UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS GUARAPUAVA
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

FELIPE GOMES

SISTEMA DE APOIO À GESTÃO DE COMÉRCIO ELETRÔNICO POR INTELIGÊNCIA DE NEGÓCIOS

TRABALHO DE CONCLUSÃO DE CURSO

GUARAPUAVA

2016

FELIPE GOMES

SISTEMA DE APOIO À GESTÃO DE COMÉRCIO ELETRÔNICO POR INTELIGÊNCIA DE NEGÓCIOS

Trabalho de Conclusão de Curso apresentado à disciplina Trabalho de Conclusão de Curso II do Curso de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná como requisito parcial para aprovação.

Orientador: Prof. Me. Emerson André Fedechen

GUARAPUAVA

2016

RESUMO

GOMES, Felipe. SISTEMA DE APOIO À GESTÃO DE COMÉRCIO ELETRÔNICO POR INTELIGÊNCIA DE NEGÓCIOS. 59 f. Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2016.

Os sistemas de comércio eletrônico, mediante o comportamento do cliente no site, podem gerar informações, que se devidamente coletadas e trabalhadas fornecem ao gestor da loja a possibilidade de entender aspectos do seu estabelecimento virtual. Visto que plataformas de comércio eletrônico normalmente não tem como foco fornecer dados de inteligência de negócios, o objetivo deste trabalho foi desenvolver um sistema para coletar dados da plataforma de comércio eletrônico durante o processo de compra, e com isso extrair valores que informem pontos de comportamento do cliente, como por exemplo, percentual de abandono de carrinho de compras, taxa de conversão. Por fim, gerar relatórios que expressem inteligivelmente essas informações, para que sirvam de apoio à tomada de decisão do gestor de forma a melhorar a experiência de compra de seu comércio eletrônico.

Palavras-chave: Inteligência de Negócios, Comércio Eletrônico, Sistema web

ABSTRACT

GOMES, Felipe. Work Title. 59 f. Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2016.

E-commerce systems, through the behavior of the customer on the site, can generate information, which if properly collected and worked give the store manager the possibility of understanding aspects of their virtual establishment. Since e-commerce platforms typically do not focus on providing business intelligence data, the objective of this work was to develop a system for collecting data from the e-commerce platform during the purchasing process, thereby extracting values that inform behavior points Of the customer, such as percentage of shopping cart abandonment, conversion rate. Finally, generate reports that intelligibly express this information, to support the manager's decision making in order to improve the e-commerce buying experience.

Keywords: Business Intelligence, E-Commerce, Web System

LISTA DE FIGURAS

LISTA DE SIGLAS

HTML	HyperText Markup Language
API	Application Programming Interface
W3C	World Wide Web Consortium
CSS	Cascading Style Sheets
DOM	Document Object Model
URL	Uniform Resource Locator
HTTP	Hypertext Transfer Protoco
XML	eXtensible Markup Language
HTTP	Hypertext Transfer Protocol
NoSQL	Not Only SQL
BSON	Binary Object Notation
JSON	JavaScript Object Notation
BI	Business Inteligence
OECD	Organisation for Economic Cooperation and Development
SEO	Search Engine Optimization

SUMÁRIO

1 INTRODUÇÃO

Comércio eletrônico é uma forma de comprar e vender pela *internet*, objetivando lucro, mas também visa aumentar a eficiência do negócio, por meio de novos canais, alcançando mais clientes em novos mercados. Ele utiliza os sistemas de informação para realizar transações de compra e venda de bens e serviços, entre pessoas e empresas ou entre empresas e empresas (MADEIRA, 2007).

Sistemas de comércio eletrônico comumente chamados de *e-commerce* ou *loja virtual*, correspondem a forma de compra e venda de produtos por meio da Internet. Eles atuam através do formato eletrônico e comunicam com outros sistemas eletrônicos, propiciando que uma empresa conduza seus negócios por meio da rede mundial de computadores (SMITH et al., 2000).

As soluções de comércio eletrônico, principalmente as de código aberto como o *Magento Community*[1], uma das plataformas de comércio eletrônico de código aberto mais usada atualmente (E-COMMERCE, 2014); geralmente não focam em disponibilizar relatórios mais robustos ou específicos sobre as atividades dos clientes no processo de compra do produto. Eles geralmente proporcionam relatórios sobre as ordens de compra, totais em vendas e armazenam dados sobre os clientes como: nome, *e-mail*, endereços; dados pouco relevantes para extrair informações. Essa limitação é a motivação para este trabalho: desenvolver um sistema de apoio à gestão. Sistemas de apoio à gestão proporcionam informações para a tomada de decisão, mediante a manipulação de dados de forma a deixá-los compreensíveis para os gestores tomarem decisões ao examinarem situações e desempenhos corporativos (TURBAN et al., 2009).

A análise de uma gama de dados relacionados, tem grande importância para as empresas de comércio eletrônico. Isso permite avaliar o contexto da empresa, para decisões mais assertivas, com pouca ou nenhuma margem de erro, o que pode significar o sucesso da loja virtual (CONTENT, 2015a). Mas esse tipo de análise torna-se impossível de ser feita sem a ajuda de um sistema, que possa computar o grande montante de dados oriundos ao volume de acessos

[1]Magento Community: plataforma de código aberto para lojas virtuais: Disponível em: https://www.magentocommerce.com/download

da loja virtual, e as possibilidades de cruzamentos entre eles de modo a obter uma informação de interesse, como por exemplo, avaliar os perfis de clientes, quais clientes são mais assíduos.

O presente trabalho visa desenvolver um sistema de apoio à gestão estratégica de comércio eletrônico, na forma de um módulo para uso universal de fácil utilização onde o cliente poderá, baseando-se na coleta, processamento e pela análise dos dados coletados, entender o comportamento dos clientes, ou grupos de clientes, por meio da compreensão de diversos dados oriundos ao processo de compra e venda de produtos.

No Capítulo 2, há uma descrição de serviços semelhantes ao sistema, a fundamentação teórica com a descrição das tecnologias e conceitos usados no projeto. No Capítulo 3, há uma descrição das etapas concernentes ao projeto. No Capítulo 4, há a descrição da idealização do projeto. No Capítulo 5, um detalhamento da implementação e aplicação das tecnologias. No Capítulo 6, há um relato da aplicabilidade do projeto em duas lojas virtuais reais. No Capítulo 7, descreve-se as considerações sobre o projeto e das atividades desenvolvidas nele e o sobre os trabalhos futuros.

1.1 OBJETIVOS

1.1.1 OBJETIVO GERAL

O objetivo desse trabalho foi desenvolver um sistema de apoio à decisão aplicado a sistemas de comércio eletrônico.

1.1.2 OBJETIVOS ESPECÍFICOS

- Levantar dados bibliográficos sobre a importância e aplicação de inteligência de negócios, para a melhor gestão do comércio eletrônico;
- Definir os tipos de informações que espera-se gerar com o sistema;
- Desenvolver um sistema para coleta de dados referentes ao comportamento do cliente durante o processo de compra;
- Criar uma forma de armazenamento organizado dos dados considerados relevantes;
- Desenvolver um módulo de processamento de dados, para processar os dados primários armazenados de forma a encontrar dados reduzidos que possam gerar informação;
- Desenvolver um módulo de apresentação para mostrar os dados reduzidos de forma a alimentar relatórios e gráficos, para que o gestor possa compreender à informação, e

entender aspectos de seus negócios para tomada de decisões e melhor conhecimento do seu comércio virtual;

1.2 DIFERENCIAL TECNOLÓGICO

O sistema proposto visa oferecer, objetivamente, dados referentes à loja virtual, sem a necessidade de configurar painéis complexos, como o *Google Analytics*. Ao mesmo tempo, o sistema pretende apresentar indexadores de desempenho focados para o comércio eletrônico, sem generalidades de uso. Também não será necessário o usuário informar manualmente os dados, como é o caso do *Qlik Sense*, um *script* vai coletar dados pré-configurados para posterior análise. Terá documentação totalmente em português brasileiro, diferente do *Wunderdata*.

As métricas propostas visam oferecer ao gestor informações como:

- Informativo de produtos comprados juntos;
- Agrupamentos de clientes, perfis de clientes;
- Caminhos comuns de clientes no *site*;
- Informações de desempenho de estoque;

Essas informações podem propiciar ao gestor maximizar lucros e reduzir custos, pela melhoria do serviço prestado por sua loja virtual.

2 RESENHA LITERÁRIA

Neste capítulo serão abordados os trabalhos semelhantes já desenvolvidos na área do projeto. Assim, como as tecnologias que foram utilizadas no desenvolvimento do projeto.

2.1 ESTADO DA ARTE

Nesta seção serão abordados trabalhos correlatos, sendo apresentado os mais relevantes a essa pesquisa e desenvolvimento.

2.1.1 GOOGLE ANALYTICS

Entre sistemas para gerar informações de inteligência de negócios o *Google Analytics*, é um sistema para uso em *websites* e oferece diversas métricas para serem usadas (ANALYTICS, 2015). Algumas métricas proporcionadas pelo *Google Analytics* para comércio eletrônico, segundo a agência de *marketing RockContent* (CONTENT, 2015b) são:

- Sessão ou visitas, número total de quantas pessoas estão acessando o *site* nos últimos 30 dias;
- Páginas por visita, número médio de páginas acessadas por um visitante;
- Duração média por sessão, tempo gasto pelo visitante no *site*;
- Visitantes únicos, número de pessoas que visitaram o *site* somente uma vez nos últimos 30 dias;

O sistema do *Google Analytics* como informa o *site 5Seleto*, é uma ferramenta genérica que possui mais de 500 métricas para *websites* (MEDEIROS; BELEZA, 2015). Isso torna-o complexo para usuários menos experientes como citado pela *Academia do Marketing* (MARKETING, 2015). Dada a complexidade de configuração do sistema, percebeu-se a necessidade

de um treinamento prévio ou estudo mais aprofundado, de forma a extrair um melhor rendimento do sistema, o que para usuários comuns pode ser uma dificuldade a ser superada.

2.1.2 QLIK SENSE

Outro sistema é o *Qlik Sense*, que é uma aplicação de inteligência de negócios para diversos serviços, como *telecom*, saúde, serviços de aluguel e loja virtual (SENSE, 2015). Para lojas virtuais dispõe de painéis para informar dados sobre:

- País: total de vendas, venda ao longo do tempo, margem de lucro, valor das vendas pela loja física ou pela loja virtual;
- Clientes: percentual de sexo, idade média dos clientes, clientes por região, quantidade de pedidos de compra por cliente;
- Vendas e o clima: apresenta uma relação gráfica das vendas em relação a média de precipitação, e temperatura;

A solução *Qlik Sense*, também é uma ferramenta genérica, conforme indica a empresa detentora do software, com seus diversos usos possíveis, permitindo ao usuário integrar diversas fontes de dados. O sistema é distribuído na versão *desktop*, mas os resultados podem ser compartilhados e salvos no serviço *Qlik Cloud* [1]. Destaca-se que este não coleta dados automaticamente, passando esta responsabilidade ao usuário que deve informar os dados e tentar extrair informações pelo cruzamento dos dados, conceito do *self-service BI* (OLIVEIRA, 2015).

2.1.3 WUNDERDATA

Um terceiro sistema, é o serviço da *Wunderdata*[2] de inteligência de negócios para lojas *on-line*. Suas métricas disponíveis informam dados sobre:

- Quantidade de novos usuários;
- Visitantes únicos;
- Origem das receitas;

[1] Qlik Cloud é uma plataforma para compartilhar dados dos aplicativos Qlik Sense, para que seja possível colaborar com outras pessoas e fazer descobertas de dados juntos. Disponível em: https://goo.gl/OSRLyy . Acesso em: 15/11/2015.

[2] Wunderdata disponível em : http://wunderdata.com

- Vendas acumuladas;
- Receita semanal por categoria;

Este serviço de inteligência de negócios da *Wunderdata* é focada em comércio eletrônico, e é totalmente *web*. O *site* oficial da empresa oferece uma página para testar o produto, onde é possível ver indicadores básicos de desempenho para comércio eletrônico, como os citados acima, no entanto é apresentada em inglês, o que pode ser uma limitação com o público brasileiro.

2.1.4 COMPARATIVO

A Tabela 1 demonstra uma comparação das características citadas entre os sistemas de inteligência de negócios. Assim o sistema proposto pretende atender especificamente a lojas virtuais, focando suas métricas em proporcionar dados de interesse desse seguimento. No sistema proposto, irá ocorrer uma coleta automática dos dados vindos da loja virtual, sem o gestor se preocupar em colocar dados em planilhas ou outras fontes para depois inserir manualmente no sistema. Com essas características e simplicidade o sistema visará uma menor complexidade na configuração e adaptação da loja virtual. O sistema também será totalmente em português para facilitar o entendimento ao público brasileiro.

	Específico pra Loja Virtual	Coleta automática	Maior dificuldade de Configuração	Português Brasileiro
Google Analytics		x	x	x
Qlik Sense			x	x**
Wunderdata	x	x		
Sistema proposto	x	x		x

** Português na versão Desktop.

Tabela 1: Comparativo de características entre os sistemas de inteligência de negócios.
Fonte: autor

2.2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são apresentadas as tecnologias e conceitos utilizados no desenvolvimento do sistema que constitui o projeto. Tem-se como intenção realizar um estudo sobre as lingua-

gens de programação, banco de dados, conceitos teóricos de inteligência de negócios e loja virtual. Assim é possível se ter uma visão mais concreta do projeto e de como ele será colocado em prática.

2.2.1 LINGUAGEM DE MARCAÇÃO HTML

HTML é a sigla em inglês para *HyperText Markup Language* que traduzindo para o português se compreende como linguagem de marcação de hipertexto. Hipertexto pode ser descrito como todo conteúdo colocado em um página *web*, que tem por elementar característica se interligar a outros documentos da *web* por meio de *links* (SILVA, 2010a).

HTML é entendida como uma linguagem de marcação e não programação, pois sua função é enviar ao navegador, informações que definem de que forma textos, imagens, *links* entre outro itens devem ser dispostos na tela. Assim o navegador recebe essas informações e de acordo com as suas marcações renderiza o documento (BRITO, 2011).

O HTML se encontra na versão 5, sendo um dos seus principais objetivos facilitar a manipulação de elementos, permitindo que CSS e Javascript, trabalhem da melhor maneira possível. Essa versão possibilita o uso de novas *tags*[3] e melhora a função de outras. Possibilitando uma melhor maneira de organizar informações no documento HTML, permitindo códigos mais semânticos. Há novos recursos para multimídia, utilização de *Application Programming Interface* (APIs) e novos elementos de formulários também estão presentes nessa versão(IAMASHITA; MAGALHÃES, 2013).

Empresas como Apple, Inc. [4], Google, Inc. [5], Microsoft Corporation [6], utilizam HTML em seus produtos. Além de que, também participam como membros da *World Wide Web Consortium* (W3C)[7], organização responsável pela padronização da web (W3C, 2015).

HTML foi utilizado nesse projeto para propiciar ao gestor do painel administrativo, a apresentação de seus dados, de forma a conter imagens, gráficos, campos de formulários e textos que permitam a utilização e interatividade com o sistema.

[3]Tag significa etiqueta, que informam ao navegador como interpretar o conteúdo contido nela. Disponível em: http://www.htmlprogressivo.net/2013/04/O-que-sao-tags-em-HTML-para-que-servem-e-como-usar.html

[4]https://www.apple.com

[5]https://www.google.com

[6]https://www.microsoft.com

[7]W3C: http://www.w3c.org

2.2.2 CSS

CSS é a abreviação do inglês *Cascading Style Sheet*, que em português se traduz como "folhas de estilo em cascata". Tem por finalidade apresentar o conteúdo HTML, cores de fontes, posicionamento dos elementos, tamanho dos textos, todo o aspecto visual pertence ao CSS (SILVA, 2011).

CSS provê auxílio na definição de estilos da página, permitindo ajustar de forma mais precisa qualquer aspecto de um elemento presente na página. Folhas de estilos propicia a separação do conteúdo da página e do estilo, deixando mais simples a manutenção com páginas mais limpas e claras. Outra benesse, é aproveitar a qualidade dos navegadores em armazenar em *cache* a declaração dos estilos da página que está em arquivo separado, economizando na transferência de dados, visto que somente no primeiro acesso do usuário, é transferido o estilo, e posteriores acessos terão um carregamento mais rápido (OLIVEIRA; LEANDRO, 2007).

