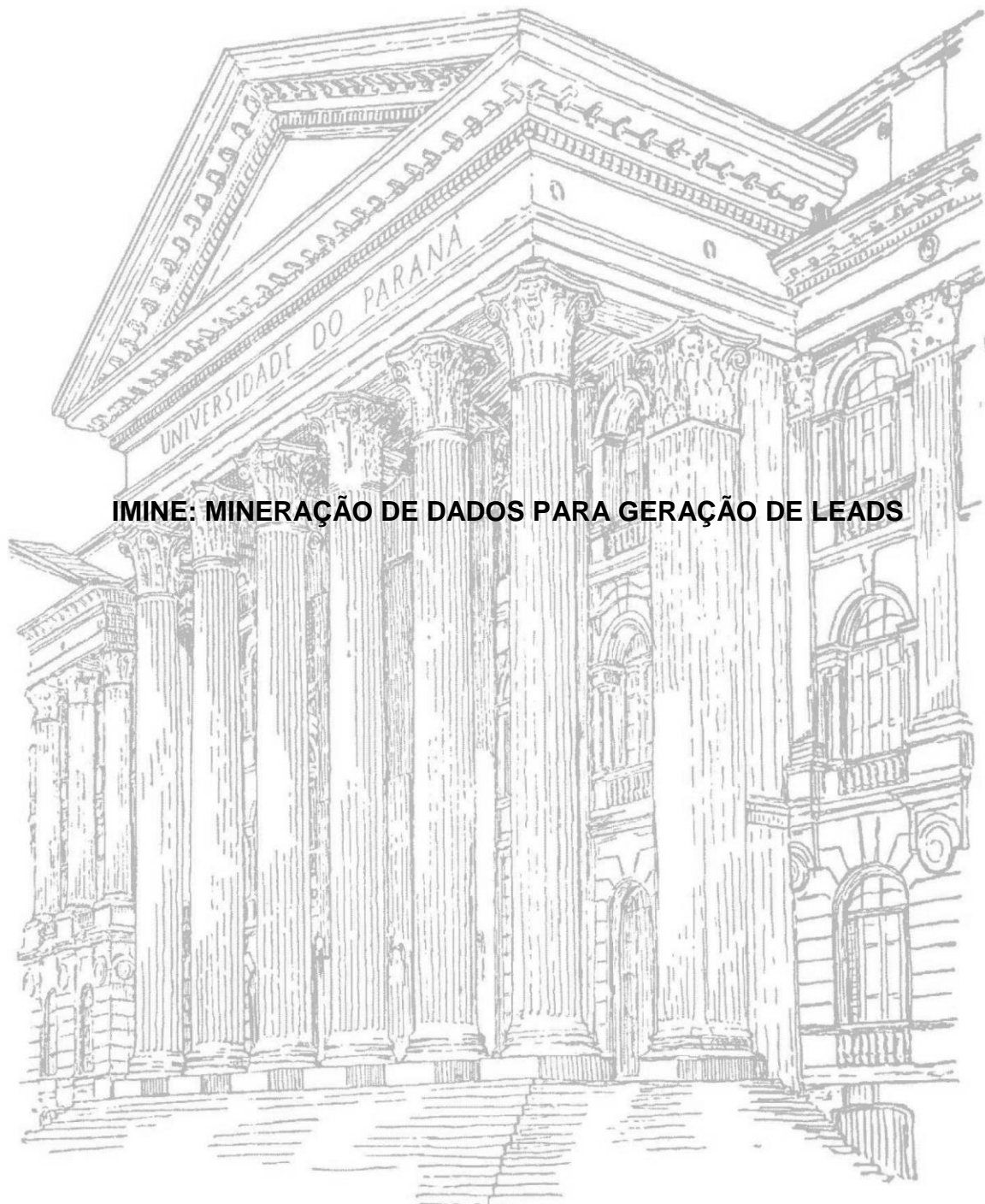


UNIVERSIDADE FEDERAL DO PARANÁ

NICOLE ANATER

PAULO ROBERTO DE QUEIROZ TELLES FILHO

RAISSA DA SILVA MORGADO



**IMINE: MINERAÇÃO DE DADOS PARA GERAÇÃO DE LEADS**

CURITIBA

2017

NICOLE ANATER  
PAULO ROBERTO DE QUEIROZ TELLES FILHO  
RAISSA DA SILVA MORGADO

## **IMINE: MINERAÇÃO DE DADOS PARA GERAÇÃO DE LEADS**

Trabalho de Conclusão de Curso apresentado ao Curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Federal do Paraná como requisito à obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientadora: Prof.<sup>a</sup> Dra. Rafaela Mantovani Fontana

CURITIBA  
2017



MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DO PARANÁ  
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO  
DE SISTEMAS

**TERMO DE APROVAÇÃO**

**PAULO TELLES FILHO**

**RAISSA MORGADO**

**NICOLE ANATER**

**iMine**

Trabalho apresentado como requisito  
Parcial para obtenção do Título de  
Tecnólogo em Análise e  
Desenvolvimento de Sistemas da  
Universidade Federal do Paraná

Curitiba, 29 de Junho de 2017.

**BANCA EXAMINADORA**

---

Professora Rafaela Mantovani Fontana

---

Professor Jaime Wojciechowski

---

Professor Dieval Guizelini

---

Aos nossos pais, familiares e amigos, que foram grandes incentivadores e que sempre acreditaram nos nossos sonhos.

## **AGRADECIMENTOS**

Agradecemos a nossa orientadora, Professora Dra. Rafaela Mantovani Fontana, pelo acompanhamento, orientação e amizade. Agradecemos também ao e Professor Dr. Jaime Wojciechowski pelo auxílio prestado. Agradecemos também a todos os professores envolvidos em nosso processo de formação profissional, e a todos os funcionários nessa trajetória.

Ao Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, do Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná, pelo apoio recebido.

Agradecemos à banca avaliadora por aceitar qualificar o presente trabalho.

Agradecemos também à minestore pela confiança depositada na equipe no desenvolvimento do produto a que este documento faz referência.

Agradecemos aos nossos familiares e amigos por todo o apoio, paciência, palavras de motivação e incentivo que foram essenciais para que continuássemos focados na execução deste trabalho.

## RESUMO

Este trabalho baseia-se e uma necessidade da empresa minestore em analisar uma saturada base de dados e visualizar métricas de seus clientes. Atualmente, a empresa possui uma grande base de dados de lojistas, mas poucos trazem retorno financeiro significativo. Visando a extração de conhecimento desses dados, o objetivo é desenvolver, utilizando metodologia de desenvolvimento ágil, uma aplicação personalizada de apoio a decisão (SAD), para o auxílio nas tomadas de decisões de sucesso ao negócio por meio de dados corporativos. Para isso, a aplicação foi desenvolvida contendo funcionalidades de otimização de indicadores gerais de clientes, *dashboards* e análise de potenciais futuros clientes para a empresa. As tecnologias utilizadas para construir essa aplicação foram PHP para o *back-end*, Bootstrap para *front-end*, Java, Nodejs, WEKA para a mineração de dados, e MySQL para o banco de dados. O sistema proporciona uma visualização que otimiza a interpretação do progresso de métricas e, por meio da mineração de dados, são encontrados potenciais clientes com perfil similar aos que trazem um melhor faturamento para a minestore. A esse sistema foi dado o nome iMine.

Palavras-chave: Dados corporativos, Conhecimento, Sistema de gerenciamento, iMine, Mineração de Dados, futuros clientes.

## **ABSTRACT**

This work is based on a necessity of the company minestore in analyzing a saturated database and visualizing metrics of its clients. Currently the company has a large database of storekeepers, but only a few bring significant financial returns. Aiming at discovering knowledge from these data, the objective of this work was to develop a personalized decision support system (DSS) application for the minestore platform, using agile development methodology, aiding in making successful business decisions with corporate data. For this, the application should contain functionalities as optimization of general customer indicators, dashboards and analysis of potential future customers for the company, for example. The chosen technologies to build up this application were PHP for the back-end, Bootstrap for front-end, Java, Nodejs and WEKA for data mining, and MySQL for the database. The system provides a view that optimizes the interpretation of metric progress and through the process of data mining, the discover of potential customers with similar profile to those who bring the best revenues for the company. This application was given the name iMine.

Keywords: Business data, Knowledge, Management system, iMine, Data Mining, Future customers.

## LISTA DE FIGURAS

FIGURA 1 - FUNIL GENÉRICO .....	22
FIGURA 2 - AS SETE ETAPAS DE UM FUNIL DE VENDAS .....	24
FIGURA 3 - O FUNIL DE VENDAS DA MINESTORE.....	25
FIGURA 4 - FASES DO MODELO .....	26
FIGURA 5 - DIFERENTES FONTES DE DADOS DA MINESTORE.....	29
FIGURA 6 - ESQUEMA REPRESENTATIVO DO PROCESSO <i>SCRUM</i> .....	31
FIGURA 7 - EXEMPLO DE UTILIZAÇÃO DA APLICAÇÃO TRELLO .....	32
FIGURA 8 - PROCESSO DE CRIAÇÃO DE NOVA LOJA NA MINESTORE .....	35
FIGURA 9 - INDICADORES GERAIS DA MINESTORE .....	36
FIGURA 10 - DIAGRAMAS DO UML .....	38
FIGURA 11 - OPÇÕES DE TECNOLOGIAS.....	42
FIGURA 12 - ARQUITETURA DO IMINE .....	48
FIGURA 13 - AQUITETURA DO GATEWAY .....	50
FIGURA 14 - FORMATO DO JSON CONSUMIDO PELO GATEWAY .....	51
FIGURA 15 - ARQUITETURA <i>DATA MINING</i> .....	53
FIGURA 16 - COMPARAÇÃO <i>LEAD A LEAD</i> .....	56
FIGURA 17 - TELA DE INDICADORES.....	57
FIGURA 18 - TELA DE GRÁFICO DE LINHA DO INDICADOR.....	58
FIGURA 19 - TELA DE PESQUISA DE INDICADOR .....	58
FIGURA 20 - TELA DE CATEGORIA DE INDICADOR.....	59
FIGURA 21 - TELA DE IMPORTAÇÃO DE DADOS .....	59
FIGURA 22 - TELA DE <i>DASHBOARD</i> .....	60
FIGURA 23 - GRÁFICO DE NÚMERO DE PEDIDOS .....	60
FIGURA 24 - GRÁFICO DE NÚMERO DE VISITAS NAS LOJAS .....	61
FIGURA 25 - GRÁFICO DE NÚMERO DE PEDIDOS EMITIDOS .....	61
FIGURA 26 - GRÁFICO DE NÚMERO DE LOJAS ATIVAS .....	62
FIGURA 27 - TELA DE PROMESSAS .....	62
FIGURA 28 - TELA DE PROGRESSO DA MINERAÇÃO .....	63
FIGURA 29 - TELA DE VISUALIZAÇÃO DE <i>LEADS</i> .....	63
FIGURA 30 - DIAGRAMA DE CASOS DE USO RELACIONADO AO USUÁRIO .....	71
FIGURA 31 - DIAGRAMA DE CLASSES .....	77

FIGURA 32 - DIAGRAMA DE SEQUÊNCIA VISUALIZAR DADOS AUTOMATIZADO .....	78
FIGURA 33 - DIAGRAMA DE SEQUÊNCIA VISUALIZAR DASHBOARD .....	80
FIGURA 34 - DIAGRAMA DE SEQUÊNCIA CARREGAR ARQUIVOS PARA MINERAÇÃO.....	81
FIGURA 35 - DIAGRAMA DE SEQUÊNCIA REALIZAR DESCOBERTA DE LEADS .....	83
FIGURA 36 - DIAGRAMA DE SEQUÊNCIA VISUALIZAR FUTUROS CLIENTES ...	85
FIGURA 37 - MODELAGEM DO BANCO DE DADOS.....	91

## LISTA DE TABELAS

TABELA 1 - CRONOGRAMA DE ATIVIDADES.....	33
TABELA 2 - TECNOLOGIAS DO IMINE .....	42
TABELA 3 - DEPENDÊNCIAS DE PACOTES .....	44
TABELA 4 - MÉTODOS DISPONÍVEIS NO GATEWAY .....	52
TABELA 5 - DOCUMENTAÇÃO UC VISUALIZAR DADOS AUTOMATIZADOS .....	72
TABELA 6 - DOCUMENTAÇÃO UC VISUALIZAR DASHBOARD .....	73
TABELA 7 - DOCUMENTAÇÃO UC CARREGAR ARQUIVOS DA MINERAÇÃO....	74
TABELA 8 - DOCUMENTAÇÃO UC REALIZAR DESCOBERTA DE CLIENTES .....	75
TABELA 9 - DOCUMENTAÇÃO UC VISUALIZAR FUTUROS CLIENTES .....	76
TABELA 10 - LISTA DE ATRIBUTOS PARA A MINERAÇÃO DE DADOS.....	86
TABELA 11 - DICIONÁRIO DE DADOS .....	88

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>13</b>
1.1.	OBJETIVO GERAL.....	15
1.2.	OBJETIVOS ESPECÍFICOS .....	16
1.3.	PROBLEMA.....	16
1.4.	JUSTIFICATIVA.....	16
1.5.	ESTRUTURA DO DOCUMENTO .....	17
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>19</b>
2.1.	E-COMMERCE NO MERCADO .....	19
2.2.	CRM COM E-COMMERCE .....	20
2.3.	FUNIL DE VENDAS.....	21
2.4.	FUNIL DE VENDAS DA MINESTORE .....	24
2.5.	PREDIÇÃO DE <i>PROSPECTS</i> .....	26
2.6.	PLANILHA DE MÉTRICAS NA MINESTORE.....	29
<b>3</b>	<b>MATERIAIS E MÉTODOS .....</b>	<b>30</b>
3.1.	DESENVOLVIMENTO ÁGIL.....	30
3.2.	ADAPTAÇÃO DA METODOLOGIA ÀS NECESSIDADES DA EQUIPE .....	31
3.3.	CRONOGRAMA DE ATIVIDADES E RESPONSABILIDADES .....	32
3.3.1	Sprint 1 .....	34
3.3.2	Sprint 2 .....	35
3.3.3	Sprint 3 .....	36
3.3.3.1	Histórias de Usuário .....	37
3.3.3.2	Diagrama de Caso de Uso .....	38
3.3.3.3	Diagrama de Classes .....	39
3.3.3.4	Diagrama de Sequência .....	39
3.3.4	Sprint 4 .....	39
3.3.5	Sprint 5 .....	40
3.3.5.1	Tecnologias utilizadas .....	41
3.3.6	Sprint 6 .....	44
3.3.7	Sprint 7 .....	46
3.3.8	Sprint 8 .....	46
3.3.9	Sprint 9 .....	46
<b>4</b>	<b>APRESENTAÇÃO DO SISTEMA .....</b>	<b>48</b>

4.1.	ARQUITETURA DO SISTEMA.....	48
4.1.1	Gateway .....	49
4.1.2	Mineração de dados .....	53
4.2.	APRESENTAÇÃO DO IMINE.....	57
<b>5</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>65</b>
	<b>REFERÊNCIAS.....</b>	<b>67</b>
	<b>APÊNDICE A – HISTÓRIAS DE USUÁRIO .....</b>	<b>70</b>
	<b>APÊNDICE B – DIAGRAMA DE CASOS DE USO .....</b>	<b>71</b>
	<b>APÊNDICE C – DOCUMENTAÇÃO UC VISUALIZAR DADOS AUTOMATIZADOS</b> <b>72</b>	
	<b>APÊNDICE D – DOCUMENTAÇÃO UC VISUALIZAR <i>DASHBOARD</i> .....</b>	<b>73</b>
	<b>APÊNDICE E – DOCUMENTAÇÃO UC CARREGAR ARQUIVOS PARA A</b> <b>MINERAÇÃO .....</b>	<b>74</b>
	<b>APÊNDICE F – DOCUMENTAÇÃO UC REALIZAR DESCOBERTA DE CLIENTES</b> <b>75</b>	
	<b>APÊNDICE G – DOCUMENTAÇÃO UC VISUALIZAR FUTUROS CLIENTES .....</b>	<b>76</b>
	<b>APÊNDICE H – DIAGRAMA DE CLASSES.....</b>	<b>77</b>
	<b>APÊNDICE I – DIAGRAMA DE SEQUÊNCIA VISUALIZAR DADOS</b> <b>AUTOMATIZADOS.....</b>	<b>78</b>
	<b>APÊNDICE J – DIAGRAMA DE SEQUÊNCIA VISUALIZAR <i>DASHBOARD</i> .....</b>	<b>80</b>
	<b>APÊNDICE K – DIAGRAMA DE SEQUÊNCIA CARREGAR ARQUIVOS PARA</b> <b>MINERAÇÃO .....</b>	<b>81</b>
	<b>APÊNDICE L – DIAGRAMA DE SEQUÊNCIA REALIZAR DESCOBERTA DE</b> <b>LEADS 83</b>	
	<b>APÊNDICE M – DIAGRAMA DE SEQUÊNCIA VISUALIZAR FUTUROS CLIENTES</b> <b>85</b>	
	<b>APÊNDICE N – LISTA DE ATRIBUTOS PARA MINERAÇÃO DE DADOS .....</b>	<b>86</b>
	<b>APÊNDICE O – DICIONÁRIO DE DADOS.....</b>	<b>88</b>
	<b>APÊNDICE P – MODELAGEM DO BANCO .....</b>	<b>91</b>

## 1 INTRODUÇÃO

O comércio eletrônico, mais conhecido pelo termo em inglês *e-commerce*, encontra-se presente desde o início da Internet. Na verdade, comprar sem sair de casa, com o auxílio de um aparelho eletrônico, já era realidade desde 1984. Tudo isso graças a Ross Davies, um professor da Universidade de Newcastle, que decidiu fazer um experimento social para estudar as necessidades do consumidor inglês. Sua ideia propunha a utilização de uma televisão modificada conectada a uma linha telefônica, ligando consumidores idosos à rede de supermercados Tesco (FELDMAN, 2011).

Mesmo antes da Internet, o interesse por sistemas que traziam informação para as pessoas (e não o contrário) já era muito grande e, ao longo do tempo, direcionou-se cada vez mais: primeiro para as residências e negócios, por meio de televisores modificados e computadores, depois diretamente para cada usuário, por meio dos *tablets* e *smartphones*. De certa forma, este interesse pela informação foi um dos pilares para a evolução da tecnologia. O comércio e consumo sempre foram nichos muito explorados neste meio (FELDMAN, 2011).

Nas últimas décadas, o comércio online cresceu e continua crescendo a cada ano. No Brasil, mais da metade da população - cerca de 52% em julho de 2015 - possuía acesso à Internet, principal meio de acesso às lojas virtuais. Aproximadamente 38% dos brasileiros faz compras pela Internet todos os meses. Apenas 34 dos 1000 consumidores brasileiros entrevistados em pesquisa sobre a indústria de varejo realizada em 2016 pela PwC (2016), nunca fizeram uma compra online.

Existem diversas razões para os *e-commerces* terem se popularizado de forma tão avassaladora e veloz. É prático e fácil fazer uma compra online, pois a conveniência está em todos os aspectos: em nenhum outro lugar é possível fazer compras no meio da noite, usando pijamas, tendo privacidade tanto em relação aos produtos comprados quanto aos dados de pagamento, que trafegam de modo seguro utilizando padrões de segurança como *Secure Socket Layer (SSL)* (SALVADOR, 2013).

Diferente do comércio comum, na Internet não existem filas ou vendedores ocupados com outros clientes. Mesmo que o próprio consumidor seja quem está “trabalhando”, é possível fazer uma compra em minutos, poupando mais tempo. Além disso, economia, comodidade e agilidade são fatores decisivos na hora de se agradar o cliente, por isso a precificação de artigos comercializados online é um grande diferencial, pois existem menos entidades envolvidas na transição da cadeia, desde o fabricante até a sua casa (BERTHOLDO, 2017).

Existem diversas estratégias de retenção de clientes, que oferecem desde cupons de desconto para primeira compra a frete grátis em períodos como *Black Friday* e Natal. Neste aspecto existem os comparadores de preço, como Zoom, Bondfaro e Buscapé, que impulsionam a livre concorrência: pequenas empresas conseguem competir com grandes anunciantes e o consumidor se preocupa com algo além de somente a marca da loja - o preço e o prazo de entrega passam a ter um grande peso nessa escolha. O produto ofertado vira a grande estrela, e isso faz com que pequenas empresas consigam mais espaço no mercado. Por fim, é impossível bater a diversidade de produtos no comércio eletrônico: a variedade não está mais presa à limites geográficos. É possível adquirir um *smartphone* de última geração diretamente de seu fabricante na China, e inclusive parcelar o pagamento utilizando seu cartão de crédito nacional (SALVADOR, 2013).

Com este crescimento exponencial no mercado de *e-commerce*, a cada dia aumenta o interesse por parte de empresas e pequenos empreendedores de criarem seu espaço virtual e começar a vender *online*. Mas apesar de vender *online* ser algo teoricamente fácil, nada disso é possível se o empreendedor não tiver em suas mãos a “ferramenta perfeita” para o seu negócio. Visando isso, a ateliware - empresa desenvolvedora de *software* - criou um produto chamado minestore, que possibilita a criação de uma loja *online* sem necessidade de conhecimentos técnicos para desenvolver um *website* e colocá-lo no ar. Uma solução simples e rápida para pessoas interessadas em ter seu negócio expandindo para a internet.

Entretanto, apesar de ser uma grande plataforma no mercado, a minestore vem enfrentando alguns problemas em seu funil de vendas<sup>1</sup>. Muitas pessoas interessadas criam sua loja *online*, mas poucos se tornam lojistas e chegam a etapa final, que é possuir uma loja considerada ativa e que traga um retorno financeiro para ambos os interessados. Segundo a analista comercial da empresa, de 40.000 lojas, 17.000 estão ativas, e menos de 800 demonstram retorno significativo. Isso mostra a possível ausência de algo importante para os clientes, mas ainda desconhecido pela empresa, uma vez que tão poucos mantêm-se ativos no negócio.

A minestore busca, então, melhorar suas métricas, a fim de encontrar uma solução que faça a vazão de seu funil de vendas aumentar e trazer mais lojistas ativos. Para isso, espera-se poder identificar potenciais lojistas que demonstram interesse no produto, a fim de que se tornem futuros clientes para a empresa, dando origem a mais um termo importante ao funil de vendas: o de *leads*. Sendo um conceito popularizado com o *marketing* digital, um lead é um consumidor potencial de uma marca, caracterizado por ter tido algum tipo de interação com ela. Ao ceder seu e-mail para a base de vendas, o *lead* avança no funil de vendas da empresa. Esse é o momento em que o negócio começa a tratar seus “visitantes” como clientes em potencial, desenvolvendo estratégias para gerar contatos e enviá-los para a parte comercial.

Com isso, *lead* tornou-se sinônimo de qualquer visitante que informe seus contatos em troca de algum tipo de conteúdo. Para a minestore, esses *leads* enquadram-se em quais lojistas, após o cadastro, têm a maior probabilidade em criar e manter uma loja ativa e rentável para eles, tornando-se um cliente no final.

### 1.1. OBJETIVO GERAL

O objetivo deste trabalho é desenvolver o *software* iMine que, em conjunto com a minestore, seja capaz de realizar uma consulta aos dados da plataforma *e-commerce* minestore e suas métricas de eventos, buscando uma solução para otimizar a visualização de dados sobre os clientes e suas lojas, bem como também descobrir potenciais clientes para a empresa.

---

<sup>1</sup> Funil de vendas (ou *pipeline*) é um modelo estratégico que mostra cada etapa seguida pelo consumidor até o fechamento da compra (CONTENT, s.d.).

## 1.2. OBJETIVOS ESPECÍFICOS

- Realizar a análise do negócio minestore;
- Permitir a visualização otimizada de uma base de indicadores referentes aos dados de acesso e utilização gerais da minestore;
- Disponibilizar gráficos para uma melhor visualização dos indicadores anteriores;
- Analisar os dados de *leads* no funil de vendas, por meio de gráficos;
- Disponibilizar um algoritmo de descoberta de *leads* que potencialmente se tornarão lojistas de sucesso;
- Documentar o sistema.

