

**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**

PROJETO DO TRABALHO DE CONCLUSÃO DE CURSO

GUARAPUAVA

2015

FELIPE GOMES

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

Projeto do Trabalho de Conclusão de Curso apresentado à disciplina Trabalho de Conclusão de Curso I 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

2015

RESUMO

GOMES, Felipe. SISTEMA DE APOIO À GESTÃO DE COMÉRCIO ELETRÔNICO POR INTELIGÊNCIA DE NEGÓCIOS. 43 f. Projeto do Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2015.

Os sistemas de comércio eletrônico comumente não oferecem relatórios mais elaborados que visam compreender o comportamento de seus clientes durante o processo de compra ou durante períodos específicos de forma a verificar e extrair informações que possam ajudar o gestor na melhoria do seu negócio. Visto que plataformas de comércio eletrônico normalmente não tem como foco fornecer dados de inteligência de negócios, o objetivo deste trabalho é 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 e interesses dos clientes, como por exemplo, os caminhos comuns de clientes no site. 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. 43 f. Projeto do Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2015.

E-commerce systems often do not provide more elaborate reports aimed at understanding the behavior of their customers during the buying process or during specific periods in order to verify and extract information that can help the manager in improving their business. Since e-commerce platforms typically has focused on providing business intelligence data, the aim of this study is to develop a system for collecting data from electronic trading platform for the buying process, and thereby extract values that inform behavior points and customer interests, such as common ways for customers on the site. Finally, generate reports intelligibly express this information, so that they serve to support the manager's decision-making in order to improve the purchase of your e-commerce experience.

Keywords: Business Intelligence, E-Commerce, Web System

LISTA DE FIGURAS

Figura 1	– Diagrama de caso de uso do painel administrativo.	29
Figura 2	– Dinâmica de funcionamento do sistema proposto.	30
Figura 3	– Exemplo do código de rastreamento.	31
Figura 4	– Exemplo de documento JSON para ser persistido no MongoDB.	33
Figura 5	– Diagrama de classe para o módulo de coleta de dados.	34
Figura 6	– Banco de dados painel administrativo.	35
Figura 7	– Diagrama de sequência cadastro do gestor.	36
Figura 8	– Diagrama de sequência da coleta de dados.	37

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
DRY	Don't Repeat Yourself
NoSQL	Not Only SQL
BSON	Binary Object Notation
JSON	JavaScript Object Notation
SQL	Structured Query Language
SSL	Secure Sockets Layer
BI	Business Inteligence
OECD	Organisation for Economic Cooperation and Development
SEO	Search Engine Optimization

SUMÁRIO

1	INTRODUÇÃO	7
1.1	OBJETIVOS	8
1.1.1	Objetivo Geral	8
1.1.2	Objetivos Específicos	8
1.2	DIFERENCIAL TECNOLÓGICO	9
2	RESENHA LITERÁRIA	10
2.1	ESTADO DA ARTE	10
2.2	FUNDAMENTAÇÃO TEÓRICA	12
2.2.1	Html	12
2.2.2	CSS	13
2.2.3	Bootstrap	14
2.2.4	Javascript	14
2.2.5	jQuery	16
2.2.6	Node.js	17
2.2.7	Ruby com framework Rails	18
2.2.8	MongoDB	19
2.2.9	PostgreSQL	20
2.2.10	Inteligência de Negócios	20
2.2.11	Comércio Eletrônico	22
2.2.12	Inteligência para lojas virtuais	23
3	METODOLOGIA	24
3.1	LEVANTAMENTO DOS REQUISITOS DO SISTEMA	24
3.2	IDEALIZAÇÃO DA ARQUITETURA DO SISTEMA	24
3.3	ESTUDO DAS TECNOLOGIAS À SEREM UTILIZADAS	24
3.4	TESTAR A APLICAÇÃO EM AMBIENTE DE PRODUÇÃO REAL	25
3.5	ANÁLISE DOS RESULTADOS OBTIDOS	25
3.6	CRONOGRAMA	25
4	DESENVOLVIMENTO PRELIMINAR	27
4.1	REQUISITOS FUNCIONAIS DO SISTEMA	27
4.1.1	Atores e seus papéis na aplicação	28
4.2	DINÂMICA DE FUNCIONAMENTO	29
4.3	BANCO DE DADOS - PAINEL ADMINISTRATIVO	34
4.4	DIAGRAMA DE SEQUÊNCIA DO CADASTRO DO GESTOR	36
4.5	DIAGRAMA DE SEQUÊNCIA DA COLETA DE DADOS	37
5	CONSIDERAÇÕES FINAIS	39
	REFERÊNCIAS	41

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*, são 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 Community¹, uma das plataformas de lojas virtuais 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 do volume de acessos da loja virtual, e as possibilidades de cruzamentos entre eles de modo a obter uma informação

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

de interesse, como por exemplo, avaliar os perfis de clientes, quais clientes são mais assíduos e compram mais.

