UNIVERSIDADE FEDERAL DO ABC
CENTRO DE ENGENHARIA, MODELAGEM E CIÊNCIAS SOCIAIS APLICADAS

Kauan Carvalho Calasans
Luiz Henrique Speht Reis de Oliveira

# GOVERNANÇA DE DADOS EM BIG DATA NO CENÁRIO DE DESENVOLVIMENTO DE SOFTWARE PÓS LGPD

Santo André
2023

Kauan Carvalho Calasans
Luiz Henrique Speht Reis de Oliveira

# GOVERNANÇA DE DADOS EM BIG DATA NO CENÁRIO DE DESENVOLVIMENTO DE SOFTWARE PÓS LGPD

Trabalho apresentado como requisito obrigatório para a conclusão do curso de Engenharia de Informação da Universidade Federal do ABC

Orientador(a): Prof. Dr. Ricardo Suyama

Santo André
2023

# Dedicatória

Dedicamos este trabalho ao nosso núcleo familiar, que empregaram, e empregam, um papel de apoio fundamental em nossas vidas e que se dedicaram fielmente para nos dar toda educação adequada. Com o mesmo vigor dedicamos este trabalho aos nossos amigos próximos que nos trouxeram leveza na execução durante um período complicado, com trabalho e graduação correndo em paralelo.

# Agradecimentos

Agradecemos ao nosso orientador, Prof. Dr. Ricardo Suyama, que se mostrou extremamente flexível durante a jornada do trabalho de graduação, nos incentivando e se fazendo presente em todos os momentos. Agradecemos também ao Prof. Dr. João Henrique Kleinschmidt e ao Prof. Dr. Roberto Sadao Yokoyama, que compuseram, juntamente com nosso orientador, a banca avaliadora, pelos conselhos e sugestões que melhoraram a estrutura do presente trabalho. E por fim, agradecemos também aos nossos familiares, que sempre se mostraram do nosso lado, nos motivando e orientando, além dos nossos companheiros de graduação que nos auxiliaram compartilhando suas experiências.

# Sumário

# Resumo

A necessidade pela busca de informações cruzadas entre diferentes plataformas ganha força com o passar do tempo devido ao aumento da volumetria de dados gerados pelos diversos *Software*s existentes, sejam eles complexos, como os *wearables* ou simples, como planilhas de controle. Atualmente o maior poder que alguma companhia ou entidade pode deter, são os dados, perfeitamente validados, limpos, e direcionais de pessoas. Com isso, diversas técnicas de cruzamentos de dados, independente de sua origem, seja via cadastro direto em um aplicativo, envios de qualidade de um aplicativo operando no seu dispositivo, gravações de áudio periódicas com finalidade de aprimoramento social ferramental (como o *google assistant*) estão sendo aplicadas, chegando a criar empresas com baixo balanço financeiro, mas que valem milhões ou até mesmo bilhões pela quantidade de informação agregada que as mesmas detêm. Neste estudo de caso é construído uma aplicação que coleta dados direcionais, em diferentes formatos, cunhando o termo *Big Data*, mas simples, onde uma das fontes de informação se trata do *Twitter*, uma das maiores redes sociais do mundo. O estudo tem como objetivo elucidar a trativa de dados nos diferentes pontos de uma aplicação, que será construída como instrumento de estudo para esse trabalho, com coleta, processamento e armazenamento de dados, que estão em posições diferentes de geolocalização. Visando analisar a influência da governança de dados perante a legislação vigente, a LGPD, e a administração dos dados, assim como elucidar técnicas que podem simplificar tratativas de dados até se tornar informações, baseando-se em um cenário atual de desenvolvimento de*Software*.

Palavras-chave: *Big Data*, *Software*, aplicação, dados, geolocalização, governança, legislação.

# Abstract

The need to search for cross-information between different platforms gains strength over time due to the increase in the volume of data generated by the various existing software, whether complex, such as wearables or simple, such as control spreadsheets. Currently the greatest power that any company or entity can hold, is the data, perfectly validated, clean, and directional of people. As a result, various data crossing techniques, regardless of their origin, either via direct registration in an application, quality submissions from an application operating on your device, periodic audio recordings for the purpose of improving social tools (such as google assistant) are being applied, even creating companies with low balance sheets, but worth millions or even billions for the amount of aggregated information they hold. In this case study, an application is built that collects directional data, in different formats, coining the term Big Data, but simple, where one of the sources of information is Twitter, one of the largest social networks in the world. The study aims to elucidate the transmission of data at different points of an application, which will be built as a study instrument for this work, with collection, processing and storage of data, which are in different positions of geolocation. Aiming to analyze the influence of data governance under current legislation, the LGPD, and data management, as well as to elucidate techniques that can simplify data processing until it becomes information based on a current scenario of software development.

Key-words: Big Data, Software, application, data, geolocation, governance, legislation.

# CAPÍTULO 1

# Introdução

## 1.1 Motivação

Vivemos em uma sociedade que está cada vez mais conectada à internet. Segundo dados da Pesquisa Nacional por Amostra de Domicílios (PNAD), em 2021 aproximadamente 90 % dos lares brasileiros tinham acesso à internet através de um ou mais dos seguintes equipamentos: tablets, microcomputadores, televisões e celulares. Outro dado muito interessante é que, o telefone celular é utilizado em 99,5 % dos domicílios que possuem acesso à internet. [1].

Toda essa conectividade gera um volume enorme de dados, e como o número de pessoas com acesso à internet continua aumentando, assim como o número de dispositivos conectados à internet, esse volume de dados não irá parar de crescer. Um estudo realizado pela *International Data Corporation* (IDC), principal fornecedora global de inteligência de mercado que envolve tecnologia da informação, prevê que até o ano de 2025, teremos produzido 175 Zettabytes de dados [2]. É interessante ressaltar que, esses dados podem vir de diversas fontes e em diferentes formatos, como por exemplo: redes sociais, vídeos, áudios, *e-mails*, mensagens, transações bancárias, imagens, documentos, geolocalização, etc.

Esse volume enorme de dados, que está crescendo exponencialmente a cada dia e vindo de diversas fontes e em diferentes formatos (dados estruturados e não estruturados) começou a ser conhecido como *Big Data*. Não demorou muito para que as empresas percebessem que informações valiosas e conhecimento estavam encobertos por esse enorme volume de dados, somente aguardando o garimpo. E foi exatamente isso que aconteceu, profissionais da área começaram a "garimpar" esses dados, realizando análises e as mais diversas combinações dos mais diferentes tipos de dados, visando extrair informações que auxiliariassem na tomada de

decisão das empresas e até mesmo no surgimento de novas ideias. Assim, dados que até então estavam, aparentemente, só ocupando espaço e consumindo recursos de armazenamento, agora se tornam essenciais para o sucesso de uma empresa.

Através dessas análises, por exemplo, empresas mapearam perfis de seus principais consumidores e direcionaram seus esforços em *marketing* para esse grupo de pessoas, outras através de um histórico de dados fizeram uma previsão de demanda para os meses do ano, e alinharam sua produção para atender seus clientes sem, ou com o menor desperdício possível de matéria prima, e tantos outros casos, dentre uma infinidade de cenários.

Contudo, como muitos desses dados, fornecidos por pessoas, começaram a ser utilizados para um propósito diferente do proposto inicialmente, e ocorreram casos grandes e graves de vazamentos, uma preocupação com relação à proteção de dados pessoais começou a surgir. Nesse contexto surgiram leis e regulamentos como por exemplo, o *General Data Protection Regulation* (GDPR) na União Europeia e a Lei Nº 13.709, de 14 de agosto de 2018, conhecida como Lei Geral de Proteção de Dados (LGPD) no Brasi [3].

Essas leis e regulamentos foram criadas com o intuito de proteger a privacidade das pessoas, um direito fundamental, analisar até onde o uso dos dados coletados está sendo juridicamente correto e estipular punições e trativas, no caso de vazamentos e má administração dos dados coletados. Também se faz necessário uma conscientização dos titulares dos dados quanto ao seu valor e confidencialidade, uma vez que muitas pessoas compartilham dados sensíveis sem se atentar para os cuidados necessários. [4].

Esse cenário é relativamente novo, não somente no Brasil, mas no mundo. A LGPD, por exemplo, embora seja uma lei de 2018, entrou em vigor a partir de agosto de 2020. Tendo isso em vista, muitas empresas estão ainda buscando se enquadrar no que é proposto pela lei e ainda existem muitas dúvidas com relação à captura, armazenamento e análise dos dados coletados. Sendo assim, este trabalho se propõe a analisar esse processo (coleta, tratamento, armazenamento, gerenciamento e análise) sob a óptica das leis que regulamentam o assunto.

## 1.2 Casos de Vazamento de Dados

Embora as tecnologias atuais tenham conferido um alto nível de proteção de dados, é possível encontrar na literatura vários casos em que houve vazamento de dados importantes, com grande impacto para as empresas.

Um dos casos que atraiu grande atenção da comunidade internacional envolveu a empresa de *marketing* e publicidade *Cambridge Analytica*, em 2018, que oferecia como serviço principal, análises de dados de comportamento de usuários com o intuito de direcionar as propagandas de forma mais acertiva ao público visado, que poderia ser consumidores e até mesmo eleitores [5]. No dia 17 de março de 2018, os jornais *The Observer*, *The Guardian* e *The New York Times*,

publicaram um artigo intitulado como "*How Trump Consultants Exploited the Facebook Data of Millions*", que pode ser traduzido de forma livre como: "Como os consultores de Trump exploraram os dados do Facebook de milhões" [6]. Em resumo, o artigo relatava o vazamento de uma enorme quantidade de dados de usuários do *Facebook* por parte da empresa *Cambridge Analytica*. Esses dados foram adquiridos pela empresa ao lançarem um aplicativo de teste psicológico na rede social [7].

Essa questão ficou ainda mais complicada quando foram levantadas suspeitas sobre o uso desses dados para influenciar nas eleições de 2016 dos Estados Unidos da América e no *Brexit* (saída do Reino Unido da União Europeia (UE)). Os dados coletados através do aplicativo *thisisyourdigitallife* foram utilizados para um propósito diferente do apresentado inicialmente, que era, fins acadêmicos. Vale ressaltar que o aplicativo foi desenvolvido por Aleksandr Kogan, um pesquisador da Universidade de *Cambridge*, no Reino Unido, que já tinha uma pesquisa sobre como deduzir a personalidade de uma pessoa e suas inclinações políticas através de seus perfis no *Facebook*. [8].

O problema se agravou ainda mais quando foi obsevado que não apenas os dados pessoais do perfil dos usuários do *Facebook* que utilizaram o aplicativo foram coletados, mas também, os dados dos amigos de cada um dos usuários. Segundo Nicole Nguyen, devido à maneira como a plataforma de terceiros do *Facebook* funcionava naquela época, os mais de 270.000 participantes da pesquisa que consentiram em entregar seus dados, também deram a Kogan acesso a dezenas de milhões de dados de seus amigos no *Facebook*. Além disso, a jornalista diz que Kogan informou ao *Facebook* e aos seus usuários que os dados seriam anonimizados, mas não foram [8].

No tocante a efetividade do tratamento dos dados, Aleksandr Kogan declarou em um entrevista para a BBC o seguinte: "Dado o que sabemos agora, nada, literalmente nada - a ideia de que esses dados são precisos, eu diria que é cientificamente ridícula" [9]. David Sumpter, um professor de matemática aplicada na Universidade de Uppsala, na Suécia, analisou a precisão dos modelos da *Cambridge Analytica*, e também chegou na mesma conclusão que Kogan, em seu livro "*Outnumbered*" [10].

No Brasil, dois casos de vazamento de dados foram destaques na imprensa nacional. O primeiro deles está associado à operação da Polícia Federal denominada de *Deepwater*, criada para investigar atos criminosos relacionados à obtenção, divulgação e comercialização de dados pessoais de brasileiros, inclusive autoridades públicas. O que aconteceu foi que, em janeiro de 2021, uma quantidade enorme de dados pessoais (CPF, CNPJ, nome completo, endereço e etc) foram disponibilizados em um fórum na *Internet*, onde eram realizadas trocas sobre atividades cibernéticas. Uma parcela desses dados foi divulgada de forma gratuita, como que para atrair o interesse dos compradores. Em seguida, o restante dos dados foi posto à venda, podendo ser adquiridos por meio do pagamento em criptomoedas [11].

Após uma longa e complexa investigação, a Polícia Federal identificou o suspeito, bem como um segundo *hacker*, que estava vendendo os dados através de suas redes sociais. Esse foi um caso muito sério pois, mais 223 milhões de CPFs, além de outras informações como nomes, endereços, renda, imposto de renda, fotos, beneficiários do Bolsa Família e *scores* de crédito, foram colocadas à venda [12]. Essa é uma situação muito delicada pois, com posse desse dados, pessoas mal intencionadas podem prejudicar, e muito, os titulares desses dados.

O outro caso de grande destaque no Brasil ocorreu em 2022, quando o Banco Central informou em setembro do mesmo ano, havia ocorrido o vazamento de dados vinculados a 137,3 mil chaves Pix. Como existem situações em que um cliente pode ter mais de uma chave Pix, o Banco Central notificou que o número de pessoas afetadas chega a 137.122. Segundo eles, os dados que vazaram eram apenas dados cadastrais (não afetam a movimentação de dinheiro), e dados mais sensíveis como saldos, senhas e extratos não foram expostos. O vazamento ocorreu entre os dias 1 e 14 de setembro e os dados vazados foram: nome do usuário, Cadastro de Pessoas Físicas (CPF), instituição de relacionamento, agência, número e tipo da conta, data de criação da chave Pix. [12]. O Banco Central disse que todas as pessoas que tiveram informações expostas serão notificadas [3], mas o fato é, infelizmente, que não há como garantir que tais informações não serão utilizadas de maneira mal intencionada.

## 1.3 Objetivos

### 1.3.1 Objetivo Geral

O objetivo geral do trabalho é mapear, contextualizar e analisar impactos em projetos de desenvolvimento de *Software* gerados pela governança de dados pós sancionamento da LGPD em seu nível macro e em seu nível micro. Onde para o nível macro tem-se as relações legislativas, momento na qual o principal controle da informação é restrito a localização geográfica da informação, enquanto para o nível micro, tem-se a forma na qual uma instituição garantirá a qualidade do dado, sua segurança e principalmente a distribuição da informação de forma sigilosa entre as áreas de negócio apresentando técnicas de como lidar com o fluxo corrente da informação entre os envolvidos, dos titulares as tomadas de decisões geradas pelas informações coletadas.

### 1.3.2 Objetivos específicos

Os objetivos específicos desse trabalho são:

1. Analisar, no cenário atual, principalmente de infraestrutura em nuvem, o impacto de ter dados distribuídos em localizações distintas ao redor do globo perante a LGPD e a distribuição da informação.

2. Analisar o percurso de uma informação na aplicação construída, que captura dados proativamente de seus titulares e reativamente na base de processamento assíncrono buscando informações em redes sociais, no caso o *Twitter*, até o momento que passe a ser gerida por analistas de dados, e entender como a governança dessas informações são aplicadas.

3. Apresentar medidas tratativas que podem ser adotadas pelas empresas no quesito da legislação vigente (LGPD) sobre o dado e no âmbito da convenção de dados interna.

4. Divulgar boas práticas quando o assunto é captura, armazenamento e análise de dados, assim como, difundir o conhecimento sobre a LGPD.

# CAPÍTULO 2

# Fundamentação Teórica

## 2.1 *Big Data*

A tradução literal de *Big Data* significa “grandes dados”, mas isso não nos ajuda a compreender esse termo tão falado nos dias de hoje. Realmente, definir *Big Data* não é algo trivial, e muitas pessoas ao longo dos últimos anos tem tentado fazê-lo. Doug Laney, da empresa *Gartner Group*, definiu o termo como os 3 V’s: Volume, Velocidade e Variedade [13]. Portanto, pode-se entender como *Big Data*, um grande volume de dados, que foi gerado a uma alta velocidade, tendo origens diversas e estando em formatos diferentes.

Com o passar do tempo, foram adicionados mais 2 V’s à definição de *Big Data*, são eles: Veracidade e Valor. Formando assim os 5 V’s do *Big Data*.

Figura 2.1: Os 5 V’s do *Big Data*


Fonte: https://www.cortex-intelligence.com/blog/os-5-vs-do-big-data

Para melhor compreensão do termo, faz-se necessário o detalhamento de cada um dos V's, considerados os pilares do *Big Data*, de forma sucinta:

**Volume:** O principal pilar está estruturado com a quantidade de informação, pois ela é extremamente relevante neste assunto, como nos dias atuais praticamente qualquer atividade realizada é uma fonte potencial informativa, como consequência, tem-se uma grande massa de dados para filtrar e ajustar com a tentativa de coletar a métrica objetiva.

**Velocidade:** Quanto ao uso estratégico deste conceito, tem-se um outro ponto que é estritamente importante, que pode superar até o pilar Volume em determinadas situações, pelo fato de ser a força que rege o mercado competitivo e se trata da velocidade na qual os dados são administrados. O conceito pode variar dependendo do objetivo da utilização do dado, um exemplo seria a taxa a qual é processada uma certa quantidade de espaço de disco, ou até mesmo uma velocidade que nos permitir processar uma grande quantidade de informações em tempo real.

