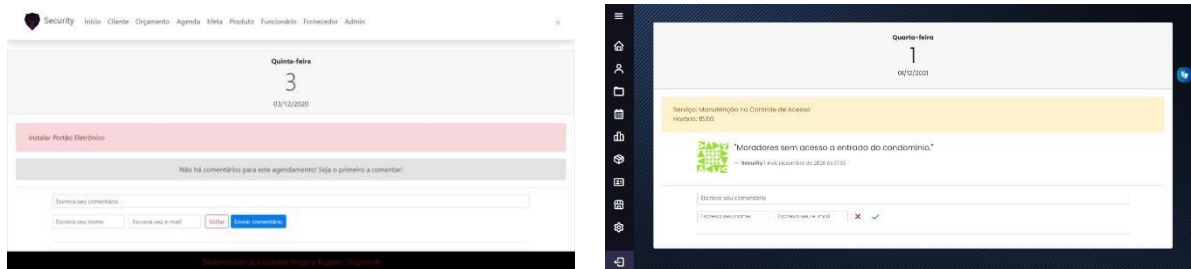
Figura 11: Agenda de Hoje.



Fonte: o autor.

Na área de comentário, podemos fixar um comentário em cada item agendado caso necessário, conforme a Figura 12.

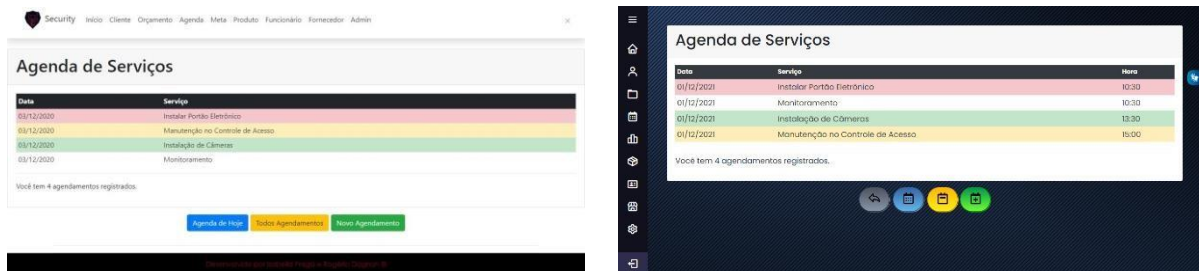
Figura 12: Comentário no Agendamento.



Fonte: o autor.

Na área de todos os agendamentos, podemos verificar todos os agendamentos realizados e quantos agendamentos temos registrado, conforme a Figura 13.

Figura 13: Todos os Agendamentos.

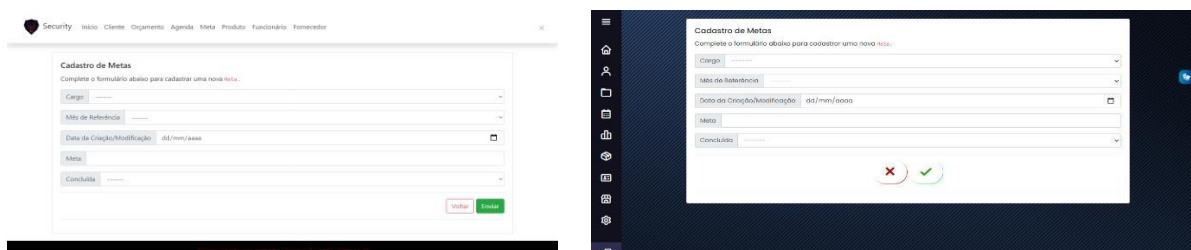


Fonte: o autor.

6.5 Tela Meta

Na tela das metas, podemos realizar o cadastro de novas metas, verificar metas cadastradas. Ao realizar um novo cadastro de meta temos os campos obrigatórios para cadastro como, cargo, mês de referência, data da criação/modificação, meta e concluída. Caso esteja algum campo incorreto ou faltando ocorre uma notificação para que possa ser corrigido antes de finalizar o cadastro.

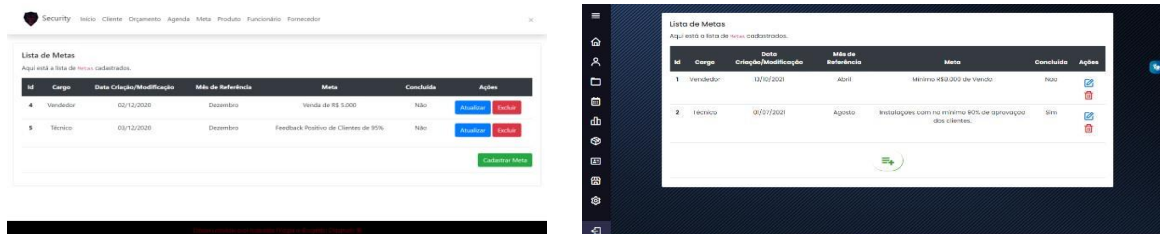
Figura 14: Cadastro Meta.



Fonte: o autor.

Na lista de metas, podemos conferir todas as metas e seus dados, podendo atualizar e excluir.

Figura 15: Lista Meta.



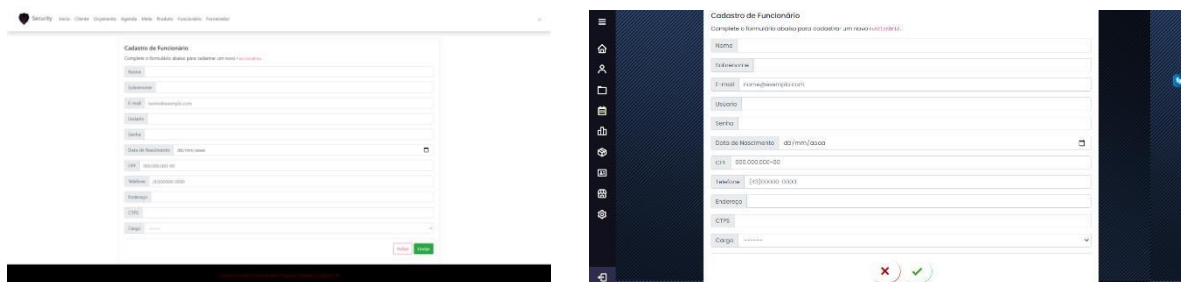
Fonte: o autor.

6.6 Tela Funcionário

Na tela do funcionário, podemos realizar o cadastro de novos clientes e verificar os clientes já cadastrados.

Ao realizar um novo cadastro de cliente temos os campos obrigatórios para cadastro, conforme a Figura 16, nome, sobrenome, e-mail, usuário, senha, data de nascimento, CPF, telefone, endereço, CTPS e cargo. Caso esteja algum campo incorreto ou faltando ocorre uma notificação para que possa ser corrigido antes de finalizar o cadastro.

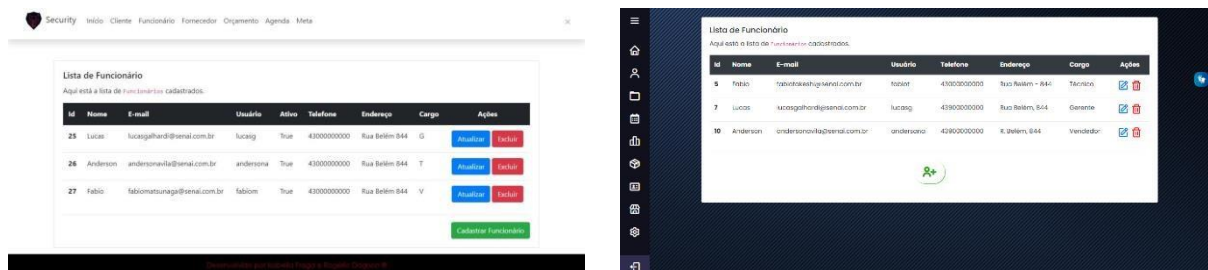
Figura 16: Cadastro Funcionário.



Fonte: o autor.

Na lista de funcionários, podemos conferir todos os funcionários e seus dados, podendo atualizar, excluir e bloquear, conforme a Figura 17.

Figura 17: Lista de Funcionário.

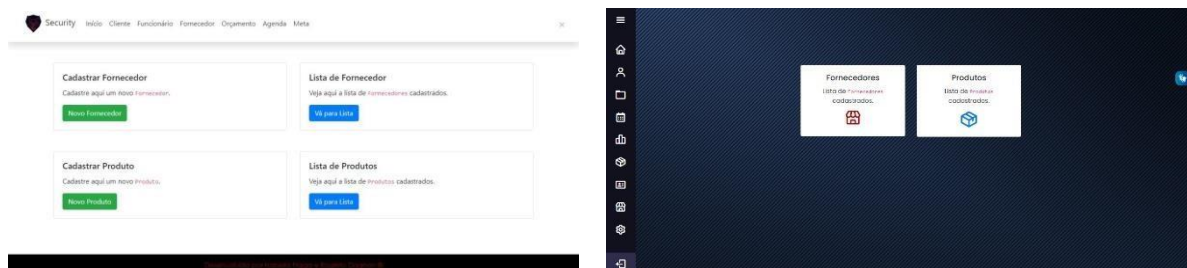


Fonte: o autor.

6.7 Tela Fornecedor

Na tela do fornecedor, podemos realizar o cadastro de novos fornecedores, verificar fornecedores cadastrados, cadastrar produtos e verificar produtos cadastrados, conforme a Figura 17.

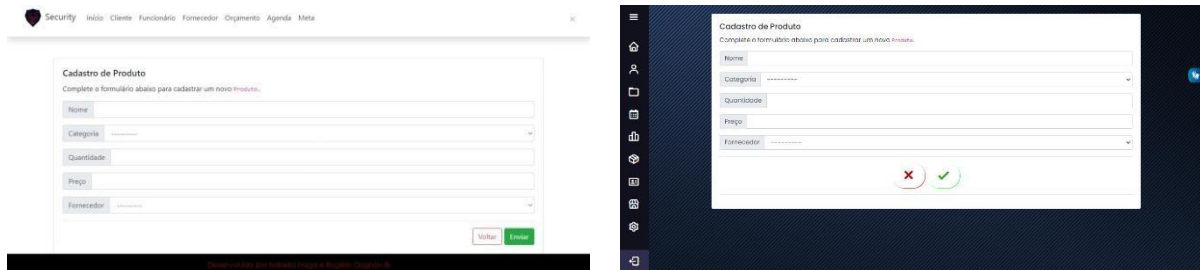
Figura 17: Tela de Fornecedor.



Fonte: o autor.

Ao realizar um novo cadastro de fornecedor, conforme a Figura 18, temos os campos obrigatórios para cadastro como, nome, e-mail, telefone, endereço, CNPJ e categoria. Para um novo cadastro de produto, também temos os campos obrigatórios de cadastro como, nome, categoria, quantidade, preço e fornecedor. Caso esteja algum campo incorreto ou faltando ocorre uma notificação para que possa ser corrigido antes de finalizar o cadastro tanto para os fornecedores quanto para os produtos.

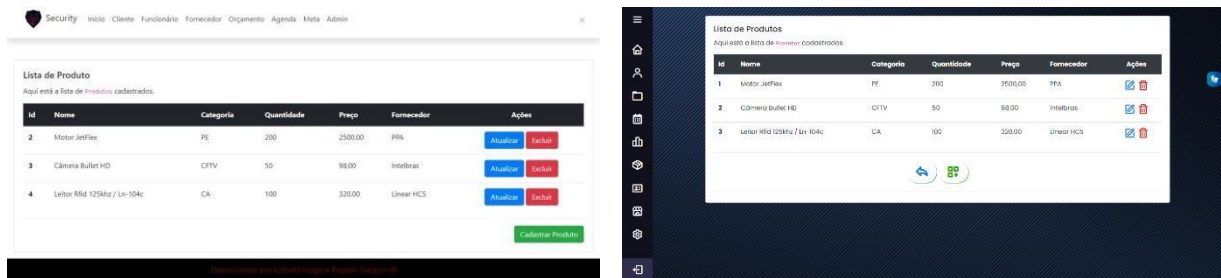
Figura 18: Cadastro de Produto.



Fonte: o autor.

Na lista de produtos, também podemos conferir todos os fornecedores e seus dados, podendo atualizar e excluir.

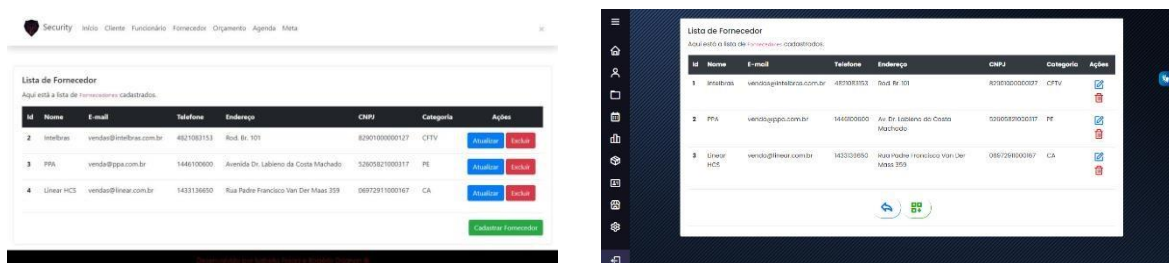
Figura 19: Lista de Produto.



Fonte: o autor.

Na lista de fornecedores, podemos conferir todos os fornecedores e seus dados, podendo atualizar e excluir.

Figura 20: Lista de Fornecedor.



Fonte: o autor.

7. APLICATIVO MOBILE

O aplicativo mobile desenvolvido para este projeto tem o objetivo de estender o sistema web auxiliando na interação do usuário. Ele foi desenvolvido em React Native que é uma biblioteca Javascript para construir interfaces de usuários, que fornece um conjunto de plataformas e componentes nativos agnósticos como View, Text e Image além de interagir com APIs nativas por meio do paradigma de IU declarativo do React e do Javascript.

Para realizar a conexão do aplicativo mobile com o sistema web, utilizamos a API Fetch do próprio React Native para atender todas as necessidades de rede, com ela conseguimos acessar e manipular partes do HTTP, buscando os arquivos em JSON para apresentar os dados direto do banco no aplicativo.

Com a API Fetch também trabalhamos as solicitações e requisições POST, GET, DELETE e PATCH, para atuarem na parte das anotações, armazenando assim no banco de dados do sistema onde não teremos problemas de percas ou falhas como ocasionado se fosse salvo localmente no próprio aplicativo.

8. CONSIDERAÇÕES FINAIS

Através desse trabalho conseguimos dar continuidade ao sistema web, adicionando mais requisitos e interações, passando assim a conhecer um pouco mais sobre a interação do aplicativo mobile com o software e a facilidade gerada ao usuário em sua utilização. Tornando possível o gerenciamento de diversos setores para as empresas de segurança.

REFERÊNCIAS

SAUDATE, Alexandre. **A. Rest – Construa API’S inteligentes de maneira simples**. 1. Ed. São Paulo: Casa do código, 2013. 314 p.

FACEBOOK OPEN SOURCE. Disponível em: <<https://reactnative.dev>>. Acessado em: 20 ago. 2021.

MOZILLA AND INDIVIDUAL CONTRIBUTORS. Disponível em:

<<https://developer.mozilla.org/en-US/>> Acessado em: 22 ago. 2021.

STREPHONSAYS - Diferença entre Perl e Python, Disponível em:

<<https://pt.strephonsays.com/perl-and-vs-python-9014>> Acessado em: 11 nov 2021.

CETAX - Python: Comparando Python com outras linguagens, Disponível em:

<<https://www.cetax.com.br/blog/comparando-python-com-outras-linguagens/>>
Acessado em: 11 nov 2021.

STREPHONSAYS - Diferença entre JavaScript e TypeScript, Disponível em:
<<https://pt.strephonsays.com/javascript-and-vs-typescript-13697>> Acessado em: 11 nov 2021.

Redação Impacta - Typescript x JavaScript: entenda o que é e saiba as diferenças, Disponível em: <<https://www.impacta.com.br/blog/typescript-javascript-entenda-quesaiba-diferencas>> Acessado em: 11 nov 2021.

Ipea – Objetivos do Desenvolvimento Sustentável, Disponível em:

<<https://www.ipea.gov.br/ods/index.html>>, Acessado em: 15 nov. 2021.

Estratégia ODS – O QUE SÃO OS ODS?, Disponível em:

<<https://www.estrategiaods.org.br/conheca-os-ods/>>, Acessado em: 15 nov. 2021.

Polen Portal Crédito de Logística Reversa – Objetivos de Desenvolvimento

Sustentável (ODS) da ONU: tudo o que você precisa saber, Disponível em:

<<https://www.creditodelogisticareversa.com.br/post/t-objetivos-de-desenvolvimento-sustentavel-ods-da-onu-tudo-o-que-voce-precisa-saber>>, Acessado em: 15 nov. 2021.

MERGO - Avaliação Heurística na análise de interfaces, Disponível em:
<<https://uxdesign.blog.br/avalia%C3%A7%C3%A3o-heur%C3%ADstica-naan%C3%A1lise-de-interfaces-218c2dd46164>>, Acessado em: 2 dez. 2021.

WayCarbon. Conheça os 17 objetivos do desenvolvimento sustentável da ONU, Disponível em: <<https://blog.waycarbon.com/2015/10/conheca-os-17-objetivos-dodesenvolvimento-sustentavel-da-onu/>>, Acessado em: 15 nov. 2021.

EQUIPE CORE. Bootstrap, Disponível em: <<https://getbootstrap.com.br/>>. Acessado em: 30 set. 2020.

DJANGO SOFTWARE FOUNDATION. DjangoProject, Disponível em:

<<https://www.djangoproject.com/>>. Acessado em: 10 nov. 2020

W3.CSS. W3schools, Disponível em: <<https://www.w3schools.com/>>. Acessado em: 20 out. 2020.

PYTHON ACADEMY. Desenvolvimento Web com Python e Django, Disponível em: <<https://pythonacademy.com.br/blog/desenvolvimento-web-com-python-e-djangotemplate/>>. Acessado em: 30 nov. 2020.