As folhas de estilos se encontra na versão 3, entre os pontos essenciais dessa versão, destaca-se a possibilidade de selecionar elementos específicos em um grupo, gradiente em texto e elementos, bordas arredondadas, manipulação de opacidade, animações, controle de perspectiva, rotação e estruturação independente do posicionamento no código HTML(W3C, 2015).

As CSS são padronizadas pelo W3C. Algumas das empresas membro do consórcio que adotam seus padrões *web* tem-se Facebook[8], Opera Software [9], PayPal [10] (W3C, 2015).

Neste projeto, o uso de folhas de estilo, foi essencial para proporcionar páginas *web* de uso agradável e inteligível, com elementos posicionados e estilizados de modo a garantir a diagramação de imagens, *links*, textos entre outros itens que a página venha a apresentar.

2.2.3 BOOTSTRAP

Bootstrap é um *framework* que inclui HTML, CSS, JavaScript, para desenvolvimento de aplicações *web* com *layout* responsivo [11]. Facilita o desenvolvimento *front-end* tornado-o mais rápido e fácil. O *Bootstratp* foi desenvolvido para que seja usado com HTML5; muitas de suas funcionalidades usufruem de elementos exclusivos dessa linguagem de marcação, como também propriedades avançadas de CSS que dependem de HTML5 (SILVA, 2015).

Entre os recursos o *Bootstrap* oferece, *reset* de CSS, estilo visual básico para maioria

[8] https://www.facebook.com
[9] http://www.opera.com
[10] https://www.paypal.com/br
[11] Responsivo: um layout que se adapta aos diversos tamanhos de telas.

das tags, conjuntos de ícones, *Grids* [12], *plugins* JavaScript, tudo é responsivo (CAELUM, 2015).

Esse *framework front-end*, foi utilizado para o desenvolvimento do painel do cliente, para facilitar na estruturação do *layout*, pelo uso de elementos já estilizados e outros recursos.

2.2.4 JAVASCRIPT

JavaScript é uma linguagem de programação da *Web*, amplamente usada pela maioria dos *sites* e navegadores modernos - computadores de mesa, consoles de jogos, *tablets* e *smartphones* - incluem interpretadores JavaScript, tornando-a a linguagem de programação mais onipresente da história (FLANAGAN, 2013).

JavaScript proporciona interatividade entre o usuário com o *Document Object Model* (DOM) da página em que se encontra. Permite manipular elementos HTML de forma a definir, alterar e controlar dinamicamente a apresentação de um documento HTML; aspectos como cores, textos, posição de elementos além da possibilidade de manipulação das folhas de estilos, inserindo ou anulando características existentes dinamicamente, sem a necessidade de recarregar a página *web*.

Com JavaScript é possível capturar eventos do mouse, teclado, janela e do documento da página acessada, apresentar mensagens ao usuário, acessar e validar campos de formulários, realizar cálculos, fornecer dicas de preenchimento de campos, validar campos de formulários, criar janelas *pop-up*, fechar e abrir janelas. Em conjunto com outras linguagens, ajuda a cumprir tarefas relacionadas ao fluxo de programação (SILVA, 2010b)

Algumas características do JavaScript incluem, como elenca Grillo (2008):

- É uma linguagem de *script*, onde o interpretador lê o código e executa instrução por instrução, a partir do próprio código fonte, toda vez que o *script* é executado;
- Tipagem dinâmica, nesse modo as variáveis podem assumir qualquer valor, podendo mudar seu tipo durante a execução do *script*, assumindo o tipo implícito da variável a qual recebe seu valor;
- Função de ordem superior, funções que recebem uma ou mais funções como argumentos ou que têm uma função como saída;
- Segurança, por ser uma linguagem que é executada no cliente, necessita de restrições para evitar códigos maliciosos que podem causar problemas. Limitações essas que incluem: a

[12] Técnica que permite criar layout responsivos

proibição de ler arquivos no computador do usuário, criar arquivos (exceto *cookies*), ler configurações do sistemas operacional, acessar *hardware*, iniciar outros programas;

- Pode ser executado no lado do servidor, usando usando Node.js um interpretador Javascript no lado do servidor.

Também é possível o uso de *XMLHttpRequest* que é um objeto JavaScript que foi projetado pela Microsoft e adotado pela Mozilla, Apple e Google. Ele agora está sendo padronizado no *W3C*. Ele fornece uma maneira fácil de recuperar dados em uma *Uniform Resource Locator* (URL). Apesar do nome, *XMLHttpRequest* pode ser usado para recuperar qualquer tipo de dado, e não apenas XML, e suporta protocolos diferentes de *Hypertext Transfer Protocol* HTTP incluindo file e ftp como mostra a documentação *web* da *Mozilla* (MOZILLA, 2015).

Neste projeto, JavaScript foi amplamente usado. As informações que são coletadas no computador do cliente(consumidor) estão formatadas como um objeto JavaScript. No computador do cliente tem um *script* usando essa tecnologia que age como um gatilho e solicita outro *script* em JavaScript, que coleta os dados do computador cliente e envia para o módulo de coleta de dados. Esse módulo usa a tecnologia Node.js, para processar os dados e salvá-los no banco de dados. Ainda assim, no painel de configurações e relatórios que cada gestor tem acesso, usa-se JavaScript para manipulação dos elementos do DOM e gráficos.

2.2.5 JQUERY

jQuery é uma biblioteca Javascript que foca principalmente na simplicidade. Ela facilita o trabalho em criar efeitos selecionar elementos do DOM. Isso tudo feito com Javascript é muito custoso e gera muitas linhas de códigos. jQuery permite a criação de extraordinários efeitos de forma muito simples, facilitando o desenvolvimento de *scripts* de forma imediata, sem necessitar de uma grande curva de aprendizado (SILVA, 2008).

Entre o que é possível fazer com o a jQuery e suas vantagens como elenca Rutter (2012):

- Eventos que incluem interações de mouse, teclado, formulário e do usuário;
- Efeitos que incluem exibir ou esconder, deslizar, transições ou animações personalizadas entre os elementos;
- Animações que permitem a movimentação de objetos com CSS e efeitos nativos;

- Métodos Ajax: *Asynchronous JavaScript and XML*(JavaScript e XML assíncrono), é muito mais que a junção de JavaScript com *eXtensible Markup Language*(XML), é todo um conceito de navegação e atualização de páginas *Web* para a interface com o processamento de formulários no servidor com XML (SOARES, 2006). O XML não é uma linguagem de marcação predefinida (como o HTML) e possibilita ao autor do documento projetar sua própria marcação (ALMEIDA, 2002);
- Extensibilidade para a criação de *plug-ins* pessoais que ampliam a base da API da jQuery;
- Manipulação do DOM;
- Manipulação da CSS;
- Utilitários que fornecem a detecção de navegadores e interfaces mais fáceis para funções comuns do Javascript;

Neste projeto jQuery é muito utilizado para criar efeitos, manipular elementos e dar suporte a *plugins* de animações no painel administrativo, facilitando *feedbacks* e a interação do usuário com o sistema.

2.2.6 NODE.JS

O Node.js é um interpretador JavaScript rápido, implementado em C++, com vínculos para as APIs Unix de baixo nível para trabalhar com processos, arquivos, soquetes de rede, também para clientes HTTP e APIs de servidor entre outros. A não ser alguns métodos síncronos com nomes especiais, os vínculos do Node são todos assíncronos e, por padrão, os programas Node nunca são bloqueados, isso quer dizer que lidam com cargas grandes de forma eficiente. Como as APIs são assíncronas, o Node conta com rotinas de tratamento de evento, as quais são frequentemente implementadas com funções aninhadas e *closures* [13](FLANAGAN, 2013).

Node.js veio para resolver o problema de arquitetura bloqueante. Entende-se por bloqueante, por exemplo, sistemas *hardware* desenvolvidos em *PHP, Ruby, Python*, que tem uma característica em comum, que é a paralisação do processamento enquanto realizam uma operação de entrada ou saída no servidor, isso é conhecido como modelo bloqueante (bloqueio de *thread*). Assim, Node.js permite que a aplicação tenha mais eficiência e leveza, essencial para atender sistemas que demandam grande troca de dados. Em um sistema bloqueante cada

[13]Closure: Um closure é uma função interior que tem acesso a variáveis de uma função exterior – cadeia de escopo.

requisição é enfileirada, e depois processada uma a uma, isso não permite o múltiplo processamento dessas requisições, pois, enquanto existe o processamento de uma requisição as demais mantém-se na fila, fazendo com que o processo tenha tempo ocioso. Node.js inova em sua arquitetura por ser totalmente não-bloqueante, o que propicia performance e bom consumo de memória, aumentando a eficiência em servidores que possuem alta carga (PEREIRA, 2014).

Entre as empresas que usam Node.js pode-se citar Dow Jones o usa em alguns de seus serviços [14], GoDaddy [15], LinkedIn [16] usa Node.js como interface de servidor para aplicações *mobile* (GITHUB, 2015).

O projeto em questão, usou Node.js no desenvolvimento do módulo de coleta de dados, que serviu também como uma API. A ideia é o sistema atender centenas de milhares de requisições, oriundas de vários computadores clientes das lojas virtuais. Devido as características citadas acima, os benefícios alcançados com o Node.js vem propiciar uma melhor gestão de uma grande quantidade de requisições de forma mais performática.

2.2.7 PHP COM FRAMEWORK LARAVEL

PHP é um acrônimo recursivo para "*PHP: Hypertext Preprocessor*", é uma linguagem de programação interpretada, que possibilita o pré-processamento de páginas HTML. Com isso é possível gerar páginas *web* com conteúdo de forma dinâmica, permitindo alterações do conteúdo HTML apresentado, antes de ser enviado pelo servidor ao computador do usuário. Permite também receber dados de entradas de formulários para um melhor forma de interação (BENTO, 2013).

Laravel é um *framework* criado na linguagem de programação PHP. Por *framework* entende-se ser um esqueleto em cima do qual se desenvolve uma aplicação completa, provendo uma funcionalidade genérica (ANTONIO, 2015).

Foi criado com o ideal de ser um *framework* simples e divertido de usar. Apresenta facilidade de uso, uma comunidade ativa, documentação clara e estabilidade. Entre suas características, segundo Bezerra (BEZERRA; SCHIMIGUEL, 2016), tem-se:

- Módulo de *Routing* (Utilizado para criar rotas HTTP);
- Módulo *Middleware* (Utilizado para filtro de requisições HTTP);

[14] Sítio usando Node: http://online.wsj.com/
[15] http://www.godaddy.com
[16] http://linkedin.com/

- Módulo *Responses* (Utilizado para respostas HTTP, como por exemplo: JSON, *File*);
- Módulo *Cache* (Utilizado para o cache entre sua aplicação e o banco de dados);
- Módulo *Eloquent* (ORM ou *Object Relational Mapping*, módulo que faz o relacionamento do banco de dados em forma de objetos);
- Módulo *Mail* (Responsável por funções relacionadas ao envio de *e-mails*);

Apresenta também uma modularidade robusta que permite ao desenvolvedor utilizar diversos padrões de projeto.

Entre as empresas que adotam o PHP é possível citar Wikipedia [17] uma enciclopédia *on-line* e gratuita, o Facebook [18] uma rede social *web*.

Neste projeto, o uso de PHP com o *framework* Laravel, foi necessário para criar o sistema que foi usado para o painel administrativo, onde os clientes (gestores) e o administrador do sistema, tem acesso a funções de cadastrado e edição de seus dados e da loja virtual, obtenção dos *scripts* e identificador da loja, visualizar os gráficos com as métricas estabelecidas, e acompanhar os serviços disponíveis pelo sistema.

2.2.8 MONGODB

MongoDB é um de banco de dados NoSQL *schema-less*, que não possui esquema fixo, permite gravar dados em qualquer estrutura, orientados a documentos, têm alta performance, alta disponibilidade e fácil escalabilidade (SCHROEDER; SANTOS, 2014).

Documentos em um banco de dados NoSQL, são coleções de atributos e valores, onde um atributo pode ter múltiplos valores. Com isso não é preciso definir um esquema rígido de dados. Essas coleções podem ser entendidas como uma tabela de um banco relacional. Caso um documento necessite conter um campo diferente, isso pode ser feito simplesmente incluindo ou excluindo esse campo, essa é uma vantagem, pois campos com valores vazios não precisam ser armazenados. Armazena os dados em coleções de documentos no formato BSON, que é uma serialização binária do formato JSON um formato leve para troca e armazenamento de dados computacionais(MALAGOLI et al., 2013).

O NoSQL é um novo paradigma de armazenamento de dados, que objetiva melhorar e facilitar o escalonamento dos dados, ou seja permite uma otimizada implementação de sistema

[17] https://www.wikipedia.org/
[18] https://www.facebook.com/

em *clusters* [19] de servidor, mais flexíveis e simples do que com os bancos de dados relacionais. Outra vantagem é o grande desempenho na busca de grandes quantidade de dados, o que motiva grandes empresas a usar esse tipo de serviço (ROSA, 2009).

No próprio *site* do Mongodb [20], tem-se alguns casos de uso por empresas entre elas: Forbes [21], Hudl [22], Pearson[23], entre outros.

Esse projeto tem como objetivo armazenar dados de vários clientes oriundos de várias lojas virtuais, cada acesso a uma página vai gerar dados variados. Esse armazenamento tende a tomar grandes proporções, por isso a necessidade de um banco escalável e que trabalhe de forma performática com um grande volume de dados;

2.2.9 MYSQL

MySQL é um sistema de gerenciamento de banco de dados do tipo objeto-relacional. Esses sistemas tem um modelo de dados que consiste de uma coleção de relações com nome, contendo atributos definidos especificamente.

Como gerenciador de banco de dados, possui uma coleção de programas que permitem a manipulação de dados para as mais diversas finalidades. Entre suas características está o controle de redundância para evitar inconsistências, controle de acesso de usuários que define o que cada um pode fazer, interfaceamento para acesso de diversas formas, controle de integridade dos dados (ELETRÔNICO, 2016).

Neste projeto, o Mysql serviu basicamente para persistir os dados sobre os usuários do painel administrativo: dados das empresas, das lojas virtuais, dados de configuração gerais do sistema. A ideia em usar dois bancos diferentes, é devido o MongoDB ter um propósito distinto, que é salvar e processar o massivo envio de dados das lojas virtuais. Como o painel administrativo é um módulo distinto dentro do projeto e os dados processados e persistidos por ele são menos volumosos, preferiu-se separar o banco, para trabalhar com esses dados.

[19] Cluster: um sistema que relaciona dois ou mais computadores para que estes trabalhem de maneira conjunta no intuito de processar uma tarefa. Disponível em: http://www.infowester.com/cluster.php

[20] https://www.mongodb.org/community/deployments?group=use-cases

[21] http://www.forbes.com

[22] http://www.hudl.com

[23] https://www.pearson.com

2.2.10 INTELIGÊNCIA DE NEGÓCIOS

Inteligência de Negócios (do inglês *Business Inteligence* BI), refere-se ao conjunto de técnicas e ferramentas que possibilitam transformar dados brutos em informações significativas e inteligíveis com o objetivo de proporcionar uma análise mais eficiente do negócio onde é aplicado. Este campo permite suportar grandes quantidades de dados, para ajudar a identificar, ou criar novas oportunidades de negócios. (OLIVEIRA; PEREIRA, 2008)

Assim, entre os principais objetivos da inteligência de negócios é trazer aos gestores informações de forma que eles possam avaliar a aspectos da empresa, que estavam contidos nesse montante de dados e agregar valor a gestão empresarial. Como também define Turban:

> *Permitir o acesso interativo aos dados (às vezes, em tempo real), proporcionar a manipulação desses dados e fornecer aos gerentes e analistas de negócios a capacidade de realizar a análise adequada. Ao analisarem dados, situações e desempenhos históricos e atuais, os tomadores de decisão conseguem valiosos* insights *que podem servir como base para decisões melhores e mais informadas.* (TURBAN et al., 2009, p.27)

Aplicações de inteligência de negócios podem auxiliar em vários aspectos a empresa, e segundo Ceci (S.V.; PEREIRA, 2009 apud CECI, 2012) podem auxiliar na análise de:

- tendências de transformação do mercado;
- alterações no comportamento de clientes e padrões de consumo;
- preferências de clientes;
- recursos das empresas;
- condições de mercado;

Segundo Cardoso (ATRE; MOSS, 2003 apud BARBIERI; CARDOSO, 2015), define que iniciativas de suporte a decisão inteligência de negócios, deve preencher no mínimo uma das cinco categorias de benefícios entre as citadas:

- Aumento de receitas, que podem vir na forma de identificação de outro nichos de mercado, melhorias nas sugestões de vendas, mais rápido conhecimento de oportunidades, rapidez na comercialização de produtos;
- Aumento de lucro, oriundos das melhora nos focos de metas e campanhas de marketing, identificação de linhas de produtos ou produtos abaixo da linha de performance, identificação de deficiências internas, mais eficiente gerenciamento de *merchandising*;

- Melhora na satisfação do cliente, que podem vir através de um melhor entendimento da preferência do cliente, melhora da customização do produto, mais rápida identificação de problemas dos clientes;

- Aumento de economia por meio da redução de *merchandising* [24] vencida;

- Ganho de *Market-Share* [25] por meio do aumento do número de clientes que abandonam o concorrente, mais alta taxa de retenção do consumidor comparado com os anos anteriores e com a concorrência;

Logo, inteligência de negócios vem ser o recurso do qual as empresas buscam com a finalidades de desafiar o volume de dados multiplicados a todo tempo no ambiente empresarial. Esses desafios que as empresas enfrentam, proporcionam o desenvolvimento de tecnologias que se prontifiquem a ajudá-las.