**Variedade:** As informações de amostragem devem vir de diversas formas, ou seja, devem existir várias formas de dados, podendo ser dados existentes localmente via banco de dados, aplicativos, *Internet of Things* (IoT), *Global Positioning System* (GPS), *cookies*, entre outras. A realidade é que nos dias atuais é mais simples classificar algo inventado que não gere informação do que o contrário. Neste pilar já tem-se a referência de maior complexidade, onde normalmente não está em relacionar informação, mas sim, no seu pré processamento onde o conteúdo é tratado e classificado, como exemplo um áudio que é transcrito e ganha seus metadados apropriados.

**Veracidade:** Considerado um dos pilares recentes, e está relacionado diretamente a quanto uma informação obtida pode ser considerada verdadeira, e este pilar pode ser de certa forma enxugado da forma que a informação é obtida. Porém, se tratando de *Big Data* a maneira na qual a informação é minerada faz-se necessário gastar uma certa quantidade de energia neste processo. Em processos empresariais, como ações de *marketing*, uma grande quantidade de dados podem ser obtidos através de redes sociais, um bom exemplo é a coleta de informações via *Twitter* que pode ser corrompida por falsas informações, como *fake news* (notícias falsas).

**Valor:** Um outro excelente pilar, e considerado o principal no quesito negócio, é o valor de negócio na qual o tratamento da informação pode gerar, ou seja, nada adianta o tratamento e processamento da informação se nenhum ganho ou métrica pode ser gerada com seu resultado, se o seu espalhamento for muito alto.

Uma outra definição bem interessante, é a do Anil Maheshwari, que em seu livro define o termo *Big Data* como dados extremamente grandes, muito rápidos, diversos e complexos, que não podem ser gerenciados através de ferramentas tradicionais. O autor ainda traz em sua definição a função do *Big Data*, que segundo ele, tem como principal papel, fornecer as informações certas, para as pessoas certas, para ajudar a tomar as decisões certas. [14].

Com toda certeza existem outra definições para *Big Data*, que variam dependendo da área de conhecimento em que são aplicadas, porém para o trabalho em questão, será adotado uma visão simplista, enxergando o termo como uma quantidade razoável de dados, gerados de formas distintas, que quando tratados e analisados corretamente, geram valor para seus proprietários.

### 2.1.1 História do *Big Data*

Embora o termo seja recente, quando analisado sua definição, que por sinal é relativa ao tempo, surge um questionamento do quão grande um conjunto de dados deve ser para ser considerado *Big Data*. Como não existe uma definição precisa, alguns acadêmicos dizem que é a quantidade suficiente para uma máquina convencional não ser capaz de conseguir analisar, enquanto alguns estipulam um valor fixo de espaço em memória. quando analisado da primeira maneira, John Graunt pode ser considerado um dos primeiros cientistas de dados, pois em 1663 o mesmo utilizou uma grande quantidade de informação, de fontes diferentes, para estudar a epidemia da peste bubônica na Europa, e considerando a capacidade de processamento e armazenamento de informações, evidentemente manual, da época, pode-se dizer que Graunt estava lidando com *Big Data*. [15].

Um outro marco aconteceu em 1937 durante a administração de Franklin D. Roosevelt nos EUA. Com a iniciativa da lei de seguridade social, fez-se necessário que o governo acompanhasse a contribuição de aproximadamente 26 milhões de americanos e seus 3 milhões de empregadores. A empresa *International Business Machines Corporation* (IBM) então, conseguiu o contrato para desenvolver uma máquina de leitura de cartões perfurados, que lidava de certa forma com uma grande quantidade de dados com um grande valor. Seguindo esta linha cronológica chega-se nos grandes acontecimentos que envolvem grandes quantidades de pessoas no globo, com processamento de informação dentro do século XIX, alguns exemplos são [16]:

- Máquina decifradora de códigos Nazista (alta taxa de processamento).
- Construção do primeiro centro de dados nos EUA.
- Invenção da *World Wide Web* (WWW).

Seguindo este fluxo, nos dias atuais, onde têm-se uma melhora extremamente significativa em todos os pilares do *Big Data*. É incomparável o aumento da capacidade de processamento

e armazenamento de dados, porém, em contrapartida temos a maior geração de informação de todos os tempos, com estudos indicando seu aumento de forma exponencial.

### 2.1.2 *Big Data* no Contexto Atual

A aplicabilidade do *Big Data* pode ser observada nas mais diversas áreas, trazendo sempre, melhorias nos processos organizacionais e apoiando a tomada de decisões [17]. Para muitos, os dados são considerados o novo "petróleo", e essa analogia é muito interessante pois, assim como o petróleo, a exploração de dados, é complexa. Todavia, apesar dessa complexidade, hoje têm-se muitos exemplos, em diversas áreas, de como esse processo é recompensador. A seguir, serão listados alguns exemplos de como o *Big Data* vem sendo aplicado em algumas áreas.

#### Mobilidade urbana

Algumas empresas como *Waze* (Aplicação mobile que utiliza navegação por GPS e tem como funcionalidade traçar rotas entre dois pontos do globo) e UPS (uma das maiores empresas do mundo no ramo de logística), utilizam *Big Data* para cruzar informações de um certo trajeto com o cenário atual das vias, e dessa forma conseguem tomar a decisão de qual a rota menos custosa para se atingir um certo objetivo.

#### *Ecommerce*

Se tratando de grandes empresas, deve-se notar que a maioria delas, senão todas, fazem o uso de técnicas de *Big Data* para criarem vantagens competitivas em relação aos seus concorrentes. Essas empresas utilizam os dados do perfil de seus consumidores e observam seus perfis de navegação (quais *sites* acessam, o que buscam, etc) para definir em tempo real quais produtos serão oferecidos a esse público [18].

Empresas com o foco em vendas, como o grupo Pão de Açúcar, Amazon, dentre outras, através do histórico de vendas, podem determinar quais produtos são mais vendidos e planejar de uma forma mais acertiva seus estoques, evitando assim que produtos de baixo giro ocupem o lugar que poderia estar sendo utilizado para produtos com maior demanda.

Um outro dado que pode ser bem explorado é o histórico de compras do usuário, somado aos seus dados cadastrais (sexo, idade, região onde vive, etc), que podem ajudar as empresas a criarem um mecanismo de recomendação, dessa forma, quando o usuário acessar a página da empresa irá ver ali anúncios de produtos que tenha intenção de comprar.

Tudo isso é muito interessante, pois é possível classificar o consumo em diversos níveis de granularidade, as empresas podem classificar o padrão de consumo de uma pessoa, assim como os padrões de consumo de regiões, cidades e até estados.

**Saúde**

Recentemente o mercado recebeu inúmeros dispositivos que estão coletando dados e gerando uma massa de dados incrível para saúde, um exemplo são os *wearables* que representam quaisquer dispositivos tecnológicos que possam ser usados como acessórios, como os *smartwatches*, ou mesmo ser vestido por completo. Ou seja, no caso dos relógios inteligentes, os padrões de monitoramentos vitais estão sendo comparados a todo momento com possíveis doenças, e dessa forma algumas marcas estão fazendo sugestões de consultas para seus clientes baseado nessas informações.

**Serviços Financeiros**

Uma área que faz uso das técnicas e ferramentas de *Big Data* é a aréa de análise de risco no setor financeiro. Quando uma empresa concede crédito a um consumidor existe uma probabilidade do cliente não honrar o compromisso de quitar sua dívida. Os dados considerados nas análises são, por exemplo, a situação financeira atual do solicitante, se o solicitante está empregado, sua renda mensal, o percentual de sua renda que já está comprometido em outras dívidas, seu histórico com a empresa (caso já tenha solicitado outros empréstimos) e seu histórico com o mercado, se ele honrou os compromissos com outras empresas do mesmo ramo. [19].

Hoje em dia, os resultados financeiros das empresas são divugaldos por mês, trimestre, quadrimestre, semestre ou ano. Os bancos, por exemplo, tem acesso a esses dados, e na hora de conceder um empréstimo a esses empresas, pode analisar seus resultados financeiros, suas transações nos últimos meses, suas dívidas, resumindo, a situação financeira da empresa como um todo, e depois decidir se ira conceder o empréstimo e sob qual percentual de juros.

## 2.2 Governança de Dados

Os benefícios gerados através de soluções tecnológicas que utilizam *Big Data* para as organizações e empresas são inquestionáveis. Conforme já dito anteriormente, hoje em dia, os dados são os ativos mais valiosos que as empresas possuem. Todavia, administrar essa tão grande quantidade de dados não é uma tarefa fácil. Para suprir essa demanda, uma área que até então estava ”escondida” dentro das empresas, passa a ganhar cada vez mais espaço, a área de governança de dados.

Dentre as responsabilidades da área de governança de dados estão o exercício de autoridade e controle de estratégias, políticas, papéis e atividades ligadas aos ativos de dados dos negócios [20].

A governança de dados se atenta para como as decisões são tomadas sobre os dados que a empresa ou organização possui e como as pessoas e os processos se comportam em relação aos

dados. Fato é que todas as empresas ou organizações tomam decisões sobre dados (quais dados vão coletar, quais tratamentos vão aplicar, como vão armazenar, etc), tendo, ou não, uma função formal de governança de dados. Todavia, empresas que possuem uma área de governança de dados apresentam maior autoridade e controle sobre os dados, e consequentemente, conseguem explorar melhor esse ativo tão precioso, que são os dados. [21].

### 2.2.1 Fundamentos da LGPD

Antes de 2014, no Brasil, não existia uma norma específica que garantisse a privacidade e a segurança das informações dos usuários da *Internet*. Devido a essa falta de regulamentação específica, na ocorrência de violação da privacidade de dados de um usuário da *Internet*, era utilizado o inciso X, do art. 5 da Constituição Federal de 1988, que afirma que "são invioláveis a intimidade, a vida privada, a honra e a imagem das pessoas, assegurado o direito a indenização pelo dano material ou moral decorrente de sua violação" [22].

Dando sequência, com o objetivo de regulamentar o uso da *Internet*, no Brasil, em 2014, foi sancionada a Lei Nº 12.965, no dia 23 de abril, que ficou conhecida, popularmente, como o Marco Civil da *Internet*. Tal lei estabelece princípios, garantias, direitos e deveres para o uso da *Internet*. Tendo em vista o objetivo do trabalho, vale ressaltar que, no art. 7 desta lei, são assegurados, dentre outros, os seguintes direitos ao usuário: não fornecimento a terceiros de seus dados pessoais, salvo mediante consentimento livre, expresso e informado do usuário; informações claras e completas sobre a coleta, uso, armazenamento, tratamento e proteção de seus dados pessoais, sendo que estes, só poderão ser utilizados para fins que justifiquem sua coleta e que tenham sido especificados nos contratos de prestação de serviços e/ou termos de uso de aplicações de *Internet*. Vale ressaltar a importância da leitura desses documentos pelo usuário, antes do fornecimento dos dados requisitados. [23].

Um outro fundamento muito importante para a LGPD foi, e é, o *General Data Protection Regulation* (GDPR). Aprovado em 27 de abril de 2016, o GDPR propunha abordar a proteção das pessoas no tocante ao tratamento e circulação de seus dados pessoais [24]. A GDPR é composta por 99 capítulos que dentre outras coisas, concedem aos titulares mais direitos sobre os seus dados e mais responsabilidades aos processadores de dados, que devem atuar de forma mais segura e responsável [25].

### 2.2.2 LGPD

Tendo como base a GDPR, aplicada nos países da União Europeia, a Constituição Federal de 1988 e o Marco Civil da *Internet*, em 14 de agosto de 2018, foi sancionada a Lei Nº 13.709, conhecida como a Lei Geral de Proteção de Dados Pessoais (LGPD). Tal lei tem, como objetivo, proteger os direitos fundamentais de liberdade e de privacidade e o livre desenvolvimento da

personalidade da pessoa natural [3]. De forma simplista, a LGPD trata os direitos dos titulares e obrigações dos agentes de tratamento. A LGPD é composta por 10 capítulos, que serão abordados individualmente a seguir. Vale ressaltar que, os capítulos não serão explorados em sua íntegra, mas terão ênfase nos pontos principais que servirão de base para este trabalho.

**Capítulo 1**

O Capítulo 1 da LGPD trata das disposições preliminares. No artigo 2 são apresentados os fundamentos da lei em questão, dentre eles estão: o respeito à privacidade, a inviolabilidade da intimidade, da honra e da imagem e o desenvolvimento econômico e tecnológico e a inovação. Fundamentos esses muito importantes para a construção de uma lei que tem como objetivo sim, proteger os dados pessoais, mas também entende que esses dados integram os pilares para o desenvolvimento econômico e tecnológico do país.

No artigo 5 ela traz definições importantes, como por exemplo a definição de dado pessoal e dado pessoal sensível, que sim, são coisas diferentes. Segundo a LGPD, dado pessoal é uma informação relacionada a pessoa natural identificada ou identificável e dado pessoal sensível, são dados que estão relacionados à origem racial ou étnica, convicção religiosa, opinião política, filiação a sindicato ou a organização de caráter religioso, filosófico ou político, dado referente à saúde ou à vida sexual, dado genético ou biométrico. Tais definições são muito importantes, principalmente para os profissionais que trabalham com dados. Saber mapear quais dados são dados pessoais e dados pessoais sensíveis é de suma importância para o desenvolvimento de um processo seguro e alinhado com a lei para coletar, armazenar, tratar e gerenciar dados. Também nota-se as definições de titular, controlador e operador. O titular é a pessoa a quem se referem os dados pessoais, o controlador é a pessoa a quem compete as decisões referentes ao tratamento de dados pessoais e por fim, o operador, é quem realiza o tratamento.

Para finalizar, uma outra definição importante é o que seria o tratamento de dados, a LGPD no art. 5º, define tratamento como:

> X - tratamento: toda operação realizada com dados pessoais, como as que se referem a coleta, produção, recepção, classificação, utilização, acesso, reprodução, transmissão, distribuição, processamento, arquivamento, armazenamento, eliminação, avaliação ou controle da informação, modificação, comunicação, transferência, difusão ou extração [3].

E para fechar o capítulo 1 da LGPD, o artigo 6 nos informa quais princípios o tratamento de dados deve observar. O tratamento de dados precisa ter uma **finalidade**, que precisa ser apresentada para o titular de forma muito clara e após o consentimento do titular dos dados, o tramento deve seguir o que foi acordado, o que nos leva ao segundo princípio, **adequação**. Também deve ser levado em consideração a **necessidade** de dados, considerando a finalidade do tratamento. Deve-se usar o mínimo de dados pessoais possíveis, evitando assim problemas

desnecessários. Outros 3 princípios são, o **livre acesso**, os proprietários dos dados pessoais devem ter acesso faciltado às informações disponibilizadas e quando necessário devem ser capazes de terem seus dados atualizados, garantido a **qualidade dos dados** e toda essa relação deve ser fundamentada na **transparência**, para que o titular dos dados saiba exatamente qual(is) tratamento(s) está(ão) sendo realizado(s) e sua(s) finalidade(s). Tudo isso sempre se atentando para a **segurança** e **prevenção**, fazendo uso de ferramentas que possíbilitem a proteção dos dados e adotando medidas preventivas para evitar incidentes. E por fim, de forma alguma os dados podem ser utilizados para fins discriminatórios, **não discriminação**, e no caso de incidentes (vazamento de dados), deve-se ser observado o princípio de **responsabilização e prestação de contas**, onde devem ser apresentados aos titulares dos dados as medidas que estão sendo tomadas como trativa do incidente ocorrido.

### Capítulo 2

O capítulo 2 da LGPD aborda o tratamento de dados pessoais. No primeiro artigo do capítulo, art. 11, são apresentadas as hipóteses onde o tratamento de dados pessoais sensíveis pode ocorrer. Salvo as exceções dispostas, o tratemento só pode ocorrer mediante o consentimento do titular, ou em caso de menoridade, de seu responsável, para finalidades específicas, ou seja, os dados não podem ser usados para um propósito diferente daquele informado inicialmente ao usuário. Vale ressaltar que o consentimento pode ser revogado a qualquer momento mediante manifestação expressa do titular, conforme previsto no § 5º do art. 8.

Segundo o art. 12, quando os dados forem anonimizados, estes não serão mais considerados dados pessoais pela LGPD, a não ser que o processo de anonimização seja revertido. Finalizada a etapa de tratamento, os dados devem ser eliminados, uma vez que não são mais necessários, e o objetivo inicial do tratamento foi alcançado. O portador dos dados só poderá manter esses dados para atender as seguintes finalidades, dispostas no art.16: cumprimento de obrigação legal ou regulatória pelo controlador (inciso I); estudo por órgão de pesquisa, garantindo sempre que possível a anonimização dos dados pessoais (inciso II); compartilhamento com terceiros respeitando os requisitos da LGPD (inciso III); uso exclusivo do controlador dos dados anonimizados (inciso IV). Nos demais cenários, os dados pessoais devem ser eliminados.

### Capítulo 3

O capítulo 3 da LGPD é um capítulo muito importante, pois apresenta os direitos dos titulares, onde são respeitados a liberdade, a intimidade e a privacidade. Nesse capítulo é evidenciado a importância dos portadores dos dados terem absoluto controle sobre todos os dados que estão gerenciando, pois o titular pode requisitar, em qualquer momento, o acesso aos dados, a correção de dados, a anonimização dos dados, dentre outras coisas, dispostas no art.18.

