

DAS Departamento de Automação e Sistemas
CTC **Centro Tecnológico**
UFSC Universidade Federal de Santa Catarina

**Detecção de anomalias na captura de dados em
lojas de e-commerce**

*Relatório submetido à Universidade Federal de Santa Catarina
como requisito para a aprovação da disciplina:
DAS5511 - Projeto de Fim de Curso*

Vinicius Kiatkoski Neves

Florianópolis, agosto de 2017

Detecção de anomalias na captura de dados em lojas de e-commerce

Vinicius Kiatkoski Neves

Esta monografia foi julgada no contexto da disciplina
DAS5511 - Projeto de Fim de Curso
e aprovada na sua forma final pelo
Curso de Engenharia de Controle e Automação

Prof. Ubirajara Franco Moreno, PhD.

Assinatura do orientador

Detecção de anomalias na captura de dados em lojas de e-commerce

Vinicius Kiatkoski Neves

Esta monografia foi julgada no contexto da disciplina DAS5511 - Projeto de Fim de Curso e aprovada na sua forma final pelo Curso de Engenharia de Controle e Automação

Florianópolis, 09 de agosto de 2017

Banca examinadora

Wagner Francisco Mezaroba
Orientador na empresa
Linx Sistemas e Consultoria Ltda.

Prof. Ubirajara Franco Moreno
Orientador no curso
Universidade Federal de Santa Catarina

Prof. Carlos Barros Montez
Avaliador
Universidade Federal de Santa Catarina

Arthur Seidelmann Schweder
Debatedor
Universidade Federal de Santa Catarina

Matheus Sperb Machado
Debatedor
Universidade Federal de Santa Catarina

Resumo

O presente trabalho foi desenvolvido na empresa Linx Sistemas e Consultoria Ltda., líder no Brasil em personalização de comércio eletrônico. Neste trabalho foi desenvolvido um algoritmo de detecção de falhas na coleta de eventos em lojas de comércio eletrônico. O algoritmo foi testado em dados reais provenientes de coleta da empresa e cenários de falha foram isolados para seu desenvolvimento. O mesmo procura melhorar o processo de detecção de falhas, hoje feito por um outro algoritmo com escopo limitado de detecção e muito suscetível a falsos-positivos. O algoritmo desenvolvido baseia-se no histórico do cliente (conforme o dia da semana) para então concluir sobre uma possível falha, utilizando uma abordagem estatística para tal, baseada em médias e desvio padrão dos dados dos clientes. Os dois algoritmos são comparados diariamente e como resultado salienta-se as melhorias do algoritmo proposto, tanto no escopo de detecção, uma abrangência maior de falhas podem ser detectadas, quanto na redução de falsos-positivos.

Palavras-chave: detecção de falhas. comércio eletrônico. falsos-positivos. coleta de eventos.

Abstract

The current work was developed at Linx Sistemas e Consultoria Ltda., Brazil's leading e-commerce personalization company. In this work, a fault-detection algorithm was developed for capturing events in e-commerces. The algorithm was tested within company's real data and failure scenarios were isolated for its development. The algorithm intends to improve fault-detection process, which is currently performed by another algorithm with limited scope and very susceptible to false positives. The developed algorithm uses client's historical data (according to the day of the week) to detect a possible failure through a statistical approach based on the means and standard deviation of the customer data. Both algorithms are compared on a daily basis. As a result of the study, we highlight the improvements perceived with the proposed algorithm, both in the scope of detection, as a greater range of failures can be detected, as well as in the reduction of false positives.

Palavras-chave: fault-detection. e-commerce. false positive. event capturing.

Lista de ilustrações

Figura 1 - Funil de vendas do e-commerce.....	17
Figura 2 - Visualizações de página ao longo de um dia.....	18
Figura 3 - Diferenças de visualizações de página entre dias de semana e finais de semana.....	19
Figura 4 - Visualizações de página, produto, carrinho e compras ao longo de um dia.....	20
Figura 5 - Ausência completa de eventos após falha.....	24
Figura 6 - Redução na captura de eventos após falha.....	25
Figura 7 - Aumento abrupto de eventos após falha.....	26
Figura 8 - Visualizações de página por dia da semana.....	43
Figura 9 - Comportamento de acesso por hora para 8 semanas.....	44
Figura 10 - Comportamento de acessos por hora para 24 semanas.....	45
Figura 11 - Variação em relação a média para 5 clientes nos 6 primeiros meses do ano.....	46
Figura 12 - Variação em relação a média para 5 clientes nos 6 primeiros meses do ano (ampliada).....	47
Figura 13 - Distribuição ao entorno da média com região de rejeição.....	48
Figura 14 - Distribuição de médias e limite de dois desvios padrões de rejeição.....	49
Figura 15 - Exemplo de falha com queda na captura de eventos.....	51
Figura 16 - Falha detectada pelas duas ferramentas.....	52
Figura 17 - Falso-positivo detectado por promoção na página do cliente.....	53

Lista de tabelas

Tabela 1 - Tipos de páginas disponíveis.....	13
Tabela 2 - Página de produto simplificada.....	14
Tabela 3 - Página de produto completa.....	15
Tabela 4 - Página de carrinho.....	16
Tabela 5 - Página de confirmação de compra.....	16
Tabela 6 - Índices de severidade.....	28
Tabela 7 - Índices de ocorrência.....	32
Tabela 8 - Índice de ocorrência para cada modo potencial de falha.....	32
Tabela 9 - Índices de detecção.....	36
Tabela 10 - Índice de detecção para cada controle atual.....	37
Tabela 11 - Expansão do NPR para as falhas apontadas.....	39
Tabela 12 - Comparação de 15 dias entre ferramentas.....	50
Tabela 13 - Quadro FMEA completo para PageView.....	58

Sumário

1 INTRODUÇÃO	10
2 DESCRIÇÃO DO PROCESSO	12
2.1 Modelo de integração	12
2.2 Comportamento de acesso de e-commerces	16
3 ANÁLISE DE FALHAS DO PROCESSO	21
3.1 Componentes	21
3.2 Função dos componentes	21
3.2.1 Visualização de página (<i>PageView</i>).....	22
3.2.2 Visualização de produto (<i>ProductView</i>).....	22
3.2.3 Visualização de carrinho (<i>CartView</i>).....	22
3.2.4 Confirmação de compra (<i>Transaction</i>).....	22
3.3 Modo potencial de falha	23
3.4 Efeitos	23
3.4.1 Remoção de script (MP1).....	24
3.4.2 Erro de script (MP2).....	24
3.4.3 Inicialização duplicada (MP3).....	25
3.4.4 Preenchimento incorreto do meta (MP4).....	26
3.4.5 Não inicialização do script (MP5).....	26
3.4.6 Falhas no servidor (MP6).....	27
3.5 Severidade	27
3.5.1 Gráfico zerado (E1).....	28
3.5.2 Redução na captura de eventos (E2).....	29
3.5.3 Aumento na captura de eventos (E3).....	29
3.6 Causas potenciais da falha	29
3.6.1 Remoção de script (MP1).....	29
3.6.2 Erro de script (MP2).....	30

3.6.3 Inicialização duplicada (MP3).....	30
3.6.4 Preenchimento incorreto do meta (MP4).....	30
3.6.5 Não inicialização do script (MP5).....	31
3.6.6 Falhas no servidor (MP6).....	31
3.7 Ocorrência.....	31
3.8 Controles atuais.....	33
3.8.1 Remoção de script (MP1).....	33
3.8.2 Erro de script (MP2).....	34
3.8.3 Inicialização duplicada (MP3).....	34
3.8.4 Preenchimento incorreto do meta (MP4).....	35
3.8.5 Não inicialização do script (MP5).....	35
3.8.6 Falhas no servidor (MP6).....	35
3.9 Detecção.....	36
3.10 Número de prioridade de risco (NPR).....	37
3.11 Ações recomendadas.....	40
3.11.1 Detecção baseada no histórico do cliente.....	41
4 FERRAMENTAS PARA DETECÇÃO DE FALHAS.....	42
4.1 Análise do problema.....	42
4.2 Solução proposta.....	47
5 RESULTADOS.....	50
6 CONCLUSÃO.....	54
REFERÊNCIAS.....	56
ANEXO A - Tabela FMEA Completa.....	57

1 INTRODUÇÃO

O presente trabalho, *Detecção de Anomalias na Captura de Dados em Lojas de E-commerce*, foi desenvolvido na empresa Chaordic, hoje conhecida como Linx Sistemas e Consultoria Ltda. após aquisição da mesma em setembro de 2015. A Chaordic atua no setor de sistemas de recomendação para comércio eletrônico no Brasil e no exterior, sendo sua sede localizada na cidade de Florianópolis - SC.