GIOVANNI NUNES. Exemplo em Django Autenticação, Disponível em:

<<https://giovannireisnunes.wordpress.com/2019/09/27/exemplo-em-django-autenticacao/>>. Acessado em: 28 nov. 2020.

ZKoder. Django Rest Framework - MySQL CRUD RESTful APIs, Disponível em <<https://bezkoder.com/django-crud-mysql-rest-framework/>>. Acessado em: 20 nov. 2020.

Capítulo II – NOBLEWARE: Estimativas de projeto de software

André Nogueira da Costa

Fernando Hideki Watanabe

Pedro Andrade Galdana

Anderson Paulo Ávila Santos⁴

Lucas Busatta Galhardi⁵

Fábio Takeshi Matsunaga⁶

RESUMO

Este artigo tem como objetivo expor os resultados obtidos durante o desenvolvimento do projeto proposto na Jornada de Aprendizagem, que consiste em construir um sistema como forma de solução de um problema real do mercado de softwares. Neste contexto, tratou-se dos equívocos e dificuldades encontradas nas estimativas de orçamentos de softwares realizadas por equipes de consultoria, obtendo como solução um sistema que por meio de uma Inteligência Artificial, tendo como base o conhecimento prévio dos participantes, realizasse uma estimativa de horas e custos para a construção de um software, voltado para os usuários que não possuem fácil acesso à uma consultoria presencial ou para empresas interessadas na ferramenta.

Palavras-chave: Software. Consultoria. Estimativa. Inteligência Artificial.

NOBLEWARE: Software Project Estimate

ABSTRACT

This article aims to expose the results obtained during the development of the project proposed in the Learning Journey, which consists of building a system as a way to solve a real problem in the software market. In this context, the mistakes and difficulties found in the estimates of software budgets carried out by consulting teams were addressed, obtaining as a solution a system that, through an Artificial Intelligence, based on the prior knowledge of the participants, carried out an estimate of hours and costs for the construction of a software, aimed at users who do not have easy access to a face-to-face consultancy or for companies interested in the tool.

Keywords: Software. Consultancy. Estimate. Artificial Intelligence.

⁴ Mestre em Ciência da Computação. E-mail: anderson.avila@sistemafiep.org.br

⁵ Mestre em Ciência da Computação. E-mail: lucas.galhardi@sistemafiep.org.br

⁶ Mestre em Ciência da Computação. E-mail: fabio.matsunaga@sistemafiep.org.br

1. INTRODUÇÃO

O Processo de desenvolvimento de Software é um processo de alta complexidade, pois são diversas fases realizadas durante todo o processo, e uma das principais fases é a fase de Levantamento de Requisitos. Durante esta fase, são realizadas a coleta dos requisitos necessários (Requisitos Funcionais, Requisitos Não Funcionais), e é justamente nesta fase que surgem muitos problemas relacionados ao processo de desenvolvimento do Software, sendo um dos problemas mais comuns, a falta dos requisitos, acarretando assim mais problemas de custos e entrega do projeto.

A equipe analisou estes problemas, e assim, desenvolveram uma ideia de uma possível solução, sendo este um sistema Web de consultoria para estimativas de custo e horas para determinado projeto.

Nossas principais justificativas para realização deste projeto são solucionar os problemas envolvendo o processo de desenvolvimento de Software e trazer uma maior praticidade, acessibilidade e facilitador para aqueles que não possuem conhecimento a respeito do processo (seja tanto no quesito informativo quanto no quesito custos e valores).

O projeto Nobleware contribui para um dos ODS, mais especificamente o objetivo **9 – Indústria, Inovação e Infraestrutura**, que se preocupa com a construção de infraestruturas confiáveis para promover a industrialização e estimular a inovação. De acordo com os tópicos presentes no objetivo, o nosso projeto se enquadra no propósito de elevar o acesso das pessoas às tecnologias.

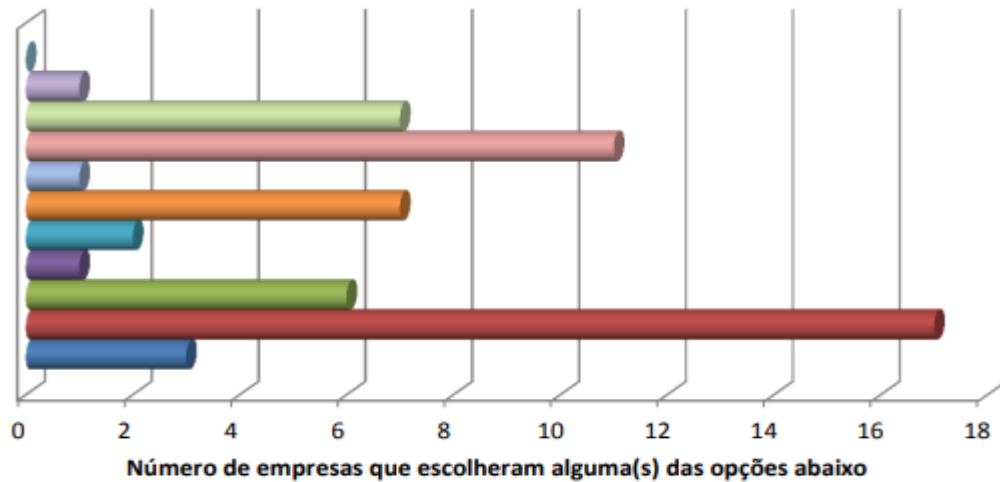
2. FUNDAMENTAÇÃO TEÓRICA

Durante nossas pesquisas realizadas, o site Whow! (2020) afirma que muitas empresas, durante o período da quarentena, tiveram a necessidade repentina de utilização de serviços e plataformas como solução para manter seus negócios ativos.

Pelas pesquisas realizadas por FIGUEIRA (2012), este aponta que muito dos desenvolvedores encontram dificuldades durante o processo de levantamento de requisitos como é indicado no gráfico abaixo:

Figura 1 – Dificuldades no levantamento de requisitos.

Principais dificuldades no levantamento de requisitos estão relacionadas:



- Outro
- Não enfrento dificuldades.
- Ao analista ter que exercer vários papéis durante a atividade.
- Às mudanças nos requisitos.
- À dificuldade em capturar informações no domínio.
- À dificuldade em validar o que o cliente solicitou.
- À análise das informações coletadas do usuário.
- Às técnicas e ferramentas existentes.
- À existência de muitos stakeholders envolvidos com interesses distintos.
- À dificuldade do cliente expressar o que realmente deseja.
- À dificuldade em se relacionar com o cliente.

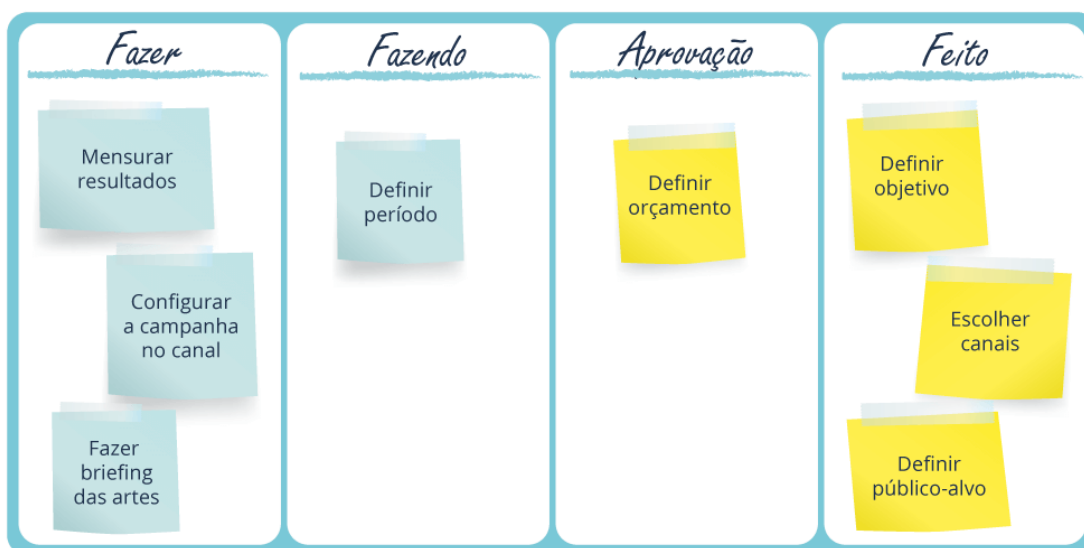
Fonte: Anderson Marques da Silva Figueira (2012)

3. METODOLOGIA

A metodologia aplicada para o desenvolvimento do trabalho foi o SCRUM, que consiste em definir backlogs — um conjunto de tarefas a serem realizadas. Além de dividir as etapas de desenvolvimento em ciclos, também chamados de Sprints, normalmente cada Sprint tem um tempo de duração, por exemplo, uma semana ou duas semanas. Para definir uma Sprint, existe uma fase de planejamento para selecionar as atividades que serão realizadas durante o período determinado, esta fase é chamada de Sprint Planning. Após a finalização da Sprint, é realizada uma reunião, para verificar se os objetivos da Sprint — Product Backlog — foram atingidos e um feedback da equipe sobre o andamento do projeto, isto é chamado de Sprint Retrospective.

Para controlar os fluxos de produção é utilizado o Kanban, que é uma metodologia de gerenciamento ágil, que conta com a ajuda de um painel que organiza o progresso do projeto, dividindo-o em etapas: “para fazer”, “em andamento”, “aprovação” e “concluído”. Assim é possível ter uma visão geral de como está o andamento do projeto. Atualmente, existem algumas ferramentas que auxiliam na criação do Kanban, com uma interface digital por meio de um software. A ferramenta escolhida para esta função foi o Trello. Nela é possível criar o planejamento das ações do projeto, determinar prazos e atualizar o andamento de cada tarefa.

Figura 2 – Exemplo tabela Kanban



Fonte: Roberto Gil Espinha, 2019.

Após a definição da metodologia de desenvolvimento, foi iniciado o processo de modelagem do sistema, por meio dos diagramas UML, que servem para determinar os fluxos e as funções que o sistema irá conter. Nesse projeto foram utilizados os diagramas de caso de uso e diagrama de classe, que são a base para começar o desenvolvimento do software. Sendo assim, foi iniciada a construção do sistema por meio de códigos, utilizando as tecnologias escolhidas pela equipe, que no caso foi o ReactJS (JavaScripts) e o Django (Python). Para salvar as mudanças no código, foi utilizado o GitHub, cuja sua função é controlar o que cada integrante da equipe alterou em determinada parte do software, possibilitando o versionamento de código.

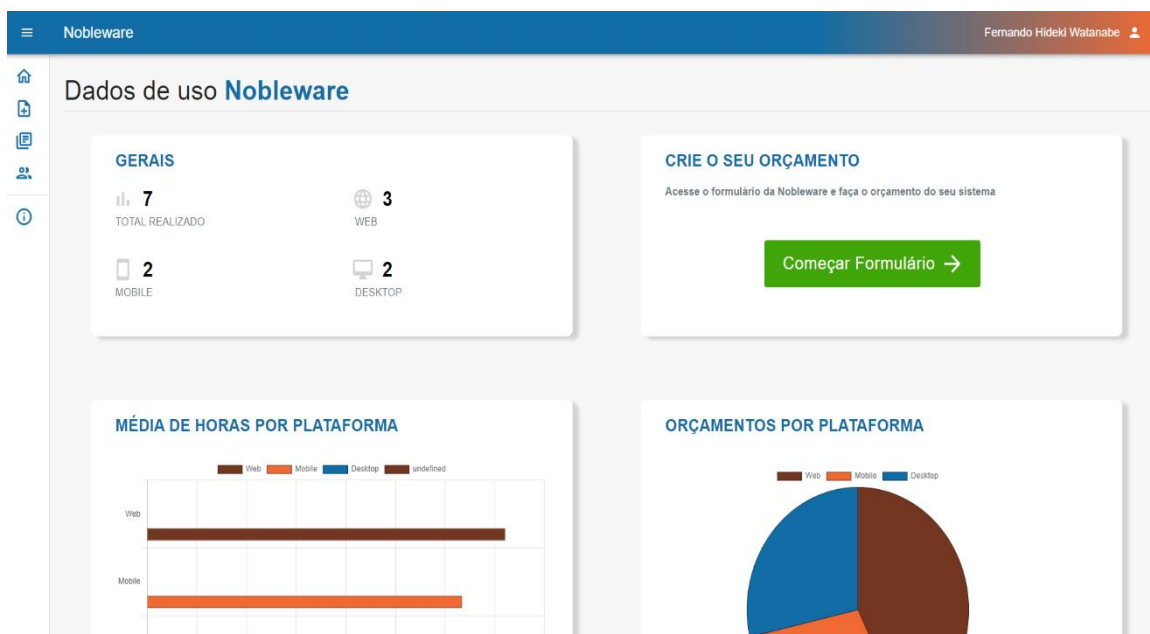
Para o desenvolvimento da camada de back-end (Django), foi utilizado a arquitetura API Rest, que é um conjunto de restrições, onde as requisições feitas pela camada client-side, tenham um padrão de comunicação com a camada server-side. E para o desenvolvimento da camada front-end (ReactJS), foi utilizado o padrão de reutilização de componentes, para organizar e facilitar a construção das interfaces.

4. DESENVOLVIMENTO E APRESENTAÇÃO DO SISTEMA

Com a modelagem do sistema realizada, a equipe decidiu definir para o front-end, o modelo arquitetural com base na reutilização de componentes, como opção para o desenvolvimento ágil e organizado da interface do sistema. E para o back-end, a equipe decidiu utilizar o modelo de API Rest, visando estabelecer um padrão de comunicação com a camada de client-side, possibilitando a identificação do resultado de cada requisição através dos códigos de status de resposta HTTP.

Após a aplicação dos modelos arquiteturais, o desenvolvimento da solução como produto foi iniciado com o objetivo de construir as telas “Home”, “Questionário” e “Orçamento”.

Figura 3 – Tela Home.



Fonte: Elaborado pelo autor (2021).

A figura 3 representa a tela inicial da aplicação, onde é exibido um dashboard contando informações e estatísticas com base nos orçamentos gerados no histórico do sistema.

Figura 4 – Tela de Questionário.

The questionnaire screen includes a progress bar with four stages: "Conhecendo o sistema", "Requisitos", "Detalhes", and "Orçamento". The current stage is "Requisitos".

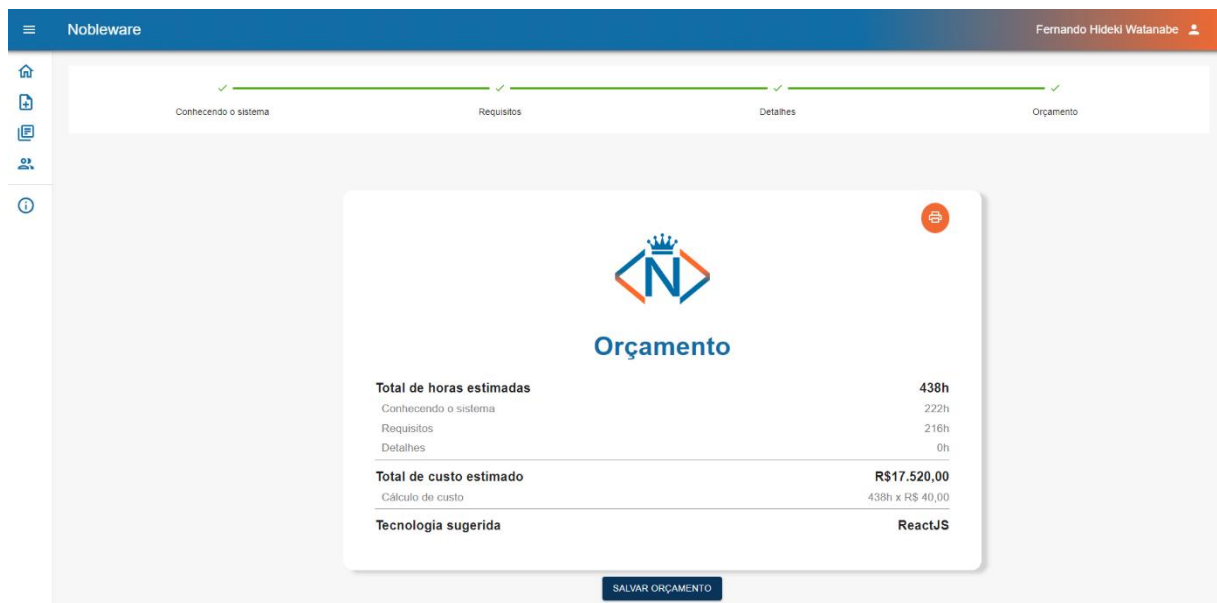
Questions and options:

- Qual a plataforma de desenvolvimento?
 - Web
 - Mobile
 - Desktop
- Deverá rodar em quais navegadores?
 - Opera
 - Mozilla
 - Chrome
 - Internet Edge
- Deverá rodar em quais sistemas operacionais?
 - Android
 - IOS (Apple)
- Deverá ter acesso a internet para utilizar o sistema?
 - Sim
 - Não

Fonte: Elaborado pelo autor (2021).

A figura 4 representa a tela de questionário, uma das principais funcionalidades da aplicação, contendo as questões de múltipla escolha com o objetivo de colher os interesses do usuário em relação ao software que ele deseja. As questões foram divididas em etapas, que simulam as fases de uma entrevista de consultoria com o cliente. Ao final do questionário, as respostas são enviadas para a camada server-side, onde são processadas pela Inteligência Artificial (IA), que por meio de um conjunto de dados de estimativas, executa a previsão de tempo de desenvolvimento através de um modelo treinado. O resultado obtido na atuação da IA, é exibido para o usuário na tela de orçamento, conforme figura 5.