O art.20 traz um direito bem importante dos titulares, que é o direito de solicitar a revisão de decisões, tomadas a partir do tratamento de dados que afetem os seus interesses. Por exemplo, um banco faz uso de dados pessoais de uma pessoa para determinar o seu limite de crédito, se essa pessoa entender que a análise trouxe prejuízo para ela, ela pode solicitar a revisão.

Todos os capítulos da LGPD são importantes, mas esse capítulo se destaca por apresentar os direitos dos titulares, que devem sim, se interessar e aprender sobre, para garantir a segurança de suas informações pessoais.

**Capítulo 4**

O capítulo 4 da LGPD trata sobre o tratamento de dados pessoais pelo poder público. Um ponto interessante é que o Poder Público pode coletar informações pessoais, sem o consentimento dos titulares, conforme previsto no art.11.

**Capítulo 5**

O capitulo 5 da LGPD trata da transferência internacional de dados. Logo no primeiro artigo do capítulo 5, art.33, a LGPD nos informa que a transferência de dados para outros países só pode ocorrer, quando esses países apresentarem leis que garantam a proteção dos dados e estejam em conformidade com a LGPD. Isso é bastante interessante, pois hoje em dia os dados são considerados um dos maiores, senão o maior, ativos das empresas, e países que querem continuar se desenvolvendo economicamente e tecnologicamente, precisarão propor leis e ou regulamentos que garantam a proteção de dados, para poderem consumir dados advindos de outros lugares.

**Capítulo 6**

O capítulo 6 da LGPD vai tratar sobre os agentes de tratamento de dados pessoais. Primeiro vale o entendimento de quem são os agentes de tratamento, que são o controlador e o operador. De forma bem simples e resumida o controlador é quem toma as decisões sobre o tratamento, por exemplo, qual(is) tratamento(s) será(ão) realizado(s), o que se busca obter com o(s) tratamento(s), e assim por diante. Já o operador é quem realiza o tratamento em si, portanto é essencial que domine as ferramentas para tal.

Falar sobre tratamento de dados, sem falar sobre organização é impossível. Os agentes de tratamento devem ser muito organizados e manter todos os registros das operações de tratamento realizadas, isso não somente para a sua própria proteção, mas também para a proteção dos dados.

**Capítulo 7**

O capítulo 7 da LGPD fala sobre segurança e boas práticas, que devem ser observadas e adotadas pelos agentes de tratamento, conforme proposto no art.46:

> Art. 46. Os agentes de tratamento devem adotar medidas de segurança, técnicas e administrativas aptas a proteger os dados pessoais de acessos não autorizados e de situações acidentais ou ilícitas de destruição, perda, alteração, comunicação ou qualquer forma de tratamento inadequado ou ilícito [3].

Os últimos anos nos mostraram que incidentes acontecem, e houveram vários casos de vazamento de dados. Quando isso acontece, o controlador, é responsável por comunicar a autoridade nacional sobre o ocorrido, informando, dentre outras coisas, a descrição dos dados pessoais afetados, quais medidas de segurança estavam sendo adotadas, possíveis riscos, conforme previsto no § 1º do art. 48.

Os controladores e operadores devem adotar boas práticas, como por exemplo, a implementação de governança em privacidade. É importante que essa governança apresente, dentre outras coisas, planos de resposta a incidentes e como estes incidentes serão remediados.

**Capítulo 8**

O capítulo 8 da LGPD fala sobre a fiscalização. No caso de incidentes, os agentes de tratamento ficam sujeitos à sanções, que vão aumentando em grau. Eles podem receber advertência, multa simples, multa diária, bloqueio dos dados, eliminação dos dados e até suspensão parcial do banco de dados. Além disso, após averiguação, caso seja realmente confirmada a ocorrência da infração, pode ser solicitado a publicização da infração, ou seja, comunicação do ocorrido para o público. Nota-se que as consequências podem ser mais brandas, ou mais severas, sendo os fatores que influenciam nessa decisão a gravidade da infração, os direitos pessoais afetados, as ações para remediar e minimizar o dano, bem como a velocidade para a adoção das mesmas.

**Capítulo 9**

O capítulo 9 da LGPD vai tratar da Autoridade Nacional de Proteção de Dados (ANPD) e do Conselho Nacional de Proteção de Dados Pessoais e da Privacidade. Também nos é apresenta a composição da ANPD e seu escopo. Cabe a ANPD, dentre outras coisas, zelar pelos dados pessoais através da fiscalização e aplicação de sanções, quando necessário. Outra resposabilidade muito importante da ANPD é a conscientização e o ensino da população sobre a LGPD. Por fim, também é apresentado a composição do Conselho Nacional de Proteção de Dados Pessoais e da Privacidade e suas competências, e novamente, ele também é responsável por disseminar conhecimento sobre a proteção de dados à população.

**Capítulo 10**

O capítulo 10 da LGPD nos traz as disposições finais e transitórias.

A LGPD, com toda certeza, foi um marco na área de proteção de dados no Brasil, e embora recente, ela vem pavimentando o caminho para um melhor armazenamento e tratamento dos

dados coletados de milhões de brasileiros. O futuro é muito promissor, e com a adoção e implementação da LGPD será mantido o progresoo econômico e tecnológico.

### 2.2.3 Gerenciamento de Dados

Neste ponto já se faz necessário a capacidade de uma organização em garantir um nível elevado em relação à qualidade dos seus dados durante todo o percurso da informação e seu ciclo de vida, além disso deve garantir certas seguranças a esta informação e acessos concedidos a privilégios a nível do negócio. Os principais pilares para uma alta qualidade de tratamento de dados são:

- **Disponibilidade:** Basicamente, na jornada da informação, desde a coleta, o dado tem que estar disponível adequadamente para os analistas de dados ou o que (cenário de aplicações em tomadas de decisão) irá operar. O dado não pode estar abundante, ou seja, enriquecido com outros valores que podem gerar vazamento e interconexões problemáticas e explícitas do usuário, como por exemplo, uma equipe de atendimento de um *e-commerce* que lida com processo de estorno não deveria operar sobre os dados de cartão de um cliente bruto, mas sim pelos seus dados tratados, como últimos dígitos, bandeiras entre outras informações.

- **Usabilidade:** Uma vez que o dado existe, ele deve ser de fácil acesso para as áreas de interesse, e de seu fornecedor, no caso do *e-commerce* um cliente deve poder ver as informações relacionadas a ele com extrema facilidade.

- **Consistência:** Os dados devem refletir o mundo externo, e não serem arbitrários, deve ter correlação com um processo factível.

- **Integridade:** O dado tem que ser confiável durante todo seu ciclo de vida, não pode ser corrompido por processos ou agentes externos.

- **Segurança:** Segurança de dados possui uma definição ampla, mas resumidamente, o dado tem que estar sob medidas que protejam contra acessos não autorizados, e dessa forma é evitado os demais pontos acima, geralmente deve ser tratado com criptografia caso sejam exposto, ou seja, deve existir medidas de seguranças em cima da informação mesmo com os devidos acessos.

O principal ponto a se lembrar é que quando se trata de gerenciamento de dados está relacionado à pessoas e processos, assim como suas ferramentas de manuseio, dessa forma faz-se necessário um alto nível de padronização para que a informação seja enriquecida adequadamente, uma expertise na qual geralmente a companhia vai se aprimorando com o tempo para se enquadrar

em certas classificações, que passam uma maior confiabilidade para seus clientes. Um bom exemplo disso são, as exigências que uma empresa precisa ter com seus dados para que possa fazer o *Initial Public Offering* (IPO), ou seja, abrir capital na bolsa.

## 2.3 Metodologias de desenvolvimento de *Software* alinhadas com a LGPD

Nessa seção serão apresentadas algumas metodologias que podem sem utilizadas em projetos de desenvolvimento de *Software* para auxiliar na adequação à LGPD.

### 2.3.1 Mapeamento de dados

Mapeamento de dados, *mapping* ou inventário de dados, é uma das formas ideias de manter *Softwares* em conformidade com a LGPD. Consiste em conhecer profundamente as atividades e sua relação com os dados pertencentes à organização em questão, ou seja, o fluxo de vida de uma informação desde que ela é coletada, até o momento em que ela é descartada. Dessa forma é possível garantir a qualidade da informação e criar um plano de governança apropriada à legislação.

Entender este fluxo pode nos trazer a visibilidade dos potenciais pontos de falhas, desde invasões até mesmo problemas de distribuição de dados ou informações dentro da própria organização. Uma boa prática seria adequar os dados as categorias existentes, por exemplo:

- **Tipos de dados** - Dividir as modalidades de dados em seus tipos baseado em fluxos, como por exemplo, dados cadastrais, especiais, trabalhistas, sensíveis, entre outros. Isso permite direcionar melhor o acesso a informação para certas áreas, e evidentemente restringir os acessos sensíveis a quem não deveria interagir com tal informação.

- **Volume de dados** - A volumetria de dados navegados em um determinado fluxo e a sazonalidade dessa informação.

- **Etapas do fluxo de dados** - Entendimento dos valores de entradas e saídas nas etapas de coleta, armazenagem, sanitização, enriquecimento, processamento, segmentação, inferências, transferências, descarte. Dessa forma, fica evidente a redundância da informação em certos pontos da aplicação.

- **Tecnologias** - Apresentar as tecnologias utilizadas em cada fluxo no formato de documentação. Um bom exemplo é o fato de que com recorrência bibliotecas de encriptamento e análises de fraude ficam depreciadas e com falhas de segurança.

- **Locais de armazenamento** - Informar os locais onde os dados sofrem processamento, coleta e distribuição, pois em caso de falha de informação fica extremamente mais simples identificar a falha descoberta.
- **Origem dos dados** - Deixar explícito a forma de coleta de quaisquer informações que trafegam dentro de um sistema.
- **Campanhas de *Marketing*** - Deixar transparente para os usuários a forma que os dados serão utilizados para esta finalidade.
- **Compartilhamento de dados com parceiros** - Deixar evidente os parceiros com os quais as informações coletadas são compartilhados, sejam eles parceiros de negócio ou de tecnolgoia.
- **Empresas coligadas** - Deixar transparente as empresas coligadas dentro do grupo econômico onde os dados são transacionados.
- **Localidades do tratamento** - Deixar transparente as localidades onde existe atividade para esta informação.
- **Base legal** - Informar a nível de legislação a capacidade de operação dessa informação.
- **Política de privacidade** - Em quaisquer pontos de coleta de dados, a política de privacidade da informação deve estar atualizada e dentro do âmbito da LGPD.
- **Dados de menores de idade** - No cenário onde existe a coleta de informações de menores, deve haver o consentimento do responsável maior de idade, além da confirmação dos documentos informados.
- **Segurança da informação** - Deixar evidente os principais controles de segurança que participam nos processos de transferências de dados.
- **Direito dos titulares** - Garantir que os titulares dos dados possam exercer todos os direitos previstos ao longo da LGPD.
- **Transferência internacional de dados** - Garantir a segurança e integridade dos dados, mesmo com a transferência para outro país. Segundo o Art. 3º da LGPD, a Lei aplica-se a qualquer operação de tratamento realizada independentemente do país de sua sede ou do país onde estejam localizados os dados, desde que os dados pessoais tenham sido coletados no território nacional [3].

Estes são os pontos essenciais no mapeamento de dados que devem ser contemplados.

### 2.3.2 *Privacy by Design*

O conceito de *Privacy by Design* foi criado por um canadense, Ann Cavoukian, e se encontra previsto dentro do artigo 25 do GDPR e também ao longo dos princípios da LGPD. Tem como base sete princípios, sendo eles:

1. **Empresas devem adotar abordagem proativa e não reativa** - Basicamente a ideia inserida neste princípio é não esperar um risco acontecer para tomar uma ação, mas sim se antecipar a este risco de forma com que ele não ocorra.

2. **Sistemas, serviços e produtos devem proteger os dados pessoais de titulares** - A privacidade sempre deve ser o comportamento padrão de qualquer atividade desenvolvida que coloque em risco dados pessoais, e com isso, o usuário não precisa agir para estar dentro das normas mais seguras, ao menos que seja necessário. Com isso, o tratamento de dados deve ser específico, limpo, claro e relevante, priorizando coleta de dados não identificáveis, assim como o uso a retenção da informação deve ser usada somente para o necessário previamente estabelecido na política de dados.

3. ***Design* deve ser incorporado às medidas adotadas para a proteção de dados de titulares** - A segurança das informações dos usuários deve ser incorporada como um critério sistêmico, sempre levada em consideração perante o ponto de vista de arquitetura das aplicações. Para isso faz-se necessário uma abordagem holística, integrativa e criativa.

4. **Empresas não devem coletar mais dados do que o necessário** - A coleta de informações excedentes, que não são usadas nem para o bem da organização, quanto para melhora da segurança do usuário, pode ser um grande fator de atrito caso algum risco aconteça, pois maiores informações do que as realmente utilizadas estarão envolvidas.

5. **Deve ser adotada segurança de ponta a ponta** - Durante todo o ciclo de vida de qualquer informação, deve ser adotado a forma de segurança mais adequada, do contrário, caso algum risco aconteça, deve ser enxergado como falha de segurança grave, durante todo o processo devem ser assegurados a confidencialidade, a integridade e a disponibilidade do dado.

6. **Práticas empresariais devem ser dotadas de visibilidade e transparência** - O objetivo é garantir que todos os processos aconteçam da forma com que está acordado, através da responsabilização, abertura e transparência e compliance.

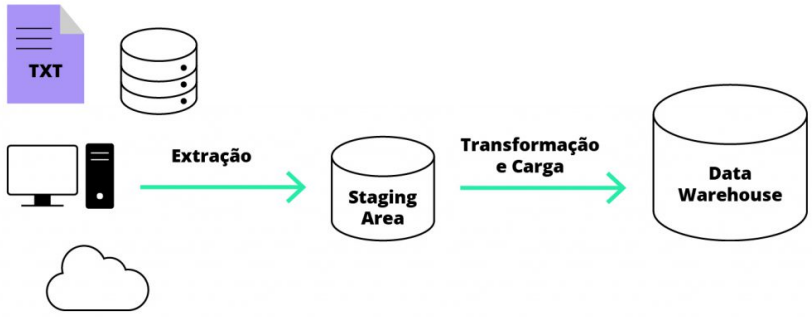
7. **Deve ser respeitada a privacidade do usuário** - A finalidade é manter os interesses e segurança dos dados do usuário acima de tudo, dessa forma, tudo deve ser desenvolvido em prol do usuário, porém com o radar em sua segurança.

Estes princípios representaram uma mudança radical na forma com que a proteção dos direitos e liberdade interagem com o titular, devido ao fato de se posicionar antes mesmo do acontecido. Uma outra consequência positiva de toda essa mudança no âmbito legal, é o fato dos usuários começarem a questionar e posicionar companhias baseado no seu histórico de tratamento de dados e política de informação. Não é por acaso que as empresas mais bem vistas hoje tem uma forte presença na confiabilidade das suas informações e como guardam as informações de quem aposta com elas.

## 2.4 *Extract, Transform, Load* (ETL)

ETL, que pode ser traduzido como extrair, tranformar e carregar, é um processo que combina dados de diversas fontes em um ambiente relacional. Nesse processo, dentre outras coisas, é levado em consideração as regras de negócio da empresa para se realizar a limpeza e o tratamento dos dados, que serão armazenados para auxiliar em análises futuras, projetos de *Machine Learning* e tomada de decisões. [26].

Figura 2.2: Diagrama ilustrando o processo ETL.




Fonte: https://www.mjvinnovation.com/pt-br/blog/o-que-e-etl-como-funciona/

O ETL é composto por três etapas, são elas: extração, transformação e carregamento.

### Extração

Nessa etapa são identificados os dados que o negócio deseja, e estes são transportados para uma área de preparação. Aqui, vale ressaltar a importância dos dados que o negócio possui estarem mapeados, pois sem isso o processo se torna muito mais difícil. Não são poucos os negócios que não conhecem os dados que possuem, e isso limita, e muito, o surgimento de novas análises que poderiam auxiliar na tomada de decisões importantes. Esses dados podem vir de fontes estruturadas, como por exemplo, bancos de dados, planilhas de dados, arquivos (CSV,

XML e JSON), ou de fontes não estruturadas, como por exemplo, imagens, aúdios, mensagens de texto, dentre outras.

#### Transformação

Os dados capturados estão em sua forma original. Normalmente se utiliza o termo ”dados brutos” para se referir a esse conjunto de dados. Por serem ”dados brutos”, eles precisam passar por um processo de lapidação, para que possam atender as necessidades do negócio e essa é a etapa da transformação. Durante esse processo os dados são validados, para garantir veracidade (um dos pilares do *Big Data*), ocorre também a deduplicação, que é um processo onde dados repetidos são eliminados. Também é uma prática comum a junção de dados, conseguindo assim extrair informações novas, derivadas de dados já existentes, e tantos outros tratamentos que visam garantir o melhor uso possível desses dados e a geração de valor para o negócio.