O presente trabalho visa desenvolver um sistema de apoio à gestão estratégica de comércio eletrônico, na forma de um módulo 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 3.6, apresenta-se o cronograma de execução das atividades. No Capítulo 4, há a descrição do que já foi realizado no projeto. No Capítulo 5, descreve-se as considerações sobre o projeto e das atividades desenvolvidas nele.

1.1 OBJETIVOS

1.1.1 OBJETIVO GERAL

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 a informação, e entender aspectos de seus negócios para tomada de decisões;

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 produtos com venda em potencial;

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 já desenvolvidos na área do projeto. Assim, como as tecnologias que serão utilizadas futuramente no desenvolvimento do projeto.

2.1 ESTADO DA ARTE

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.

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).

Um terceiro sistema, é o serviço da Wunderdata² de inteligência de negócios para lojas on-line. Suas métricas disponíveis informam dados sobre:

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

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

¹Qlik 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.

²Wunderdata disponível em : <http://wunderdata.com>

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 serão apresentadas as tecnologias e conceitos utilizados no desenvolvimento do sistema que constitui o projeto. Tem-se como intenção realizar um estudo sobre as linguagens de programação, banco de dados, conceitos teóricos de inteligência de negócios e loja virtual. Assim poderá se ter uma visão mais concreta do projeto e de como ele será colocado em prática.

2.2.1 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 outros 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 Corporation⁶, utilizam HTML em seus produtos, pois também participam como membros da *World Wide Web Consortium* (W3C)⁷, organização responsável pela padronização da web (W3C, 2015).

HTML tem por finalidade nesse projeto, 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.

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,

³Tag 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>

⁷W3C: <http://www.w3.org>

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 Facebook⁸, Opera Software⁹, PayPal¹⁰ (W3C, 2015).

Nesse projeto, o uso de folhas de estilo, é 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* responsivo¹¹. Facilita o desenvolvimento *front-end* tornando-o mais rápido e fácil. O Bootstrap 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 das tags, conjuntos de ícones, *grids*¹², plugins JavaScript, tudo é responsivo (CAELUM, 2015).

Esse *framework front-end*, vai ser 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

⁸<https://www.facebook.com>

⁹<http://www.opera.com>

¹⁰<https://www.paypal.com/br>

¹¹Responsivo: um layout que se adapta aos diversos tamanhos de telas.

¹²Técnica que permite criar layout responsivos

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 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).

Nesse projeto, JavaScript é amplamente usado. As informações que serão coletadas no computador do cliente(consumidor) estarão formatadas como um objeto JavaScript. No computador do cliente existirá um script usando essa tecnologia que agirá como um gatilho que

solicitará outro script em JavaScript, que vai coletar os dados do computador cliente e enviar 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 vai ter acesso, será também usado 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;

Nesse projeto jQuery será 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 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 normalmente mudam bem de escala e 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 web 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 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 mantêm-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, vai usar Node.js no desenvolvimento do módulo de coleta de dados, que vai servir 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 caracte-

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

¹⁴Sítio usando Node: <http://online.wsj.com/>

¹⁵<http://www.godaddy.com>

¹⁶<http://linkedin.com/>

terísticas citadas acima, os benefícios alcançados com o Node vem propiciar uma melhor gestão de uma grande quantidade de requisições de forma mais performática.

2.2.7 RUBY COM FRAMEWORK RAILS

Ruby é uma linguagem de programação interpretada, multi paradigma, que possui tanto programação orientada a objetos como programação funcional. Possui tipagem dinâmica, logo não é preciso declarar o tipo de variável, mas essa tipagem é forte o que indica que a linguagem não realiza *cast* automáticos para outros tipos de dados.

Entre seus recursos de destaques Ruby apresenta uma sintaxe simples, orientação a objetos completa e aberta, herança única, implementa *closures*, o coletor de lixo é totalmente funcional e tem um sistema de *threading* independente do sistema operacional, a linguagem também possui um gerenciador de pacotes chamado *Ruby Gems*, com o qual é possível gerenciar várias bibliotecas de códigos reutilizáveis (BRODBECK, 2007).

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

Projetado para ser uma solução completa de desenvolvimento, escrito em somente uma linguagem, e ter suas camadas se comunicando de forma transparente e uniforme, segue a arquitetura MVC¹⁷. Isso provê ao Ruby com Rails, produtividade no desenvolvimento com uma baixa curva de aprendizagem (CRUZ et al., 2011).