## 1.3. PROBLEMA

Antes da implantação do sistema desenvolvido no presente trabalho o cliente identificou três problemas principais. Esses viriam a ser o objeto de justificativa para a parceria entre o cliente e a equipe no desenvolvimento do sistema.

O primeiro problema é relacionado ao modo que o cliente visualizava os indicadores gerais das lojas da minestore. Segundo ele, a planilha de indicadores era pouco usada, uma vez que a alimentação de seus dados era feita de forma manual em um certo dia específico da semana. A inconsistência de dados também foi levantada, já que haviam semanas onde a interação manual não era feita e as colunas ficavam vazias.

O segundo problema, ainda relacionado com os indicadores gerais, era que o cliente não possuía uma forma dinâmica de visualizar esses dados, não conseguindo chegar a conclusões relevantes com tantos números armazenados.

O terceiro problema era o de que muitos lojistas se cadastravam no site da minestore, porém poucos mantinham a loja ativa e rentável para a empresa.

As soluções para esses problemas são apresentadas no próximo tópico do documento como forma de justificativa para a execução do presente trabalho.

## 1.4. JUSTIFICATIVA

A solução para o problema da visualização dos indicadores gerais foi resolvida com a implementação de um *gateway* para a obtenção automática de diferentes fontes de dados. Nesse primeiro momento, o *gateway* está adaptado para

a leitura de arquivos .xls manualmente, mas o cliente pode configurar o *gateway* para receber sua fonte de dados específica. A planilha se tornou otimizada, mostrando gráficos de linha para a evolução de cada um dos indicadores. Desse modo, os dados se tornam confiáveis.

Para o problema do dinamismo dos indicadores gerais, o sistema conta com os principais gráficos que o cliente solicitou, sendo eles os de indicadores gerais (números de visitas nas lojas, lojas com vendas, pedidos pagos, lojas ativas, total de visitas - por semana).

Por fim, como solução para o terceiro problema levantado, o sistema conta com uma análise de promessas de potenciais clientes que se tornarão futuros lojistas. Isso foi alcançado com a proposta de utilização de técnicas de *Data Mining*, que são a “mineração do conhecimento a partir de uma grande quantidade de dados” (HAN; KAMBER, 2006, p. 5). Desse modo, com base nos dados da empresa, realiza-se uma série de processamentos, descobrindo valores mais precisos e encontrando os potenciais futuros clientes. Com o resultado em mãos, é possível então a empresa tomar decisões e realizar mudanças para o seu sucesso.

## 1.5. ESTRUTURA DO DOCUMENTO

Este documento apresenta as informações do desenvolvimento do projeto e estrutura. O Capítulo 2 tem como objetivo demonstrar os embasamentos teóricos do projeto.

O Capítulo 3 possui a descrição dos materiais e métodos utilizados pela equipe de desenvolvimento. Este capítulo possui detalhamento sobre a metodologia de desenvolvimento, passando por tecnologias utilizadas, cronograma de atividades e responsabilidades.

No Capítulo 4 é possível encontrar informações da arquitetura e do funcionamento do sistema iMine em si, regras de negócio, descrição de funcionalidades e exemplos de telas implementadas.

O Capítulo 5 conclui o trabalho e apresenta aspectos positivos e negativos encontrados no decorrer do desenvolvimento do presente trabalho, contando também com as suas possíveis melhorias futuras.

Por fim, no final do documento, podem ser encontrados apêndices contendo diagramas e documentações UML. Estes podem ser consultados para maior

elucidação conceitual e teórica da implementação que foi utilizada para o desenvolvimento da aplicação iMine.

## 2 FUNDAMENTAÇÃO TEÓRICA

Como citado por Salvador (2013, p.11), o *e-commerce* ou comércio eletrônico se trata de uma “transação comercial feita através de meios eletrônicos, com transferência de fundos e dados por meios eletrônicos e troca de informações”. Em um conceito mais amplo e simples, pode-se pensar em uma compra realizada a distância. Basicamente, é ter o poder de ter conforto e comodidade ao alcance de seus dedos, em qualquer lugar e, principalmente, por meio de qualquer dispositivo compatível (celular, *tablet*, computador).

### 2.1. E-COMMERCE NO MERCADO

*E-commerce* podem ser classificados em várias categorias, tais como: B2B (*business-to-business*), de empresas para empresas; B2G (*business-to-government*), de empresas para órgãos públicos; B2C (*business-to-consumer*), de empresas para consumidores; e C2C (*consumer-to-consumer*), de consumidores para consumidores. Cada uma dessas categorias possui opções políticas comerciais diferentes para orquestrar seus negócios, pois cada uma lida com portes variados de clientes (VALLE, 2016).

Por exemplo, políticas B2C não funcionam com e-commerces B2B. As necessidades comerciais são diversas: algumas empresas são mais sensíveis a descontos para pagamento à vista, outras preferem ter parcelamento estendido em 15, 30, 45, 60 dias; outras compram em grandes quantidades e com periodicidade e precisam de preços baixos, em contrapartida com outras que compram em pequenas quantidades e vendem unidades com preços mais elevados. O fator determinante do modelo de *e-commerce* que uma empresa adota, segundo Valle (2014), “não são os produtos em si ou as quantidades negociadas, mas sim o destino das mercadorias vendidas”. O modelo B2B, adotado pela minestore – cliente deste projeto – possui as seguintes opções de políticas comerciais: formas de pagamento, rotinas contábeis diferenciadas, navegação orientada para este tipo de negócio, uma vez que não lida com um cliente final, mas sim outra empresa.

Aprofundando-se na temática de *e-commerce* é fácil de entender tamanho favoritismo no comércio pela *internet*. Entretanto, com grandes crescimentos,

grandes mudanças devem ser feitas no gerenciamento de um negócio. Uma dessas mudanças é supervalorizar o relacionamento com o cliente, elevando-o a um patamar especial. O cliente deixa de ser apenas mais “uma venda finalizada” e passa a ser uma pessoa importante para a evolução contínua do negócio. Sua voz passa a ter a necessidade de ser mais ouvida do que seria em uma loja física, por exemplo, já que se trata de um relacionamento empresa - cliente por meio da *internet*, onde nenhum dos dois jamais se viu ou verá pessoalmente. Estratégias e sistemas integrados de gestão com foco específico são necessários, resultando na centralização de processos e tarefas organizadas e integradas em torno desses clientes (WENNINGKAMP, 2009).

## 2.2. CRM COM E-COMMERCE

Em menos de dez anos, *Customer Relationship Management* (Gestão de Relacionamento com o Cliente), ou CRM, tornou-se um tópico de bastante importância. Conceituado como uma estratégia de negócio voltada à otimização da rentabilidade da empresa, promovendo satisfação e fidelidade com o cliente, o CRM provou ser essencial em negócios que desejam priorizar sua automação de força de vendas no mercado. Suas tecnologias habilitam estratégia, identificando e gerenciando o relacionamento com os clientes, pessoal ou virtualmente. *Softwares* de CRM fornecem funcionalidades para as companhias em quatro segmentos: vendas, *marketing*, serviço ao cliente e comércio digital (GARTNER, 2013).

Apesar de aparecer apenas no final dos anos 90, os princípios que o baseiam são muito mais antigos, uma vez que são originários desde o início do comércio comum. De qualquer modo, os responsáveis pela repentina fama da temática do CRM foram uma série de significantes tendências que, em conjunto, deram forma a melhores oportunidades para servir clientes por meio de um *marketing* de relacionamento habilitado à informação (*information-enabled relationship marketing*). Existem inúmeros fatores que impactam nas habilidades das organizações de construir e gerir relações mais duradouras, especialmente em negócios com muitos clientes. São eles: aumento do poder dos computadores; baixo custo dos computadores; aumento de armazenamento dos computadores; redução significativa de custo de armazenamento de megabytes de dados; aumento da variedade de ferramentas para realizar análises dados, mineração de dados e

visualização de dados; aumento do *e-commerce* e sucesso em atingir clientes pela *internet* sob um custo muito mais baixo; aumento do reconhecimento da importância da retenção do cliente e valor de vida útil do cliente; e por fim, mas não menos importante, a sofisticação na abordagem de *marketing* e desenvolvimento de melhores soluções para atingir clientes, incluindo o *one-to-one marketing (1:1)*, abordagem onde o vendedor lembra os detalhes sobre as preferências de cada cliente e suas características, usando esse conhecimento para providenciar um serviço melhor (PAYNE, 2005).

Mais do que nunca vemos que o mercado atual está muito diferente daquele que conhecíamos, ainda mais com o progresso da tecnologia. Os antigos conceitos de que o cliente deveria se contentar com o que a empresa tem a oferecer já não mais se adaptam ao escopo presente. O cliente possui maior poder de decisão em suas mãos, sem falar sobre uma maior necessidade no dia-a-dia, optando por toda acessibilidade que possa lhe auxiliar a atingir seu objetivo com rapidez e segurança. Para isso, ele escolhe trabalhar com empresas que utilizam uma forma de venda diferente, a qual não tenta “empurrar” produtos e serviços a ele, mas sim permitindo ao cliente tomar decisões de compra no seu tempo. Para a empresa, sua vantagem está em encontrar o cliente na hora certa, abordá-lo na hora exata, e sugerir o que ele precisa na hora perfeita, concretizando assim a sua venda (PAYNE, 2005).

E é por meio desse diferencial que é utilizado o termo de funil de vendas, que é uma ferramenta essencial para otimizar o processo de vendas, entender o processo de compra do cliente e aumentar sua eficiência. Basicamente, é a jornada de guiar o cliente pelas fases, desde o contato inicial com a empresa, até a concretização da compra de seu produto/serviço, mostrando o perfil do cliente sobre a solução, julgamento, aprovação e, finalmente, o compartilhamento da experiência da marca da empresa pelo seu cliente (ANDRADE, 2017).

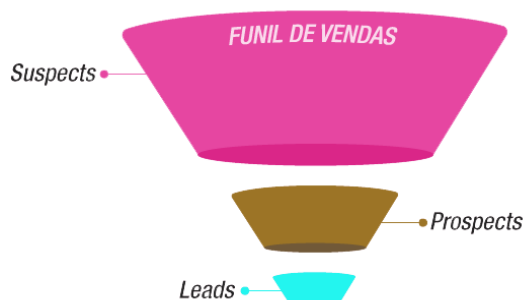
### 2.3. FUNIL DE VENDAS

Segundo os grupos de empreendedorismo Endeavor e Agendor (s.d.), cada empresa pode possuir o seu próprio Funil de Vendas. O primeiro passo de sua criação seria uma análise do perfil de seus clientes. Ter o perfil bem construído vai ajudar a determinar onde focar seu tempo, guiar o desenvolvimento de produtos, e

manter todos da empresa no mesmo ponto, principalmente seu time de vendas, na hora de criar iniciativas de atração de potenciais clientes.

Feito isso, é necessário definir as etapas de seu funil de vendas. Não existe uma regra definida, mas geralmente ele pode simplesmente ter três etapas genéricas, como *suspect*, *prospect* e *lead*, como mostrado na Figura 1. Como as definições desses termos podem ser muito abrangentes, embasaremos nos conceitos do CEO da empresa InVert Inc., que possui definições universais desses termos. Segundo Hawk Mikado (2016), um *suspect* pode ser qualquer um que talvez compre sua oferta. A empresa nem ao menos sabe se um *suspect* está interessado no seu mercado. Pode ser um indivíduo ou organizações com potencial necessidade do serviço ou produto. Um *prospect* é um indivíduo ou uma organização que é um possível cliente para comprar produtos oferecidos e já passou por alguns processos de qualificação. Um *prospect* que está interessado irá se associar com a empresa consistentemente, enquanto um *suspect* apenas irá se associar quando achar seguro. Enquanto *suspects* são um universo potencial maior de clientes, *prospects* são auxiliares ao longo do funil. Eles manifestam alguns fortes indicadores de que estão interessados em comprar o produto e que estão dispostos a dividir informações pessoais para isso. *Prospects* chegam ao nível final se tornando *leads* capacitados. *Leads* são *prospects* que satisfazem os critérios da empresa. Eles precisam de produto ou serviço oferecidos e tem a condição de comprar. Um *prospect* se torna um *lead* quando se reconhece uma necessidade potencial. Quanto mais urgente a necessidade, mais qualificado o *lead* é para se tornar um cliente (MIKADO, 2016).

FIGURA 1 - FUNIL GENÉRICO



Fonte: Adaptado de MOURA (2016).

Indiferente dos nomes e números das etapas do funil, a ideia é conduzir o cliente pelo processo de *marketing* (comunicação e relacionamento) até a concretização da venda. As etapas genericamente podem ser divididas em topo meio e fundo, e diferentes estratégias são recomendadas em cada uma delas.

O topo do funil seria responsável por atrair o maior número de clientes em potencial para o negócio, e todas essas pessoas, interessadas, mas que ainda não demonstraram nenhuma intenção de compra ou de interesse para solucionar o problema são os *suspects*. O melhor tipo de conteúdo para fornecer nessa etapa são informações e conceitos que apresentem soluções para suprir as necessidades dos interessados. Como o usuário ainda não conhece a empresa, materiais mais abrangentes e educacionais são bem-vindos, como guias, lista de dicas e explicação de contextos desconhecidos pelo *suspect*. Exemplo: “9 dicas para quem está começando a fotografar”, “O que é Escada de valor? ”, Vídeos informativos, infográficos (DAGOSTIM, 2016; ANDRADE, 2017; FUZZATO, s.d.).

O meio do funil é a etapa onde os clientes demonstram que estão interessados na empresa, desejam resolver algum problema e estão em busca da melhor solução e estão dispostos a oferecer dados pessoais como nome e e-mail em troca de seu conteúdo. É percebido como um *prospect* quando ele faz parte de seu público alvo de seu marketing, com o conhecimento de que precisa do seu produto/serviço, disponibilidade financeira e capacidade de tomada de decisões. É o cliente em potencial, alinhado com a empresa, mas às vezes ele ainda não está pronto para comprar, precisando de mais maturidade. Para isso, a empresa deve continuar nutrindo-o com conteúdo de qualidade para que ele possa passar para a próxima fase. O melhor tipo de conteúdo para essa fase é não tentar vender o produto, pois o cliente ainda está analisando qual será a melhor solução e precisa-se convencê-lo de que a opção oferecida é melhor que a do concorrente. O conteúdo deve ter mais profundidade, mostrando quais são os benefícios de adquirir o seu produto. Exemplo: publicações do tipo “Como fazer - O que é e como fazer uma oferta Tripwire? ”, *e-mails* personalizados, guias e manuais, *ebooks* (DAGOSTIM, 2016; ANDRADE, 2017; FUZZATO, s.d.).

O fundo do funil é onde o interessado é chamado de *lead* qualificado, pois ele já está de fato considerando a compra. O tipo de conteúdo fornecido precisa

reforçar os benefícios, valor e credibilidade da marca, esclarecendo dúvidas específicas e demonstrando porque a marca é melhor que a concorrência. Por exemplo, pode-se citar estudos de casos de sucesso, testemunhos de clientes, pesquisas de mercado e tabelas de preço. Um contato mais pessoal por *e-mail* ou telefone pode ajudá-lo a esclarecer dúvidas e então motivá-lo efetuar a compra. *Free-trials* podem também incentivá-lo para conhecer o produto e criar confiança.

Se o funil genérico não for o suficiente para a empresa, ele pode ser ainda dividido em outras categorias, como, por exemplo, em sete etapas mais detalhadas, mostradas na Figura 2. São elas: prospecção (busca de potenciais clientes), qualificação (se cliente quer/pode/precisa da sua solução), apresentação (primeiro contato com o produto), maturação (o cliente analisa suas opções disponíveis e condições), negociação (reajuste de preços), fechamento (clareza na entrega), pós-venda (satisfação dos clientes e *follow-up*) (ENDEAVOR; AGENDOR, s.d.).

FIGURA 2 - AS SETE ETAPAS DE UM FUNIL DE VENDAS



Fonte: Adaptado de ENDEAVOR; AGENDOR (s.d.)

#### 2.4. FUNIL DE VENDAS DA MINESTORE

O modelo de funil de vendas da minestore apresenta a seguinte estrutura: *Lead*, *Lead* qualificado, Oportunidade e Cliente, representados na Figura 3.

FIGURA 3 - O FUNIL DE VENDAS DA MINESTORE



Fonte: Adaptado de PEREIRA (2016).

Para a empresa, um *Lead* é uma pessoa que somente cadastra seu *e-mail* no sistema, enquanto um *Lead* qualificado é alguém que tentou criar a loja eletrônica. Passando dessa etapa, ao completar o *checklist*, formado por 8 passos referentes ao gerenciamento básico da loja, o *Lead* Qualificado se torna uma Oportunidade. Finalmente, na base do funil, ele deixa de ser uma Oportunidade assim que realiza 2 vendas, tornando-se um Cliente.

Nota-se que, com o funil de vendas da minestore, é possível classificar os usuários de acordo com sua evolução nas etapas de forma simples, mas também que os “curiosos” acabam ocupando 50% do funil, enquanto as oportunidades e os clientes ocupam apenas 25% cada. Ainda assim, apenas essa visualização dos usuários não é capaz de resolver o maior problema atual do sistema, que é a enorme quantidade de *Leads* e *Leads* qualificados que dificilmente chegam até a última camada, a de Clientes.

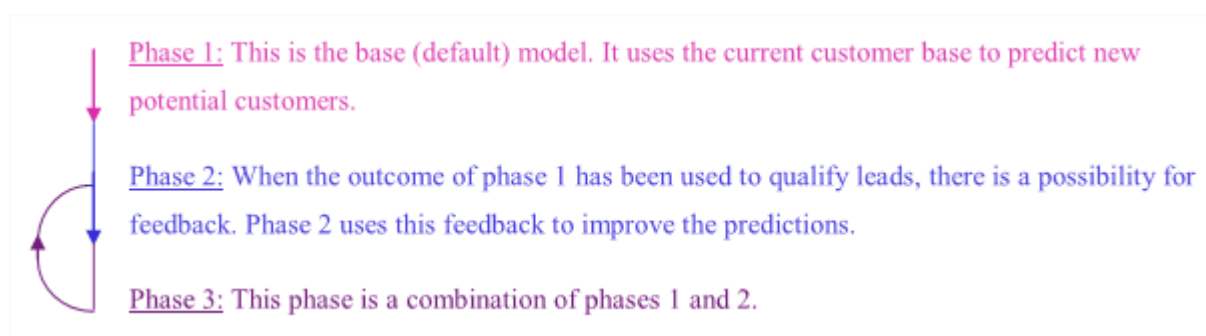
Dessa forma, muitas vezes apenas saber como funciona o funil não é o bastante para garantir vendas e clientes duradouros. Muitos *Leads*, *Leads* qualificados e até mesmo Oportunidades desistem antes de chegar ao fundo do funil, o que não os classifica como verdadeiros clientes. É necessária uma análise de dados a fundo, afim de se descobrir resultados que contribuirão para a resolução do problema. E é por causa deste ponto que a solução abordada no artigo de D’Haen e Poel (2013) foi mais do que escolhida para este projeto, pois os autores

desenvolvem um jeito de identificar um perfil de um cliente do negócio e de descobrir novas pessoas com grande potencial em se tornarem clientes. Esses métodos citados seriam utilizados com os clientes da minestore, como será explicado no tópico seguinte.

## 2.5. PREDIÇÃO DE *PROSPECTS*

D’Haen e Poel (2013) desenvolveram um modelo de predição de *prospects* com suporte a B2B (*business-to-business*). Esse modelo consegue usar dados dos clientes e inferir quais pessoas têm a maior probabilidade de virarem *leads*, baseando-se no perfil de pessoas que já são clientes. Isso gera uma lista de *prospects* com maior qualidade e com maiores chances de virarem *leads*. Basicamente, o modelo é dividido em 3 fases (Figura 4): a primeira utiliza informações de clientes já existentes e utiliza clusterização e um algoritmo de vizinho mais próximo para encontrar predições; a segunda usa o *feedback* das predições da primeira fase, sabendo agora quais das predições não viraram clientes e otimiza o algoritmo; a terceira combina as fases 1 e 2 (D’HAEN; POEL, 2013).

FIGURA 4 - FASES DO MODELO



Legendas: Fase 1: Este é o modelo base. Utiliza a base atual de clientes para predizer novos clientes em potencial; Fase 2: Quando o resultado da fase 1 já tiver sido usado para qualificar *leads*, há uma possibilidade de *feedback*. A fase 2 usa o *feedback* para melhorar a previsão; Fase 3: esta fase é uma combinação das fases 1 e 2. Fonte: Adaptado de D’HAEN; POEL (2013).

O modelo proposto se baseia em CRM (*Customer relationship Management*) analítica, que envolve minerar dados relacionados ao cliente para conseguir novos clientes. A maioria das literaturas de CRM ignoram a etapa de aquisição de cliente, focando mais na parte de retenção pois geralmente as estratégias têm menor custo. Há diversas razões para se preocupar também com a aquisição de novos clientes, e para o caso da minestore é a parte mais importante, pois o cliente passa apenas

uma vez pelo funil e é nessa etapa que está ocorrendo a maior perda de clientes (D'HAEN; POEL, 2013).

Um ponto forte para esse trabalho é que o autor cita que o modelo pode ser usado indiferente das características da empresa, funcionando tanto para uma empresa grande no setor automobilístico como para uma empresa pequena no ramo de alimentação. Apesar do algoritmo funcionar indiferente de que companhia a use, a ferramenta de análise funcionará melhor em mercados que estão saturados, com muitos clientes em potencial. Acredita-se que isso seja bom para o presente projeto, pois a minestore tem um funil de vendas com muitos interessados (D'HAEN; POEL, 2013).

O funil de vendas apresentado por D'Haen e Poel (2013) ajuda a descrever a aquisição de clientes dividindo o funil em estágios: *suspect*, *prospect*, *lead*, definições já apresentadas neste documento. Dentre eles, o *lead* é a “promessa” proposta. A ideia do algoritmo é buscar nessa lista de *prospects* quem são os *leads*, ou promessas, usando como base os clientes já estabelecidos na loja (D'HAEN; POEL, 2013).

A primeira fase do modelo proposto consiste em alimentar o algoritmo com uma base de *prospects*, tentando identificar características similares entre eles. Para isso, os *prospects* similares recebem uma clusterização, que se trata de uma classificação não supervisionada onde não se conhece padrão, nem o número total de classes a serem encontradas durante a classificação. O conjunto de dados é particionado em grupos, baseados em características específicas, tais que os pontos dentro de um *cluster* (agrupamento) sejam mais similares do que os pontos de outros grupos.

Em seguida, é executado um algoritmo de vizinho mais próximo para qualificar melhor esses possíveis clientes. Assim que é obtida a resposta de quais dessas promessas realmente virarão clientes. Inicia-se, então, a fase 2, pois com esse *feedback* é possível aprimorar o algoritmo. A fase 3 combina a fase 1 e 2 e gera mais *feedback*, alimentando a fase 2 novamente e entrando em um *loop*, sempre em aprendizado (D'HAEN; POEL, 2013).

Ao fim do artigo, os autores apresentam que foram feitos 2 pseudo-testes para analisar a qualidade do algoritmo, com dados verdadeiros, embora não sejam testes de uma situação da vida real. O primeiro usou uma lógica reversa, utilizando uma lista de 56 *prospects* qualificados positivamente pela empresa para tentar encontrar os 107 clientes originais numa base de dados. Com esses dados de entrada, foram obtidos 228 *prospects*, os quais 8 eram clientes reais. Presumindo que apenas estes 8 virariam clientes, a taxa de conversão de *prospect* para cliente seria de 3,5%, o que está dentro do esperado, ainda que só rodando a fase 1 (D'HAEN; POEL, 2013).