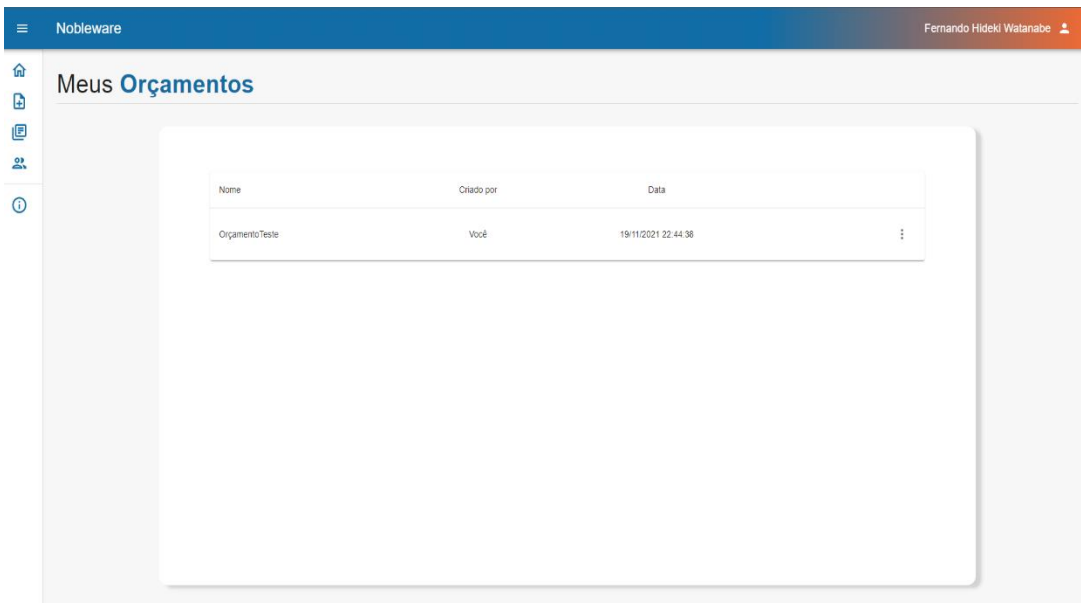
Figura 5 – Tela de Orçamento.



Fonte: Elaborado pelo autor (2021).

A figura 6 representa a tela de orçamentos realizados, contendo todos os orçamentos salvos pelo usuário. Cada orçamento é listado com botões de ação, dando as opções de visualizar e compartilhar o arquivo do orçamento.

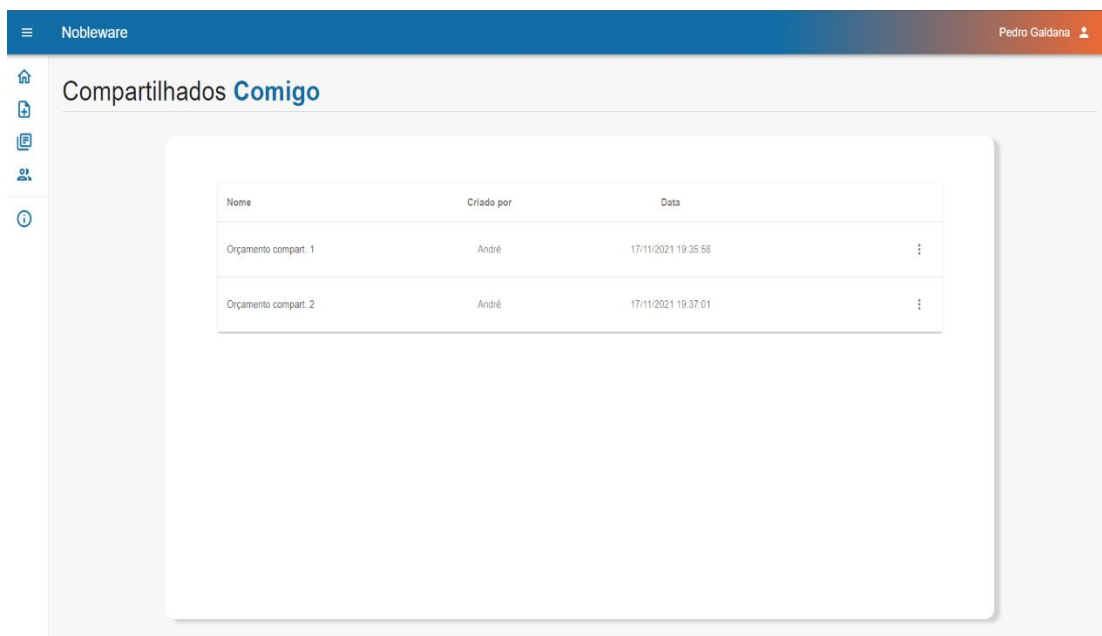
Figura 6 – Tela de Meus Orçamentos.



Fonte: Elaborado pelo autor (2021).

Na figura 7 é apresentada a Tela de Orçamentos Compartilhados Comigo, onde todos os orçamentos que foram compartilhados com aquele determinado usuário serão listados naquela tela, permitindo a visualização mais detalhada do orçamento compartilhado (Figura 5).

Figura 7 – Tela de Orçamentos Compartilhados Comigo



Fonte: Elaborado pelo autor (2021).

5. COMPARANDO LINGUAGENS DE PROGRAMAÇÃO

Em nosso projeto, utilizamos de algumas tecnologias disponíveis para o desenvolvimento do Sistema, sendo essas tecnologias utilizadas o Framework ReactJS, feito em JavaScript, para o desenvolvimento do Front-End, e o Framework Django, feito em Python, para o desenvolvimento do Back-End do projeto.

Analizamos alguns comparativos de algumas outras linguagens que são equiparadas as quais utilizamos para o desenvolvimento do nosso projeto, sendo essas linguagens, mais especificamente o C# e o PHP.

5.1 VANTAGENS

- **Python:** O Python, por se tratar de uma linguagem Open Source, ela é muito utilizada no mercado de Tecnologia num geral, e esta é de certa forma uma linguagem muito prática e com uma certa facilidade de se aprender pelo fato dela ser dinamicamente tipada, além de ser uma linguagem bem versátil com possibilidade de aplicação para diversas soluções. Essa linguagem é bem consolidada no mercado atual e possui suporte para diversas plataformas diferentes, então esta possui muitas soluções disponíveis tanto por parte da documentação original da linguagem quanto por parte da comunidade de desenvolvedores, o que a torna uma boa opção de uso para desenvolvimento geral.
- **C#:** A linguagem C# é uma linguagem que é fortemente tipada que segue o paradigma orientado à objetos, mas isto não significa que ela é difícil de se aprender ou algo do gênero, muito pelo contrário, caso a pessoa possua experiência com outras linguagens como Java, C++, ela não encontrará dificuldades. Assim como a linguagem Python, o C# também é uma linguagem muito consolidada no mercado de Tecnologia, sendo assim, também possui muitos recursos e soluções disponíveis. Essa linguagem, diferente da Python, possui suporte para diversas plataformas, podendo ser Web, Desktop e até mesmo mobile, o que também possibilita uma possibilidade grande de aplicabilidade em diversas soluções diferentes. No quesito desempenho, C# é uma linguagem compilada,

sendo assim, garante um pouco mais de velocidade e segurança na sua execução.

- **PHP:** Assim como as linguagens Python e C#, o PHP é uma linguagem open source, permitindo sua utilização para qualquer situação, pessoas e/ou empresas. Essa linguagem também possui um certo grau de facilidade, visto que pessoas que já tiveram contato com C, Java e afins também não encontraram grandes dificuldades em aprender essa linguagem, já que o PHP utiliza de nomenclaturas presentes nessas linguagens. A linguagem também é muito versátil, visto que ela é capaz de rodar em diversas plataformas como Linux, Windows, Mac OS etc. Além disso, essa linguagem

5.2 DESVANTAGENS

- **Python:** Uma das principais desvantagens que pode ser destacada nesta linguagem é em relação à sua performance. Pelo fato de a linguagem ser interpretada, isso significa que sua execução ocorre com o auxílio de um interpretador ao invés de um compilador, sendo executado linha por linha. Uma das desvantagens que também vale ser citada é a questão do design da linguagem, isso faz com que seja necessário realizar muitos mais testes no código pois os erros aparecem somente no tempo de execução do código. O Python também conta com uma limitação em relação a plataformas mobiles, onde é raramente usada e possui poucos recursos e soluções disponíveis.
- **C#:** Apesar do fato da linguagem C# ser compilada trazer um certo ganho de desempenho, existem algumas complicações que podem ser ocasionadas durante o desenvolvimento. Por exemplo, se caso seja necessário realizar alguma modificação no código, será necessário recompilar a aplicação inteira, o que pode acabar ocasionando um ganho no tempo de execução. A estrutura da linguagem C# é bem estrita se comparada com linguagem que utilizam de tipagem dinâmica, então muita das vezes alguma solução

feita em C# pode acabar se tornando muito maior e mais complexa que soluções realizadas em Python por exemplo.

- **PHP:** A linguagem PHP é um tipo de linguagem interpretada, que é executada pelo lado do servidor, portanto, a chance de algum invasor conseguir o acesso ao código fonte é relativamente alta. A linguagem PHP não possui uma determinada padronização, sendo assim, muitos recursos de versões anteriores das linguagens podem acabar se tornando obsoletos e até mesmo podem parar de funcionar conforme versões novas forem sendo implementadas, além disso, em relação à documentação da linguagem, a documentação dela é bastante incompleta e nem sempre todos os recursos estarão disponíveis na documentação oficial da linguagem, podendo encontrar facilmente recursos implementados que ainda não foram inseridos na documentação oficial.

6. APRESENTAÇÃO DOS RESULTADOS E CONSIDERAÇÕES FINAIS

A equipe conseguiu concluir com êxito a realização do desenvolvimento do Sistema, conseguindo realizar a implementação de todas as funcionalidades que estavam previstas além de conseguir realizar a implementação do algoritmo de Inteligência Artificial.

Além da realização do desenvolvimento do Sistema, a equipe realizou testes para garantir que as funcionalidades do sistema estão funcionando de acordo com as especificações que foram documentadas. Com o auxílio da ferramenta TestLink, a equipe pode ser capaz de realizar o controle dos testes que foram realizados sobre a aplicação, além de manter o controle de versões liberadas.

Foram realizados diversos testes sobre a aplicação, sendo os principais os testes funcionais, para garantir que o sistema esteja funcional, testes de usabilidade, para garantir que o sistema possuirá uma usabilidade relativamente simples e sem muitas complicações que possam atrapalhar a experiência de usuário, e alguns testes exploratórios gerais.

A plataforma também permite gerar o relatório da execução dos testes, que permite visualizar os cenários realizados, com o passo a passo, resultados esperados e resultados obtidos.

Concluiu-se que ao final do projeto que, por ser algo incomum no mercado de sistemas Web, a aplicação desenvolvida pela equipe possui um certo potencial de mercado, visto que até o presente momento não existe nenhum produto semelhante ao implementado pela equipe. Vale ressaltar também que, existem inúmeras pesquisas e artigos, como por exemplo, em “Machine learning approaches to estimating software development effort” (K. Srinivasan and D. Fisher, 1995), discutindo sobre esse assunto e até mesmo, Inteligências Artificiais treinadas para realizar o orçamento de um software, porém não encontramos nenhum sistema no âmbito comercial, que realizasse o que o nosso sistema faz.

Para planos futuros, a equipe planeja realizar a implementação de features novas, que irão complementar e melhorar o Sistema como um todo, além de continuar treinando a Inteligência Artificial implementada para que esta esteja funcionando da melhor maneira possível e esteja sempre atualizada.

REFERÊNCIAS

FIGUEIRA, S. M. A. ANÁLISE DAS TÉCNICAS DE LEVANTAMENTO DE REQUISITOS PARA DESENVOLVIMENTO DE SOFTWARE NAS EMPRESAS DE VITÓRIA DA CONQUISTA – BA. 74f. Monografia (Graduação em Engenharia de Software). Universidade Estadual do Sudoeste da Bahia, Bahia, 2012. Disponível em: < <http://www2.uesb.br/computacao/wp-content/uploads/2014/09/ANÁLISE-DAS-TÉCNICAS-DE-LEVANTAMENTO-DE-REQUISITOS-PARA-DESENVOLVIMENTO-DE-SOFTWARE-NAS-EMPRESAS-DE-VITÓRIA-DA-CONQUISTA---BA.pdf>> Acessado em: 31 mai. 2021.

FONSECA, A. Aumento na demanda por desenvolvedores na quarentena. Disponível em: <<https://www.whow.com.br/tecnologia/aumento-demanda-desenvolvedores-quarentena/>>. Acesso em: 27 mai. 2021.

NETO, N. A. J. C# vs. Python. Disponível em: <https://linkedin.com/pulse/c-vs-python-jos%C3%A9-antonio-das-neves-neto/?trk=public_profile_article_view>. Acesso em: 11 out. 2021.

Stack Overflow Developer Survey 2021. Disponível em:
<<https://insights.stackoverflow.com/survey/2021>>.
Acesso em: 30 out. 2021.

FERREIRA, K. O que é PHP e por que você precisa conhecer essa linguagem de programação web. Disponível em: <<https://rockcontent.com/br/blog/o-que-e-php/>>.
Acesso em: 30 out. 2021.

Um tour pelo C#. Disponível em: <<https://docs.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/>>.
Acesso em: 30 out. 2021.

PHP: Documentation. Disponível em: <<https://www.php.net/docs.php>>.
Acesso em: 30 out. 2021.

ROVEDA, U. O que é Python, para que serve e por que aprender. Disponível em:
<<https://kenzie.com.br/blog/o-que-e-python/>>.
Acesso em: 30. out. 2021.

O tutorial de Python. Disponível em: <<https://docs.python.org/pt-br/3.10/tutorial/index.html>>.
Acesso em: 30. out. 2021.

K. Srinivasan and D. Fisher, "Machine learning approaches to estimating software development effort," in IEEE Transactions on Software Engineering, vol. 21, no. 2, pp. 126-137, Feb. 1995, doi: 10.1109/32.345828.

Capítulo III – Uma solução *web* e *mobile* para predição e análise de sentimentos sobre o avanço da Covid-19

Alexandre Marcos de Souza Junior⁷

Rubens Fujita Hirata⁸

Anderson Paulo Ávila Santos⁹

Fabio Takeshi Matsunaga¹⁰

Lucas Busatta Galhardi¹¹

RESUMO

A comunidade acadêmica tem se mobilizado para buscar soluções para realizar o monitoramento dos avanços da pandemia da Covid-19. Diante do exposto, o objetivo do presente projeto é realizar a projeção do avanço da Covid-19 de modo que os dados sejam acessíveis pela população por meio de uma aplicação *web* e *mobile*, a partir dos quais serão disponibilizados gráficos sobre dados oficiais. Estes dados serão processados por meio de regressão linear, para realizar estimativas sobre avanços. Além disso, será utilizada uma biblioteca em linguagem Python para realizar a análise de sentimentos da população sobre a pandemia, cujo dados serão buscados em postagens do Twitter. De acordo com os resultados, o número de óbitos representa em torno de 2% dos casos totais, com uma faixa etária entre 41 e 59 anos, sendo a maioria do sexo feminino.

Palavras-chave: API, *back-end*, *front-end*, Inteligência Artificial, *Machine Learning*.

A web and mobile solution for predicting and analyzing sentiments about the advance of Covid-19

ABSTRACT

The academic community has mobilized to seek solutions to monitor the advances of the Covid-19 pandemic. The main objective of this project is to carry out the projection of the advance of Covid-19 so that the data are accessible by the population. For this, a web and mobile application will be developed to provide graphics on official data.

⁷ Graduando em Engenharia de Software.

⁸ Graduando em Engenharia de Software.

⁹ MSc. Anderson Paulo Ávila Santos. E-mail: anderson.avila@sistemafiep.org.br

¹⁰ MSc. Fabio Takeshi Matsunaga. E-mail: fabio.matsunaga@sistemafiep.org.br

¹¹ MSc. Lucas Busatta Galhardi. E-mail: lucas.galhardi@sistemafiep.org.br

These data will be processed through linear regression, to make estimates of disease progress. In addition, the data will be processed through a library in Python language to carry out the analysis of the population's feelings about the pandemic, whose data will be searched on Twitter posts. According to the results, the number of deaths represents around 2% of the total cases, with an age group between 41 and 59 years, the majority being female.

Key-words: API, Artificial Inteligente, back-end, front-end, Machine Learning.

1. INTRODUÇÃO

O Brasil sofre em 2020 com a chegada de uma pandemia originada na China. Esta pandemia é efeito de uma doença causada pelo coronavírus SARS-CoV-2, que apresenta um quadro clínico que varia de infecções assintomáticas a quadros respiratórios graves (MINISTÉRIO DA SAÚDE, 2020a).

O evento abalou fortemente a sociedade brasileira, tanto em aspectos econômicos, como o fechamento de 1,1 milhão de vagas de emprego e retração de 5,89% do PIB do mercado brasileiro (G1, 2020), mas também sociais, com o óbito de mais de 131.274 falecidos e 4.315.858 pessoas infectadas em 13/09/2020, segundo Ministério da Saúde (2020b).

Diante desta tragédia no cenário mundial, a comunidade acadêmica e científica tem realizado diversas pesquisas e publicações sobre o assunto. No trabalho de Neves e Barros (2020), é utiliza um modelo matemático não linear proposto por Gompertz para verificar se as medidas impostas pelos governantes estão tendo resultados, já que este tipo de análise é aplicado em estudos populacionais interpretativos, levando a deduzir a velocidade e a aceleração dos casos.

Já Fernandes (2020) relata que os modelos matemáticos epidemiológicos usados em pandemias passadas, foram pouco eficazes de predizer números reais. De forma simplificada, obtendo resultados estatísticos, buscou dados dessa pandemia em 30 países que já haviam atingido o pico máximo e estavam ao menos com 5 dias de queda ou estabilização, tendo 18 desses 30 países considerados concluídos o ciclo completo da pandemia por apresentar novos casos menor que 70% dos seus picos de novos caso por dia. O Brasil tem o terceiro maior número de leitos de UTI do mundo, atrás do EUA e da Alemanha.

Diante do cenário atual da pandemia na região de Londrina-PR com base em dados clínicos disponibilizado pelo órgão público SESA-PR (Secretaria da Saúde Paraná), o objetivo do presente projeto é realizar a projeção do avanço da doença por meio de uma aplicação *web* e *mobile*. Para isso, serão coletados dados oficiais sobre a Covid-19 na base de dados do SESA-PR, os quais passarão por processo de filtragem, processamento e aplicação de regressão linear, um algoritmo de *Machine Learning* para predição dos dados e análise de sentimentos da população em rede social ao enfrentar tal pandemia e o isolamento social.