Atualmente existem 3 produtos principais oferecidos pela empresa: vitrines personalizadas, e-mails personalizados e busca personalizada, todos com a premissa de tornar única a experiência do usuário com o *e-commerce*, e assim melhorar a conversão de possíveis consumidores. Além disso, a empresa trabalha no desenvolvimento de outros produtos no mesmo setor, porém alguns em fase projeto e outros que já foram descontinuados pela mesma.

A principal fonte de informação para a empresa são os dados capturados da navegação dos usuários nas lojas de comércio eletrônico e a base de produtos de cada loja. Nesse cenário, foco principal do trabalho, pretende-se minimizar o efeitos de possíveis falhas nas capturas desses dados de modo que o impacto para a empresa que contratou a solução seja a menor possível.

A falha na captura de dados pode impactar diretamente na qualidade das recomendações oferecidas pela empresa, uma vez que os dados alimentam o processo de geração de recomendações. Como os produtos da empresa aumentam a conversão do *e-commerce*, recomendações ruins diminuem essa conversão, fazendo com que a loja deixe de ganhar dinheiro caso o problema não seja detectado o quanto antes.

Os principais dados capturados que influenciam na geração de recomendações são os dados provenientes da navegação do usuário na loja, principalmente as páginas visitadas pelo mesmo (páginas de produto, categorias, carrinho de compras, etc) e os produtos comprados (chamadas de páginas de transação). Outros dados são capturados, porém estes servem, entre outras coisas, para medir o desempenho do produto.

O objetivo deste trabalho consiste então em detectar falhas nos dados de navegação de usuário e apresentar um algoritmo para melhorar a detecção de falhas e redução de falso-positivos em comparação com o sistema atual. O enfoque é dividido entre páginas de produto, páginas de carrinho de compras, páginas de transação e outras páginas (incluem qualquer visitada, incluindo as mencionadas anteriormente). As falhas podem acontecer por diversos motivos e os impactos podem ser variados: redução de captura de dados, aumento na captura de dados por duplicação de eventos, e não captura de dados (caso específico de redução de captura).

A metodologia utilizada para estudar as possíveis falhas do sistema parte de um processo de análise do problema utilizando FMEA (*Failure Mode and Effect Analysis* - Análise dos Modos de Falha e seus Efeitos). Em seguida, com base no resultado da aplicação do FMEA, é estudada uma ação para reduzir os modos de falha levantados.

A ação escolhida é implementada e validada, comparando-a com sistemas semelhantes já existentes na empresa. Além disso, após o estudo e levantamento dos dados da ação escolhida, outros problemas e possíveis ações a serem aplicadas nos mesmos são apresentadas, de modo a deixar novas oportunidades de estudo para o problema em aberto.

O presente documento encontra-se dividido em cinco capítulos, a partir da introdução. O capítulo dois introduz o processo de captura de eventos, como ele funciona e sua importância para empresa. No capítulo três é feita uma análise de modos de falha e seus efeitos, FMEA, do processo, de modo a justificar a ferramenta desenvolvida no presente trabalho.

A descrição da ferramenta desenvolvida e os métodos estatísticos utilizados são apresentados no capítulo quatro. Já o capítulo cinco apresenta os resultados obtidos, fazendo um comparativo entre a solução desenvolvida e a ferramenta utilizada atualmente pela empresa.

O capítulo seis sumariza toda a análise e desenvolvimento feito no presente documento, e explora possíveis pontos de melhoria para o algoritmo desenvolvido, abrindo portas para continuidade no desenvolvimento da ferramenta.

2 DESCRIÇÃO DO PROCESSO

O processo de captura de dados dos usuários depende de uma integração entre o site do cliente e a solução oferecida pela empresa. Essa integração acontece pela inserção de um *script* na página do cliente. Esse *script*, também conhecido por *loader*, é mantido pela empresa e é responsável por iniciar todo o processo de captura.

O *loader* captura informações da página que são preenchidas pelo cliente. Essas informações são preenchidas em uma variável da página chamada *chaordic_meta*, conhecida também por *meta*. O *meta* contém informações gerais da página (como o tipo de página sendo visitada) e também informações mais específicas de alguns tipos de página (como o identificador de um produto caso a página visitada seja de produto).

2.1 Modelo de integração

A tabela 1 apresenta o valor que pode ser atribuído a cada tipo de página do cliente no *meta*. O preenchimento do tipo de página define quais dados adicionais serão necessários.

Tabela 1 - Tipos de páginas disponíveis.

Tipo de página	<i>chaordic_meta.page</i>
Home (Página inicial)	home
Categoria/Departamento (1º nível)	category
Subcategoria (2º nível ou mais)	subcategory
Produto	product
Carrinho	cart
Checkout (Etapas do carrinho)	checkout
Confirmação de compra	confirmation
Busca	search
Busca vazio (Sem resultados)	emptysearch
Landing page	landing_page
Perfil do usuário	userprofile
Erro 404	not_found
Outras	other

Fonte - [1]

Como o presente trabalho estará focado na análise de páginas de produto, carrinho e confirmação de compra, será preciso apresentar os dados necessários em cada uma dessas páginas, já que serão mencionados posteriormente no documento.

As páginas de produto são preenchidas de acordo com as informações das tabelas 2 e 3. A tabela 2 apresenta uma integração simplificada, onde menos dados do produto são fornecidos, e a tabela 3 apresenta uma integração completa, onde mais dados do produto são fornecidos. A diferença entre integrações decorre do fato de que a versão simplificada é uma novidade do modelo de integração da empresa. Por esse motivo ainda existem dois tipos de integração sendo utilizados.

Tabela 2 - Página de produto simplificada.

Propriedade	Valor
*page	<i>"product"</i>
*pid	<i>String</i> - Identificador único do produto
sku	<i>String</i> - Identificador da variação do produto
price	<i>Number</i> - Preço de comercialização produto

* Indicam propriedades com preenchimento obrigatório

Fonte - [2]

Tabela 3 - Página de produto completa.

Propriedade	Valor
*id	<i>String</i> - Identificador único do produto
*name	<i>String</i> - Título do produto
*description	<i>String</i> - Descrição do produto
*url	<i>String</i> - URL da página do produto
*images	<i>Object</i> - Caminho da imagem do produto.
*status	<i>String</i> - Disponibilidade do produto em estoque. Valores permitidos: "AVAILABLE" e "UNAVAILABLE".
*categories	<i>Array de Strings</i> - Conjunto de categorias relacionadas ao produto.
*price	<i>Number</i> - Preço de comercialização do produto.
**specs	<i>Object</i> - Características e valores das variantes do produto.
**skus	<i>Array de Objetos</i> - Informações para cada variante do produto.
details	<i>Object</i> - Detalhes do produto.
installment	<i>Object</i> - Informações do parcelamento do produto.

* Indicam propriedades com preenchimento obrigatório.

** Obrigatório para produtos com mais de uma variante.

Fonte - [3]

As página de carrinho e confirmação de compra são preenchidas de forma semelhante para as duas integrações, e são apresentadas nas tabelas 4 e 5, respectivamente.

