Universidade Federal do ABC
Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas
Trabalho de Graduação em Engenharia de Informação

Implantação de Ferramenta de Geração de Leads para Organização Não Governamental

Jonas Santos Marma

Santo André

Outubro de 2022

Jonas Santos Marma

Implantação de Ferramenta de Geração de Leads para Organização Não Governamental

Trabalho de Graduação apresentado ao concluir o curso de Graduação em Engenharia de Informação, como parte dos requisitos necessários para a obtenção do Título Bacharel em Engenharia de Informação.

Universidade Federal do ABC

Orientador: Kenji Nose Filho

Santo André

Outubro de 2022

Resumo

Este trabalho se constituiu na proposição, desenvolvimento e implantação de uma ferramenta para a prospecção de potenciais doadores financeiros, também chamados de *leads*, para uma Organização Não Governamental (ONG). A ferramenta desenvolvida consiste em uma extensão de navegador para ser utilizada ao visitar perfis do *LinkedIn*. Essa extensão envia dados para um sistema que extrai informações do perfil e os armazena em um banco de dados, de forma a automatizar boa parte do processo, tornando-o mais rápido e menos propenso a erros.

Palavras-chaves: Webscrapping. Geração de leads. Organização Não Governamental.

Abstract

This work consisted of the proposition, development and implementation of a tool for prospecting potential financial donors, also called *leads*, for a Non-Governmental Organization (NGO). The developed tool consists of a browser extension to be used when visiting *LinkedIn* profiles. This extension sends data to a system that extracts information from the profile and stores it in a database, in order to automate much of the process, making it faster and less prone to errors.

Keywords: Webscrapping. Lead Generation. Non-Governmental Organization.

Lista de ilustrações

Figura 1 – Funil de vendas a ONG. À direita são mostrados os departamentos responsáveis pelas etapas de captação à direita. 2

Figura 2 – Planilha utilizada pelos arquitetos de conexões para o registro de *leads*. Os arquitetos de conexões faziam o preenchimento manual dos dados dessa planilha, um processo que demandava muito tempo e era propenso a erros. O o processo foi quase totalmente automatizado após a implementação do sistema. 2

Figura 3 – Padrão observer. 10

Figura 4 – Obtenção do *link* do perfil pelo *Sales Navigator*. Para obter o *link*, é necessário clicar no botão com reticências (a) e em seguida clicar no botão "Copiar URL do *Linkedin*.com"(b) para que ele seja copiado para a área de transferência. Analisando com o inspetor de elementos, como em (a), o botão e elementos próximos, não é possível encontrar o *link* do perfil. 12

Figura 5 – Fluxo recomendado ao arquiteto de conexões para gerar *leads*. Destacam-se as tarefas e decisões obrigatoriamente manuais, que devem ser feitos pelo arquiteto de conexões a cada vez que desejar gerar *leads*, as tarefas e decisões opcionalmente manuais, que o arquiteto pode optar por deixar que o *software* realize ou decidir realizar manualmente caso deseje ter mais controle, e as tarefas e decisões automáticas, que são realizadas sem intervenção humana. 15

Figura 6 – Diagrama de blocos do funcionamento do gerador de *leads*. 16

Figura 7 – Funcionamento da extensão 18

Figura 8 – Interface gráfica. Observa-se que há quatro modos de geração de *lead*: preenchimento e envio automático, preenchimento manual e envio automático, preenchimento automático e envio manual e preenchimento e envio manual. 20

Figura 9 – Exemplo de aviso exibido caso seja detectado um erro no preenchimento dos dados do *lead*. 21

Figura 10 – Aviso exibido caso o *lead* já tenha sido gerado. 21

Figura 11 – Sistema de login utilizando o *Google Identity Platform*. 22

Figura 12 – Esquema de renderização da página do gerador de *leads*. 23

Figura 13 – *Hooks* acionados para a atualização do *link* do *lead* e envio dos dados do mesmo caso o envio automático esteja ativado. 23

Lista de abreviaturas e siglas

ONG	Organização Não-Governamental
PF	Pessoa Física
LGPD	Lei Geral de Proteção de Dados
CRM	Customer Relationship Management
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
HTTP	Hypertext Transfer Protocol
URL	Uniform Resource Locators
API	Application Programming Interface

Sumário

1 INTRODUÇÃO . . . 1

2 FUNDAMENTAÇÃO TEÓRICA . . . 5
2.1 Conceitos atrelados ao marketing . . . 5
2.2 Legalidade do projeto . . . 5
2.2.1 Lei Geral de Proteção de Dados e Marco Civil da Internet . . . 6
2.2.2 Legalidade de *web scrapers* . . . 6
2.3 Tecnologias já utilizadas pela ONG . . . 7
2.4 Tecnologias utilizadas no projeto . . . 8
2.4.1 LinkedIn e Sales Navigator . . . 8
2.4.2 Tecnologias *Web* . . . 9

3 METODOLOGIA . . . 11
3.1 Objetivos . . . 11
3.1.1 Rapidez na geração de *leads* . . . 11
3.1.2 Interface gráfica . . . 12
3.1.3 Segurança de dados . . . 13
3.1.4 Compatibilidade e flexibilidade . . . 13
3.2 Utilização do ponto de vista do arquiteto de conexões . . . 14
3.3 Resumo do funcionamento . . . 16
3.4 Extensão de navegador . . . 17
3.5 API e banco de dados . . . 18
3.6 Interface gráfica . . . 19
3.7 Transferência dos dados para o Google Sheets . . . 24

4 CONCLUSÕES E TRABALHOS FUTUROS . . . 25

5 REFERÊNCIAS . . . 26

1 Introdução

Em empresas, nos departamentos de captação de recursos, existe o conceito de funil de vendas. Um funil de vendas é uma representação visual, como a da Figura 1, em que todos os potenciais compradores, ou *leads*, são colocados no topo do funil e esses são divididos em outras classificações de acordo com o seu grau de intenção de compra até a base do funil, onde estão os compradores de fato. Este tipo de diagrama é chamado de funil de vendas porque cada etapa do processo é representada com uma área proporcional à quantidade de pessoas presentes na área e, naturalmente, a quantidades de pessoas que passam de uma etapa para outra é cada vez menor, o que afunila a representação.

O projeto discutido aqui foi desenvolvido com o objetivo de ajudar o departamento de captação de recursos de uma Organização Não-Governamental (ONG) sem fins lucrativos. Nessa ONG o conceito do funil de vendas utilizado para modelar o processo de captação de doações. Contextualizando, a ONG tem como principal público jovens de baixa renda que conseguiram ingressar no ensino superior: tais jovens, após passar por um processo seletivo, recebem recursos financeiros mensais por 1 ano, treinamentos e acompanhamento de um mentor para ajudá-los a ingressar no mercado de trabalho em um ambiente saudável e que permita a continuação dos estudos. A ONG também promove outros eventos relacionados a empregabilidade para um público mais aberto.

A sustentação financeira para essa estrutura vem de parcerias com empresas e doações de pessoas físicas. Existe dentro da ONG um departamento chamado de "*Hunting* de Pessoa Física", ou "*Hunting* PF", que faz a busca por doadores com alto potencial de doação (valores acima de R$1000,00 em 2022). Os potenciais doadores, ou *leads*, são encontrados utilizando a rede social *LinkedIn* [1] e convidados para atividades de engajamento onde o trabalho da ONG é apresentado, contribuindo para uma possível doação financeira. O departamento de *Hunting PF* se divide em três especialidades:

1. **Arquitetos de conexão:** responsáveis por analisar sistematicamente funcionários de empresas no *LinkedIn* e selecionar *leads* entre eles.

2. **Executivos de contas:** responsáveis por se comunicar com os *leads* encontrados pelos arquitetos de conexões, fazer o agendamento de reuniões e otimizar a utilização do tempo dos desenvolvedores de conexões.

3. **Desenvolvedores de conexões:** responsáveis por conduzir as reuniões com os *leads*, convencendo-os do valor social e credibilidade da ONG e pedindo as doações.

Como mostrado na Figura 1, essa estrutura de captação do departamento de

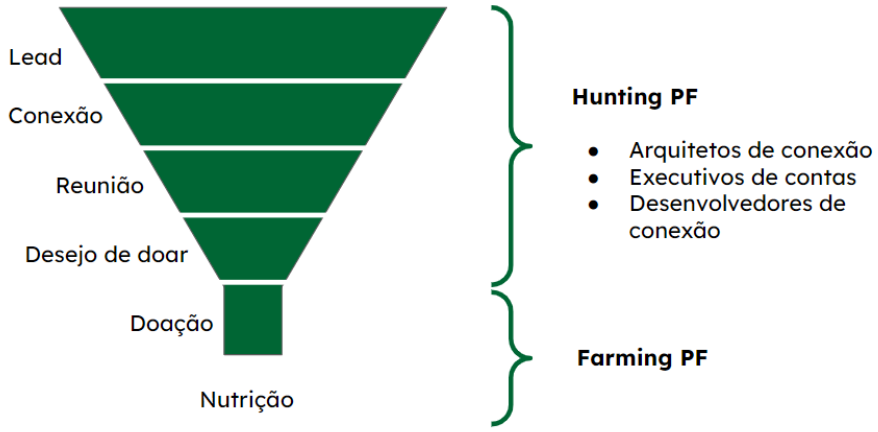

Figura 1 – Funil de vendas a ONG. À direita são mostrados os departamentos responsáveis pelas etapas de captação à direita.