2. FUNDAMENTAÇÃO TEÓRICA

Com a frase “Sem precedentes!”, Giannella e Velho (2020) desenvolveram a pesquisa sobre o tema levantando a seguinte questão: “quando isso vai acabar?”. Tal questão é uma dúvida emergente em momento de larga e acelerada produção de dados, que mesmo com o excesso de informação, não parece aclarar a visualização. Segundo Giannella e Velho, a Matemática e o Design da Informação baseadas em evidências, podem ser: i) a evidenciação de eventos concretos e finalizados, com conjuntos de dados com início, meio e fim e ii) a evidenciação de eventos controlados e em processo, com conjuntos de dados atualizados em tempo real em ambiente controlado.

Contudo, a pandemia de Covid-19, descreve um fenômeno de evento em processo em ambiente não controlado, ou parcialmente controlado, baseados em velocidade, imprevisibilidade e leitos disponíveis. Giannella e Velho, fizeram uma nova abordagem de visualização de dados hipotéticos, evidenciando eventos concretos usando conceitos e modelos matemáticos em simulações para chegar ao gráfico de achatamento da curva com o uso da função exponencial, da função logarítmica que é inversa da função exponencial e da função Logística, que apresenta o decrescimento até se chegar a um patamar aceitável, utilizando o modelo SIR (Susceptíveis, Infectadas e Removidas) em equações não lineares.

Farias *et al.* (2020) descrevem a utilização da Regra de Bayes para representar um número esperado de novos casos infecciosos secundários gerados a partir de um caso infectado em uma população suscetível, a partir do número de contatos que cada

indivíduo infectado realiza com outros indivíduos em uma unidade de tempo médio de duração do processo infeccioso.

3. METODOLOGIA

3.1 Pesquisa inicial e tecnologias

O sistema efetuará previsões sobre o avanço da doença em Londrina, com dados comparativos e informativos dos casos confirmados na região, além de efetuar uma análise em rede social de pessoas na cidade. A partir destas informações, serão feitas as previsões do sentimento geral da cidade ao enfrentar a pandemia e o isolamento social.

Com a intenção de coletar dados verídicos e estáveis sobre a região de Londrina, utilizou-se a plataforma SESA (Secretaria do Estado da Saúde), que disponibiliza as informações oficiais sobre a Covid-19. O órgão atende a região pertinente e a estrutura dos dados foram considerados de fácil utilização e disponibilizados diariamente.

Tratando-se de aplicações web e mobile, muitas tecnologias estão disponíveis para uso no mercado e no meio acadêmico. Para o armazenamento de dados, foi utilizado o banco de dados MongoDB, por se tratar de um banco NoSQL e permitir o armazenamento de dados diretamente no formato JSON (*JavaScript Object Notation*), o qual permite que os dados sejam acessados e manipulados em quaisquer aplicações, independente da tecnologia e da linguagem de programação.

Para *back-end*, foi utilizada a linguagem de programação Python, devido a disponibilidade de bibliotecas para operações e cálculos matemáticos com previsão de dados por meio de regressão linear, além de possuir recursos para realizar a análise dos sentimentos da população. A conexão entre o banco de dados MongoDB com a aplicação Python foi realizada com a biblioteca Pymongo.

Na parte de *front-end* web foi utilizado JavaScript, HTML, CSS e React JS para construção das páginas, além de componentes como React Google Charts para criação dos gráficos. O ambiente de execução NodeJS foi aplicado para subir a aplicação e realizar testes. Para aplicação mobile, foi utilizado o React Native para o desenvolvimento das telas e das funcionalidades e a plataforma Expo para a simulação da aplicação. A comunicação entre o *front-end* e o *back-end* foi realizada

por meio de uma arquitetura API Rest, que permite com que tanto o *front-end* web como o mobile sejam integrados com o mesmo *back-end*. Esta integração com a API foi realizada por meio da biblioteca Flask Cors.

3.2 Documentação e planejamento de desenvolvimento

Baseado nas informações contidas nos dados SESA-PR, foi definido as seguintes funcionalidades do sistema que são apresentados por meio de gráficos na forma de dashboard:

- **Representação de avanço de casos (confirmado, recuperado e óbito) e predição de óbito:** esta funcionalidade da aplicação é responsável por exibir os registros de casos confirmados, recuperados e óbitos na cidade de Londrina, além de utilizar algoritmo de *Machine Learning* para predizer os casos de óbitos da cidade.
- **Representação de afetados agrupado por sexo masculino ou feminino:** por meio de gráfico, representa um agrupamento de casos confirmados filtrados pelo sexo juntamente com atendimentos efetuados na cidade de Londrina.
- **Relação de atendimentos de pacientes agrupado por cidade:** exibe um agrupamento de casos confirmados, filtrados pela cidade de residência do infectado juntamente com atendimentos efetuados na cidade de Londrina.
- **Representação de infectados, agrupados por idade:** demonstra o agrupamento de casos confirmados, filtrados pela idade juntamente com atendimentos efetuados na cidade de Londrina.
- **Análise de sentimentos:** exibe resultado de análise de sentimento, executado por algoritmo de *Machine Learning* que classifica os posts do Twitter em "Positivo", "Negativo" e "Neutro" filtrados pela região do Brasil.

As seguintes funcionalidades, foram separadas com foco na comunicação e apresentação de informações sobre a estrutura do sistema ao usuário.

- **Quadro de comunicado:** esta funcionalidade tem por objetivo, exibir comunicados da aplicação para o usuário, trazendo informações e esclarecimentos da falta de dados, informações ou erros.

- **Informações do projeto:** tem como objetivo explicar ao usuário, como os dados exibidos no dashboard foram obtidos.

Para detalhar como deveria ser feito o desenvolvimento, processos e tecnologia a ser seguida durante desenvolvimento, foi construído documentos para cada caso de uso do sistema, onde havia descrições do objetivo do caso, público alvo, atores, descrição, pré-condições, fluxos básicos, fluxos alternativos, fluxos de exceção, pós-condição, requisitos funcionais e não funcionais. Para ter uma visão geral da aplicação foi construído um diagrama de caso de uso.

4. APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

Antes de realizar processamento e armazenamento dos dados, foi necessário efetuar tratamento das informações contidas nos arquivos disponibilizados pelo SESA. Este tratamento permitiu filtrar informações irrelevantes para aplicação, como dados de outras cidades e dados sem possibilidade de uso.

A estrutura das informações em formato CSV (*Comma-separated values*), que ao longo da pandemia tiveram alterações, necessitaram de implementação de código para tratar essas alterações, mapeando e localizando as informações relevantes. Todo processo de coleta foi executado manualmente, por meio do *download* os arquivos CSV do site da SESA, renomeando de acordo com sua data no padrão dd_MM_yyyy (dd=dia, MM=mês, yyyy=ano). Por exemplo, um arquivo disponibilizado no dia 04 de Dezembro de 2020 recebe o nome de "04_12_2020".

Um dos arquivo disponibilizado pela plataforma que foram processados foi o que representava os casos e óbitos (Figura 1), que é utilizado no processamento de registro de casos confirmados, recuperados e óbitos pelo Covid-19. Outro arquivo processado é o que mostra os dados gerais, onde é trabalhado os dados de casos confirmados agrupado por idade, sexo e cidade.

Figura 1 – Arquivo CSV de casos e óbitos.

```

IBGE;RS;MACRO;Município;Casos;Óbitos;Recuperados;Em_investigacao
4100103;18;NORTE;Abatiá;93;1;86;0
4100202;02;LESTE;Adrianópolis;191;4;168;0
4100301;02;LESTE;Agudos do Sul;122;1;77;0
4100400;02;LESTE;Almirante Tamandaré;2728;65;719;14
4100459;11;NOROESTE;Altamira do Paraná;21;2;16;1
4100509;12;NOROESTE;Altônia;237;3;173;1
4100608;14;NOROESTE;Alto Paraná;196;1;146;0
4100707;12;NOROESTE;Alto Piquiri;124;4;108;2
4100806;17;NORTE;Alvorada do Sul;150;5;7;0
  
```

Fonte: O autor (2020).

Através da API Tweepy, uma biblioteca do Python, foram pesquisados tweets filtrados pela região do Brasil, com palavra-chave “covid” e incluiu-se o parâmetro “extended” no “tweet_mode”, para que seja retornado toda mensagem da postagem. O retorno da API é anexado e armazenado no banco de dados.

Para gerar os dados de evolução da doença na cidade contendo os casos de confirmados, óbitos e recuperados, foram utilizados todos dados já salvo no banco referente ao arquivo “Casos e Óbitos”. Estes dados, são chamados pelo *front-end* que interage com a API, buscando os dados salvos no banco e retornando no formato JSON, onde o front-end carrega e gera os gráficos utilizando React Google Charts.

Para gerar os dados de predição de óbito, foi utilizado a biblioteca Scikit Learn, onde consumiu-se todos os dados existentes de casos confirmados (variável independente) e óbitos (variável dependente) registrado no sistema, conforme mostra a Figura 2. Estes dados foram passados no modelo de regressão linear como dados para treino e teste. Os dados de predição, foram preditos por uma média de evolução de casos confirmados dos últimos cinco dias, que foram acrescentados a partir do último registro, simulando a evolução dos casos nos dias seguintes. Os dados simulados foram passados como parâmetro no modelo de regressão (variável independente) já treinado que retornou a predição de óbito para cada dia. Estes dados foram salvos no banco e ficam disponível para consumo do front-end através da API que segue o mesmo processo descrito na comunicação entre API e *front-end*.

Figura 2 – Evolução dos casos.

```

},
{
  "Confirmado": 13846,
  "Data": "11142020",
  "Obito": 320,
  "Recuperado": 10375
},
{
  "Confirmado": 13907,
  "Data": "11152020",
  "Obito": 320,
  "Recuperado": 10392
},
{
  "Confirmado": 13960,
  "Data": "11162020",
  "Obito": 321,
  "Recuperado": 10419
},
{
  "Confirmado": 14098,
  "Data": "11172020",
  "Obito": 321,
  "Recuperado": 10568
}
]

```

Fonte: O autor (2020).

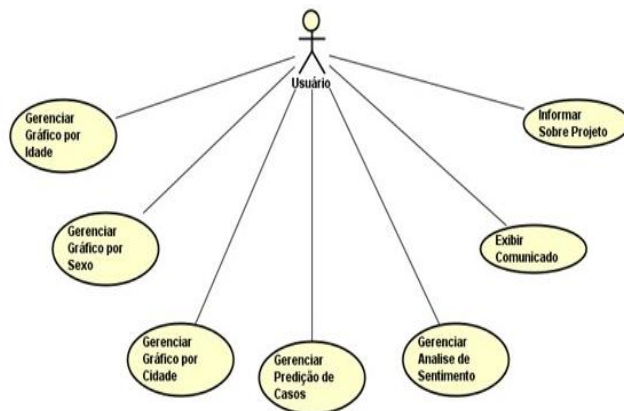
Os dados de casos confirmados classificados pelo sexo, foram obtidos através de uma consulta no banco referente aos dados salvos do tipo “Geral”, onde foram contabilizados os valores de “M” para masculino e “F” para feminino. Após totalizar os valores, foram anexados e salvos no banco, ficando disponível para consumo.

Referente aos casos confirmados agrupados pela idade, seguiu-se o mesmo processo dos casos classificados pelo sexo, porém são contabilizados o número de registros dentro das faixas 0-10, 11-17, 18-28, 29-40, 41-59 e maior que 60 anos. Para os dados do tipo atendimento por cidade, utilizaram-se registros salvos no banco referente aos dados “Geral”, a contabilização dos dados leva em consideração o município de atendimento, que no caso foi a cidade Londrina e o município de residência. Por fim, os dados foram salvos no banco de dados e disponíveis para o consumo.

As análises de sentimentos, utilizam-se os tweets salvos no banco que processam uma segunda camada de filtro. Primeiramente, foram retirados links, pontos e vírgulas das mensagens, já que estes elementos são considerados resíduos para análise e posteriormente são retirados artigos definidos e indefinidos das frases. Os resultados foram manipulados com a biblioteca TextBlob, que por característica, realiza processamento computacional de linguagens naturais, no qual os dados são

traduzidos para idioma inglês para que funcione com a biblioteca. Para cada texto, é chamado a função de análise de sentimento do TextBlob, que retorna a polaridade da mensagem com valores entre 1, 0, -1, onde 1 é considerado positivo, -1 negativo e 0 neutro. A polaridade da mensagem foi salva no banco e por fim, é realizado a contabilização, ficando disponível para consumo.

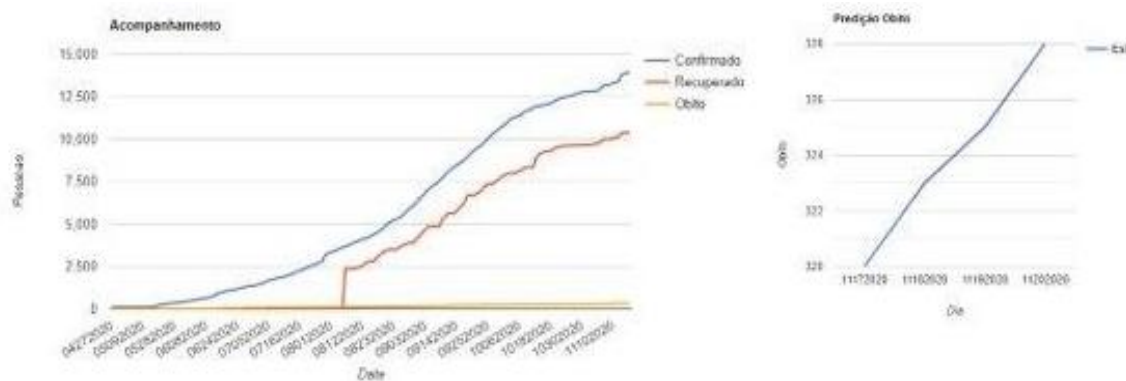
Figura 3 – Diagrama de casos de uso do sistema.



Fonte: O autor (2020).

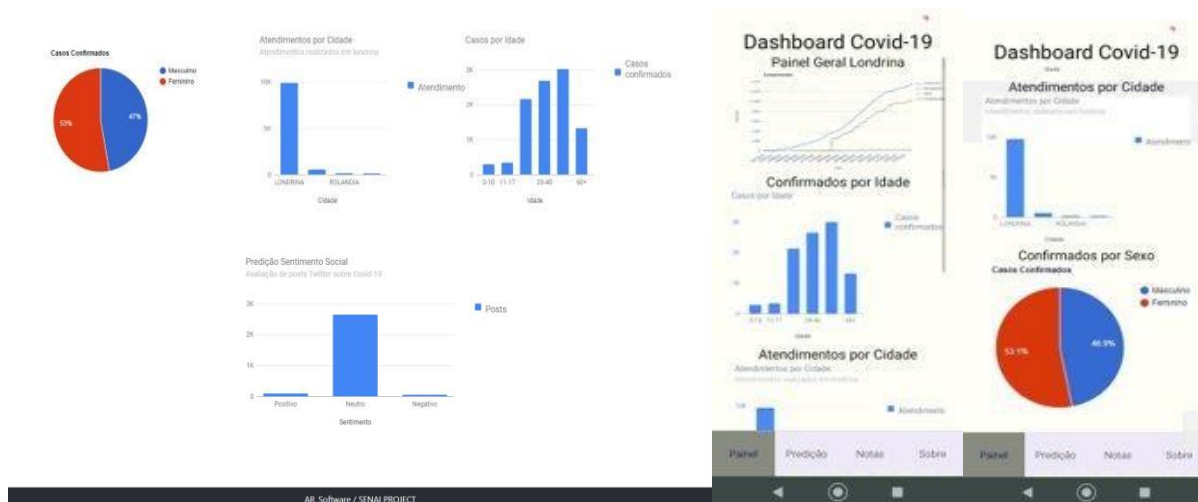
Após todo processo de desenvolvimento, este projeto chegou ao resultado proposto, cumprindo todos os requisitos e seguindo toda documentação estabelecida. Os requisitos funcionais estão sendo ilustrados no diagrama de casos de uso da Figura 3, um exemplo de um gráfico de acompanhamento da Covid-19 e a respectiva predição na Figura 4 e um dashboard da aplicação *web* e *mobile* na Figura 5.

Figura 4 – Gráfico de acompanhamento dos casos confirmados, ocorrência de óbitos e recuperações, junto com a predição realizada.



Fonte: O autor (2020).

Figura 5 – Dashboard da aplicação *web* (à esquerda) e *mobile* (à direita).



Fonte: O autor (2020).

Pode-se observar na Figura 5 que na cidade de Londrina, o número de casos de óbitos representa em torno de 2% dos registros totais e que a faixa etária com maior número de infecção estão entre 41 e 59 anos. O sexo feminino possui maior número de casos confirmados.

5. CONSIDERAÇÕES FINAIS

De acordo com a pesquisa realizada neste projeto, as estimativas relacionadas aos avanços da Covid-19 tiveram suas premissas em cálculos estatísticos matemáticos. O desafio foi aplicar tecnologias digitais, incluindo *Machine Learning*, inteligência artificial e finalizando com aplicação *web* e *mobile*, a fim de disponibilizar a população os aplicativos para os usuários.

A qualificação de dados, a vulnerabilidade populacional e diferentes ações dos gestores públicos, tornaram desafiadora a utilização da predição por algoritmos de aprendizagem de máquina. É importante frisar que para melhor eficácia, há a dependência de grande volume de dados para o aprendizado, ausentes no início desta pandemia.

Por fim, apesar do momento difícil na história da humanidade, a necessidade de devolver soluções na velocidade necessária, fez com que esse surto de Covid-19 proporcionasse um grande avanço no desenvolvimento intelectual dos cientistas e

pesquisadores, um avanço muito positivo. Certamente, esses esforços já trouxeram benefícios e deverá acelerar a adoção de soluções digitais na saúde e principalmente, salvar vidas.