O segundo teste realizado foi uma combinação das 3 fases. Foram selecionadas características para o perfil aproximado das companhias clientes, como número de vendas maior que 100.00 e menor que 190000, número de empregados maior que 4 e menor e igual a 50 entre outras, o que trouxe 36 resultados. Dez foram selecionadas randomicamente como clientes atuais e utilizados na fase 1, para tentar encontrar as 26 restantes. O algoritmo do vizinho mais próximo foi então rodado e 10 *prospects* tiveram a mais alta similaridade, todos dentre essas 26 companhias. Na segunda execução o vizinho mais próximo foi executado novamente pois não houve *feedback* negativo para aprimorar o algoritmo, e foram encontrados 123, nenhum entre os perfis de cliente. A terceira execução encontrou 10176 *prospects*, dos quais 12 eram os clientes. Na quarta execução foram encontrados mais 2 *prospects*, e ambos estavam na base de clientes. Execuções subsequentes não conseguiram encontrar os dois clientes. Conclui-se que em apenas 4 execuções do segundo teste, foi possível encontrar 24 dos 26 clientes da base em uma lista de *suspects* que possuía mais de 16 milhões de companhias (D'HAEN; POEL, 2013).

Resumidamente, o artigo mostra uma solução adequada e que se enquadra na necessidade da empresa minestore, uma vez que a descoberta dos potenciais clientes pode ser alcançada no final da primeira fase, sendo as seguintes melhores filtragens e ajustes a descoberta de conhecimento.

Por último nessa sessão, mas não menos importante, a minestore levanta mais um problema, que é a atual planilha de métricas gerais que eles utilizam para computar os dados gerais das lojas, a qual será descrita a seguir.

## 2.6. PLANILHA DE MÉTRICAS NA MINESTORE

Atualmente a minestore utiliza uma planilha de dados online como fonte das métricas gerais das lojas, preenchidas manualmente em um dia específico da semana pela analista comercial da empresa.

Como resultado de um processo manual, a consistência dos dados deixa a desejar, já que a inserção dos números depende da ação humana, que nem sempre é possível. Muitos desses dados estão inconsistentes ou simplesmente nulos. Outra dificuldade enfrentada ainda é a troca de tecnologia de rastreamento de eventos de usuários anteriormente realizada pela empresa Intercom, desenvolvedora de plataformas responsáveis em computar eventos específicos como, por exemplo, quantas vezes uma loja é visitada. Com a finalização da parceria, a obtenção de muitas métricas foi descontinuada, já que utilizavam essa fonte para serem atualizadas. Sem falar em outros rastreadores que também foram deixados para trás, como o Piwik. A Figura 5 mostra uma relação das diferentes fontes de dados utilizados pela minestore.

FIGURA 5 - DIFERENTES FONTES DE DADOS DA MINESTORE

	A	BG	BH	BI	BJ	BK	BL
1		S138	S139	S140	S141	S142	
2		01/02/2017	08/02/2017	15/02/2017	22/02/2017	01/03/2017	
3	<b>Financeiro período</b>						
5	Lojas com vendas no período						Mixpanel: People > Revenue > Escolher data de quarta
6	Pedidos Pagos						
7	Lojas ativas						Mixpanel: People > Explore > Bookmarks > Lojas Ativas
8							
9	<b>Crescimento</b>						
10	Visitas na home						Analytics > Visibilidade "Landing" > Aquisição > Todo o
11	Blog						Analytics > Visibilidade "Landing" > Aquisição > Todo o
12	Direto						Analytics > Visibilidade "Landing" > Aquisição > Todo o
13	Lojas clientes						Analytics > Visibilidade "Landing" > Aquisição > Todo o
14	Busca Orgânica						Analytics > Visibilidade "Landing" > Aquisição > Todo o
15	Novas lojas						
16	20 - setup para cliente (doing setup for customer)						intercom > People
17	21 - primeira loja online						intercom > People
18	22 - já vende online (already sell online)						intercom > People
19	Visitas nas lojas						piwik_soma do total de "visitas" dos dias
20	Canções nas lojas						piwik_soma do total de "canções" dos dias
21	bcash						Order.where('orders.paid_at >= ? AND orders.paid_at <= ?
22	pagseguro						Order.where('orders.paid_at >= ? AND orders.paid_at <= ?
23	Pedidos cancelados						Order.where('orders.created_at >= ? AND orders.creat
24	moip						Order.where('orders.created_at >= ? AND orders.creat
25	bcash						Order.where('orders.created_at >= ? AND orders.creat
26	pagseguro						Order.where('orders.created_at >= ? AND orders.creat

Legenda: Exemplo das diferentes fontes de dados usadas pela minestore. MixPanel, Google Analytics, Intercom, Piwik e o próprio banco de dados da empresa são usados para armazenar informações diferentes. Fonte: Arquivo interno da minestore.

Ao fim desse capítulo, conclui-se a fundamentação teórica do projeto, expondo conceitos a abordagens relacionadas ao assunto. No Capítulo 3 são apresentados os materiais e a metodologia adotada para o desenvolvimento desse trabalho.

### 3 MATERIAIS E MÉTODOS

Este capítulo apresenta o método e etapas utilizados para o desenvolvimento da solução iMine, bem como as tecnologias empregadas. O primeiro tópico a ser descrito é a metodologia de desenvolvimento ágil.

#### 3.1. DESENVOLVIMENTO ÁGIL

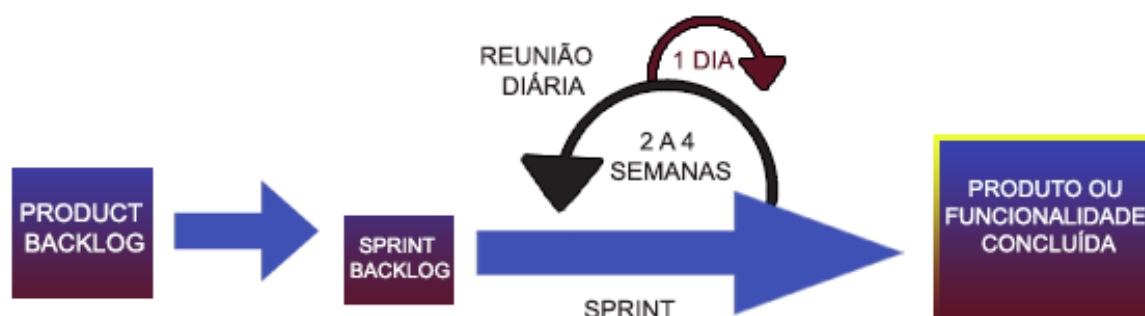
Na década de 90, o desenvolvimento de software já existia, mas ele era lento, tradicional e cheio de regras regulamentadas. Esse conceito começou a mudar com a necessidade de mudanças no mercado de *software*, que experimentava novas formas de agilizar a entrega de produtos e tornar o desenvolvimento uma tarefa mais ágil. Em 2001, foi assinado o documento conhecido como Manifesto Ágil, que tem como valores que os indivíduos, interações, *software* (ou produto) em funcionamento, colaboração com o cliente e resposta a mudanças são mais importantes para o sucesso do projeto do que a utilização de processos, ferramentas, documentação, contratos e planos (BECK, et al. 2001).

O desenvolvimento ágil pode ser aplicado de diferentes formas em um projeto, como pela utilização do *Scrum*, que é um *framework* simples para gerenciar projetos complexos. O *Scrum* é aplicado avaliando o nível de conhecimento dos requisitos e tecnologias que a equipe tem sobre o projeto. Caso as funcionalidades sejam habituais e as tecnologias bem dominadas, a utilização de metodologias com processos definidos, baseados em estilo cascata, é mais aconselhada (VIEIRA, 2014).

O *Scrum* proporciona uma melhoria no nível de entrosamento dentro da equipe, e com isso o rendimento do projeto aumenta. A Figura 6 é um esquema de como o *Scrum* funciona. Nela pode-se observar que há uma série de funcionalidades desejadas que são colocadas em um estado chamado *backlog*. As funcionalidades selecionadas para execução estão na etapa chamada *Sprint backlog*. Inicia-se o processo de *Sprint* (período de tempo para realização de tarefas específicas), em que todos os dias deve-se fazer uma reunião rápida para informar toda a equipe sobre as tarefas e atividades que estão em execução. O processo de

*Sprint* leva, normalmente, entre duas e quatro semanas. Ao final de cada *Sprint* são geradas como produto funcionalidades entregáveis ao cliente. É realizada então uma reunião de retrospectiva e um novo conjunto de funcionalidades é separado para que uma nova *Sprint* seja iniciada (SCHWABER, et al. 2002; BISSI, 2007).

FIGURA 6 - ESQUEMA REPRESENTATIVO DO PROCESSO SCRUM



Legenda: *Product Backlog* é representado pela caixa mais à esquerda contendo várias tarefas dentro. A *Sprint Backlog* é representada pela segunda caixa contendo um número determinado de tarefas a ser utilizado na *Sprint*. O processo de *Sprint* é representado pelo loop contendo um loop secundário para a representação das reuniões diárias. Por fim, as funcionalidades concluídas são representadas pela caixa à direita. Fonte: Adaptado de VIEIRA (2014).

### 3.2. ADAPTAÇÃO DA METODOLOGIA ÀS NECESSIDADES DA EQUIPE

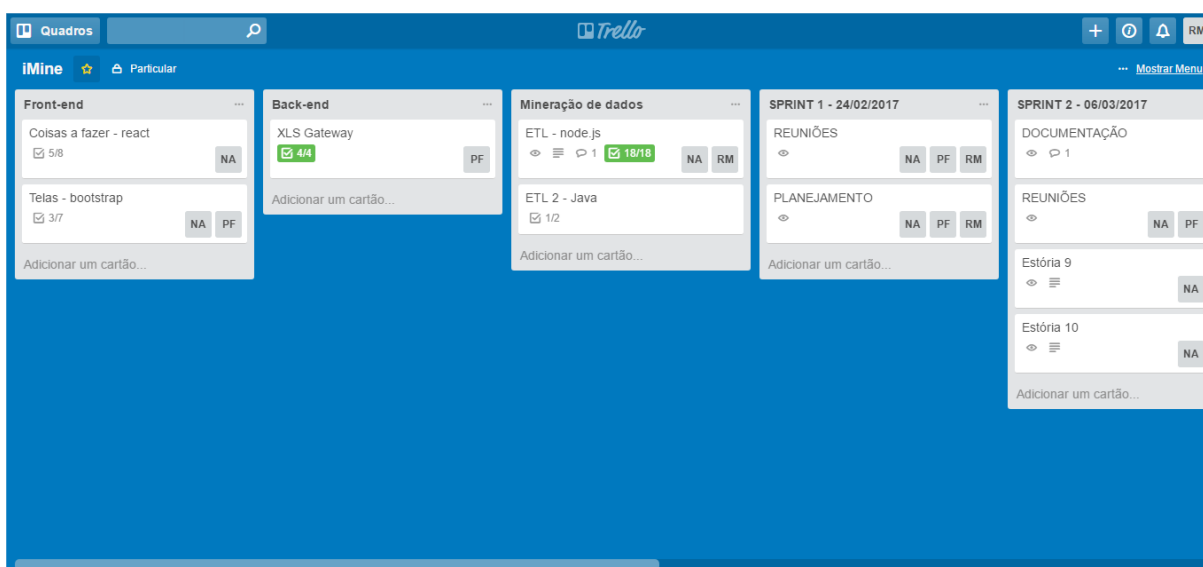
O *framework Scrum* foi adaptado para as necessidades do projeto conforme citado a seguir.

As reuniões diárias previstas pela metodologia foram executadas majoritariamente via ligações utilizando o Skype. Além disso, pelo menos uma vez por semana, era feita uma reunião com a orientadora e os integrantes. Apenas quando a pauta era um assunto mais sensível, relacionado aos estudos sobre a estrutura do *gateway* e mineração de dados, as reuniões eram presenciais. Dessa forma, cada um dos integrantes poderia continuar trabalhando nas tarefas após o término das reuniões, sem se preocupar com o deslocamento diário.

O controle das tarefas a serem executadas foi realizado em conjunto com o cronograma de atividades, na ferramenta Excel, do pacote Office do Windows. A

justificativa desta escolha se dá ao fato de que os requisitos do desenvolvimento eram poucos, considerando o tamanho do projeto, porém complexos. A utilização de uma ferramenta em estilo Kanban (Trello), onde pode-se configurar etapas de progresso de tarefas, assinalar tarefas a pessoas e controlar o estado das tarefas, foi experimentada, mas acabou se tornando uma fonte de controle pessoal para saber os *checklists* que cada integrante evoluía no desenvolvimento (Figura 7).

FIGURA 7 - EXEMPLO DE UTILIZAÇÃO DA APLICAÇÃO TRELLO



Legenda: Exemplo de utilização da aplicação Trello para auxiliar no gerenciamento das tarefas. Screenshot feita no dia 11/06/2017. Fonte: Os Autores (2017).

Também foi necessário estipular quem seria responsável por qual papel, então foi definido que a Raissa seria a *Product Owner*, responsável em definir o *backlog* que contém todo o trabalho que seria executado durante as *Sprints*, Nicole a *Scrum Master* e Paulo o líder técnico de *back-end*. Em cada reunião os membros da equipe se limitavam a passar os status de suas respectivas atividades, apresentando quais tinham sido os progressos desde a última reunião, quais eram as próximas ações que executariam para completar as tarefas a eles assinaladas e, por fim, se existiam impasses, que deveriam ser solucionados em grupo. Caso um dos membros da equipe não fosse necessário na discussão essa pessoa estaria livre para continuar suas tarefas, sem ter a necessidade de comparecer ao encontro.

### 3.3. CRONOGRAMA DE ATIVIDADES E RESPONSABILIDADES

No início do projeto foi documentado um cronograma de atividades em que foram separadas as etapas em que cada funcionalidade seria desenvolvida de forma

parcial, pois a equipe ainda não possuía todas as informações neste momento, como será elucidado detalhadamente a seguir durante as *Sprints*. Dessa maneira, o cronograma foi feito com base em estimativas e experiência dos próprios integrantes da equipe. A versão final do cronograma pode ser visualizada na Tabela 1.

TABELA 1 - CRONOGRAMA DE ATIVIDADES

<b>Nº <i>Sprint</i></b>	<b>Data início</b>	<b>Metas da <i>Sprint</i></b>
1	24/02/2017	Análise negocial; Pesquisa <i>e-commerce</i> ; Estudo da ferramenta minestore.
2	06/03/2017	Reunião com o cliente para elucidação de processos; Criação do cronograma em visão macro; Análise da planilha dos indicadores gerais; Estudo sobre funil de vendas; Estudo sobre <i>data mining</i> ; Análise do modelo de dados da minestore.
3	21/03/2017	Análise do artigo base do projeto; Análise do funil da minestore; Definição de estrutura do projeto; Realização das Histórias do Usuário. Realização da documentação UML.
4	30/03/2017	Definição dos atributos de clientes para a clusterização; Definições detalhadas dos atributos; Finalização do escopo do projeto; Planejamento das <i>Sprints</i> .
5	06/04/2017	Reunião com o cliente para elucidação dos dados; Definição de arquitetura do <i>data mining</i> ; Definição das tecnologias a serem utilizadas; Lista de variáveis para a mineração de dados;

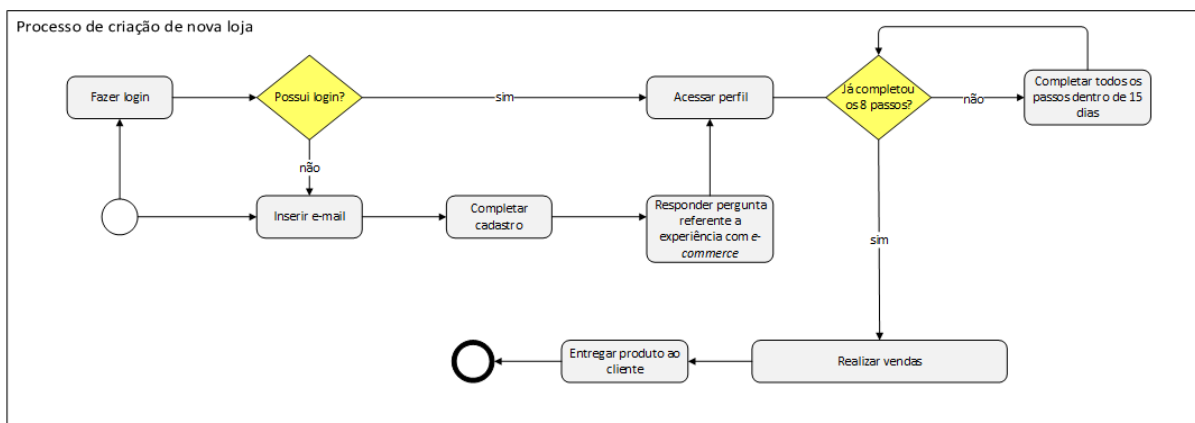
		Desenvolvimento do <i>gateway</i> dos indicadores gerais.
6	21/04/2017	Criação das prototipações; Apresentação da proposta do sistema para a minestore; Início do desenvolvimento do pré-processamento dos dados; Início do <i>front-end</i> da aplicação; Compilação do documento de TCC nas normas UFPR.
7	26/05/2017	Testes de clusterização; Estudo dos algoritmos de clusterização; Finalização do <i>gateway</i> ; Finalização dos <i>dashboards</i> ; Finalização do <i>data mining</i> .
8	26/06/2017	Entrega do sistema; Preparação da apresentação.
9	29/06/2017	Apresentação para a Banca; Entrega do sistema final ao cliente.

Legenda: Cronograma de atividades e Sprints detalhadas. Fonte: Os Autores (2017).

### 3.3.1 *Sprint 1*

Na primeira *Sprint* definida pela equipe, foi realizada uma análise do negócio do cliente, com cunho exploratório, afim de entender a plataforma criadora de *e-commerces* que eles disponibilizam no mercado. A análise do processo de criação de uma loja virtual online foi feita, assim como seus pontos críticos também foram observados e repassados a orientadora e a equipe do projeto. O Diagrama de Processo de criação de uma loja virtual pode ser visualizado na Figura 8.

FIGURA 8 - PROCESSO DE CRIAÇÃO DE NOVA LOJA NA MINESTORE



Legenda: Losangos em amarelo representam fluxos que necessitam de decisão, caixas cinza são ações simples. Fonte: Os Autores (2017).

Assim que foi finalizada essa análise, um estudo sobre *e-commerces* foi então realizado, visando sanar as dúvidas existentes sobre o assunto e aprofundar os conhecimentos do grupo, que deveria também levar ideias de soluções a empresa na reunião de elicitação de requisitos.

### 3.3.2 Sprint 2

A *Sprint 2* teve como principais objetivos definir o escopo do projeto e definir as métricas que seriam usadas para a mineração de dados.

A *Sprint* deu-se início a partir da primeira reunião com o cliente, onde necessidades foram expostas e a equipe começava a ter uma ideia da dimensão do escopo do sistema. Como resultado desse encontro e das reuniões semanais de orientação foi possível produzir o cronograma de atividades demonstrado na Tabela 1, e assim dar início à mais estudos específicos sobre novos termos aprendidos: funil de vendas e mineração de dados.

Também nessa iteração, em paralelo, houve a solicitação da planilha de indicadores gerais (Figura 9) e da modelagem do banco de dados da empresa para análise, com o objetivo de pensar em formas de otimizar a planilha no sistema e de selecionar atributos interessantes para algum tipo de extração de conhecimento.

FIGURA 9 - INDICADORES GERAIS DA MINESTORE

	C	D	E	F	G	H	I	J	K
1	S82	S83	S84	S85	S86	S87	S88	S89	
2	06/01/2016	13/01/2016	20/01/2016	27/01/2016	03/02/2016	10/02/2016	17/02/2016	25/02/2016	02/03/2016
41	<b>Crescimento</b>								
42	Visitas na home								
43	Blog								
44	Direto								
45	Lojas clientes								
46	Busca Orgânica								
47	Referência								
48	E-mail								
49	Twitter								
50	Facebook								
51	Google pago								
52	Ebook								
53	Facebook pago								
54	Ajuda								
55	Google Plus								
56	checkout minestore								
57	newsletter								
58	Adroll								
59	grupo facebook								
60	guest post								
61	novos leads								
62	20 - setup para cliente (doing setup for customer)								
63	21 - primeira loja online								
64	22 - já vende online (already sell online)								
65									
66	novos signups								
67	novos checklists completados								
68	Promessas (Ready to sell, +3 produtos e login na semana)								
69	Novas lojas ativas								
70									
71	lojas com login nos últimos 7 dias								

Fonte: Arquivo interno da minestore.

### 3.3.3 Sprint 3

Nessa *Sprint* os focos foram na procura de uma solução para aplicar a mineração de dados e na análise do funil de vendas da minestore, seguidos das documentações da arquitetura do sistema, histórias de usuário e a Linguagem de Modelagem Unificada (UML).

A mineração de dados até então era algo que a equipe tinha em mente que deveria realizar, uma vez que os dados corporativos da empresa seriam analisados em prol do melhoramento do negócio de alguma maneira, mas a forma e para o que especificamente esse conhecimento seria aplicado ainda eram desconhecidos. Foi então que, com o auxílio da orientadora, que sugeriu a leitura do artigo solução de D'Haen e Poel, e por um novo requisito do dono da empresa ateliware, a equipe descobriu o que extrair da mineração. Segundo a empresa, seria interessante de algum modo saber quais lojistas investir para que se tornassem futuros clientes rentáveis a eles. Essa nova ideia daria início a próxima *Sprint*, onde os atributos relevantes seriam selecionados e as instâncias solicitadas a empresa.

O funil de vendas da empresa também precisava ser estudado e compreendido, uma vez que deveria se entender a nomenclatura das etapas que eles usam, assim como estavam divididos os *suspects*, *leads* e *prospects* para eles. Esta análise é descrita no Capítulo 2.4 deste documento.

A arquitetura desenvolvida nesta etapa será explicada no Capítulo 4, correspondente ao próximo capítulo desse trabalho.

### 3.3.3.1 Histórias de Usuário

Em um processo tradicional de desenvolvimento de software, segundo Smith<sup>2</sup> (2009 apud Aguiar, s.d.), o analista identifica e descreve todos os requisitos do projeto. Já em um processo ágil, apenas o suficiente é descrito, para determinar o que será construído e repassado ao cliente, e é feito pela equipe, com o objetivo de agilizar o início do projeto.

No *Scrum* geralmente são usadas histórias de usuários (*user stories*), que servem para “descrever funcionalidades que serão valiosas tanto para o cliente ou para o sistema” (COHN, 2004, p. 4). Elas são compostas por 3 aspectos: uma descrição da história usada para o planejamento como um lembrete; conversações sobre a história que servem para clarear detalhes sobre a mesma; testes para seguir e detalhes de documentos que podem ser usados para determinar quando a história está completa.

A criação de uma história deve ser feita com o acrônimo INVEST, para se garantir uma boa qualidade para a equipe:

- I corresponde a independente, afirmando que uma boa história não deve depender de outra;
- N significa negociável, já que um requisito não é uma coisa estática e está em constante mudança;
- V responde a valor, onde é necessário adicionar um valor a criação da história, não apenas algo divertido para o momento;
- E corresponde a estimável, onde se há a ideia de que se a história não contém informações suficientes para se conseguir estimar o tempo necessário para se desenvolver, ela não pode ser iniciada.

---

<sup>2</sup> SMITH, G.; SIDKI, A. **Becoming Agile...** in an imperfect world. Greenwich: Manning Publications Co., 2009.