	A	B	C	E	F	G	H	I	J	K	L	M
1	Arq	Data de geração	Projeto	LinkedIn	Nome	Empresa	Setor	Nº de funcionários	Cargo	Vículo Semear	Dev	Observações
75	Maria Fernanda	16/02/2022	Projetos_2022	https://www.lir	Mart		Automobili	Mais de 5000.	1.Cordenador	Fora da rede Semear	Janyne	
76	Maria Fernanda	16/02/2022	Projetos_2022	https://www.lir	Fern		Automobili	Mais de 5000.	2. Líder	Fora da rede Semear	Janyne	
77	Maria Fernanda	16/02/2022	Projetos_2022	https://www.lir	Brun		Automobili	Mais de 5000.	1.Cordenador	Fora da rede Semear	Janyne	
78	Maria Fernanda	16/02/2022	Projetos_2022	https://www.lir	Flavi		Automobili	Mais de 5000.	1. Supervisor	Fora da rede Semear	Janyne	
79	Maria Fernanda	16/02/2022	Projetos_2022	https://www.lir	Alexa		Automobili	Mais de 5000.	1. Supervisor	Fora da rede Semear	Janyne	
80	Maria Fernanda	16/02/2022	Projetos_2022	https://www.lir	Ricar		Automobili	Mais de 5000.	1. Supervisor	Fora da rede Semear	Janyne	
81	Maria Fernanda	16/02/2022	Projetos_2022	https://www.lir	Paul		Automobili	Mais de 5000.	2. Especialista	Fora da rede Semear	Marcos	
82	Maria Fernanda	16/02/2022	Projetos_2022	https://www.lir	Sand		Automobili	Mais de 5000.	1. Supervisor	Fora da rede Semear	Marcos	
83	Maria Fernanda	16/02/2022	Projetos_2022	https://www.lir	João		Automobili	Mais de 5000.	2. Especialista	Fora da rede Semear	Marcos	
84	Maria Fernanda	16/02/2022	Projetos_2022	https://www.lir	Mari		Automobili	Mais de 5000.	1.Cordenador	Fora da rede Semear	Marcos	
85	Maria Fernanda	16/02/2022	Projetos_2022	https://www.lir	Sergi		Automobili	Mais de 5000.	1.Cordenador	Fora da rede Semear	Marcos	
86	Maria Fernanda	16/02/2022	Projetos_2022	https://www.lir	Sergi		Automobili	Mais de 5000.	1.Cordenador	Fora da rede Semear	Marcos	
87	Maria Fernanda	16/02/2022	Projetos_2022	https://www.lir	Alyss		Automobili	Mais de 5000.	1. Nível 1	Fora da rede Semear	Marcos	
88	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		[illegible]	[illegible]	[illegible]	Fora da rede [illegible]	[illegible]	

Figura 2 – Planilha utilizada pelos arquitetos de conexões para o registro de *leads*. Os arquitetos de conexões faziam o preenchimento manual dos dados dessa planilha, um processo que demandava muito tempo e era propenso a erros. O o processo foi quase totalmente automatizado após a implementação do sistema.

Hunting PF é responsável por todo o funil de vendas e, ao final desse funil, quando o doador faz a doação, ele é enviado para um ciclo denominado de nutrição, em que ele passa por atividades de engajamento periódicas e exclusivas como forma de incentivo para fazer novas doações. Essas atividades são controladas por outro departamento, o e *Farming PF*.

O sistema desenvolvido teve como foco a melhoria do processo de procura de *leads* pelos arquitetos de conexões. Antes de sua implementação, todo o processo de procura por novos *leads* era feito manualmente por meio de leitura e registro em planilhas, como a mostrada na Figura 2, pelos arquitetos de conexões. O sistema desenvolvido permite o preenchimento automático dessa mesma planilha.

A busca por *leads* é feita de acordo com análises e hipóteses criadas pelos diretores do departamento de *Hunting* PF, que criam metas de captação semanais com base no contexto de captação da ONG e fazem segmentação sobre o tipo de *lead* a ser buscado, as

empresas às quais pertencem e o tipo de cargo que ocupam.

Em 2022, com 3 arquitetos que trabalhavam em média 4 horas por dia na geração de *leads*, a cada semana havia uma meta para que os arquitetos de encontrassem ao menos 200 *leads*. De acordo com dados do departamento de *Hunting* PF, desses 200, em média 15 respondem ao contato e marcam reuniões, e dessas reuniões, em média 3 fazem doações. Observou-se que os executivos de contas e desenvolvedores de conexões tinham capacidade para agendar e fazer mais reuniões por semana e com isso foi criada a hipótese de que a melhor estratégia para aumentar a arrecadação do departamento seria aumentar a taxa de geração de *leads*.

O objetivo específico do sistema desenvolvido foi, portanto, aumentar a taxa de geração de *leads* por arquiteto de conexões, possibilitando assim o que a meta de 200 *leads* por semana fosse aumentada sem que fosse necessário contratar mais colaboradores para a ONG. Além disso, desejava-se que essa nova forma de gerar *leads* fosse o mais confortável o possível aos arquitetos de conexões, e para tanto o projeto foi desenvolvido em constante conversa com eles.

De forma secundária, a manutenção do sistema exige conhecimentos de desenvolvimento *web*. A necessidade de manutenção desse sistema faz então com que parte dos voluntários e estagiários da ONG adquiram e apliquem esses conhecimentos, o que é benéfico tendo em vista a missão de inclusão no mercado de trabalho da ONG.

O projeto também faz parte de um esforço de reestruturação da ONG para manter seus dados seguros e em boa qualidade. Sendo assim, houve também uma preocupação em se utilizar tecnologias flexíveis, compatíveis com outras tecnologias, bem documentadas e populares para facilitar migrações e diminuir riscos. Foi desenvolvido dentro do departamento de inteligência de mercado da ONG, que tem a função de, com base em pesquisas em meios internos e externos, monitorar e propor mudanças em processos, estratégias de captação de recursos, estratégias de *marketing* e uso de ferramentas tecnológicas, como é o caso deste projeto.

A procura de *leads* para alimentar o topo do funil de vendas da ONG, como mencionado anteriormente, é feita utilizando a rede social *LinkedIn* [1]. O *LinkedIn* é uma rede social utilizada conectar profissionais e empresas. Uma das aplicações dessa rede social é a prospecção de *leads*, tal como é feito pela ONG. Para tanto, é utilizado o *Sales Navigator*, uma plataforma pertencente ao *LinkedIn* com ferramentas específicas, principalmente filtros, para ajudar na procura de perfis [1].

O sistema começa a agir com extensão de navegador que, ao detectar que o arquiteto de conexões está acessando um perfil do *LinkedIn* pelo *Sales Navigator*, extrai o código-fonte da página. Em seguida, a extensão envia esse código-fonte para uma aplicação *web* que o armazena e extrai as informações úteis desse perfil do *lead* no *LinkedIn*. Finalmente,

a aplicação *web* é acessada pelo arquiteto de conexões que pode analisar essas informações e disponibilizá-las para os demais membros da ONG.

No Capítulo 2 são discutidos os conceitos de marketing utilizados para algumas decisões durante o projeto, a legalidade do mesmo, especialmente quanto à manipulação de dados e as tecnologias já utilizadas pela ONG e as escolhidas para o projeto tendo em vista a compatibilidade entre tecnologias. No Capítulo 3 é discutido o histórico de versões do projeto e como ele evoluiu a partir de críticas recebidas e melhorias pensadas a partir dessas críticas. Finalmente, no Capítulo 4, são apresentadas as conclusões e perspectivas sobre trabalhos futuros.

2 Fundamentação Teórica

Neste capítulo serão discutidos alguns dos conceitos principais que guiaram este trabalho. Primeiramente, na Seção 2.1, é explicado o conceito de *lead* e como este se relaciona com outros conceitos atrelados ao *marketing*. Em seguida, como se trata de um projeto que lida com dados de terceiros, na Seção 2.2 são discutidas as questões legais quanto à interação com esses dados. Na Seção 2.3 são apresentadas as ferramentas tecnológicas relevantes para a compatibilidade do projeto desenvolvido com as tecnologias já utilizadas e, finalmente, na Seção 2.4, são elencadas as tecnologias escolhidas para o desenvolvimento do projeto.

2.1 Conceitos atrelados ao marketing

Um funil de vendas, como apresentado na Figura 1, é um modelo que separa possíveis clientes de uma empresa pelas etapas em que eles passam até se tornarem clientes efetivos. É chamado de funil porque faz uma analogia com a forma geométrica de um funil: no início, a parte mais larga do funil, estão todos os potenciais clientes, que passam por sucessivos filtros até os clientes efetivos, que são uma fração dos potenciais clientes.

Perto das primeiras camadas do funil, existe o *lead*, que é um potencial cliente mapeado, ou no nosso caso, um potencial doador. O processo de mapeamento desses *leads* é chamado de prospecção, e envolve obter os dados de contato desses *lead* para que ele seja acessado e informado sobre o produto. Esta é a primeira etapa do funil de vendas e é decisiva para a quantidade de pessoas que chegarão ao final do funil de vendas [2].

Vale observar que os conceitos apresentados aqui foram desenvolvidos tendo em mente a venda de produtos por empresas, especialmente em modelos empresa-cliente, mas esses conceitos, por serem bem gerais, podem ser utilizados com adaptações para doadores de uma ONG, como é o caso.

2.2 Legalidade do projeto

No Brasil, existem dois principais conjuntos de leis que regulam a aquisição, manipulação e armazenamento de dados pertinentes ao projeto: a Lei Geral de Proteção de Dados (LGPD) e o Marco-Civil da internet. Como o projeto girava em torno dos dados de potenciais doadores, foi necessário dar especial atenção a esses temas.

2.2.1 Lei Geral de Proteção de Dados e Marco Civil da Internet

As duas principais leis que deve-se ter em vista ao tratar dados no Brasil são a Lei Geral de Proteção de Dados (LGPD) [3], em vigor desde agosto de 2020, e o Marco Civil da Internet [4], em vigor desde 2014. É de interesse manter o cuidado com os dados dos *leads* porque esses podem ser classificados como dados sensíveis de pessoas físicas. Portanto, o projeto foi realizado com a supervisão de um advogado voluntário da ONG.

2.2.2 Legalidade de *web scrapers*

Web scraping, ou raspagem de dados, consiste na obtenção de uma grande quantidade de dados disponíveis na internet utilizando alguma forma de automação, geralmente com *bots*. É uma prática comum em projetos acadêmicos e corporativos baseados na análise de dados. O projeto sendo desenvolvido, apesar de ainda envolver muita interação humana, pode ser considerado como um *web scraper*.

Administradores de sites tendem a utilizar técnicas para tentar blindar seus sites à ação de web *scrapers* porque estes representam um tráfego que geralmente não gera valor ao seu negócio. Por exemplo: se um site é monetizado através da exibição de anúncios aos seus visitantes, não é de interesse ter visitas de um *bot* que faz *web scraping* porque há desperdício de banda sem que ninguém veja o anúncio. Isso faz com que empresas de publicidade monitorem as medidas tomadas para evitar a ação de *scrappers* pelos sites em que veicularão seus anúncios e prefiram sites mais resistentes.