REFERÊNCIAS

FARIAS, C. M.; MEDRONHO, R. A.; TRAVASSOS, G. H. Avaliação do comportamento da Covid-19 no estado do Rio de Janeiro e seus municípios com base em R(o) calculado a partir das evoluções anteriores de R dos casos notificados à Secretaria de Estado de Saúde-RJ. **Grupo de Trabalho Multidisciplinar da UFRJ sobre a Coronavirus Disease 19 (COVID-19)**. Rio de Janeiro, 2020, 18p.

FERNANDES, J. L. Covid-19 no Brasil: aprendendo a andar no escuro sem deixar nada para trás. **Arquivos Brasileiros de Cardiologia**, v. 114, n. 6, jun. 2020.

G1. Entenda os impactos da pandemia de coronavírus nas economias global e brasileira, 2020a. Disponível em: <<https://g1.globo.com/economia/noticia/2020/02/26/entenda-os-impactos-do-avancodo-coronavirus-na-economia-global-e-brasileira.ghtml>>. Acessado em: 26 ago. 2020.

GIANNELLA, J.; VELHO, L. Visualização em Tempo de Conora Vírus – parte 1. **Laboratorio VISGRAF**: Instituto de Matemática Pura e Aplicada. Rio de Janeiro, 2020, 23p.

MINISTÉRIO DA SAÚDE (Brasil). Sobre a Doença: O que é Covid-19. [S. I.], 2020a. Disponível em: <<https://coronavirus.saude.gov.br/sobre-a-doenca#o-que-e-covid>>. Acessado em: 26 ago. 2020.

MINISTÉRIO DA SAÚDE (Brasil). Painel Coronavírus [S. I.], 2020b. Disponível em: <<https://covid.saude.gov.br>>. Acessado em: 26 ago. 2020.

NEVES, K. D; BARROS, N. Análise preditiva de casos confirmados de Covid-19 no brasil e em oito países baseada no modelo não linear de Gompertz. **SciELO Peru**, v. 202, 2020.

SECRETARIA DA SAÚDE (Paraná). Boletim – Informe Epidemiológico Coronavírus (COVID-19) – Arquivos CSV (Dados Abertos) [S. I.], 2020. Disponível em: <<http://www.saude.pr.gov.br/Pagina/Coronavirus-COVID-19>>. Acessado em: 26 ago. 2020.

Capítulo IV – Melhorando a comunicação com a imobiliária por meio de um sistema web

Silvio Aparecido Capelo¹²

Cadmiel Matioli Donato¹³

Rafael Augusto Balabem¹⁴

Anderson Paulo Ávila Santos¹⁵

Fabio Takeshi Matsunaga¹⁶

Lucas Busatta Galhardi¹⁷

RESUMO

Com o advento da tecnologia, o uso de serviços online tem crescido nos últimos anos, especialmente com a recente pandemia da Covid-19, em que o *lockdown* é exigido. Um dos setores que tem mais sofrido demanda é o de mercado imobiliário. Existem problemas como falta de comunicação direta com a imobiliária, falta de eficiência nas buscas e localização e baixa qualidade das imagens. Considerando-se o exposto, o objetivo deste projeto é melhorar a comunicação entre os consumidores e a imobiliária por meio de UX. Para isso, será desenvolvido um sistema *web* com arquitetura MVC. O front-end foi desenvolvido em HTML, CSS e Javascript e o back-end em framework Laravel, que foi responsável por gerenciar os dados dos usuários, imóveis e as imagens. O sistema web torna o acesso ao serviço mais fácil, podendo ser realizado por qualquer dispositivo que tenha acesso à Internet.

Palavras-chave: API, *back-end*, *front-end*, Inteligência Artificial, *Machine Learning*.

Improving communication with the real estate company through a web system

ABSTRACT

¹² Graduando em Engenharia de Software.

¹³ Graduando em Engenharia de Software.

¹⁴ Graduando em Engenharia de Software.

¹⁵ MSc. Anderson Paulo Ávila Santos. E-mail: anderson.avila@sistemafiep.org.br

¹⁶ MSc. Fabio Takeshi Matsunaga. E-mail: fabio.matsunaga@sistemafiep.org.br

¹⁷ MSc. Lucas Busatta Galhardi. E-mail: lucas.galhardi@sistemafiep.org.br

With the advent of technology, the use of online services has grown in recent years, especially with the recent Covid-19 pandemic, in which the lockdown is required. One of the sectors that has suffered the most demand is the real estate market. There are problems such as lack of direct communication with the real estate company, lack of efficiency in searches and location and low quality of images. Considering the above, the objective of this project is to improve communication between consumers and real estate through UX. For this, a web system with MVC architecture will be developed. The front-end was developed in HTML, CSS and Javascript and the back-end in Laravel framework, which was responsible for managing user data, properties and images. The web system makes access to the service easier and can be performed by any device that has access to the Internet.

Key-words: API, Artificial Intelligente, back-end, front-end, Machine Learning.

1. INTRODUÇÃO

Com o crescimento da quantidade de sites imobiliários, a tendência é que as pessoas deixem de se deslocar até uma imobiliária procurar um imóvel e passem a acessar estes serviços de forma *online*. O surgimento da pandemia do Covid-19 acelerou esse processo, uma vez que para conter a doença em vários estados foram tomadas medidas de segurança, tais como o *lockdown* (confinamento). Fato este que gerou um aumento no número de pessoas que buscam sites imobiliários ou realizam transações comerciais na *web*. (BALENA e PEGORINI, 2019)

De acordo com Giglio *et al.* (2006) o uso da internet pode representar um grande diferencial, pois ela proporciona maior agilidade na prestação de serviço. A plataforma Google em um de seus artigos afirma que houve um aumento de 34% na procura de imobiliárias no período de maio de 2021. Este quadro que se apresenta, sugere que as empresas que não se adequarem a esse novo formato de negócio perderão espaço no mercado.

Segundo Moro e Andrade (2020), em seu artigo Qualidade de Sites Imobiliários por meio de satisfação do cliente, a pesquisa dicômica, chega à conclusão de que 67% dos clientes que fizeram negociação com imobiliárias através da internet não estão satisfeitos com os serviços recebidos e tiveram um score abaixo de zero. Os pontos críticos apontados na pesquisa e que servem de base para a idealização desses projetos foram: falta de Informação dos produtos, buscas complexas, fotos

fornecidas dos imóveis fora de padronização, falta de comunicação direta com imobiliária, responsabilidade o site entre outros.

De acordo com os dados obtidos pela pesquisa acerca do mercado imobiliário *online*, que demonstra uma demanda crescente na procura de imóveis através da internet, a experiência do cliente tem se mostrado deficitária em alguns aspectos, tais como insatisfação com informações sobre os imóveis, falta de comunicação direta com a imobiliária, insatisfação com filtros de buscas, insatisfação com a geolocalização dos imóveis, baixa qualidade de fotos dos imóveis e a falta de responsabilidade para acesso em dispositivos móveis (MORO e ANDRADE, 2020).

De encontro com a velocidade em que o mercado imobiliário se desenvolve na *web*, o objetivo deste presente projeto é desenvolver um sistema web que correspondam as expectativas dos usuários utilizando conceito de UX (*User Experience* ou Experiência do Usuário) e implementar as ferramentas necessárias para melhorar a comunicação entre consumidores e imobiliárias.

2. METODOLOGIA

2.1 Processo de *software* e gestão

Para este projeto, a metodologia aplicada para o desenvolvimento do *software* foi o Scrum, que se trata de uma metodologia ágil de desenvolvimento incremental e iterativo por meio da qual são produzidas e entregues partes dos softwares a cada iteração. De acordo com Cruz (2013), a ideia principal do Scrum é que um pequeno time de pessoas pode tratar e resolver problemas complexos e adaptativos enquanto entrega produtos de forma criativa.

Para gerenciar as tarefas, foi utilizada o Trello, que é uma ferramenta colaborativa que organiza projetos em quadros (*boards*), em que são inseridas listas de tarefas a serem seguidas individualmente ou em equipe.

Para o versionamento dos códigos-fontes do projeto, foi utilizado o GIT, que é um sistema de controle de versões distribuído, usado principalmente no desenvolvimento de *software*. Para o compartilhamento das versões do projeto, foi utilizado o GitHub, que é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o GIT.

2.2 Tecnologias

Para o desenvolvimento do sistema *web*, foram utilizados alguns *frameworks* para o desenvolvimento do *front-end* e *back-end*. Um *framework* é uma estrutura-base que contém um conjunto de funções e componentes pré-definidos, funções e

componentes estes que se relacionam para disponibilizar funcionalidades específicas ao desenvolvimento de *software*.

O Laravel foi utilizado para o desenvolvimento do *back-end*. Trata-se de um *framework* MVC de código aberto para PHP, robusto que fornece fácil desenvolvimento de aplicações web com recursos como um sistema de empacotamento modular com um gerenciador de dependências dedicado, acesso a bancos de dados relacionais e outros utilitários para implantação e manutenção de aplicações (STAUFFER, 2019).

Para o *front-end*, foi utilizado HTML para gerenciar o conteúdo e a estrutura de uma página na *web*, Javascript para a interação da página e o CSS (*Cascading Style Sheets*) para a estilização da página. Javascript é uma linguagem de script ou programação que permite implementar recursos complexos em páginas da *web*. Para o projeto, o Javascript foi utilizado em conjunto com o Ajax, que é o acrônimo para JavaScript assíncrono + XML, utilizado para tornar páginas Web mais interativas com o usuário. Além disso, em conjunto com o CSS, foi utilizado o Bootstrap, um *framework* CSS que organiza e gerencia o *layout* de um site.

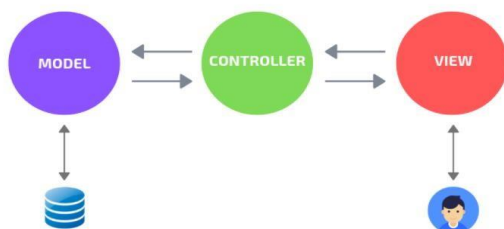
O sistema foi desenvolvido utilizando MVC, que é o acrônimo de *Model-View-Controller* (LUCIANO e ALVES, 2011). O MVC define um padrão arquitetural, que funciona da seguinte forma:

Model é a camada responsável pela parte lógica da aplicação, que gerencia todos os recursos (consultas ao BD, validações e notificações).

View é a camada responsável por exibir dados para o usuário, seja em páginas HTML, JSON, XML. A camada *View* não possui responsabilidade de saber quando vai exibir os dados, apenas como irá exibi-los.

Controller recebe todas as requisições do usuário. Seus métodos chamados *actions* são responsáveis por uma página, controlando qual *model* usar e qual *view* será mostrado ao usuário.

Figura 1 – Arquitetura MVC.



Fonte: os autores.

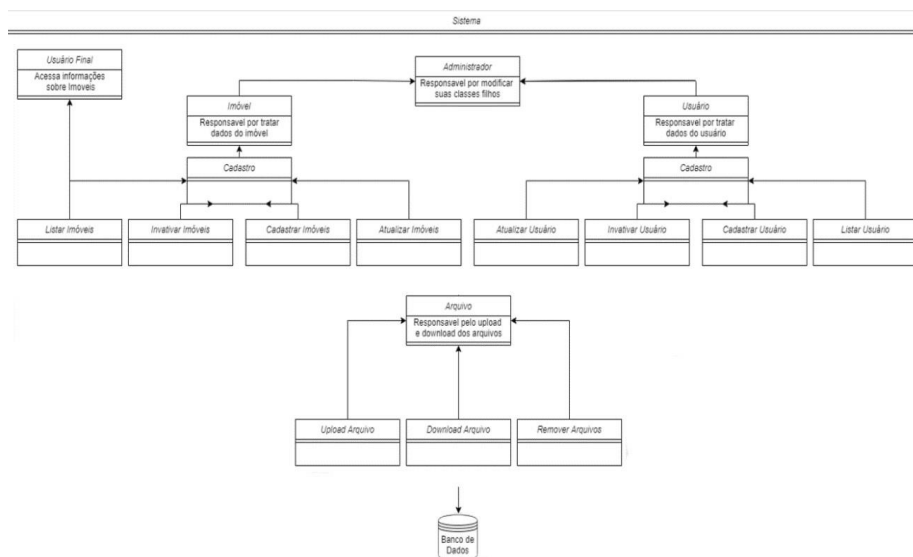
Ao receber uma requisição, o *Controller* solicita ao *Model* as informações necessárias (que virão do banco de dados), que as obtém e retorna ao *Controller*. De posse dessas informações, o *Controller* as envia para a *View* que irá renderizá-las, como mostra a Figura 1.

3. APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

3.1 Diagramas de modelagem

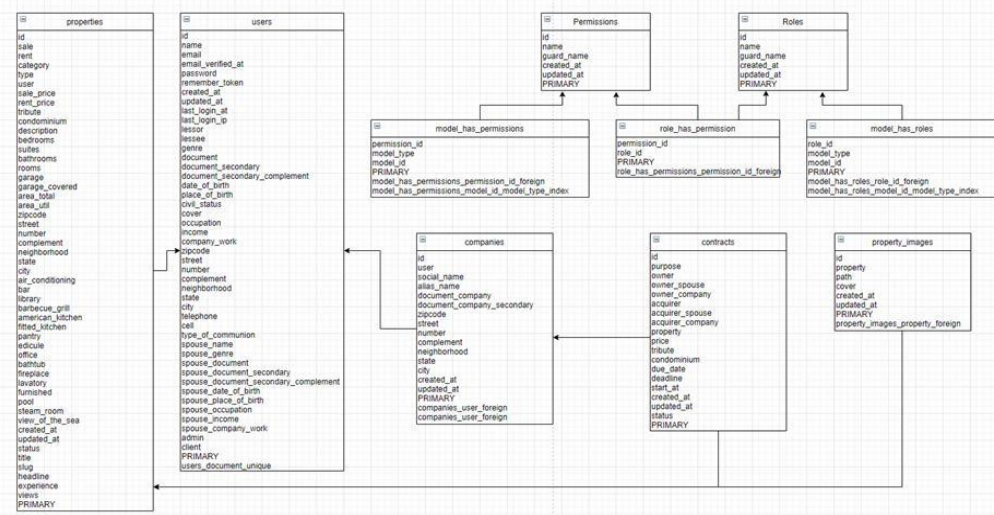
Os diagramas foram utilizados para facilitar o entendimento da solução arquitetônica adotada. É um recurso bastante didático, em que vários aspectos do projeto são explicados de maneira direta e visual. O diagrama de arquétipos auxilia na visualização da arquitetura do projeto como um contexto geral, possibilitando entender quais os pontos principais de acordo com a Figura 2.

Figura 2 - Arquétipos arquiteturais.



Fonte: os autores.

Figura 3 - Diagrama de entidade e relacionamento (DER).



Fonte: os autores.

3.2 Desenvolvimento do sistema

Nesta sessão serão apresentados os principais códigos para exemplificar o desenvolvimento do sistema. A partir deste ponto será explicado como foi desenvolvido a classe de Usuário, assim como seus métodos que servirão de base para implementação das demais classes. O início da implementação da classe se dá a partir das importações de bibliotecas e componentes necessários para desenvolvimento de todas as funcionalidades.

Para otimizar o código foi utilizado o ORM (*Object Relational Mapping*), responsável por aprimorar as buscas de dados e pelo auxílio no envio de informações para o banco de dados através das classes. Com o objetivo de demonstrar com mais clareza temos um exemplo abaixo da implementação de um método que realiza buscas através da classe de Usuário com utilização do ORM (Figura 4).

Figura 4 - Criação da classe e método de listagem de usuários

```
class UserController extends Controller
{
    public function index(){
        $users = User::get();
        return view('admin.users.index', compact('users'));
    }
}
```

Fonte: os autores.

Figura 5 – Método de criação de usuários

```
public function create(){
    return view('admin.users.create');
}

public function store(UserRequest $request){

    $userCreate = User::create($request->all());

    return redirect()->route('admin.users.edit', [
        'user' => $userCreate->id
    ])->with(['color' => 'green', 'message' => 'Cliente cadastrado com sucesso!']);
}
```

Fonte: os autores.

Seguindo modelo de implementação de classe fornecido pelo Framework Laravel, foram criados vários métodos com a finalidade de tramitar as informações entre o *back-end* e *front-end*. O método *create* tem a função de chamar o código que mostra a parte visual da aplicação, assim como passar dados recebidos do banco de dados para que seja mostrado em tela, enquanto o método *store* se encarrega de enviar de fato as informações para o banco de dados, conforme mostra a Figura 5.

Do mesmo modo se dá a implementação dos métodos de edição, onde a função *edit* se encarrega de mostrar o formulário, assim como servir de condutor para enviar as informações necessárias para a edição do usuário. O método *update* se encarrega de receber as informações vindas do formulário através da classe *Request* (requisição vida do cliente pelo formulário) e enviar essas informações para o banco de dados. Na figura 6, é apresentado o método *update* que recebe todos os dados de usuários provenientes da classe *Request* e posteriormente serão salvos no banco de dados.

Figura 6 – Método update, para edição de usuários.

```
public function edit(User $user){
    return view('admin.users.edit', compact('user'));
}

public function update(Request $request, User $user){
    $user->update($request->all());
    return redirect()->route('admin.users.index')
        ->with(['color'
            => 'green', 'message'
            => __('messages.users.update.success')]);
}
```

Fonte: os autores.

Tendo como base a implementação da classe de usuário, a classe *properties* matém a mesma lógica de funcionamento. A figura 7 apresenta a criação e definição do nome da classe com seu primeiro método, a listagem de propriedades e retorno da interface. A lista é adquirida do banco de dados de forma decrescente.

Figura 7 - Criação da classe e método de listagem de propriedades.

```
class PropertyController extends Controller
{
    public function index(){
        $properties = Property::orderBy('id', 'DESC')->get();
        return view('admin.properties.index', compact('properties'));
    }
}
```

Fonte: os autores.

Para criar uma propriedade e fazer o registro no banco de dados é necessário ter dois métodos: o *create* que fica responsável de retornar os devidos dados para contemplar o relacionamento entre classes, possibilitando o registro que será feito através do método *store*, pelo qual serão enviados os dados a serem salvos, retornando para a tela de edição acordo com a Figura 8.