#### Carregamento

Nessa última etapa, os dados já prontos são movidos para um ambiente relacional, onde a área de nogócio terá acesso para realizar suas análises e assim construir um embasamento mais sólido para a tomada de decisões. Um ponto interessante é que é possível restringir o acesso a esses dados, criando perfis de consulta. Dessa forma, é possível garantir que cada pessoa só tenha acesso aos dados necessários para a execução do seu trabalho, minimizando a chance de problemas futuros com a LGPD.

Uma informação adicional é que também existe um outro processo chamado *Extract, Load, Transform* (ELT), que nada mais é que uma variação das etapas do ETL. O que diferencia o ELT do ETL, é que ao invés de mover os dados para uma área de preparação, todos os dados já são direcionados para um ambiente relacional, ficando à disposição para serem transformados/tratados conforme for necessário. Não possui relevância para este trabalho comparar ambos processos, pois isso vai depender muito do contexto de cada negócio.

## 2.5 *Software*

*Software*, nada mais é, do que um conjunto de instruções, dados ou programas que são usados para operar sistemas de computadores e executar tarefas específicas, diferente do hardware que representa aspectos físicos da máquina em questão. Geralmente quando há uma referência a *softwares*, ela se relaciona a aplicativos, páginas, scripts (trechos de código com alguma funcionalidade) ou programas que são executadas em um computador.

Seus dois principais desdobramentos são “*Softwares* de aplicação”, onde é buscado atender

uma necessidade específica, atreladas a regras de negócio e "*Softwares* de sistema" que são projetados para executar o hardware e por sinal acabam se tornando a plataforma de execução de alguma aplicação. O mais importante é entender que seus diversos tipos estão interligados, e suas fragilidades pode corromper o outro tipo de *Software*, porém, os demais tipos, "*Softwares* de programação" que possibilitam que desenvolvedores gerem novos programas (IDE's, compiladores, editores de textos, entre outros), "*Software* de comunicação", basicamente aplicações de comunicação, sejam elas em tempo real ou não (*Whatsapp*, *gmail*, entre outras) e "*Software* para jogos" que são usados como fonte de entretenimento.

Quando se trata de legislação de proteção de dados, esta relacionado, principalmente, as aplicações que executam uma determinada tarefa e possui necessidade de entradas de usuários, e com isso conseguem alterar positivamente ou negativamente a experiência de vida de um possível usuário válido.

### Engenharia de *Software*

Segundo as palavras de Boehm (1976), um renomado professor de ciência da computação americano, dado uma tradução livre, Engenharia de *Software* tem como definição: "Uma aplicação prática de conhecimento científico no design e na construção de programas computacionais gerando documentação associada necessária para desenvolvê-las, operá-las e mantê-las.".

*Design* refere-se às atividades necessárias em elaborar as necessidades do *Software* em questão, evidentemente, que para este ramo deve-se considerar alguns parâmetros, que começam a dar um grau de dificuldade sistêmico, sendos eles, local de armazenamento (data center local ou em nuvem), utilização de ferramentas *open source* (código aberto) ou privada, partir de uma solução pronta ou ter uma ideia nova. E justamente nessas tomadas de decisão, junto a forma na qual o *Software* é escrito que as brechas de segurança, quebras de legislação ou arranjo de dados/informação começam a aparecer.

Uma etapa importante ao desenvolver uma aplicação, é a sua modelagem a nível de sistemas, e neste cenário se detaca um profissional que é extremamente requisitado junto aos engenheiros e desenvolvedores de *Software*, o arquiteto de *Software*, que junto aos desenvolvedores costuma definir a distribuição das aplicações em termos de governança da empresa, e até mesmo decisões de onde os dados devem se encontrar perante ao globo.

### Arquitetura de *Software*

Consiste em definir as funcionalidades dos componentes do *Software*, suas propriedades externas e suas coligações com outros componentes, gerando uma documentação adequada do comportamento do sistema a nível das interligações. A arquitetura estabelecida influência nas decisões da estrutura do sistema, nos tipos de controle e monitoramento, nos seus protocolos de comunicação (síncronos via protocolo HTTP, assíncrono via protocolo AMQP), limitam

o escopo de funcionalidades de um certo componente, além de definir sua distribuição física limitando sua escalabilidade e fatores de qualidade.

# CAPÍTULO 3

# Metodologia

Para atingir os resultados esperados, o trabalho de graduação envolveu a construção de uma aplicação simples, mas com desenho bastante usual, para avaliar e destacar possíveis pontos de vulnerabilidade frente à LGPD. A aplicação desenvolvida manipula dados de usuários diretamente, através dos cadastros, e indiretamente, através de uma busca ativa de engajamento em redes sociais, de forma que tenha uma alta distribuição geográfica de seus segmentos.

Para isso, o projeto seguiu a seguinte estratégia de desenvolvimento, descrita nos seguintes passos:

1. Definição da forma com o que o cadastro seria feito pelos usuários;
2. Definição de como dados seriam obtidos para ver engajamento de usuários em redes sociais a respeito de um tema;
3. Construção da aplicação responsável por permitir o cadastro de usuários proativamente;
4. Construção da aplicação responsável por coletar dados de redes sociais sobre um determinado tema e inserir essas informações em um banco de dados para que possam, ser tratadas e gerar um valor de negócio a nível de tomada de decisões;
5. Definição das ferramentas gratuitas que ajudariam a colocar ambas aplicações em produção de forma distribuída geograficamente em mais de um país;
6. Gerar um relatório com as informações obtidas, unindo as informações coletadas proativamente pelos usuários e o processo de busca de engajamento em redes sociais, utilizando ferramentas com um grande valor de mercado para compreensão de possíveis pontos de vulnerabilidade perante a LGPD;

## 3.1 Métodos e ferramentas utilizadas

Ao longo da execução do trabalho foi necessário utilizar algumas metodologias e ferramentas, comuns à área de desenvolvimento de *Software*, e que serão apresentadas brevemente abaixo.

### 3.1.1 *Test Driven Development* (TDD)

Com o intuito de entregar um *Software* que funcione apropriadamente dentro do tempo estimado para a construção do Trabalho de Graduação, foi tomado como princípio a importância dos testes automatizados, dessa maneira, o código gerado por nós fica de fácil manutenção e com garantias de sua funcionalidade. Como testar um *Software* não se trata de algo tão simples, principalmente falando do grande número de componentes que uma aplicação passa a ter, o mundo do desenvolvimento de *Software* contempla diversas metodologias de como lidar com estes testes. [27].

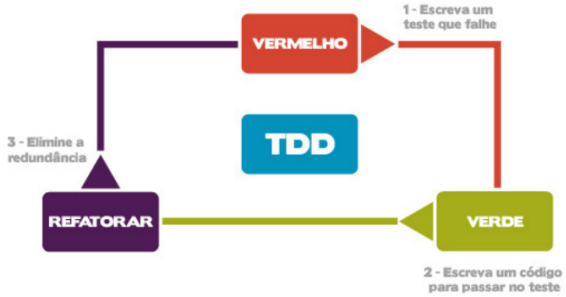
Uma dessas formas, inclusive a contemplada por nós, é o TDD, onde o teste unitário, testes que verificam uma parte específica do código, ou os testes de integração, testes de fluxo de ponta a ponta como a jornada de cliente em uma aplicação, são escritos primeiro, ou seja, antes mesmo do código executor da responsabilidade existir. TDD é o acronimo para *Test Driven Development*, que traduzido significa desenvolvimento orientado a teste, a ideia por trás dessa metodologia é respeitar um ciclo que ocorre na seguinte ordem:

1°. Escrever um teste, unitário ou de integração, que na primeira execução irá falhar, principalmente pelo fato do código executor não existir ainda.

2°. Criar o código executor que satisfaça minimamente o teste escrito.

3°. Refatorar o código executor até que a legibilidade do mesmo se encontre em um nível adequado, e evidentemente o teste quando executado deve passar sem falhar, indicando que a funcionalidade está adequada e com uma boa legibilidade.

Figura 3.1: *Test Driven Development* (TDD)




Fonte: https://www.devmedia.com.br/tdd-fundamentos-do-desenvolvimento-orientado-a-testes/28151

### 3.1.2 *Ruby*

*Ruby* é uma linguagem de programação orientada principalmente para objetos, embora possa ser funcional, é um projeto *open-source* (código aberto) baseado em outras linguagens mais antigas como *Perl* e *Lisp*. Dessa forma os objetos são criados e encapsulados pelas classes, estes modelos possuem os métodos que são responsáveis por guardar tanto os valores dos atributos, quanto executar ações em cima dos objetos. [28]. Como características principais:

- Enxergar tudo como objeto, onde qualquer variável criada implementa as funcionalidades de um objeto.
- Flexibilidade, onde até mesmo as classes bases podem ser sobrescritas.
- Capacidade de tratamento de exceções;
- Presença do *garbage collector*, gerenciador de memória.
- Facilidade no momento de criação de extensões em C.
- Linguagem interpretada.
- Tipagem dinâmica.

O motivo de sua escolha se deve pelo fato da grande material de referência do processamento síncrono em aplicações *WEBs*, o que seria o formulário de cadastro preenchido por um usuário, e nas grandes variações de processamentos assíncronos como a busca de dados de engajamento de usuários, além de ser uma linguagem sólida com muitas bibliotecas que simplificam o caminho, as bibliotecas são conhecidas como gems. Alguns exemplos de bibliotecas utilizadas podem ser vistos abaixo.

#### Gem *Twitter*

Uma interface de comunicação com a API RESTful, modelo de arquitetura de *software*, do *Twitter*, ou seja, uma interface entre dois sistemas de computador que é utilizada para trocar informações seguras pela *Internet*.

#### Gem *Rails* (*framework Ruby on Rails - RoR*)

O *Framework* principal da aplicação, utilizada para desenvolvimento de aplicações *web*, foi criado com o intuito de facilitar o desenvolvimento deste tipo de aplicação. O *framework* utiliza o conceito de *model-view-controller* (MVC) como organização. Nesta abordagem a lógica de programação é dividida em três grandes camadas baseado na sua responsabilidade. A camada de modelo (*Model*), basicamente é responsável por conter as regras de negócio da aplicação e ser a

parte responsável pela integração com o banco de dados. A camada de controladores (*Controller*) tem como responsabilidade receber solicitações ativas do usuário e com isso cria gatilho para algumas ações enquanto a camada de visualização (*View*) representa a parte visual da aplicação e de retorno ao usuário [29]. Dentre seus componentes, os principais utilizados foram:

- *Mongoid* - É considerado um ODM (*Object-Document Mapper*), basicamente possui a responsabilidade de interação com o SGBD (Sistema Gerenciador de Banco de Dados) usado pela aplicação (MongoDB), permite a interação via leitura, escrita e atualização dos registros como documento.
- *Active View* - Componente responsável pela exibição dos dados do lado do *frontend*, interface gráfica vista pelo usuário relativo à aplicação, permite a escrita de páginas HTML com código *Ruby* embutido, conhecido como ERB (*Embedded Ruby*), além de apresentar camadas de visualização totalmente em HTML (Linguagem de marcação de Hipertexto) e JSON (*Java Script Object Notation*) no caso se necessário a utilização de *API's* [29].
- *Active Controller* - Componente responsável pela manipulação dos controladores, agindo como intermediador entre a camada de modelo e de visualização, ou seja, no contexto aqui descrito possuem o papel de receber uma comunicação síncrona e renderizar algo apropriado para o usuário [29].

**Gem *Sidekiq e Sidekiq Cron***

Biblioteca responsável por realizar e gerenciar o processamento assíncrono do lado de uma aplicação *Rails*. O gerenciamento necessita de um banco de dados do tipo chave-valor, que é um tipo de banco de dados não relacional que usa um método de chave-valor simples para armazenar dados, e neste cenário foi escolhido o *Redis*, o *Sidekiq Cron* se responsabiliza no agendamento de tarefas, gerando os próprios gatilhos para aplicação, baseado em um formato específico do tempo.

**Gem *Redis***

Biblioteca responsável pela comunicação com o banco *Redis*, utilizada pelo *Sidekiq* para controlar os gatilhos de processamento assíncrono.

**Gem *RSpec***

Responsável por trazer uma estrutura de testes para o *framework Ruby on Rails*, embora o *framework* MVC tenha por padrão uma outra biblioteca, esta apresenta uma aceitação maior da comunidade da linguagem, pela sua simplicidade e capacidade de adicionar *plugins* que facilitam a etapa de desenvolvimento.

**Gem *Capybara***

*Capybara* é um *framework* de testes de aceitação para aplicações *web*, responsável por simular como um usuário real interage com o aplicativo, para seu funcionamento faz-se necessário um *driver* que roda como intermediador, aqui foi escolhido o *Selenium*,

**Gem *Selenium WebDriver***

*Selenium WebDriver* é uma biblioteca que oferece uma interface que facilita na execução de testes de ponta a ponta. Essa ferramenta realiza chamadas de forma direta ao navegador, fazendo uso do suporte à automação nativo de cada *browser*. O *Selenium WebDriver* não é uma ferramenta de teste independente, é uma *API* que permite através de programação interagir com elementos de uma página, por esse motivo, para um aproveitamento maior costuma ser usado com outra biblioteca de execução de testes como o *Capybara* e *RSpec*.

### 3.1.3 *MongoDB*

*MongoDB* é um banco de dados não relacional que é orientado a documentos, e com isso não usa o conceito de tabelas, colunas ou outra estrutura pré definida e assim, armazena objetos através do formato JSON, que traz a vantagem de ser flexível, não sendo necessário que os registros possuam o mesmo molde de informação. Ele aceita as mesmas operações que os bancos de dados relacionais, sem a necessidade de estabelecer o padrão previamente. Sua divisão como banco de dados se dá nos seguintes componentes [30]:

- Documentos - São as versões binárias dos JSON, que são compostas por chave-valor. Equivalem aos registros na estrutura relacional.
- Coleção - Se dá pelo agrupamento dos documentos, seu equivalente em uma estrutura relacional seriam as tabelas.
- *Database* - Se tratam do agrupamento das coleções, assim como os *databases* na estrutura relacional.

Atualmente existe um formato comum de disponibilizar *Softwares* e soluções de tecnologia por meio da *Internet* com o intuito de diminuir os atritos em ingressar em alguma ferramenta, dado que todo cenário de manutenção, instalação ficam por conta do serviço, este formato é conhecido como SaaS, *Software as a Service*, que traduzido do inglês significa *Software* como serviço. Grandes exemplos são: *Google Drive*, solução de armazenamento e compartilhamento de arquivos em Nuvem, Trello, solução para gerenciamento de projetos e atividades. Dados estes motivos foi utilizado o *MongoDB Atlas*, SaaS, um serviço de hospedagem de instâncias do

*MongoDB* na Nuvem que possuí uma grande parcela de armazenamento gratuita e distribuída para o globo [30].

### 3.1.4 *Redis*

*Redis* é um banco de dados do tipo chave-valor, é de código aberto e uma de suas características é a utilização da memória principal para armazenamento da informação, o que te dá uma alta performance em sua execução.

### 3.1.5 *PowerBI*

Ferramenta da Microsoft para *Business Intelligence* (BI), com a responsabilidade de coletar dados de uma certa origem, pode ser um banco de dados, uma *API*, entre outros, agregar, e unir, gerando algo fácil de ter um resultado a nível de informação. O principal artefato gerado, são os *dashboards* utilizados pela área de negócio, com o intuito de monitoria e tomadas de decisões.

### 3.1.6 *Power Query*

Basicamente é um mecanismo de transformação, onde o objetivo é a transformação com o viés de preparação de dados. Dispõem de uma interface gráfica onde é possível selecionar e trabalhar certas informações, pode ser usado com outras ferramentas para gerar artefatos, como por exemplo o *PowerBI*.

### 3.1.7 *Railway* - SaaS

A *Railway* é uma plataforma de implantação na qual você pode provisionar a infraestrutura, desenvolver com essa infraestrutura localmente e, em seguida, implantar na nuvem. O objetivo do *Railway* é ser a maneira mais simples de desenvolver, implantar e diagnosticar problemas com seu aplicativo.