O *framework* absorve algumas características como o *Don't Repeat Yourself (DRY)*, conceito para reaproveitar código em outros lugares quando necessário. Convenção sobre configuração, assumir valores padrão baseados em convenção; essa padronização ajuda entender o código e aumenta a produtividade. Mapeamento objeto-relacional através do *Active Record*, módulo que trata do acesso ao banco de dados. Também propicia facilidade para as operações básicas como criar, ler, alterar e apagar em uma tabela do banco de dados, oferecendo ferramentas para gerar parte do código necessário para manipular modelo de dados, procedimento conhecido como *Scaffolding* (CRUZ et al., 2011).

Entre as empresas que adotam o Ruby é possível citar o Github ¹⁸ repositório de

¹⁷MVC é um padrão de arquitetura de software, separando sua aplicação em 3 camadas. A camada de interação do usuário(view), a camada de manipulação dos dados(model) e a camada de controle(controller). Disponível em: <http://goo.gl/CJ74hU>

¹⁸<http://github.com/>

códigos com elementos de rede sociais, Shopify¹⁹ uma popular plataforma de e-commerce, Scribd²⁰ uma grande empresa de publicação social.

Nesse projeto, o uso de Ruby com o framework Rails, será necessário para criar o sistema que vai ser usado para o painel administrativo, onde os clientes(gestores), e o administrador do sistema, terão 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; SANTOS1, 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 em *clusters*²¹ 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 mongodb²², tem-se alguns casos de uso por empresas entre elas: Forbes²³, Hudl²⁴, Pearson²⁵, entre outros.

Esse projeto tem como objetivo armazenar dados de vários clientes oriundos de várias

¹⁹<http://www.shopify.com/>

²⁰<http://www.scribd.com/>

²¹Cluster: 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>

²²<https://www.mongodb.org/community/deployments?group=use-cases>

²³<http://www.forbes.com>

²⁴<http://www.hudl.com>

²⁵<https://www.pearson.com>

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 POSTGRESQL

PostgreSQL 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.

PostgreSQL, é um sistema de gerenciamento de banco de dados de alta performance, mas é bem amigável na utilização no dia-a-dia e de fácil administração. Ele também é um sistema *multithread*, utiliza o padrão SQL, oferece altos níveis de acesso com rigorosos controle. Tem ainda regras de acesso aos dados, por exemplo, um conjunto de dados pode ser acessado por dois ou mais usuários, mas modificado somente por um. Na questão de segurança, ele utiliza ligações encriptadas pelo protocolo SSL (CALDEIRA, 2015).

Nesse projeto, o PostgreSQL vai servir basicamente para persistir os dados sobre os usuários do painel administrativo, dados das empresas, e suas respectivas lojas virtuais. E também dados de configuração, identificação e outros dados relacionados ao usuários gestores ficará sobre responsabilidade desse sistema de gerenciamento de banco de dados.

2.2.10 INTELIGÊNCIA DE NEGÓCIOS

Inteligência de Negócios (do inglês Business Intelligence - 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*²⁶ vencida;
- Ganho de *Market-Share*²⁷ 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.

²⁶Técnica de marketing para apresentação de um produto no mercado.

²⁷Grau de participação de uma empresa no mercado em termos das vendas de um determinado produto.

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.

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)²⁸ 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.

²⁸SEO: Otimização para mecanismos de buscas

3 METODOLOGIA

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

3.1 LEVANTAMENTO DOS REQUISITOS DO SISTEMA

Serão relacionados os requisitos do sistema e descritos na forma de histórias. Essas histórias tem como objetivo demonstrar 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.

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 será 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.

3.3 ESTUDO DAS TECNOLOGIAS À SEREM UTILIZADAS

Nessa etapa serão estudadas as tecnologias a serem empregadas. Busca-se entender de que forma cada tecnologia pode ajudar efetivamente para um resultado satisfatório, com o auxílio de literaturas especializadas pretende-se atinar de forma essencial como empregar cada tecnologia.

3.4 TESTAR A APLICAÇÃO EM AMBIENTE DE PRODUÇÃO REAL

Após desenvolver o sistema de coleta de dados, pretende-se testá-lo em um sistema de comércio eletrônico, sendo sua posterior análise apresentada no sistema em um painel administrativo.

Entrar em contato com empresários para verificar a possibilidade de usar a loja virtual, para coleta de dados por alguns meses, de maneira a ter uma base de dados real onde seja possível aplicar e submeter a apreciação do gestor as métricas de desempenho apuradas, mediante relatórios no painel do cliente.

3.5 ANÁLISE DOS RESULTADOS OBTIDOS

Após coletados os dados em um determinado período será verificado se o sistema cumpriu com sua proposta. Assim, será verificado se foi possível 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.