2.2.11 COMÉRCIO ELETRÔNICO

Comércio eletrônico refere-se ao tipo de transação comercial feita especialmente por meio de equipamentos eletrônicos, computadores, *tablets*, *smartphones* e também dependem da infraestrutura da internet. A empresa que cria um *site* onde disponibiliza seus produtos para comercialização na forma de uma vitrine virtual, com imagens, informações técnicas, preços, formas de pagamento, etc.

Sobre os modelos de comércio eletrônico, pode-se citar, segundo Nascimento (NASCIMENTO et al., 2009), os mais conhecidos:

- B2B (*Business-to-Business*): refere-se as transações que são realizadas entre empresas. Geralmente se trata de um sistema homogêneo com informações em uma linguagem em comum entre as empresas, onde as transações são realizadas;

- B2C (*Business-to-Consumer*): trata-se das transações realizadas entre uma empresa e um consumidor final. Permite a empresa oferecer uma canal diferenciado para atender uma gama maior de clientes, aumentando seu fluxo de vendas;

- C2C (*Consumer-to-Consumer*): é o tipo de relação de vendas entre consumidores finais, não envolve empresa no tramite. Como exemplo desse modelo temos o leilão pela internet, onde uma grande gama de produtos é negociada com pessoas do mundo inteiro;

[24] Técnica de marketing para apresentação de um produto no mercado.
[25] Grau de participação de uma empresa no mercado em termos das vendas de um determinado produto.

- B2G (*Business-to-Governamet*): nesse modelo as transações ocorrem entre as empresas e o Governo. É um tipo de transação que visa reduzir custos, nas operações de compras oficializadas pelas esferas do poder governamental;

Mesmo o comércio eletrônico sendo muito conhecido a *Organisation for Economic Cooperation and Development* (OECD), o definiu de maneira a deixar claro seu conceito:

> *Uma transação de comércio eletrônico é a venda ou compra de bens ou serviços, realizadas através de redes de computadores, por métodos projetados especificamente para a finalidade do recebimento ou da colocação de ordens de compra. Os bens ou serviços são ordenados por esses métodos, mas o pagamento e a entrega final dos bens ou serviços não têm de ser realizadas on-line. Uma transação de comércio eletrônico pode ser entre empresas, famílias, indivíduos, governos e outras organizações públicas ou privadas.*(PUBLISHING, 2011, p.72)

Essa forma de negócio trouxe para as empresas, ferramentas que proporcionam vantagens e melhor competitividade no mercado. Entre essas vantagens o *Instituto Infnet* menciona em uma matéria intitulada “O comércio eletrônico não sabe o que é crise”, bons motivos viabilizados pela relação eletrônica tanto para o cliente como para o comerciante (INFNET, 2015).

Para o comerciante destaca-se:

- Reduz os custos com estrutura física dos estabelecimentos;
- Possibilita vender para um mercado livre da região onde estaria estabelecida a loja presencial;
- Permite a inteligência de relacionamento com consumidores em potencial;
- Fortalece a capacidade de negociação diante da redução do movimento em momentos de crise;

Para o consumidor destaca-se:

- Reduz os custos de aquisição com a possibilidade das pesquisas online;
- Reduz o stress causado pela logística de deslocamento, estacionamento e alimentação;
- Permite o agendamento da entrega com maior facilidade;

Essa forma de negócio apesar de não ter um contato direto com o cliente, pode se beneficiar com o montante de dados gerados do acesso a loja virtual. Acessos que podem ser a fonte de informações para desenvolvimento de campanhas de marketing, propaganda direcionada e atendimento especializado.

2.2.12 INTELIGÊNCIA PARA LOJAS VIRTUAIS

Para uma loja virtual, a inteligência de negócios permite ao gestor tomar decisões por base em informações oriundas de dados reais e atuais, ao contrário de avaliar situações com base em sua intuição empreendedora. Mesmo em pequenos negócios é possível identificar deficiências, rotinas, oportunidades, custos a serem minimizados e traçar estratégias de atuação no mercado. Informações que podem auxiliar em tomadas de decisão nem sempre estão claras e traduzidas em linguagem natural, assim, avaliando o montante de dados gerados pelos clientes, com o auxílio de um sistema computacional que permita padronizar esse montante de dados, e com isso oferecer dados que possam ajudar o gestor (BUENO, 2015).

Dos pontos que se destacam no cenário de uma loja virtual, um dos mais importantes é a oportunidade de analisar o comportamento do consumidor; métricas podem identificar o que as pessoas compram e com que frequência, a média de compras por clientes, ciclo de vendas, relacionamento entre produtos, perfis de clientes. Essa compreensão de comportamento se torna vital para a definição de *Buyer Personas* do inglês "Cliente comprador". *Buyer persona*, é uma representação semi fictícia do seu cliente ideal, baseado em pesquisas de mercado e em dados reais dos clientes. Com dados da demografia dos clientes, padrões de comportamento, motivações e objetivos, oportuniza-se a potencialização de campanhas de marketing digital, por meio do entendimento das melhores formas de abordagem e tipos de conteúdo com maior adesão, promovendo um potencial aumento de acesso de clientes na loja virtual (BELEZA; MEDEIROS, 2015).

Ainda em se tratando de negócios online é possível melhorar as estratégias de *Search Engine Optimization* (SEO)[26] e com isso avaliar técnicas para trazer novos clientes vindo de mecanismos de busca, e assim, melhorar o ranking da loja virtual, nos resultados das pesquisas nesses mecanismos de busca. O Google oferece ferramentas sobre demanda e associação de palavras-chave, porém só com um sistema de inteligência permite entender quais funcionam melhor para o ramo de atuação da loja e público-alvo.

[26]SEO: Otimização para mecanismos de buscas

3 METODOLOGIA

Neste Capítulo descreve-se a metodologia da solução proposta para o problema apresentado. Os passos metodológicos estabelecidos são relacionados e descritos na sequência.

3.1 LEVANTAMENTO DOS REQUISITOS DO SISTEMA

Foram relacionados os requisitos do sistema e descritos na forma de histórias. Essas histórias tem como objetivo demostrar de forma simples como os usuários do sistema vão interagir com o sistema e obter dele as respostas às suas requisições. Nessa etapa há uma descrição sucinta e sem detalhes técnicos de implementação. O Capítulo 4 explana melhor essa etapa.

3.2 IDEALIZAÇÃO DA ARQUITETURA DO SISTEMA

Planejar a forma de coleta, armazenamento e os módulos do sistema, sua inter-relação, comunicação e tratamento dos dados. Aqui foi definido de que forma, basicamente, o sistema deve funcionar. O objetivo de cada módulo, como eles devem trabalhar de forma atender o propósito do sistema. o Capítulo 4 e o Capítulo 5 apresentam melhor como foi trabalhada essa etapa.

3.3 ESTUDO DAS TECNOLOGIAS À SEREM UTILIZADAS

Nessa etapa foram estudadas as tecnologias a serem empregadas. Buscou-se entender de que forma cada tecnologia pode ajudar efetivamente para um resultado satisfatório, com o auxílio de literaturas especializadas pretendeu-se compreender de que forma empregar cada tecnologia. A Seção 2.2 explica com mais detalhes sobre as tecnologias empregadas no projeto.

3.4 TESTAR A APLICAÇÃO EM AMBIENTE DE PRODUÇÃO REAL

Durante o processo de desenvolvimento do sistema de coleta de dados, buscou-se testá-lo em um sistema de comércio eletrônico real, sendo sua posterior análise apresentada no sistema no painel administrativo. Foi possível coletar dados de duas lojas virtuais reais, os resultados e um parecer da experiência podem ser vistos no Capítulo 6.

3.5 ANÁLISE DOS RESULTADOS OBTIDOS E EXPERIÊNCIA NO DESENVOLVIMENTO

Após coletados os dados em um determinado período foi verificado se o sistema cumpriu com sua proposta. Assim, foi verificado a possibilidade em oferecer ao gestor dados de maneira que ele pôde apreciar e tomar por base para agregar valor a sua tomada de decisão. Assim, concluir se o sistema é apto para a ideia proposta ou se fracassa em seu propósito. No Capítulo 7 fica o relato da experiência com o projeto.

4 DESENVOLVIMENTO

Neste capítulo são apresentados as funcionalidades no que concerne ao desenvolvimento do projeto. Foram levantados e desenvolvidos os requisitos do sistema e dispostos na tabela de requisitos funcionais do sistema. Foram também desenvolvidos o diagrama de caso de uso para o painel administrativo e a modelagem do banco de dados. Todo o desenvolvimento inicial e projeção do sistema são explicados nesse capítulo.

4.1 REQUISITOS FUNCIONAIS DO SISTEMA

Na Tabela 2 estão dispostos os requisitos funcionais do sistema. Essa tabela elenca as funcionalidades necessárias para a interação do usuário gestor e do usuário administrativo com o sistema.

Tabela 2: Tabela de requisitos funcionais

ID	Descrição
RF01	O sistema deve ter a funcionalidade de gerenciar o cadastro de usuários administrativos.
RF02	O sistema deve ter a funcionalidade de gerenciar o cadastro de usuários gestores.
RF03	O sistema deve ter a funcionalidade de gerenciar o cadastro de empresas .
RF04	O sistema deve ter a funcionalidade de gerenciar o cadastro de lojas virtuais.
RF05	O sistema deve ter a funcionalidade de gerenciar o cadastro de códigos de rastreamento.
RF06	O sistema deve permitir ao gestor consultar seu código de rastreamento.
RF07	O sistema deve ter a funcionalidade de marcar uma empresa como inativa.
RF08	O sistema deve permitir ao administrador desativar lojas virtuais.
RF09	O sistema deve permitir ao gestor e ao administrador consultar suas métricas mediante relatórios expositivos.
RF10	O sistema deve ser capaz de solicitar de tempo em tempo, a sumarização das métricas .

4.1.1 ATORES E SEUS PAPÉIS NA APLICAÇÃO

O sistema proposto poderá ser usado em diversas plataformas de loja virtual. Nesse sentido, o sistema terá interação com dois atores:

- O papel do **gestor da loja virtual**, permitirá a ele, cadastrar-se no sistema, cadastrar a empresa e as lojas virtuais, consultar o código de rastreamento da loja, consultar as métricas e relatórios sobre sua loja;
- Ao papel do **administrador do sistema**, será permitido, cadastrar gestores, empresas e lojas virtuais, consultar as métricas e relatórios de todos os clientes, excluir e deletar os gestores, empresas ou lojas, consultar os códigos de rastreamento de todos os clientes do sistema, cadastrar códigos de rastreamento para plataforma de loja virtual;

A Figura 1, ilustra seus papéis:

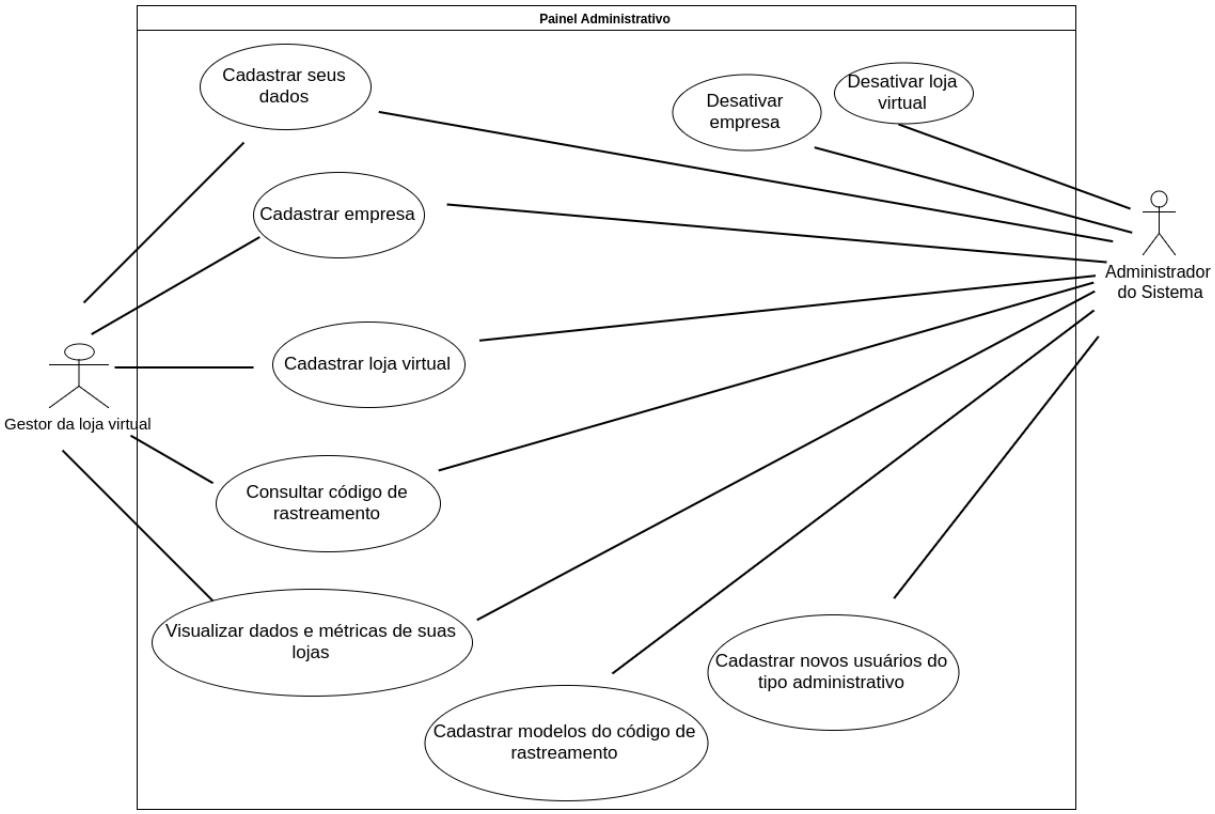

Figura 1: Diagrama de caso de uso do painel administrativo.
Fonte: autor

Assim, a loja virtual pode ser qualquer software *web* que é externo a aplicação proposta, desde que esse software tenha um *script* de rastreamento compatível, que foi previamente cadastrado pelo administrador do sistema. Pois, esse *script*, vai ser a ponte de ligação para extrair do sistema da loja virtual os dados que são preparados para serem coletados.

Há uma diferença, entre cadastrar a empresa e cadastrar a loja virtual, nesse caso, o cadastro da empresa é onde ficam registrados a razão social, o número do cadastro de pessoa jurídica, nome fantasia. Nada impede que uma empresa tenha várias lojas virtuais, por exemplo uma empresa com uma gama de produtos grande, pode querer separar os departamentos de produtos em lojas virtuais distintas, uma loja para infanto-juvenil, outra para moda casa, outra para calçados; nesse sentido é a mesma empresa, mas com lojas virtuais separadas, com comportamentos, acessos e endereço *web* distintos, que estão ligadas a mesma empresa. Com isso, surge a necessidade em ser possível cadastrar mais de uma loja virtual, com identificadores separados, para a mesma empresa.