## 3.2 Procedimentos adotados

### 3.2.1 Criação da aplicação de cadastro

Com o intuito de criar a aplicação, foi iniciado uma nova aplicação com o *framework* em *Rails*. Para isso faz-se necessário a instalação da linguagem na máquina a ser utilizada (algo como sudo apt update && sudo apt install ruby no ubuntu). Com a linguagem instalada na máquina, foi necessário utilizar o *framework* principal para lidar com a estrutura da aplicação *web*, o *Ruby on Rails*, e foi seguido as seguintes interações necessárias até o mínimo da aplicação estar

com as configurações desejadas, *Ruby on Rails* com *MongoDB*, *Redis*, *Sidekiq* e a biblioteca do *Twitter*. Dado esse cenário, a nossa estrutura ficou as seguintes dependências, sendo as relevantes explicadas acima:

```
ruby '3.1.2'

gem 'bootsnap', '>= 1.4.4', require: false
gem 'dotenv-rails'
gem 'jbuilder', '~> 2.7'
gem 'mongoid', '~> 8.0', '>= 8.0.2'
gem 'puma', '~> 5.0'
gem 'rails', '~> 6.1.7', '>= 6.1.7.2'
gem 'redis', '~> 4.4'
gem 'sass-rails', '>= 6'
gem 'sidekiq'
gem 'sidekiq-cron'
gem 'turbolinks', '~> 5'
gem 'twitter'
gem 'webpacker', '~> 5.0'

group :development, :test do
  gem 'awesome_print'
  gem 'pry-byebug'
  gem 'pry-rails'
  gem 'rspec-rails'
  gem 'rubocop'
  gem 'rubocop-performance'
  gem 'rubocop-rails'
end

group :development do
  gem 'listen', '~> 3.3'
  gem 'rack-mini-profiler', '~> 2.0'
  gem 'spring'
  gem 'web-console', '>= 4.1.0'
end

group :test do
  gem 'capybara', '>= 3.26'
  gem 'factory_bot_rails'
  gem 'selenium-webdriver', '>= 4.0.0.rc1'
  gem 'shoulda-matchers'
  gem 'webdrivers'
  gem 'webmock'
end
```

```
42
43 gem 'tzinfo-data', platforms: %i[mingw mswin x64_mingw jruby]
```

Trecho de código: Lista de dependências da aplicação

Com isso, foi criado dentro dos modelos, os atributos mínimos que são necessários inserir no banco de dados (*MongoDB*) a respeito do usuário.

```
1  # frozen_string_literal: true
2
3  class CustomLead
4    EMAIL_REGEXP = /\A[a-zA-Z0-9._\+-]+@[a-zA-Z0-9._-]+\.[a-zA-Z]{2,}\z/
5
6    include Mongoid::Document
7    include Mongoid::Timestamps
8
9    validates :first_name, :last_name, presence: true
10   validates :email, format: { with: EMAIL_REGEXP }, presence: true,
         uniqueness: true
11
12   field :first_name, type: String
13   field :last_name, type: String
14   field :email, type: String
15   field :interested_in, type: String
16   field :city, type: String
17 end
```

Trecho de código: Classe CustomLead que representa o modelo de cadastro de usuários.

Respeitando a formatação MVC, foi definido o controlador para tal modelo *Custom Lead*, com o intuito de receber as informações de cadastro.

```
1  # frozen_string_literal: true
2
3  class CustomLeadsController < ApplicationController
4    def new
5      @custom_lead = CustomLead.new
6    end
7
8    def create
9      @custom_lead = CustomLead.new(custom_lead_params)
10
11     if @custom_lead.save
12       redirect_to new_custom_lead_path, notice: I18n.t('comum_lead.
             successfully_created')
13     else
14       render :new, status: :unprocessable_entity
```

```
15      end
16    end
17
18    private
19
20    def custom_lead_params
21      params.require(:custom_lead).permit(:first_name, :last_name, :email, :
          interested_in, :city)
22    end
23  end
```

Trecho de código: Controlador resposável pela criação de usuários.

E por fim, foi construída uma tela mínima que permite ser feito o cadastro desta informação a nível do banco de dados.

```
1  <%= form_with(model: custom_lead) do |form| %>
2    <% if custom_lead.errors.any? %>
3      <div id="error_explanation">
4        <% plural = custom_lead.errors.count > 1 %>
5
6        <h2><%= "#{custom_lead.errors.count} #{plural ? 'Erros' : 'Erro'}
             n o #{plural ? 'permitiram' : 'permitiu'}" %> o lead de ser salvo
              </h2>
7
8        <ol>
9          <% custom_lead.errors.each do |error| %>
10           <li><%= error.full_message %></li>
11         <% end %>
12       </ul>
13     </div>
14   <% end %>
15
16   <br>
17
18   <div class="field">
19     <%= form.label :first_name %>
20     <%= form.text_field :first_name %>
21   </div>
22
23   <div class="field">
24     <%= form.label :last_name %>
25     <%= form.text_field :last_name %>
26   </div>
27
28   <div class="field">
```

```
29      <%= form.label :email %>
30      <%= form.text_field :email %>
31    </div>
32
33    <div class="field">
34      <%= form.label :interested_in %>
35      <%= form.select :interested_in, ["Selecionar", "ps", "xbox"] %>
36    </div>
37
38    <div class="field">
39      <%= form.label :city %>
40      <%= form.select :city, ["Selecionar", "Sao Vicente", "Sao Paulo"] %>
41    </div>
42
43    <br>
44
45    <div class="actions">
46      <%= form.submit %>
47    </div>
48  <% end %>
```

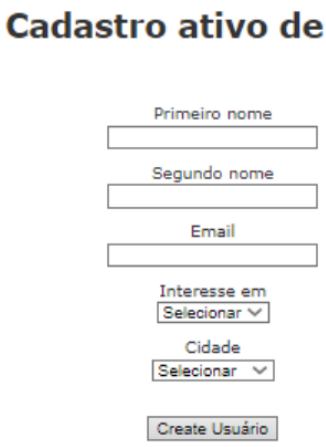
Trecho de código: HTML responsável pela geração do formulário do cadastro.

O que nos resulta em duas possibilidades de inserção das informações que serão contempladas nas subseções a seguir.

### 3.2.2 Cadastro via interface

Neste ponto, é possível realizar o cadastro ativo de um registro via uma interface gráfica:

Figura 3.2: Interface Gráfica Cadastro




Fonte: Autoria pória.

### 3.2.3 Cadastro via *API*

*API* vem do acrônimo *Application Programming Interface*, que traduzido significa Interface de Programação de Aplicação. Neste contexto, a palavra aplicação se refere a qualquer *Software* que exerce uma função distinta. A interface pode ser pensada como um contrato de serviço entre duas aplicações, neste caso se relaciona a etapa que roda em qualquer outro processo diferente da aplicação, podendo ser um outro *Software* por exemplo.

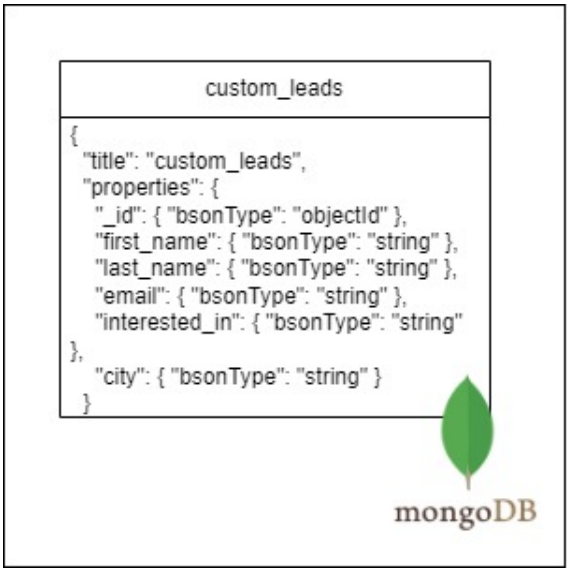
Com isso é necessário utilizar uma ferramenta de transferência de dados de um servidor para outro, como um dos pacotes mais usados na distribuição *Linux*, o *Curl*, que suporta diversos protocolos, assim como o HTTP e o HTTPS, que são os mais comumente utilizados para transmissão de dados via *APIs*.

```
curl --verbose --location --request POST '<APP_PROTOCOL>://<APP_HOST>/
    custom_leads' \
--header 'Content-Type: application/json' \
--data-raw '{
  "custom_lead": {
    "first_name": "Nome de exemplo",
    "last_name": "Sobrenome de exemplo",
    "email": "email@exemplo.com.br",
    "interested_in": "Interesse de exemplo",
    "city": "Cidade de exemplo"
  }
}'
```

Trecho de código: Modelo de requisição com o pacote "curl"para criação de um usuário.

Em ambas as formas, o valor é inserido no *MongoDB* com o seguinte formato na coleção de *custom_leads*:

Figura 3.3: Formato na coleção de *custom_leads*




Fonte: Autoria própria.

### 3.2.4 Criação da aplicação de processamento assíncrono

Como mencionado anteriormente, foi escolhida a abordagem de coletar informações de forma assíncrona para enriquecimento, como se faz necessário escolher o local de coleta, foi decidido coletar informações do *Twitter*, uma rede social e um serviço de microblog que permite aos seus usuários enviar e receber atualizações pessoais, informativas positivamente ou negativamente. O motivo dessa escolha se deve pelo fato da companhia permitir o acesso à informação, via *API* por estudantes e outras empresas, sendo que estas informações possuem um baixo valor estrutural, variando de textos, imagens, vídeos, links. Como o propósito em questão em lidar com uma quantidade razoável de dados de baixa estrutura, encaixando perfeitamente quando comparado ao pilar apresentado anteriormente de variedade do Big Data.

Um outro ponto importante é refletir se o fato da manipulação de dados de uma outra aplicação, sendo mais assertivo, da rede social *Twitter* implica que está sendo quebrado a privacidade do usuário pela rede social, porém quando analisado a política de privacidade, a plataforma deixa explicito que o objetivo maior é divulgar as informações mencionadas de forma mais ampla possível, onde até mesmo pessoas fora da redes possam ser impactadas, em outras palavras, não existe diferenciação para eles um usuário anônimo navegando na plataforma, ou o consumo de recursos via API pública, que foi abordada neste estudo de casa, a única diferenciação é o fato de isso ser realizada automaticamente por um processo em outra aplicação, no caso, o processamento assíncrono a seguir.

Como partida foi criado a camada de comunicação com o *Twitter* através de um serviço, uma classe na linguagem *Ruby*, que é responsável por buscar *tweets*, que são basicamente publicações nesta rede social.

Com o intuito de restringir um pouco o escopo, e ser um tanto quanto assertivo, foram colocadas regras nesta pesquisa para que no futuro, essas informações pudessem ser agrupadas com algum valor. Quanto ao tema, foi escolhido o que estava sendo considerado as duas palavras mais em alta relacionadas a um produto pela plataforma (*Twitter*) no momento do estudo, que eram "*playstation*" e "*xbox*", além de seus derivados, como abreviações e consoles mais recentes, no caso do "*playstation*" tem-se o "*playstation* 5" enquanto no do "*xbox*" tem-se o "*xbox one*". Vale ressaltar que o objetivo em questão é analisar o fluxo de coleta, processamento e armazenamento de informações provenientes de origem externa, não possuindo qualquer relevância o tema em questão.

Tabela 3.1: Palavras Procuradas

| Produto | Palavras Procuradas |
|---|---|
| XBOX | xbox one; xboxone; xbox; #xone; #xboxone; #xbox |
| PLAYSTATION | ps5; playstation 5; playstation5; playstation; #ps5; #playstation5; #playstation |

Com o intuito de reduzir o ruído com reportagens, ato de quando alguém simplesmente postar o conteúdo de outra pessoa em sua página, foram removidos os *retweets*, e filtrados as mensagens com valores sensíveis dados pela própria inteligência artificial da rede social, para isso foram informado os seguintes filtros comuns:

**'filter:safe -filter:retweets'**

Com o viés de reduzir em amostras mais significativas, foram escolhidas publicações de somente dois lugares do Brasil, sendo eles a cidade de Santos e a cidade de São Paulo. A *API* fornecida para viés estudantil, nos permite passar a latitude e a longitude de postagem, com um determinado raio. Esses valores foram escolhidos e podem ser visualizados na Tab. 2. Lembrando que se partiu do centro de cada município.

Tabela 3.2: Configurações

| Cidade | Latitude | Longitude | Raio [km] |
|---|---|---|---|
| Santos | -23.533773 | -46.625290 | 50 |
| São Paulo | -23.961800 | -46.332200 | 20 |

Com isso, um exemplo de chamada para a *API* do *Twitter*, seria (cenário de santos - playstation - 10 últimos registros):

```
1 santos_results = client.search(
2   'ps5 OR "playstation 5" OR playstation5 OR playstation OR #ps5 OR #
      playstation5 OR #playstation filter:safe -filter:retweets',
3   geocode: '-23.9618,-46.3322,20km',
4   result_type: 'recent',
```

```
  lang: 'pt').take(10)
```

Trecho de código: Exemplo de chamda para a API do twitter em ruby.

A Classe completa pode ser observada abaixo:

```
# frozen_string_literal: true

module Twitter
  class Api
    class << self
      def search(item)
        query = case item
                when :playstation5
                  playstation5_query
                when :xbox_one
                  xbox_one_query
                else
                  item
                end

        complete_query = "#{query} #{basic_filters}"

        santos_results = client
                         .search(complete_query, geocode: santos_geocode,
                             result_type: 'recent', lang: portuguese)
                         .take(limit)

        sp_results = client
                     .search(complete_query, geocode: sao_paulo_geocode,
                         result_type: 'recent', lang: portuguese)
                     .take(limit)

        {
          santos: santos_results,
          sp: sp_results
        }
      end

      private

      def basic_filters
        'filter:safe -filter:retweets'
      end

      def playstation5_query
```

```
39          'ps5 OR "playstation 5" OR playstation5 OR playstation OR #ps5 OR #
                playstation5 OR #playstation'
40        end
41
42        def xbox_one_query
43          '"xbox one" OR xboxone OR xbox OR #xone OR #xboxone OR #xbox'
44        end
45
46        def sao_paulo_geocode
47          '-23.533773,-46.625290,50km'
48        end
49
50        def santos_geocode
51          '-23.9618,-46.3322,20km'
52        end
53
54        def portuguese
55          'pt'
56        end
57
58        def limit
59          10
60        end
61
62        def client
63          @client ||= ::Twitter::REST::Client.new do |config|
64            config.consumer_key      = ENV.fetch('TWITTER_CONSUMER_KEY', nil)
65            config.consumer_secret = ENV.fetch('TWITTER_CONSUMER_SECRET', nil
                  )
66          end
67        end
68      end
69    end
70  end
```

Trecho de código: Classe responsável pela comunicação HTTPS com o servidor do twitter.

Como já pode ser encontrada, resta criar o gatilho para que seja possível buscar isso de forma assíncrona no servidor, sem qualquer intervenção humana. Para isso, foi escolhido a biblioteca do *Ruby, Sidekiq,* com ela são definidos os *cron jobs*, que são tarefas agendadas, a biblioteca armazena o padrão de horários no formato *cron*, no *Redis*, e com isso verifica de tempos em tempos se está no momento adequado de invocar o método da classe apontada. Como nosso objetivo seria apenas comprovar o armazenamento dessas informações, além das limitações de CPU, e o espaço em disco fornecido pelas plataformas em nuvem de hospedagem gratuitas serem

baixos, foi decidido inserir as 10 últimas postagens, não repetidas no banco de dados, de ambas cidades sobre ambos assuntos, e para isso foi acordado que este gatilho ocorreria a cada primeiro minuto de hora, independentemente do dia, para isso foi definido o seguinte cron.

```
# Configuration file for cron jobs/workers.

lead_generator_by_twitter:
  cron: "1 * * * *" # execute at 1 minute of every hour, ex: 12:01, 13:01,
      14:01, ...
  class: Lead::Generator::Performer::TwitterWorker
  description: Lead generator based on twitter messages
```

Trecho de código: Arquivo de configuração para os gatilhos do processamento em background.

O *Worker* é a classe responsável por invocar o serviço que realmente executa uma tarefa geralmente denominada *jobs*, que em nosso caso seria o Lead::Generator::Twitter. O final *Worker* no nome da classe seria uma norma de boas práticas para referenciar um processo que ocorre em *background*.

```
# frozen_string_literal: true

module Lead
  module Generator
    class Twitter
      def initialize
        @items = %i[playstation5 xbox_one]
      end

      def perform
        @items.each do |item|
          twitter_api_result = ::Twitter::Api.search(item)

          Rails.logger.info("[#{self.class}] #{item} #{log_message(twitter_
              api_result)}")

          success = failed = 0

          twitter_api_result[:santos].to_a.each do |tweet|
            persist_tweet(tweet, 'Santos') ? success += 1 : failed += 1
          end

          Rails.logger.info("[#{self.class}] Santos, sucessos: #{success},
              falhas: #{failed}")

          success = failed = 0

```

```
26            twitter_api_result[:sp].to_a.each do |tweet|
27              persist_tweet(tweet, 'Sao Paulo') ? success += 1 : failed += 1
28            end
29
30            Rails.logger.info("[#{self.class}] Sao Paulo, sucessos: #{success
                  }, falhas: #{failed}")
31          end
32        end
33
34        private
35
36        def log_message(result)
37          "Resultados para santos -> #{result[:santos].size}, para sp #{
                result[:sp].size}"
38        end
39
40        def persist_tweet(tweet, region)
41          tweet_as_hash = tweet.to_h
42
43          message    = tweet_as_hash.delete(:text)
44          user_data = tweet_as_hash.delete(:user)
45
46          lead = ::PublicationLead.new(provider: 'twitter', region:, message
                :, user_data:)
47
48          lead.save
49        end
50      end
51    end
52  end
```

Trecho de código: Classe responsável por buscar informações dos topicos de playstation e xbox por região.