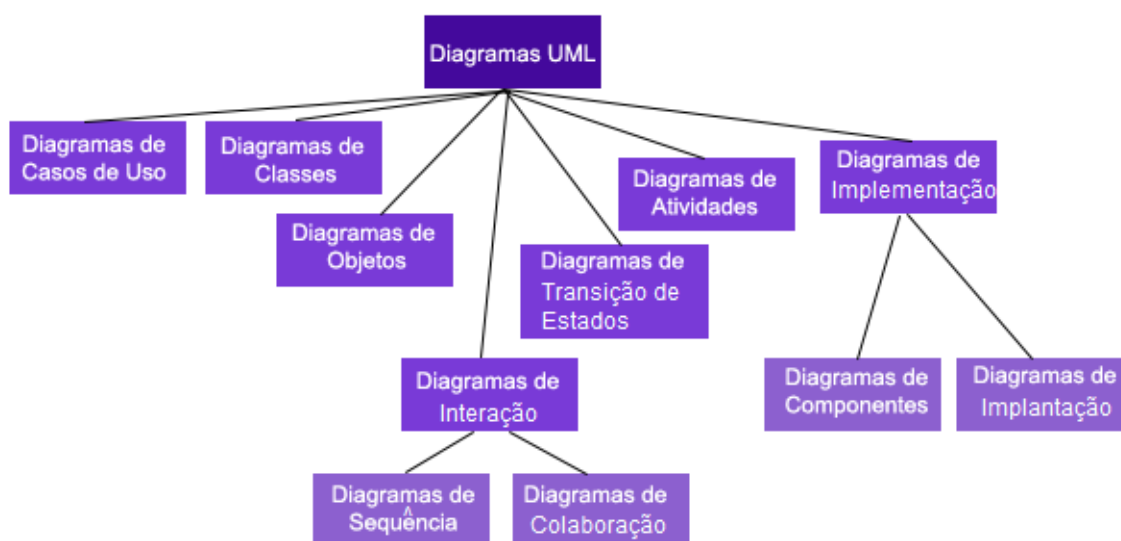
- S para pequena (*small*), definindo que a história deve ser pequena o suficiente para caber em um cartão;
- E o T, que significa testável. Se uma história não pode ser testada por falta de informação, ela não pode ser colocada no *backlog* (BERNARDO, 2014).

Neste presente projeto utilizou-se histórias de usuários para definir os requisitos, os quais podem ser encontradas no Apêndice A.

### 3.3.3.2 Diagrama de Caso de Uso

A utilização da linguagem UML como suporte à modelagem de um sistema envolve a criação de diversos documentos, uma vez que cada diagrama desenvolvido mostra uma visão diferente do sistema para o colaborador. Pode-se usar a seguinte analogia: imagine que você possui uma coleção de carrinhos, feitos em modelos tridimensionais para serem analisados por diferentes perspectivas (por cima, por dento, entre outros). Não é necessário mais de um carrinho para isso ser possível. Já os diagramas apenas existem no universo bidimensional. Para compensar a falta de uma dimensão, são desenvolvidos vários diagramas com perspectivas diferentes do sistema, como mostrado na Figura 10 abaixo (BEZERRA, 2006, p. 16-17).

FIGURA 10 - DIAGRAMAS DO UML



Fonte: Adaptado de BEZERRA (2006).

Um desses documentos é o Diagrama de Casos de Uso, que corresponde a uma visão de fora do sistema, representando graficamente os atores, casos de uso e relacionamentos entre eles. Ele tem a funcionalidade de ilustrar quais elementos externos interagem com o sistema e suas funcionalidades (BEZERRA, 2006, p. 57).

O DCU do Usuário pode ser encontrado no Apêndice B, seguido das suas seguintes descrições que vão do Apêndice C ao Apêndice G.

### 3.3.3.3 Diagrama de Classes

O Diagrama de Classes é responsável em construir um modelo das classes usadas no sistema. É o diagrama mais rico em termos de notação, considerando que nele são mostrados os nomes das classes, lista de atributos e a lista de operações das mesmas (BEZERRA, 2006, p.97).

O iMine, em consideração ao tamanho de seu escopo, não utiliza um grande número de classes. Como é dividido em duas aplicações, ele utiliza classes para o *gateway* e algumas para o *data mining*. O Diagrama de Classes deste sistema pode ser encontrado no Apêndice H.

### 3.3.3.4 Diagrama de Sequência

O Diagrama de Sequência enfatiza a troca de mensagens dentro de uma linha de tempo sequencial, exibindo as mensagens ordenadas no tempo (BEZERRA, 2006, p.148 e 156).

Neste trabalho, os Diagramas de Sequência da tela dos indicadores gerais e da tela dos gráficos podem ser visualizados nos Apêndice I ao Apêndice M, no final deste relatório.

### 3.3.4 *Sprint 4*

Essa *Sprint* foi responsável em definir os atributos do cliente para a clusterização, finalização do escopo do projeto e planejamento detalhado das *Sprints*.

Com o assunto da mineração de dados em pauta, agora era possível escolher as variáveis relevantes, no modelo de dados da minestore, que possivelmente traria o perfil dos futuros clientes através do processo de *data mining*. Para isso o grupo criou uma lista com os atributos chave a serem analisados, seguidos de uma breve explicação do que a equipe achava conter naquele campo, já que a seleção havia ocorrido somente pelos nomes das colunas do banco de dados. Essa lista foi então apresentada à analista comercial da minestore, que sanou as dúvidas da equipe na reunião seguinte.

O escopo final do sistema e a elaboração do planejamento das *Sprints* com detalhamento também foram realizados nessa etapa, já que era preciso saber o quanto antes como seria distribuído o desenvolvimento do *software* ao longo do final do semestre e também no que a equipe deveria se focar em cada momento.

### 3.3.5 *Sprint 5*

Na *Sprint 5* houve uma reunião com o cliente, a definição da arquitetura detalhada da mineração de dados, as definições das tecnologias que seriam utilizadas e a lista de atributos para a mineração. Também se iniciou o desenvolvimento do gateway nessa iteração. Essa lista de variáveis pode ser visualizada no Apêndice N.

Após a reunião com a empresa, agora com os atributos e suas definições, foi solicitada à empresa uma tabela com pelo menos 10.000 clientes para se iniciarem os testes de clusterização, os quais foram entregues na *Sprint* seguinte. A arquitetura da mineração de dados também foi criada, no intuito de gerar uma fácil visualização de todo o processamento que acontecia nela. A partir daqui já se era pensado na próxima *Sprint*, onde a elaboração do primeiro pré-processamento de dados se iniciaria.

Não menos importante, o desenvolvimento da tela de indicadores gerais do sistema também foi iniciado. É importante observar que foi nessa iteração em que o sistema começou a tomar forma e começar a ser programado, em contrapartida com todas as análises, estudos e reuniões que haviam sido feitos até o momento.

Tanto a arquitetura da mineração de dados e a elaboração do gateway da aplicação possuem uma explicação dedicada no Capítulo 4 desse documento.

### 3.3.5.1 Tecnologias utilizadas

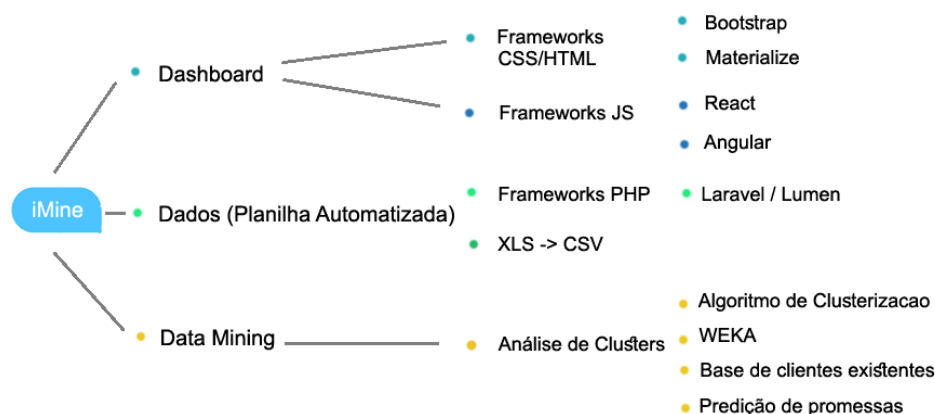
Para a escolha das tecnologias que seriam usadas no trabalho foram levados em conta três fatores cruciais: tempo disponível para o desenvolvimento, custo de utilização para a equipe e abrangência de mercado.

O primeiro fator foi levantado por razão deste projeto ser um sistema desenvolvido para um propósito acadêmico, apesar de ser elaborado para uma empresa real. Desse modo, o tempo de desenvolvimento foi bastante importante, pois o período delimitado foi de quatro meses, e o escopo planejado deveria agradar tanto a empresa quanto a instituição de ensino.

O custo de utilização também foi planejado com atenção, já que o tempo era escasso e a equipe não poderia ter o luxo de usar muitas ferramentas ou tecnologias desconhecidas, que demandariam certo tempo de aprendizado. Esse é um dos motivos da tecnologia *React*, que seria utilizada para o *front-end* da aplicação, ser rejeitada, já que o estudo da mineração de dados demandaria muito do tempo disponível da equipe. Já sobre a abrangência de mercado, todas as tecnologias utilizadas estão em alta, mantendo a aplicação nos padrões de competitividade com os sistemas que vêm sendo desenvolvidos atualmente. A Figura 11 mostra as tecnologias idealizadas para essa aplicação.

De todas essas tecnologias, as que foram usadas no desenvolvimento desta aplicação podem ser vistas de maneira resumida na Tabela 2.

FIGURA 11 - OPÇÕES DE TECNOLOGIAS



Legenda: Relação com as tecnologias e *frameworks* (biblioteca de classes que suportam uma funcionalidade) idealizados para a criação do iMine. Nem todos foram escolhidos, assim como algumas tecnologias que foram usadas não estão presentes. Fonte: Os Autores (2017).

TABELA 2 - TECNOLOGIAS DO IMINE

Tecnologia	Versão	Utilidade
Laravel Framework (PHP)	5.4	Framework baseado na linguagem de programação PHP, utilizado para <i>back-end</i> no desenvolvimento das APIs de métricas e carga inicial do data mining.
MySQL	5.7	Banco de dados.
Twitter Bootstrap	3.3.7	Framework CSS para <i>front-end</i> , com <i>design</i> responsível.
Trello	-	Quadro de atividades.
Docker	17.03.1-ce	Empacotador de aplicação e suas dependências em um recipiente virtual que pode ser executado em qualquer servidor Linux.
WEKA	3.8.0	Coleção de algoritmos de aprendizado de máquina para tarefas de <i>data mining</i> , podendo ser utilizada a partir de uma aplicação Java.
JAVA	1.7.0_80	Linguagem de programação para se comunicar com o WEKA.
Node.js	6.11	Plataforma para desenvolvimento de aplicações server-side baseadas em rede utilizando JavaScript.

Legenda: Tabela de tecnologias do iMine. Fonte: Os Autores (2017).

Por conta das necessidades trazidas pelo cliente, a aplicação iMine foi concebida com não apenas uma, mas duas funcionalidades principais: visualização

unificada de dados provenientes de diversas fontes, e *data mining* sobre uma fonte de dados específica. Desta forma, as tecnologias utilizadas são diversificadas, assim como os motivos que fizeram a equipe escolhê-las: algumas tecnologias foram selecionadas pois são padrões atuais de mercado, outras foram selecionadas pois eram a única alternativa disponível.

A primeira tecnologia escolhida foi o Laravel, *framework* PHP que utiliza a arquitetura MVC (*Model, View, Controller*) e tem como principal característica ajudar a desenvolver aplicações seguras e rápidas, com código limpo e simples, já que ele incentiva o uso de boas práticas de programação (ADRIEL, 2015). Ele foi escolhido para o *back-end* pela característica de ser um *framework* que trabalha com o PHP de baixo custo de utilização à equipe, que já estava familiarizada com a ferramenta, assim como com a tecnologia Bootstrap, usada no *front-end* da aplicação.

A respeito ao banco de dados, o MySQL foi o escolhido devido à vasta utilização no mercado e licença gratuita, além do baixo custo de utilização, novamente pela equipe já possuir certa familiaridade com as suas funcionalidades e interface. Sua fácil implementação e compatibilidade com várias linguagens de programações, as quais foram variadas nesse projeto, também foi um ponto forte na escolha.

O WEKA (*Waikato Environment for Knowledge Analysis*), que é um pacote de software de aprendizado de máquina escrito em Java, produzido pela universidade de Waikato, foi escolhido por ter se consolidado como a ferramenta de mineração de dados mais utilizada por estudantes e professores de universidades, sendo a ferramenta desse tipo que a equipe mais tinha contato em razão da grade acadêmica.

O Node.js, plataforma construída sobre o JavaScript do Google Chrome para facilmente mostrar aplicações rápidas e escaláveis, foi escolhido para a produção do primeiro ETL (*Extract-Transform-Load*) da mineração de dados. Segundo Rainardi (2008), ETL é um programa de extração, transformação e o carregamento que traz dados de várias fontes para uma área de testes. Esse programa tem a capacidade de se conectar com sistemas fontes de dados, lê-los, transformar esses dados e carregá-los em um outro local específico, já pré-processados (sem ruídos). O

Node.js foi escolhido por apresentar um baixo custo de utilização e a grande rapidez do JavaScript para enorme base de dados.

Dentre os casos de utilizar uma tecnologia obrigatória, pode-se citar a linguagem de programação Java, que é a linguagem orientada a objetos que possui uma biblioteca para interação com a ferramenta WEKA, utilizada para realizar a clusterização e o segundo ETL. No início, a ideia era utilizar o Node.js para os 2 ETLs, mas a interação necessária com o WEKA não permitiu a utilização da tecnologia no segundo processamento.

Ao longo do desenvolvimento, pela própria natureza da aplicação, algumas das tecnologias citadas adquiriram dependências de pacotes. A Tabela 3 a seguir mostra a lista resumida destas dependências e suas utilizações:

TABELA 3 - DEPENDÊNCIAS DE PACOTES

Pacote	Versão	Tecnologia	Utilização
laravel-job-status	0.1.7	Laravel	Gerenciamento de estado das tarefas de importação de arquivos CSV.
guzzlehttp	6.2.3	Laravel	Cliente HTTP para envio de requisições aos webservices.
body-parser	1.17.2	Node	Traduz o corpo da requisição para json.
express	4.15.3	Node	Servidor HTTP para atender as requisições ao webservice ETL 1.
mysql	2.13.0	Node	Interação com o banco de dados MySQL através do ETL 1.
nodemon	1.11.0	Node	Utilizado durante desenvolvimento, para monitoramento de mudanças no código.

Legenda: Tabela de dependências de pacotes das tecnologias do iMine. Fonte: Os Autores (2017).

### 3.3.6 Sprint 6

As *Sprints* 6 e 7, como pode ser observado, são as *Sprints* que demandaram mais esforço da equipe e, por essa razão, foram planejadas com um período maior para serem terminadas.

Os objetivos a serem alcançados nessa iteração eram o da apresentação da prototipação do sistema para a empresa e o início dos desenvolvimentos no *front-end* e na mineração de dados.

Após o desenvolvimento da prototipação do sistema, de acordo com o escopo já definido, uma apresentação foi realizada com a minestore, a qual aprovou o escopo do projeto da equipe.

Com as telas aprovadas, deu-se início ao desenvolvimento real das telas e a codificação do primeiro pré-processamento de dados para a mineração de dados, ou *ETL*. No caso de dados corporativos, como os quais são usados nesse projeto, um desenvolvimento de um ETL é essencial, pois os dados disponibilizados para a equipe apresentavam formatos inconsistentes de caracteres e muitos se encontravam *null* ou simplesmente vazios.

O primeiro obstáculo da equipe foi trabalhar com esses dados, que precisavam estar categorizados, sem nenhum numeral, para poder usá-los na clusterização, principalmente os campos com datas. Mais estudos e análises foram feitos com visto a esse empecilho. Foi então que, com o auxílio do grupo de pesquisa SOLID da UFPR, que estava familiarizado com situações semelhantes de temática *data mining*. A equipe começou a progredir, sanando as dúvidas que apareciam durante o desenvolvendo do primeiro ETL ao longo da *Sprint*. Conseqüentemente, um dicionário de dados foi implementado em conjunto com uma modelagem do banco de dados do sistema. Tanto o dicionário de dados quanto a modelagem de dados podem ser encontrados no final deste relatório, nos Apêndices O e P, respectivamente.

Ainda algumas outras dificuldades foram encontradas, relacionadas as colunas dos dados e instâncias nulas, que era o caso da tabela *store\_daily\_visits*, que deveriam ser categorizadas por meses, mas as datas eram de no máximo dois anos atrás (2015), sendo a mais recente de 2016. Além disso, metade das instâncias estavam vazias ou nulas. A tabela *store\_daily\_visits* foi descartada para esse projeto. Um outro problema existente foi relacionado ao atributo *confirmed\_at*, que corresponde a data que o lojista confirmou o *e-mail*, e que seria usado como a data principal para as comparações com outras datas relacionadas a loja, mas muitas das instâncias também estavam vazias, deixando impossível, por exemplo, descobrir a idade da loja. Mas isso foi resolvido com a solicitação à empresa de um novo atributo, o *created\_at*, que corresponde a data de criação da loja.

Logo, uma nova planilha de dados foi disponibilizada e o primeiro ETL finalizado. Dessa etapa se iniciaria a nova *Sprint* com os testes da clusterização e fim da implementação do sistema.

### 3.3.7 *Sprint 7*

Essa *Sprint* foi marcada pela realização dos testes de clusterização, estudo dos algoritmos de clusterização, finalização do *gateway*, *dashboards* e *data mining*, além da finalização da documentação do TCC.

Com o primeiro passo dado para a clusterização dos possíveis clientes, a equipe poderia dar início a fase de testes. Dois arquivos .csvs foram gerados: um com os *leads* e o outro com verdadeiros clientes. O campo de distinção entre os dois, no banco de dados, foi de faturamento. Caso houvesse algum valor, o lojista era considerado um cliente. Nessa divisão, foram identificados, dos 39 mil lojistas, quase 800 clientes em contrapartida com 38 mil *leads*. Alguns ajustes foram necessários no .csv de clientes para poder utilizá-lo no WEKA. Com o WEKA e o auxílio do grupo de pesquisas da UFPR, conseguiu-se salvar os clientes com o cluster ao qual pertencem por meio da clusterização, então para cada loja do arquivo de *leads*, verificar a qual cluster ele pertence.

O *gateway* também foi finalizado no começo desta *Sprint*.

Os *dashboards* foram terminados em paralelo com mineração de dados.

### 3.3.8 *Sprint 8*

A partir dessa *Sprint* houve uma diminuição no tempo das iterações novamente, considerando que os prazos finais do trabalho estavam bem próximos.

Nessa *Sprint* finalizou-se qualquer possibilidade de ajustes no sistema, pois a versão final foi corrigida e entregue. Após esse momento a equipe passou a preparar a apresentação do sistema e do trabalho à banca.

### 3.3.9 *Sprint 9*

A *Sprint 9* aconteceu em um momento em que o documento de TCC completo já havia sido devidamente entregue para a banca avaliadora.

O ciclo de *Sprints* previstas pelo escopo do trabalho de conclusão de curso é finalizado com a apresentação para a banca de avaliação. Entretanto, ainda existe mais uma interação nessa *Sprint*, que é a entrega do sistema ao cliente. Após a

avaliação da banca o sistema passará a ser utilizado pela empresa, que marcará previamente uma reunião de apresentação do iMine.

Ao fim desse capítulo, concluí-se a explicação da metodologia e materiais utilizados no projeto, demonstrando todas as etapas passadas para o desenvolvimento da solução e as dificuldades enfrentadas pelo grupo. No capítulo 4 será abordada a arquitetura detalhada do iMine e a apresentação do sistema.

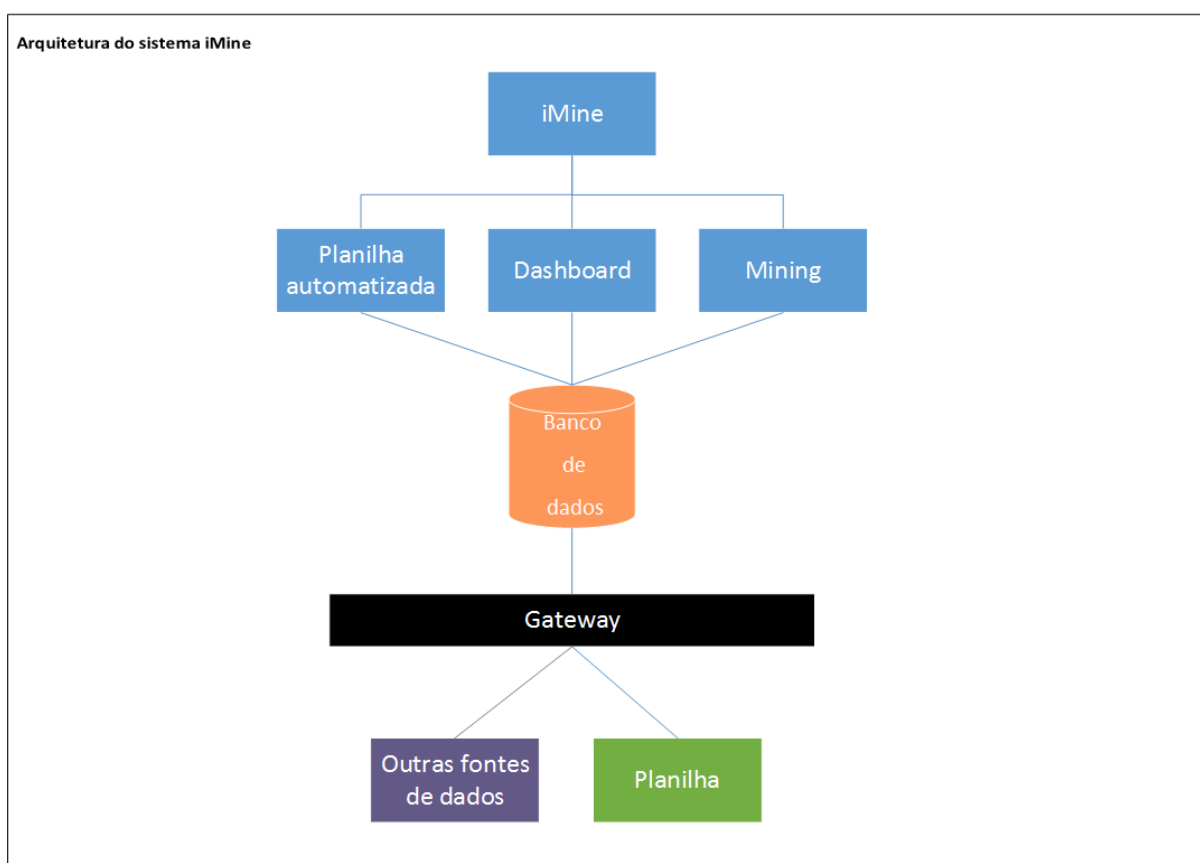
## 4 APRESENTAÇÃO DO SISTEMA

Nesse capítulo serão abordadas as arquiteturas do sistema iMine detalhadamente e a apresentação do sistema. A primeira parte detalha a estruturação enquanto a segunda apresenta o software, suas telas e funcionalidades.

### 4.1. ARQUITETURA DO SISTEMA

Na Figura 12 pode-se ver a arquitetura macro do sistema iMine. Como a aplicação é uma requisição de uso particular, para uso interno do cliente, a aquisição de um servidor para requisições *web* não foi necessária.

FIGURA 12 - ARQUITETURA DO IMINE



Legenda: Os retângulos azuis correspondem a estrutura do sistema iMine, que é composto pelas funcionalidades de uma planilha automatizada, *dashboards* e a mineração de dados. O retângulo preto corresponde ao gateway que faz comunicação com as fontes externas de dados sobre a minestore, com corpo JSON. O retângulo verde representa a planilha de indicadores disponibilizada pela empresa com alguns dados refletidos de seu banco, atualmente configurada para o gateway. O retângulo roxo representa as outras fontes de dados que devem possuir um formato JSON para serem aceitas pelo *gateway*. Fonte: Os Autores (2017).

Basicamente, a aplicação é dividida em duas partes: a primeira envolve a planilha automatizada e os gráficos, construídos em cima de um *gateway* de dados, enquanto a outra é um conjunto de subprogramas desenvolvidos para o *data mining*. Para unir todas as aplicações, no *back-end*, o iMine utiliza uma arquitetura do tipo MVC composta por três tipos de objetos: o modelo é o objeto de aplicação, a vista é a apresentação na tela e o controlador define a maneira como a interface do usuário reage às entradas do mesmo (GAMMA<sup>3</sup> et al. 2000 apud BAPTISTELLA, s.d.).

Tanto os processos do *gateway* quanto da mineração de dados serão abordados detalhadamente nos tópicos a seguir.