4.2 DINÂMICA DE FUNCIONAMENTO

Aqui é detalhada a dinâmica de funcionamento do sistema e sua ideia arquitetural. A Figura 2 apresenta uma ideia do funcionamento.

No procedimento número **1**, o gestor da loja virtual vai acessar o sistema do módulo administrativo, realizar seu cadastro de usuário, cadastrar a empresa e realizar o cadastro da loja virtual. Concluído o cadastro ele vai ter acesso ao código de rastreamento da loja virtual e ao código identificador único da loja virtual o "webstore_id". Uma empresa pode ter vários domínios de loja virtual, por isso para cada loja virtual vai ter um identificador único.

No procedimento número **2**, o gestor da loja virtual, em posse do código de rastreamento vai inserí-lo na página da loja, no documento HTML, preferencialmente no rodapé. Esse código como mostra a **Figura 3**, é um objeto em Javascript, aqui denominado "BI", que é composto de chaves e valores com os dados do cliente, da página acessada, dos produtos adicionados no carrinho, dos produtos comprados, total da compra, total do frete. No fim do código de rastreamento tem um bloco de código que serve como um gatilho, responsável de fazer a requisição do *script* de coleta de dados para a API, que validará esses dados e enviará para a API para que o passo de persistência dos dados seja realizada.

Esse *script* tem três chaves que são semelhantes no seu conteúdo, são eles: *cart_products*, *checkout_products* e *conversion_items*. Ambos mantém dados dos produtos em determinados momento da navegação da loja. Isso se dá em razão da possibilidade de rastrear, por exemplo, quantos produtos foram adicionados ao carrinho e quantos foram efetivamente comprados e com isso identificar problemas com o preço do frete, entre outros.

No procedimento número **3**, com o código de rastreamento inserido na loja virtual, a cada acesso de um cliente, o objeto JavaScript vai estar preenchido com os dados oriundos

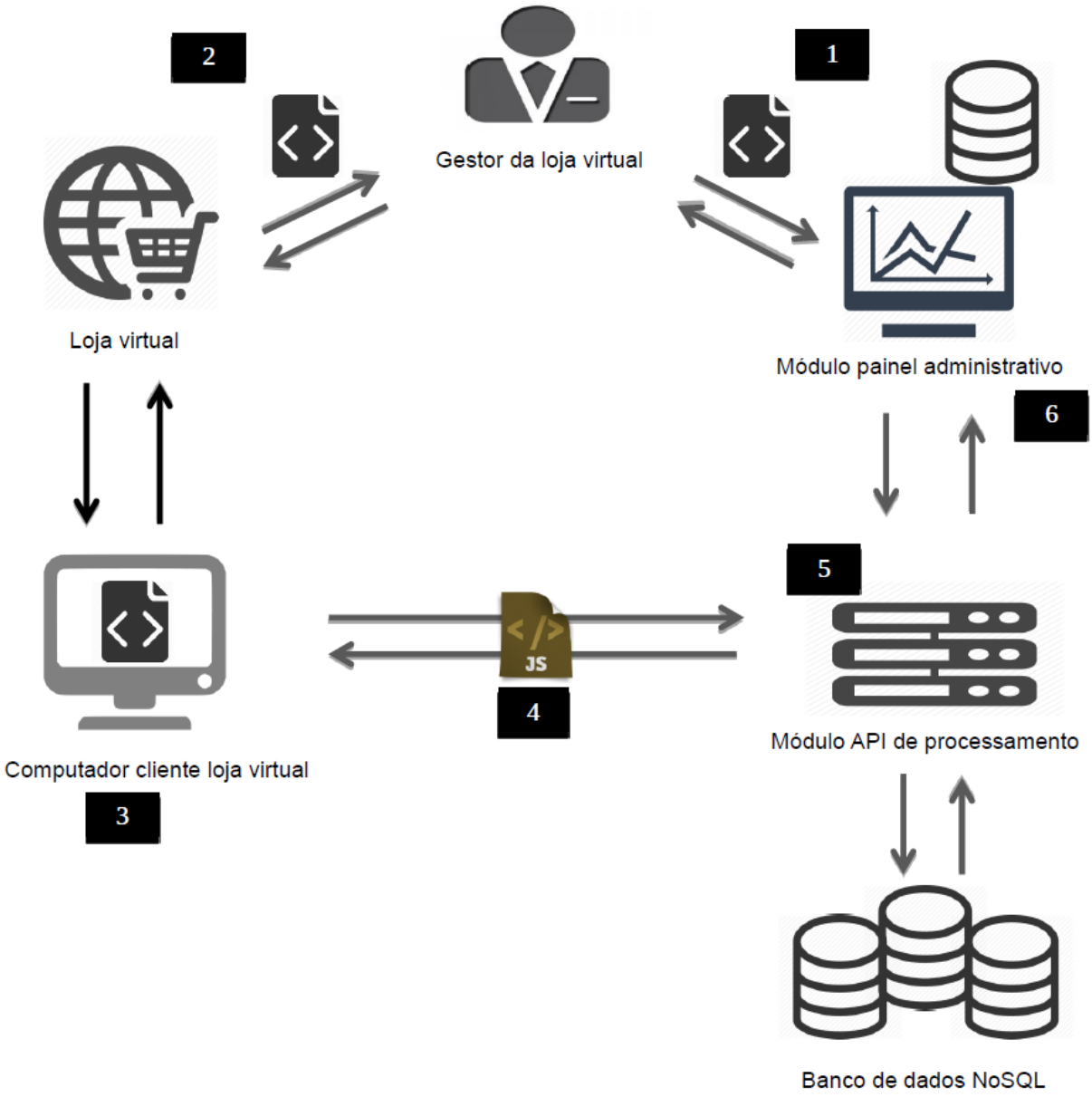

Figura 2: Dinâmica de funcionamento do sistema proposto.
Fonte: autor

daquele acesso, por exemplo, se o cliente está logado conterá seu *e-mail*; se o cliente está finalizando uma compra, conterá informações de quais produtos estão em seu carrinho, quantidade e valores, assim, quando a página da loja virtual for carregada no navegador do cliente, o bloco de código contido no fim do código de rastreamento vai executar a chamada para à API solicitando o *script* de coleta de dados.

Para o correto funcionamento do código de rastreamento, é importante que o gestor preencha os valores das chaves, com os métodos que a plataforma de loja virtual usada por ele dispõe. Logo, o sistema pode fornecer o código de rastreamento já com os métodos prontos

```
<script type="text/javascript">
    var BI = {
        "customer_name" : "###",
        "customer_email" : "###",
        "cart_products":[
            {
                "sku" : "xxx",
                "qty" : "xxxx",
                "name" : "xxx",
                "price" : "xxx.xx",
                "url" : "xxxx",
                "image" : "xxxx"
            }
        ],
        "checkout_products":[
            {
                "sku" : "xxxx",
                "qty" : "x",
                "name" : "xxxxxxx",
                "price" : "xxx.xxx",
                "url" : "xxxxx",
                "image" : "xxxxxxx"
            }
        ],
        "conversion_items":[
            {
                "sku" : "xxxxx",
                "qty" : "xxxx",
                "name" : "xxxxx",
                "price" : "xxx.xx",
                "url" : "xxxx",
                "image" : "xxxx"
            }
        ],
        "conversion_order_id" : "xxxxxxx",
        "conversion_shipping" : "xxxx.xx",
        "conversion_total":"xxxxxx"
    }
    (function(){
        BI.webstore_id = 'xxxxxxxxxxxxxxxxxx';
        var _dcs = document.createElement('script');
        _dcs.type = 'text/javascript';
        _dcs.async = true;
        _dcs.src ='10.0.0.1/bi_tracking.min.js';
        document.getElementsByTagName('head')[0].appendChild(_dcs);
    })();
</script>
```

Figura 3: Exemplo do código de rastreamento.
Fonte: autor

para plataformas de loja virtuais mais usadas, como por exemplo o *Magento Community*;

Na sequência, no procedimento número **4**, o *script* de coleta de dados, após ser requisitado, vai executar sem que o cliente da loja virtual perceba e de maneira assíncrona, a coleta dos valores contidos no objeto “BI”, e também vai adicionar a este informações extras, como tamanho da tela, tipo de dispositivo usado, endereço da página atual, endereço da página anteriormente acessada. Essas informações extras estão presentes no objeto *document* nativo no Javascript em cada navegador.

Após coletar esses dados, o código vai executar a validação deles, removendo valores não presentes e preparando um outro objeto, que será enviado à API para persistência. Todo esse processamento de coleta e validação de dados é realizado no computador do cliente, com isso se reduz a carga de processamento na API de coleta de dados. Esse processamento não sobrecarrega a máquina do cliente, pois se trata de um pequeno código Javascript.

Finalizando a etapa de coleta de dados no procedimento número **5**, a API vai receber os dados já validados para serem salvos no banco. Desse modo cabe à API somente verificar se a loja virtual está ativa no sistema e responder o *status* da requisição, com sucesso ou erro. O MongoDB salva os objetos como documentos que vem na forma de objetos JSON e são serializados para BSON. A Figura 4 mostra um exemplo de como os dados chegam até o banco para serem persistidos.

```
1
2  {
3      "webstore_id":"32d559c63e81159c63e8c63e8",
4      "customer_name":"Ludwig von Mises",
5      "customer_email":"ludwigvon@mises.com",
6      "cart_items":[
7          {
8              "sku":"23422N",
9              "qty":2,
10             "name":"Product example",
11             "price":"999.9000",
12             "url":"http://www.examplestore.com.br/product-example.html",
13             "image":"http://www.examplestore.com.br/media/catalog/product/product_example.jpg"
14         }
15     ],
16     "ua":"Mozilla/5.0 AppleWebKit/537.36 Chrome/46.0.2490.80 Safari/537.36 OPR/33.0.1990.58",
17     "referrer":"http://www.examplestore.com.br/product-example-before.html",
18     "path": "http://www.examplestore.com.br"
19     "screen":"1366x768"
20 }
21
```

Figura 4: Exemplo de documento JSON para ser persistido no MongoDB.
Fonte: autor

O banco onde serão armazenados os dados coletados fica fora do sistema servidor do cliente. Essa dados ficam armazenados em um servidor que serve como o repositório de dados, para que possam ser trabalhados a parte, sem prejudicar o desempenho da loja virtual do gestor.

Após algum tempo, rotinas são executadas quando o gestor acessa o painel administrativo, solicitando à API que processe as métricas de forma a gerar as informações; estes alimentam relatórios e gráficos no sistema. Assim, quando o gestor acessar o módulo administrativo e desejar verificar alguma métrica, o módulo administrativo vai fazer uma requisição para a API de coleta e processamento de dados, solicitando os dados que vai alimentar o relatório para apreciação do gestor.

4.3 BANCO DE DADOS - PAINEL ADMINISTRATIVO

O sistema vai contar com um banco de dados relacional, que servirá para gerenciar os dados referentes ao cadastro dos gestores, dos administradores, e outras informações de apoio. Nesse banco não ficarão armazenados dados oriundos da API referentes as métricas apuradas, mas somente informações pertinentes ao painel administrativo. A Figura 5 mostra o conceito desse banco:

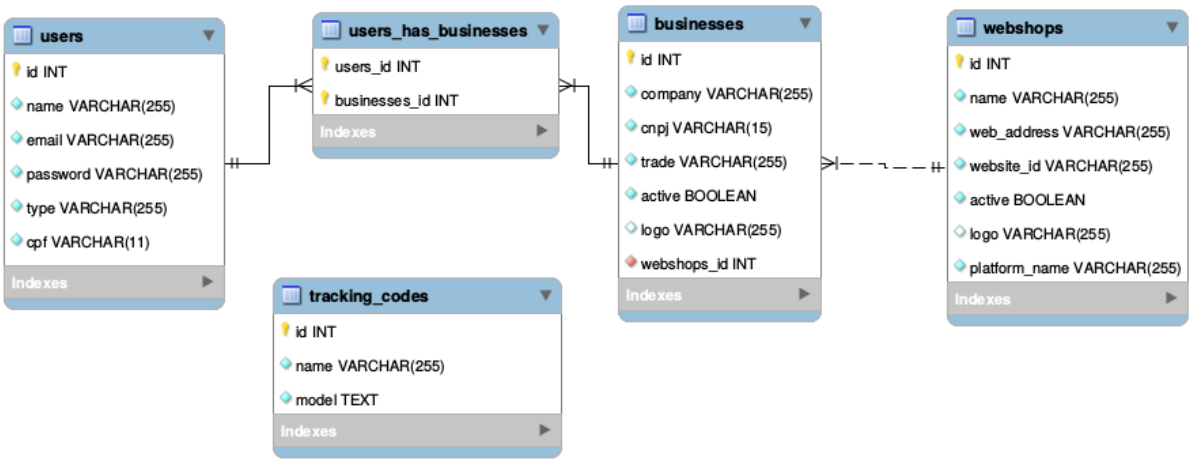

Figura 5: Banco de dados painel administrativo.
Fonte: autor

A tabela *users* é responsável pelo armazenamento dos dados dos usuários do sistema: nome, *e-mail*, cadastro de pessoa física e tipo. O campo *type* vai ser útil para controle de acesso e permissões dos usuários quanto as rotinas do sistema.

Cada usuário pode estar relacionado com várias empresas, assim, a tabela *businesses* será responsável pelo armazenamento dos dados referentes a empresa: nome, nome fantasia, cadastro de pessoa jurídica e se a empresa está ativa; O campo *active* permitirá ao administrador do sistema, bloquear ou desbloquear a empresa e suas lojas virtuais.

Como vários usuários podem ter várias empresas e empresas podem pertencem a vários usuários, a tabela *users_has_businnesses*, serve para controlar melhor essa relação *n:m*.

Cada empresa pode ter atrelada a si várias lojas virtuais, a tabela *webshops* será responsável pelo armazenamento dos dados da loja virtual em específico: como seu código identificador único o *webstore_id*, o nome da loja, o nome do domínio da loja e se a loja está ativa. O campo *active* será usado para identificar se a loja está ativa, isso permitirá ao administrador bloquear definitiva ou temporariamente uma loja em específico.

A tabela *tracking_codes*, será responsável pelo armazenamento dos códigos de rastrea-

mento. O gestor poderá verificar se já existe o código de rastreamento com os métodos prontos para sua plataforma de loja virtual ou poderá retirar um código como demonstrado na Figura 3. Esses códigos serão cadastrados pelo administrador do sistema.