3.6 CRONOGRAMA

Na Tabela 2 apresenta-se o cronograma com a conjectura para o desenvolvimento do referido sistema. As partes destacadas referem-se as atividades que em suma foram concluídas ou ainda apresentam-se em prosseguimento.

Atividades	TCC 1					TCC 2				
	Ago	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai
1. Levantamento dos requisitos do sistema	■	■								
2. Idealização da arquitetura do sistema		■	■							
3. Estudo das tecnologias à serem utilizadas		■	■	■	■					
4. Redação do projeto de TCC 1		■	■	■						
5. Defesa do projeto					■					
6. Desenvolvimento do módulo de coleta de dados e armazenamento.				■	■	■	■	■		
7. Desenvolvimento do módulo de redução de dados.					■	■	■	■	■	
8. Desenvolvimento do módulo de apresentação					■	■	■	■	■	
9. Verificar a possibilidade de testar a aplicação em ambiente de produção real					■	■	■			
10. Análise dos resultados obtidos									■	
11. Elaboração da apresentação final.								■	■	
12. Defesa final do TCC										■

Tabela 2: Cronograma de atividades.

Fonte: autor

4 DESENVOLVIMENTO PRELIMINAR

Neste capítulo serão apresentados os primeiros resultados alcançados até o momento 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. Para o sistema de coleta de dados foi desenvolvido o diagrama de sequência e o código de rastreamento.

4.1 REQUISITOS FUNCIONAIS DO SISTEMA

Na Tabela 3 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 3: 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

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.

