

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.

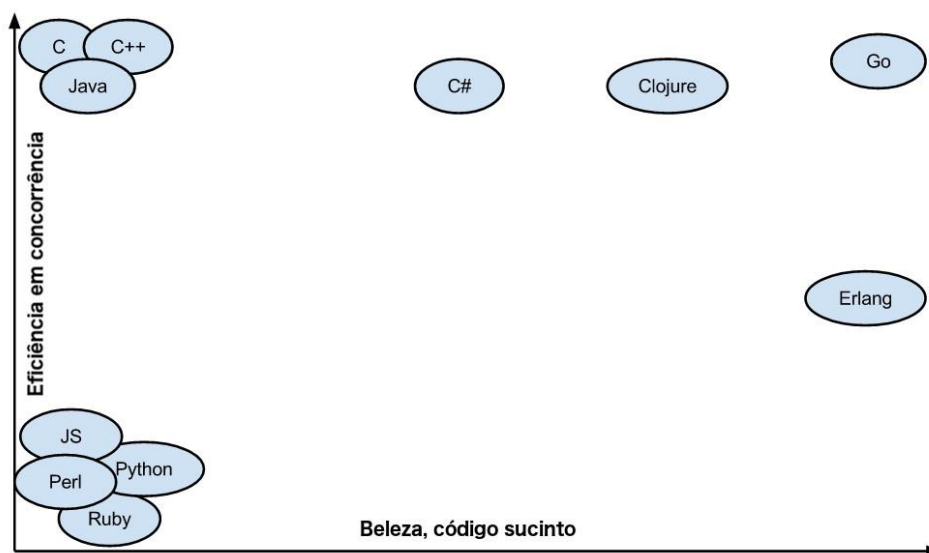
A curva de aprendizado entre Node e Go é bem parecida, mas como o NodeJs é basicamente JavaScript, sua curva tende a ser menor, porém Go é bem simples também, e desenvolvedores que já utilizaram Java, C#, C++ tem uma facilidade maior na sua compreensão.

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.

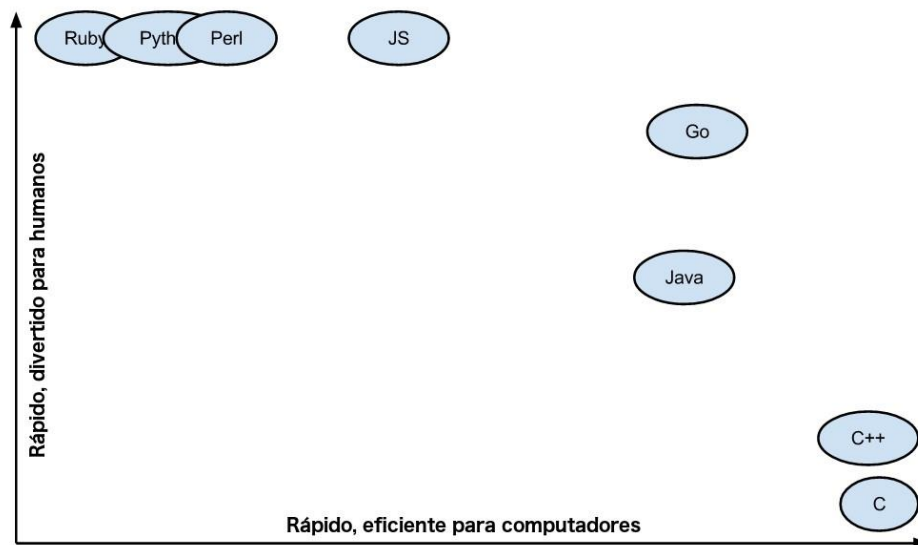
NodeJS tem frameworks muito populares e maduros, como Express, NestJS, LoopBack, etc. Enquanto o GO possui frameworks com uma popularidade menor, porém, muito eficientes, como o GIN, Fiber, etc.

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.