4.4 DIAGRAMA DE SEQUÊNCIA DO CADASTRO DO GESTOR

Para ilustrar como ocorre os procedimentos do cadastro do gestor, apresenta-se o diagrama de sequência visto na Figura 6

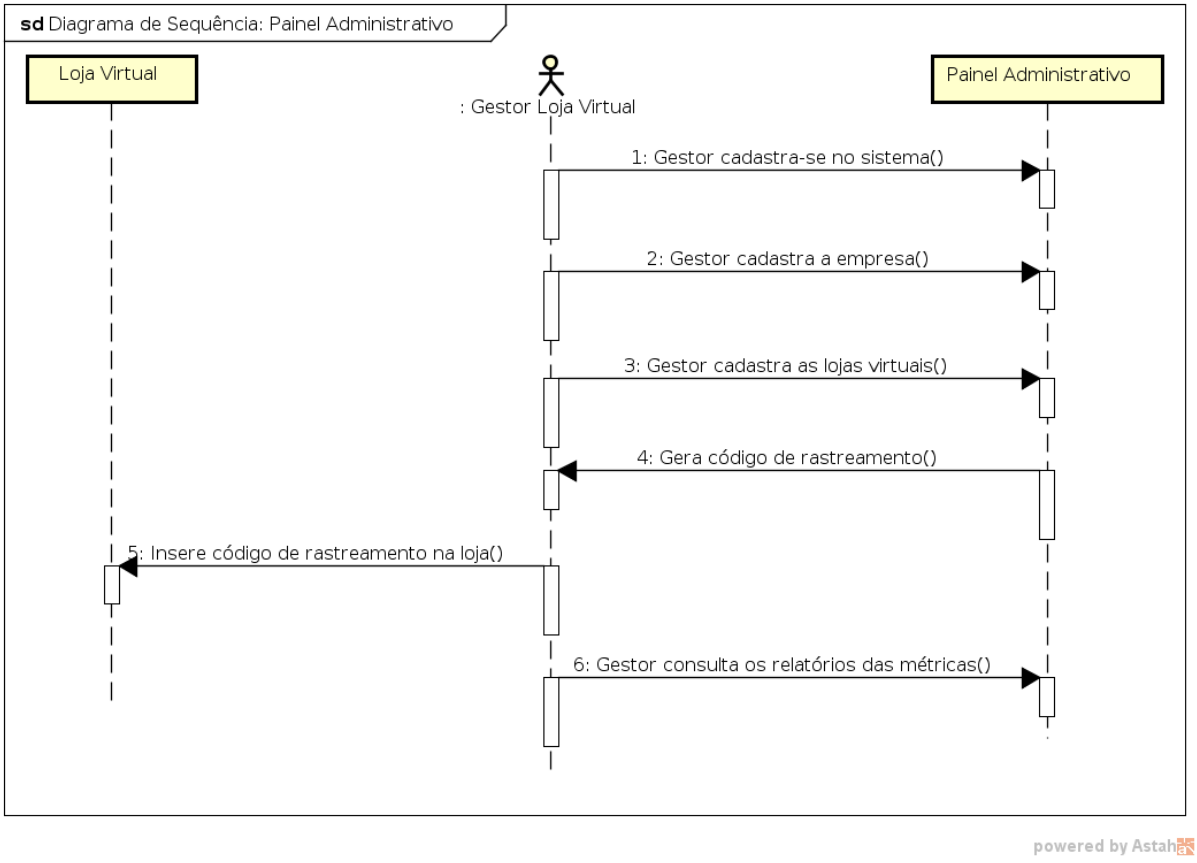

Figura 6: Diagrama de sequência cadastro do gestor.
Fonte: autor

O gestor no seu primeiro uso do sistema deverá acessar o painel administrativo, para realizar o cadastro de seus dados pessoais, seu *e-mail* e senha serão usados para futuros acessos ao painel administrativo.

Após cadastrar-se o gestor estará habilitado para cadastrar uma ou mais empresas. Com o cadastro das empresas concluído será necessário cadastrar as lojas virtuais. Ao término do cadastro da loja, o sistema redirecionará o gestor para acessar o código de rastreamento e esse código poderá ser consultado a qualquer momento.

Em posse do código o gestor irá inserí-lo na loja virtual e aguardar até que o sistema tenha dados para processar. Com isso, ao acessar o painel administrativo ele poderá verificar as métricas baseadas nos dados coletados dos acessos dos clientes na loja virtual.

4.5 DIAGRAMA DE SEQUÊNCIA DA COLETA DE DADOS

Para ilustrar como ocorre a coleta de dados, apresenta-se o diagrama de sequência visto na Figura 7.

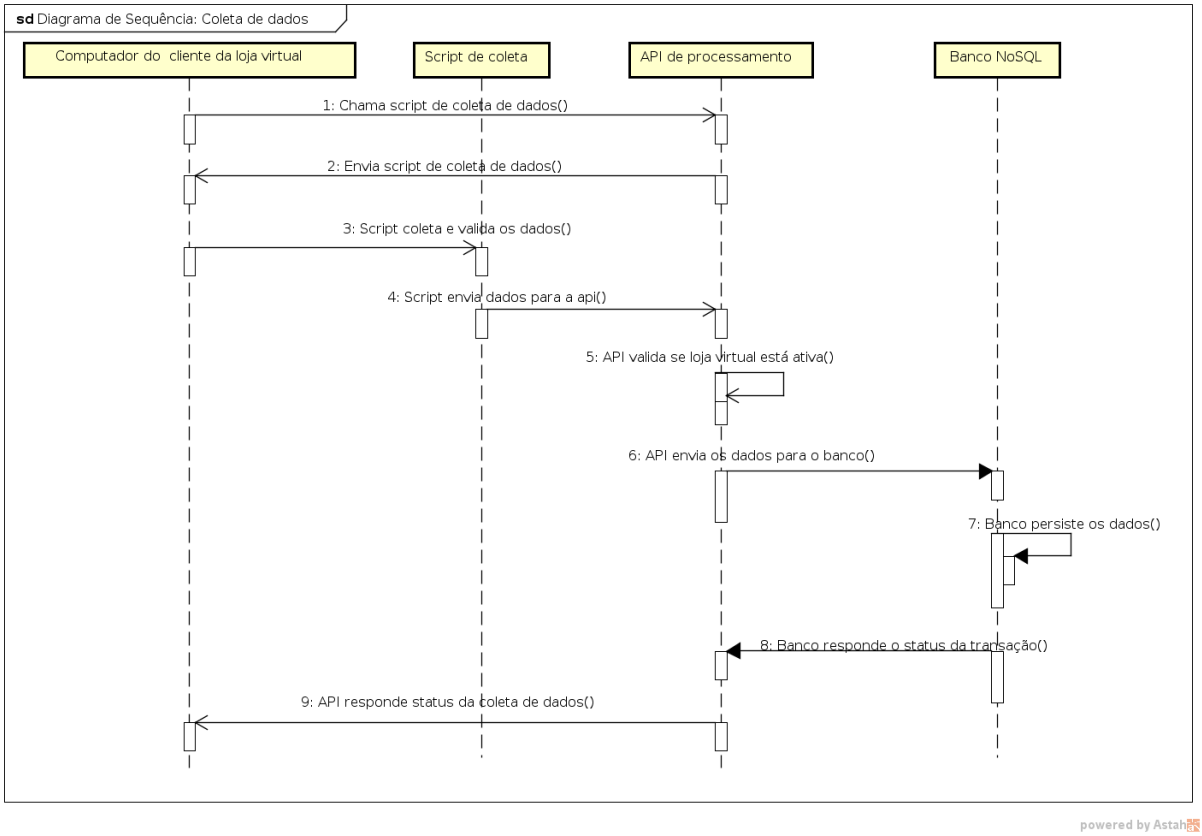

Figura 7: Diagrama de sequência da coleta de dados.
Fonte: autor

Quando o cliente acessar alguma página da loja virtual, o código de rastreamento anexo ao HTML da página, será executado quando o navegador do cliente processar a página. Isso irá disparar uma requisição a API de processamento, que enviará o *script* de coleta. Esse *script* escrito em Javascript vai ser processado no computador do cliente, validará os dados do código de rastreamento e preparará os dados em um objeto JSON para ser enviado à API.

Quando a API de processamento receber o objeto JSON, mediante o valor de *webstore_id* que estará presente nesse objeto, vai verificar se essa loja está ativa, e em caso afirmativo vai enviar esse objeto JSON para o banco de dados. O banco garantindo que salvou o docu-

mento responde à API, que por sua vez responderá a requisição de origem com sucesso ou erro.

Esse processo se repetirá para cada página acessada, a cada cliente, em cada loja virtual. Todo o procedimento será assíncrono, logo qualquer desventura no processo, não afetará a experiência de navegação do cliente com travamentos de página ou aba de navegador.

5 DESENVOLVIMENTO DO SISTEMA

Após a compreensão das funcionalidades, deu-se início a fase de desenvolvimento do sistema. Essa fase foi dividida da seguinte maneira: primeiro foi feito o código para ser inserido na loja, seguido da API de coleta de dados e *script* de rastreamento e por último o painel administrativo dos gestores. A API e o painel administrativo tiveram partes que foram desenvolvidos concomitantemente, pois, um dependia de funcionalidades do outro, principalmente no que se refere a exibição dos relatórios.

5.1 SCRIPT DA LOJA VIRTUAL

Esse *script* tem como objetivo preparar os dados para serem coletados quando o cliente requisita a página da loja virtual. Esses elementos são exclusivamente gerados pela plataforma da loja, assim que o servidor processar o pedido, ele gera os dados de acordo com a página acessada. Basicamente são informações referente ao *e-mail* do cliente logado, produtos adicionados no carrinho, produtos comprados efetivamente, número do pedido gerado, valor do frete, valor total em compras.

Assim, ao renderizar a página o navegador cria um objeto Javascript chamado “BI”, quando acontece essa renderização, o *script* da loja virtual possui um gatilho que requisita assincronamente ao servidor onde está a API de coleta de dados, o *script* que vai trabalhar os dados do objeto BI, esse passo é melhor explicado no capítulo seguinte. Exemplos desses objetos Javascript estão no apêndice A.

Ao acessar o painel administrativo e fazer o cadastro da loja virtual, quando o gestor acessar o menu do código de rastreamento, ele vai poder escolher qual loja e em qual plataforma gerar o código. A Figura 8 retrata como é feita a escolha desse código. Após a escolha, ele tem acesso ao código para copiar e inserir em sua loja virtual. A Figura 9 mostra a tela onde se visualiza o código de rastreamento.

Como existem várias plataformas de lojas virtuais, escritas em diferentes linguagens,

é preciso estar à disposição do gestor vários *scripts*, para que ele possa escolher qual é o correto para a sua loja. Cada *script*, vai estar preparado com os métodos certos para serem usados naquela plataforma que foi escolhida. Assim, basta somente ser inserido no código fonte da loja virtual. Para fins de testes, foi gerado somente o código que pode ser usado nas lojas cuja a plataforma é *Magento Community* na versão 1.9. O código completo é demonstrado no apêndice B

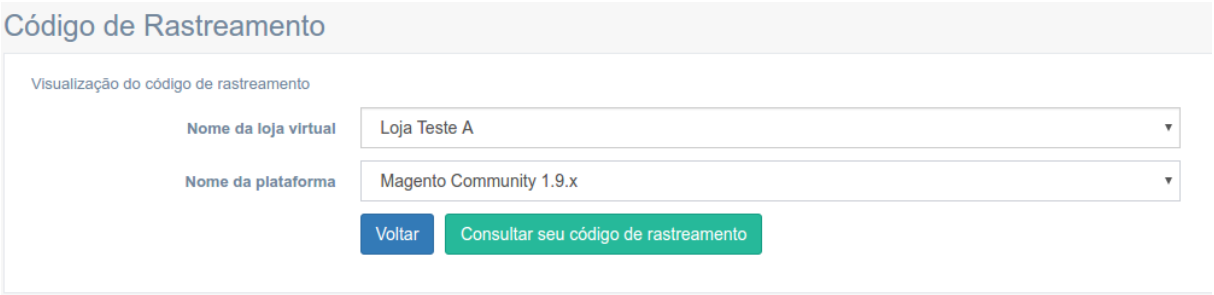

Figura 8: Tela de escolha do código de rastreamento.
Fonte: autor

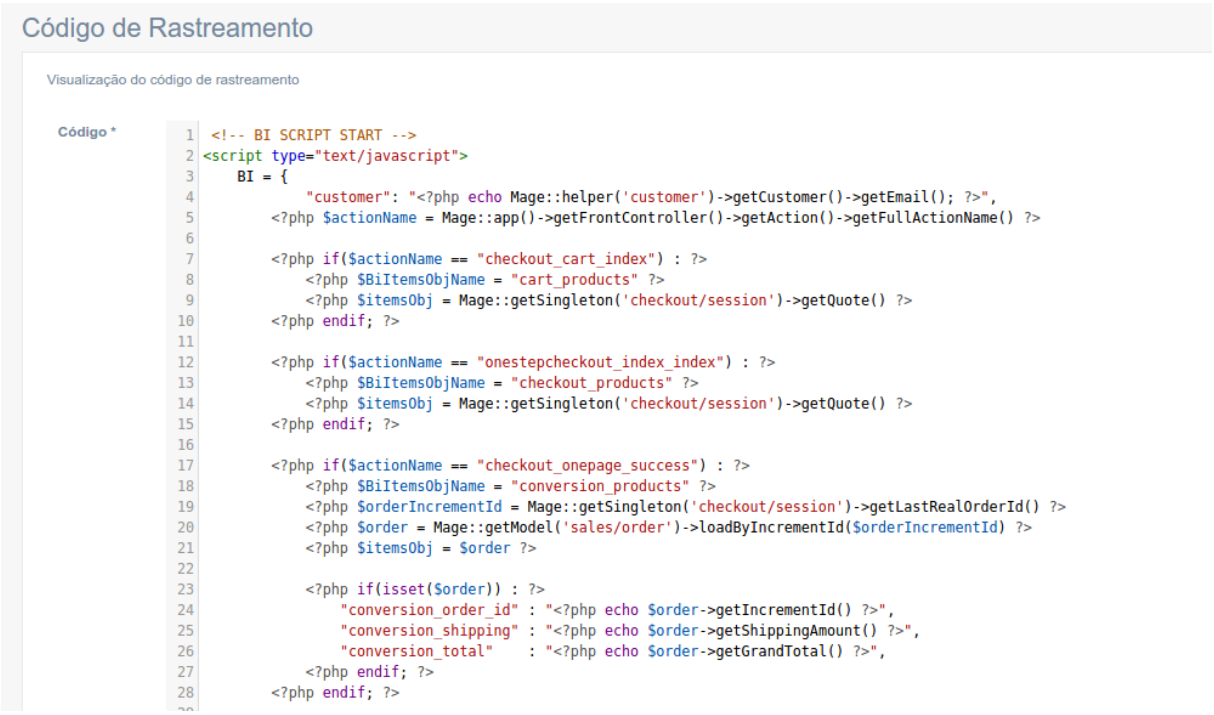

Figura 9: Tela de visualização do código de rastreamento.
Fonte: autor

5.2 SCRIPT DE COLETA DE DADOS

Como mencionado, o *script* da loja virtual, possui um gatilho que requisita a API o *script* responsável pela coleta de dados. A Figura 10 mostra o trecho de código que executa

essa ação. Esse código adiciona ao corpo do documento da página HTML o arquivo requisitado. Na resposta dessa requisição é recebido um código exclusivamente em Javascript, chamado "tracking.js" que é executado pela máquina cliente assim que é carregado pelo navegador.

```
(function(){
    BI.website_id = 'f74a24d09157bb8e7dd1435';
    var _dcs = document.createElement('script');
    _dcs.type = 'text/javascript';
    _dcs.async = true;
    _dcs.src ='https://www.enderecoapi.com.br/collector/tracking.js';
    document.getElementsByTagName('head')[0].appendChild(_dcs);
})();
```

Figura 10: Código do gatilho que requisita o script de coleta de dados.
Fonte: autor

O *script* de coleta de dados, incrementa o objeto "BI", usando informações disponíveis pelo navegador do usuário cliente da loja virtual e pelas APIs do HTML 5, essas informações incluem: url da página atual, tamanho da tela, tipo de agente de usuário usado, página anteriormente acessada, data e hora, tempo de resposta do servidor, tempo de carregamento da página, título da página, plataforma usada. Assim que o objeto é incrementado, o *script* faz um tratamento prévio dos dados para garantir a consistência deles e os envia para a API de coleta que os armazena no banco de dados não-relacional.

Esse processo de coleta, tratamento e envio se dá no momento que o navegador carrega e executa o *script*. Como o processo de execução é assíncrono, caso aconteça algum problema na execução do código, não interfere na navegação do usuário, sem causar travamentos ou impedir a execução de outros códigos Javascript.

Para enviar os dados é usado *XMLHttpRequest*, que permite enviar e receber dados diretamente para um servidor, sem precisar recarregar a página, com isso o cliente da loja virtual, nem percebe que esse procedimento está acontecendo, o que permite seguir com sua experiência de compra tranquilamente. Antes de enviar os dados, o objeto é convertido para uma string de dados e enviados usando o método POST do protocolo HTTP.

Esse procedimento é eficiente pois, o processamento maior do objeto "BI" acontece no computador do cliente da loja virtual, assim, os dados chegam prontos para serem salvos na API, ficando a égide dela somente salvá-los no banco de dados. Cada computador cliente executando e validando seus próprios dados antes de serem enviados, permite desafogar a API de processamentos extras.

5.3 API DE COLETA DE DADOS

O ponto nevrálgico para o funcionamento do sistema como um todo ficou a cargo da API de coleta de dados. Ela não somente coleta os dados como o nome sugere, mas tem outras atribuições, como fornecer os arquivos e dados sumarizados. Todas essas tarefas se originam de uma série de requisições, que podem somar centenas de milhares delas feitas em um único dia. Com isso, para atender melhor essa tarefa foi usado Node.js no desenvolvimento da API, que trabalha com arquitetura não bloqueante e assíncrona, e por isso responde bem a milhares de requisições.