Quando um site possui dados pessoais de indivíduos, tais como redes sociais, seus administradores são ainda mais encorajados a evitar a ação de *web scrapers* porque eles perdem a rastreabilidade desses dados, correndo o risco de terem complicações quanto às leis de proteção de dados. Até o momento, não existem legislações claras especificamente quanto à prática de web *scraping*, mas cabe aos praticantes fazer o uso dos dados de acordo com outras legislações a depender da aplicação [5]. No contrato de usuário em vigor para o ano de 2022 do *LinkedIn*, rede social de onde os dados deste projeto são extraídos, há uma seção intitulada "8. O que é permitido ou proibido no *LinkedIn*", de onde se destacam as seguintes cláusulas:

- 8.2. Você concorda que não irá desenvolver, dar suporte ou utilizar software, dispositivos, *scripts*, robôs ou quaisquer outros meios ou processos (incluindo *crawlers*, *plugins* e *add-ons* para navegadores ou quaisquer outras tecnologias) para fazer varredura nos Serviços ou copiar de outra forma perfis e outros dados dos Serviços;
- 8.4. Você concorda que não irá copiar, utilizar, divulgar ou distribuir quaisquer informações obtidas a partir dos Serviços, diretamente ou por

meio de terceiros (tais como ferramentas de pesquisa), sem o consentimento do *LinkedIn* [6];

A depender da interpretação, poderia-se afirmar que essas cláusulas não são totalmente cumpridas com o desenvolvimento do *software* proposto aqui e sabe-se que a consequência para tanto pode ser a suspensão de contas do *LinkedIn* e do *Sales Navigator* utilizadas para a busca dos perfis dos *leads*. Em casos mais graves de não cumprimento das cláusulas, podem haver sanções legais mais pesadas como multas, mas felizmente constata-se que há um risco baixo de acontecerem esses tipos de sanções por dois motivos:

1. Já existe no mercado uma grande quantidade de *webscrappers* para *LinkedIn* e outras redes sociais, sendo esse inclusive um nicho de mercado grande, o que dificultaria a aprovação de leis contra os mesmos.

2. Comparado com as atividades de *scrapping* maiores que já existem, o impacto do *scrapping* feito pela ONG é desprezível, o que também diminui a probabilidade de que o *LinkedIn* mova ações mais pesadas, já que não é um esforço que traria resultados pequenos.

3. Ainda que ocorra uma ação, é possível alegar boa-fé, tendo em vista a causa da ONG e a natureza sem fins lucrativos.

2.3 Tecnologias já utilizadas pela ONG

Nos primeiros anos da ONG, que foi fundada em 2010, havia apenas uma quantidade pequena de pessoas trabalhando: um grupo com menos de 20 voluntários que precisavam administrar também uma pequena quantidade de parcerias, doadores e beneficiados. Dessa forma, devido ao conhecimento já existente entre os membros e a comodidade de ter várias ferramentas compatíveis entre si, eles optaram pela utilização do *Google Workspace* para a realizar atividades como redação de documentos, envio de *emails*, reuniões e armazenamento de dados.

Após um projeto de consultoria bem-sucedido em 2016, houve, ano a ano, um crescimento quase exponencial do valor arrecadado em doações, do número de membros administrativos, da quantidade de beneficiados e, consequentemente, da quantidade de dados manipulada e armazenada. Tendo os dados em sua maioria ainda armazenados e manipulados utilizando o *Google Sheets*, um software do *Google Workspace* para a criação de planilhas, os erros de manipulação de dados cometidos por membros da organização começaram a se propagar porque a ferramenta não foi projetada para atuar com uma grande quantidade e variedade de dados.

Dessa forma, há um esforço sendo feito para migrar os dados para uma ferramenta mais robusta e escalável de armazenamento ao mesmo tempo em que os processos administrativos são adaptados para essas novas tecnologias. No momento, existe uma propensão para a criação de bancos de dados utilizando serviços do *Google*, principalmente o *Google Cloud Platform*, mas também existem considerações em se utilizar soluções de administração de relacionamento com cliente, ou do inglês, *Customer Customer Relationship Management* (CRM), da empresa *Salesforce*.

Esse cenário de mudanças exige que novas ferramentas, tais como a de geração de *leads* proposta aqui, sejam bem documentadas e flexíveis quanto às tecnologias utilizadas. Tais critérios foram levados em conta durante a escolha das tecnologias utilizadas e descritas a seguir.

2.4 Tecnologias utilizadas no projeto

2.4.1 LinkedIn e Sales Navigator

O *LinkedIn* é uma rede social utilizada pela ONG para a procura de *leads*. No *LinkedIn* pessoas criam páginas de perfil sobre elas mesmas em que relatam suas experiências profissionais, expressam suas opiniões, anunciam ou procuram vagas de emprego e se conversam com outras pessoas. Existe no *LinkedIn* uma ferramenta chamada de *Sales Navigator* [7], que é um produto pago, criado para profissionais de vendas e de *marketing*, que permite uma busca aprofundada e sistemática de perfis no *LinkedIn* a fim de encontrar oportunidades de negócio. A ONG utiliza o *Sales Navigator* para encontrar *leads*, segmentando-os por ramo de atuação, empresa em que trabalham, interesses e fazendo estimativas de seus salários para pedir doações de acordo. O processo de registro de *leads* encontrados utilizando o *Sales Navigator* era feito de forma manual pelo departamento de *Hunting PF*, que registrava esses dados em uma planilha. Os dados extraídos pela ONG sobre os *leads* são:

1. Link para a página de perfil no *LinkedIn* do *lead*;
2. Nome do *lead*;
3. Data de última atividade do *lead* no LinkedIn;
4. Empresa em que o *lead* trabalha atualmente;
5. Cargo que o *lead* ocupa na empresa;
6. Tipo de cargo, classificado de 1 a 5 em que 1 representa cargos totalmente operacionais e 5 representa cargos altos de diretoria (classificação criada pela ONG);

7. Proximidade com a ONG, em que o *lead* é considerado próximo se conhece uma pessoa que já foi bolsista ou trabalhou na ONG;

8. Email do lead.

2.4.2 Tecnologias *Web*

Páginas na Web são construídas com três linguagens principais: a Linguagem de Marcação de Hipertexto, ou *HyperText Markup Language* (*HTML*), a linguagem de Cascateamento de Folhas de Estilo, ou *Cascading Style Sheets CSS* e *JavaScript*. *HTML* é uma linguagem de marcação para a estruturação de páginas na *Web*, *CSS* é uma linguagem para estilização de páginas na *Web* e *JavaScript* é uma linguagem de programação utilizada para adicionar funcionalidades lógicas a sites. Isso faz do *JavaScript* uma linguagem de programação popular para aplicações relacionadas à Internet. Por exemplo, extensões de navegador, como a desenvolvida neste projeto, são programadas em *JavaScript* [8].

Node.js é um ambiente de execução para programas em *JavaScript* fora de navegadores de internet. A ferramenta se tornou uma ferramenta popular entre desenvolvedores para utilização do *back-end* de sites e parte dessa popularidade se deve à comodidade de se utilizar *JavaScript* tanto no *back-end* como no *front-end*. Por ser tão popular, existe atualmente uma grande variedade de ferramentas para desenvolvedores que orbitam o *Node.js* [9].

Next.js é um *framework* para o desenvolvimento de aplicações web baseado em *Node.js*. que tem por objetivo facilitar o desenvolvimento de sites resolvendo boa parte dos problemas encontrados na criação deles. Neste projeto, o *Next.js* foi utilizado para o desenvolvimento da aplicação *web* em sua segunda versão [10].

Uma biblioteca muito popular de *JavaScript*, utilizada para facilitar a inserção de lógica em sites, é o *React* [11]. A biblioteca se tornou especialmente útil para o projeto com os seus *hooks*. Um *hook* do *React* é uma subrotina que pode ser associada a um componente da página, por exemplo, um botão ou uma imagem, permitindo a criação de comportamentos especiais e uso de variáveis [12].

Os *hooks* foram utilizados neste trabalho para controlar variáveis sendo exibidas para o usuário na aplicação *web*, já que podem ser utilizados, por exemplo, para criar gatilhos de disparo de subrotinas e armazenar dados temporariamente sem a necessidade de um banco de dados, tarefas que são complexas com *JavaScript* ou *NodeJS* puro.

Uma extensão de navegador é um *software* que pode ser adicionado a um navegador de internet para modificar e expandir suas funcionalidades. Neste trabalho, foi desenvolvida uma extensão de navegador utilizando a linguagem *JavaScript* [8].

Firebase é uma plataforma do *Google* para desenvolvimento de aplicativos que

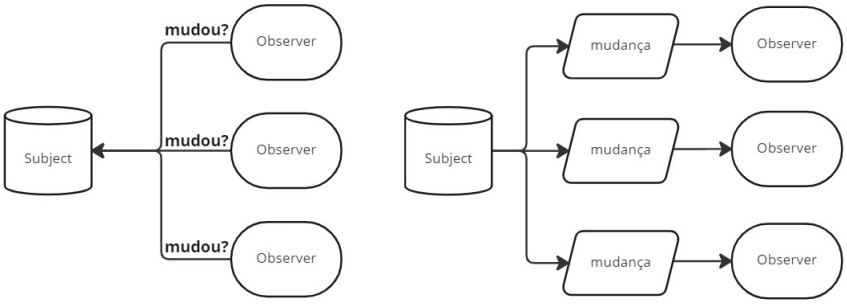

Figura 3 – Padrão observer.

fornece produtos em nuvem, funcionalidades como bancos de dados, hospedagem de sites, serviços de autenticação, inteligência artificial e outros. O produto do *Firebase* utilizado neste projeto foi o banco de dados em nuvem *Realtime Databse*. Este é um banco de dados orientado a documentos que armazena os dados como arquivos [13]. Em uma versão do projeto, foi utilizado o *MongoDB* [14], um produto muito similar ao *Firebase*, mas que parou de ser utilizado por estar menos alinhado com os requisitos do projeto.

O *Firebase* foi escolhido como plataforma para este projeto porque permite a adição futura de outras funcionalidades do *Google* com facilidade e também porque já é prática da ONG utilizar ferramentas do *Google*, o que facilita a aceitação do projeto e compatibilidade com sistemas já existentes.

Um outro motivo para a escolha do *Firebase* para armazenamento dos dados é a sua capacidade de alertar aplicações sobre mudanças nos dados armazenados em tempo real. Isso acontece porque é utilizado em seu *software* o padrão de projeto observador [15]. O padrão observador pode ser aplicado quando um conjunto de *softwares*, chamados de *observers*, precisam se atualizar constantemente sobre as informações de interesse de um outro *software*, o *subject* e, como mostrado na Figura 3, à esquerda, esses *softwares* enviam requisições a todo instante em busca de informações sobre a mudança.

Quando aplicado o padrão *observer*, o *subject*, que contém a informação de interesse, se torna o responsável por informar aos *observers* interessados sobre mudanças. Para o caso deste projeto, o *Firebase* é um *subject* e a aplicação *web* desenvolvida é um *observer*, configurado para receber informações.

Para fazer a comunicação entre todos os *softwares* de diferentes tecnologias e propósitos de um projeto como o desenvolvido aqui, são utilizadas Interfaces de Programação de Aplicações, ou do inglês, *Application Programming Interfaces* (*APIs*) [16]. Uma *API* é um *software* que tem a função de mediar a interação entre outros *softwares*, garantindo a compatibilidade entre dados trocados e mantendo a modularidade.