Tabela 4 - Página de carrinho.

Propriedade	Valor
*page	" <i>cart</i> "
*items	<i>Array de Objetos</i> - Lista dos produtos presentes no carrinho.
id	<i>String</i> - Identificador único do carrinho.

* Indicam propriedades com preenchimento obrigatório.

Fonte - [4]

Tabela 5 - Página de confirmação de compra.

Propriedade	Valor
*page	" <i>transaction</i> "
*id	<i>String</i> - Identificador único da transação.
*items	<i>Array de Objetos</i> - Lista de itens comprados.

* Indicam propriedades com preenchimento obrigatório.

Fonte - [5]

De posse dessas informações preenchidas no *meta*, são feitas requisições para os servidores da empresa enviando esses dados. As informações são salvas na base, separadas por cliente e por tipo de evento.

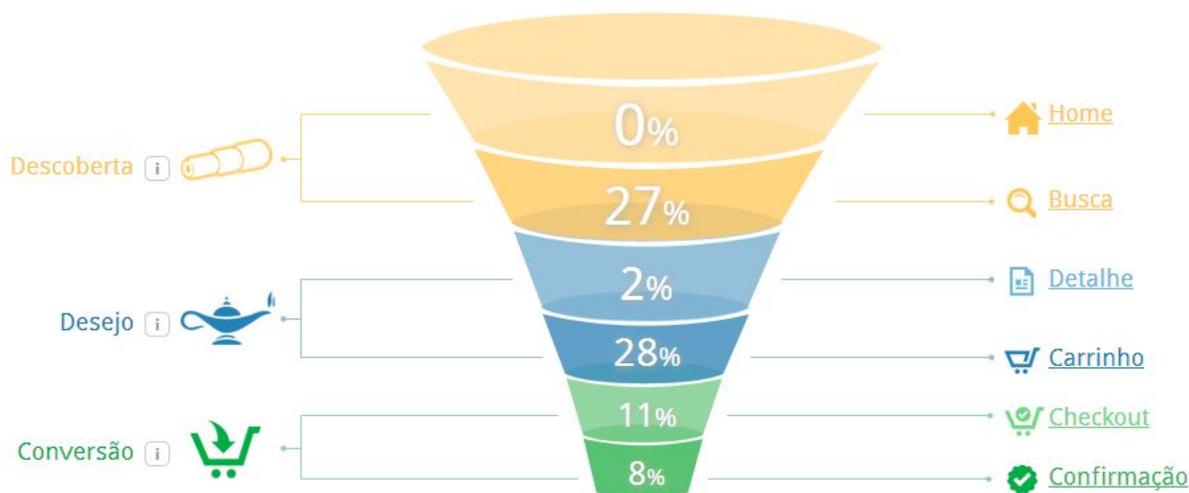
2.2 Comportamento de acesso de e-commerces

Como mencionado anteriormente, os eventos de interesse são de visualização de página, visualização de produto, visualização de carrinho de compra e confirmação de compra. A quantidade de eventos gerados obedecem uma regra de funil de vendas e variam de acordo com o tamanho do cliente e do período do dia.

O funil de vendas de um comércio eletrônico é apresentado na figura 1, onde o item "detalhe" é equivalente a página de produto. Essa figura ilustra o comportamento dos eventos gerados pela loja, em que visualizações de página

geram mais eventos, seguida por visualizações de produto, depois por visualizações de carrinho e, por fim, compras.

Figura 1 - Funil de vendas do e-commerce.



Fonte - Blog Luz Lab Ideias¹

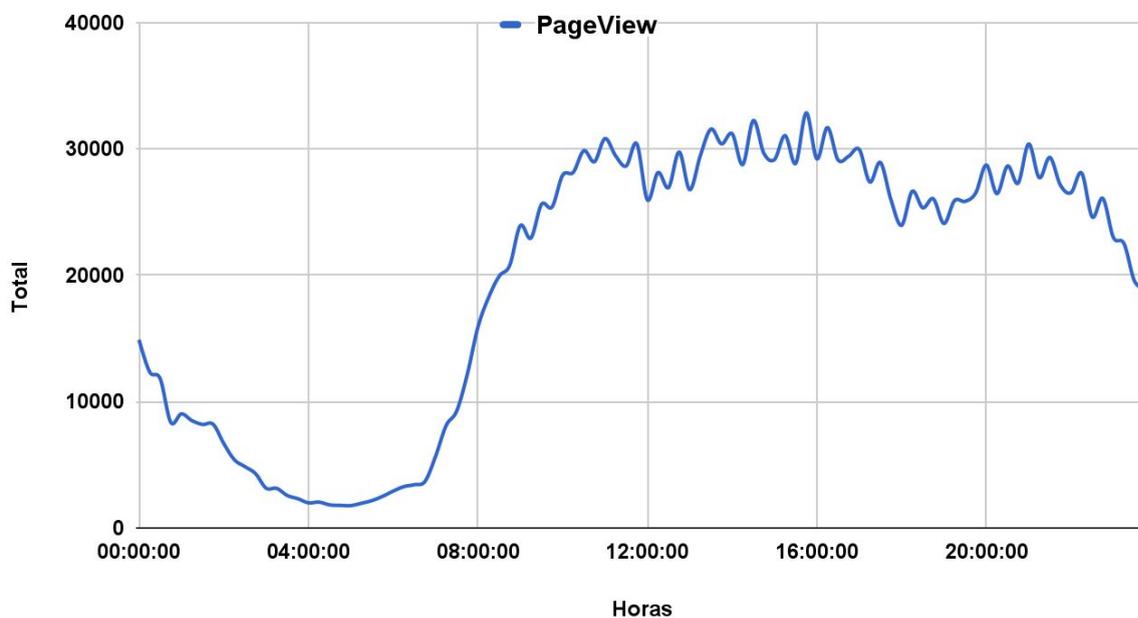
O tamanho de um cliente influencia diretamente na quantidade de dados, enquanto o período do dia tem uma influência maior na distribuição dos eventos ao longo do dia. O tamanho então influencia apenas na quantidade e não na distribuição.

Considerando um cliente situado no Brasil, durante a noite (período no qual a maioria das pessoas está dormindo) a quantidade de acessos nas lojas cai consideravelmente. Já durante o dia, ela cresce e permanece num patamar parecido ao longo da tarde, até o final da noite. Esse comportamento pode ser visto na figura 2, que reflete o comportamento de compra dos brasileiros ao longo do dia (24h que compõem um dia).

¹ Funil de vendas, utilize os números. Disponível em: <<http://blog.luz.vc/o-que-e/funil-de-vendas>>. Acesso em 20 de junho de 2017.

Figura 2 - Visualizações de página ao longo de um dia.