A escolha de Node.js para servir como sistema servidor se dá pelo fato dele atender bem um grande montante de requisições, na Figura 11 apresenta-se um comparativo onde se observa as características ao usar Node.js para atender requisições *web*.

DADOS	**NODE.JS**	**PHP+APACHE**
Nível de Concorrência	1000	1000
Tempo Total	4.061 segundos	49.219 segundos
Requisições Completas	20000	20000
Requisições Falhas	0	2427
Requisições por segundo	4924.91 segundos	406.34 segundos
Tempo por requisição	203.049 milissegundos	2460.973 milissegundos

Figura 11: Dados do teste de desempenho.
Fonte: Utilização da Tecnologia Node.js para consumo eficiente dos recursos de servidores *web* (QUIXABEIRA et al.)

Ainda sobre o desenvolvimento da API, usou-se o *framework* Express.js, que busca facilitar o desenvolvimento de aplicações *Web* usando Node.js. O Express.js propicia a criação de servidores *web* para atender a requisições HTTP de maneira simples. Proporciona a criação de um conjunto de diretórios com um estrutura padrão, para organizar o arquivos de rotas, modelos e *views* da aplicação (BONFIM; LIANG,).

A API de coleta de dados fornece o arquivo estático, o “tracking.js”, que é requisitado pelo *script* da loja virtual. Nesse ponto basicamente não existe um processamento complexo, somente o sistema serve o arquivo sem maiores problemas.

Ainda cabe a API salvar os dados enviados pelo *script* de coleta, nesse ponto como os dados já estão validados e já foram pré-processados, não há necessidade de outras validações, ficando somente a responsabilidade de persistir os dados no banco não-relacional. Para persistir e acessar os dados no MongoDB, é usado um *driver* chamado *mongojs* [1], bastando somente

[1]Mongojs módulo: https://www.npmjs.com/package/mongojs

configurar o caminho e dados de acesso ao banco. Esse driver é um módulo para ser usado com Node.js, que facilita a integração do sistema com a interface do MongoDB.

Assim que a loja já tenha dados coletados, quando o gestor acessar o sistema para conferir os relatórios e gráficos, o painel administrativo faz requisições para a API solicitando essas informações. Essas requisições são rotas específicas que atendem cada uma um determinado tipo de informação. O painel administrativo requisita à API, que por meio do verbos HTTP, GET ou POST, ao receber o pedido conecta-se ao banco para processar a consulta, o banco retorna os dados a API que prepara-os no formato JSON, que então é enviado ao painel administrativo, que está preparado para apresentá-los ao gestor.

5.4 BANCO DE DADOS NÃO-RELACIONAL

O MongoDB, banco de dados não-relacional, tem a tarefa de persistir os dados e executar as consultas para extrair as informações. Esse banco salva os dados na forma de documentos, ou seja, não existe tabelas e relacionamentos nem normalização, cada documento é único; essa característica é importante pois, cada requisição feita pelo cliente e que é salva no banco, pode conter campos diferentes.

A escolha do MongoDB se deu por conta da possibilidade de flexibilidade e escalabilidade que o modelo não-relacional permite, junte-se a isso a capacidade de aumentar a carga de processamento sem qualquer tempo de inatividade. Como a ideia é persistir dados oriundos de centenas de lojas virtuais, vindos de centenas de milhares de clientes, é preciso um banco que possa crescer e processar com facilidade esses dados, usando *sharding*[2] por exemplo, para atender essa necessidade de processamento e performance (PESSOA et al., 2012).

5.5 PAINEL ADMINISTRATIVO

O painel administrativo é onde acontece toda a interação direta com o usuário do sistema, nesse caso são os gestores das lojas virtuais ou o gestor administrativo do sistema.

Para tanto, o painel administrativo tem duas divisões básicas o acesso geral que o administrador do sistema tem, com acesso a todas as informações de todas as lojas e o acesso do gestor da loja virtual, limitado a sua empresa e lojas cadastradas.

[2]Sharding: é uma técnica muito usada atualmente para lidar com escalabilidade de massas de dados, consiste basicamente em dividir os dados de uma aplicação entre vários bancos: por exemplo, numa aplicação com 1000 usuários o sharding faria que os usuários com nomes de A a J ficassem em um servidor, e de K a Z em outro servidor. http://core.eti.br/2010/11/14/mongodb-sharding-e-mapreduce/

Para dar início ao uso do sistema, o gestor precisa se cadastrar no sistema administrativo para ter um usuário de acesso, cadastrar os dados da empresa e cadastrar uma loja virtual vinculada a essa empresa, ao fazer isso o gestor vai ter acesso a lista de códigos de rastreamento. Nessa lista ele precisa ter conhecimento da plataforma usada por sua loja e a versão, para escolher a correta para sua loja.

Com o código de rastreamento em mãos, ele precisa inserir no código fonte da loja virtual, preferencialmente antes do fechamento da *tag body*; aqui é o momento que ele pode precisar de auxílio técnico, para todos os fins esse procedimento só precisa ser feito uma vez, desde que a loja não sofra alguma atualização de versão ou um *upgrade* que cause incompatibilidade com o código de rastreamento.

Ao cadastrar a loja virtual, ela recebe um identificador único, que está presente no objeto “BI”. Assim que o *script* estiver devidamente inserido na loja, dependendo do volume de acessos, dentro de algumas horas o gestor já vai ter acesso a algumas informações sobre seu negócio virtual.

O painel administrativo permite gerenciar os dados cadastrados do cliente, das empresas e das lojas. Se o gestor tem mais de uma loja cadastrada ele pode trocar entre elas sem precisar fazer um novo *login*, basta simplesmente acessar o menu de troca de loja.

5.6 PAINEL ADMINISTRATIVO - ÍNDICES INICIAIS

No painel administrativo o gestor da loja virtual, deve primeiro efetuar o cadastro do seu usuário para aceder a tela de *login* e ter acesso as funcionalidades do sistema. Ali é preciso fazer o cadastro de uma empresa e da loja virtual, para ter acesso ao código de rastreamento. Nesses cadastros ele pode editar ou excluir os dados tanto da empresa como da loja virtual.

Após os procedimentos dos cadastros e da colocação do código de rastreamento na loja virtual, em algumas horas, o gestor já passa a ter informações no painel administrativo. No menu lateral esquerdo vai ter acesso as métricas, que são apresentadas por gráficos e tabelas e na tela de início na parte superior, acesso a índices da loja.

Ao acessar o painel administrativo, a primeira página trás algumas informações na forma de índices, essas informações são explicadas a seguir:

- **Tempo de resposta do servidor**: Aqui é possível verificar o tempo médio em milissegundos de resposta do servidor da loja virtual para cada requisição nos últimos 7 dias. Permite perceber se o tempo de resposta está satisfatório. Problemas devido ao mal funci-

onamento do *hardware*, falhas no *link* de internet, aumento no gargalo de processamento devido ao elevação natural de acessos ao *site*, podem ser sentidos com esse índice. Segundo o *site E-Commerce News*[3], uma loja cuja a resposta por página seja superior a 5 segundos, já pode ser considerado lento. Essa lentidão é frustrante para o cliente que busca praticidade nas compras *on-line*;

- **Total de requisições**: Nesse item o gestor tem uma ideia de como anda o tráfego em sua loja virtual, essas requisições demonstram o interesse do cliente na loja virtual. Quanto mais sensível o aumento do tráfego é um bom indicativo de clientes procurando a loja virtual, buscando e avaliando produtos ou também pode ser um bom termômetro a resposta de campanhas de *marketing*;

- **Total em pedidos**: Esse indicativo é somente um totalizador para o gestor ter acesso a quantia de venda no mês, normalmente as lojas virtuais tem totalizadores de venda. Como é uma informação útil é interessante concentrar em um painel somente certas informações de utilidade;

- **Taxa de conversão**: esse índice demonstra ao gestor qual é o percentual de clientes que realmente compram na loja, em relação ao total de visitantes. Quanto mais próximo de 100% indica que para cada visitante a chance de efetivamente haver uma venda é grande; Com essa métrica é possível monitorar o desempenho da loja e os reais resultados obtidos por ela, com campanha publicitárias, promoções, entre outros segundo o *site Mercado do Ecommerce*[4];

- **Abandono de Carrinho**: esse índice indica o percentual de clientes que adicionaram produtos no carrinho em relação aos que efetivamente compraram. É semelhante a Taxa de conversão. Isso é bom indicador para perceber clientes que intencionaram a comprar, mas por alguma razão não compraram. Ajuda ao gestor, a compreender se existe ou não uma deficiência e criar mecanismos motivadores, por exemplo: como cupons de desconto, frete grátis, para que o cliente sinta-se motivado a seguir o processo de compra e com isso melhorar as vendas; Quanto mais alto o valor é um sinal ruim para a loja;

- **Abandono de Checkout**: o *checkout* é o último passo para concretização de uma venda na loja virtual. Nesse índice o gestor vai notar, que o cliente estava bem motivado a

[3]E-Commerce News: "Velocidade: requisito de sucesso para plataformas de e-commerce". Acesso: https://ecommercenews.com.br/noticias/pesquisas-noticias/velocidade-requisito-de-sucesso-para-plataformas-de-e-commerce em 15/11/2016

[4]Mercado do Ecommerce: Guia do iniciante: O que é taxa de conversão e qual a sua importância .Acesso: http://www.mercadoecommerce.com.br/guia-do-iniciante-o-que-e-a-taxa-de-conversao-e-qual-a-sua-importancia em 15/11/2016

comprar, mas devido por exemplo, a falta de métodos de pagamento, falta de indicadores de segurança na página, formulários complexos entre outros, fizeram com que o cliente abandonasse a compra no seu momento final; Quanto mais alto esse valor se torna preocupante para o gestor, visto que o cliente realmente tinha intenção de comprar, mas por algum motivo ele não efetivou a compra;

Esses índices, servem como um termômetro para que o gestor, consiga perceber o andamento da loja virtual, caso ele identifique alguma deficiência, que possa tomar as medidas necessárias para garantir a melhor experiência de navegação ao seu cliente. Ainda com esses índices, é possível perceber quase que imediatamente, se suas medidas sortiram efeitos ou se ainda precisam ser trabalhadas novas soluções.

5.7 PAINEL ADMINISTRATIVO - MENU DE MÉTRICAS

O menu de métricas contém dados que por serem mais complexos demandam uma exibição mais detalhada, por gráficos ou tabelas, por isso a necessidade da exibição particular dessas informações em páginas específicas.

Como métricas propostas pelo sistema apresenta-se:

- **Informativo de produtos comprados juntos**: Essa métrica visa mostrar ao gestor a relação de produtos, que são comprados juntos. Para ficar mais compreensível, por exemplo, ao comprar um mouse alguns clientes comumente compram um *mousepad*. Assim essa informação pode ser útil para agregar vendas; um novo cliente que ao adicionar ao carrinho um mouse, porque não sugerir a ele um *mousepad*? Com essas informações, é possível aumentar as vendas, na medida que o sistema forneça essa relação de produtos, o gestor pode trabalhar de forma que certos produtos sejam sugeridos ao consumidor final dependendo de sua escolha;

- **Perfis de clientes**: em campanhas de *marketing* ou campanhas promocionais, o gestor pode priorizar certos grupos de clientes. Buscando fidelizar clientes que compraram somente uma vez na loja, tentar atraí-los ou valorizando-os com promoções especiais. Nesse sentido essa métrica, visa propiciar essa capacidade ao gestor, de informar quais clientes compram mais e com mais frequência, classificando-os em grupos distintos;

- **Caminhos comuns de clientes**: essa informação visa mostrar ao gestor, qual é a trilha mais comum que seus clientes fazem. Em uma loja física, é perceptível ao gerente ou

dono do estabelecimento perceber qual produto ou sessão é a mais procurada, ou se os clientes somente entram na loja dão uma olhada e logo saem. Mas na loja virtual, isso se torna não viável sem uma forma de coletar e processar essa informação, por isso que essa métrica busca propiciar ao gestor um entendimento de como é o tráfego no seu estabelecimento virtual;

6 EXPERIMENTO

Como experimento do protótipo desenvolvido, foram escolhidas duas lojas virtuais reais, para teste do sistema com a finalidade de coletar e analisar dados dessas lojas. Uma loja no ramo de produtos para decoração, casa e cozinha e outra loja no ramo de peças de reposição para linha de floresta e jardim como aparadores de grama, motosserras.

Para a coleta de dados, criou-se um servidor que recebeu a API de coleta de dados e o banco de dados não-relacional. Assim que a API foi colocada em funcionamento, foi inserido nas lojas que usam a plataforma *Magento Community*, o *script* de rastreamento com seus respectivos códigos identificadores, o "website_id".

Como as duas lojas virtuais estão *on-line* no mercado há pouco tempo, o fluxo de vendas é sutil, o que prejudicou a eficiência do estudo de algumas métricas, como relação de produtos comprados juntos. Por outro lado, foi uma boa oportunidade em analisar como era o comportamento dos atuais clientes das lojas em questão.

6.1 MÉTRICAS

Aqui mostra-se uma explanação de como foi o comportamento de cada grupo de métricas;

6.1.1 MÉTRICAS DE PÁGINA INICIAL

Essas métricas, como mostrado na seção 5.6, tem como objetivo dar ao gestor da loja uma ideia de como está o andamento de seu estabelecimento virtual e fornecer ao gestor informações úteis à administração da loja virtual; em poucas horas, dependendo da movimentação da loja, é possível ter noção de como está o comportamento do cliente ou se mesmo, está tendo acessos de usuários. Nesse sentido, trazem informações dos últimos dias no que concerne aos acessos dos usuários.

Com essas métricas, o gestor já consegue perceber, por exemplo, se está ocorrendo um

considerável movimento em sua loja, se os clientes estão adicionando produtos no carrinho ou até mesmo chegando ao *checkout*, com base nisso, perceber se ele está perdendo vendas ou se as finalizações estão aquém a movimentação na loja.

Dessa forma, essas métricas apresentam-se consistentes com o propósito esperado delas. A Figura 12 mostra a disposição das métricas vista pelo gestor.

Figura 12: Métricas de página inicial.
Fonte: autor

6.1.2 MÉTRICA CAMINHO COMUNS DE CLIENTES

Essa métrica não apresentou resultados satisfatórios. Para trabalhar com elas, cada acesso do cliente da loja virtual que era coletado, continha a url da página anteriormente visitada e da atual, com isso se esperava gerar um mapeamento baseado nas urls, que em suma indicaria a movimentação em comum dos clientes. Devido a grande quantidade de urls diferentes, o resultado foi um mapa onde a identificação da informação não é inteligível como nota-se na Figura 13.

A Figura 14 mostra outra apresentação do mesmo gráfico, onde é possível notar onde o tráfego massivo acontece, que nesse caso se concentra em sair da página inicial e ir para as páginas de produtos, que por serem muitas e cada página ter uma url diferente, desfigura o gráfico com a diversidade de urls mapeadas.

Assim, no sentido de melhorar essa métrica, seria possível fazer um agrupamento de todos os caminhos trafegados na loja e com isso, contabilizar os que são mais buscados ou aqueles que geralmente terminam com um fechamento de pedido. Com isso, apresentar os resultados mais relevantes.