3 Metodologia, Resultados e Discussões

Neste Capítulo, são apresentados em 3.1 os objetivos para o projeto e, posteriormente nos Capítulos 3.3, 3.4, 3.5, 3.6 e 3.7, é feita uma explicação detalhada sobre o funcionamento do sistema, de suas partes e de como são atendidos os objetivos traçados. Para facilitar a compreensão do sistema, estão disponíveis para consulta seu código-fonte [1] e um vídeo de demonstração [2].

3.1 Objetivos

A seguir, são listados os objetivos gerais que guiaram o projeto e sobre cada um deles, são listados requisitos específicos. É explicado como tais objetivos específicos surgiram ao longo do projeto e porque foi necessário traçar eles para que os objetivos gerais fossem atingidos. Além disso, são feitas explicações de como esses objetivos guiaram algumas decisões. Esses objetivos foram ajustados durante o projeto mantendo contato constantes com os arquitetos de conexões, usuários finais da solução.

3.1.1 Rapidez na geração de *leads*

Como o principal gargalo identificado no processo de captação estava na taxa de geração de *leads*, um dos principais objetivos do projeto foi desenvolver para os arquitetos de conexões uma ferramenta que os desse mais agilidade nessa geração. A solução encontrada, além de economizar tempo eliminando o procedimento de cópia e inserção de dados em uma planilha, também economiza tempo em duas outras tarefas que foram identificadas como gargalos: a verificação da existência de *leads* repetidos e a obtenção do *link* para o perfil do *LinkedIn*.

Foi identificado que a atividade mais demorada para os arquitetos de conexões em seu processo manual era a verificação de que se perfil sendo acessando já foi registrado anteriormente, ou se de fato é um *lead* novo. A ONG possui uma planilha no *Google Sheets* com mais de 10000 *leads* já encontrados no *LinkedIn* e a cada possível novo *lead* encontrado, o arquiteto de conexões fazia uma busca pelo seu nome nessa planilha. Além de ser um processo lento, é um risco jurídico porque a existência dessa planilha violava a LGPD. Esses fatos fizeram com que um dos objetivos do projeto fosse a inclusão de um sistema de detecção de *leads* repetidos.

[1] <https://github.com/JonasMarma/TG-info>
[2] <https://drive.google.com/file/d/1kwzdNt9liT3cx_LWZotDTKZDyF8rmfZk/view?usp=share_link>

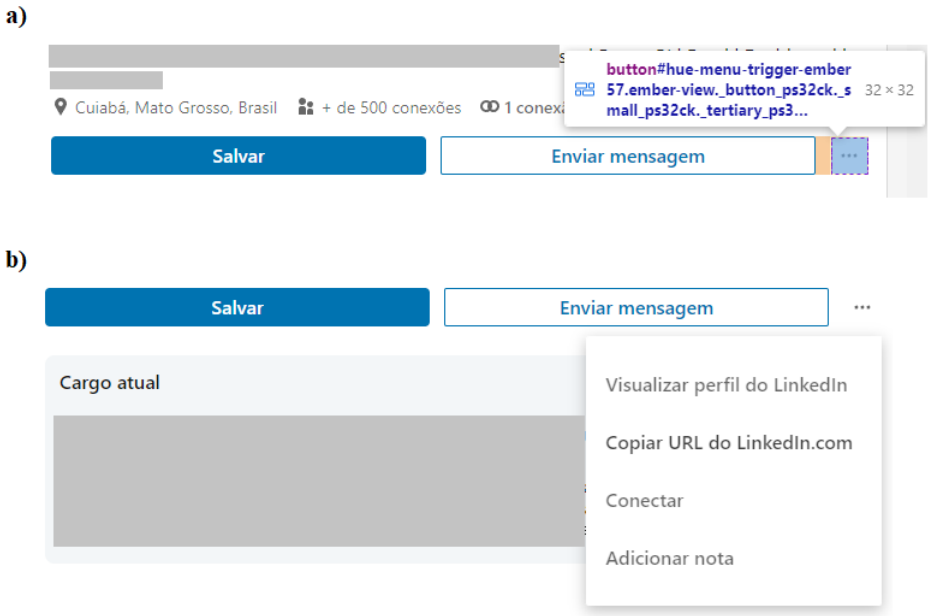

Figura 4 – Obtenção do *link* do perfil pelo *Sales Navigator*. Para obter o *link*, é necessário clicar no botão com reticências (a) e em seguida clicar no botão "Copiar URL do *Linkedin*.com"(b) para que ele seja copiado para a área de transferência. Analisando com o inspetor de elementos, como em (a), o botão e elementos próximos, não é possível encontrar o *link* do perfil.

Outro ponto de atraso era a obtenção do *link* para o perfil do *LinkedIn* quando utilizado o *Sales Navigator*. Para extrair esse *link*, o arquiteto de conexões precisava clicar em dois botões para que o *link* fosse copiado para sua área de transferência, como mostrado na Figura 4, para em seguida colar este *link* na planilha de *leads*. Dessa forma, surgiu também como objetivo no projeto da criação de um sistema que facilitasse esse procedimento.

3.1.2 Interface gráfica

No início, uma interface gráfica não era uma preocupação, mas logo se tornou uma das partes mais complexas do projeto. Sua necessidade foi percebida por meio dos contatos constantes feitos com os arquitetos de conexões, que ao avaliar as primeiras versões, relataram que precisavam de uma forma de visualizar quais dados estão sendo extraídos, escolher se um perfil sendo visitado deve ou não enviado para a planilha e ter indicadores visuais do que está acontecendo no *software* num geral.

Uma interface gráfica não foi uma preocupação inicial porque se pensava na criação de uma tecnologia que pudesse ser usada principalmente por um *bot*. O *bot* faria a geração de uma grande quantidade de *leads* de baixo e médio potencial somente enquanto os arquitetos de conexões seriam responsáveis somente pela geração de uma pequena quantidade de *leads* de alto potencial.

Tal *bot* seria responsável por navegar por uma grande quantidade de perfis, seja pelo *LinkedIn* ou pelo *Sales Navigator*. Para a construção desse *bot*, seria utilizada o *Selenium*, uma biblioteca de *Python* para a criação de testes automatizados [17]. Está disponibilizado um vídeo de demonstração de uma versão descontinuada desse *bot* [3].

A ideia para a utilização de um *bot* que fosse capaz de gerar *leads* em grande quantidade, ao menos para o momento atual da ONG, foi arquivada porque ainda não há uma estrutura grande o suficiente para interagir com todos esses *leads* e porque chegou-se à conclusão de que a melhor forma de atração desses *leads* de menor potencial seria de forma mais passiva, utilizando anúncios.

3.1.3 Segurança de dados

Como mencionado em 2.2, uma vez que o *software* desenvolvido lida com dados pessoais de potenciais doadores da ONG, é necessário ter em mente as leis de proteção de dados aplicáveis durante o desenvolvimento do projeto. Foi identificado que o principal risco jurídico dentro do escopo do projeto era a planilha no *Google Sheets* com dados de mais de 10000 *leads*, citada em 3.1.1.

Com a implementação deste projeto, a planilha se tornou desnecessária porque os dados de *leads* já gerados, como discutido em maiores detalhes a seguir, passaram a ser armazenados em um banco de dados no *Firebase Realtime Database*. Assim, foi eliminada a necessidade de manipulação manual desses dados de forma que apenas o *software* desenvolvido tem acesso aos mesmos. Além disso, existem dispositivos de segurança próprios do *Firebase*, o que torna o armazenamento desses dados mais seguros [18].

Ainda assim, a situação da segurança de dados da organização não é satisfatória porque os processos seguintes à geração de *leads* são feitos utilizando planilhas com poucas políticas de segurança. Além disso, dentro do escopo do projeto, a segurança do armazenamento de dados poderia ser melhorada utilizando técnicas de anonimização. Essas melhorias futuras são descritas com mais detalhes no Capítulo 4.

3.1.4 Compatibilidade e flexibilidade

A compatibilidade do *software* desenvolvido foi pensada considerando as outras tecnologias já utilizadas pela organização. A preocupação foi necessária para garantir que o projeto funcionasse sem atrito com essas tecnologias, para que outros membros do ONG pudessem fazer a manutenção do projeto uma vez pronto e para facilitar o trabalho no caso de alterações futuras.

A maior parte dos processos da ONG são feitos utilizando o *Google Workspace*, especialmente o *Google Sheets* para a organização de dados e processos administrativos em

[3] https://www.youtube.com/watch?v=tj85pBWBQqM

planilhas. Esse fato levou à preferência da utilização de ferramentas compatíveis. No caso, foi utilizado o *Firebase Realtime Database* para a construção dos bancos de dados quando necessários, e do *Google Apps Script* para a construção de uma *API* para fazer o envio dos dados do *lead* para uma planilha do *Google Sheets* utilizada pelos arquitetos de conexões.

Há também na organização um departamento de tecnologia, que está criando uma plataforma virtual para acesso de bolsistas, doadores e mentores. As tecnologias utilizadas pelos membros desse departamento com funções relevantes a este projeto são a biblioteca *React*, para a construção de interfaces de usuário, e o ambiente *NodeJS* para a construção de *backends* de *sites*. Essas duas tecnologias também foram utilizadas no projeto no *site* que faz o *scrapping* e contém a interface para o arquiteto de conexões. A utilização dessas tecnologias é conveniente porque já existem na organização membros especializados nela que poderão ser acionados caso necessário.

Por fim, tendo em vista a migração dos dados e processos da organização para o *CRM* da *Salesforce*, o projeto foi desenvolvido tendo em mente a modularidade, ou seja, o funcionamento independente de diferentes partes do sistema com pouca dependência técnica de outras partes. Isto é importante porque facilita a mudança de tecnologias e deixa o sistema mais flexível à mudanças e inserção de novas funcionalidades.

3.2 Utilização do ponto de vista do arquiteto de conexões

Na interface gráfica desenvolvida existem controles que podem ser utilizados para ativar ou desativar o preenchimento e envio automático. Isso permite a utilização do *software* de uma forma mais rápida e automática nos casos em que é necessário gerar uma grande quantidade de *leads* em pouco tempo, ou de uma forma mais lenta e manual para momentos em que é necessário gerar uma quantidade menor *leads*, mas com mais atenção aos seus dados.

Na Figura 5 são mostrados os fluxos possíveis ao arquiteto de conexões ao utilizar a ferramenta de geração de *leads* após sua instalação da extensão de navegador. Para iniciar a geração, deve-se manter duas janelas abertas: uma com o *Sales Navigator*, onde são visitados perfis de possíveis *leads*, e outra com o *site* que contém a interface gráfica. A partir da interface gráfica é possível configurar se será utilizada ou não a aquisição automática de dados do *LinkedIn* e o envio automático de dados do *lead* para a planilha de *leads*.