#### 4.1.1 Gateway

Procurando simplificar a importação de métricas relevantes para a análise de indicadores da minestore, a aplicação iMine conta com um *gateway* desenvolvido como API (Interface de Programação de Aplicações), possibilitando interoperabilidade entre aplicações. Esta API traz meios de comunicação com o banco de dados da aplicação iMine, possibilitando a inserção de dados para todos objetos utilizados nas funcionalidades de planilha automatizada e gráficos.

Desta forma, todas as fontes de dados (consultas SQL, Google Analytics, etc) podem ter suas respectivas métricas inseridas ao sistema de forma consolidada. A empresa poderá, então, desenvolver aplicações especializadas/microserviços que realizem de um lado a obtenção de dados (por exemplo através da API disponibilizada pelo Google Analytics) e de outro lado a inserção no sistema iMine (por meio da API mencionada). A Figura 13 apresenta o detalhamento do funcionamento do gateway.

---

<sup>3</sup> GAMMA, Erich et al. **Padrões de Projeto**: Soluções reutilizáveis de software Orientado a Objetos. Porto Alegre: Bookman, 2000.

FIGURA 13 - AQUITETURA DO GATEWAY



Legenda: A planilha .csv é a fonte atual de dados. Outras fontes de dados são simbolizadas pelo arquivo JSON, que devem possuir esse formato para conseguir se comunicar com o *gateway*. Assim que os dados entram no gateway desenvolvido na ParseController, existe o pré-processamento dos mesmos (transformação para *arrays*), então a verificação do tipo de métrica que ele apresenta: se é uma métrica já existente ou se é uma nova. Caso ela já exista, as semanas novas são adicionadas. Se forem novas métricas, o *gateway* registra as novas métricas e então registra as novas semanas. Em seguida, para ambas as opções, os valores das métricas são salvos no banco de dados. Fonte: Os Autores (2017).

Como já comentado, o *gateway* é usado apenas na hora da inserção de dados novos no banco. Ou seja, quando o usuário acessa as telas de indicadores e de gráficos, sem realizar uma nova importação de dados, o *gateway* não é acionado, apenas os dados do banco são requisitados. Esse componente encontra-se implementado como a controladora ParseController no sistema e consome requisições contendo corpo JSON, utilizando métodos de padrão REST para seu funcionamento. Atualmente, o *gateway* está adaptado para receber planilhas .csv, que são pré-processadas, transformadas em *array* e inseridas no banco. Caso a minestore queira alimentar o *gateway* com outra fonte de dados, eles deverão usar a API desenvolvida, implementando uma aplicação cliente que envie JSON via *post* para o iMine, o qual será recebido e processado pelo *gateway* genérico. O formato do JSON pode ser visto na Figura 14.

FIGURA 14 - FORMATO DO JSON CONSUMIDO PELO GATEWAY

```

▼ [
  ▼ {
    "name": "Lojas ativas",
    "slug": "lojas_ativas",
    "categories": "Financeiro período",
    "description": "Número de lojas ativas",
    ▶ "values": [...] // 61 items
  },
  ▼ {
    "name": "Checkout 1 (Nome/Entrega)",
    "slug": "checkout1_nomeentrega",
    "categories": "Financeiro período",
    "description": "Usuários que clicaram em comprar e informaram nome",
    ▼ "values": [
      ▼ {
        "week": "S82",
        "value": "623"
      },
      ▼ {
        "week": "S83",
        "value": "841"
      },
      ▼ {
        "week": "S84",
        "value": "819"
      },
      ▼ {
        "week": "S85",
        "value": "831"
      },
      ▼ {
        "week": "S86",
        "value": "943"
      },
    ],
  },
]

```

Legenda: O corpo do JSON é composto por um *array* contendo objetos JSON com os atributos *name*, *slug*, *categories*, *description* e *values*. O atributo *values* é também um *array* JSON, contendo objetos JSON, desta vez com atributos *week* e *value*. Fonte: Os Autores (2017).

O processo de inserção de dados por meio da API pode ser realizado com a implementação de um cliente HTTP que utilize os padrões REST, enviando requisições contendo um corpo JSON. A Tabela 4 a seguir lista os métodos disponíveis em sua API:

TABELA 4 - MÉTODOS DISPONÍVEIS NO GATEWAY

#	Endereço	Função	Objetos envolvidos	Requisição
1	/metrics	Listar as métricas existentes	Metric	GET
2	/metric/{slug}	Listar a métrica correspondente ao slug solicitado, com seus valores.	Metric, MetricValue, Week	GET
3	/metrics/json/{id}	Listar os valores da métrica correspondente ao ID solicitado	Metric, MetricValue, Week	GET
4	/weeks	Listar as semanas existentes	Week	GET
5	/weeks/{name}	Listar valores (e respectivas métricas) para a semana correspondente ao nome solicitado.	Week, MetricValue, Metric	GET
6	/metrics	Inserir nova métrica	Metric	POST
7	/weeks	Inserir nova semana	Week	POST
8	/metric/values	Inserir novo valor de métrica	MetricValue, Metric, Week	POST

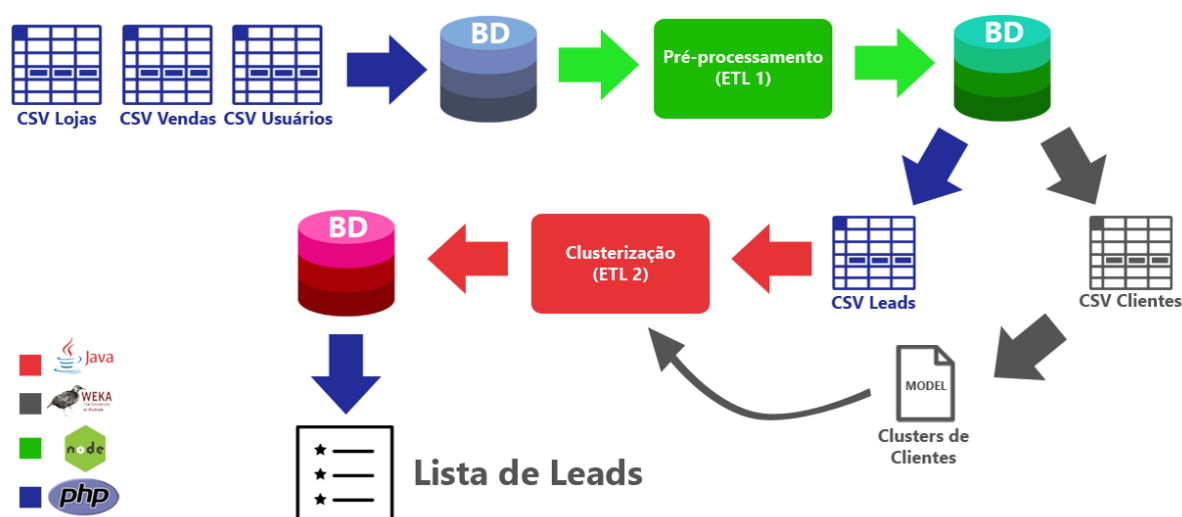
Fonte: Os Autores (2017).

Tendo em vista que atualmente a visualização de dados pela equipe da minestore baseia-se na utilização de planilhas, a aplicação inclui interface nativa para a importação de arquivos CSV (comma-separated values) gerados a partir de planilhas. Desta forma, torna-se mais fácil a adaptação para utilização da aplicação iMine por parte da minestore.

#### 4.1.2 Mineração de dados

O processo de mineração de dados desse projeto, para a descoberta de potenciais clientes para a empresa minestore, conta com uma série de processos, que são abordados na Figura 15.

FIGURA 15 - ARQUITETURA DATA MINING



Legenda: Arquitetura do processo de mineração de dados da aplicação iMine. Os artefatos em formato de planilha correspondem a arquivos .csv; as bases de dados da aplicação iMine; os retângulos arredondados aos subprogramas de processamento de dados; o caminho em cinza (flechas, planilha e model) ao processamento no programa WEKA; a lista corresponde ao resultado de todos os processos, que é uma lista de *leads*. Fonte: Os Autores (2017).

O processo de mineração de dados começa quando o usuário fornece ao sistema, por meio de *upload*, três arquivos .csv com os seguintes formatos:

- O .csv de lojas (stores.csv) no formato *id, is\_company, created\_at, published\_at, already\_have\_ecommerce, company\_segment, metrics, responsible\_gender, responsible\_birthdate*;
- O .csv de vendas das lojas (store\_sales.csv), no formato *id, faturamento, pedidos*;

- O .csv de usuários (users.csv), no formato *store\_id, confirmed\_at, current\_sign\_in\_at, last\_sign\_in\_at, sign\_in\_count*.

Assim que esses arquivos são adicionados ao sistema, o iMine é responsável em separar todos os campos pelos delimitadores ( , ou ; ) e salvar as informações no banco de dados, nas respectivas tabelas *stores, store\_sales* e *users*.

Dessa forma, os dados no formato que a empresa disponibiliza são mantidos em persistência e de fácil acesso a todos os processos do projeto.

A partir daí, com os dados armazenados, é possível dar continuidade para a próximo passo, que é o pré-processamento de dados (ETL 1), onde é feita a limpeza de ruídos dos dados, retiradas as inconsistências, o campo *metrics* é dividido em colunas e é realizada a categorização de todos os dados, preparando-os para a clusterização.

Essa etapa de pré-processamento de dados é de extrema importância pelo fato de que o algoritmo escolhido não funciona com atributos não categorizados. Por exemplo, datas e números foram categorizados de acordo intervalos de tempo. Se o número fosse 80, ele seria categorizado em “Entre 50 e 100”, entre outros.

O algoritmo utilizado para clusterização foi o SimpleKMeans. Este algoritmo é um dos mais simples existentes para realização de clusterização e foi utilizado com a base de clientes para agrupá-los por semelhança. Ele realiza o particionamento de um conjunto de dados em um pequeno número de *clusters*, utilizando observações através de variáveis discretizadas. O usuário deve especificar o número de *clusters* a serem utilizados, e então um centróide<sup>4</sup> *k* é encontrado para cada *cluster*. Os centróides devem idealmente estar localizados com a maior distância possível uns dos outros, uma vez que diferentes posições trazem diferentes resultados. Em seguida, cada ponto no conjunto de dados é associado ao centróide mais próximo, e ao fim, novos centróides são calculados com base nos baricentros dos clusters e os pontos são reassociados, e assim sucessivamente até que os centróides não mudem. Essa execução foi feita algumas

---

<sup>4</sup> Ponto ou coordenada de uma forma geométrica que estabelece o seu centro geométrico. Igual baricentro. (PRIBERAM, s.d.).

vezes, etapa geralmente necessária para avaliar o resultado obtido com diferentes valores de *clusters* nas opções do algoritmo. (MATTEUCCI, s.d.).

Após o pré-processamento, os dados são inseridos na tabela\_mae. Com os dados dessa forma já é possível seguir para a próxima etapa, que é a criação de dois arquivos .csv, separando os *leads* dos clientes. A separação foi feita por meio de duas *queries* no banco, onde a condição a ser satisfeita era a de que em um arquivo ficariam todos os lojistas que possuíam algum faturamento (clientes) e no outro os outros que não tinham faturamento algum.

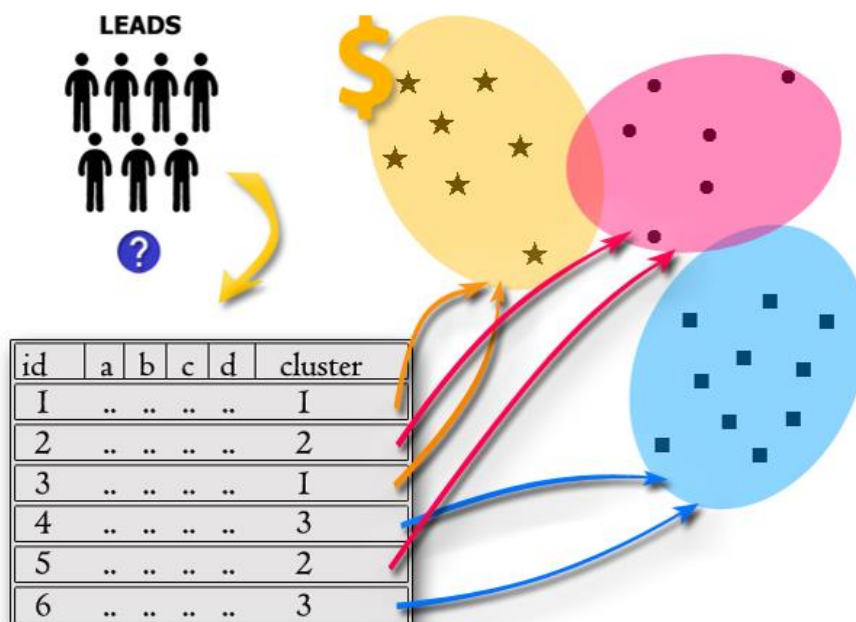
O arquivo de clientes foi então clusterizado no programa WEKA, o qual agrupou em 5 *clusters* (0 a 4) os diferentes perfis de clientes da minestore. Foram escolhidos 5 *clusters* pois não foram percebidas diferenças relevantes com números maiores - os novos agrupamentos possuíam características com extrema similaridade a grupos já existentes anteriormente. Analizando os resultados, notou-se que, dos 5 *clusters*, o *cluster* 1 obteve destaque pois se diferenciou de todos os outros no atributo faturamento, que é a característica em evidência mais importante para descobrir um perfil de um bom cliente. Esse *cluster* 1 recebeu a menor quantidade de registros (23 clientes de um total de 793) e foi escolhido para a comparação *lead a lead* (Figura 16). Sendo assim, todos os *leads* que posteriormente foram clusterizados para o *cluster* 1 foram classificados como potenciais clientes, descoberta que auxiliaria na próxima etapa do processo.

Como essa clusterização de clientes é realizada no programa WEKA, ela precisava ser acessível de alguma maneira para poder ser usada com o arquivo de *leads*. Por isso, o arquivo .model, gerado pelo programa após a clusterização, foi utilizado na etapa seguinte, junto do segundo ETL desenvolvido obrigatoriamente na linguagem Java, em função da utilização desse arquivo modelo.

Nesse ponto da mineração de dados definiu-se uma maneira de utilizar o arquivo modelo da clusterização dos clientes reais no WEKA com os potenciais clientes no arquivo de *leads*. Isso foi alcançado por meio da elaboração de um segundo ETL, onde o enfoque seria transformar o arquivo *leads.csv* em um arquivo do tipo .arff, para ser compatível com o arquivo modelo, e então cruzá-lo com o .model, descobrindo a que *cluster* cada *lead* pertenceria. Os que pertencessem ao

cluster 1 seriam os potenciais clientes, já que teriam mais características semelhantes aos dos clientes reais da loja.

FIGURA 16 - COMPARAÇÃO LEAD A LEAD



Legenda: O grupo de leads reflete registros do banco de dados que não possuem faturamento. Eles são comparados um a um com os *clusters* de clientes do arquivo modelo. Fonte: Os Autores (2017).

Após rodar o ETL 2, na etapa de pós-processamento de dados, uma lista de *ids* de lojas foi encontrada. Esta lista é a mostrada na tela “Promessas” pela aplicação iMine. Dessa forma, o processo de mineração de dados é terminado, e pode ser refeito quando necessário pela empresa a qualquer momento. O usuário somente precisa fazer o carregamento das planilhas iniciais e esperar o processo terminar.

A geração do arquivo modelo é realizada apenas uma vez manualmente no WEKA, já que a base de dados de clientes reais deverá sofrer uma mudança considerável de clientes para atrapalhar as predições. Por essa razão, o único componente não automatizado foi o de geração do arquivo modelo por meio do programa WEKA.

## 4.2. APRESENTAÇÃO DO IMINE

O iMine conta com três interfaces no sistema, sendo a primeira tela a de visualizações de indicadores, a segunda de visualizações de gráficos e a terceira de geração de promessas (ou descoberta de potenciais clientes para a empresa).

Na primeira tela, como é mostrado na Figura 17, é possível visualizar os indicadores gerais da empresa em uma grande tabela, que pode ser percorrida na horizontal e vertical.

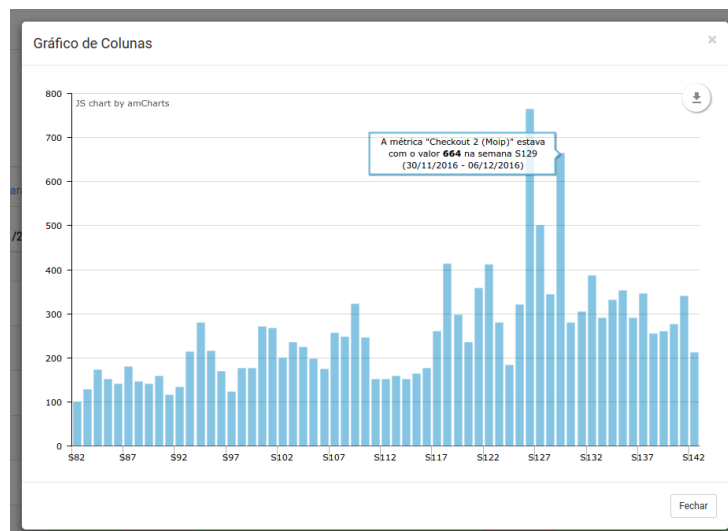
FIGURA 17 - TELA DE INDICADORES



Legenda: A cor vermelha na célula da tabela significa que o valor do indicador diminuiu em relação ao da semana anterior. O verde mostra que este indicador aumentou. Já a cor amarela mostra que o valor se manteve. Fonte: Os Autores (2017).

Ao clicar sobre um dos indicadores, um gráfico de linha é aberto, revelando a evolução daquele indicador ao longo do tempo, como é exemplificado na Figura 18 abaixo. Esse gráfico pode ser baixado nos formatos .png, .jpg, .svg ou .pdf e salvo para o computador do usuário nos formatos .csv, .json ou .xlsx.

FIGURA 18 - TELA DE GRÁFICO DE LINHA DO INDICADOR



Legenda: No gráfico ainda é possível ver uma mensagem detalhada. Fonte: Os Autores (2017).

Nessa tela ainda é possível realizar pesquisas de indicadores, como mostrado na Figura 19. A pesquisa funciona da seguinte forma: o usuário digita o nome do indicador desejado e, ao mesmo tempo, o sistema retorna os possíveis resultados existentes. Assim que encontrado, o indicador deve ser selecionado e o sistema fica responsável em trazer a métrica para a tela. No caso de nenhum resultado ser encontrado, nenhuma lista de resultados é montada.

FIGURA 19 - TELA DE PESQUISA DE INDICADOR

Dados

76 métricas  
61 semanas

Buscar métrica por nome

pedido

Pedidos Emitidos  
Pedidos Emitidos (Moip)  
Pedidos Emitidos (Bcash)

Pedidos Emitidos (Pagueseguro)

Pedidos Pagos  
Pedidos pagos (Moip)

Dica: clique sobre uma métrica na tabela para visualizar o gráfico de barras correspondente

Métrica	Categoria	S82 06/01/2016	S83 13/01/2016	S84 20/01/2016	S85 27/01/2016	S90 02/03/2016	S91 09/03/2016	S92 16/03/2016	S93 23/03/2016	S94 30/03/2016	S95 06/04/2016	S96 13/04/2016	S97 20/04/2016
Lojas ativas	N/A	181	186 +	189 +	194 +								
Checkout 1 (Nome/Entrega)	N/A	623	841 +	819 +	831 +	1037 +	1129 +	1482 +	0 +	0 +	1274 +	1359 +	
Checkout 2 (Nome/Entrega)	N/A	370	492 +	454 +	479 +	517 +	536 +	565 +	461 +	542 +	612 +	703 +	551 +
Checkout 2 (Moip)	Checkout 2 (Pagto/Entrega)	100	129 +	172 +	151 +	141 +	180 +	147 +	140 +	159 +	116 +	134 +	214 +
Checkout 2 (Bcash)	Checkout 2 (Pagto/Entrega)	144	134 +	100 +	109 +	152 +	229 +	173 +	107 +	126 +	210 +	160 +	122 +
Checkout 2 (Pagueseguro)	Checkout 2 (Pagto/Entrega)	126	229 +	182 +	219 +	224 +	127 +	245 +	214 +	257 +	286 +	409 +	215 +
Pedidos Emitidos	N/A	277	335 +	305 +	334 +	372 +	405 +	431 +	326 +	401 +	459 +	521 +	403 +
Pedidos Emitidos (Moip)	Pedidos Emitidos	80	89 +	126 +	112 +	106 +	131 +	112 +	106 +	120 +	93 +	101 +	154 +
Pedidos Emitidos (Bcash)	Pedidos Emitidos	116	95 +	70 +	91 +	121 +	197 +	141 +	79 +	97 +	170 +	133 +	95 +
Pedidos Emitidos (Pagueseguro)	Pedidos Emitidos	81	151 +	109 +	131 +	145 +	77 +	178 +	141 +	184 +	196 +	287 +	154 +

+ IMPORTAR DADOS

Fonte: Os Autores (2017).

A tela conta também com uma seleção de indicadores por meio da escolha de uma categoria de métricas específica, como mostrado na Figura 20. Para utilizá-la, o usuário deve apenas clicar sobre a categoria de indicadores desejados.

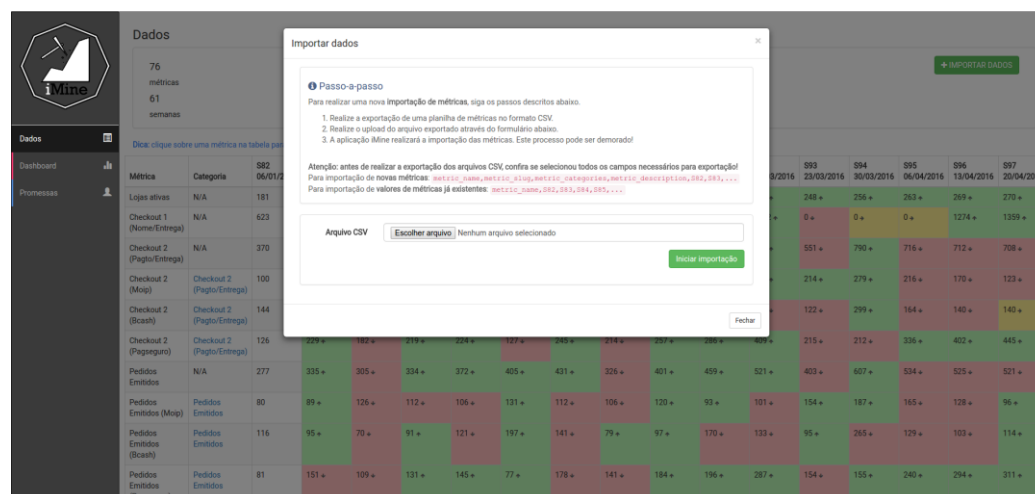
FIGURA 20 - TELA DE CATEGORIA DE INDICADOR



Fonte: Os Autores (2017).

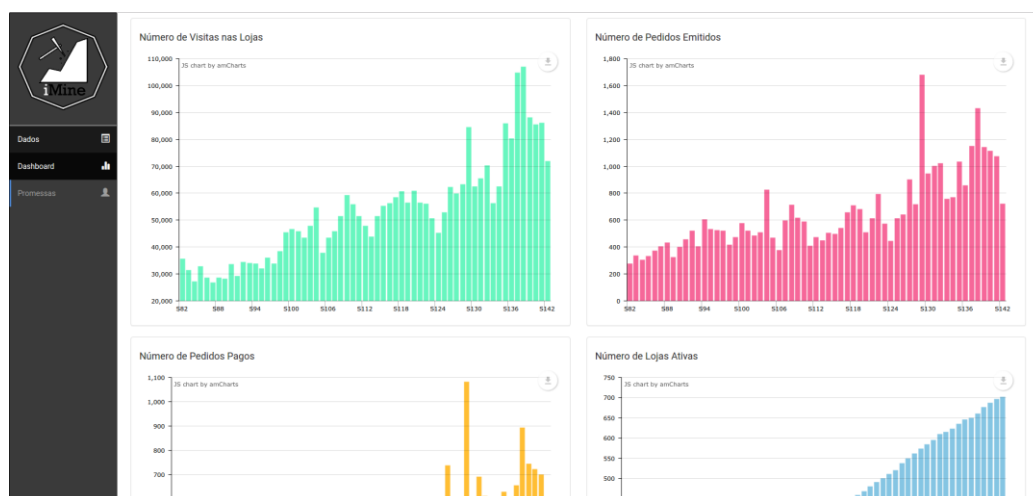
Por fim, mas não menos importante, o usuário é capaz de carregar uma nova planilha de indicadores gerais ao clicar no botão “Importar dados”, localizado no canto direito superior da tela, como demonstrado na Figura 21, e seguir as instruções apresentadas.