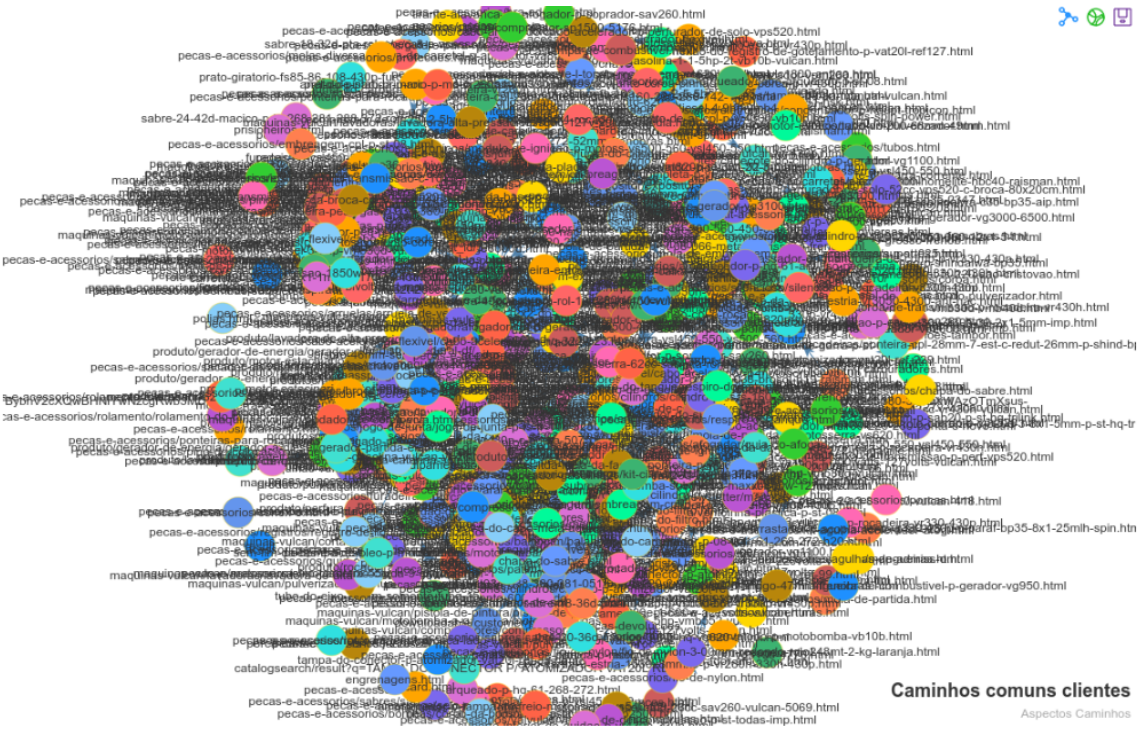

Figura 13: Apresentação do resultado com a métrica de caminhos comuns.
Fonte: autor

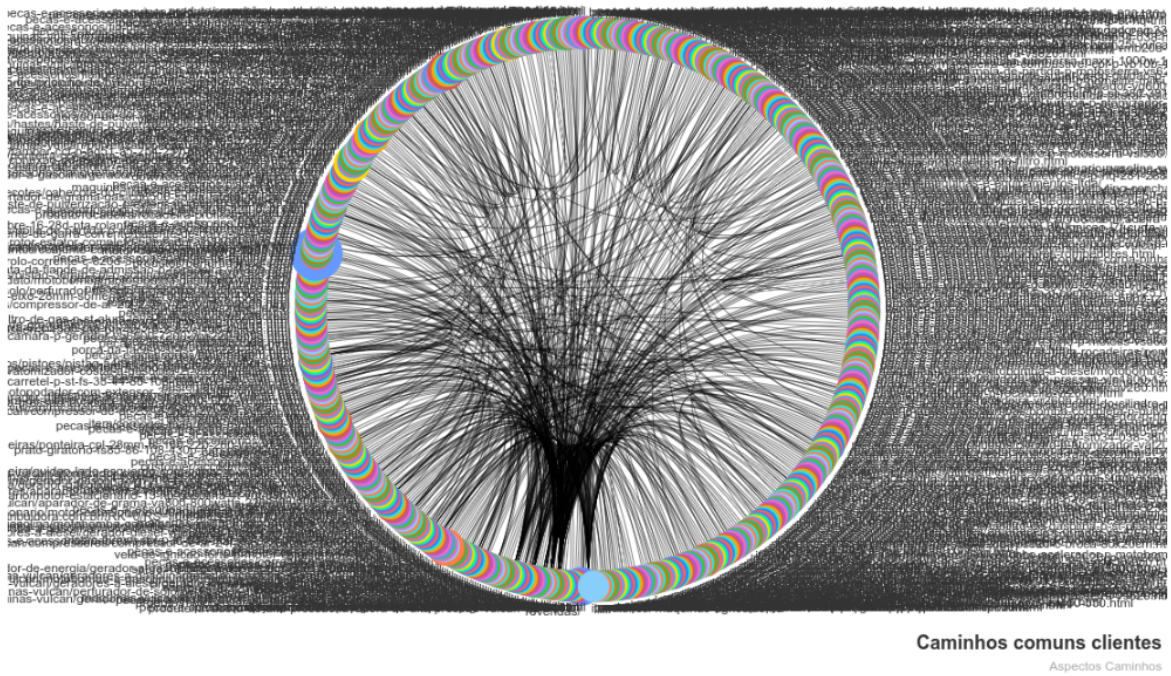

Figura 14: Outra forma de apresentação do resultado com a métrica de caminhos comuns.
Fonte: autor

6.1.3 MÉTRICA PERFIS DE CLIENTES

Essa métrica apresentou um comportamento consistente com o esperado. Conhecer perfis dos clientes é importante para se trabalhar concomitantemente com campanhas de *mar-*

keting, que visam atender um público mais específico, por exemplo, uma loja tem um produto em pré-lançamento, de valor superior, onde pode-se realizar uma campanha especial para oferecer aos clientes de maior poder aquisitivo esse lançamento.

6.1.4 MÉTRICA DE ABANDONO DE CARRINHO - CLIENTES

Essa métrica também apresentou um comportamento condizente com seu propósito. Relacionar os clientes que adicionaram produtos no carrinho e não apresentam registro de compra, mostrou que é um comportamento presente nas lojas testadas neste trabalho.

7 CONSIDERAÇÕES FINAIS

Aqui são elencados qual a contribuição que se almeja oferecer ao gestor de loja virtual.

Com esse sistema, objetivou oferecer uma ferramenta para ajudar o gestor a obter informações sobre sua loja virtual. Um *e-commerce* pode ter centenas ou milhares de acessos diários e esses acessos contém uma riqueza de informações, que podem ajudar o gestor a melhorar seu negócio. Mas é preciso um sistema computacional que possa processar com eficiência esse montante de dados.

Assim, o desafio principal foi pensar uma arquitetura onde fosse possível processar dados coletados de várias lojas virtuais. Com isso, a ideia foi dividir o sistema em partes distintas, uma parte para gerenciar os usuários e a interação deles com o sistema, outra parte para coletar e validar os dados de uma loja virtual, outra para salvar e processar esses dados para obter informações diversas.

Com essa arquitetura definida, o próximo passo foi pensar nas tecnologias necessárias para atender as necessidades do sistema. Onde armazenar esses dados? Como processá-los sem interferir no desempenho das lojas virtuais? Como mostrar esses dados ao gestor? Assim, a escolha de uma arquitetura com funções independentes e distribuídas que interagem entre si favorece o cumprimento dos objetivos, pois, para cada problema encontrado na arquitetura foi escolhida a tecnologia mais adequada, por exemplo, para uma API que poderá processar um grande número de requisições foi usado Node.js; para armazenar e trabalhar com um grande volume de dados foi usado MongoDB, que permite escalabilidade no armazenamento e processamento melhorando a performance. Foi uma boa experiência trabalhar com uma arquitetura não muito comum, com mais de um tipo de banco de dados e mais de um tipo de linguagem de programação.

Definidas as tecnologias, o passo seguinte foi pensar mais minuciosamente os detalhes e funcionalidades do sistema, diagramas de caso de uso, diagrama de sequência que propiciam uma melhor compreensão do funcionamento do sistema.

O primeiro passo prático para a criação do sistema foi desenvolver o *script* da loja

virtual, por não haver grande complexidade envolvida, foi relativamente fácil implementá-lo.

Após, foi criada a estrutura base da API de coleta de dados de forma que ela conseguisse fornecer arquivos estáticos. Com isso, se deu o desenvolvimento do *script* de coleta de dados, de maneira que ele estivesse funcional e cumprindo com suas atribuições. Até esse ponto como o desenvolvimento foi basicamente feito com Javascript, acabou sendo relativamente fácil fazer as atividades, as dúvidas que surgiam foram sanadas com o uso da documentação das tecnologias.

Com o *script* de coleta de dados funcionando, o próximo passo foi persistir os dados no banco, as configurações são relativamente fáceis de fazer, mas o uso do módulo "mongojs" demanda mais experiência, visto que ele trabalha semelhantemente ao funcionamento das consultas do MongoDB.

O desenvolvimento do módulo administrativo propiciou o conhecimento do *framework* Laravel, que realmente cumpre com a proposta de ser fácil e divertido de trabalhar.

Sobre as dificuldades encontradas, o MongoDB foi o maior gargalo no desenvolvimento do sistema, devido sua curva de aprendizado, mas por outro lado contribuiu para agregar conhecimento e experiência ao usar um banco não-relacional com uma proposta diferente de sistema. Outra dificuldade encontrada, foi devido as lojas virtuais estarem a pouco tempo no mercado, não ofereceram dados em grandes quantidades, mas isso por outra óptica foi bom, pois algumas deficiências puderam ser observadas, corroborando com a a importância na utilização de um sistema como o proposto.

Logo, é sensível a necessidade em se ter um sistema que ajude o gestor a captar pontos estratégicos da sua loja virtual. Realmente o sistema mostra que o acompanhamento do comércio eletrônico se faz necessário, mas para tanto há o que melhorar para se ter um sistema robusto, essas melhorias estão elencadas na Seção 7.1 referente aos trabalhos futuros.

7.1 TRABALHOS FUTUROS

Durante o desenvolvimento do projeto, constatou-se necessidades, que se implementadas em esforços futuros, vem contribuir com a melhor aplicabilidade e melhor atendimento da demanda dos clientes gestores. Essas necessidades estão listadas a seguir:

- Servir o *script* de coleta de dados na forma de uma CDN (Content Delivery Network). CDNs são servidores cujo o objetivo é fornecer arquivos para o usuário final, sem outros processamentos complexos. O *script* de coleta é um arquivo estático, solicitado a cada

requisição feita pelo computador cliente da loja virtual, com o aumento da demanda de requisições fica mais fácil atendê-la usando um serviço de CDN.

- Também é importante melhorar a segurança no tratamento dos dados com métodos criptográficos em todos os tramites do sistema. Criar uma forma de realizar *backups* dos dados coletados dos clientes, visto que esses dados são o cerne do sistema;

- Dividir a API de coleta de dados de forma que sejam dois serviços distintos, um somente para receber os dados coletados das lojas virtuais e outra para atender as requisições oriundas do painel administrativo, solicitadas pelos gestores das lojas virtuais. Com isso, espera-se melhorar a gestão da API no que diz respeito a manutenção e escalabilidade dos serviços, tratando-os de forma independente;

- Estudar novas métricas que podem ser aplicadas ao sistema, que venham contribuir com o conjunto de informações gerados. Na medida que os clientes vão se fidelizando e gerando meses e anos de dados com suas lojas, se torna importante explorar esses dados, para que venham a contribuir com alguma informação importante;

- Ainda no sentido de melhorias de métricas, criar uma funcionalidade que permite ao gestor verificar o histórico de qualquer métrica, em um intervalo de data escolhida por ele, para que assim seja possível estudar melhor o andamento da sua loja virtual ao longo do tempo. Para isso, criar filtros onde ele possa selecionar o intervalo de datas;

- Na medida que os dados coletados e seu processamento demandem mais recursos computacionais, trabalhar na escalabilidade do sistema, para que seja possível atender satisfatoriamente a gama de lojas virtuais usuárias do serviço;

- Desenvolver a funcionalidade que ajude o gestor no gerenciamento do estoque de sua loja. Coletando informações do estoque dos produtos, para propiciar informações sobre o desempenho de estoque de cada produto; Indicando produtos que estão parados a mais tempo, ou que não tem muita saída. Isso ajuda na manutenibilidade do estoque;

- Desenvolver um serviço que identifique padrões de produtos comprados juntos, onde o gestor possa configurar sua loja virtual com base nessas informações, para oferecer ao cliente produtos para venda cruzada com maior relevância;

- Elaborar a documentação do projeto, de forma que novos desenvolvedores tenham condições de compreender a arquitetura do sistema;

- Fazer um estudo para melhorar a usabilidade do sistema, criando uma interface mais intuitiva e agradável de usar;

REFERÊNCIAS

ALMEIDA, M. B. Uma introdução ao xml, sua utilização na internet e alguns conceitos complementares. **Ciência da Informação**, SciELO Brasil, v. 31, n. 2, p. 5–13, 2002.

ANALYTICS. **Google Analytics - Google Inc.** set 2015. Disponível em: http://www.qlik.com. Acesso em: 08/09/2015.

ANTONIO, C. S. **Aprendendo Ruby On Rails**. 2015. Disponível em: http://goo.gl/FHksQo. Acesso em: 08/11/2015.

ATRE, S.; MOSS, L. T. **Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications.** 1a ed. ed. Boston, USA: Addison-Wesley, 2003.

BARBIERI, C.; CARDOSO Érico de S. **Business Intelligence: Modismo ou Necessidade?** 2015. Disponível em: http://ericocardoso.com/publicacoes/BI.pdf. Acesso em: 24/11/2015.

BELEZA, C.; MEDEIROS, S. **O QUE É BUYER PERSONA? POR QUE CRIAR PERSONAS PARA A SUA EMPRESA?** 2015. Disponível em: http://5seleto.com.br/o-que-e-buyer-persona-por-que-criar-personas-para-empresa/. Acesso em: 22/11/2015.

BENTO, E. J. **Desenvolvimento web com PHP e MySQL**. São Paulo, SP: Casa do Código, 2013.

BEZERRA, P. T.; SCHIMIGUEL, J. **DESENVOLVIMENTO DE APLICAÇÕES MOBILE CROSS-PLATFORM UTILIZANDO PHONEGAP**. 2016.

BONFIM, F. L.; LIANG, M. Aplicações escaláveis com mean stack. **Monografia (Graduação**.

BRITO, K. **Fundamentos do Desenvolvimento Web**. 1. ed. 2011.

BUENO, A. J. F. Utilização do business intelligence no comércio eletrônico. 2015.

CAELUM. **Desenvolvimento Web com HTML, CSS e JavaScript**. 2015. Disponível em: https://www.caelum.com.br/apostila-html-css-javascript/bootstrap-e-formularios-html5/. Acesso em: 11/11/2015.

CECI, F. **Business Intelligence**. 2a edição. ed. Palhoça, SC: Ed. UnisulVirtual, 2012. ISBN 978-85-7817-465-1.

CONTENT, R. **Business Intelligence pode significar o sucesso do seu e-commerce**. set 2015. Disponível em: http://www.profissionaldeecommerce.com.br/business-intelligence-no-e-commerce/. Acesso em: 22/11/2015.

CONTENT, R. **Google Analytics: guia completo**. Belo Horizonte, MG: Rock Content Ebooks, 2015.

E-COMMERCE, G. de. **Pesquisa sobre plataformas de e-commerce**. 2014. Disponível em: http://www.guiadeecommerce.com.br/pesquisa-sobre-plataformas-de-ecommerce/. Acesso em: 22/11/2015.

ELETRÔNICO, C. D. G. **Apostila de MySQL**. 2016. Disponível em: https://goo.gl/oCBcal. Acesso em: 30/10/2016.

FLANAGAN, D. **JavaScript: O guia definitivo**. 6. ed. Porto Alegre, RS: Bookman Companhia Editora Ltda, 2013. ISBN 978-85-65837-19-4.

GITHUB nodejs on. **Projects, Applications, and Companies Using Node**. Nov 2015. Disponível em: https://goo.gl/K4gDy0 . Acesso em: 11/11/2015.

IAMASHITA, C. H.; MAGALHÃES, W. B. Html 5, um estudo sobre seus novos recursos. 2013.

INFNET, I. **O comércio eletrônico não sabe o que é crise**. 2015. Disponível em: http://www.infnet.edu.br/negocios/o-comercio-eletronico-nao-sabe-o-que-e-crise. Acesso em: 22/11/2015.

MADEIRA, M. N. **Comércio Eletrônico: Livro didático**. 4a edição. ed. Palhoça, Santa Catarina: UnisulVirtual, 2007. ISBN 978-85-60694-78-5.

MALAGOLI, F. et al. Testes de performance utilizando o db4o e mongodb. **e-RAC**, v. 3, n. 1, 2013.

MARKETING, A. do. **Configuração do Google Analytics**. 2015. Disponível em: http://www.academiadomarketing.com.br/configuracao-do-google-analytics/. Acesso em: 22/11/2015.