Após feita a configuração, inicia-se na janela de navegador a busca pelo perfil de um *lead* utilizando o *Sales Navigator*. Quando este perfil é encontrado e aberto, caso a aquisição automática de dados esteja ativada, os dados esse perfil do *LinkedIn* aparecerão automáticamente na interface gráfica e o *link* para a página de perfil aparecerá quando o arquiteto de conexões copiá-lo clicando nos botões indicados na Figura 4. Caso a aquisição

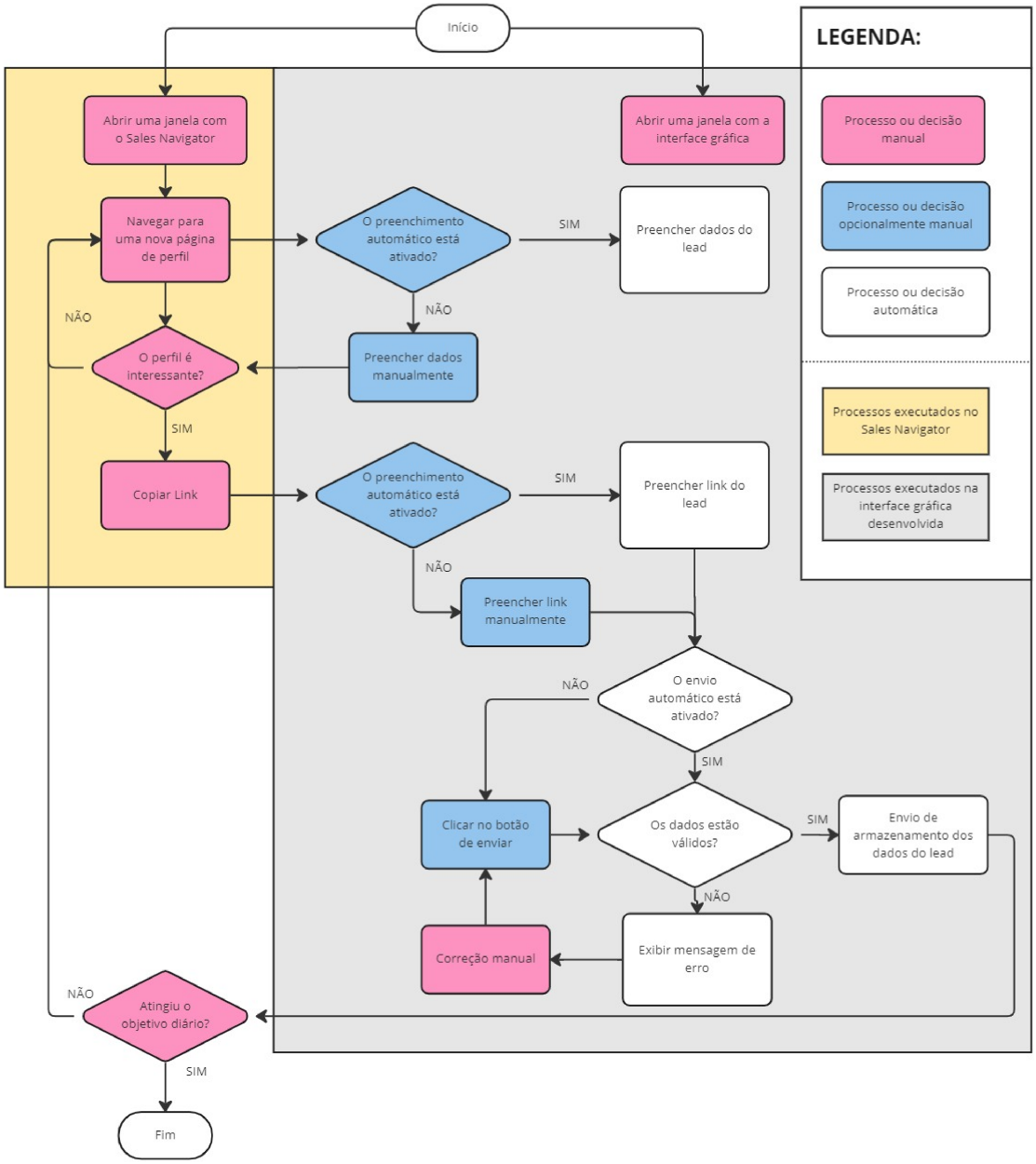

Figura 5 – Fluxo recomendado ao arquiteto de conexões para gerar *leads*. Destacam-se as tarefas e decisões obrigatoriamente manuais, que devem ser feitos pelo arquiteto de conexões a cada vez que desejar gerar *leads*, as tarefas e decisões opcionalmente manuais, que o arquiteto pode optar por deixar que o *software* realize ou decidir realizar manualmente caso deseje ter mais controle, e as tarefas e decisões automáticas, que são realizadas sem intervenção humana.

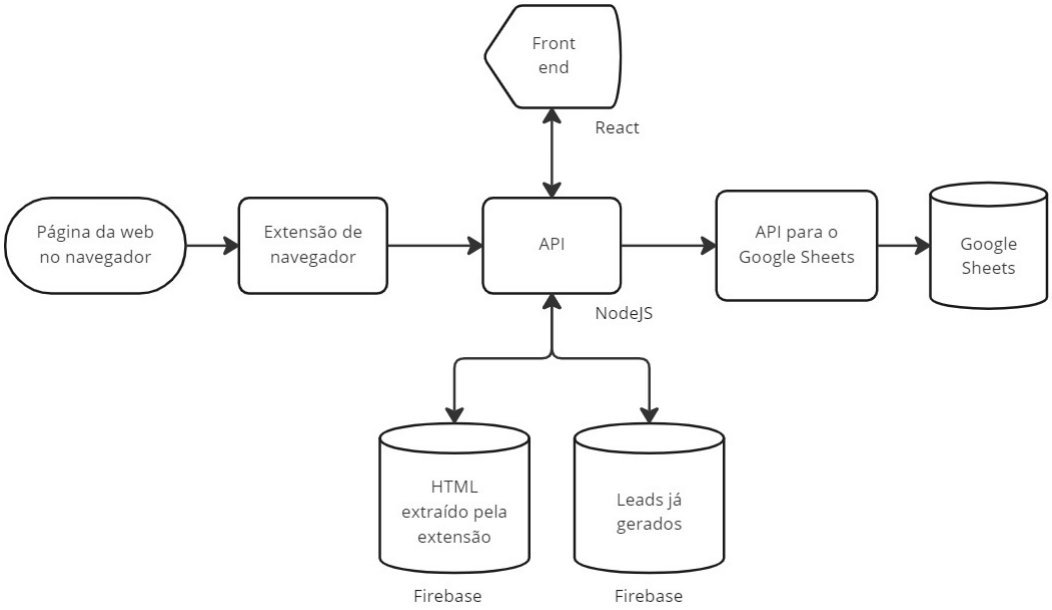

Figura 6 – Diagrama de blocos do funcionamento do gerador de *leads*.

automática esteja desativada, o arquiteto deve preencher esses dados manualmente, ou ainda, caso haja um erro na extração automática desses dados, o arquiteto pode desativá-la e corrigir esses dados.

Caso o envio automático de dados do *lead* esteja ativado, no momento em que o *link* para o perfil foi copiado e preenchido na interface gráfica os dados do perfil são validados, inclusive quanto à duplicidade, e enviados automaticamente para a planilha de *leads*. Nesse momento, o arquiteto de conexões está pronto para gerar um novo *lead* acessando um novo perfil e repetindo o processo. Caso o envio automático esteja desativado, deve-se clicar no botão de envio para que as validações sejam feitas e os dados sejam enviadas.

Nos Capítulos 3.3 a seguir, é feita uma explicação, de forma resumida, do funcionamento do sistema desenvolvido. Em seguida, nos Capítulos 3.4, 3.5, 3.6 e 3.7, são feitas explicações aprofundadas sobre o funcionamento de cada um dos componentes desse sistema, a saber: a extensão de navegador, *APIs*, banco de dados e a interface gráfica.

3.3 Resumo do funcionamento

Na Figura 6 é mostrado um diagrama de blocos do funcionamento do sistema desenvolvido. O processo se inicia com o acesso do arquiteto de conexões à página de um *lead* em potencial e a extração automática do código *HTML* dessa página pela extensão de navegador instalada no computador do arquiteto de conexões, como mostrado nos dois blocos à esquerda.

Em seguida, a extensão de navegador envia o código *HTML* extraído para uma *API*, hospedada em um servido externo ao computador do arquiteto de conexões. A *API*

é responsável por receber dados da extensão de navegador e enviar os dados do *lead*, já processados, para uma planilha do *Google Sheets* através de uma segunda *API* para o *Google Sheets*, no ramo direito do diagrama. A extração dos dados do *lead* do código *HTML* enviado pela extensão é feita pela interação entre os bancos de dados e a interface do usuário (*front end*), ao centro do diagrama, mediados pela *API*.

3.4 Extensão de navegador

A extensão de navegador foi a forma encontrada para extrair dados do *LinkedIn* com o baixo risco de que o mesmo detectasse uma atividade suspeita. A ideia é aproveitar o fato de que quando uma página da *web* é acessada, é feito o *download* do conteúdo a ser exibido no computador e esses arquivos são abertos e interpretados por um navegador. A partir do momento em que o *download* do conteúdo foi feito, esse conteúdo pode ser manipulado a necessidade de se conectar novamente com o servidor, no caso do *LinkedIn*, que o enviou originalmente. Chegou-se à conclusão de que uma extensão de navegador era adequada para esse reenvio porque permite fácil acesso e manipulação desses arquivos pelo usuário através da linguagem *JavaScript*.

Na Figura 7, é mostrado o diagrama de blocos do funcionamento da extensão. É um ciclo que se repete a cada segundo em que primeiro se verifica se o usuário está acessando uma página do *LinkedIn*. Em caso afirmativo, verifica-se se há algo em sua área de transferência ou se houve alguma alteração no código-fonte da página (essa alteração sempre ocorre quando o usuário sai de uma página da *internet* qualquer para uma página do *LinkedIn*, iniciando o ciclo). Em caso afirmativo, verifica-se se está sendo visitada uma página de perfil do *Sales Navigator* e, nesses casos, é enviado o código-fonte para a *API* através de uma requisição *HTTP* com o objetivo de atualizar o banco de dados no *Firebase Realtime Database*.