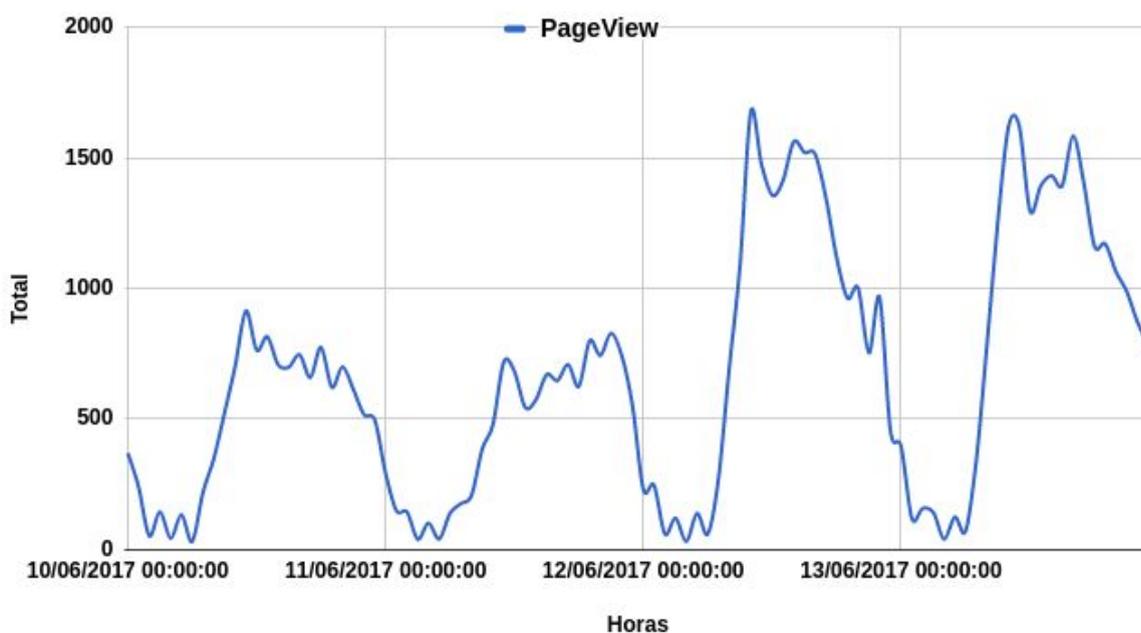
Visualizações de página



Além disso, vale ressaltar que para a maioria dos clientes (relacionado ao nicho no qual o cliente atua) o comportamento dos dias úteis difere dos finais de semana (em quantidade), onde há menos eventos em geral. A figura 3 ilustra esse comportamento, em que os dias 10/06 e 11/06 representam sábado e domingo, respectivamente.

Figura 3 - Diferenças de visualizações de página entre dias de semana e finais de semana.

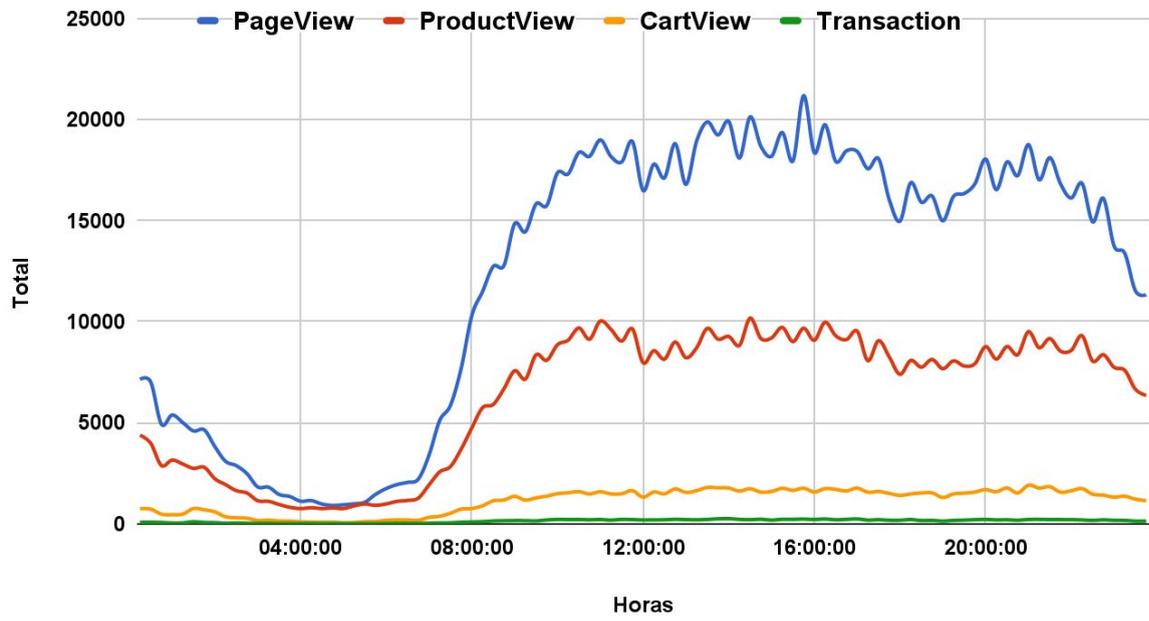
Visualizações de página



A figura 4 exemplifica o processo descrito anteriormente como funil de vendas para uma loja específica. *PageView* corresponde a visualizações de página (neste caso desconsiderando páginas de produto, carrinho e confirmação de compra), *ProductView* como visualizações de produto, *CartView* como visualizações de carrinho e *Transaction* como compras. As proporções dos eventos segue o que foi apresentado no funil, ou seja, a quantidade de visitas no site é muito maior que o de compras.

Figura 4 - Visualizações de página, produto, carrinho e compras ao longo de um dia.

Visualizações de página, produto, carrinho e compras



3 ANÁLISE DE FALHAS DO PROCESSO

A captura de eventos apresentada no capítulo anterior pode sofrer falhas, como será apresentado neste capítulo. Estas falhas podem resultar em prejuízos para o cliente, que vão desde pagar por uma solução que não está funcionando até uma má performance dos produtos oferecidos pela empresa. Portanto, analisar as falhas que possam ocorrer no processo e desenvolver ferramentas capaz de capturá-las se faz de extremo interesse.

Para análise de falhas do processo de captura de eventos, utilizou-se do processo de FMEA (*Failure Mode and Effect Analysis* - Análise de Modos de Falhas e Efeitos) que é bem conhecido e conceituado no mundo da engenharia.

Análise FMEA (Failure Mode and Effect Analysis) é uma metodologia que objetiva avaliar e minimizar riscos por meio da análise das possíveis falhas (determinação da causa, efeito e risco de cada tipo de falha) e implantação de ações para aumentar a confiabilidade. - [6]

3.1 Componentes

Os componentes são de fato o que será analisado no processo. Como já mencionado, serão analisados os eventos de visualização de página, visualização de produtos, visualização de carrinho e compra. Vale destacar que toda página visitada, seja ela produto, carrinho ou compra, gera uma visualização de página. Os eventos de produto, carrinho e compra são extras à visualização de página gerada.

Como descrito no capítulo 2, cada um desses eventos possui dados específicos que os caracterizam. Portanto o preenchimento correto das informações é de fundamental importância para o registro correto dos eventos.

3.2 Função dos componentes

A função dos componentes é descrita por [7] como “[...] a intenção, propósito, meta ou objetivo do componente”. A seguir será descrita a função de cada componente mencionado.

3.2.1 Visualização de página (*PageView*)

Visualização de página é o evento mais geral coletado pela empresa. Toda vez que uma página é aberta ou atualizada, um novo evento é gerado. Ele indica que uma página foi visitada e ainda especifica essa página, de acordo com a tabela 1.

3.2.2 Visualização de produto (*ProductView*)

Visualização de produto é equivalente a uma visualização de página, porém específica para páginas de produto, ou seja, toda vez que uma página de produto for visitada será contabilizado um *ProductView* e também um *PageView*, pois esses dois eventos serão enviados para o sistema. As páginas de produto ainda requerem que informações dos produtos sejam enviadas nos eventos - tabelas 2 e 3 - de modo que essa informação sirva de insumo para as soluções oferecidas pela empresa.

3.2.3 Visualização de carrinho (*CartView*)

Assim como a visualização de produto, a visualização de carrinho também envia um evento de *PageView*. Toda vez que uma página identificada como carrinho for visitada ela enviará um evento de *CartView*, com informações as informações extras descritas na tabela 4. As informações enviadas pelos eventos de *CartView* também servem de insumo para as soluções oferecidas pela empresa.

3.2.4 Confirmação de compra (*Transaction*)

Semelhante a *CartView* e *ProductView*, porém nesse caso as informações relacionadas à compra dos produtos estão presentes nos eventos. Esse evento também é usado como insumo para as soluções oferecidas pela empresa, além disso, é extremamente importante para avaliar o impacto das soluções usadas pelo cliente em seu faturamento.