Com isso, o valor é inserido no *MongoDB* com o seguinte formato na coleção de *publication_leads*:

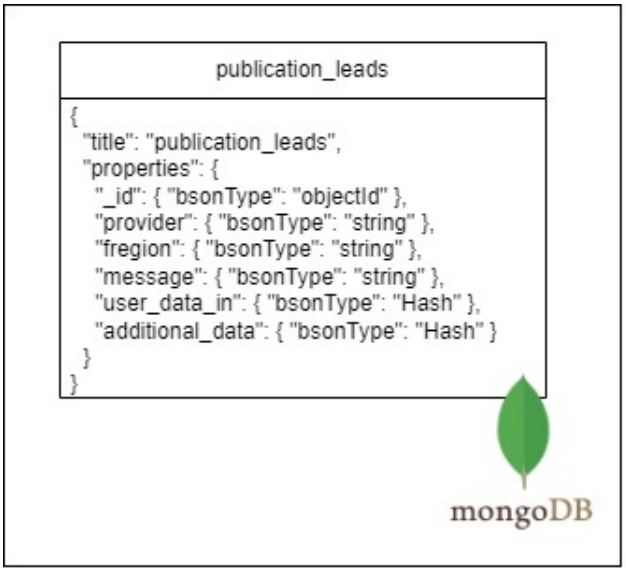
```
1  # frozen_string_literal: true
2
3  class PublicationLead
4    include Mongoid::Document
5    include Mongoid::Timestamps
6
7    validates :provider, :message, presence: true
8
9    field :provider, type: String
```

```
10    field :additional_data, type: Hash
11    field :region, type: String
12    field :message, type: String
13    field :user_data, type: Hash
14  end
```

Trecho de código: Classe PublicationLead que representa o modelo de busca de tweets.

O formato da coleção *publication_leads*, pode ser observado na Fig. 4.

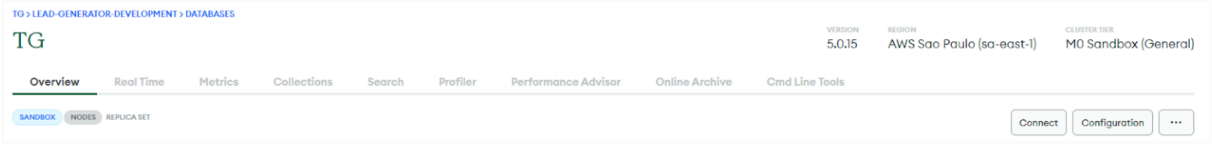
Figura 3.4: Formato na coleção de *publication_leads*




Fonte: Autoria própria.

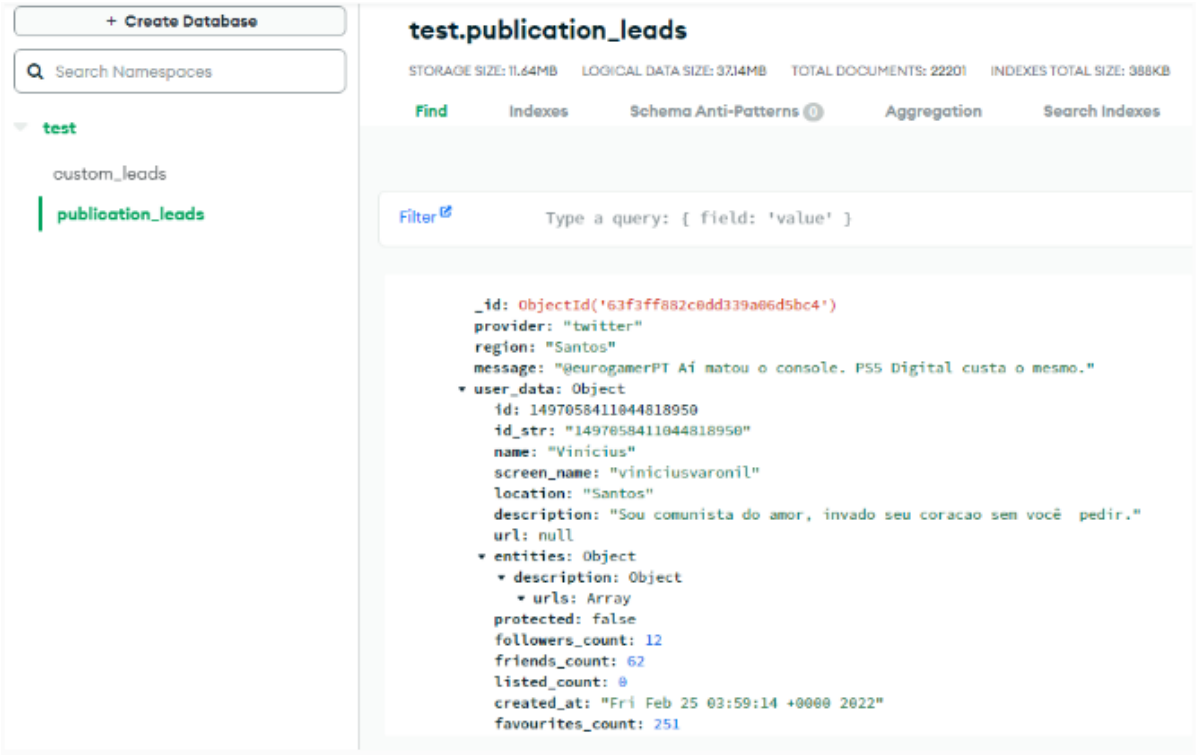
A Seguir, tem-se a evidência de criação no *MongoDB* na Nuvem, utilizando *Mongo Atlas*, um exemplo de documento dentro de *publication_leads* e um exemplo de documento dentro de *custom_leads*, que podem ser visualizados nas Figs. 5, 6 e 7 respectivamente.

Figura 3.5: Criação no MongoDB



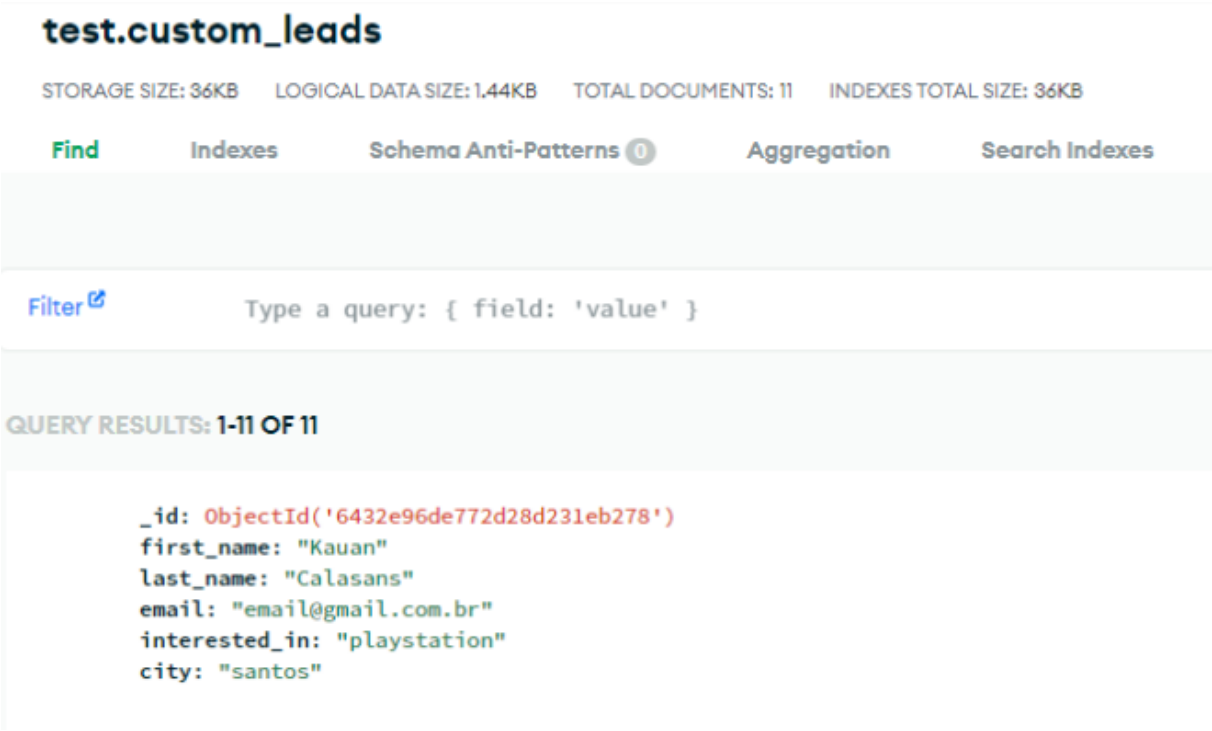

Fonte: Autoria própria.

Figura 3.6: Exemplo de documento dentro de *publication_leads*




Fonte: Autoria própria.

Figura 3.7: Exemplo de documento dentro de *custom_leads*



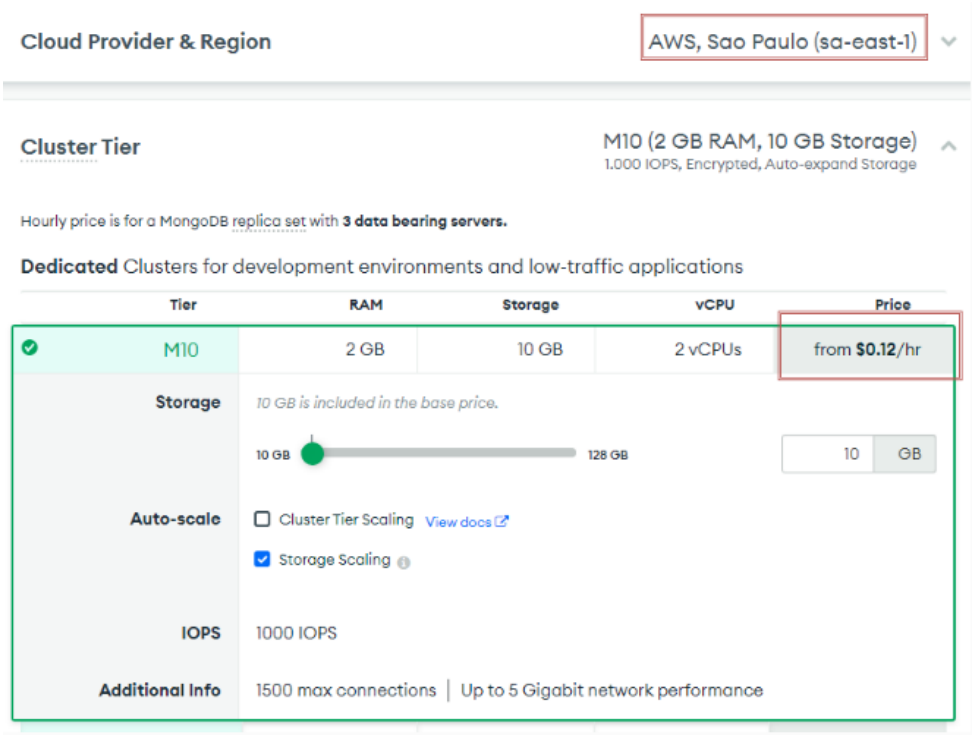

Fonte: Autoria própria.

### 3.2.5 Criação do processo de ETL das informações

Como normalmente as informações obtidas estão contidas em um banco de dados não relacional, e com uma grande variação de formatos, para melhor manipulação e gerenciamento de *dashboards*, faz-se necessário uma manipulação ou transformação da informação. Um outro ponto relevante, que costuma ser frequente em empresas que utilizam plataformas na Nuvem, é utilizar um banco de réplica, para que não onere o que está sendo executado em produção. O mesmo será feito aqui, baseado no que existe no *MongoDB*, será recriado uma outra instância com os mesmo valores, inclusive será mantido o padrão. Essas informações serão manipuladas em um lugar, geograficamente, com o custo menor por armazenamento, visto que geralmente são produzidas uma vez ao dia, e não impacta o negócio.

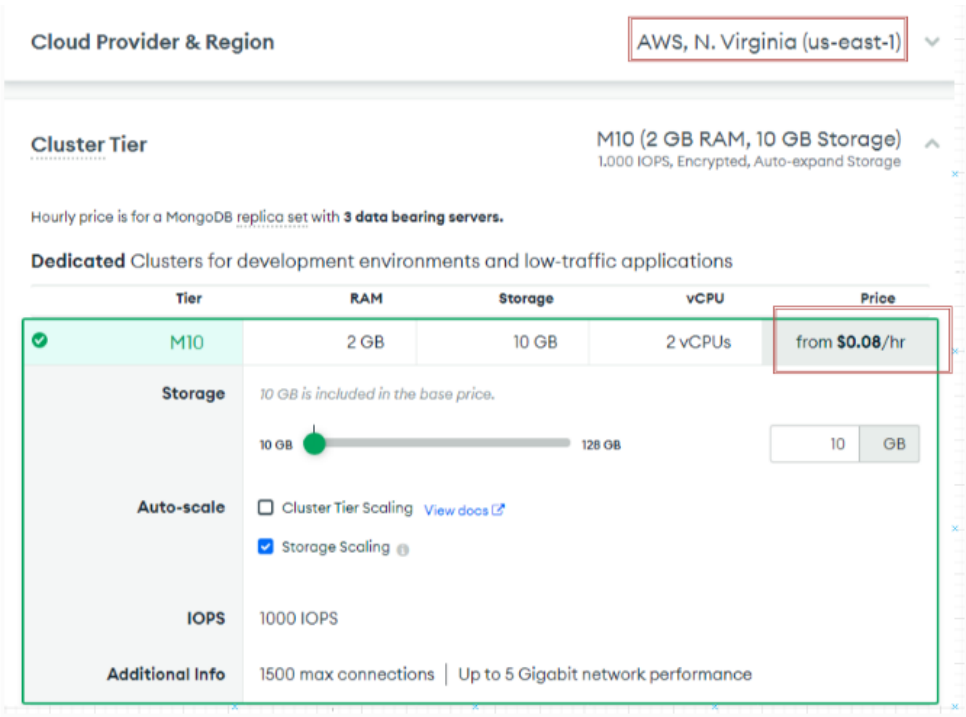
Abaixo, podem ser visualizados os preços de armazenamento na plataforma *MongoDB Atlas* em São Paulo - Brasil e Norte Virgínia - Estados Unidos, nas Figs 8 e 9 respectivamente.

Figura 3.8: Preço de armazenamento na plataforma *MongoDB Atlas* em São Paulo - Brasil




Fonte: Autoria própria.

Figura 3.9: Preço de armazenamento na plataforma *MongoDB Atlas* em Norte Virgínia - Estados Unidos

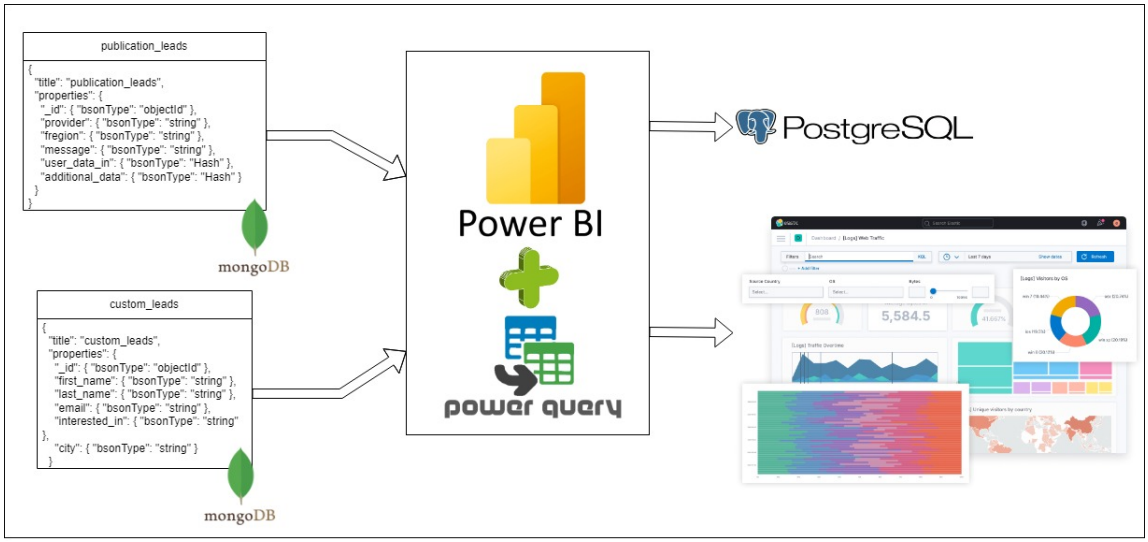



Fonte: Autoria própria.

Isto implica que foram obtidas as informações coletas sincronamente e assincronamente em regiões geográficas distintas. O papel das ferramentas *PowerBI* e *PowerQuery* é gerar tanto o artefato em um banco de dados relacional, como *PostgreSQL*, algo semelhante a um *data warehouse* que é responsável por armazenar dados atuais e históricos de uma empresa, assim, melhora o poder analítico corporativo, além de criar os *dashboards* de negócio.

Por fim o processo de ETL pode ser obeservado na Fig. 10.

Figura 3.10: Processo de ETL




Fonte: Autoria própria.