Em versões iniciais, a extensão realizava também a extração dos dados do *lead*, enviando-os diretamente para a *API* do *Google Sheets*. A delegação dessa função da extensão para outras partes do sistema foi benéfica porque a extensão é um *software* que deve ser instalado no computador do arquiteto de conexões, sendo assim, é mais conveniente mantê-la o mais simples o possível para evitar que os usuários tenham que instalar novas versões a cada vez que que a extensão precisa ser atualizada, o que também deve ocorrer raramente porque a complexidade do sistema não se concentra na extensão.

Outra vantagem em se remover complexidade da extensão está em suas limitações técnicas. Não há suporte, por exemplo, para executar funções do *NodeJS* em uma extensão de navegador, o que torna difícil a utilização de diversas bibliotecas úteis como o *Cheerio*, utilizado para *web scrapping*. Por esse motivo, na primeira versão do projeto, o algoritmo de *scrapping* foi implementado puramente em *JavaScript* no código-fonte da extensão, o

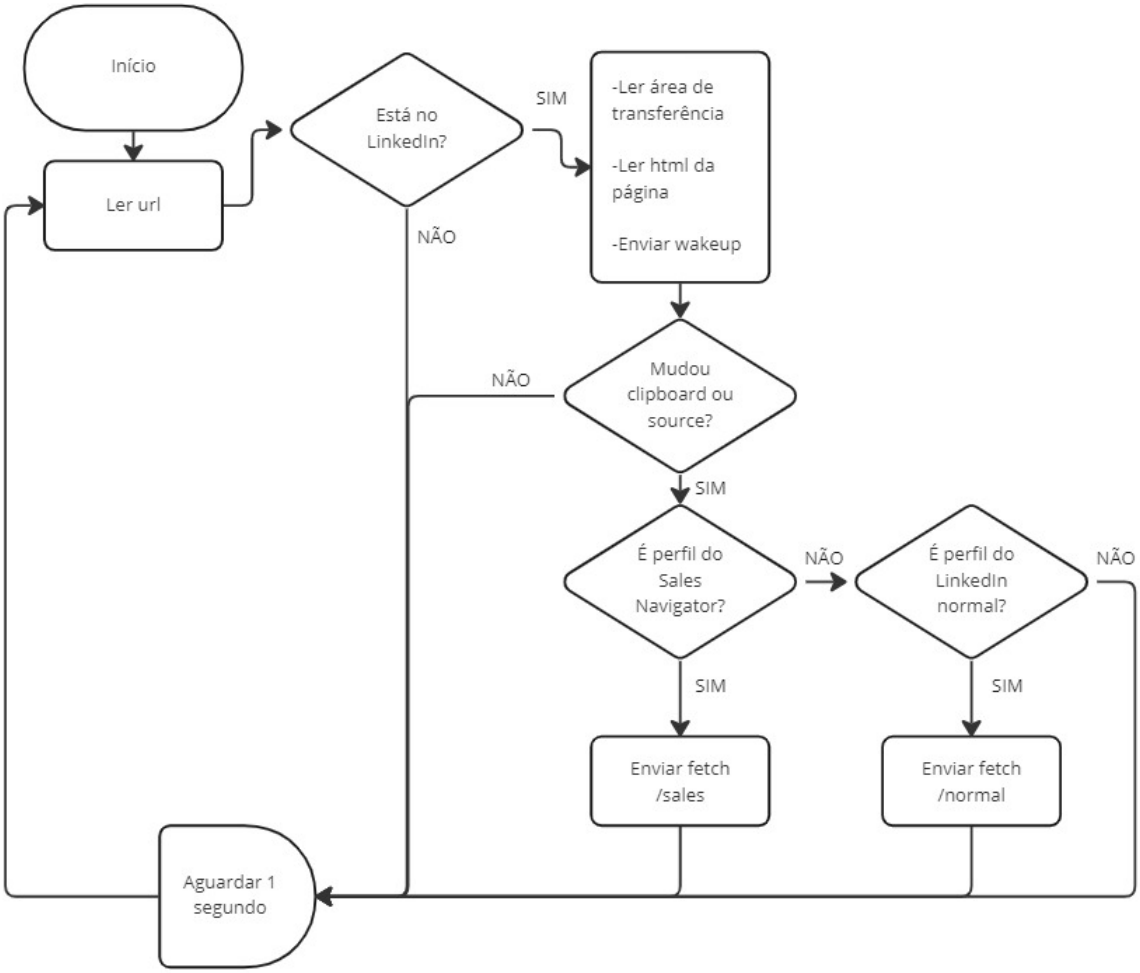

Figura 7 – Funcionamento da extensão

que dificultava sua manutenção. Este é um dos problemas que foram resolvidos com a delegação das funções da extensão.

3.5 API e banco de dados

A extensão de navegador extrai e envia o código-fonte *HTML*, que contém as informações do *lead* misturadas ao conteúdo do *site*, para uma *API*, hospedada em um servidor externo. Essa *API* é responsável por receber o código e armazená-lo em um banco de dados para consulta posterior. Esse banco de dados é utilizado porque, a depender a velocidade de conexão, conforme o a página de perfil é carregada, a extensão no computador do usuário pode ver várias versões do arquivo *HTML*. O banco de dados mantém sempre a versão mais recente pronta para ser acessada pela *API*.

Essa segunda consulta é feita quando o arquiteto de conexões acessa a interface gráfica da ferramenta de geração de *leads*, ou seja, o *front-end* da aplicação *web* descrita a seguir. No momento do acesso a *API* é acionada novamente para fazer uma consulta no banco de dados, recuperar o código-fonte em *HTML* da página de perfil e enviá-lo para a

interface gráfica.

Para a criação do banco de dados, foi escolhido o *Firebase Realtime Database* porque uma de suas principais funcionalidades é a obtenção de dados *realtime* com a utilização do padrão de projetos observador. Isso foi aproveitado no projeto para permitir a obtenção de dados de *lead* mais recentes sem a necessidade de recarregamento manual pelo usuário.

A aplicação *web* se inscreve como observador do nó correspondente ao usuário conectado. Quando o nó é atualizado, ou para esta aplicação específica, quando é recebido um novo código-fonte de uma página de perfil do *lead*, esse novo dado, e somente ele, é enviado do *Firebase Realtime Database* para a aplicação *web* que, por sua vez, aciona automaticamente uma função para o tratamento desse dado, que para esta aplicação específica, é a atualização dos dados do *lead* no formulário caso a opção de atualização automática esteja ligada.

3.6 Interface gráfica

A interface gráfica mostrada na Figura 8 é, na verdade, um *site* acessado pelo arquiteto de conexões. O *site* foi pensado para ser utilizado em uma janela do navegador que fica ao lado da janela onde está aberto o *Sales Navigator*. Na Figura 8 são mostrados quatro estados possíveis, correspondentes às combinações dos modos ativado e desativado do preenchimento e envio manual. A interface consiste de um formulário dos dados relevantes do *lead* e alguns botões.

Destaca-se a existência de dois botões: um para controlar o preenchimento automático ou manual do formulário e outro para controlar o envio automático ou manual dos dados do *lead*. Ao clicar nos botões, essas opções são ativadas ou desativadas, levando a interface a quatro estados possíveis: preenchimento e envio automático, preenchimento manual e envio automático, preenchimento automático e envio manual e preenchimento e envio manual.

A ativação ou não desses botões depende dos objetivos de captação semanais: pode ser mais conveniente gerar os *leads* de forma mais lenta, fazendo uma leitura criteriosa do perfil antes de enviá-lo para o funil, ou pode ser mais conveniente registrar dezenas de perfis em pouco tempo se baseado apenas na empresa que trabalham e no cargo que ocupam.

Caso o envio automático do *lead* esteja ativado, o botão de envio do *lead* desaparece e os dados do *lead* serão enviados para a planilha do *Google Sheets* sempre que o *link* de perfil for trocado por outro *link* válido. Caso o envio automático esteja desativado, o botão de envio aparece novamente e o arquiteto de conexões deverá clicar nesse botão para fazer

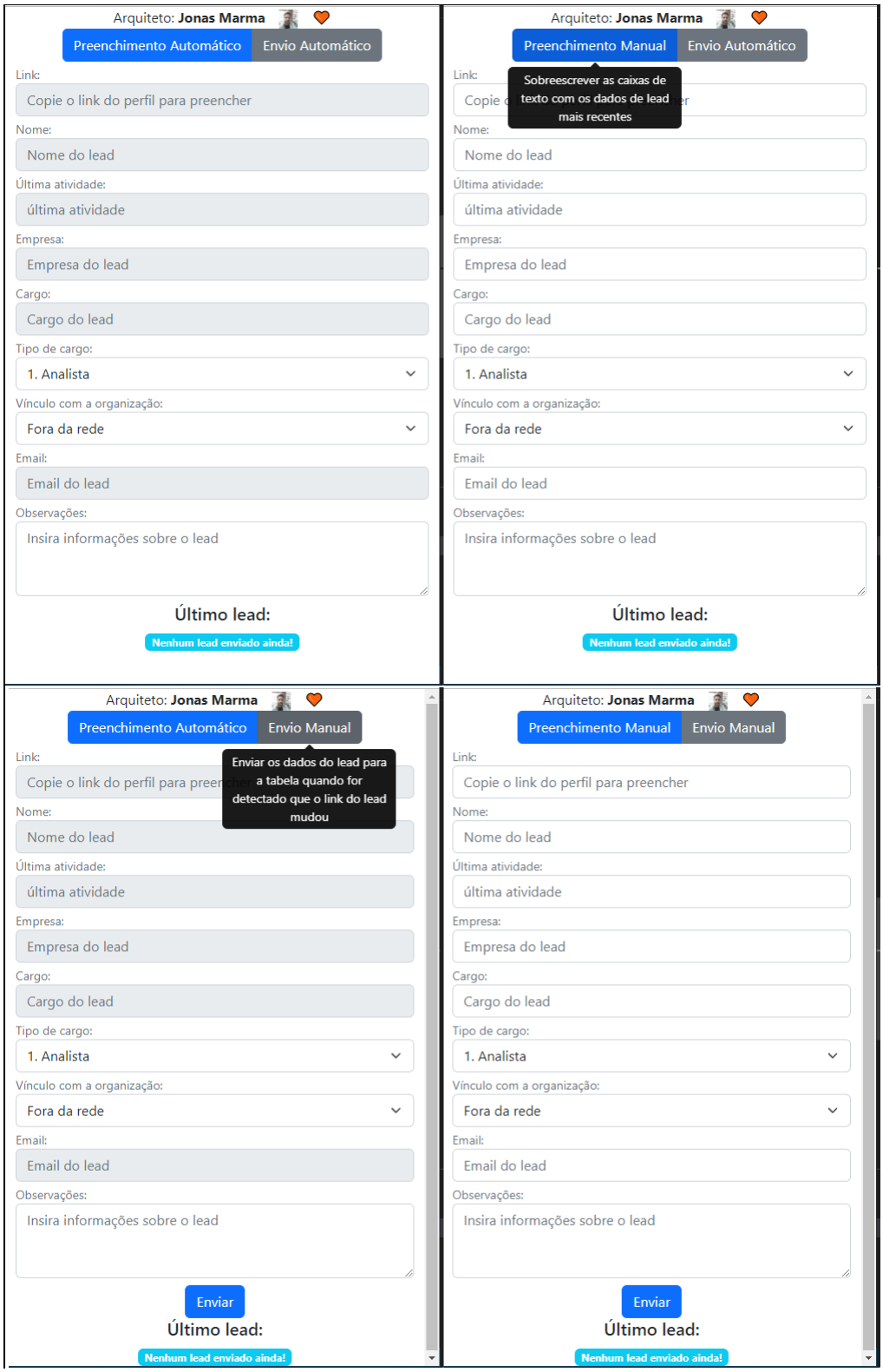

Figura 8 – Interface gráfica. Observa-se que há quatro modos de geração de *lead*: preenchimento e envio automático, preenchimento manual e envio automático, preenchimento automático e envio manual e preenchimento e envio manual.