FIGURA 21 - TELA DE IMPORTAÇÃO DE DADOS



Fonte: Os Autores (2017).

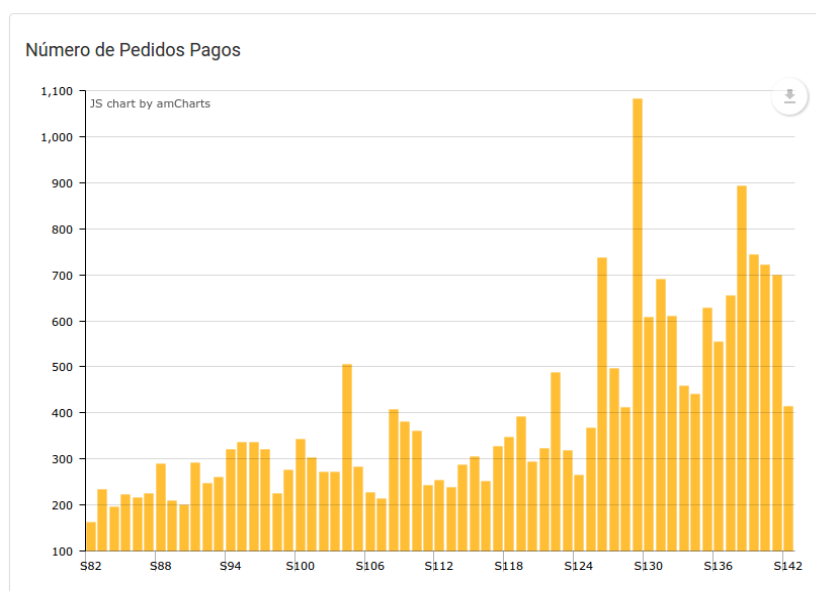
Na segunda tela do sistema, acessível por meio do segundo item do menu “Dashboard”, na lateral da aplicação, é possível encontrar os dados anteriores por meio de gráficos autoexplicativos, como mostrados na Figura 22. Todos os gráficos podem ser baixados ou salvados para o computador do usuário, nos formatos já citados anteriormente.

FIGURA 22 - TELA DE *DASHBOARD*

Fonte: Os Autores (2017).

Nessa tela encontra-se quatro gráficos diferentes. O primeiro a ser comentado é o de Número de Pedidos Pagos, que mostra a evolução na quantidade de pedidos pagos ao longo das semanas (Figura 23).

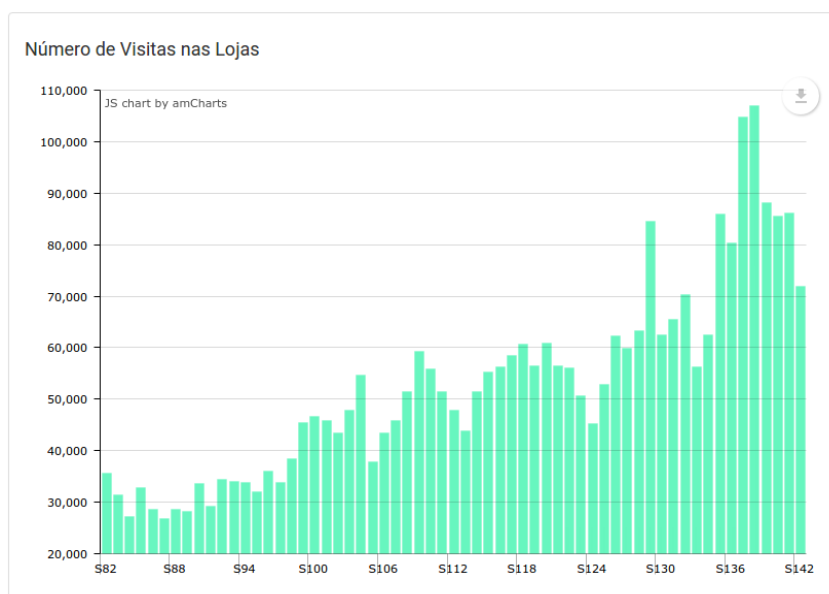
FIGURA 23 - GRÁFICO DE NÚMERO DE PEDIDOS



Fonte: Os Autores (2017).

Outro gráfico presente é o de Número de Visitas nas Lojas, que mostra a evolução na quantidade de visitas nas lojas ao longo das semanas (Figura 24).

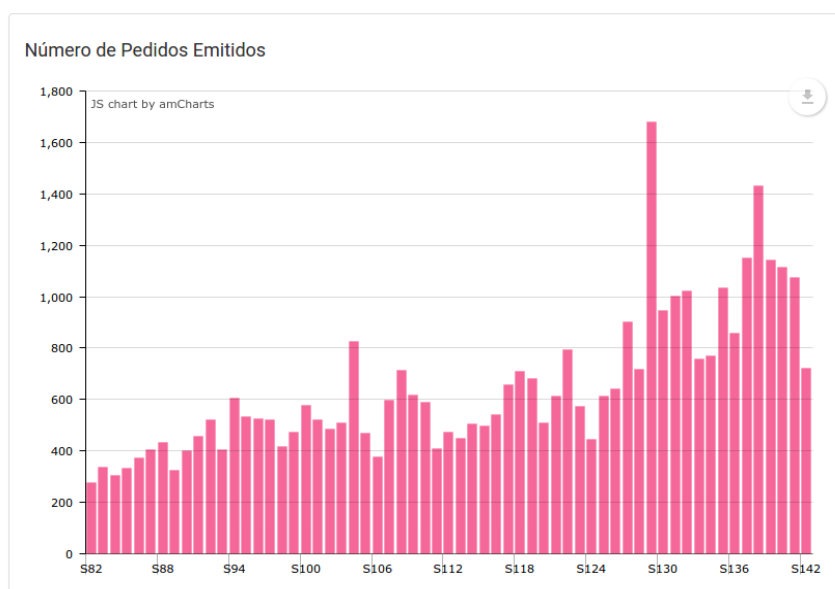
FIGURA 24 - GRÁFICO DE NÚMERO DE VISITAS NAS LOJAS



Fonte: Os Autores (2017).

Também é possível encontrar o gráfico de Número de Pedidos Emitidos, representando a quantidade de pedidos emitidos em todas as lojas da minestore ao longo das semanas (Figura 25).

FIGURA 25 - GRÁFICO DE NÚMERO DE PEDIDOS EMITIDOS



Fonte: Os Autores (2017).



Na Figura 28 pode-se ver a tela de aguardo do processo de mineração. A evolução das etapas pode ser observada por meio das barras azuis de progresso. No fim, uma lista com os potenciais clientes é mostrada. É importante destacar que o usuário, após realizar o processo de descoberta de clientes pelo menos uma vez, sempre terá acesso a última lista de potenciais clientes, podendo visualizá-la como mostrado na Figura 29.


FIGURA 28 - TELA DE PROGRESSO DA MINERAÇÃO

Tarefas de Importação Em Andamento			
Tarefa	CSV	Progresso Atual	Status
Importação de CSV Lojas	cut1.csv	100% (5297/5297)	Finalizado
Importação de CSV Vendas	minestore_ufpr_store_sales.csv	100% (838/838)	Finalizado
Importação de CSV Usuários	minestore_ufpr_users_export.csv	14%	Executando

[ATUALIZAR PROGRESSO](#)

Fonte: Os Autores (2017).

FIGURA 29 - TELA DE VISUALIZAÇÃO DE LEADS



Dados

Dashboard

Promessas

### Análise de Promessas

A aplicação **iMine** utiliza uma técnica de mineração de dados denominada **agrupamento k-means**, que possibilita realizar clusterizações.

1. Particionamos grupos de clientes da *minestore* que possuam características comuns;
2. A partir dos grupos gerados, é possível inferir um grupo de clientes que trazem bons resultados;
3. Por fim, analisamos a proximidade entre este grupo de clientes e novos clientes, gerando assim as **promessas!**

### Lista de Leads

ID da Loja	Segmento	Data de Criação	Data de Publicação
12	N/A	14/03/2014 18:44:54	20/05/2015 18:23:45
15	N/A	20/03/2014 19:18:56	20/05/2015 18:23:45
25	N/A	15/04/2014 16:48:48	20/05/2015 18:23:45
27	N/A	15/04/2014 16:50:15	20/05/2015 18:23:45
28	N/A	15/04/2014 16:52:16	20/05/2015 18:23:45
30	N/A	15/04/2014 17:03:43	20/05/2015 18:23:45
32	N/A	15/04/2014 17:07:08	20/05/2015 18:23:45
33	N/A	15/04/2014 17:07:11	20/05/2015 18:23:45
35	N/A	15/04/2014 17:20:55	20/05/2015 18:23:45
36	N/A	15/04/2014 17:26:45	20/05/2015 18:23:45

« Anterior [Próxima](#) »

[Realizar nova análise de promessas](#)

Legenda: Tela de resultado da última mineração de dados realizada no sistema, com sua respectiva lista de potenciais clientes. Caso o usuário queira realizar uma nova descoberta de promessas, ele pode clicar no botão azul “Realizar nova análise de promessas”. Fonte: Os Autores (2017).

Este capítulo apresentou da arquitetura do sistema e suas funcionalidades. No Capítulo 5, a seguir, serão apresentadas as considerações finais da equipe sobre o projeto e sua elaboração ao longo dos meses.

## 5 CONSIDERAÇÕES FINAIS

Este documento apresentou o desenvolvimento do software iMine. O projeto iniciou-se com base na necessidade da empresa em realizar uma consultoria aos dados da plataforma minestore e suas métricas de eventos, buscando uma solução para os seguintes problemas: a visualização de dados sobre os clientes e suas lojas, e a descoberta de potenciais *leads*. Perante esses problemas apresentados e as soluções desenvolvidas, o iMine está disponível para ser utilizado pela empresa ateliware no auxílio de tomadas de decisões do seu negócio de *e-commerces*.

O primeiro objetivo específico foi realizar a análise do negócio, etapa crucial para que a equipe pudesse entender as funcionalidades requeridas pelo cliente. A visualização otimizada das métricas e a elaboração de gráficos autoexplicativos, referentes aos indicadores gerais da minestore, foram completados, mesmo com as dificuldades de manipulações de dados e inconsistências existentes. Além disso, o desenvolvimento de um *gateway* genérico de comunicação que pudesse atender as necessidades da empresa também foi alcançado como o previsto.

A disponibilização de um algoritmo de descoberta de *leads*, que potencialmente se tornariam lojistas de sucesso, também foi feita com êxito. O processo de *data mining*, também adotado como o diferencial do sistema, desde o início ao fim, apresentou extrema dificuldade à equipe, que precisou que todos realizassem várias sessões de estudos e análises de dados para poder simplesmente iniciá-lo. Sendo o objetivo específico que mais demandaria o trabalho da equipe, essa funcionalidade foi acompanhada desde a *Sprint 1* até a última do ciclo, provando que a mensuração de sua dificuldade já era esperada.

De todo o tempo disponível para o projeto, mais de 50% foi dedicado ao pré-processamento dos dados corporativos, antes mesmo de se iniciar a interação com o algoritmo de clusterização, demonstrando a grande importância em se ter dados consistentes nas organizações para a realização de procedimentos de predições como o realizado.

Após a realização da mineração de dados, o resultado obtido foi uma grande quantidade de *leads*. Uma solução futura para filtrar essa lista seria a utilização de

um segundo algoritmo, conhecido como KNN (*K-Nearest Neighbor*), para ordenar as promessas por proximidade do cliente, classificando os melhores *leads*.

O último objetivo específico consiste da confecção do presente documento, em que todas as funcionalidades críticas foram descritas, sempre se baseando nas literaturas existentes e sobre as regras negociais colocadas pelo cliente. Além de documentar os processos e estratégias de desenvolvimento de software utilizados.

Mesmo com todos os objetivos atingidos, ainda identifica-se a possibilidade de implementação de novas funcionalidades, como:

- Multi-sessões de usuários com permissões de acesso ao sistema;
- A aplicação do algoritmo de KNN;
- A aplicação de um algoritmo de rede neural, supervisionado, para treinar um modelo de clientes afim de poder facilmente classificar *leads* de acordo com os resultados obtidos anteriormente;
- A possibilidade de integrar o WEKA com o sistema, para a criação do arquivo *model* da clusterização dos clientes, de forma embarcada, sem precisar fazer manualmente fora do iMine, já que o arquivo *model* é baseado nos clientes existentes na loja e tende a precisar ser gerado novamente quando houver uma mudança considerável nessa base.

Futuramente, há ainda a possibilidade de disponibilizar o sistema a outros clientes com negócios semelhantes. Caso isso ocorra, haverá a necessidade de modificar algumas implementações e personalizá-las para a nova empresa.

## REFERÊNCIAS

ADRIEL, W. **Introdução ao Laravel Framework PHP**, 2015. Disponível em: <<http://www.devmedia.com.br/introducao-ao-laravel-framework-php/33173>>. Acessado em: 18 jun. 2017.

AGUIAR, P. **COMO ESPECIFICAR REQUISITOS PARA DESENVOLVIMENTO DE SOFTWARE EM METODOLOGIAS ÁGEIS?**, 18 f. Dissertação (Pós graduação em Engenharia de Software Centrada em metodologias Ágeis) – Centro Universitário UNA.

BAPTISTELLA, J., A. **Abordando a arquitetura MVC, e Design Patterns: Observer, Composite, Strategy**. Disponível em: <<http://www.linhadecodigo.com.br/artigo/2367/abordando-a-arquitetura-mvc-e-design-patterns-observer-composite-strategy.aspx>>. Acesso em: 19 junho 2017.

BECK, et al. O manifesto Ágil. Disponível em: <<http://www.knowledge21.com.br/sobreagilidade/agilidade/o-manifesto-agil/>>. Acessado em: 15 jun. 2017.

BERNARDO, K. **Estória de usuário. Você saberia contar?** 2014. Disponível em: <<https://www.culturaagil.com.br/estoria-de-usuario-voce-saberia-contar/>>. Acessado em: 15 abr. 2017.

BEZERRA, E. **Princípios de Análise e Projeto em UML**, 3º tiragem. Editora Campus. 2006. Ebook.

BISSI, W. Scrum – **Metodologia de Desenvolvimento Ágil**, Campo Dig. Campo Mourao, v. 2, n. 1, p 3-6, 2007.

COHN, M. **USER STORIES APPLIED** For Agile Software Development. Publisher Addison-Wesley Professional. Mar 2004.

COMÉRCIO eletrônico no Brasil. Disponível em: <<https://www.b2wdigital.com/institucional/comercio-eletronico-no-brasil>>. Acesso em: 25 mar. 2017.

CONTENT, R. **O que é e como desenvolver um funil de vendas?** Disponível em: <<http://marketingdeconteudo.com/funil-de-vendas/>>. Acesso em: 02 mai. 2017.

CRM: o que é e como funciona? Disponível em: <<http://www.administradores.com.br/artigos/marketing/crm-o-que-e-crm-e-como-funciona/34063/>>. Acesso em: 18 abr. 2017.

CUSTOMER Relationship Management (CRM). Disponível em: <<http://www.gartner.com/it-glossary/customer-relationship-management-crm/>>. Acesso em: 19 abr. 2017.

D'HAEN J.; POEL **Model-supported business-to-business prospect prediction based on an iterative customer acquisition framework**. 7f, Faculty of Economics and Business Administration, Ghent University, Belgium, 2013.

ENDEAVOR, AGENDOR **O Funil de Vendas na Prática**, s.d. Ebook. Disponível em: <<http://info.endeavor.org.br/funil-vendas-pratica>>. Acessado em: 01 mai. 2017.

BERTHOLDO, 3 tendências do e-commerce para 2017. Disponível em: <<https://www.bertholdo.com.br/blog/tendencias-do-e-commerce-2017/>>. Acesso em: 19 jun. 2017.

FELDMAN, J. Computers and Thought - The Back Story. **Revista IEEE Annals of the History of Computing**. [on-line]. v. 33, n. 4, Abril 2011. Disponível em: <<http://ieeexplore.ieee.org/document/6096527/>> ISSN: 1058-6180. Acesso em 25 março 2017.

FUNIL de Marketing e Vendas: O Melhor Conteúdo Para Cada Etapa. Disponível em: <<http://blog.escoladomarketingdigital.com.br/funil-de-marketing-e-vendas-o-melhor-conteudo-para-cada-etapa/>>. Acesso em: 10 abr. 2017.

FUNIL de Vendas: qual a diferença entre suspect, prospect e lead. Disponível em: <<http://motordeconteudo.com.br/funil-de-vendas-diferenca-entre-suspect-prospect-lead/>>. Acesso em: 10 abr. 2017.

FUZZATO, M. A diferença entre suspect, prospect e lead qualificado. Disponível em: <<http://www.automacaodemarketing.com.br/gerenciamento-de-leads/a-diferenca-entre-suspect-prospect-e-lead-qualificado/>>. Acesso em: 10 abr. 2017.

HAN, J.; KAMBER, M. **Data Mining: Concepts and Techniques**. San Francisco, 2006.

MATTEUCCI, M. **K-Means Clustering**. Politecnico di Milano. Dipartimento di Elettronica Informazione e Bioingegneria. Disponível em: <[https://home.deib.polimi.it/matteucc/Clustering/tutorial\\_html/kmeans.html](https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html)>. Acessado em: 5 jul 2017.

MIKADO, H. **The Difference Between a Suspect, a Prospect, and a Lead**, 2016. Disponível em: <<https://www.linkedin.com/pulse/difference-between-suspect-prospect-lead-hawk-mikado>>. Acesso em: 10 jun. 2017.

MOURA, A. **Funil de vendas: como aumentar a conversãodo seu e-commerce**, 2016. Disponível em: <<https://moip.com.br/blog/funil-de-vendas-como-usar-para-aumentar-as-vendas-do-e-commerce/>>. Acessado em: 15 abr. 2017.

PAYNE A. **HANDBOOK OF CRM: Achieving Excellence in Customer Management**. London, England, 2005. Disponível em: <<http://testa.roberta.free.fr/My%20Books/Mainframe/jcl%20cobol%20tutorials/Handbook%20of%20CRM.pdf>>. Acessado em: 20 abr. 2017.

PEREIRA, V. **O que é e para que serve um funil de vendas?**, 2016. Disponível em: <<http://resultadosdigitais.com.br/blog/funil-de-vendas-o-que-e-como-funciona/>>. Acessado em: 21 abr. 2017.

PLATAFORMAS de e-commerce B2B. Disponível em: <<http://www.guiadeecommerce.com.br/plataformas-ecommerce-b2b/>>. Acesso em: 18 abr. 2017.

PRIBERAM dicionário “**Centróide**”. Disponível em: <<https://www.priberam.pt/dlpo/centr%C3%B3ide>>. Acessado em: 06 jul 2017.

RAINARDI, V. **Building a Data Warehouse** with examples in sql server. Publisher Apress. 2008.

SALVADOR, M. **Gerente de ecommerce**. São Paulo: Ecommerce School, 2013. Ebook. Disponível em: <<http://www.ecommerceschool.com.br/Livro-Ecommerce/Livro-Gerente-de-Ecommerce-Mauricio-Salvador.pdf>>. Acesso em: 25 mar. 2017.

SANTOS, F, R. **Escrevendo Estórias do Usuário Eficazes**, 2010. Disponível em: <<https://www.slideshare.net/Ridlo/escrevendo-estrias-do-usurio-eficazes>>. Acessado em: 10 jun. 2017.

SCHWABER, K.; BEEDLE, M. Agile Software Development with Scrum. **Prentice Hall**, 2002

TOTAL retail 2016. PWC. Disponível em: <[http://www.pwc.com.br/pt/setores-de-atividade/varejo-e-consumo/assets/2016/total\\_retail\\_16\\_brasil.pdf](http://www.pwc.com.br/pt/setores-de-atividade/varejo-e-consumo/assets/2016/total_retail_16_brasil.pdf)>. Acesso em: 25 de março.

VALLE, A. **Modelos de E-commerce**, 2016. Disponível em: <<http://www.guiadeecommerce.com.br/modelos-de-ecommerce/>>. Acessado em: 20 abr. 2017.

VIEIRA, D. **Scrum: A Metodologia Ágil Explicada de forma Definitiva**, 2014. Disponível em: <<http://www.mindmaster.com.br/scrum/>>. Acessado em: 1 jun. 2017.

## APÊNDICE A – HISTÓRIAS DE USUÁRIO

### Agrupamento de histórias sobre o tema “Indicadores gerais”

- Como usuário preciso visualizar os indicadores gerais da minestore por meio de uma planilha automatizada.
- Como usuário gostaria de não precisar “alimentar” os indicadores gerais manualmente. Um “gateway” poderia fazer isso!
- Sendo um usuário quero que a planilha automatizada seja dividida em semanas e por categorias importantes para que eu encontre mais fácil o que eu queira.
- Sendo um usuário gostaria que a planilha tivesse um campo de pesquisa rápida para eu encontrar um indicador específico mais rapidamente.
- Como usuário gostaria de poder ver um gráfico de linha em cada indicador, vendo sua evolução durante as semanas.
- Como usuário gostaria que fosse possível visualizar o número de pedidos pagos, em um gráfico por semanas.
- Como usuário gostaria que fosse possível visualizar o número das visitas nas lojas, em um gráfico por semanas.
- Como usuário gostaria que fosse possível visualizar o número de pedidos emitidos, em um gráfico por semanas.
- Como usuário gostaria que fosse possível visualizar o número de lojas ativas, em um gráfico por semanas.
- Eu como usuário preciso que os gráficos que aparecerão no dashboard sejam relevantes para mim.

### Agrupamento de histórias sobre o tema “Futuros clientes”

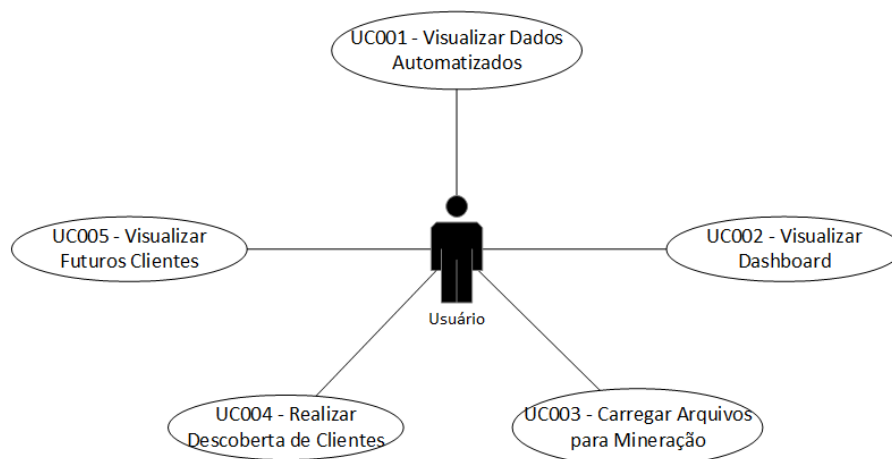
- Eu como usuário gostaria que fosse possível identificar os possíveis lojistas que tendem a se tornar futuros clientes para que eu saiba o grupo em que devo investir.
- Eu como usuário desejo visualizar os possíveis futuros clientes no sistema de forma simples.

## APÊNDICE B – DIAGRAMA DE CASOS DE USO

O iMine conta com cinco casos de uso relacionados ao ator Usuário do sistema, como é mostrado na Figura 30.

FIGURA 30 - DIAGRAMA DE CASOS DE USO RELACIONADO AO USUÁRIO

UC - iMine



Legenda: Diagrama de Casos de Uso relacionados ao ator Usuário. Fonte: Os Autores (2017).