3.3 Modo potencial de falha

O modo potencial de falha é o que se deseja analisar, ou seja, como as falhas podem ocorrer. Todos os componentes analisados apresentam os mesmos modos potenciais de falha, uma vez que os eventos são apenas separados por tipo de página. A análise parte de vários eventos, pois não necessariamente um problema nos eventos de visualização de carrinho impactam em visualização de produto, já que são páginas totalmente diferentes.

Os modos potenciais de falha são listados abaixo e uma breve explicação de cada um é apresentada:

- MP1 - Remoção de *script*: caso o *script* da empresa seja removido da página do cliente, não é possível coletar dados, uma vez que esse é o único ponto de entrada das soluções da empresa na página do cliente.
- MP2 - Erro de *script*: ocorre alguma exceção na execução do código do *script* e a mesma não é tratada. Dessa forma, o *script* não é executado por inteiro.
- MP3 - Inicialização duplicada: o *script* acaba sendo inicializado múltiplas vezes por algum *script* externo presente na página do cliente.
- MP4 - Preenchimento incorreto do *meta*: o cliente é responsável por preencher os dados de coleta, mas por algum motivo os dados podem ser preenchidos de forma incorreta.
- MP5 - Não inicialização do *script*: por mais que o *script* esteja presente na página, ele precisa ser inicializado (Caso não configurado para inicialização automática).
- MP6 - Falhas no servidor: servidores que recebem as requisições com os eventos podem falhar e com isso os eventos acabam sendo perdidos.

3.4 Efeitos

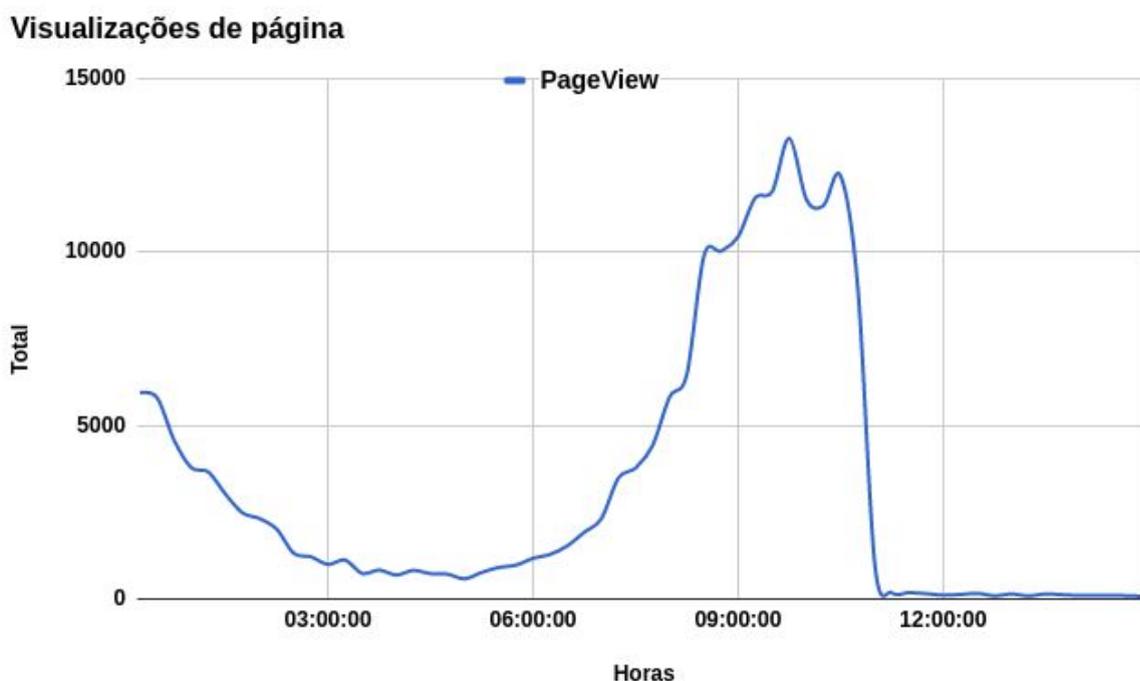
Os efeitos “são as conseqüências que surgem no sistema causadas pelos modos de falha” [7]. Portanto, cada modo potencial de falha pode ter um ou mais

efeitos no processo, além disso, modos de falha diferentes podem resultar nos mesmos efeitos, como é apresentado a seguir.

3.4.1 Remoção de script (MP1)

Ao remover o script de coleta de eventos da página nenhum evento é coletado. Por consequência, os gráficos de registros de eventos terão uma queda abrupta e aparecerão zerados. A figura 5 apresenta esse cenário, onde próximo às 12h houve uma queda abrupta nos eventos de *PageView* registrados.

Figura 5 - Ausência completa de eventos após falha.



3.4.2 Erro de *script* (MP2)

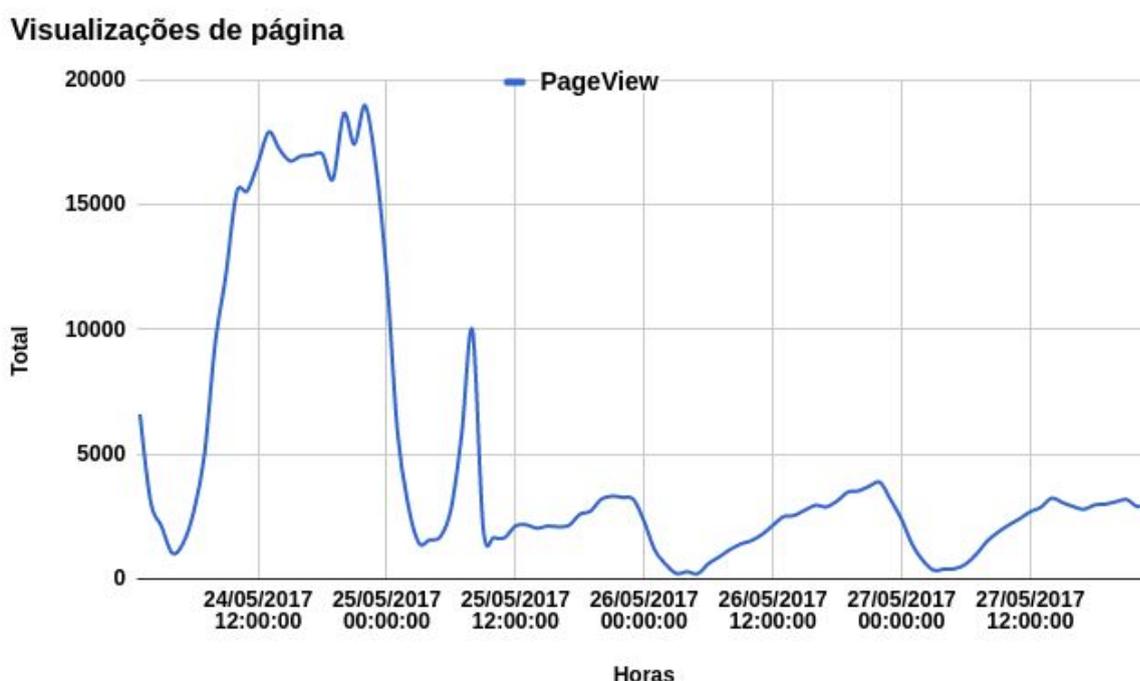
Um erro de *script* é causado por algum código mal projetado, que acaba lançando uma exceção que não é tratada durante sua execução. Um caso muito comum é o acesso a variáveis não existentes.

Os efeitos de um erro de *script* podem se manifestar de duas maneiras: nenhum evento coletado (semelhante ao caso explorado anteriormente) ou eventos

coletados de forma intermitente, ou seja, o código mal projetado lança exceção somente para algumas circunstâncias. Dessa forma há uma queda não abrupta no registro de eventos.