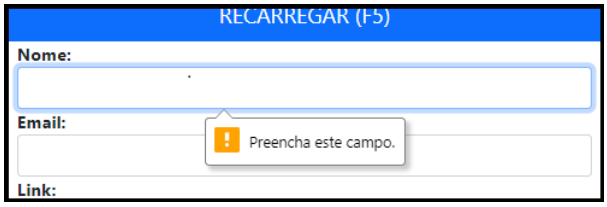

Figura 9 – Exemplo de aviso exibido caso seja detectado um erro no preenchimento dos dados do *lead*.

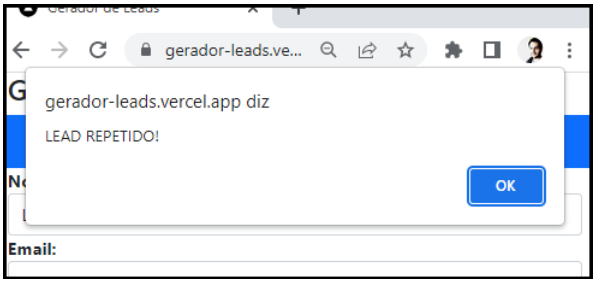

Figura 10 – Aviso exibido caso o *lead* já tenha sido gerado.

o envio do *lead*.

Caso o preenchimento automático de dados do *lead* esteja ativado, os campos do formulário se fecham para edições e passam a ser atualizados sempre com os últimos dados do último perfil visitado pelo *Sales Navigator*. Caso essa opção esteja desativada, o arquiteto de conexões deve inserir esses dados de forma manual. Isso pode ser usado para corrigir eventuais erros da extração de dados.

A fim de evitar erros de preenchimento, o formulário verifica se foram preenchidos ao menos os dados de nome, *link*, cargo, tipo de cargo e vínculo com a instituição, já que tais dados são indispensáveis para a continuação do processo de arrecadação e estão sempre disponíveis no perfil. Caso dados inválidos sejam encontrados em alguma dessas verificações, é mostrado um aviso como na Figura 9 e os dados do *lead* não são enviados até que o arquiteto de conexões corrija o erro. Além disso, quando o *link* para o perfil é preenchido, é feita uma verificação no banco de dados que contém os *links* dos perfis dos mais de 10000 *leads* já gerados anteriormente. Caso seja um perfil repetido, é mostrado um aviso como na Figura 10 e o *lead* não é gerado.

Antes da apresentação da interface gráfica, há um sistema de login baseado no *Google Identity Platform*, como mostrado na tela de *login* da Figura 11. Essa tela exige que seja feito um *login* com uma conta da organização antes que se possa utilizar o *site*. Sua utilização, além de tornar o sistema mais seguro, permite a utilização do sistemas por múltiplos usuários simultaneamente: os códigos-fonte *HTML* são armazenados no banco de dados com uma identificação do arquiteto que utilizou a extensão para gerá-los e quando esse arquiteto acessa a aplicação *web*, somente os dados correspondente a ele são acessados. Após o *login*, o nome e foto de perfil da conta *Google* do usuário aparecem no

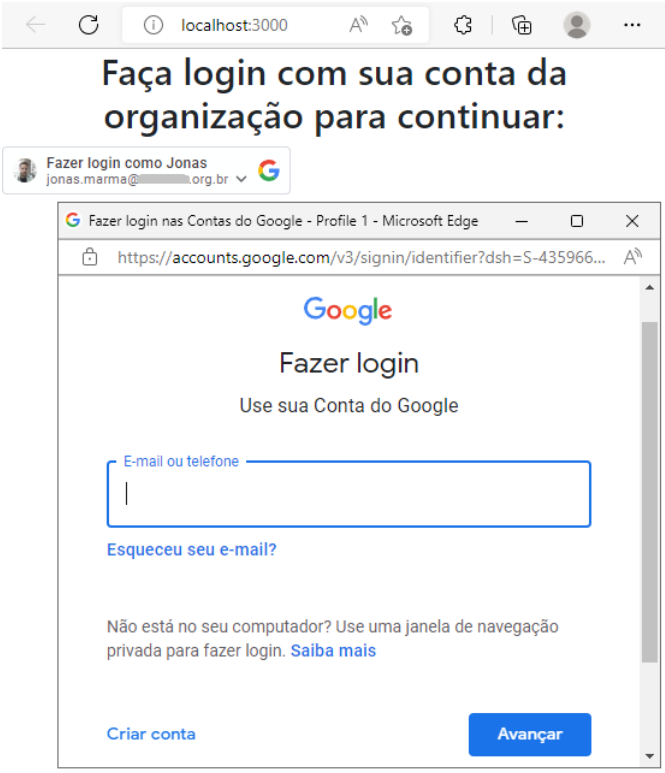

Figura 11 – Sistema de login utilizando o *Google Identity Platform*.

canto superior do gerador de *leads*, como mostrado na Figura 8.

Na Figura 12 é mostrado o fluxograma da lógica de renderização e atualização da página utilizando o *React*. Primeiramente, verifica-se se o usuário fez o *login* em uma conta da organização utilizando a biblioteca do *Google Identity Platform*. Enquanto o *login* não é feito, é renderizado apenas o botão de *login*. Após feito o *login*, os botões de preenchimento e envio automático ou manual, bem como o formulário, são renderizados e, em um estado inicial, o preenchimento de dados do *lead* e o envio desses dados ficam com o modo automático ligado.

Caso o preenchimento de dados automático esteja ativado, a cada vez que a extensão de navegador envia uma atualização sobre o código-fonte de página de perfil para o banco de dados no *Firebase* e, consequentemente, o serviço do *Firebase* repassa essa atualização para a aplicação *web*, é feito um *scrapping* dessa página utilizando a biblioteca *Cheerio* e os dados do *lead* são alterados.

Após a renderização dos botões e do formulário, o programa entra em um ciclo de monitoramento das variáveis dos estados automáticos ou manuais e das variáveis que armazenam os dados do *lead*. Caso o envio automático seja desativado, o botão de envio é exibido, e caso haja alguma alteração nos dados do *lead*, seja por um *scrapping* de código-fonte *HTML* com o preenchimento automático esteja ativado, ou seja por alteração manual do usuário, o formulário é renderizado novamente com as novas informações do *lead*.

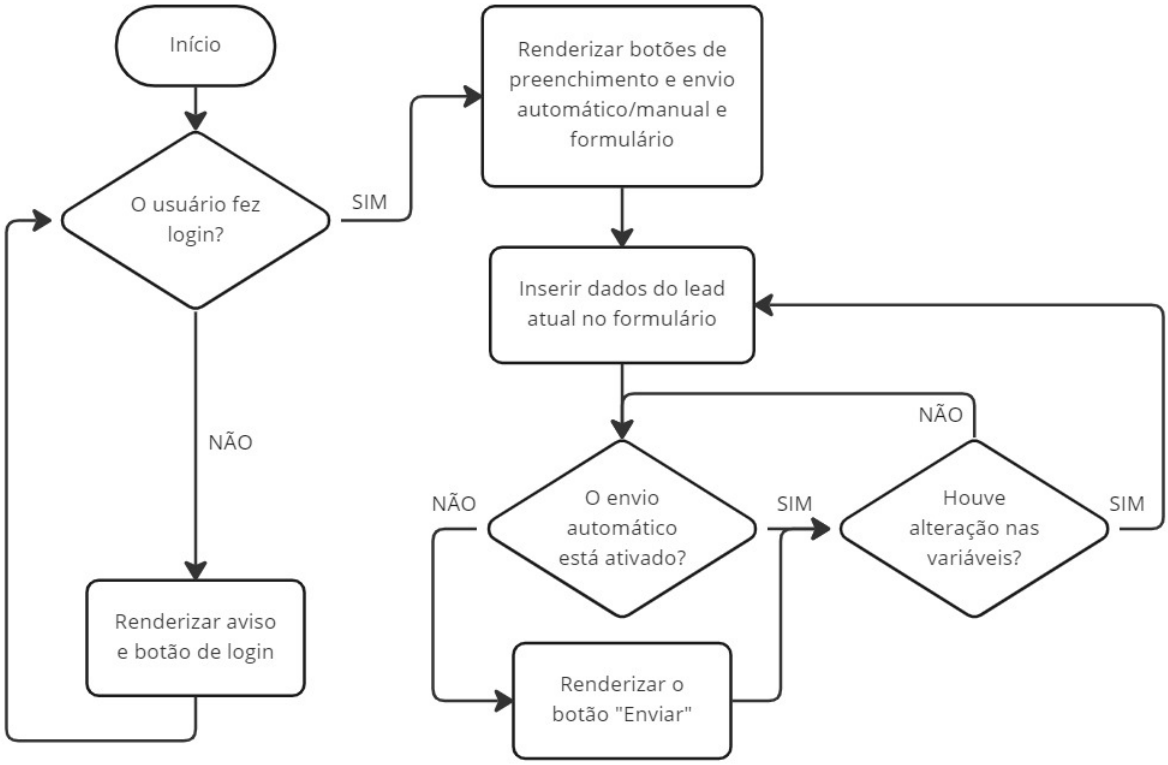

Figura 12 – Esquema de renderização da página do gerador de *leads*.