Para proceder na edição de propriedades são necessários dois métodos: o *edit* e o *update* já anteriormente citados. Através do método *imageSetCover* é possível definir qual será a imagem principal do anúncio da propriedade. O método *imageRemove* removerá uma imagem da propriedade de acordo com a Figura 9.

Figura 8 – Criação de propriedades.

```
public function create(){
    $users = User::orderBy('name')->get();
    return view('admin.properties.create', compact('users'));
}

public function store(PropertyRequest $request){
    $createProperty = Property::create($request->all());

    $validator = Validator::make($request->only('files'), ['files.*' => 'image']);

    if ($validator->fails() == true) {
        return redirect()->back()->withInput()->with([
            'color' => 'orange',
            'message' => 'Todas as imagens devem ser do tipo jpg, jpeg ou png.',
        ]);
    }

    foreach ($request->allFiles()['files'] as $image) {
        $propertyImage = new PropertyImage();
        $propertyImage->property = $createProperty->id;
        $propertyImage->path = $image->store('properties/' . $createProperty->id);
        $propertyImage->save();
        unset($propertyImage);
    }

    return redirect()->route('admin.properties.edit', [
        'property' => $createProperty->id,
    ])->with(['color' => 'green', 'message' => 'Imóvel cadastrado com sucesso!']);
}
```

Fonte: os autores.

Figura 9 – Definir e remover imagens de propriedades.

```
public function imageSetCover(Request $request){
    $imageSetCover = PropertyImage::where('id', $request->image)->first();
    $allImage = PropertyImage::where('property', $imageSetCover->property)->get();

    foreach ($allImage as $image) {
        $image->cover = null;
        $image->save();
    }

    $imageSetCover->cover = true;
    $imageSetCover->save();

    $json = [
        'success' => true,
    ];

    return response()->json($json);
}

public function imageRemove(Request $request){

    $imageDelete = PropertyImage::where('id', $request->image)->first();

    Storage::delete($imageDelete->path);
    Cropper::flush($imageDelete->path);
    $imageDelete->delete();

    $json = [
        'success' => true,
    ];
    return response()->json($json);
}
```

Fonte: os autores.

Fundamentando-se na implementação descrita até o momento, pode-se observar que há um padrão que norteia o desenvolvimento do sistema. A criação de cada classe que o compõe segue um padrão que se repete durante todo o projeto. Portanto, o que foi abordado demonstra claramente como se desenvolverá a implementação das demais classes. O projeto do *front-end* é constituído de um *layout* que representa toda a parte visual do projeto. Os principais componentes gerados são:

Painel administrativo para a gestão de Funcionários, Cadastros e Permissões;

Cadastros de Contratos;

Painel Administrativos com fluxo de Caixa;

Implementações de Geolocalização e mapas com Street View;

Integração com Redes Sociais;

Filtros simples e avançados para buscas mais detalhadas;

Chat para facilitar a comunicação com a imobiliária;

Módulo de conexão com WhatsApp.

Deste modo, vale ressaltar que todas estas funcionalidades foram implementadas levando em consideração as pesquisas referentes a satisfação dos clientes, já mencionada nesta pesquisa e que serviu como razão primordial o modo o qual foi concebido.

4. CONCLUSÃO

O desenvolvimento deste projeto visa fornecer um software de gestão Imobiliária, melhorando a qualidade e agilidade dos processos administrativos. A linguagem de programação utilizada, o PHP, foi utilizada devido à grande familiaridade dos desenvolvedores com a linguagem e por ter uma vasta quantidade de biblioteca disponível para integração na aplicação, suprimindo as demandas e requisitos. O próximo passo foi a escolha por aplicação *web* para que pudesse ser acessível de qualquer máquina com acesso à internet, seja de um computador, celular ou tablet, simplificando o acesso.

Para além de tudo que já foi implementado, mais recursos serão agregados futuramente para melhorar ainda mais a experiência do usuário tanto do ponto de vista do empresário, quanto do ponto de vista do cliente. Serão implementadas melhorias como: alertas de ajustes de contratos, notificações, notas fiscais, baixa automática de boletos, melhorias na área de cliente, melhorias no *dashboard*, entre outros.

O presente trabalho trouxe um grande aprendizado, ambos relacionados à modelagem, análise de demanda e implantação de sistema. Durante todo este processo, muitos testes e alterações foram feitos até que um resultado mais próximo do ideal fosse obtido, podendo superar as expectativas.

No processo de execução do trabalho, foram encontradas algumas dificuldades. Entre as que se destacam podemos citar os mapas e documentos. Através dos conceitos apresentados neste artigo, será possível desenvolver novas aplicações que buscam como principal objetivo a experiência do usuário com foco na resolução de problemas, proporcionando maiores benefícios e informações aos usuários.

REFERÊNCIAS

BALENA. V. M.; PEGORINI, V. Sistema web para gestão imobiliária. 2019. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Universidade Tecnológica Federal do Paraná (UTFPR), Pato Branco, 2019.

CRUZ, Fabio. **Scrum e PMBOK unidos no Gerenciamento de Projetos**. Editora Brasport, 2013. 416 p.

GIGLIO, E. M.; PEREIRA, P. G.; RYNGELBLUM, A. Investigação sobre as relações entre a internet e as mudanças estratégicas, exemplificadas no mercado imobiliário. **Revista Brasileira de Gestão de Negócios**, v. 8, n. 21, p. 43-54, 2006.

LUCIANO J.; ALVES, W. J. Padrão de arquitetura MVC: Model-view-controller. **Revista EPeQ Fafibe 3ª. Ed.**, v. 1, 2011.

MORO, M. F.; ANDRADE, D. F. Avaliação da Qualidade de WebSites de Imobiliária por Meio da Avaliação dos Usuários, novembro de 2020. In. XL Encontro Nacional de Engenharia de Produção, 2020. Foz do Iguaçu. **Anais...** Foz do Iguaçu: S.N., 2020.

STAUFFER, M. **Laravel: Up & running**: A framework for building modern PHP apps. O'Reilly Media, 2019.

Capítulo V – SENAI METRICS: IMPULSIONANDO SEU NEGÓCIO COM DADOS.

Marcia Regina Ramalho Rodrigues¹⁸

Matheus Shiniti Honda Rocco¹⁹

Thiago Moraes Silva de Santana²⁰

Fábio Takeshi Matsunaga²¹

Lucas Busatta Galhardi²²

Anderson Paulo Avila Santos²³

RESUMO

O presente artigo descreve o sistema SenAI Metrics, um sistema focado em coletar dados de tópicos em alta a partir de uma palavra-chave e com isso gerar métricas em um formato visual. Essas métricas têm como objetivo auxiliar micro e pequenas empresas que tiveram seus estabelecimentos fechados temporariamente em decorrência do Lockdown, adotado durante a pandemia do COVID-19 como uma medida restritiva e meio de prevenção. Este evento ocasionou uma dificuldade na retomada dos rendimentos de inúmeras empresas, forçando a grande maioria dos empreendedores a remodelarem sua visão sobre o mercado do seu produto ou serviço. O artigo também apresenta que estes dados podem auxiliar na administração dos recursos da empresa, aplicando-os em estratégias que muitas das vezes são simples, mas que nem sempre são vistas e praticadas da maneira mais rentável possível.

Palavras-chave: Métricas, IA, tópicos em alta, COVID-19.

¹⁸ Graduando em Engenharia de Software.

¹⁹ Graduando em Engenharia de Software.

²⁰ Graduando em Engenharia de Software.

²¹ MSc. Fabio Takeshi Matsunaga. E-mail: fabio.matsunaga@sistemafiep.org.br

²² MSc. Lucas Busatta Galhardi. E-mail: lucas.galhardi@sistemafiep.org.br

²³ MSc. Anderson Paulo Ávila Santos. E-mail: anderson.avila@sistemafiep.org.br

SENAI METRICS: BOOST YOUR BUSINESS WITH DATA

ABSTRACT

This article describes the SenAI Metrics system, a system focused on collecting data on hot topics from a keyword and thereby generating metrics in a visual format. These metrics are intended to help micro and small businesses that had their establishments temporarily closed because of the Lockdown, adopted during the COVID-19 pandemic as a restrictive measure and a means of prevention. This event caused a difficulty in recovering the income of numerous companies, forcing most entrepreneurs to reshape their vision of the market for their product or service. The article also shows that these data can help in managing the company's resources, applying them in strategies that are often simple, but which are not always seen and practiced in the most profitable way possible.

Keywords: Metrics, AI, Trending Topics, COVID-19.

INTRODUÇÃO

A pandemia da COVID-19 tem causado diversos fechamentos temporários de pequenas e médias empresas em decorrência do *Lockdown*, uma medida restritiva adotada como meio de prevenção. Após estas medidas restritivas, tais empresas enfrentam diversas dificuldades em retornar suas economias devido à reabertura gradual. Existe um grande desafio destas empresas em recuperarem as vendas e ganhos financeiros perdidos durante o fechamento temporário.

Com a entrada dessas medidas, as empresas tiveram que reconsiderar a maneira do gerenciamento financeiro, pois em alguns casos, o gerenciamento incorreto pode ocasionar a falência da empresa. Para contornar essa situação, há diversas formas de restabelecer a parte financeira, sendo uma, a realização de campanhas publicitárias. Essas campanhas são extremamente importantes para dar relevância ao seu produto ou serviço, além de aumentarem as vendas, por isso, é fundamental transmitir aos seus consumidores quais são os seus objetivos e buscar conhecer o mercado alvo, focando justamente em que os seus clientes buscam.

Diante do exposto, a proposta do projeto é a realizar a agregação de dados de diferentes fontes de dados, a fim de gerar informação para auxiliar tomadas de decisão sobre o gerenciamento do setor de financeiro e de campanhas publicitárias de empresas. As informações fornecidas em forma de métricas relacionadas ao negócio previamente informado, disponibilizam conhecimento a partir dos dados, e assim direcionar campanhas de marketing.

Desta forma, há a possibilidade de atingir níveis de engajamento superiores, bem como uma maior visibilidade nas redes sociais, auxiliando na definição de novos mercados e de públicos-alvo da empresa, aumentando a eficácia das campanhas de marketing e por consequência, adquirir um número maior de vendas de produtos.

FUNDAMENTAÇÃO TEÓRICA

Líderes corporativos consideram a capacidade de analisar dados e gerar insights como uma possibilidade de investimento. Muitas empresas, entretanto, negligenciam a importância de utilizar os dados disponíveis de seu negócio. Ou, em um cenário pior, algumas empresas nem mesmo coletam dados.

Nos modelos de negócio da atualidade, vê-se que a atividade de coletar os dados é essencial, pois a partir de sua análise, é possível gerar a proposição de novos produtos e ferramentas, ou ainda, auxiliar em decisões estratégicas.

Em um cenário não ideal, o processo de estruturação de dados começa com uma planilha. Embora o Excel seja uma ótima ferramenta para uma diversidade de aplicações, usá-lo para armazenar, manipular e gerar insights com um grande volume de dados acaba se tornando uma ideia não eficiente. Isso pode ser suficiente para operações em escala pequena, mas assim que ocorra a necessidade em se ampliar a escala, a eficiência da aplicação fica comprometida.

A melhor maneira de capitalizar os dados e continuamente gerar insights é com a automação. Existem uma diversidade de formas de se coletar dados de forma contínua, entre elas estão: campanhas de e-mail automatizadas, formulários integrados, consulta à bases de dados via protocolos *web*, *chatbots*, entre outros.

Tais ferramentas podem ser usadas para coletar dados e usá-los de forma eficaz para a geração de conteúdo e de informação ao usuário.

Neste sentido, a aplicação proposta nesse projeto é a coleta e a agregação de dados disponíveis publicamente via consulta à serviços *web*, com o objetivo de gerar métricas e informações de assuntos diversos, e que serão disponibilizadas ao usuário final. As fontes de dados consultadas para o protótipo inicial do projeto são

as bases de dados da NewsAPI (NEWS API, 2021) e do Google Trends (GOOGLE TRENDS, 2021).

Google Trends é um aplicativo que disponibiliza a consulta a sua base de dados via API (*Application Programming Interface*), a qual fornece dados com as tendências dos termos de pesquisa ao longo do tempo para diferentes geografias. Ele mostra tendências de pesquisa em tempo real e tendências de pesquisa diárias para a maioria das cidades do mundo.

Com a consulta aos dados fornecidos pelo Google Trends, é mais fácil entender a intenção dos usuários em diferentes épocas para diferentes consultas. Além ainda, eventos em tempo real e tendências do mundo real podem ser observados.

Em diferentes referências científicas, muitos estudos mostram que o Google Trends pode indicar os estágios iniciais de surtos para permitir que os governos tomem medidas (Holistic Seo, 2021), entre outros tipos de uso, como por exemplo, a tendência de crescimento de consumo de produtos e de serviços específicos.

Dessa forma, avalia-se que os dados coletados e analisados do Google Trends podem auxiliar em campanhas de publicidade e de estratégias de negócio para verificar o poder da marca e as tendências de consultas relacionadas ao longo do tempo para criar uma estratégia de campanhas inteligentes.

NewsAPI é uma aplicação agregadora de notícias, que tem como recurso uma API REST que retorna metadados para manchetes relevantes com base em uma consulta. Ele cobre uma ampla gama de mercados em todo o mundo, incluindo fontes em diferentes países. Com a consulta em sua base de dados, é possível obter recursos como as principais fontes de notícias do conteúdo buscado, assim como tipos específicos de mídia.

Ambas as fontes de dados utilizados no desenvolvimento deste projeto constituem uma importante fonte de informação para compreender como a maior parte da população pensa, quais são as tendências observadas no mercado, quais conteúdos estão sendo produzidos e assim delinear um perfil de consumo atualizado.

METODOLOGIA

A pesquisa foi realizada e planejada seguindo a metodologia ágil conhecida como Scrum. Apesar da palavra ágil remeter a lembrança de algo rápido, a ideia principal do ágil não é ser rápido, mas de segmentar o projeto em partes menores, facilitando a gestão de cada componente do projeto e assim garantindo uma qualidade melhor. Segundo Ken e Jeff (2013), o Scrum é um framework para desenvolver e manter

produtos complexos, sendo uma metodologia para gestão e planejamento de projetos de software.

No Scrum, os projetos são divididos em ciclos, nomeados de *Sprints*. Uma *Sprint* representa um determinado período no qual deve ser realizado um conjunto de atividades. Estes ciclos possuem durações curtas, geralmente de duas a quatro semanas, sendo iniciada uma nova *Sprint* após finalizar a anterior. As *Sprints* são compostas por uma *Sprint Planning Meeting* (Reunião de Planejamento da Sprint), *Dailys Scrum* (Reuniões diárias), uma *Sprint Review* (Revisão da Sprint) e uma *Sprint Retrospective* (Retrospectiva da Sprint).

No início da Sprint, é realizado uma *Sprint Planning Meeting*, que consiste em uma reunião de planejamento para priorizar os itens estabelecidos no *Product Backlog*, sendo estes, as funcionalidades definidas que serão implementadas no projeto.

Após a classificação das atividades, são disponibilizadas em um quadro virtual que segue o modelo do Kanban, garantindo a visibilidade a todos os integrantes do projeto, de maneira que após a conclusão da atividade, possa ser iniciado uma nova sem a necessidade de ter que realizar outra discussão sobre o que ainda está pendente.

Foi utilizado em um dos backends o NodeJs, que pode ser definido como um ambiente para executar JavaScript no lado do servidor. É utilizado por grandes empresas do mercado como por exemplo Uber, LinkedIn e Netflix. Um dos principais motivos de sua adoção é sua alta escalabilidade, flexibilidade, baixo custo e também, uma boa alternativa para implementação de microsserviços e arquitetura *serverless*.

Uma alternativa para a tecnologia utilizada, seria o Go ou Golang, que é uma linguagem criada pelo Google em 2007 e tem como proposta trazer boa performance, segurança e aumentar a produtividade ao desenvolver as aplicações. Go geralmente é utilizado para construir API's, scripts e aplicações embarcadas.

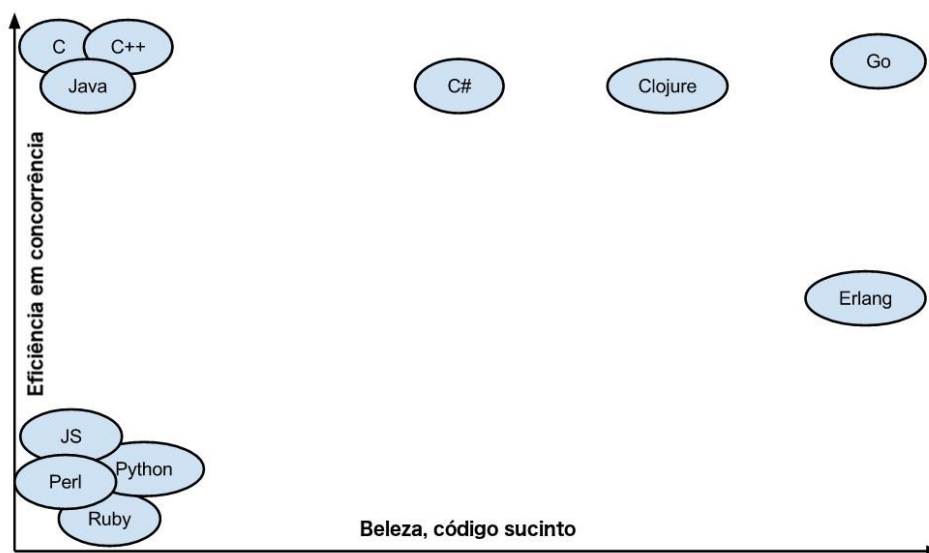
A curva de aprendizado entre Node e Go é bem parecida, mas como o NodeJs é basicamente JavaScript, sua curva tende a ser menor, porém Go é bem simples também, e desenvolvedores que já utilizaram Java, C#, C++ tem uma facilidade maior na sua compreensão.

A respeito do gerenciamento de dependências, Go é infinitamente superior ao NodeJs. É criado uma pasta no repositório local do usuário, e mesmo que múltiplos projetos utilizem as mesmas dependências, elas são armazenadas uma única vez, ocupando muito menos espaço ao comparar com o *node_modules*, que é necessário um por projeto.