A figura 6 ilustra o segundo caso, onde há uma queda não abrupta no registro de eventos. Percebe-se que ainda existem eventos sendo registrados, porém em menor quantidade após a manhã do dia 25/05 e os dias seguintes.

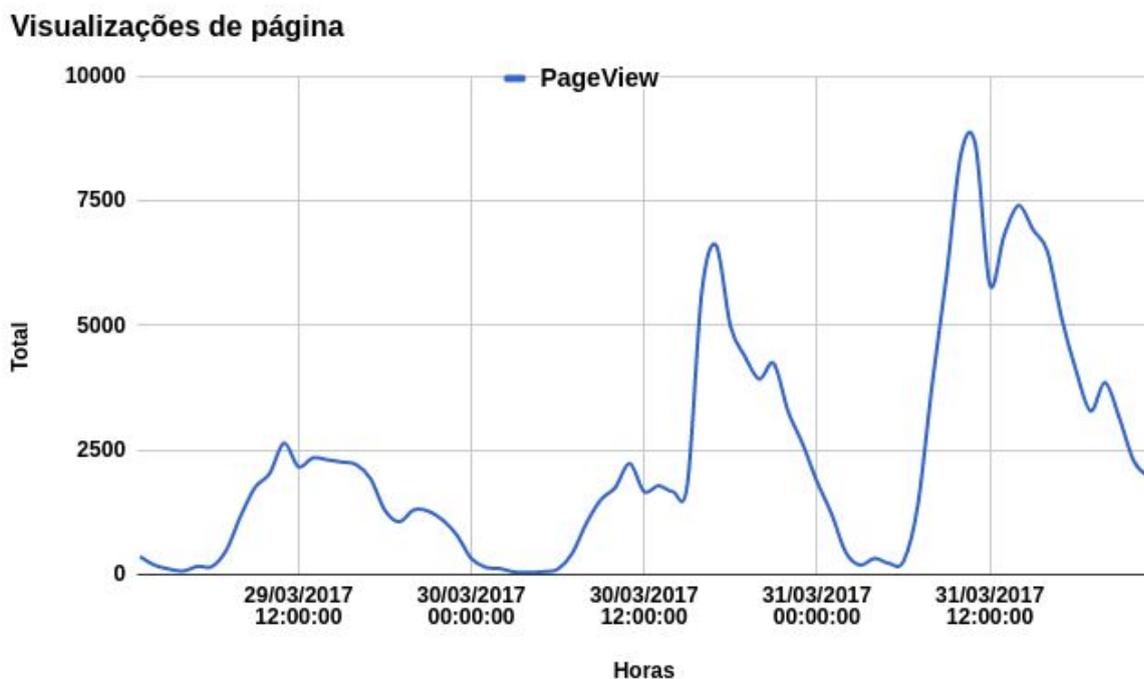
Figura 6 - Redução na captura de eventos após falha.



3.4.3 Inicialização duplicada (MP3)

Caso o *script* seja inicializado mais de uma vez, ocorrerá uma duplicação dos eventos enviados, pois para cada inicialização os dados são coletados e enviados. O efeito nesse caso é oposto aos já mencionados, ou seja, há um aumento brusco na quantidade de eventos. A figura 7 ilustra esse caso, onde após às 12h do dia 30/03 os eventos aumentaram de forma abrupta e permanecem dessa maneira para os dias seguintes.

Figura 7 - Aumento abrupto de eventos após falha.



3.4.4 Preenchimento incorreto do *meta* (MP4)

O preenchimento incorreto das informações da página pode levar a dois casos já apresentados: gráfico zerado ou diminuição na captura de eventos. Ao preencher uma informação errada, pode-se ocasionar uma captura equivocada de um dado. Por exemplo, caso seja visitada uma página de produto e os dados preenchidos forem para uma página inicial, o evento registrado estará equivocado e não será registrado um *ProductView*.

Caso os dados preenchidos incorretamente ocorram para todos os casos, teremos um gráfico zerado. Caso esse preenchimento seja eventual, para uma determinada condição, teremos uma diminuição na captura de eventos, semelhante a figura 6.

3.4.5 Não inicialização do *script* (MP5)

Se o *script* for inserido na página, porém não inicializado (por padrão a inicialização do *script* é automática), não haverá captura dos eventos. O resultado

disso é o mesmo apresentado para MP4, uma vez que a inicialização também pode estar relacionada a uma condição específica, portanto ocorrendo eventualmente.

3.4.6 Falhas no servidor (MP6)

Os servidores que recebem os eventos (requisições) podem falhar, ocasionando como resultado um gráfico zerado - caso todos os servidores falhem. Ou ainda pode haver uma diminuição de captura de eventos, caso apenas alguns servidores falhem.

3.5 Severidade

A severidade é definida como “um índice que indica o quão sério é o efeito do modo de falha potencial. [...] Quanto mais grave e crítico o efeito, maior é o índice de severidade” [7]. A tabela 6 apresenta uma lista de critérios para se definir o grau de severidade de um modo de falha.

Tabela 6 - Índices de severidade.

Escore	Critério
1	O cliente provavelmente não tomará conhecimento
2	Leve aborrecimento
3	
4	Insatisfação do cliente
5	
6	
7	Alto grau de insatisfação
8	
9	Atinge as normas de segurança
10	

Fonte - [8]

Os índices devem ser distribuídos para cada um dos efeitos mapeados na seção anterior: gráfico zerado, redução na captura de eventos e aumento na captura de eventos. Como os impactos são os mesmos, independente do modo de falha, não é necessário explorar individualmente os índices de severidade por efeito de modo de falha.

3.5.1 Gráfico zerado (E1)

Um gráfico zerado, onde nenhum dado está sendo carregado, e, por consequência a solução oferecida pela empresa está comprometida, é considerado de alta severidade. Em alguns casos, o gráfico pode zerar para apenas algum tipo de página, diminuindo a severidade da falha, porém esses são casos isolados.

Um índice 7 é devidamente aplicado a este caso, já que, como mencionado, o cliente acaba percebendo que a solução não está funcionando, causando grande descontentamento pelo fato do problema não ser resolvido pró-ativamente.

3.5.2 Redução na captura de eventos (E2)

A redução na captura de eventos, se não tratada rapidamente, pode afetar o desempenho da solução e, por sua vez, gerar descontentamento do cliente. Um índice 4 é adequado para este caso, uma vez que a deterioração do desempenho da solução não é imediata.

3.5.3 Aumento na captura de eventos (E3)

Um aumento na captura de eventos dificilmente é percebido pelo cliente, somente quando comportamentos anômalos são apresentados pela solução. Um índice 2 justifica essa condição. O impacto maior é sentido do lado da empresa onde uma quantidade alta de eventos duplicados precisam ser descartados.

3.6 Causas potenciais da falha

“Causas e condições que podem ser responsáveis pelo tipo de falha em potencial” [6], ou seja, a origem da falha. Dado que a solução oferecida pela empresa depende de um processo de integração, que pode ser demorado e complexo, diversos erros podem ocorrer durante e após esse processo, de modo que diversas causas podem ser mapeadas e analisadas separadamente.

Para cada modo potencial de falha podemos ter mais de uma causa. Algumas já foram exploradas no capítulo 3.3, porém serão detalhadas a seguir.

3.6.1 Remoção de *script* (MP1)

A inserção do *script*, que irá inicializar a coleta e a solução, na página do cliente é de inteira responsabilidade do mesmo. Portanto, a causa está relacionada a alguma alteração que o próprio cliente fez em sua página, geralmente causada após uma atualização de conteúdo/*layout*.

3.6.2 Erro de *script* (MP2)

A manutenção do *script* é de responsabilidade total da empresa que oferece a solução, portanto qualquer erro causado em sua inicialização está relacionado a um código mal projetado. Geralmente, este erro está relacionado a customizações mal projetadas e que já foram oferecidas ao cliente, para tratamento de outros erros.