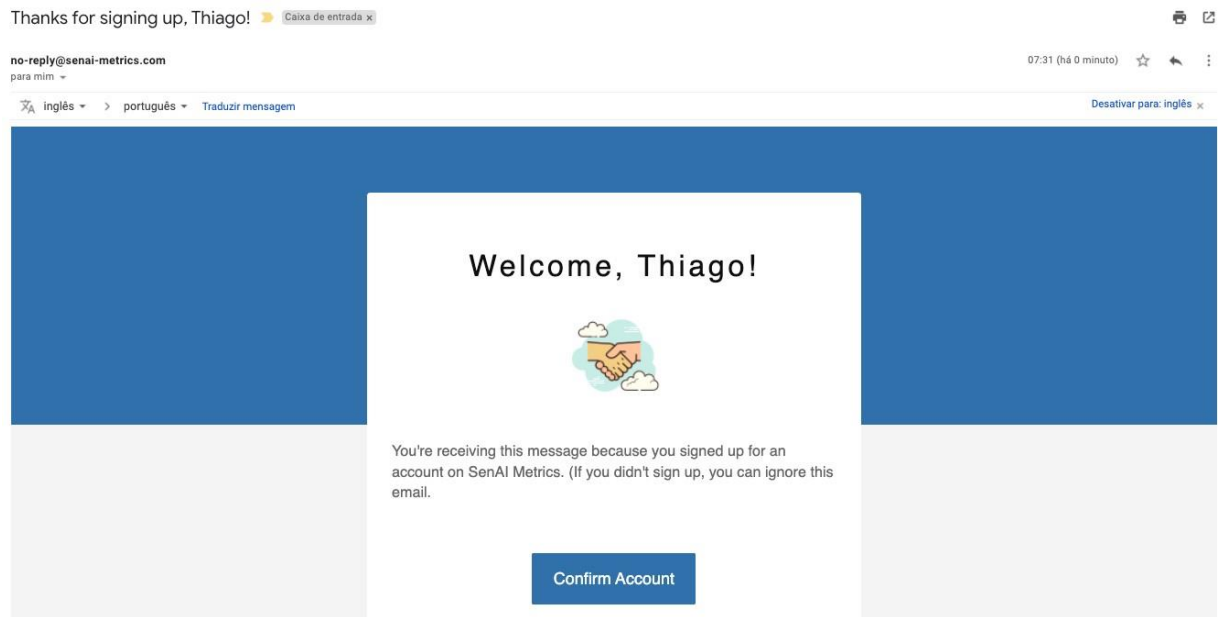
Foi realizada a implementação de um serviço de e-mails, cujos objetivos eram o de realizar a validação do e-mail de cadastro de usuário, e para a comunicação automatizada com os usuários a fim de realizar o envio de conteúdos de interesse.

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)

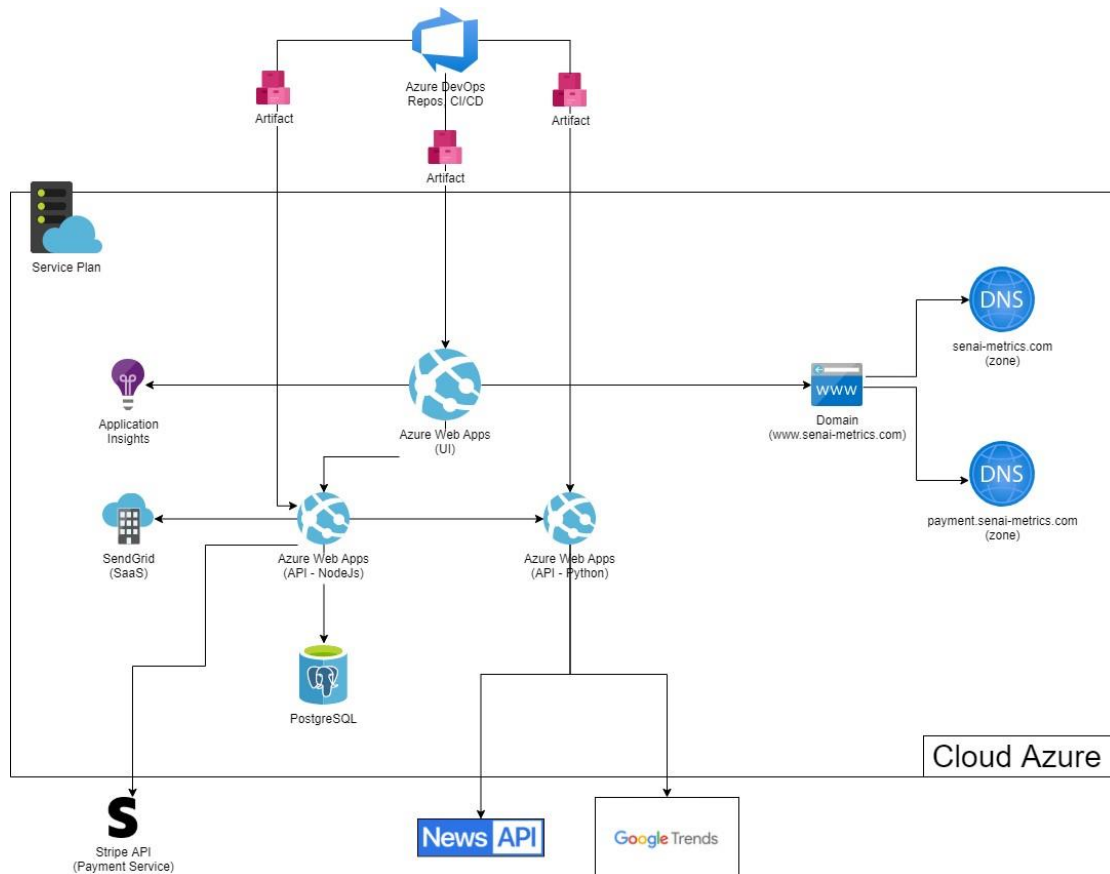
ARQUITETURA E TECNOLOGIAS

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.