Em questão de bibliotecas, NodeJs tem uma grande vantagem, possuindo uma comunidade maior, com mais variedades e vários projetos open source maduros e consolidados. Já Go possui uma comunidade que está em crescimento constante.

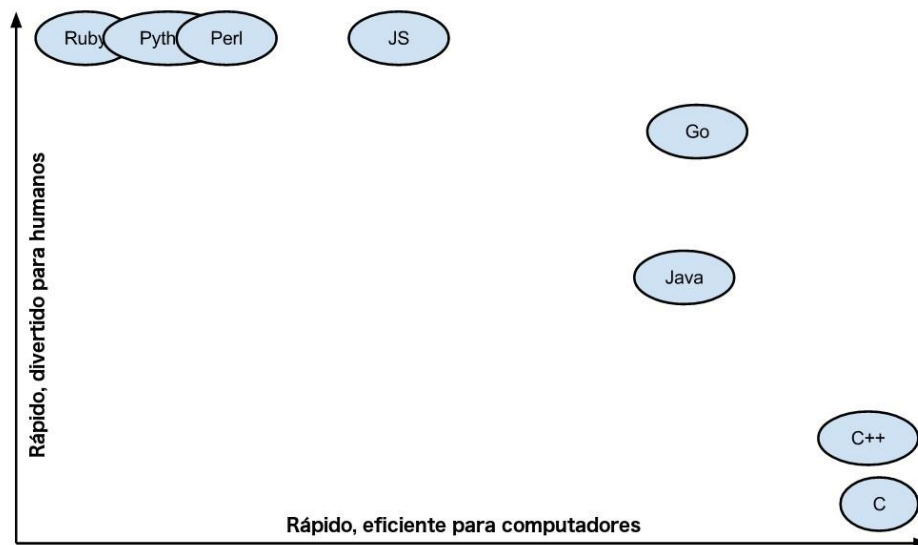
NodeJS tem frameworks muito populares e maduros, como Express, NestJS, LoopBack, etc. Enquanto o GO possui frameworks com uma popularidade menor, porém, muito eficientes, como o GIN, Fiber, etc.

Figura 2. Gráfico de eficiência em concorrência x sintaxe simples.



Fonte: Marcelo Cavalcante (2019)

Figura 3. Gráfico de performance e eficiência.



Fonte: Marcelo Cavalcante (2019)

Go não traz uma série de recursos de linguagens orientadas a objetos de forma intencional, ou seja, não possui classes, utiliza apenas structs, não suporta heranças, tornando o entendimento do código mais simples, não possui construtores, anotações, genéricos e exceções. Essas características fazem com que seja quase tão performático e eficiente quanto C/ C++, porém muito mais simples.

AQUISIÇÃO De DADOS

Para a aquisição dos dados utilizados no sistema, foi utilizado um framework web denominado Flask para a construção de uma aplicação web cujo objetivo é a de realizar consultas em API's que disponibilizam dados publicamente e encaminhá-los continuamente às outras camadas do sistema.

Uma das API's consultadas e contida no escopo deste projeto, é a News API, uma API agregadora de notícias que tem como recurso uma API REST que retorna dados de manchetes relevantes com base em uma consulta. A partir dela, é possível obter como resposta as principais fontes de notícias do conteúdo buscado, assim como tipos específicos de mídia.

Para a News API são obtidos os seguintes dados: as notícias mais relevantes a partir de uma palavra-chave, a fonte de cada notícia, o título, um resumo do conteúdo, a URL, e o conteúdo do texto para cada notícia.

A segunda fonte de dados utilizados no sistema é a do Google Trends, sendo este um website que fornece dados com as tendências de crescimento de assuntos ao longo do tempo, os assuntos relacionados a um tema pesquisado, e quantidade de buscas realizadas para diferentes regiões em um determinado espaço de tempo. Os dados obtidos a partir da consulta ao Google Trends são: a quantidade de buscas realizadas por uma palavra-chave para cada estado no Brasil; o comportamento do interesse ao longo do tempo pelo tempo buscado; palavras relacionadas ao tema buscado por ordem de relevância.

PRÉ-PROCESSAMENTO DOS DADOS

O pré-processamento é um conjunto de atividades que envolvem preparação, organização e estruturação dos dados. Trata-se de uma etapa fundamental que precede a realização de análises e previsões.

Ao se trabalhar com dados de texto, o pré-processamento é uma etapa essencial antes dos dados estarem prontos para análise. Por natureza, a maioria dos dados em formato de texto é altamente desestruturado e cheio de ruídos, não sendo possível identificar uma organização clara. Para gerar insights sobre estes dados é preciso realizar um intenso pré-processamento para recuperar a informação e para construir melhores algoritmos.

Uma das primeiras etapas realizada foi a filtragem dos dados originais em séries mais adequadas para o processo visualização. Como primeiro item, foi realizada a seleção de atributos. Nesta ação, atributos de interesse foram selecionados para compor o conjunto de dados de análise, bem como novos atributos foram gerados a partir do conjunto de atributos fornecido.

A etapa seguinte foi a remoção de caracteres HTML, uma vez que os dados obtidos a partir da web normalmente contém uma grande quantidade de entidades HTML como & Lt; & Gt; & Amp; que vêm embutida nos dados originais. Assim, foi necessário remover tais entidades.

Outra parte do processo foi a decodificação dos dados, realizando a transformação de símbolos complexos para caracteres simples e mais fáceis de entender. Dados de texto podem estar sujeitos a diferentes formas de decodificação como “Latin-1”, “UTF-8”, etc. Portanto, para melhor análise, foi necessário manter os dados completos no formato de codificação padrão UTF-8.

Foi realizada a remoção e a substituição de determinadas pontuações, para que todas as marcas de pontuação fossem tratadas de acordo com as prioridades.

Por exemplo: “”, “?”, “,”, são pontuações importantes que foram retidas, enquanto outros, por exemplo “\”, “\”, ou pontuações sem sentido gramatical como “, \. ! ? :]”, foram removidas.

A abordagem utilizada para as etapas destacadas foi a através da utilização de expressões regulares específicas.

Para geração de um tipo de informação denominada *WordCloud*, foi feita uma etapa de pré-processamento adicional, na qual foi realizada a remoção de palavras não-significativas, uma vez que a análise de dados precisa ser orientada a dados no nível de palavra que acrescentam alguma informação ao estudo. Foi criado, assim, uma lista de palavras alimentada manualmente, assim como o uso de bibliotecas de *stopwords* previamente implementadas e específicas de um idioma pré-definido.

MODELO DE PREDIÇÃO

Neste estudo é realizada a proposta para a construção de um modelo preditivo o qual apresenta qual será o comportamento futuro da tendência de um determinado assunto ao longo do tempo.

Para isto, foi utilizado um modelo autorregressivo estatístico denominado ARIMA, sendo esta, uma sigla que significa *AutoRegressive Integrated Moving Average*. Este modelo captura um conjunto de diferentes estruturas temporais em dados de série temporal (BROWNLEE, 2017).

A diferença entre modelos regressivos e autorregressivos é que os primeiros preveem o valor de uma variável de interesse usando uma combinação linear (equação) das variáveis explanatórias. Já os segundos usam uma combinação linear de valores passados da própria variável (Gonzaga, S. T., 2019). Para este estudo, é avaliada apenas uma variável: a variação da tendência de um assunto ao longo do tempo, e desta forma é utilizado um modelo autorregressivo.

Os modelos ARIMA baseiam-se no pressuposto de que os valores anteriores têm algum efeito residual nos valores atuais ou futuros. Exemplificando o ARIMA no contexto deste artigo, uma pessoa que usa esta modelagem para prever a tendência de crescimento/decréscimo de um determinado produto presumirá que os novos compradores/vendedores deste produto são influenciados por transações recentes de mercado.

Em outras palavras, este é um método para prever resultados futuros com base em uma série histórica. É baseado no conceito estatístico de correlação serial, onde os pontos de dados passados influenciam os pontos de dados futuros (Prabhakaran, 2021).

Resumidamente, segundo Brownlee (2017) os componentes da sigla ARIMA significam:

AR: Autorregressão. Um modelo que usa a relação dependente entre uma observação e um certo número de observações defasadas.

I: integrado. O uso de diferenciação de observações brutas (por exemplo, subtraindo uma observação de uma observação na etapa de tempo anterior) a fim de tornar a série temporal estacionária.

MA: Média móvel. Um modelo que usa a dependência entre uma observação e um erro residual de um modelo de média móvel aplicado a observações defasadas.

Para realizar a modelagem dos dados temporais, foi utilizado o ARIMA implementado pela biblioteca *StatsModel*. Para utilizar o modelo desta biblioteca, é necessário fornecer parâmetros específicos da modelagem, os quais são definidos como:

O número de observações de defasagem incluídas no modelo, também chamado de ordem de defasagem ou lags.

O número de vezes que as observações brutas são diferenciadas, também chamado de grau de diferenciação.

O tamanho da janela da média móvel, também chamada de ordem da média móvel.

Com isso, um modelo autorregressivo é construído incluindo os parâmetros fornecidos, e os dados são preparados por um grau de diferenciação a fim de torná-los estacionários, ou seja, para remover estruturas de tendência e sazonais que afetam negativamente o modelo de regressão.

Um ponto fundamental para a construção deste modelo é a característica estacionária dos dados, uma vez que o termo 'autorregressivo' em ARIMA significa que este é um modelo de regressão que usa seus próprios atrasos como preditores (Prabhakaran, 2021).

Tais modelos funcionam melhor quando os preditores não estão correlacionados e são independentes uns dos outros. Assim, a abordagem mais comum para tornar os dados estacionários é diferenciá-los, ou seja, subtrair o valor anterior do valor atual. Este valor representa, portanto, o número mínimo de diferenciação necessária para tornar a série estacionária (BROWNLEE, 2017).

Após a construção do modelo de predição, é obtido como resultado os valores estimados de crescimento ou decréscimo de tendência dos dados para 12 meses à frente, juntamente com os valores de erro médio apresentados pelo modelo.

MODELAGEM DO BANCO DE DADOS

O banco de dados se tornou uma peça muito valiosa nos sistemas desde sua criação e o mais comum de armazenar os dados é no modelo tradicional do banco de dados relacional, devido a sua baixa complexidade.

Utilizado o PostgreSQL, que é um sistema de banco de dados relacional que usa e estende a linguagem SQL, além de, possuir muitos recursos destinados a ajudar os desenvolvedores a construir aplicativos, administradores para proteger a integridade dos dados e construir ambientes tolerantes a falhas, e ajudá-lo a gerenciar seus dados, não importa o tamanho do conjunto de dados.

SERVIÇO DE ENVIO DE EMAILS

O e-mail é um dos métodos de comunicação escrita mais amplamente usados, especialmente para empresas. Seja para compartilhar informações de marketing, registros transacionais ou qualquer outra informação.

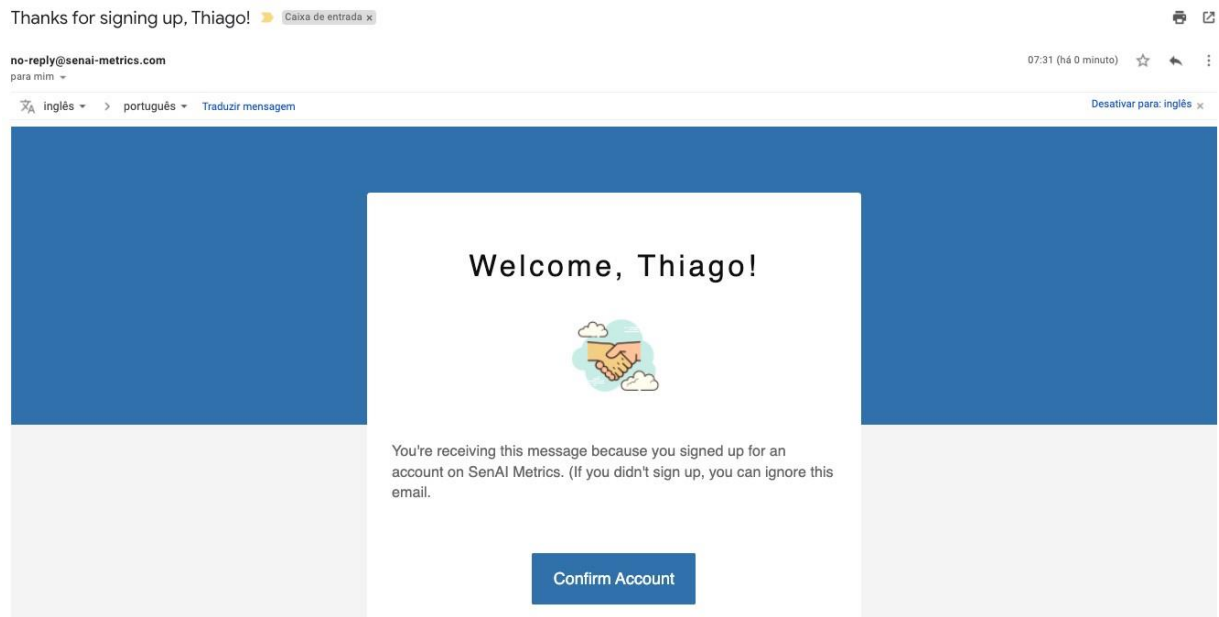
Foi realizada a implementação de um serviço de e-mails, cujos objetivos eram o de realizar a validação do e-mail de cadastro de usuário, e para a comunicação automatizada com os usuários a fim de realizar o envio de conteúdos de interesse.

Essa implementação foi realizada utilizando o provedor SendGrid, o qual dispõe de um serviço baseado em nuvem que permite enviar e-mail sem ter que manter servidores de e-mail.

Conforme descrição da própria empresa, o SendGrid fornece o gerenciamento de detalhes técnicos, como o dimensionamento da infraestrutura e o alcance do provedor de internet. Baseia-se na tecnologia SMTP, ou Simple Mail Transfer Protocol, sendo o método pelo qual os servidores da internet enviam mensagens de e-mail.

O serviço de cadastro e recebimento de e-mails segue demonstrado na figura seguinte, e necessitará de futuras implementações no corpo e design no texto.

Figura 4. E-mail de validação da conta após cadastro.



Fonte: O autor (2021)

APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

REQUISITOS DO SISTEMA

Após a aplicação da metodologia ágil conhecida como Scrum, foram levantados os requisitos funcionais e não-funcionais do sistema, os quais podem ser visualizados na Tabela 1. Estes requisitos são fundamentais para validar que o projeto entregue está em conformidade com o esperado.

Tabela 1: Requisitos Funcionais e não-funcionais do SenAI Metrics.

TIPO (RQF e RQNF)	Requisito
RQF = Requisito Funcional	
RQNF = Requisito não funcional	

RQF	Fornecer assuntos por nível de popularidade, através de um filtro.
RQF	Fornecer notícias dos principais veículos de mídia por meio de uma busca por uma palavra-chave.
RQF	Fornecer métricas sobre determinado assunto após a seleção das palavras-chaves na interface.
RQF	Fornecer métrica sobre a tendência de evolução de determinado tópico (palavrachave) com base em um modelo de predição de dados.
RQF	A ferramenta de assuntos permite rastrear a popularidade do tema ao longo do tempo, gerando gráficos que demonstram o interesse no tema da pesquisa ao longo do período selecionado.
RQF	Fornecer a possibilidade de cadastro na plataforma.

RQF	Fornecer a possibilidade da visualização constante das métricas sobre os temas préselecionados no cadastro individual na plataforma.
RQF	Fornecer um serviço de e-mail que possibilite envio de relatórios semanais/mensais.
RQF	Fornecer a possibilidade da visualização constante das métricas sobre os temas préselecionados no cadastro individual na plataforma.

RQF	Fornecer histórico de pesquisa ao usuário, por meio da persistência dos dados de busca.
RQF	Fornecer como tela inicial da aplicação um dashboard das palavras-chaves mais pesquisadas no site.
RQF	Fornecer busca personalizada para o usuário cadastrado.
RQF	Fornecer análise preditiva sobre a curva de tendencia para uma palavra-chave. (avaliar viabilidade)
RQNF	Utilizar o SendGrid como serviço de e-mail.
RQNF	Criar um domínio próprio que permita a configuração de zonas de DNS para o serviço do SendGrid.
RQNF	Os filtros selecionados serão aplicados por meio de uma consulta a uma API de agregação de assuntos.
RQNF	Utilização de uma base de dados para armazenar as consultas feitas na API para futuras métricas, uma vez que as API's só retornam os dados dos últimos 30 dias.
RQNF	Utilização de um banco de dados relacional.
RQNF	O sistema será subdividido em 2 servidores, um em Python e outro em NodeJs.

RQNF	Todo o ecossistema será inserido em um ambiente na nuvem, isso inclui o gerenciamento de versão do sistema, quadro de atividades, banco de dados e outros.
RQNF	Armazenamento de documentos, como imagens de wordcloud e perfil do usuário em um storage em cloud.