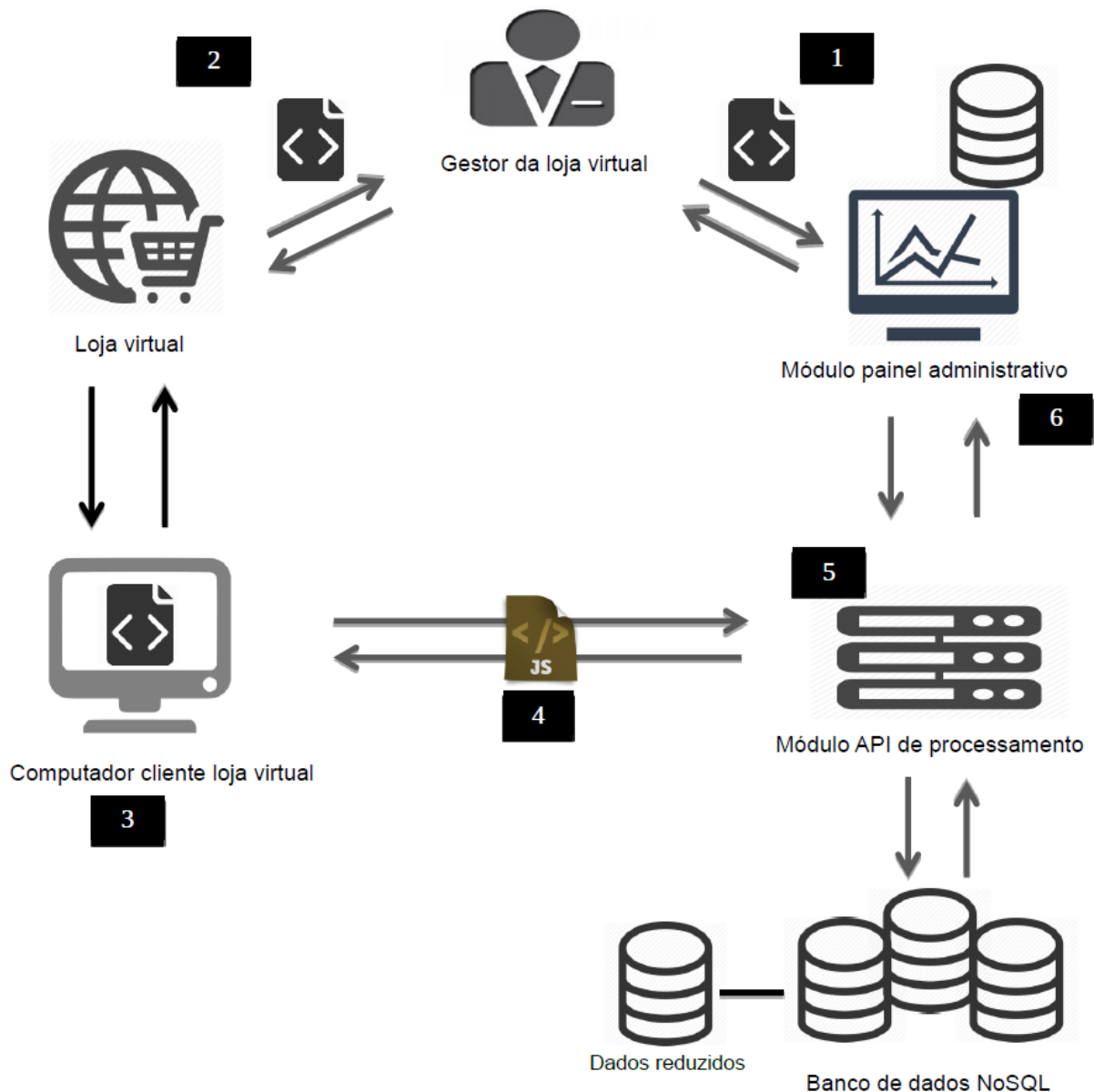


Figura 2: Dinâmica de funcionamento do sistema proposto.

Fonte: autor

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.

```

1  <script type="text/javascript">
2      var BI = {
3          "customer_name" : "###",
4          "customer_email" : "###",
5          "cart_products":[
6              {
7                  "sku" : "xxx",
8                  "qty" : "xxxx",
9                  "name" : "xxx",
10                 "price" : "xxx.xx",
11                 "url" : "xxxx",
12                 "image" : "xxxx"
13             }
14         ],
15         "checkout_products":[
16             {
17                 "sku" : "xxxx",
18                 "qty" : "x",
19                 "name" : "xxxxxxxx",
20                 "price" : "xxx.xxx",
21                 "url" : "xxxxx",
22                 "image" : "xxxxxxxx"
23             }
24         ],
25         "conversion_items":[
26             {
27                 "sku" : "xxxxx",
28                 "qty" : "xxxx",
29                 "name" : "xxxxx",
30                 "price" : "xxx.xx",
31                 "url" : "xxxx",
32                 "image" : "xxxx"
33             }
34         ],
35         "conversion_order_id" : "xxxxxxx",
36         "conversion_shipping" : "xxxx.xx",
37         "conversion_total": "xxxxxx"
38     }
39     (function(){
40         BI.webstore_id = 'xxxxxxxxxxxxxxxxxxxx';
41         var _dcs = document.createElement('script');
42         _dcs.type = 'text/javascript';
43         _dcs.async = true;
44         _dcs.src = '10.0.0.1/bi_tracking.min.js';
45         document.getElementsByTagName('head')[0].appendChild(_dcs);
46     })();
47 </script>

```

Figura 3: Exemplo do código de rastreamento.

Fonte: autor

Esse script possui três chaves que são semelhantes no seu conteúdo, são eles: *cart_products*, *checkout_products* e *conversion_items*, nas linhas 5, 15 e 25 respectivamente. Ambos mantêm

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 daquele acesso, por exemplo, se o cliente está logado vai conter seu e-mail, seu nome; se o cliente está finalizando uma compra, vai ter 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 para plataformas de loja virtuais mais usadas, como por exemplo o Magento Community; caso o gestor não encontrar o código pronto para a plataforma que ele usa, é disponibilizado o código com valores em branco, ficando sob a responsabilidade dele inserir esses valores.