MEDEIROS, S.; BELEZA, C. **TUDO SOBRE GOOGLE ANALYTICS, A MELHOR FERRAMENTA DE ESTATÍSTICAS PARA SEU SITE**. jun 2015. Disponível em: http://5seleto.com.br/tudo-sobre-google-analytics-a-melhor-ferramenta-de-estatisticas-do-site/. Acesso em: 22/11/2015.

MOZILLA, F. **XMLHttpRequest**. 2015. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/API/XMLHttpRequest. Acesso em: 22/11/2015.

NASCIMENTO, A. R. d.; SILVA, B. F. d.; SANTOS, G. G. d. E-commerce: o melhor caminho no mercado atual. 2009.

OLIVEIRA, D. T.; PEREIRA, O. J. Um estudo do business intelligence no ambiente empresarial. 2008.

OLIVEIRA, J. D. M.; LEANDRO, J. L. D. C. Desenvolvimento de sistema eletrônico de protocolo com framework prado. Universidade Católica de Goiás, 2007.

OLIVEIRA, R. **Qlik Sense – O que é?** ago 2015. Disponível em: http://robertooliveira.com.br/qlik-sense-o-que-e/. Acesso em: 22/11/2015.

PEREIRA, C. R. **Node.js: Aplicações web real-time com Node.js**. São Paulo, SP: Casa do Código, 2014.

PESSOA, B. C. et al. Banco de dados mongodb vs banco de dados sql server 2008. **RE3C-Revista Eletrônica Científica de Ciência da Computação**, v. 7, n. 1, 2012.

PUBLISHING, O. **OECD Guide to Measuring the Information Society 2011**. Paris, France: Organisation for Economic Co-operation and Development, 2011. ISBN 978-92-64-09598-4.

ROSA, A. G. Análise da estrutura do banco de dados mongodb: Testes de desempenho mongodb x mysql. Clube de Autores, 2009.

SCHROEDER, R.; SANTOS, F. dos. **Arquitetura e testes de serviços web de alto desempenho com node. js e mongodb**. 2014.

SENSE. **Qlik Sense**. set 2015. Disponível em: http://www.google.com/analytics/. Acesso em: 08/09/2015.

SILVA, M. S. Jquery. **Biblioteca do Programador JavaScript. São Paulo–SP. Editora Novatec**, 2008.

SILVA, M. S. **HTML 5 A Linguagem de Marcação que revolucionou a web**. 1. ed. São Paulo, SP: Ed. Novatec, 2010. ISBN 978-85-7522-261-4.

SILVA, M. S. **JavaScript: Guia do programador**. 1. ed. São Paulo, SP, 2010.

SILVA, M. S. **CSS3: desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3**. São Paulo, SP: Novatec Editora, 2011. ISBN 978-85-7522-289-8.

SILVA, M. S. **Bootstrap 3.3.5 Aprenda a usar o framework Bootstrap para criar layouts CSS complexos e responsivos**. 1. ed. São Paulo, SP: Novatec Editora Ltda, 2015. ISBN 978-85-7522-460-1.

SMITH, R.; SPEAKER, M.; THOMPSON, M. **O Mais Completo Guia Sobre E-commerce**. 1. ed. 2000.

SOARES, W. **AJAX (Asynchronous JavaScript And XML): guia prático**. São Paulo, SP: Ed. Érica, 2006. ISBN 9788536501109.

S.V., F.; PEREIRA, L. B. R. **SQL dos Conceitos às Consultas Complexas**. Rio de Janeiro, RJ: Ed. Ciência Moderna, 2009. ISBN 9788573938593.

TURBAN, E. et al. **Business Intelligence: um enfoque gerencial para a inteligência do negócio**. Porto Alegre, RS: Ed. Bookman, 2009. ISBN 978-0-13-234761-7.

W3C. **Consortium Member List**. Nov 2015. Disponível em: http://www.w3.org/Consortium/Member/List. Acesso em: 11/11/2015.

W3C. **W3C (2015). CSS Curso W3C Escritório Brasil**. 2015. Disponível em: Disponível em: http://goo.gl/z16rTe. Acesso em: 07/11/2015.

APÊNDICE A – OBJETOS JAVASCRIPTS SCRIPT DE RASTREAMENTO

Aqui elenca-se exemplos de objetos Javascript gerado pelo script de rastreamento:

Figura 15: Objeto Javascript gerado na página inicial.
Fonte: autor

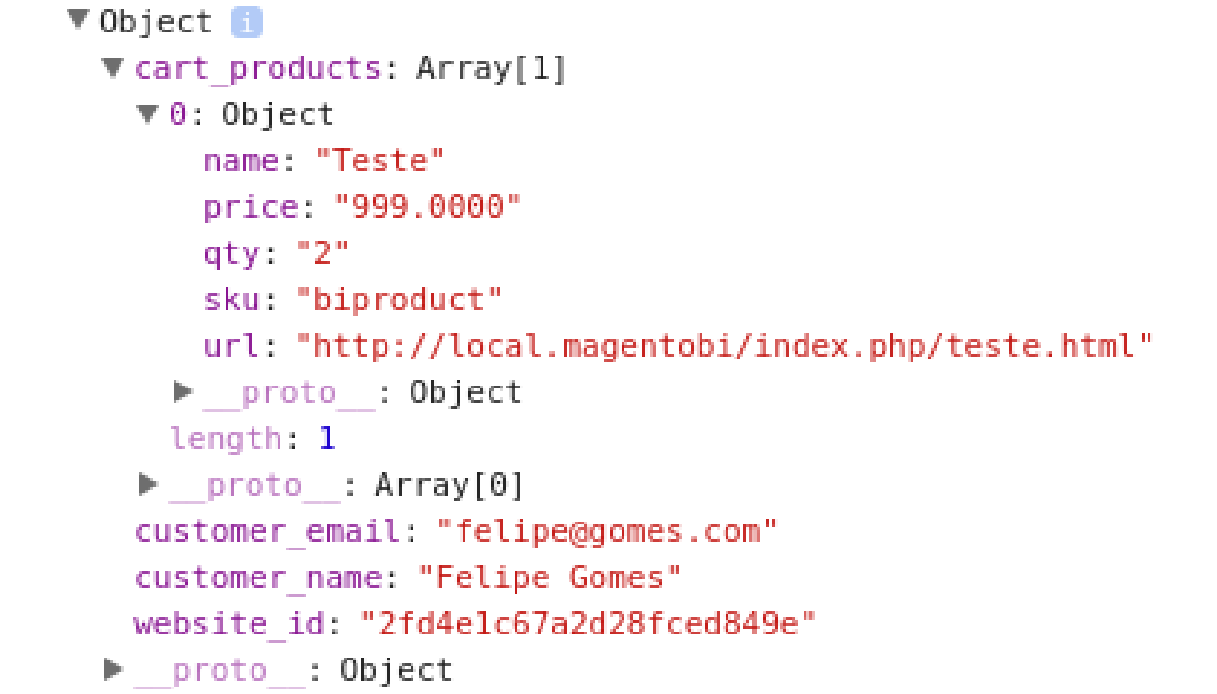

Figura 16: Objeto Javascript gerado na página do carrinho.
Fonte: autor

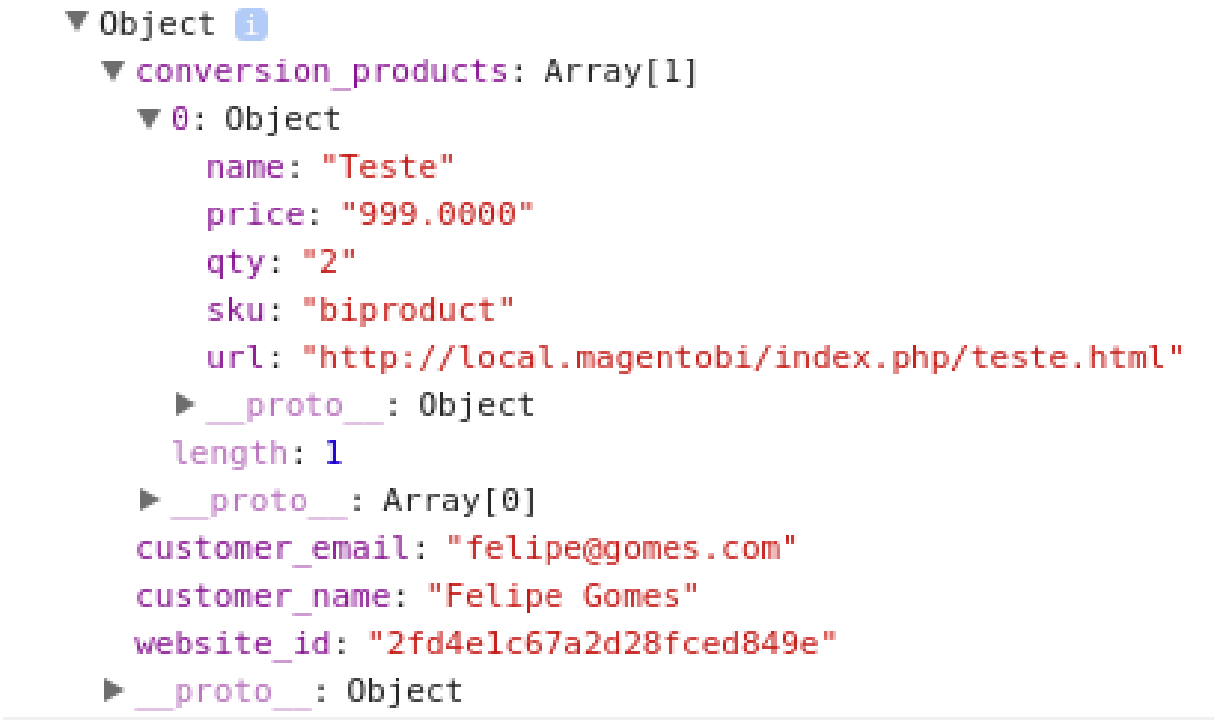

Figura 17: Objeto Javascript gerado na página de checkout.
Fonte: autor

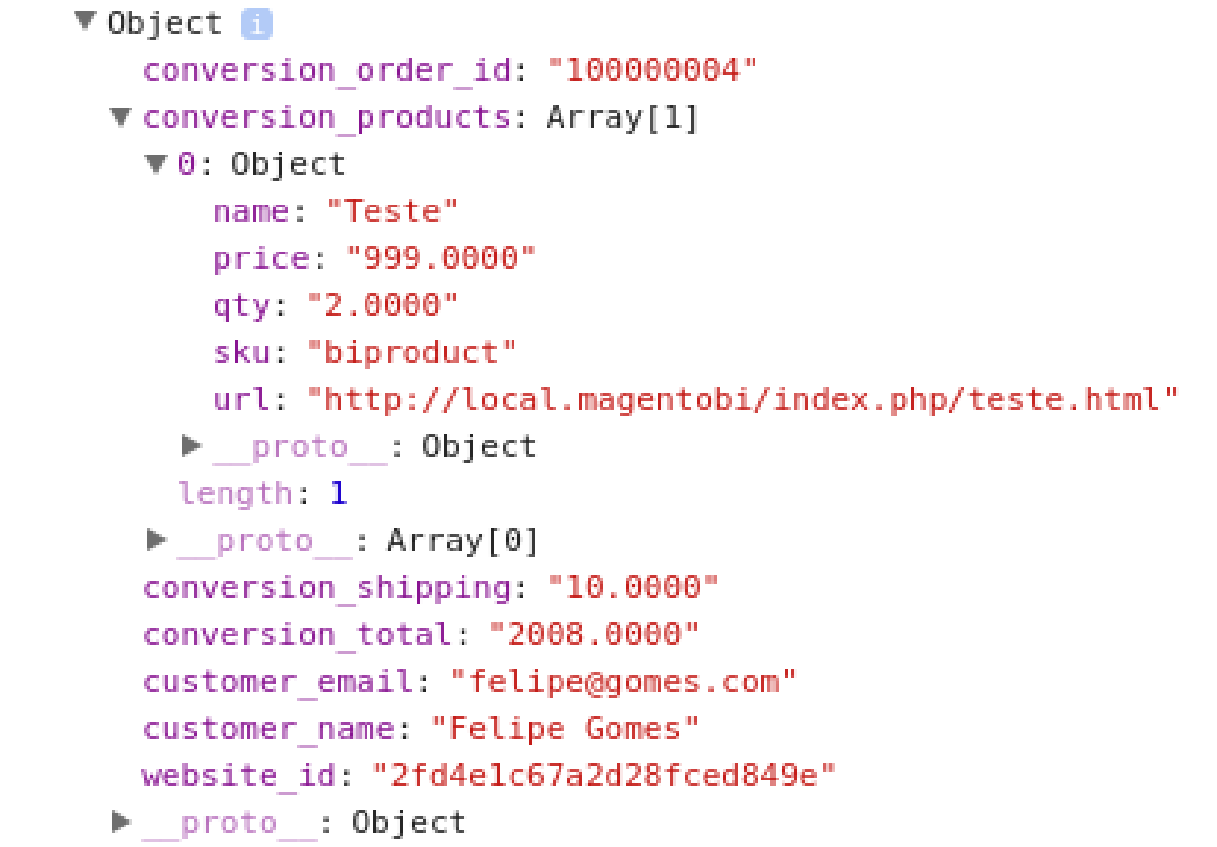

Figura 18: Objeto Javascript gerado na página de sucesso de compra.
Fonte: autor

APÊNDICE B – CÓDIGO DA LOJA VIRTUAL

```
<!-- BI SCRIPT START -->
<script type="text/javascript">
    BI = {
            "customer": "<?php echo Mage::helper('customer')->getCustomer()->getEmail(); ?>",
        <?php $actionName = Mage::app()->getFrontController()->getAction()->getFullActionName() ?>

        <?php if($actionName == "checkout_cart_index") : ?>
            <?php $BiItemsObjName = "cart_products" ?>
            <?php $itemsObj = Mage::getSingleton('checkout/session')->getQuote() ?>
        <?php endif; ?>

        <?php if($actionName == "onestepcheckout_index_index") : ?>
            <?php $BiItemsObjName = "checkout_products" ?>
            <?php $itemsObj = Mage::getSingleton('checkout/session')->getQuote() ?>
        <?php endif; ?>

        <?php if($actionName == "checkout_onepage_success") : ?>
            <?php $BiItemsObjName = "conversion_products" ?>
            <?php $orderIncrementId = Mage::getSingleton('checkout/session')->getLastRealOrderId()
                ?>
            <?php $order = Mage::getModel('sales/order')->loadByIncrementId($orderIncrementId) ?>
            <?php $itemsObj = $order ?>

            <?php if(isset($order)) : ?>
                "conversion_order_id" : "<?php echo $order->getIncrementId() ?>",
                "conversion_shipping" : "<?php echo $order->getShippingAmount() ?>",
                "conversion_total"    : "<?php echo $order->getGrandTotal() ?>",
            <?php endif; ?>
        <?php endif; ?>

        <?php if(isset($itemsObj)) : ?>
            <?php $items = $itemsObj->getAllVisibleItems() ?>
            "<?= $BiItemsObjName ?>":[
                <?php foreach($items as $item) : ?>
                {
                    "sku" : "<?php echo $item->getProduct()->getSku() ?>",
                    "qty" : "<?php echo ($item->getQtyOrdered() == "" ? $item->getQty() : $item->
                        getQtyOrdered()) ?>",
                    "name" : "<?php echo $item->getProduct()->getName() ?>",
                    "price" : "<?php echo $item->getProduct()->getPrice() ?>",
                    "url" : "<?php echo $item->getProduct()->getProductUrl() ?>"
                },
            <?php endforeach; ?>
        ],
        <?php endif; ?>
    },

    (function(){
        BI.website_id = 'f74a24d09157bb8e7dd1435';
        var _dcs = document.createElement('script');
        _dcs.type = 'text/javascript';
        _dcs.async = true;
        _dcs.src ='http://45.55.244.135:3000/collector/tracking.js';
        document.getElementsByTagName('head')[0].appendChild(_dcs);
    })();
</script>
<!-- BI SCRIPT END -->
```

Figura 19: Código preparado para a lojas cujas as plataformas usam o Magento Community versão 1.9.x

Fonte: autor