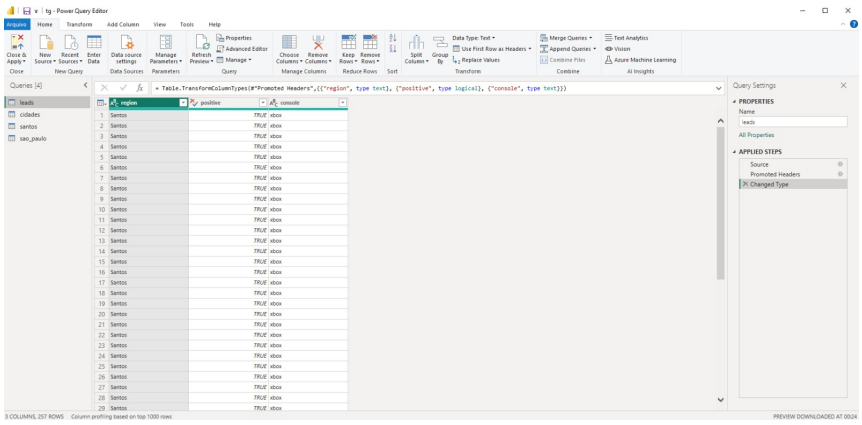
O processo de ETL consistiu em combinar as duas coleções existentes, *PublicationLeads* e *CustomLeads*, para gerar *dashboards*, ferramenta que auxilia na visualização de dados e métricas importantes para uma companhia, que agregasse em uma possível tomada de decisão corporativa, dessa forma, como ilustrado na figura acima, foi gerado alguns, com as ferramentas de análise de dados mencionadas, *PowerBi* em conjunto com o *PowerQuery*, nesta etapa foi realizada o processo de retirada de identidade da informação, pois o que interessa para quem consumirá os relatórios, são as métricas por região.

O primeiro passo consiste em importar ao *PowerBI* as informações, e como os dados participantes da agregação se encontram em um banco de réplica na plataforma MongoDB Atlas, mais especificamente em Illinois, faz-se necessário a presença de um ODBC (*Open Database Connectivity*), que é responsável por traduzir o formato que os registros são inseridos no banco de dados do MongoDB para o formato conhecido pelo *PowerBI*, com isso, foi adicionado o *driver* fornecido em conjunto com a equipe desenvolvedora do MongoDB (MongoDB Inc) juntamente com a equipe desenvolvedora do PowerBI (Microsoft), o *Mongo ODBC Driver*, e com isso se torna possível apontar uma base de dados inserida em um MongoDB como tabela de dados no *PowerBI*, basta inserir as credenciais de conexão presentes no banco de dados de réplica na plataforma MongoDB Atlas e tem-se todas informações inseridas.

O Segundo passo se dá em sanitizar os dados para melhor visualização dos relatórios, retirar informações excedentes, ajustar colunas, e eliminar registros. Nesta etapa, não se fez-se necessário a deduplicação, remoção de registro duplicados, pelo fato da parte do Modelo da aplicação já se encarregar de não persistir no banco de dados registros duplicados, visto que a própria API do Twitter retorna um identificador da mensagem assim, já se sabe se ela foi utilizada. Mas para a construção de um dashboard simples, foram removidos as colunas existentes até que

ficassem com as seguintes: *region* (representa a região de coleta do Tweet, podendo ser Santos ou São Paulo), *positive* (um valor retornado pela API do Twitter que representa se a postagem trata-se de algo positivo/neutro ou negativo) e console (representa a identidade que foi buscada pela aplicação, que no caso em questão pode ser "xbox" ou "playstation"). Todo esse manuseio foi realizado através da ferramenta *PowerQuery*, como evidenciado abaixo, o mesmo processo foi feito para as demais informações presentes na aplicação.

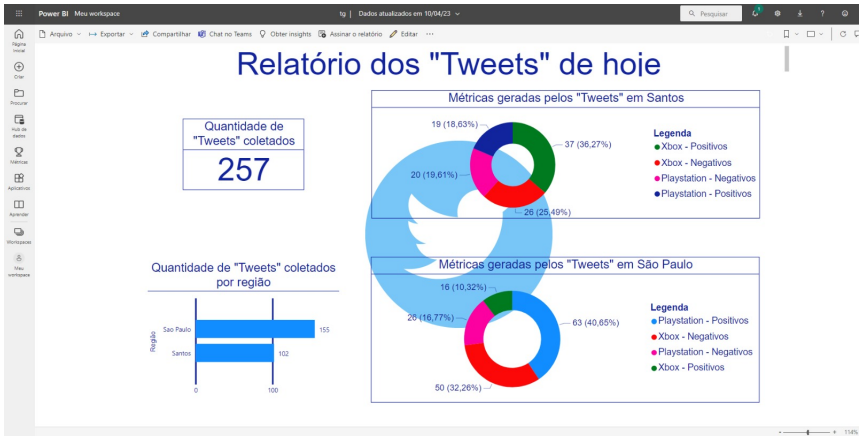
Figura 3.11: Evidência da agregação da informação no *PowerQuery*



Fonte: Autoria própria.

Dessa forma, foram gerados diversas métricas, um dos *dashboards* pode ser evidenciado abaixo com as métricas coletadas no twitter no dia 10 de abril de 2023 na parte da manhã.

Figura 3.12: *Dashboard* gerado com as informações coletadas do Twitter no dia 10 de abril




Fonte: Autoria própria.

O código completo do sistema pode ser visualizado no repositório presente no rodapé da página[1].

[1]https://github.com/KauanCarvalho/lead-generator

# CAPÍTULO 4

# Resultados e Discussão

Utilizando apenas ferramentas e plataformas gratuitas foi possível esboçar um sistema completo que atua de diversas formas, sendo ela proativamente, com a interação do usuário cadastrador, ou reativamente, através dos processos assíncronos que rodam em *background*, e para que isso acontecesse em conformidade com o esperado pela legislação, algumas etapas foram modificadas e estudadas, e com isso os seguintes tópicos foram analisados olhando para o sistema construído:

1. Entendimento da distribuição geográfica dos segmentos do ecossistema.
2. Mapeamento de Dados.
3. Análise de requisitos.
4. Anonimização dos dados.
5. Breve discussão dos casos de vazamentos de dados, apresentados na Introdução do presente trabalho.

## 4.1 Entendimento da distribuição geográfica dos segmentos do ecossistema

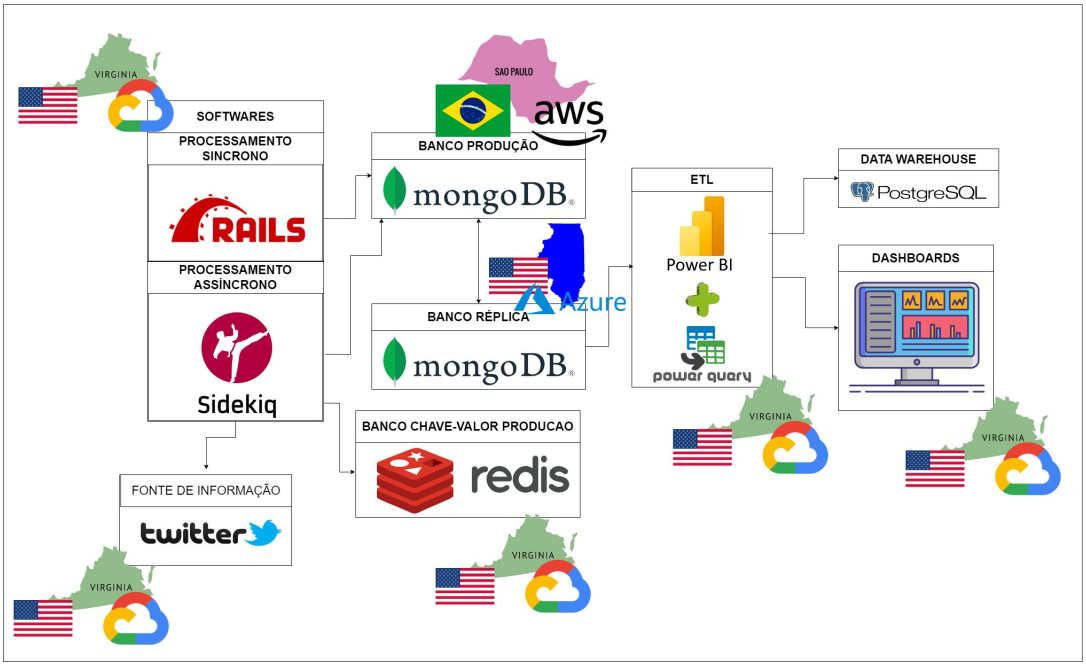
Com o intuito de baratear o projeto, além de forçar a distribuição geográfica dos segmentos da aplicação para análise da Governança de dados perante a LGPD, foi obtido as seguintes regiões por segmento do *software*.

Tabela 4.1: Distribuição geográfica dos segmentos

| Segmento | SAS - Plataforma | Localização |
|---|---|---|
| Banco produção | MongoDB Atlas - AWS | São Paulo |
| Software de cadastro (*frontend e backend*) | Railway - GCP | Virgínia |
| Software de coleta de tweets | Railway - GCP | Virgínia |
| Redis | Railway - GCP | Virgínia |
| Banco de réplica | MongoDB Atlas - Azure | Illinois |
| ETL | Railway - GCP | Virgínia |
| Data warehouse | Railway - GCP | Virgínia |

Para facilitar a visualização da distribuição geográfica do *software*, foi elaborado uma figura que pode ser visualizada abaixo.

Figura 4.1: Distribuição geográfica dos segmentos




Fonte: Autoria própria.

A figura 4.1 nos permite evidenciar a distribuição geográfica dos diversos segmentos que compõem a aplicação, que em seguida será relacionado a governança de dados em relação a uma análise de requisitos, isto nos permite constatar que um dos objetivos do trabalho de distribuição geográfica foi atingido com sucesso, e segundo a LGPD qualquer operação de tratamento de dado de um titular, independentemente do país onde estejam localizado (como no cenário da ETL que acontece em Virgínia com dados coletados no Brasil, e a fonte de dados está salva em Illinois, ou então no cenário de coleta de *Tweets*, onde o *hardware* se encontra em Virgínia, e persiste no banco de dados em São Paulo, dados provenientes do Brasil) devem atender os requisitos expostos na LGPD mesmo estando em um região geográfica diferente do Brasil.

## 4.2 Mapeamento de Dados

Na Fundamentação Teórica, foram apresentadas algumas metodologias que podem ser utilizadas durante o processo de desenvolvimento de *Softwares*, para garantir que estejam alinhados com a LGPD. Nessa seção, abordaremos algumas categorias do Mapeamento de Dados, relacionando-as com o nosso estudo de caso. Para facilitar a visualização, as informações foram disponibilizadas na Tabela 4.2.

Tabela 4.2: Categorias do mapeamento de dados relacionadas ao estudo de caso

| Categoria do Mapeamento de Dados | Exemplo do Estudo de Caso |
|---|---|
| Tipos de dados | Os arquivos de *logs* das aplicações construídas, deveriam ser acessados somente pela equipe de desenvolvimento de *Software*, pois estas informações podem conter valores sensíveis a nível da aplicação. |
| Volume de dados | Métricas geradas pelas aplicações, como latência (tempo de resposta médio da aplicação), números de conexões ativas com o banco de dados, entre outras informações correlatas devem estar de fácil acesso para identificação de possíveis vulnerabilidades. Uma anomalia desses valores deve indicar que algo não está acontecendo como deveria. |
| Etapas do fluxo de dados | Não faria sentido na etapa geração dos *dashboards* levar as informações pessoais de usuários, pois a este ponto, quaisquer dados deveriam ter sofrido o processo de anonimização de informação. |

| | |
|---|---|
| Tecnologias | Um desafio da engenharia de *Software* é o fato de manter as versões mais recentes dos *frameworks* e linguagens utilizadas. A atualização dessas dependências não melhora apenas a performance, mas sim a segurança, um bom exemplo se dá pelo *Ruby on Rails* que recorrentemente ganha atualizações novas que ajustam medidas de seguranças. Ter essas informações documentadas ajuda nas tratativas ligadas a melhoria da aplicação. |
| Locais de armazenamento | O processamento em *background* acontece em uma certa localidade do globo, em uma máquina isolada das restantes. |
| Origem dos dados | Os dados ativos são obtidos via cadastro, enquanto dados de postagens são recolhidos de um serviço externo do *Twitter*. |
| Compartilhamento de dados com parceiros | Um bom cenário é a política de termos de serviço e a política de privacidade, que são estabelecidas ao criar uma conta no *Twitter*. Essas permitem com que *APIs* privadas possam ser disponibilizadas, assim como foi utilizado nesse sistema. |
| Transferência internacional de dados | No sistema construído foram utilizadas diversas plataformas de nuvem, dispostas em diferentes localidades do globo que devem respeitar as demais condições. |

## 4.3 Análise de requisitos

Evidentemente pode existir diferenças legítimas entre os interesses da companhia e dos usuários, e o objetivo principal de ambos os lados deve ser encontrar o ponto de equilíbrio, na qual nenhuma vertente se prejudique, a organização deve ter como objetivo principal oferecer políticas de privacidade consistentes e eficazes, além de escolher as medidas que melhor se adequam ao seu perfil e da sua base de usuários.

Um outro ponto, é que nenhum sistema construído é perfeito a todos os pontos de falhas, por isso o segredo está no quanto a organização tolera em termos de risco de informação, e dado o

cenário, o quanto os usuários estão dispostos a arriscar e com isso se torna mandatório a restrição de acesso a certos tipos de dados baseado em diferentes tipos de perfis.

Um bom exemplo, embora a simplicidade, é analisar os tipos de informações presentes no sistema desenvolvido, onde se deve criar alguns perfis baseados em suas funções, sendo eles no estudo de caso:

- **Desenvolvedores** - Profissional que escreve e cria *softwares*, neste caso se refere a uma aplicação *web* que pode ser divida tanto em *frontend* (tela de formulário) quanto *backend* (servidor que recebe requisição proveniente da tela, e que lida com o processamento em *background*).

- **Analistas de qualidade** - Profissional que implementa e elabora ferramentas, como testes automatizados de todos os tipos, para garantir a integridade e qualidade de um *Software*.

- ***SREs / Devops*** - Responsáveis pelas etapas de infraestrutura, de o monitoramento a sua construção, como por exemplo colocar a aplicação com todos os cenários positivos em produção junto com os desenvolvedores.

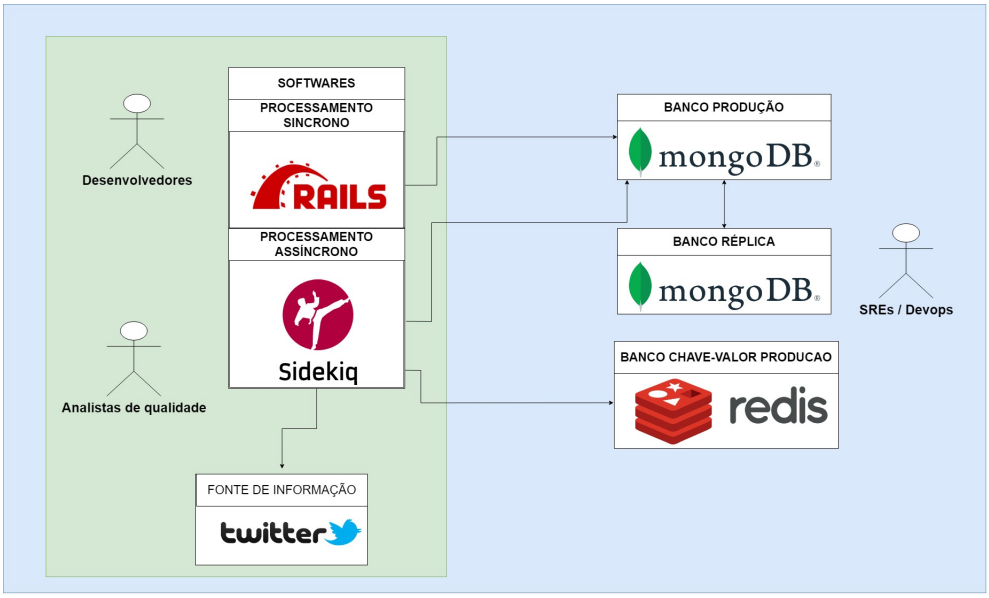
- **Engenheiros de dados** - Responsável por planejar e executar *pipeline* de dados, garantindo que os dados estejam disponíveis para serem usados com segurança, um bom exemplo aqui seria o profissional encarregado pelas ETLs.

- ***Business Intelligence*** **(BI)** - Tem como responsabilidade fazer coleta de dados e informações, de uma base de dados, que permitam identificar problemas ou oportunidades, como a construção de *dashboards* para a área de negócio.

- ***Product Owner*** **(PO)** - Responsável por trabalhar com um gerenciamento de um certo produto em qualquer área, inclusive engenharia de *Software*.

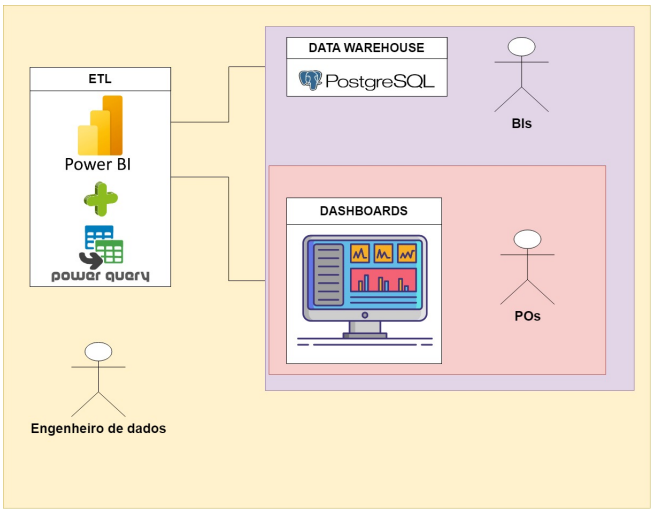
Existem diversas formas de distribuir acesso a certas informações, mas sem dúvidas a mais comum é baseado no exercimento da sua função profissional, pegando o sistema construído como exemplo, pode-se chegar, com as devidas simplificações, em algo relacionado a essa composição de distribuição de acesso aos dados. Na Fig. 1, tem-se o acesso aos dados da equipe de desenvolvimento e infraestrutura enquanto na Fig. 2 o acesso aos dados pelos analistas de dados e de produto.