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

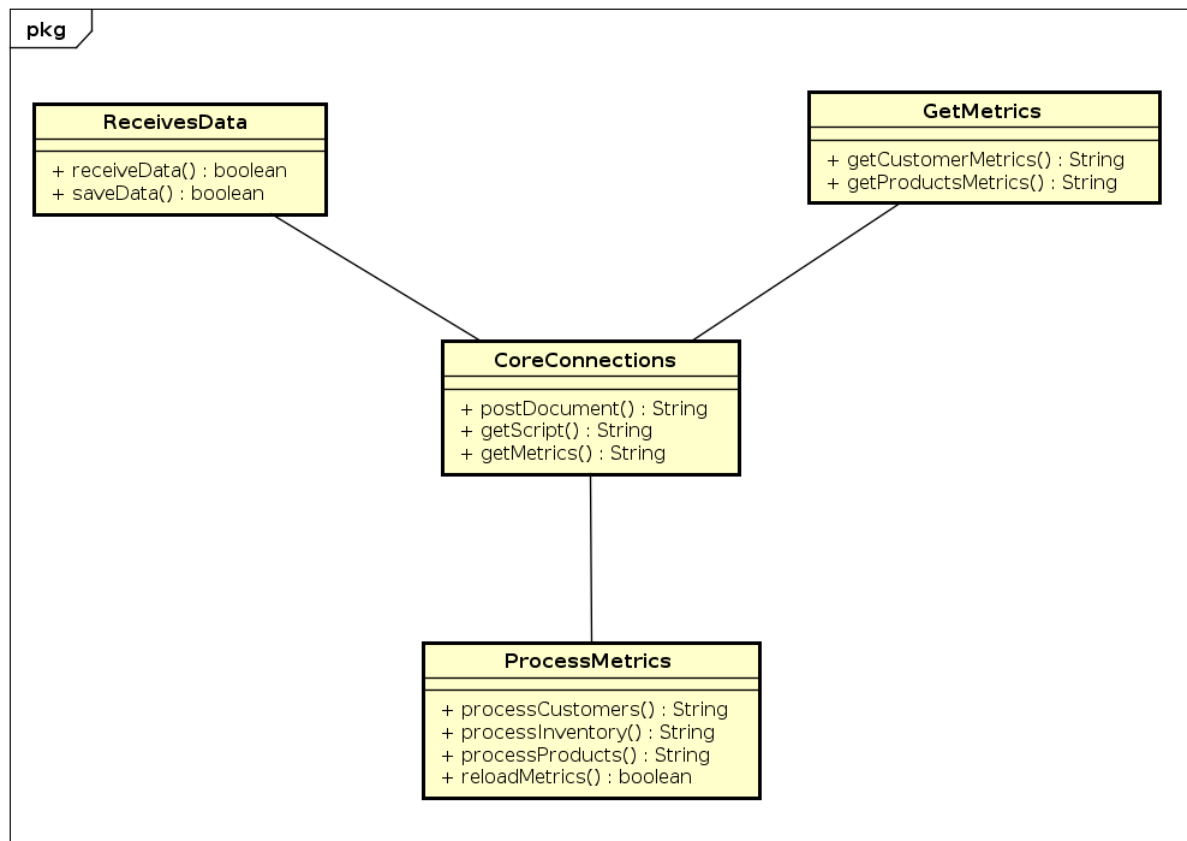
Figura 4: Exemplo de documento JSON para ser persistido no MongoDB.

Fonte: autor

O banco onde serão armazenados os dados coletados, fica fora do sistema servidor do cliente. Os dados armazenados nesse banco, ficam hospedados em um servidor distinto, que servirá como repositório de dados, para que esses possam ser trabalhados a parte, não prejudicando o desempenho da loja virtual.

Após algum tempo, rotinas automáticas, que também poderão ser executadas manualmente pelo administrador do sistema, irão solicitar à API que processe as métricas de forma a gerar novas coleções de dados; estes irão alimentar 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.

Para ilustrar melhor, na Figura 5 observa-se o diagrama de classes de como deve ser o funcionamento do módulo de coleta de dados.



powered by Astah

Figura 5: Diagrama de classe para o módulo de coleta de dados.

Fonte: autor

A classe *CoreConnections* vai ser a classe administradora das requisições, responsável de controlar o fluxo das solicitações para a API. A classe *ReceivesData* será responsável em receber os dados verificar se a loja está ativa e enviar para o banco de dados. A classe *ProcessMetrics* será responsável em processar as métricas e aplicar as funções de sumarização, e com isso gerar novas coleções ou incrementar coleções já existentes com os novos valores apurados. A classe *GetMetrics*, será responsável em servir os dados para o módulo administrativo.

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 6 mostra o conceito desse banco:

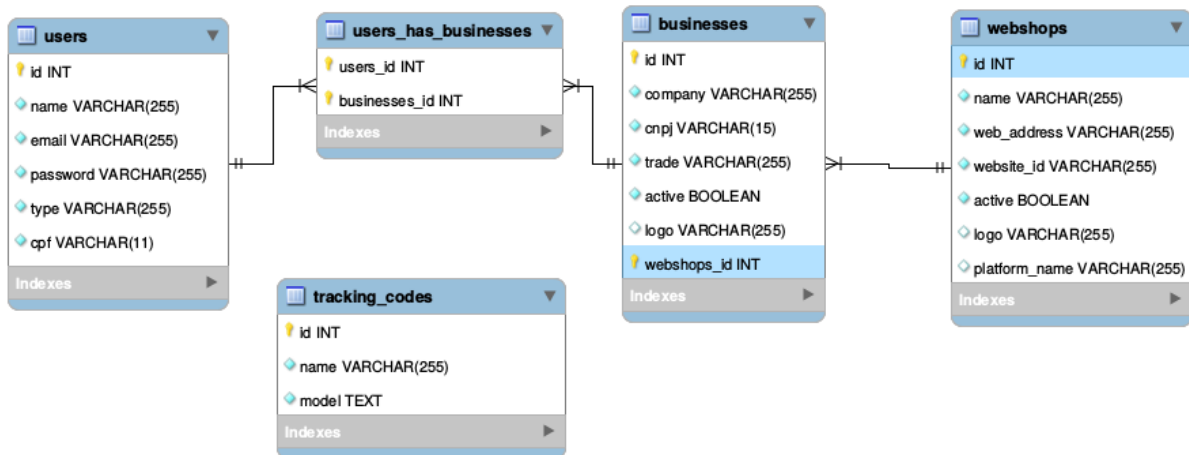


Figura 6: Banco de dados painel administrativo.