3.6.3 Inicialização duplicada (MP3)

A inicialização duplicada pode ser de responsabilidade tanto do cliente quanto da empresa. No caso da empresa, a causa está relacionada a uma customização mal projetada para o cliente. Esses casos são mais simples e rápidos de resolver.

Quando a inicialização duplicada está relacionada ao cliente, geralmente está relacionado a um código mal projetado do mesmo, porém a empresa fica responsável por alertar o cliente de tal situação, uma vez que para o mesmo é quase imperceptível o resultado de tal falha.

3.6.4 Preenchimento incorreto do *meta* (MP4)

Cientes que utilizam plataformas de comércio eletrônico possuem soluções prontas, oferecidas pela empresa, de integração (conhecidas por módulos). Esses módulos podem ser customizados para cada cliente (quando necessário) e, caso alguma customização seja mal projetada ou haja alguma atualização da plataforma, o *meta* pode vir a ser preenchido incorretamente. A empresa fica responsável pela correção de tal falha.

Cientes que possuem integração própria podem ficar com dúvidas em relação a documentação ou podem cometer erros ao atualizar a página, de modo que o *meta* acaba preenchido incorretamente. Tais situações precisam ser mapeadas pela empresa para que as correções sejam solicitadas ao cliente. Portanto, trata-se de um processo mais demorado de correção.

3.6.5 Não inicialização do *script* (MP5)

A não inicialização automática do *script* é uma opção do cliente, onde o mesmo assume controle total de quando deseja iniciar o processo. Portanto, a causa de não inicialização está totalmente atrelada a falhas que ocorrem do lado do cliente. Essa situação precisa ser mapeada pela empresa e repassada ao cliente em grande parte dos casos, uma vez que, em sua maioria, os clientes contratam agências para efetuar a integração e o acompanhamento de erros é muito distante.

3.6.6 Falhas no servidor (MP6)

As falhas de servidor são causadas na maioria dos casos por problemas relacionados a má projeção de código e são rapidamente detectadas. Outra causa possível é falha no provedor de serviços utilizado para hospedagem dos servidores, que na maioria dos casos também é rapidamente detectada e corrigida.

3.7 Ocorrência

Assim como a severidade, a ocorrência também é dado por um índice, “que corresponde a um número estimado das falhas que poderiam ocorrer” [7]. Portanto também existe uma tabela de índices equivalentes para uma quantidade de falhas. A tabela 7 apresenta esses índices de acordo com [8]. Como não há mensuração da quantidade de ocorrência, todos os índices foram atribuídos empiricamente aos modos de falha.

Tabela 7 - Índices de ocorrência.

Probabilidade de ocorrência	Chances de ocorrência	Escore
Remota	0	1
Baixa	1/20.000	2
	1/10.000	3
Moderada	1/2.000	4
	1/1.000	5
	1/200	6
Alta	1/100	7
	1/20	8
Muito alta	1/10	9
	1/2	10

Fonte - [8]

A tabela 8 apresenta a distribuição dos índices para cada modo de falha. Como não existe mensuração para determinar as chances de ocorrência, os índices foram atribuídos baseados na probabilidade de ocorrência da tabela 7.

Tabela 8 - Índice de ocorrência para cada modo potencial de falha.

Modo de falha	Probabilidade de ocorrência	Índice
MP1	Baixa	3
MP2	Moderada	5
MP3	Baixa	2
MP4	Moderada	6
MP5	Baixa	2
MP6	Remota	1

3.8 Controles atuais

“Medidas preventivas e de detecção que já tenham sido tomadas e/ou são regularmente utilizadas nos produtos/processos da empresa.” [6]. Portanto qualquer medida que já exista para diminuir os riscos de ocorrência dos modos potenciais de falha é definido como controle.

Como a maioria dos problemas ocorre do lado do cliente, grande parte dos controles devem ser feitos de forma reativa, ou seja, ao ocorrer o problema, precisa-se detectá-lo e em seguida comunicar o cliente do mesmo. A seguir são analisados os controles atuais para cada falha já apresentada.

3.8.1 Remoção de *script* (MP1)

A detecção da remoção de *script* da página do cliente pode se dar de duas formas. A primeira é através de uma ferramenta conhecida como *bot*, que acessa a página do cliente e verifica a presença do *script* na página. Essa ferramenta é pouco utilizada pois envolve um alto grau de customização devido a navegação das páginas dos clientes serem muito diferentes. Portanto, simular um processo de compra para garantir que o *script* está presente em uma página de carrinho pode não ser trivial.

A segunda ferramenta utilizada baseia-se no histórico de dados do cliente. Dado um comportamento existente para um cliente, caso um gráfico apareça zerado, semelhante a figura 5 um alerta é gerado e a verificação é feita manualmente. Um dos problemas desse sistema é a quantidade de falsos positivos detectados.

Portanto existem 2 tipos de controle atualmente utilizados:

- C1.1 → *Bot*
- C1.2 → Histórico do cliente (gráfico zerado)

3.8.2 Erro de *script* (MP2)

Nesse caso, pode ser aplicada a mesma ferramenta que utiliza o histórico de dados do cliente para detectar uma falha. Porém, a mesma funciona apenas para gráficos zerados e não detecta falhas semelhantes a apresentada pela figura 6.

Outra ferramenta existente no processo é uma ferramenta de detecção de falhas no código, conhecida como *lint*. Pelo fato da linguagem não ser compilada, a maioria dos erros são apresentados no momento da execução do código. Esse tipo de ferramenta faz uma pré-verificação de sintaxe e algumas outras regras definidas para evitar problemas simples no código.

Existem ainda dois processos para evitar que erros cometidos pela equipe de desenvolvimento da empresa cheguem no cliente. Um deles é conhecido por *code review*, que é um processo bem conhecido pela indústria de *software*, onde uma segunda pessoa revisa o código produzido por alguém. Dessa maneira, códigos com defeitos podem ser evitados. Além disso, um outro processo é o de *quality assurance*, ou somente QA. Nesse processo, uma segunda pessoa é responsável por testar o código desenvolvido, ou seja, simular o comportamento na página do cliente, e assim avaliar qualquer efeito colateral que esse código possa causar.

Apesar de todos esses processos e ferramentas, códigos com defeito ainda podem ser inseridos nas páginas dos clientes. Foram definidas 4 formas de controle:

- C2.1 → Histórico do cliente
 - 1 → Gráfico zerado
 - 2 → Diminuição da captura de eventos
- C2.2 → *Lint*
- C2.3 → *Code Review*
- C2.4 → QA

3.8.3 Inicialização duplicada (MP3)

As maneiras de controle atuais para detecção de inicialização duplicada são as mesmas apresentadas para MP1 em 3.8.1, porém, neste caso, o histórico do

cliente está relacionado com o aumento do número de eventos e não com gráfico zerado, ou seja:

- C3.1 → *Bot*
- C3.2 → Histórico do cliente (Aumento da captura de eventos)

3.8.4 Preenchimento incorreto do *meta* (MP4)

As maneiras de controle são as mesmas apresentadas em 3.8.1, porém neste caso temos duas situações com o histórico do cliente, de gráfico zerado e diminuição da captura de eventos. Portanto os controles são:

- C4.1 → *Bot*
- C4.2 → Histórico do cliente
 - 1 → Gráfico zerado
 - 2 → Diminuição da captura de eventos

3.8.5 Não inicialização do *script* (MP5)

Para essa falha, os controles são os mesmos apresentados em 3.8.4, portanto:

- C5.1 → *Bot*
- C5.2 → Histórico do cliente
 - 1 → Gráfico zerado
 - 2 → Diminuição da captura de eventos

3.8.6 Falhas no servidor (MP6)

Os controles para falhas no servidores são feitos por sistemas externos de monitoria, sendo assim, este controle não será explorado a fundo no presente trabalho. Portanto:

- C6 → Sistemas externos

3.9 Detecção

A detecção “é a probabilidade de que os sistemas de controle detectem a falha [...] antes que esta atinja os clientes.” [7]. A detecção também é dada por um índice que estima a “habilidade para cada um dos controles identificados [...] detectar a falha antes que ela alcance o cliente.” [7]. A tabela 9 apresenta os índices de detecção de acordo com [8].