Figura 4.2: Acesso aos dados pela equipe de desenvolvimento e de infraestrutura




Fonte: Autoria própria.

Figura 4.3: Acesso aos dados pelos analistas de dados e de produto




Fonte: Autoria própria.

Nas figuras acima, seguindo as boas práticas descritas em evitar o fornecimento de informações excedentes para certos grupos de equipes, pode-se restringir o acesso a certas informações baseado em sua responsabilidade como cargo. Sendo assim, os cargos responsáveis pelos acessos relacionados aos *softwares*, seriam tanto os desenvolvedores e analistas de qualidade quanto a equipe de infraestrutura, baseado em nossa experiência durante o desenvolvimento da aplicação esses times precisam de tais acessos, mesmo que de forma moderada, para ajustes no ambiente de produção, monitoramento das aplicações, monitoramentos das integrações (como a integração com o Twitter), acesso aos *logs* (registro de eventos importantes da software como um todo).

Enquanto do lado do banco de dados, independentemente se tratando do banco de produção ou da réplica, é necessário que o time de infraestrutura tenha a capacidade de gerenciamento, desde operações simples como liberar credenciais novas para novas aplicações, criação de novas instâncias, controle das variáveis de ambiente de forma que não fiquem expostas, possibilidade de fazer *resize* da instância do banco de dados (nada mais seria do que dar mais ou menos recursos de hardware para o banco de dados, processo que geralmente é realizado quando a equipe sabe que terá um grande acesso, como datas comemorativas, campanhas, entre outras).

Enquanto para a análise de dados, tem-se a necessidade de que os *PO's* consigam enxergar os *dashboards* construído pela equipe de *BI's*, com perfeita anonimização da informação levada por todo o fluxo, e para isso a equipe de *BI's* deve ter acesso aos dados agrupados pelo time de Engenharia de dados, e a criação de *dashboards*, dessa forma, o grupo de pessoas que lidam com o dado de forma bruta, com baixo valor para negócio e com informações sensíveis, é menor, e com isso estreita-se o funil de vazamento de informações.

Outra abordagem comumente utilizada, seriam ter ambientes diferentes de produção, com isso, reduz-se ainda mais o conhecimento e acesso de pessoas neste ambiente crítico, além de que geralmente, apenas cargos de senioridades mais elevadas possuem acesso a esta informação, e são tratados como cargos de confiança. Evidentemente no desenvolvimento deste projeto, foi necessário a realização de mais de processo por pessoa, uma prática que pode levar a um conhecimento exagerado por um único membro que não é tão comum nos dias atuais.

## 4.4 Anonimização dos dados

Para um dado pessoal, ou seja, uma informação que permite a identificação de uma pessoa seja ela diretamente, ou em conjunto com outras informações, sua anonimização se dá pelo processo de utilização de meios técnicos que sejam razoáveis e disponíveis em qualquer etapa de tratamento na qual um dado perde, propositalmente, a possibilidade de associação direta e indireta a um indivíduo, e assim, se torna impossível sua associação a uma certa informação [3].

A grande principal vantagem é o fato de que informações anonimizadas não entrarem no escopo da LGPD, justamente pelo fato de não poder ser possível a identificação do indivíduo, e se torna uma maneira excelente para lidar com dois cenários:

1. Geração de material estatístico, ou seja, pode-se gerar métricas genéricas a respeito de alguma informação que agregue valor para negócio, sem se comprometer com a LGPD;

2. Encontrar problemas no ambiente de produção, quando se lida com o usuário, é ideal evitar menções diretas a um usuário, desde uma mensagem de log com email, CPF, pode agravar problemas, geralmente são usados outros componentes, sendo até mesmo os identificadores únicos de registro.

## 4.5 Breve discussão dos casos de vazamentos de dados, apresentados na Introdução do presente trabalho.

Analisando o caso *Cambridge Analytica* considerando a LGPD, pode-se notar pelo menos duas violações, são elas: (i) tratamento dos dados para outra finalidade; (ii) tratamento de dados sem o consentimento do titular. A primeira violação aconteceu quando os usuários do aplicativo forneceram seus dados, consentindo que esses seriam utilizados para fins acadêmicos, mas futuramente vieram a ser utilizados para outros fins. Já a segunda violação ocorreu quando os dados dos amigos dos usuários do aplicativo foram capturados sem o consentimento deles.

Não é o objetivo desse trabalho julgar se os dados foram ou não utilizados para influenciar nas eleições de 2016 dos Estados Unidos de América e no *Brexit*, e no caso de terem sido, determinar se foi ou não efetivo. O objetivo aqui é mostrar que houve o vazamento de dados pessoais e que estes estavam sendo utilizados para fins distintos dos propostos inicialmente e em alguns casos sem o consentimento dos seus proprietários.

Já no caso de vazamento de dados de chaves Pix, o Banco Central informou que todas as pessoas que tiveram informações expostas seriam notificadas. Isso está previsto pela LGPD, no Capítulo VIII, Seção I, que trata das sanções a serem aplicadas nos casos de infração da Lei, sendo uma delas, a publicização da infração após devidamente apurada e confirmada.

# CAPÍTULO 5

# Conclusão

Esse trabalho pretendeu explorar os impactos da governança de dados em *Big Data* no cenário atual, trazendo um olhar mais atento para como a LGPD (Lei Geral de Proteção de Dados) tem impactado em projetos de desenvolvimento de *Software*, por exemplo. Para tal, foi desenvolvida uma aplicação que coleta, trata e armazena dados obtidos diretamente por meio do cadastro de usuários e indiretamente por uma busca ativa de engajamento em redes sociais com uma alta volumetria e baixa estruturação da informação, consolidando o termo *Big Data*.

Para que o propósito inicial do trabalho fosse atingido, foram definidos 4 objetivos específicos. O primeiro objetivo específico foi analisar o impacto de se ter dados distribuídos em localizações distintas. Após a construção da aplicação que se encontra distribuída geograficamente por diversas regiões do Brasil e Estados Unidos, verificou-se que a LGPD aplica-se a qualquer operação de tratamento independentemente do país onde estejam localizados os dados, contanto que esses tenham sido coletados em território nacional, cenário englobado pelo estudo de caso. Desta forma, os dados utilizados nesse trabalho, ainda que armazenados ou processados em outras regiões, foram capturados no Brasil, e portanto, o tratamento, e qualquer operação realizada com esses dados deve atender aos requisitos expostos na LGPD. O segundo objetivo específico foi analisar o percurso completo dos dados, desde a coleta até a disponibilização para o negócio, o que foi feito na seção 3 de forma detalhada nas etapas de construção da aplicação hospedada na plataforma *Railway*, e da construção da ETL utilizando *PowerBI* em conjunto com o *PowerQuery* e o devido *Driver* OBDC para MongoDB.

O terceiro objetivo específico do trabalho foi apresentar algumas medidas trativas que podem ser adotadas pelas empresas. Verificou-se que existem princípios adotados pelo mercado como o *Privacy by Design*, que auxiliam na adequação das empresas à LGPD, e que não possuí uma

alta taxa de complexidade de implementação quando junto a uma assertiva análise de requisitos, assim como no estudo de caso. E por fim, mas não menos importante, o quarto objetivo específico foi divulgar e compartilhar boas práticas no tocante a captura, tratamento, armazenamento e análise de dados e difundir o conhecimento sobre a legislação vigente no Brasil, no caso, a LGPD. Esse objetivo também foi concluído, uma vez que foram apresentadas boas práticas, como por exemplo a anonimização dos dados que aconteceu nos processos de construção da ETL, onde foi rejeitado qualquer identificação do dado que poderia levar ao titular novamente, e foram abordados os assuntos principais tratados em cada um dos 10 capítulos da LGPD em nossa fundamentação teórica.

No corpo do trabalho também está presente 3 casos de vazamento de dados que corroboram para a justificativa desse trabalho. Com o atual aumento de volume de dados, e estes são usados para os mais diversos fins, faz-se mais do que necessário um olhar cuidadoso aos processos que os utilizam, principalmente dados pessoais, para evitar, ou pelo menos minimizar o número de incidentes como vazamentos, sequestros e perdas de informações pessoais, que se utilizadas de forma errada, podem trazer prejuízos altíssimos para seus titulares.

Embora sancionada em 14 de agosto de 2018, a LGPD só entrou em vigor em setembro de 2020, dessa forma, pode ser considerada uma lei bastante recente. Muitas empresas ainda estão alinhando seus processos para que esses estejam em conformidade com a LGPD, e essa, com certeza não é uma tarefa fácil. No primeiro trimestre de 2022, o Brasil ficou em 12º lugar entre os países que mais contabilizaram episódios de vazamento de dados, segundo a *SurfShark*, uma empresa especializada em privacidade [31].

Com isso, concluímos que o caminho a ser trilhado ainda é bastante longo e iniciativas como a desse trabalho se fazem necessárias para compartilhar boas práticas com relação à proteção de dados, principalmente dados sensíveis. Assim, ao longo do desenvolvimento da presente monografia foi possível vislumbrar algumas vertentes para continuidade do trabalho, que incluem:

- Aumentar o número de fontes de informações, utilizando até mesmo fontes com outras estruturas de dados, como arquivos de mídia de outras plataformas.
- Expandir a aplicação da LGPD entre os subgrupos dos domínios dos dados, baseando-se em uma nova análise de requisitos, onde existe a presença de uma equipe com maior quantidade de ramificações de especializações, enfatizando a área de Segurança da Informação.
- Expandir como a LGPD se aplica aos processos de infraestrutura, como nos conceitos de disponibilização da aplicação no ambiente de produção.

De qualquer forma, alinhar o processo de desenvolvimento de *Software* com a LGPD pode ser na maioria das vezes um processo demorado, mas necessário primeiramente para garantir

a privacidade e a segurança dos titulares dos dados e depois para evitar que a empresa sofra sanções, que sim, podem ser mais brandas, como advertências, mas também podem ser mais rigorosas como multas diárias, eliminação dos dados e até suspensão parcial do banco de dados.

# Bibliografia


1 COMUNICAÇÕES, M. das. *Aumenta para 90% o número de domicílios com internet no Brasil*. 2022. Disponível em: ⟨https://www.gov.br/mcom/pt-br/noticias/2022/setembro/aumenta-o-numero-de-domicilios-com-internet-no-brasil⟩. Acesso em: 08 de jan. de 2023.

2 REINSEL, D.; GANTZ, J.; RYDNING, J. *The Digitization of the WorldFrom Edge to Core*. 2018. Disponível em: ⟨chrome-extension://oemmndcbldboiebfnladdacbdfmadadm/https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf⟩. Acesso em: 15 de dez de 2022.

3 BRASIL. Lei nº 13.709, de 14 de agosto de 2018. Brasília, DF, 2018. Disponível em: ⟨https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm⟩.

4 CERVANTES, V.; RODRIGUES, D. F. Big data e proteção de dados: O desafio está lançado.

5 GUENKA, T. T. Big data: vigiar e rastrear o caso da cambridge analytica. Pontifícia Universidade Católica de São Paulo, 2019.

6 FORNASIER, M. de O.; BECK, C. Cambridge analytica: escândalo, legado e possíveis futuros para a democracia. *Revista Direito em Debate*, v. 29, n. 53, p. 182–195, 2020.

7 BBC. *Entenda o escândalo de uso político de dados que derrubou valor do Facebook e o colocou na mira de autoridades*. Disponível em: ⟨https://www.bbc.com/portuguese/internacional-43461751⟩. Acesso em: 14 de fevereiro de 2023.

8 NGUYEN, N. *Here's How Facebook Got Into This Mess: A Timeline*. Disponível em: ⟨https://www.buzzfeednews.com/article/nicolenguyen/cambridge-analytica-facebook-timeline⟩. Acesso em: 10 de março de 2023.

9 BBC. *Reality Check: Was Facebook data's value 'literally nothing'?* Disponível em: ⟨https://www.bbc.com/news/technology-43502366⟩. Acesso em: 12 de março de 2023.

10 SUMPTER, D. *Outnumbered: From Facebook and Google to Fake News and Filter-Bubbles - The Algorithms That Control Our Lives*. [S.l.]: Bloomsbury SIGMA, 2018. Anotação. ISBN 147294741X.

11 GOV. *Polícia Federal deflagra a Operação Deepwater que combate a obtenção e vazamento ilegal de dados pessoais de brasileiros pela internet*. Disponível em: ⟨https://l1nk.dev/qvpSD⟩. Acesso em: 16 de fevereiro de 2023.

12 AGÊNCIABRASIL. *PF prende hacker suspeito do maior vazamento de dados do Brasil*. Disponível em: ⟨https://agenciabrasil.ebc.com.br/geral/noticia/2021-03/pf-prende-hacker-suspeito-do-maior-vazamento-de-dados-no-brasil⟩. Acesso em: 10 de fevereiro de 2023.

13 ENOMURA, B. Y. Big data: A era dos grandes dados já chegou. 2014. 22fs. *Trabalho de Conclusão de Curso (Curso de Jornalismo)–Departamento de Comunicação e Expressão. Universidade Federal de Santa Catarina*, 2014.

14 MAHESHWARI, A. *Big Data Made Accessible*. [S.l.: s.n.], 2016. 2020 edition Kindle Edition.

15 LÚCIO, R. *Big Data*. 2021. Disponível em: ⟨https://energiainteligenteufjf.com.br/tecnologia/big-data/⟩. Acesso em: 10 de dezembro de 2022.

16 SILVA, V. M. da. *Big Data: Definição e Um Breve Histórico*. 2019. Disponível em: ⟨https://encurtador.com.br/qsyT0⟩. Acesso em: 15 de janeiro de 2022.

17 GALDINO, N. Big data: ferramentas e aplicabilidade. In: *Congresso De Engenharia*. [S.l.: s.n.], 2016.

18 VILLELA, A. O fenômeno 'big data'e seu impacto nos negócios. *https://canaltech. com. br/bigdata/O-fenomeno-Big-Data-e-seu-impacto-nos-negocios/¿. Acesso em 13 de janeiro de 2023*, v. 5, p. 12, 2018.

19 SERASA. *Entenda o que é risco de crédito e por que ele deve ser calculado*. Disponível em: ⟨https://www.serasa.com.br/ecred/blog/entenda-o-que-e-risco-de-credito-e-por-que-ele-deve-ser-calculado/⟩. Acesso em: 10 de fevereiro de 2023.

20 ALGARTECH. *Governança de dados: como criar uma estratégia efetiva de big data*. Disponível em: ⟨https://algartech.com/pt/blog/governanca-de-dados-como-criar-uma-estrategia-efetiva-de-big-data/⟩. Acesso em: 15 de abril de 2023.

21 ENAP. *Governança de Dados - Módulo 3: Gestão Inteligente de Dados*. 2019. Disponível em: ⟨https://encurtador.com.br/EKSY8⟩. Acesso em: 25 de janeiro de 2022.

22 BRASIL. Constituição da república federativa do brasil de 1988. 1988. Disponível em: ⟨https://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm⟩.

23 BRASIL. Lei nº 12.965, de 23 de abril de 2014. Brasília, DF, 2014. Disponível em: ⟨http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2014/lei/l12965.htm⟩.

24 PINHEIRO, P. P. *Proteção de dados pessoais : comentários à Lei n.13.709/2018 (LGPD)*. São Paulo: Saraiva Educação, 2018. ISBN 9788553608317.

25 LORENZON, L. N. Análise comparada entre regulamentações de dados pessoais no brasil e na união europeia (lgpd e gdpr) e seus respectivos instrumentos de enforcement. *Revista do Programa de Direito da União Europeia*, v. 1, p. 39–52, 2021.

26 AMAZON. *O que é ETL?* Disponível em: ⟨https://aws.amazon.com/pt/what-is/etl/⟩. Acesso em: 12 de fevereiro de 2023.

27 BECK, K. *Test Driven Development: By Example*. 1ª edição. ed. [S.l.]: Addison-Wesley Professional, 2002. ISBN 9780321146533.

28 RUBY. *Sobre o Ruby*. Disponível em: ⟨https://www.ruby-lang.org/pt/about/⟩. Acesso em: 15 de fevereiro de 2023.

29 HARTL, M. *Ruby on Rails Tutorial: Learn Web Development with Rails*. 6ª edição. ed. [S.l.]: Addison-Wesley Professional, 2020. Anotação. ISBN 0136702651.

30 MONGODB. *MongoDB Documentation*. Disponível em: ⟨https://www.mongodb.com/docs/⟩. Acesso em: 03 de março de 2023.

31 EMBRATEL. *Brasil ocupa 12º lugar no ranking de vazamento de dados*. Disponível em: ⟨https://proximonivel.embratel.com.br/brasil-ocupa-12o-lugar-no-ranking-de-vazamento-de-dados/⟩. Acesso em: 06 de abril de 2023.