## APÊNDICE C – DOCUMENTAÇÃO UC VISUALIZAR DADOS AUTOMATIZADOS

O caso de uso “Visualizar Dados Automatizados” pode ser observado detalhadamente na Tabela 5.

TABELA 5 - DOCUMENTAÇÃO UC VISUALIZAR DADOS AUTOMATIZADOS

<b>Nome do caso de uso</b>	Visualizar Dados Automatizados.
<b>Ator Principal</b>	Usuário.
<b>Resumo</b>	Esse caso de uso descreve as etapas do usuário para poder visualizar os dados automatizados da planilha de indicadores gerais.
<b>Pré-condições</b>	O usuário precisa acessar o sistema.
<b>Pós-condições</b>	O usuário terá acesso aos indicadores gerais da empresa.
<b>FLUXO PRINCIPAL</b>	
<ol style="list-style-type: none"> <li>1. O usuário acessa o sistema iMine.</li> <li>2. O sistema carrega todos os indicadores e suas semanas na tela.</li> <li>3. O usuário seleciona e clica no indicador da tabela que deseja visualizar com detalhe.</li> <li>4. O sistema mostra um gráfico de linha da evolução do indicador ao longo do tempo.</li> </ol>	
<b>FLUXO ALTERNATIVO – PESQUISAR INDICADOR</b>	
<ol style="list-style-type: none"> <li>1. O usuário escreve na caixa de texto de pesquisa o nome do indicador que deseja procurar.</li> <li>2. O sistema realiza a pesquisa ao mesmo tempo que o usuário digita.</li> <li>3. O sistema mostra uma lista de indicadores para o usuário escolher na tela.</li> <li>4. O sistema carrega a tabela com a métrica sozinha na tela.</li> </ol>	
<b>FLUXO ALTERNATIVO – AGRUPAR POR CATEGORIA</b>	
<ol style="list-style-type: none"> <li>1. O usuário clica em uma categoria de indicadores.</li> <li>2. O sistema mostra na tela apenas os indicadores daquela categoria.</li> </ol>	
<b>FLUXO ALTERNATIVO – CARREGAR NOVO ARQUIVO</b>	
<ol style="list-style-type: none"> <li>1. O usuário clica no botão “Importar dados”.</li> <li>2. O usuário seleciona o arquivo para carregar no sistema.</li> </ol>	
<b>FLUXO DE EXCEÇÃO – INDICADOR NÃO EXISTE</b>	
<ol style="list-style-type: none"> <li>1. O sistema não mostra nada na lista de resultados na pesquisa.</li> </ol>	
<b>FLUXO DE EXCEÇÃO – FORMATO DE ARQUIVO INVÁLIDO</b>	
<ol style="list-style-type: none"> <li>1. O sistema comunica o usuário que o tipo de arquivo não é compatível.</li> </ol>	

Legenda: Documentação do caso de uso “Visualizar Dados Automatizados”. Fonte: Os Autores (2017).

## APÊNDICE D – DOCUMENTAÇÃO UC VISUALIZAR *DASHBOARD*

O caso de uso “Visualizar *Dashboard*” pode ser observado detalhadamente na Tabela 6.

TABELA 6 - DOCUMENTAÇÃO UC VISUALIZAR *DASHBOARD*

<b>Nome do caso de uso</b>	Visualizar <i>Dashboard</i> .
<b>Ator Principal</b>	Usuário.
<b>Resumo</b>	Esse caso de uso descreve as etapas do usuário para poder visualizar os gráficos dos indicadores gerais no <i>dashboard</i> do sistema.
<b>Pré-condições</b>	O usuário precisa acessar o sistema.
<b>Pós-condições</b>	O usuário terá acesso aos gráficos dos indicadores gerais da empresa.
<b>FLUXO PRINCIPAL</b>	
<ol style="list-style-type: none"> <li>1. O usuário acessa o sistema iMine.</li> <li>2. O usuário clica no item do menu “<i>Dashboard</i>”.</li> <li>3. O sistema carrega todos os gráficos dos indicadores gerais na tela. São eles: gráfico de pedidos pagos, número de visitas nas lojas, número de pedidos emitidos e número de lojas ativas.</li> </ol>	
<b>FLUXO ALTERNATIVO (Não há fluxo alternativo)</b>	
<b>FLUXO DE EXCEÇÃO (Não há fluxo de exceção)</b>	

Legenda: Documentação do caso de uso “Visualizar *Dashboard*”. Fonte: Os Autores (2017).

## APÊNDICE E – DOCUMENTAÇÃO UC CARREGAR ARQUIVOS PARA A MINERAÇÃO

O caso de uso “Carregar Arquivos para a Mineração” pode ser observado detalhadamente na Tabela 7.

TABELA 7 - DOCUMENTAÇÃO UC CARREGAR ARQUIVOS DA MINERAÇÃO

<b>Nome do caso de uso</b>	Carregar Arquivos para a Mineração.
<b>Ator Principal</b>	Usuário.
<b>Resumo</b>	Esse caso de uso descreve as etapas do usuário para poder carregar os arquivos das lojas que serão utilizados na mineração de dados.
<b>Pré-condições</b>	O usuário precisa acessar o sistema.
<b>Pós-condições</b>	O usuário saberá os potenciais clientes para a empresa.
<b>FLUXO PRINCIPAL</b>	
<ol style="list-style-type: none"> <li>1. O usuário acessa o sistema iMine.</li> <li>2. O usuário clica no item do menu “Promessas”.</li> <li>3. O sistema mostra na tela informações sobre o formato dos arquivos permitidos e disponibiliza um local de seleção de arquivos para adicionar ao sistema.</li> <li>4. O usuário seleciona e clica no local de <i>upload</i> e seleciona os arquivos das lojas.</li> <li>5. O sistema comunica ao usuário que começou o pré-processamento daqueles arquivos para o banco de dados. Assim que o processo é terminado, o sistema direciona o usuário para a próxima tela a inicialização da mineração dos dados.</li> </ol>	
<b>FLUXO ALTERNATIVO – VISUALIZAR LISTA DE LEADS</b>	
<ol style="list-style-type: none"> <li>1. Caso uma mineração anterior de dados já tenha sido feita, o sistema mostra na tela a lista de antigos clientes em potencial existente no banco. Dessa forma, não é necessária realizar sempre o processo de descoberta de dados.</li> </ol>	
<b>FLUXO ALTERNATIVO – FAZER NOVA MINERAÇÃO DE DADOS</b>	
<ol style="list-style-type: none"> <li>1. O usuário clica no botão disponível junto da lista de potenciais clientes para realizar um novo processo de descoberta de clientes.</li> <li>2. O sistema redireciona o usuário a tela de carregamento de arquivos .csv, retornando ao UC “Carregar Arquivos para a Mineração”.</li> </ol>	
<b>FLUXO DE EXCEÇÃO – ARQUIVO NÃO SUPORTADO</b>	
<ol style="list-style-type: none"> <li>1. O sistema comunica ao usuário que não é possível processar o arquivo adicionado e pede para adicionar um arquivo dentro dos padrões requeridos.</li> </ol>	

Legenda: Documentação do caso de uso “Carregar Arquivos para a Mineração”. Fonte: Os Autores (2017).

## APÊNDICE F – DOCUMENTAÇÃO UC REALIZAR DESCOBERTA DE CLIENTES

O caso de uso “Realizar Descoberta de Clientes” pode ser observado detalhadamente na Tabela 8.

TABELA 8 - DOCUMENTAÇÃO UC REALIZAR DESCOBERTA DE CLIENTES

<b>Nome do caso de uso</b>	Realizar Descoberta de Clientes.
<b>Ator Principal</b>	Usuário.
<b>Resumo</b>	Esse caso de uso descreve as etapas do usuário para realizar o processo de descoberta de potenciais futuros clientes para empresa. Esse UC é chamado automaticamente após o término do UC “Carregar Arquivos para a Mineração”.
<b>Pré-condições</b>	O usuário precisa acessar o sistema. O usuário precisa ter realizado o caso de uso anterior, “Carregar Arquivos para a Mineração”.
<b>Pós-condições</b>	O usuário realizará a descoberta dos potenciais clientes.
<b>FLUXO PRINCIPAL</b>	
<ol style="list-style-type: none"> <li>1. O usuário é direcionado para a nova tela após o fim do UC “Carregar Arquivos para Mineração”.</li> <li>2. O usuário aguarda o término do processamento de dados até o final, podendo acompanhá-lo por meio das barras de progresso.</li> <li>3. O sistema salva no banco de dados os potenciais clientes descobertos durante o processamento.</li> <li>4. O sistema mostra uma lista de lojas potenciais para serem investidas pela empresa na tela.</li> </ol>	
<b>FLUXO ALTERNATIVO (Não há fluxo alternativo)</b>	
<b>FLUXO DE EXCEÇÃO – ERRO NO PROCESSO DE MINERAÇÃO</b>	
<ol style="list-style-type: none"> <li>1. O sistema comunica ao usuário que houve um erro durante o processamento de dados.</li> </ol>	

Legenda: Documentação do caso de uso “Visualizar Futuros Clientes”. Fonte: Os Autores (2017).

## APÊNDICE G – DOCUMENTAÇÃO UC VISUALIZAR FUTUROS CLIENTES

O caso de uso “Visualizar Futuros Clientes” pode ser observado detalhadamente na Tabela 9.

TABELA 9 - DOCUMENTAÇÃO UC VISUALIZAR FUTUROS CLIENTES

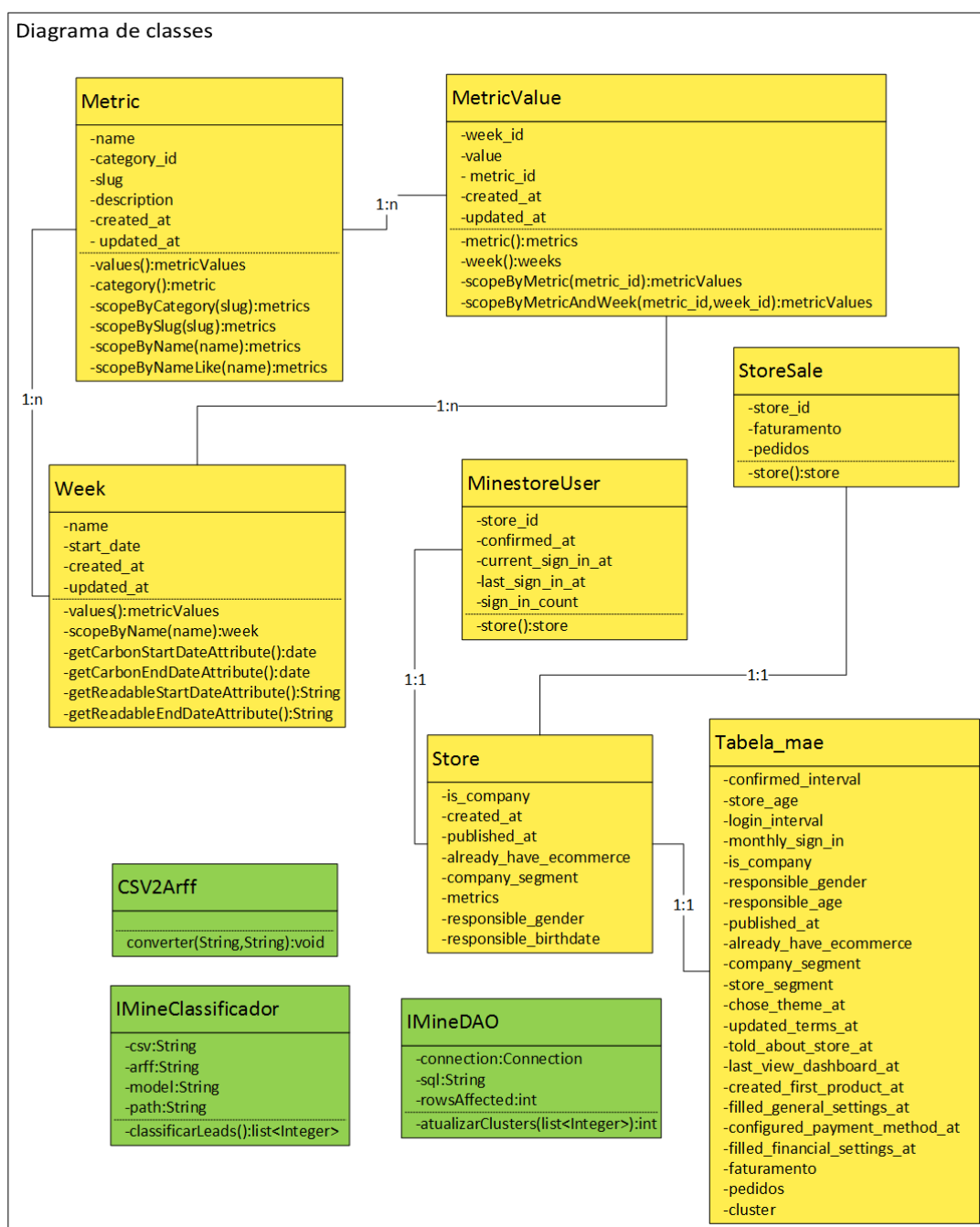
<b>Nome do caso de uso</b>	Visualizar Futuros Clientes.
<b>Ator Principal</b>	Usuário.
<b>Resumo</b>	Esse caso de uso descreve as etapas do usuário para poder visualizar a lista dos potenciais futuros clientes para empresa.
<b>Pré-condições</b>	O usuário precisa acessar o sistema. O usuário precisa ter realizado pelo menos uma vez o caso de uso anterior, “Realizar a Descoberta de Clientes”.
<b>Pós-condições</b>	O usuário poderá visualizar a lista dos potenciais clientes para a empresa.
<b>FLUXO PRINCIPAL</b>	
<ol style="list-style-type: none"> <li>1. O usuário acessa o sistema iMine.</li> <li>2. O usuário clica no item do menu “Promessas”.</li> <li>3. O sistema mostra uma lista de lojas potenciais para serem investidas pela empresa na tela do item do menu “Promessas”.</li> </ol>	
<b>FLUXO ALTERNATIVO – FAZER NOVA MINERAÇÃO DE DADOS</b>	
<ol style="list-style-type: none"> <li>1. O usuário clica no botão disponível junto da lista de potenciais clientes para realizar um novo processo de descoberta de clientes.</li> <li>2. O sistema redireciona o usuário a tela de carregamento de arquivos .csv, retornando ao UC “Carregar Arquivos para a Mineração”.</li> </ol>	
<b>FLUXO DE EXCEÇÃO (Não há fluxo de exceção)</b>	

Legenda: Documentação do caso de uso “Visualizar Futuros Clientes”. Fonte: Os Autores (2017).

## APÊNDICE H – DIAGRAMA DE CLASSES

A Figura 31 representa o Diagrama de Classes desenvolvido para a modelagem do sistema iMine. As classes em amarelo correspondem as classes do iMine e as verdes correspondem as classes do ETL 2, em Java, para melhor visualização.

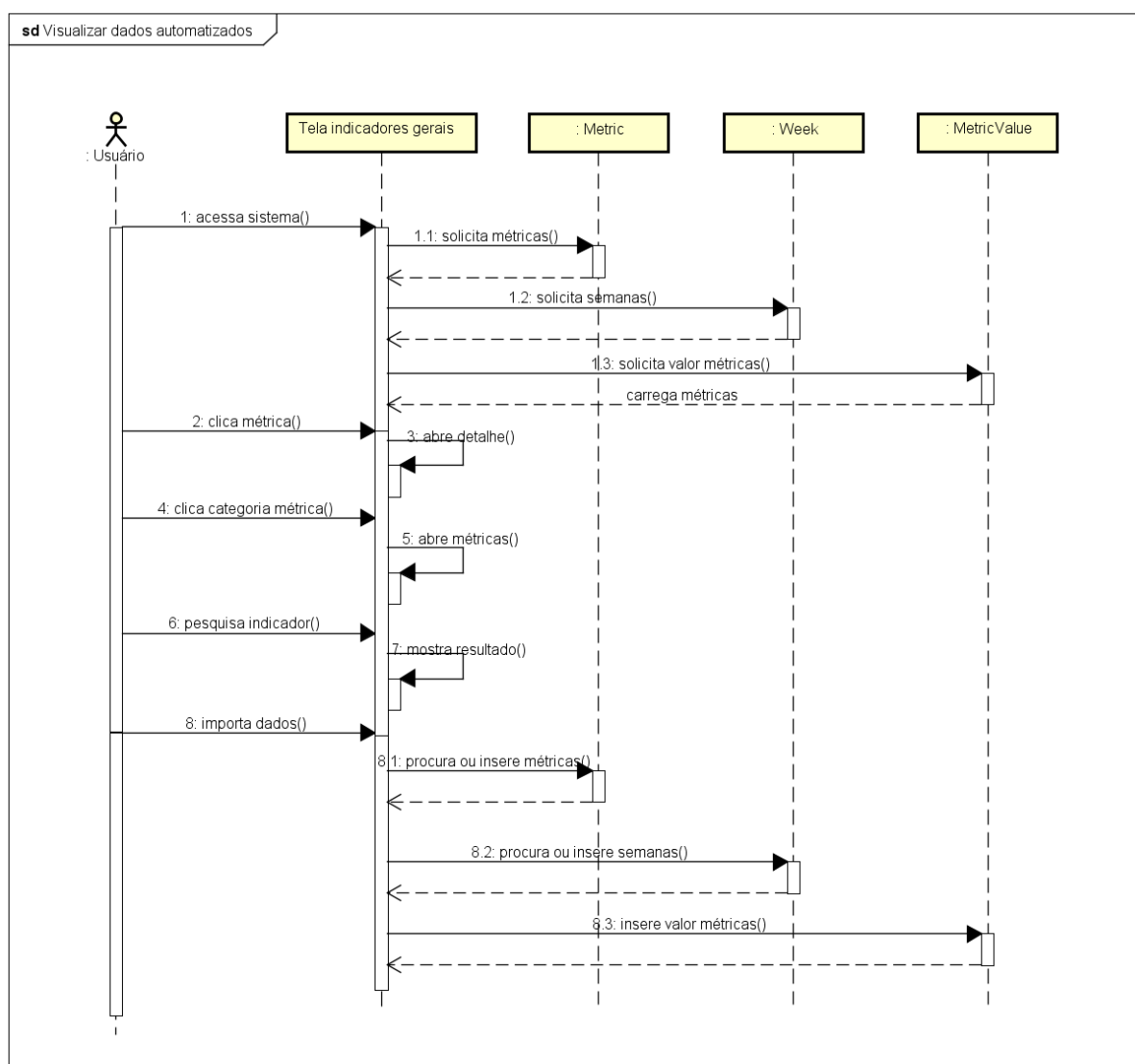
FIGURA 31 - DIAGRAMA DE CLASSES



## APÊNDICE I – DIAGRAMA DE SEQUÊNCIA VISUALIZAR DADOS AUTOMATIZADOS

A Figura 32 demonstra o diagrama de sequência do caso de uso “Visualizar Dados Automatizado”.

FIGURA 32 - DIAGRAMA DE SEQUÊNCIA VISUALIZAR DADOS AUTOMATIZADO



powered by Astah

Fonte: Os Autores (2017).

A primeira mensagem do usuário demonstrada no diagrama é a ação do mesmo de acessar o sistema, chegando a tela de indicadores gerais. Assim que essa tela é acessada, ela solicita nomes de métricas a classe Metric, semanas a

classe Week e valores de métricas para a classe MetricValue. Assim que todas as classes são instanciadas, a planilha de indicadores é preenchida na tela.

Ainda, caso o usuário deseje visualizar detalhes de uma métrica específica, a tela abre o gráfico do respectivo indicador, mostrando seu progresso. Caso ele deseje escolher uma categoria de métricas, ao clicar sobre a categoria desejada a tela mostra o grupo de indicadores.

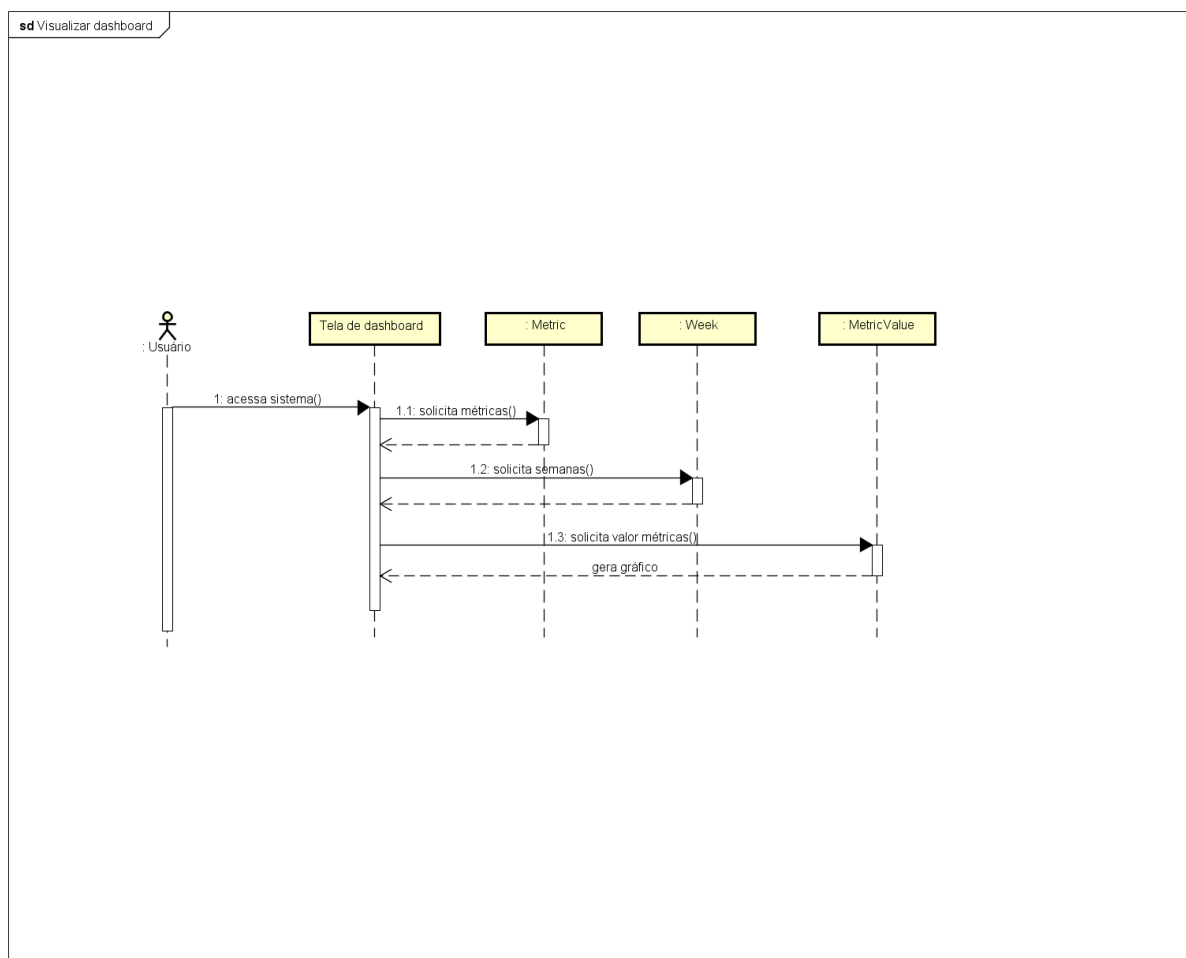
A pesquisa de métricas também está disponível ao usuário, onde é apenas necessário que o usuário digite o nome da métrica que deseja e a tela retorna a lista de resultados encontrados.

Por fim, a última mensagem mostrada é a do usuário importar dados, ação que faz a tela solicitar a procura de nome de métricas e semanas existentes. Caso elas já existam, o sistema apenas insere os novos valores. Senão o sistema insere novos nomes, semanas e valores de métricas.

## APÊNDICE J – DIAGRAMA DE SEQUÊNCIA VISUALIZAR *DASHBOARD*

A Figura 33 demonstra o diagrama de sequência do caso de uso “Visualizar *Dashboard*”.