Fonte: autor

A tabela *users* é responsável pelo armazenamento dos dados dos usuários do sistema: nome, email, 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.

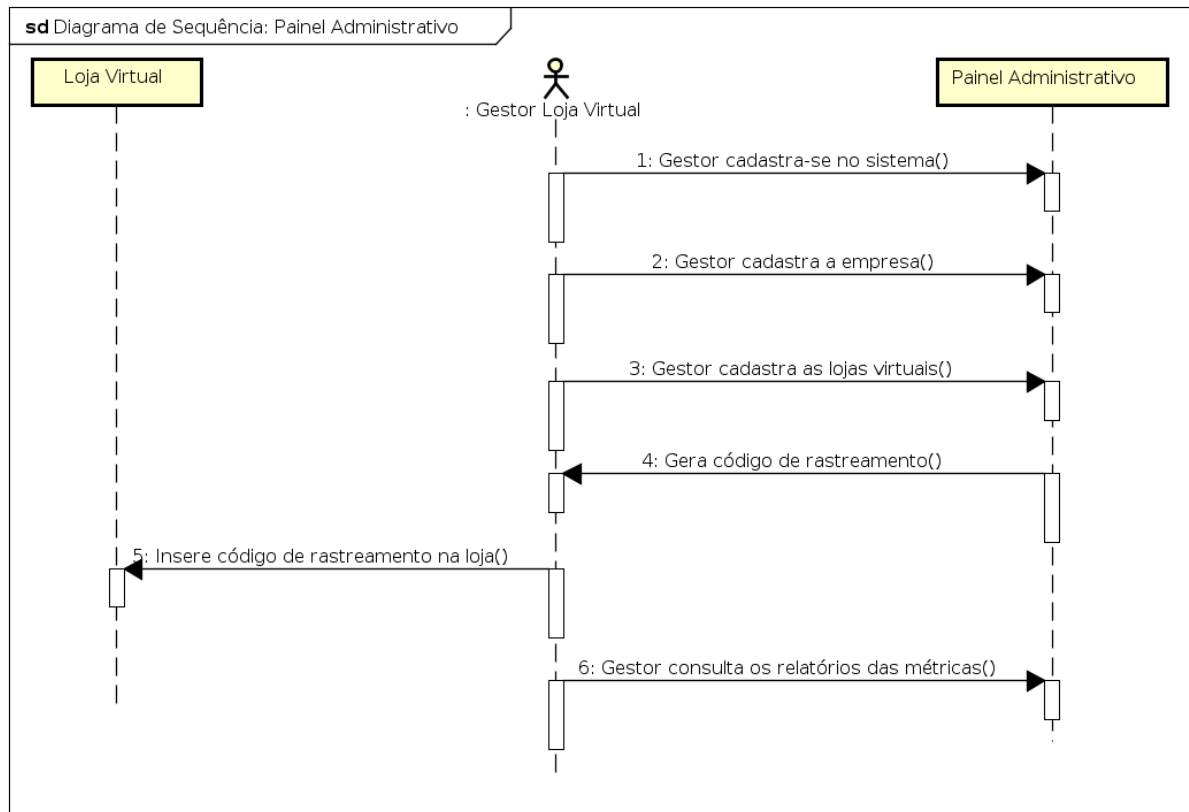
Como vários usuários podem ter várias empresas e empresas podem pertencem a vários usuários, a tabela *users_has_businesses*, serve para controlar melhor essa relação *n:m*.

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 rastreamento. 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 7



powered by Astah

Figura 7: Diagrama de sequência cadastro do gestor.