Fonte: O autor (2021)

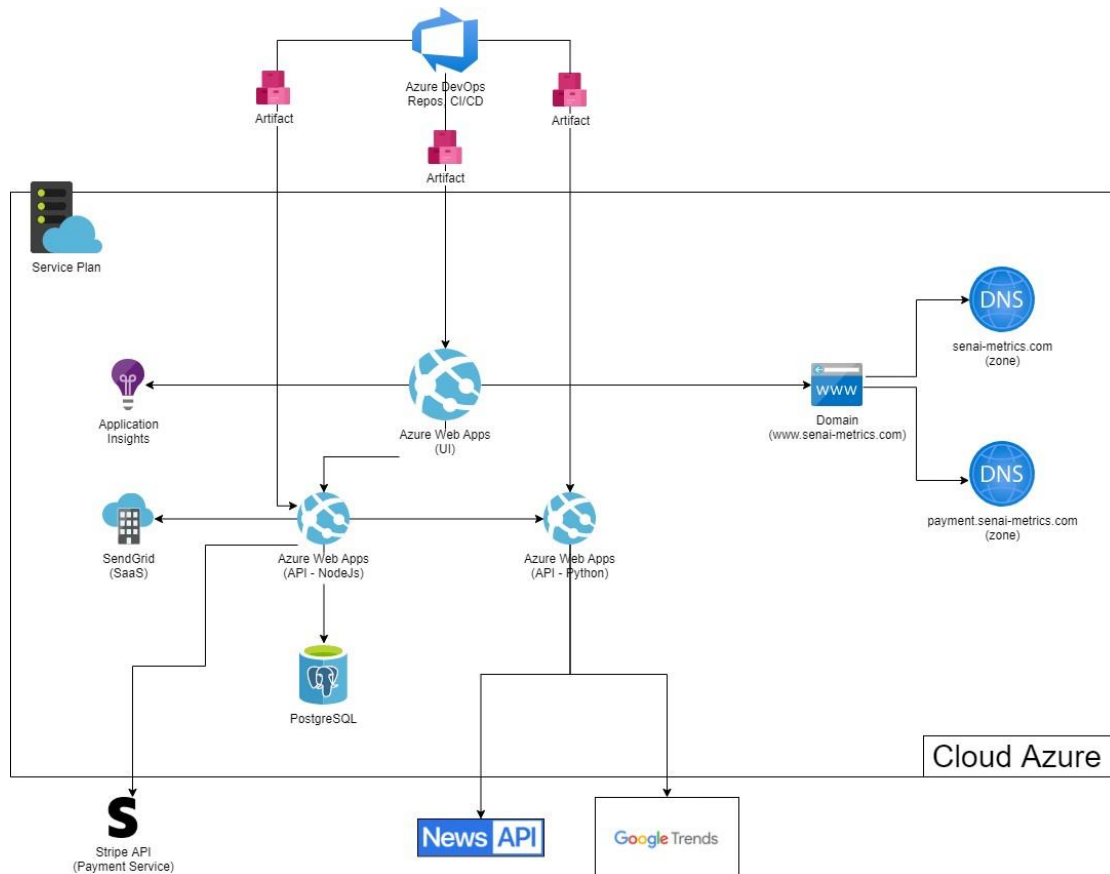
ARQUITETURA E TECNOLOGIAS

A ferramenta foi projetada para trabalhar em um ambiente de nuvem, executando seus serviços dentro de containers em Docker e uma base de dados altamente flexível e escalável.

A tecnologia Docker faz o uso do kernel e recursos do Linux para dissociar processos. Assim, podendo ser executados de maneira independente. O objetivo dos containers é criar a possibilidade de executar diversos processos e aplicações separadamente para aproveitar melhor a infraestrutura e, ainda assim, manter a segurança que normalmente se têm em sistemas separados.

Para facilitar o entendimento, a seguir, é apresentado a arquitetura que foi criada para manter os serviços em nuvem. O código fonte das aplicações fica armazenadas no sistema de versionamento do Azure Repos e o processo de deploy é realizado através dos pipelines, gerando artefatos que são inseridos em containers nos Azure Web Apps, conforme ilustra a figura 1.

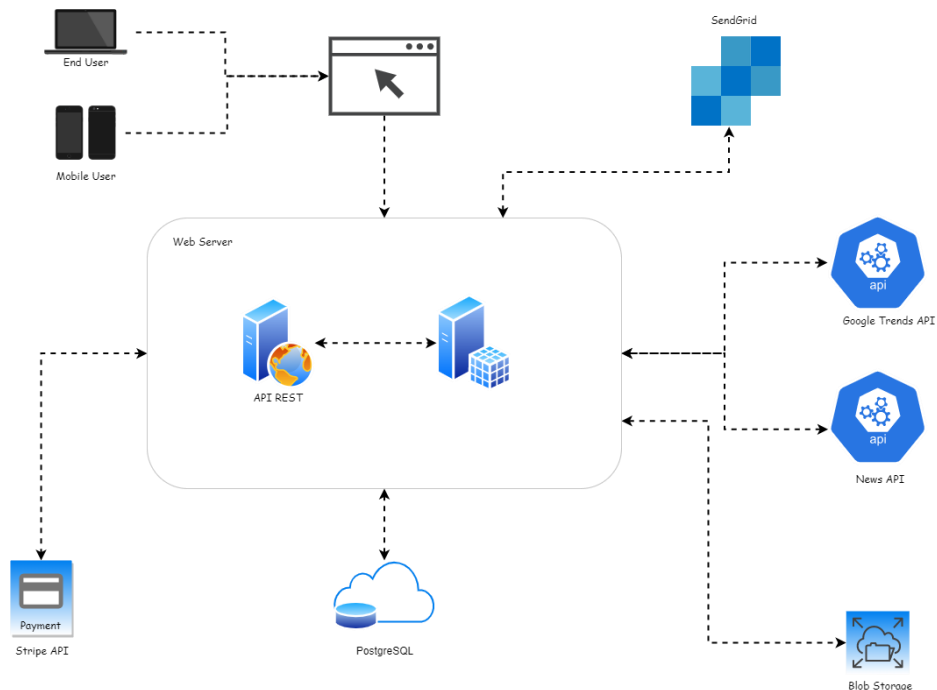
Figura 5: Arquitetura do Azure Cloud do SenAI Metrics.



Fonte: O autor (2021)

O sistema é separado em dois serviços no contexto do cliente-servidor, sendo um deles utilizando a linguagem de programação Python com o framework Flask. Já o segundo serviço em NodeJS – sendo definido como um ambiente de execução Javascript. Na Figura 1, é apresentada a arquitetura do sistema desenvolvido, evidenciando o uso de camadas, seguindo o modelo MVC.

Figura 6: Arquitetura do sistema SenAI Metrics.



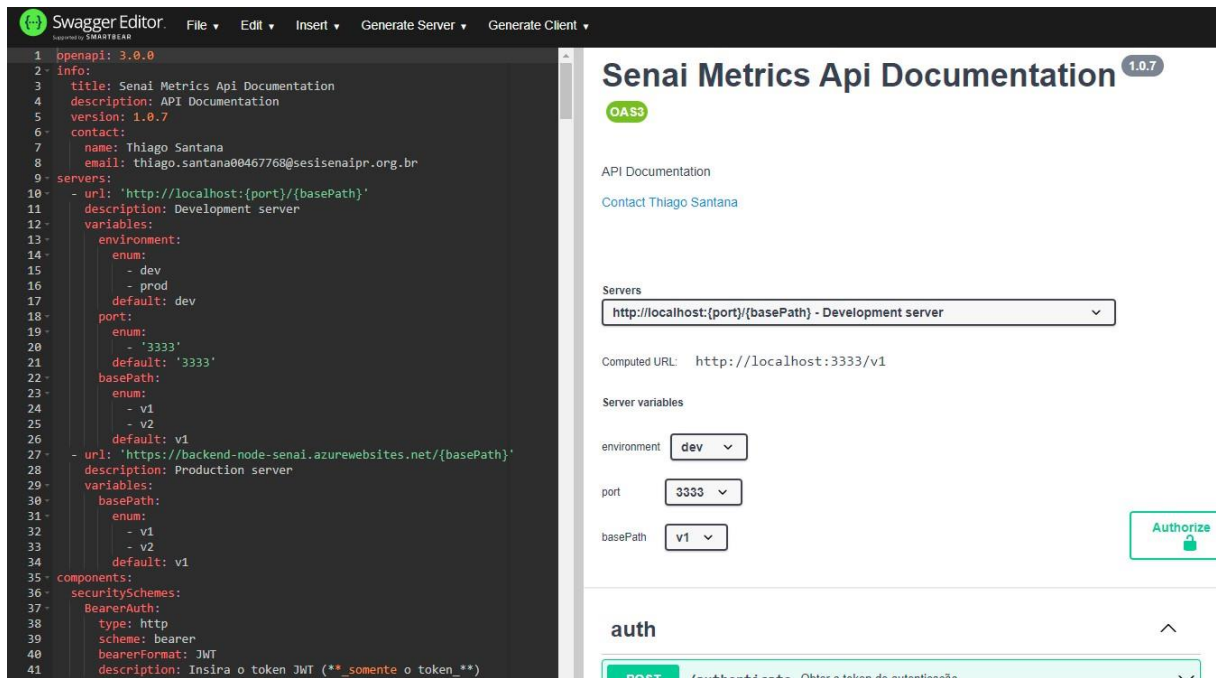
Fonte: O autor (2021)

Como é possível notar na figura acima, há um serviço de pagamento sendo utilizado para ficar responsável pelas cobranças das assinaturas mensais disponibilizadas. A Stripe oferece um conjunto de API e ferramentas programáveis que facilitam pagamentos e repasses globais para vendedores. Essa ferramenta garante a segurança no pagamento e no armazenamento das informações, sendo que a aplicação SenAI Metrics irá apenas armazenar o histórico de pagamento para que o usuário possa visualizar isso de maneira simples em seu *dashboard*.

A responsabilidade de realizar o processamento e coleta dos dados do Google Trends e da NewsAPI é por parte do Python, enquanto o NodeJS é responsável por fazer a integração entre a interface gráfica e o Python. Com a separação dos ambientes, a aplicação pode se tornar escalável com mais facilidade e torna-se possível a migração para um modelo de *microservices*. Com relação aos documentos que forem gerados na plataforma, serão armazenados temporariamente em um *storage* na nuvem, visto que há uma rotina de expurgo na aplicação.

Para facilitar na documentação e na integração da aplicação com diversas linguagens de interface gráfica, é utilizado o Swagger. O Swagger é uma linguagem de descrição de interface para descrever APIs RESTful expressas utilizando o JSON.

Figura 7: Descrição dos endpoints da aplicação pelo editor do Swagger.



Fonte: O autor (2021)

Na figura 2, o editor fornece a criação de uma API RESTful em um formato de arquivo YAML, mas que pode ser exportado em arquivo do tipo JSON, para facilitar na integração. Na figura 3, é apresentada alguns *endpoints* que a aplicação disponibiliza para a interface gráfica. Os *endpoints* são endereços mapeados que fornecem um recurso sempre com o mesmo formato, variando apenas seus valores.

Figura 8: Endpoints declarados na API RESTful do SenAI Metrics.

Autenticação	
POST	<code>/authenticate</code> Obter o token de autenticação

Usuário	
POST	<code>/user</code> Cadastro de usuários
GET	<code>/user</code> Obtém o perfil do usuário
PUT	<code>/user/{id}</code> Atualiza o perfil do usuário
POST	<code>/user/keyword</code> Armazena a palavra-chave
POST	<code>/user/verifyAccount</code> Verificar código de e-mail
POST	<code>/user/subscribe/{id}</code> Inscrição para recebimento de e-mails
POST	<code>/user/unsubscribe/{id}</code> Desinscrever do recebimento de e-mails

API Externa	
GET	<code>/metrics/suggestions</code> Sugestões de keywords mais utilizadas
GET	<code>/metrics/news-api</code> Consulta na NewsAPI
GET	<code>/metrics/google-trends</code> Consulta no Google Trends API

API Externa (Limitado)	
GET	<code>/metrics/free/news-api</code> Consulta na NewsAPI (Limitado)
GET	<code>/metrics/free/google-trends</code> Consulta no Google Trends API (Limitado)

Fonte: O autor (2021)

Conforme dito anteriormente, há dois tipos de banco de dados, relacional e não relacional. O banco de dados relacional está armazenando as informações de acesso, perfil do usuário e os dados das métricas selecionadas no cadastro para acompanhamento em período futuro. Na figura 4 é apresentado o esquema do objeto do tipo *User*, que é responsável pela autenticação e acesso a aplicação.

Figura 9: Declaração do objeto “Usuário” da aplicação SenAI Metrics.

```

User {
  _id                string($uuid)
  firstName          string
  lastName           string
  email              string
  cpf                string
  phone              string
  phone2             string
  login              string
  Login {
    username*        string
    password*        string
  }
  example: OrderedMap { "username": "admin", "password": "admin" }
}

```

Fonte: O autor (2021)

Para a parte visual da plataforma, foi utilizado o React - definido como uma biblioteca JavaScript para criação de interfaces SPA (Single Page Application), melhorando a usabilidade do usuário - juntamente com TypeScript.

O TypeScript começou a ser desenvolvido pela Microsoft em 2012, com o objetivo de adicionar recursos e ferramentas que não estão presentes nativamente na linguagem (ou que seriam muito mais complexos de serem implementados), como tipagem estática (ou seja, os tipos das variáveis são definidos explicitamente no código) e orientação a objetos (TECNOBLOG, página da internet).

4.3. EVOLUÇÃO DA APLICAÇÃO

Ao longo do desenvolvimento do projeto, algumas alterações e implementações foram acrescentadas ao escopo.

A proposta inicial do projeto se tratava da implementação de uma aplicação web, que permitiria um usuário a criar uma conta de acesso, e através de seu login, realizar consultas na plataforma. Essa consulta consistia na busca por palavras-chave, e a partir dessa consulta, a aplicação mostraria ao usuário as métricas relacionadas ao assunto pesquisado.

O protótipo inicial da aplicação realizava consulta somente a uma API, utilizando o NewsAPI para a aquisição de dados. Em etapa posterior foi inserido no escopo uma

nova fonte de dados, sendo esta, a API do Google Trends para aquisição de novas informações para o sistema.

Durante a evolução da segunda fase do projeto, implementações foram desenvolvidas tornando a aplicação mais robusta. Para esta segunda etapa, foram inseridos um sistema de pagamentos, para que houvesse um plano mensal/anual para os usuários possuírem alguns benefícios além dos usuários que utilizarem a ferramenta de forma gratuita. Além disso, foi adicionado um serviço de e-mail para envio de relatórios, e-mail de confirmação de conta, entre outros.

CONSIDERAÇÕES FINAIS

Este trabalho se propôs, como objetivo geral, apresentar os dados mais comentados pela internet referente a um tema previamente informado de uma forma visual, sendo ela através de gráficos e imagens, de tal modo que a visualização mais dinâmica trouxesse aos empreendedores uma maneira de conhecer o mercado o qual estão investindo seus recursos.

Com isso, ao optar pelo uso da aplicação com frequência, as métricas geradas são baseadas em dados de grande escala e auxiliam em tomadas de decisão, assim como influência diretamente em campanhas publicitárias que possuem um foco em aumento das vendas e adquirir novos clientes.

REFERÊNCIAS

BROWNLEE, Jason. How to Create an ARIMA Model for Time Series Forecasting in Python. [S. l.], 9 jan. 2017. Disponível em: <https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/>. Acesso em: 30 set. 2021.

CAVALCANTE, Marcelo. Go ou Golang: Porquê Adotei Go Como Minha Linguagem Favorita. Disponível em: <https://blog.marcelocavalcante.net/go-ou-golangporqu%C3%AA-adotei-go-como-minha-linguagem-favorita/>. Acesso em: 01 dez. 2021.

CODESH. Go (Golang): conheça a linguagem criada pelo Google. Disponível em: <https://codelist.com/blog/candidates/carreiras/go-lang-developers/>. Acesso em: 01 dez 2021.

DUARTE, Luis. O que é Node.js e outras 5 dúvidas fundamentais. Disponível em: <https://www.luiertools.com.br/post/o-que-e-nodejs-e-outras-5-duvidasfundamentais/#5>. Acesso em: 01 dez. 2021.

GOOGLE TRENDS. Veja o que o mundo está pesquisando. Disponível em: <https://trends.google.com/trends/?geo=BR>. Acesso em: 25 mai. 2021.

GONZAGA, Sillas Teixeira. Curso de Séries Temporais. [S. l.], 7 jun. 2019. Disponível em: http://sillasgonzaga.com/material/curso_series_temporais/index.html. Acesso em: 21 out. 2021.

GUEDES Marylene. Metodologias ágil x tradicional: Quais as diferenças?. Disponível em: <https://www.treinaweb.com.br/blog/metodologias-agil-x-tradicionalquais-as-diferencas>. Acesso em 31 mai. 2021.

HOLISTIC SEO. PyTrend Guideline: Create Dashboard for Google Trends with Python. Disponível em: <https://www.holisticseo.digital/python-seo/google-trends/>. Acesso em: 25 mai. 2021.

INDICADORES Brasileiros para os Objetivos de Desenvolvimento Sustentável. [S. l.], 1 jan. 2021. Disponível em: <https://odsbrasil.gov.br/>. Acesso em: 30 set. 2021.

INSTITUTO QUALIBEST. Campanha Publicitária: O que sua empresa deve considerar na criação?. Disponível em:

<https://www.institutoqualibest.com/marketing/campanha-publicitaria-o-que-suaempresa-deve-considerar-na-criacao/>. Acesso em: 12 jun. 2021.

NEWS API. Search worldwide news with code. Disponível em: <https://newsapi.org/>. Acesso em: 25 mai. 2021.

O QUE é Docker?. [S. I.], 7 jan. 2018. Disponível em: <https://www.redhat.com/ptbr/topics/containers/what-is-docker>. Acesso em: 1 dez. 2021.

PRABHAKARAN, Selva. ARIMA Model – Complete Guide to Time Series Forecasting in Python. [S. I.], 22 ago. 2021. Disponível em: <https://www.machinelearningplus.com/time-series/arima-model-time-seriesforecasting-python/>. Acesso em: 21 out. 2021.

POSTGRESQL. Disponível em: <https://www.postgresql.org/about/>. Acesso em: 13 mai. 2021.

SAS. How to capitalize on your data: It's not as hard as you think. Disponível em: https://www.sas.com/en_ph/customers/axis-bank-india-capitalize-data.html. Acesso em: 25 mai. 2021.

SCHWABER, Ken; SUTHERLAND, Jeff. Um guia definitivo para o Scrum: As regras do jogo. Jul. 2013. Disponível em: <https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>. Acesso em: 31 mai. 2021.

SERASA EXPERIAN. Ir à falência: veja os principais motivos e como evitá-los. Disponível em: <https://empresas.serasaexperian.com.br/blog/ir-a-falencia-veja-osprincipais-motivos-e-como-evita-los/>. Acesso em: 12 jun. 2021.

SCRUM. Disponível em: <https://www.desenvolvimentoagil.com.br/scrum/>. Acesso em 31 mai. 2021.

TECNOBLOG. O que é TypeScript? [Guia para iniciantes]. Disponível em:
<https://tecnoblog.net/426754/o-que-e-typescript-guia-para-iniciantes/>. Acesso em: 22
jun 2021.