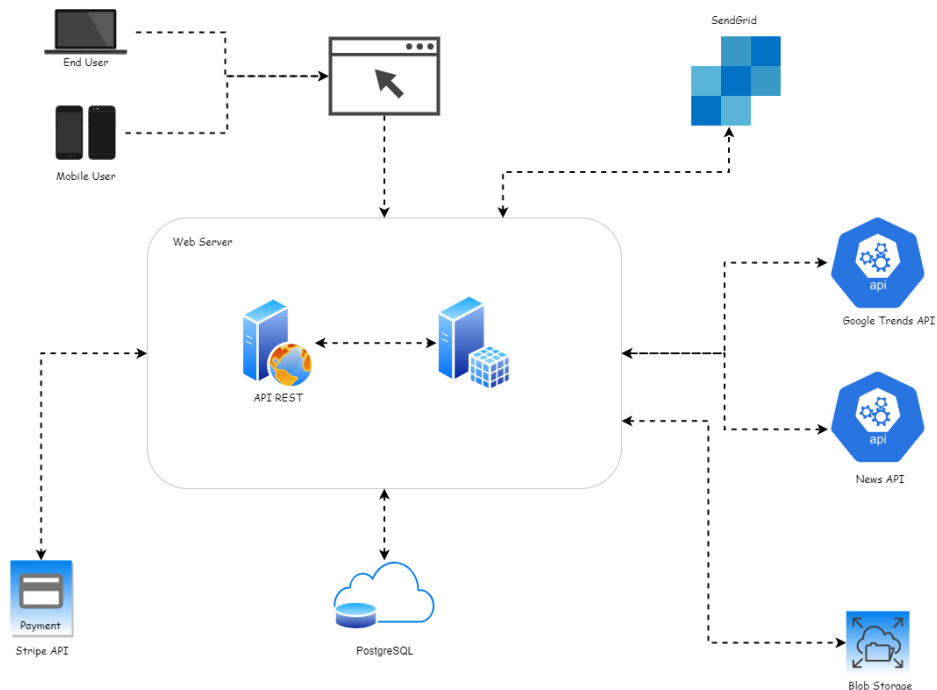
Figura 5: Arquitetura do Azure Cloud do SenAI Metrics.



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.



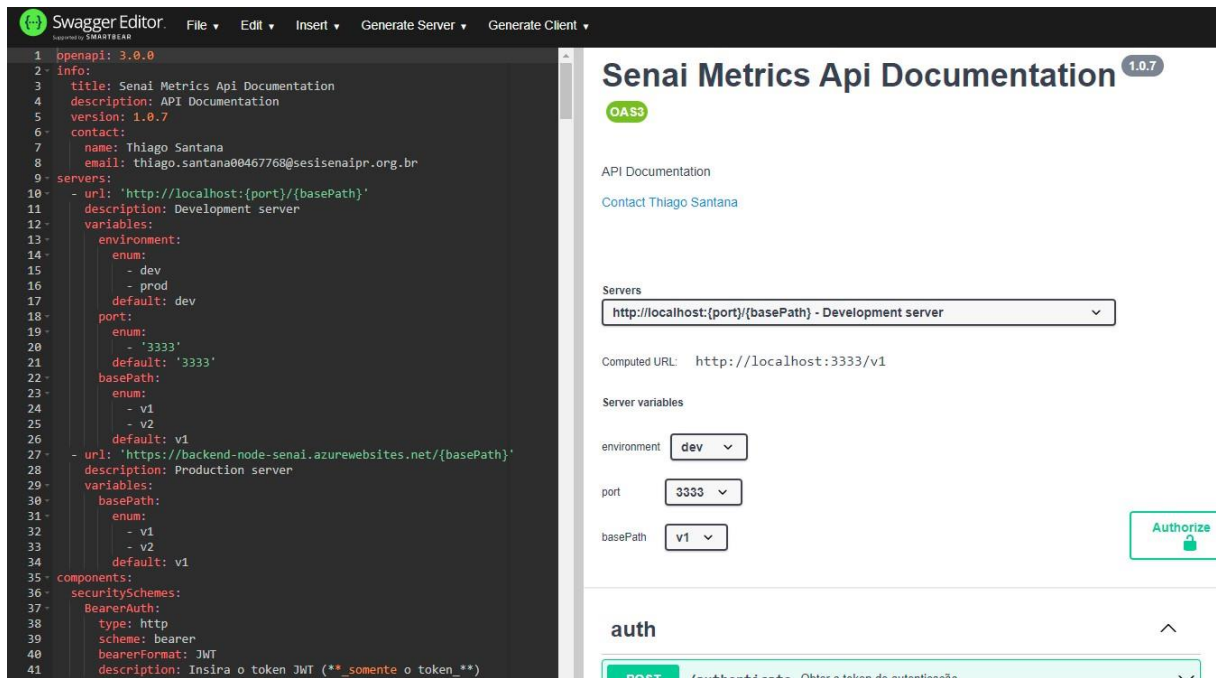
Fonte: O autor (2021)

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              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/arimafor-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.