Fonte: autor

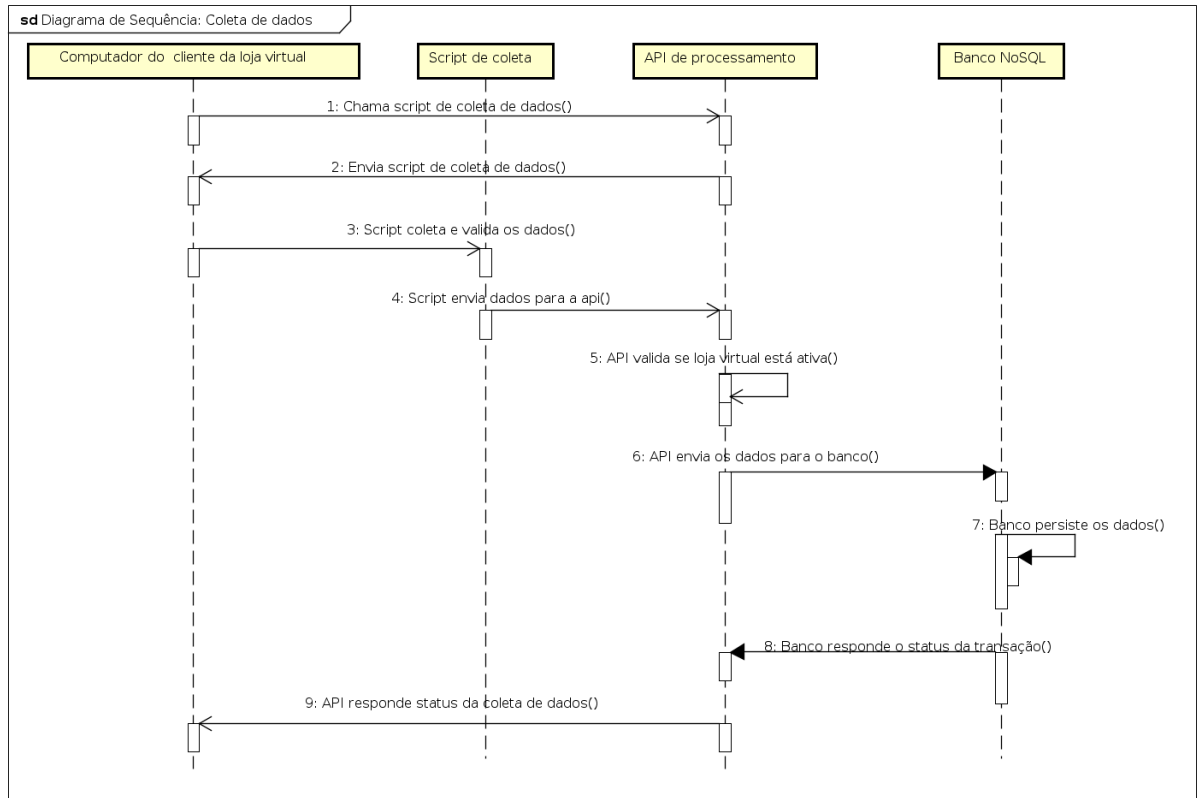
O gestor no seu primeiro uso do sistema deverá acessar o painel administrativo, para realizar o cadastro de seus dados pessoais, seu e-mail e senha serão usados para futuros acessos ao painel administrativo.

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 8.



powered by Astah

Figura 8: 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 documento 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 vir-

tual. Todo o procedimento será assíncrono, logo qualquer desventura no processo, não afetará a experiência de navegação do cliente com travamentos da página ou da aba do navegador.

5 CONSIDERAÇÕES FINAIS

Aqui elenca-se os resultados que deseja-se obter com o sistema proposto e qual a contribuição que se almeja oferecer ao gestor de loja virtual.

Com esse sistema, espera-se oferecer uma ferramenta para ajudar o gestor a obter informações sobre sua loja virtual. Uma loja virtual 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 será usado Node.js; para armazenar e trabalhar com um grande volume de dados será usado MongoDB, que permite escalabilidade no armazenamento e processamento melhorando a performance; para o sistema web será usado Ruby com Rails que favorece a produtividade na programação. É 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, diagrama de

classe, que propiciam uma melhor compreensão do funcionamento do sistema.

Entre as dificuldades está prever todas as necessidades e dependências entre os módulos, qual o melhor forma para processar as métricas e a melhor maneira para armazenar os valores obtidos e incrementá-los na medida que se obtém mais dados, mas essas situações receberão mais atenção durante o desenvolvimento do projeto.

Com o que se tem elaborado até o momento já é possível desenvolver a aplicação sugerida. Durante o processo a aplicação será amplamente testada de forma a atingir de maneira satisfatória os objetivos. Para tanto a aplicação ainda pode passar por diversas mudanças.

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.
- BRITO, K. **Fundamentos do Desenvolvimento Web**. 1. ed. 2011.
- BRODBECK, M. H. J. **Comparando ambientes de desenvolvimento Java e Ruby on Rails**. Tese (Doutorado) — UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL, 2007.
- 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.
- CALDEIRA, C. P. **PostgreSQL: Guia Fundamental**. 1ª edição. ed. Lisboa: Edições Sílabo Lda., 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.
- CRUZ, M. A. de A. S.; FIGUEIREDO, R. M.; ÁVILA, R. Mourat da R. Sistemas de controle de processos em ruby on rails. **Tecnologia & Cultura**, v. 13, n. 18, p. 69–77, 2011.

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

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.

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.

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.; SANTOS¹, 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.