FIGURA 33 - DIAGRAMA DE SEQUÊNCIA VISUALIZAR DASHBOARD



powered by Astah

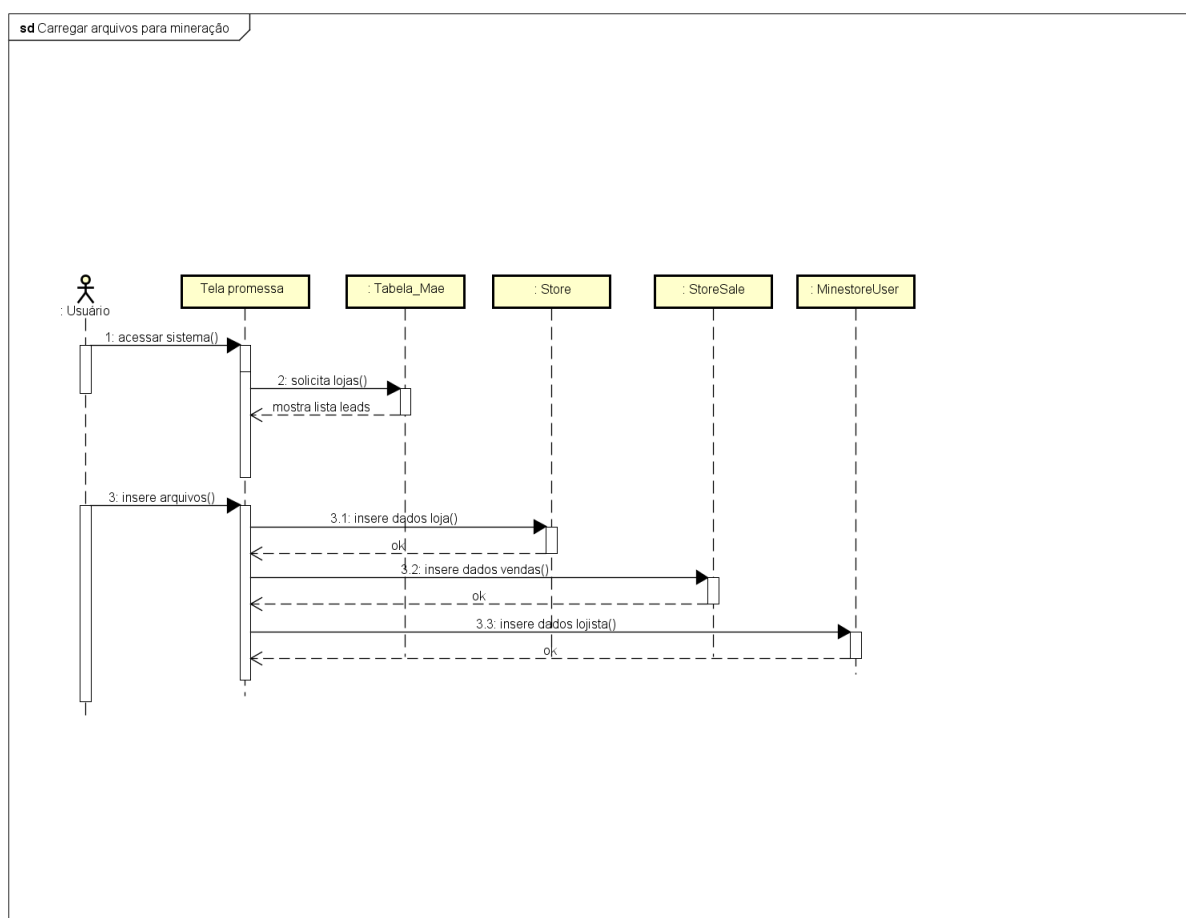
Fonte: Os Autores (2017).

A primeira mensagem do usuário demonstrada no diagrama é a ação do mesmo de acessar o sistema, chegando a tela de dashboard. Assim que essa tela é acessada, ela solicita nomes de métricas a classe Metric, semanas a classe Week e valores de métricas para a classe MetricValue. Assim que todas as classes são instanciadas, o banco então retorna uma mensagem de sucesso e os gráficos são gerados na tela.

## APÊNDICE K – DIAGRAMA DE SEQUÊNCIA CARREGAR ARQUIVOS PARA MINERAÇÃO

A Figura 34 demonstra o diagrama de sequência do caso de uso “Carregar Arquivos para Mineração”.

FIGURA 34 - DIAGRAMA DE SEQUÊNCIA CARREGAR ARQUIVOS PARA MINERAÇÃO



powered by Astah

Fonte: Os Autores (2017).

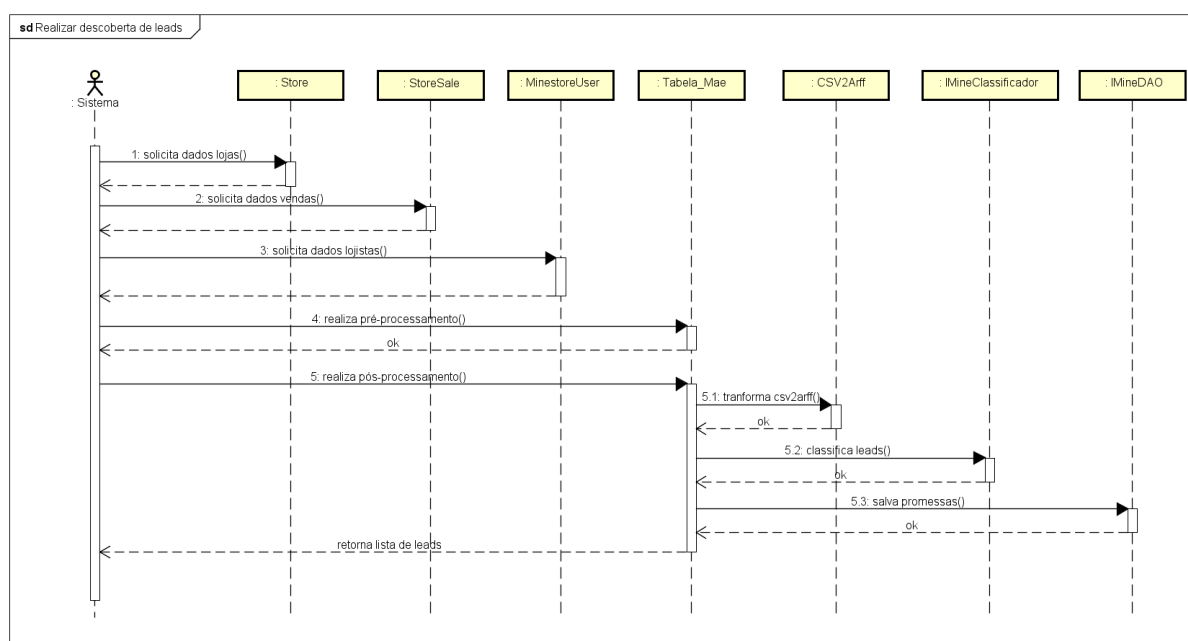
O diagrama inicia-se com o usuário acessando o sistema e indo a tela de promessas. O sistema então verifica se existe registros da última mineração de dados realizada, solicitando as lojas para a classe Tabela\_Mae. Caso esses registros não existam, a tela de promessas não mostra nenhuma lista de leads, mostrando passos para realizar o carregamento dos arquivos para começar o processamento.

O usuário insere os arquivos para a mineração, fazendo com que a tela insira dados das lojas para a classe Store, dados de vendas para a classe StoreSale e dados de lojistas para a classe MinestoreUser. Assim que inseridos no banco de dados, o mesmo retorna uma mensagem de ok para o sistema.

## APÊNDICE L – DIAGRAMA DE SEQUÊNCIA REALIZAR DESCOBERTA DE LEADS

A Figura 35 demonstra o diagrama de sequência do caso de uso “Realizar Descoberta de Leads”.

FIGURA 35 - DIAGRAMA DE SEQUÊNCIA REALIZAR DESCOBERTA DE LEADS



powered by Astah

Fonte: Os Autores (2017).

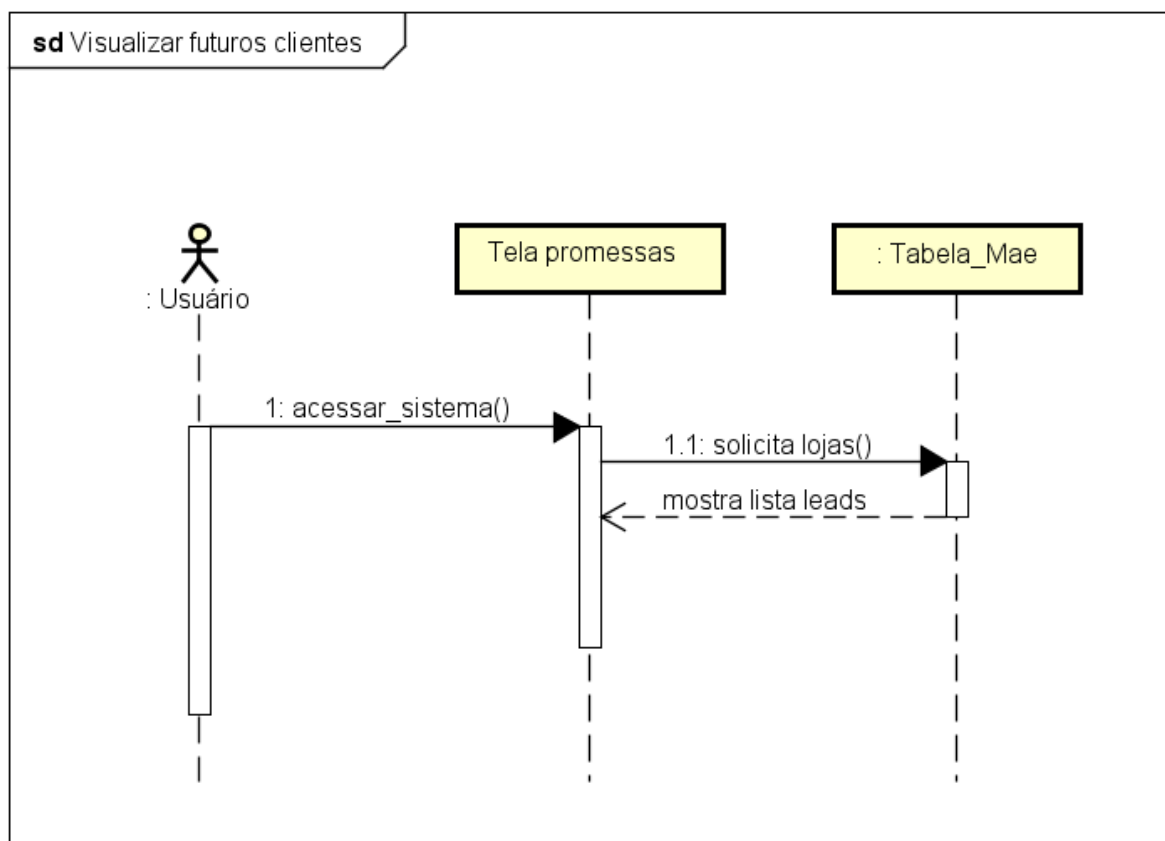
Como esse caso de uso é realizado automaticamente pelo sistema, a primeira mensagem do sistema demonstrada no diagrama é a ação do mesmo de solicitar os dados das lojas, vendas e lojistas as classes Store, StoreSale e MinestoreUser. Assim que os dados são disponibilizados para o sistema, o mesmo solicita a realização do pré-processamento de dados, instanciando a classe Tabela\_Mae no final dessa etapa, que retorna uma mensagem de sucesso caso haja sucesso. O sistema então passa para a segunda etapa, o pós-processamento de dados, onde o sistema solicita a realização do ETL de comparação de leads. Para isso, o ETL desenvolvido em Java utiliza a classe CSV2Arff para converter o .csv de leads, a classe IMineClassificador para comparar os leads e associa-los aos clusters e a classe IMineDAO para salvar a lista de leads no banco de dados. Por

fim, o banco retorna uma mensagem de sucesso e a tela mostra a lista de potenciais leads da empresa.

## APÊNDICE M – DIAGRAMA DE SEQUÊNCIA VISUALIZAR FUTUROS CLIENTES

A Figura 36 demonstra o diagrama de sequência do caso de uso “Visualizar Futuros Clientes”.

FIGURA 36 - DIAGRAMA DE SEQUÊNCIA VISUALIZAR FUTUROS CLIENTES



powered by Astah

Fonte: Os Autores (2017).

O diagrama inicia-se com o usuário acessando o sistema e indo a tela de promessas. O sistema então verifica se existe registros da última mineração de dados realizada, solicitando as lojas para a classe Tabela\_Mae. A lista de leads do último processamento é então mostrada na tela.

## APÊNDICE N – LISTA DE ATRIBUTOS PARA MINERAÇÃO DE DADOS

TABELA 10 - LISTA DE ATRIBUTOS PARA A MINERAÇÃO DE DADOS

Nome da tabela: Store	
Atributo	Definição
is_company	A loja é uma empresa e possui CNPJ.
published_at	Quando o lojista clica em abrir loja para clientes.
already_have_ecommerce	É para saber se ele vende pela internet de alguma maneira, isso muda as mensagens automáticas para eles.
company_segment	São as categorias de loja.
doing_setup_for_customer	Às vezes o lojista tem alguém contratado para configurar/estilizar a minystore, então eles fornecem material diferenciado, para criar temas customizados, html, css.
metrics	Data e hora para todos os passos: escolher tema, configurar primeiro produto, publicar a loja, clicar no banner de advertise, entre outros.
piwik_analytics_id	Não utiliza mais o piwik. Irão migrar para o Google Analytics.
responsible_gender	Gênero do lojista.
responsible_birthdate	Data de nascimento do lojista.
Nome da tabela: StoreDailyVisit	
Atributo	Definição
cart_views	Número de visitas ao carrinho, no total, por dia.
checkout_views	Número de visitas ao checkout, no total, por dia.
count	Número de visitas, no total, por dia.
date	Data das visitas.
Nome da tabela: User	
Atributo	Definição
confirmed_at	Data que foi confirmado cadastro.
current_sign_in_at	Data desse login ativo.
last_sign_in_at	Última vez que ele logou.
sign_in_count	Quantas vezes o usuário entrou
Nome da tabela: Customer	
Atributo	Definição
current_sign_in_at	Funcionalidade para clientes <i>primes</i> . O lojista tem acesso a data de quando o cliente de sua loja entrou pela última vez.
last_sign_in_at	Funcionalidade para clientes <i>primes</i> . O lojista tem acesso a

---

	data de quando o cliente da loja entrou pela penúltima vez.
sign_in_count	Funcionalidade para clientes <i>primes</i> . O lojista tem acesso a quantas vezes o cliente da loja entrou.

---

**Nome da tabela: Hooklog**

**Definição:** Hooklog é quando recebe informação de API externa. Essa tabela não foi usada para a mineração de dados.

**Nome da tabela: ImpersonateRequest**

**Definição:** Quando o administrador da minestore faz login na loja para mexer em alguma coisa, esse registro do “suporte” é salvo no banco. É possível logar no perfil do lojista para ajudar com problemas. Essa tabela não foi usada para a mineração de dados.

---

Fonte: Os Autores (2017).

## APÊNDICE O – DICIONÁRIO DE DADOS

TABELA 11 - DICIONÁRIO DE DADOS

Nome da tabela: metrics		
Campo	Fórmula	Significado
id int	-	Identificador.
category_id int	-	Identificador da categoria.
name varchar(255)	-	Nome da métrica.
slug varchar(255)	-	Tag da métrica.
description varchar(255)	-	Descrição da métrica.
created_at timestamp	-	Data de criação da métrica.
updated_at timestamp	-	Data de atualização da métrica.
Nome da tabela: metric_values		
Campo	Fórmula	Significado
id int	-	Identificador.
value varchar(255)	-	Valor.
created_at timestamp	-	Data de criação.
updated_at timestamp	-	Data de atualização.
metric_id int	-	Identificador da métrica.
week_id int	-	Identificador da semana da métrica.
Nome da tabela: minestore_users		
Campo	Fórmula	Significado
store_id bigint	-	Identificador.
confirmed_at date	-	Data de confirmação do e-mail.
current_sign_in date	-	Data do sign_in mais recente.
last_sign_in date	-	Data do último sign_in.
sign_in_count bigint	-	Contador de todos os sign_in.
Nome da tabela: store_sales		
Campo	Fórmula	Significado
store_id bigint	-	Identificador da loja.
faturamento bigint	-	Faturamento da loja.
pedidos bigint	-	Número de pedidos realizados a loja.
Nome da tabela: stores		
Campo	Fórmula	Significado
id bigint	-	Identificador.

<b>is_company varchar(5)</b>	-	Identifica se é empresa com CNPJ.
<b>created_at timestamp timestamp</b>	-	Data de criação da loja.
<b>published_at timestamp</b>	-	Data de publicação da loja.
<b>alredy_have_ecommerce varchar(5)</b>	-	Identifica se já tem outro e-commerce.
<b>company_segment varchar(5)</b>	-	Segmento da empresa (Comida, Perfumaria...)
<b>metrics varchar(600)</b>	-	Conjunto de valores de passos ou etapas durante o cadastro e configuração da loja.
<b>responsible_gender varchar(10)</b>	-	Sexo do dono da loja.
<b>responsible_birthdate date</b>	-	Data de aniversário do responsável.

**Nome da tabela: tabela\_mae**

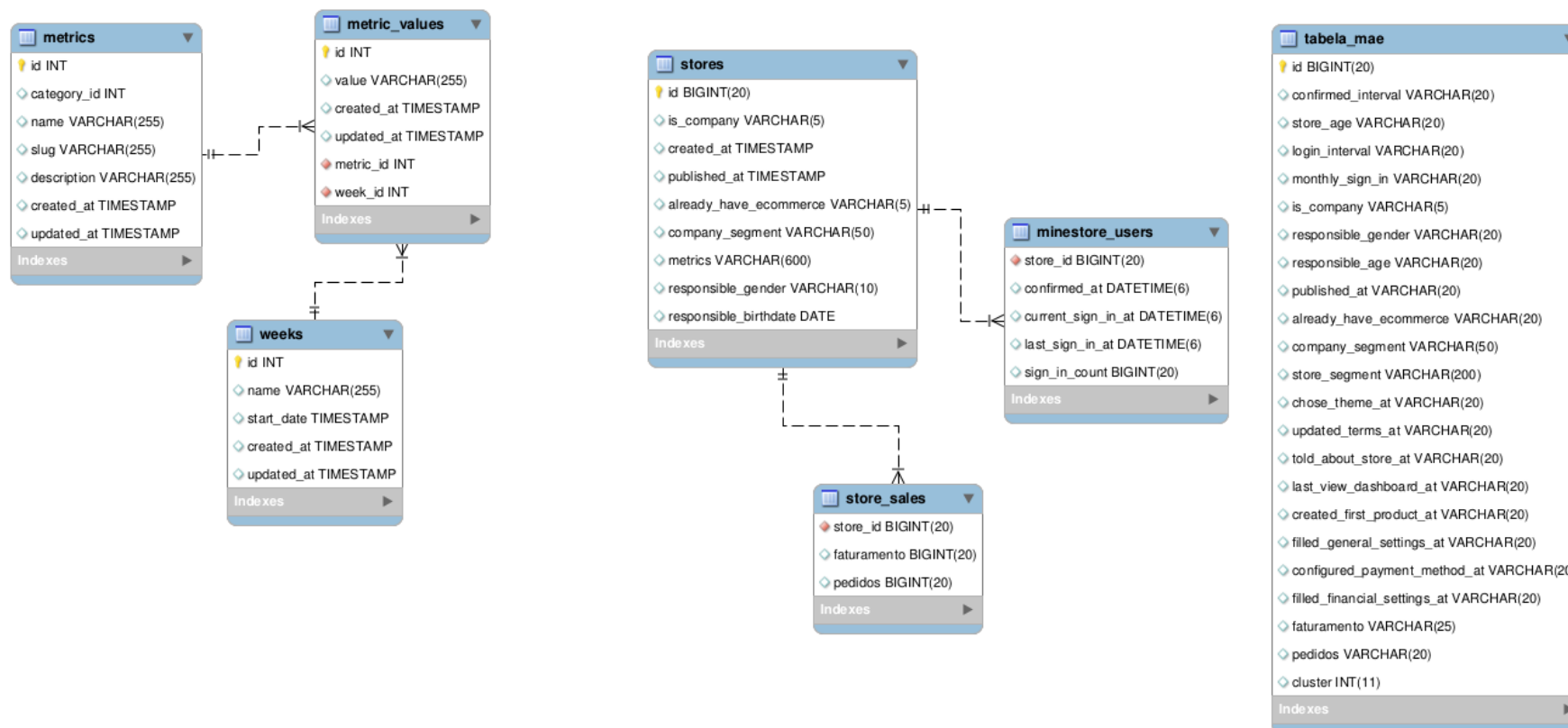
<b>Campo</b>	<b>Fórmula</b>	<b>Significado</b>
<b>id bigint</b>	-	Identificador.
<b>confirmed_interval varchar(2)</b>	confirmed_at - created_at	Intervalo de tempo que o usuário demorou para confirmar o cadastro pelo email (se nulo => nd)
<b>store_age varchar(2)</b>	sysdate - created_at	Reflete na idade da loja.
<b>login_interval varchar(2)</b>	current_sign_in_at - last_sign_in_at	Intervalo do último login para o penúltimo.
<b>monthly_sign_in varchar(2)</b>	sign_in_count / months	Média de logins mensais.
<b>is_company varchar(5)</b>	-	Identifica se é empresa com CNPJ.
<b>responsible_gender varchar(10)</b>	-	Sexo do dono da loja.
<b>responsible_age varchar(2)</b>	sysdate - responsible_birthdate	Idade do dono da loja.
<b>published_at varchar(2)</b>	published_at - created_at	Intervalo de quando ele publicou a loja (abrir loja para clientes, isso não significa que ele completou todos os passos)
<b>alredy_have_ecommerce varchar(5)</b>	-	Identifica se já tem outro e-commerce.
<b>company_segment varchar(50)</b>	-	Segmento da empresa (Comida, Perfumaria...)
<b>store_segment varchar (20)</b>		Segmento da loja (Primeira loja, Já vende, Loja para cliente - algum desenvolvedor pago para fazer o setup)
<b>chose_theme_at varchar(2)</b>	chose_theme_at - created_at	Intervalo de tempo de quando ele confirmou a loja e completou o Passo 1 - escolher o tema.
<b>last_view_dashboard_at varchar(2)</b>	last_view_dashboard_ at - created_at	Intervalo de tempo entre hoje e a última vez que visualizou o dashboard.
<b>created_first_product_at varchar(2)</b>	created_first_product_ at - created_at	Intervalo de tempo de quando ele confirmou a loja e completou o Passo 3 - criou o primeiro produto.
<b>filled_general_settings_at varchar(2)</b>	filled_general_settings_ at - created_at	Intervalo de tempo de quando ele confirmou a loja e completou o Passo 4 - complete seus dados gerais.
<b>configured_payment_metho d_at varchar(2)</b>	configured_payment_ method_at -	Intervalo de tempo de quando ele confirmou a loja e completou o Passo 6

	created_at	- configurar meio de pagamento.
<b>filled_financial_settings_at</b> varchar(2)	filled_financial_settings_at - created_at	Intervalo de tempo de quando ele confirmou a loja e completou o Passo 5 - preencher informações financeiras.
<b>cluster integer(11)</b>	-	Classifica o cluster do lojista.
<b>Nome da tabela: weeks</b>		
<b>Campo</b>	<b>Fórmula</b>	<b>Significado</b>
<b>id int</b>	-	Identificador.
<b>name varchar(255)</b>	-	Nome da semana.
<b>start_date timestamp</b>	-	Data de início da semana.
<b>created_at timestamp</b>	-	Data de criação da semana.
<b>updated_at timestamp</b>	-	Data de atualização da semana.

Fonte: Os Autores (2017).

## APÊNDICE P – MODELAGEM DO BANCO

FIGURA 37 - MODELAGEM DO BANCO DE DADOS



Fonte: Os Autores (2017).