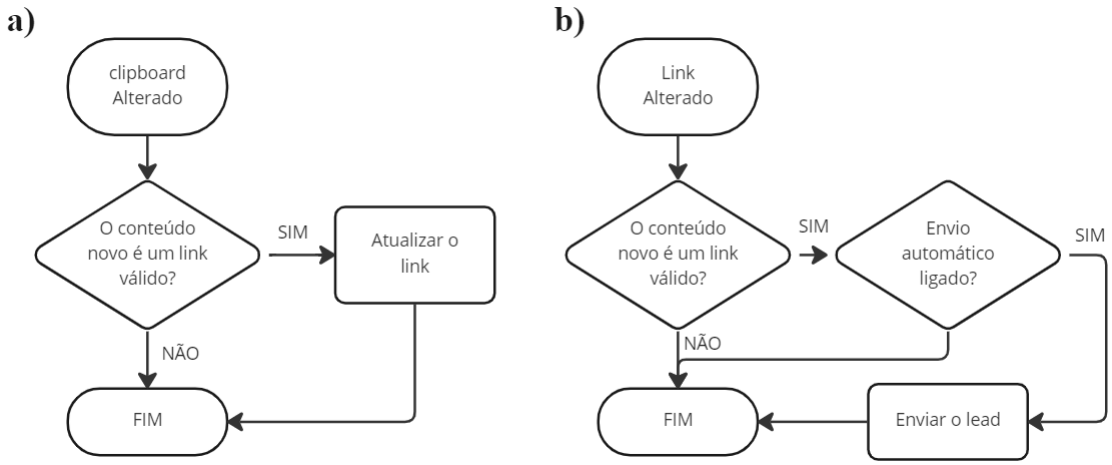

Figura 13 – *Hooks* acionados para a atualização do *link* do *lead* e envio dos dados do mesmo caso o envio automático esteja ativado.

A extração do *link* é feita de forma separada dos outros dados do *lead* porque tal *link* não está presente no código-fonte *HTML* da página de perfil no *Sales Navigator*. Na Figura 13, é mostrado como a atualização de *links* é feita utilizando *hooks* de *React*. O processo começa quando o arquiteto de conexões copia um *link* do *LinkedIn*, como demonstrado na Figura 4. Ao fazer a cópia do *link*, este é passado para a área de transferência de seu computador, ou *clipboard*. A extensão de navegador detecta essa mudança e envia o texto copiado para o banco de dados no *Firebase Realtime Database*.

Nesse momento, seguindo o padrão de observador, é enviado um sinal do banco de

dados para a aplicação *web*, informando sobre a atualização. É então ativado o *hook* **a)** da Figura 13: um algoritmo verifica se o texto na área de transferência do usuário é válido e, em caso afirmativo, atualiza a variável que armazena o *link*.

Essa alteração no *link* causa o disparo do *hook* **b)** da Figura 13. Esse segundo fluxo é responsável por verificar se o *link* é válido e, em caso afirmativo, se o envio automático estiver ativado, enviar os dados do *lead* para a planilha do *Google Sheets*. Após feita a extração das informações relevantes do *lead* e antes do envio, tais informações são procuradas no banco de *leads* já gerados através da *API* a fim de verificar se o *lead* é repetido. Caso o *lead* seja repetido, é mostrado o avisado da Figura 10.

3.7 Transferência dos dados para o Google Sheets

Ao clicar no botão de envio, a *API* é acionada mais uma vez após as validações mencionadas anteriormente e os dados são enviados para a *API* do *Google Sheets*, desenvolvida com o *Google Apps Script*, para a inserção em uma planilha. Ao mesmo tempo, o banco de dados de *leads* já gerados é atualizado com a inserção dos dados do novo *lead*.

A inserção de novos registros é feita em uma planilha do *Google Sheets* porque os demais processos do departamento de *Hunting* PF utilizam ferramentas do *Google*, mas bastaria refazer esta última *API* para enviar os dados para um sistema de CRM mais robusto, como desejado no capítulo 4. A *API* consiste de uma rotina simples que se hospeda em um servidor do *Google* e aguarda o recebimento de requisições *HTTP*. A aplicação *web* é responsável pelas seguintes tarefas:

1. Receber as requisições feitas pelo computador que utiliza a extensão;
2. Decodificar os dados do *lead* recebidos;
3. Inserir em uma planilha do *Google Sheets* um novo registro de *lead*, agora acessível para os executivos de contas.

Após o envio dos dados do *lead*, é encerrada sua geração e se iniciam os outros processos da organização, fora do escopo deste projeto já que a geração de um *lead* pelo arquiteto de conexões é somente o início do funil de vendas.

4 Conclusões e Trabalhos Futuros

Este projeto faz parte do trabalho do departamento de inteligência de mercado da ONG com o objetivo de pesquisar novas tecnologias para automatizar processos da organização. Uma outra linha de pesquisa que impactará o futuro do projeto é a migração do armazenamento e processamento de dados para as soluções de *CRM* da *Salesforce* [19].

A migração é um processo demorado, mas que já iniciou e será benéfico a longo prazo porque trará mais robustez para processos e segurança aos dados, uma das principais preocupações atuais. Quanto à segurança de dados, no caso específico da geração de *leads*, o armazenamento dos dados de *leads* em tabelas do *Google Sheets*, como explicado ao final do Capítulo 3, podia ser interpretado como não estando totalmente de acordo com a LGPD e representava um risco caso haja vazamento desses dados para fora da ONG. Felizmente, o sistema desenvolvido amenizou esse risco e problemas adicionais de segurança de dados serão mitigados com essa migração.

Um dos objetivos que guiou várias decisões do projeto, como explicado no Capítulo 3.1.4, foi a flexibilidade e compatibilidade para permitir alterações e implantação de soluções adicionais. Esta flexibilidade permite, por exemplo, para o caso de uma migração para um *CRM* da *Salesforce*, a substituição da *API* de conexão com o *Google Sheets* por uma *API* de conexão com o *Salesforce* [20]. Tal mudança é possível, conforme mencionado por consultores voluntários da *Salesforce*, e os membros da organização já estão estudando os detalhes técnicos e fazendo treinamentos.

Até o momento, o sistema desenvolvido captura somente os dados essenciais dos *leads* sendo gerados, mas a partir do momento que esses dados passarem a ser coletados com mais robustez e o processo de captação se solidificar, surgirá a oportunidade de coletar dados adicionais dos perfis que poderão ser utilizados em conjunto com dados de doações e inteligência artificial ajudar no entendimento de padrões de doações e guiar as decisões de segmentação de perfis de *lead* e metas semanais do departamento de *Hunting* PF.

Por fim, a manutenção do sistema proposto neste trabalho por parte dos membros organização requer o estudo de tecnologias *web*. Apesar de que essa manutenção pareça ser mais trabalhosa do que a simples geração manual de *leads*, ela é útil para a ONG porque se alinha com o objetivo de inclusão no mercado de trabalho: é um objetivo válido tanto para os bolsistas como para os membros administrativos e acredita-se que o estudo de tecnologias *web* é benéfico nesse sentido para os envolvidos na manutenção.

5 Referências

[1] About LinkedIn. 2022. <https://about.linkedin.com/>. [Acessado em 3 de Dezembro de 2022]. Citado na página 3.

[2] KOTLER, P. Marketing essentials. [S.l.]: Prentice-Hall, 1984. ISBN 0135572320. Citado na página 3.

[3] BRASIL - LEI Nº 13.709, DE 14 DE AGOSTO DE 2018 - Lei Geral de Proteção de Dados Pessoais (LGPD). 2022 <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm>. [Acessado em 01 de Agosto de 2022]. Citado na página 4.

[4] LEI Nº 12.965, DE 23 DE ABRIL DE 2014 - Estabelece princípios, garantias, direitos e deveres para o uso da Internet no Brasil. 2022. <http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2014/lei/l12965.htm>. [Acessado em 01 de Agosto de 2022]. Citado na página 4.

[5] KROTOV VLAD. JOHNSON, L. S. L. Legality and ethics of web scraping. Communications of the Association for Information Systems, ResearchGate, v. 47, p.441–447, 2020. Disponível em: <https://doi.org/10.17705/1CAIS.04724>. Citado na página 4.

[6] LINKEDIN - Contrato do Usuário. 2022. <br.linkedin.com/legal/user-agreement?trk=content_footer-user-agreement>. [Acessado em 01 de Novembro de 2021]. Citado na página 5.

[7] LINKEDIN - Soluções de Vendas. 2022. <business.linkedin.com/pt-br/sales-solutions>. [Acessado em 01 de Novembro de 2021]. Citado na página 6.

[8] MDN - Webdocs. 2022.<https://developer.mozilla.org/pt-BR/>. [Acessado em 01 de Novembro de 2021]. Citado 2 vezes nas páginas 6 e 7.

[9] NODE.JS - JavaScript runtime built on Chrome's V8 JavaScript engine. 2022. <https://nodejs.org/en/>. [Acessado em 01 de Novembro de 2021]. Citado na página 6.

[10] NEXTJS - The React Framework for Production. 2022. <https://nextjs.org/>. [Acessado em 13 de Agosto de 2022]. Citado na página 7.

[11] DOCUMENTAÇÃO - React. 2022. <https://pt-br.reactjs.org/>. [Acessado em 25 de Outubro de 2022]. Citado na página 7.

[12] DOCUMENTAÇÃO - Hooks do React. 2022. <https://pt-br.reactjs.org/docs/hooks-intro.html>. [Acessado em 25 de Outubro de 2022]. Citado na página 7.

[13] FIREBASE Realtime Database - Store and sync data with our NoSQL cloud database. 2022. <https://firebase.google.com/docs/database>. [Acessado em 01 de Novembro de 2021]. Citado na página 7.

[14] MONGODB Documentation. 2022. <https://www.mongodb.com/docs/>. [Acessado em 23 de Outubro de 2022]. Citado na página 7.

[15] GAMMA, E. et al. Design Patterns: Soluções Reutilizáveis de Software Orientados a Objetos. 1ed. ed. [S.l.]: Bookman, 1994. ISBN 978-8573076103. Citado na página 7.

[16] MDN Web Docs - API. 2022. <https://developer.mozilla.org/en-US/docs/Web/API>. [Acessado em 25 de Outubro de 2022]. Citado na página 8.

[17] MDN Web Docs - Base64. 2022. <https://developer.mozilla.org/en-US/docs/Glossary/Base64>. [Acessado em 05 de Agosto de 2022]. Citado na página 10.

[18] SELENIUM Documentation. 2022. <https://www.selenium.dev/documentation/>. [Acessado em 10 de Agosto de 2022]. Citado na página 11.

[19] SALESFORCE - CRM: O que é? 2022. <https://www.salesforce.com/br/crm/>. [Acessado em 13 de Agosto de 2022]. Citado na página 23.

[20] DOCUMENTAÇÃO - API do Salesforce. 2022. <https://developer.salesforce.com/docs/atlas.en-us.api_rest.meta/api_rest/intro_rest.htm>. [Acessado em 25 de Outubro de 2022]. Citado na página 23.