Tabela 9 - Índices de detecção.

Probabilidade de não detectar a falha	Probabilidade (%) de um defeito individual alcançar o cliente	Escore
Remota	0 - 5	1
Baixa	6 - 15	2
	16 - 25	3
Moderada	26 - 35	4
	36 - 45	5
	46 - 55	6
Alta	56 - 65	7
	66 - 75	8
Muito alta	76 - 85	9
	86 - 100	10

Fonte - [8]

Como novamente não existem medidas para que os índices sejam dados pela probabilidade percentual de detecção, os valores dos índices são atribuídos empiricamente com base na probabilidade de detecção apresentada na primeira coluna da tabela 9. A tabela 10 explora cada controle apresentado no capítulo 3.8 e seu respectivo índice de detecção.

Tabela 10 - Índice de detecção para cada controle atual.

Controle	Probabilidade de não detectar a falha	Índice
C1.1 (<i>Bot</i>)	Baixa	2
C1.2 (Gráfico zerado)	Baixa	3
C2.1.1 (Gráfico zerado)	Baixa	3
C2.1.2 (Diminuição eventos)	Alta	7
C2.2 (<i>Lint</i>)	Alta	8
C2.3 (<i>Code Review</i>)	Moderada	6
C2.4 (QA)	Baixa	2
C3.1 (<i>Bot</i>)	Baixa	2
C3.2 (Aumento eventos)	Muito alta	10
C4.1 (<i>Bot</i>)	Baixa	2
C4.2.1 (Gráfico zerado)	Baixa	3
C4.2.2(Diminuição eventos)	Alta	7
C5.1 (<i>Bot</i>)	Baixa	2
C5.2.1 (Gráfico zerado)	Baixa	3
C5.2.2 (Diminuição eventos)	Alta	7
C6 (Sistemas externos)	Remota	1

3.10 Número de prioridade de risco (NPR)

O número de prioridade de risco, também conhecido por NPR, é “o índice resultado do produto do índice de ocorrência, pelo de severidade, pelo de detecção. Este valor define a prioridade da falha.” [7]. Sendo assim:

$$\text{NPR} = \text{Índice Severidade} \times \text{Índice Ocorrência} \times \text{Índice Detecção}$$

O cálculo do NPR auxilia na tomada de decisões, definindo onde atuar para reduzir ou eliminar os modos de falha do sistema. Com base no NPR são apontadas ações que podem ser tomadas. Esses pontos serão explorados no capítulo seguinte, com base na tabela 11 que apresenta o NPR dos índices levantados nos capítulos anteriores.

Tabela 11 - Expansão do NPR para as falhas apontadas.

Modo potencial de falha	Severidade	Ocorrência	Detecção	NPR
MP1 (Remoção de script)	E1 = 7	Baixa = 3	C1.1 = 2	42
			C1.2 = 3	63
MP2 (Erro de script)	E1 = 7	Moderada = 5	C2.1.1 = 3	105
			C2.2 = 8	280
			C2.3 = 6	210
			C2.4 = 2	70
	E2 = 4		C2.1.2 = 7	140
			C2.2 = 8	160
			C2.3 = 6	120
C2.4 = 2	40			
MP3 (Inicialização duplicada)	E3 = 2	Baixa = 2	C3.1 = 2	8
			C3.2 = 10	40
MP4 (Preenchimento incorreto do meta)	E1 = 7	Moderada = 6	C4.1 = 2	84
			C4.2.1 = 3	126
	E2 = 4		C4.1 = 2	48
C4.2.2 = 7		168		
MP5 (Não inicialização do script)	E1 = 7	Baixa = 2	C5.1 = 2	28
			C5.2.1 = 3	42
	E2 = 4		C5.1 = 2	16
			C5.2.2 = 7	56
MP6 (Falhas no servidor)	E1 = 7	Remota = 1	C6 = 1	7

Os campos marcados com fundo destacado são os que mais chamam atenção na tabela de NPR (tabela 11). Esse pontos serão explorados no próximo capítulo com ações de melhoria recomendadas.

3.11 Ações recomendadas

As ações recomendadas tem como intuito “diminuir os índices de severidade, ocorrência e detecção.” [7]. Com base na tabela de NPR (tabela 11) são levantados os pontos mais críticos na análise de falhas do processo. As ações que visam melhorar (diminuir) esses índices são estudadas e aplicadas conforme disponibilidade da empresa.

No processo estudado não é possível atuar na severidade do problema, uma vez que a captura de dados é um insumo essencial para empresa. Atuar na ocorrência também não é possível, pois os problemas apresentados fazem parte do processo e vão continuar acontecendo. Desta forma, a maneira de diminuir o impacto de tais falhas é se antecipar o quanto antes a sua ocorrência, e atuar proativamente na detecção dessas falhas, ou seja, nos seus controles.

Dos 9 destaques apresentados na tabela 11, 6 deles são referentes a erros de script (MP2). Todos os 6 estão concentrados nos controles 1, 2 e 3. Os controles 2 e 3 são *Lint* e *Code Review* respectivamente. O *Lint* corresponde a uma ferramenta externa aplicada no código desenvolvido internamente. Uma possível melhoria é uma configuração mais restrita na ferramenta, ou seja, explorar regras que não são aplicadas hoje de modo a evitar problemas recorrentes.

O *Code Review* é um processo interno que pode ser melhorado com a capacitação dos funcionários, uma vez que após um código ser desenvolvido, uma segunda pessoa analisa esse código. Quanto mais capacitada essas pessoas, mais criterioso será esse processo e a chance de um código defeituoso chegar ao cliente será diminuída.

O controle 1 referente a MP2 é o mesmo que o controle 2 de MP4. A atual ferramenta de monitoramento de falhas, detecta 100% dos casos onde há um problema de gráfico zerado. Porém, tal ferramenta raramente detecta casos em que há diminuição na quantidade de eventos. Somado a isso, o controle 3.2 também chama atenção por raramente ser detectado pela ferramenta. Dessa forma, tal condição praticamente nunca é detectada, muito em função da natureza de como a ferramenta trabalha, que será explorada nos próximos capítulos.

Os demais índices obtidos são satisfatórios e alguns já estão em processo de melhorias, como é o caso do *Bot*, que é extremamente eficiente. Mas configurá-lo é custoso, uma vez que, como já mencionado, necessita de uma configuração praticamente exclusiva para cada cliente.

3.11.1 Detecção baseada no histórico do cliente

Na análise do capítulo anterior fica evidente que a ferramenta de detecção de falhas a partir do histórico do cliente é vulnerável. A ferramenta atualmente utilizada é baseada em um projeto *open source* chamado *Skyline*² desenvolvido pela empresa Etsy³. Apesar do projeto *Skyline* oferecer alguns algoritmos nativamente, um algoritmo extra foi desenvolvido e utilizado para analisar as falhas. Tal algoritmo verifica apenas se o tempo no qual não houve registro de eventos (gráfico zerado) é maior que um limiar estabelecido. Sendo assim, a ferramenta só analisa a possibilidade de falha se não receber eventos, caso contrário as falhas não são detectadas. Além disso, esse algoritmo não resulta em uma boa análise para clientes de pequeno porte que não possuem uma quantidade “contínua” de eventos, pois a ferramenta acaba por detectar diversos falsos-positivos nestes casos.

O presente trabalho visa então melhorar o algoritmo utilizado atualmente através de uma análise estatística do comportamento do cliente ao longo do tempo. Ou seja, tentar identificar padrões de tal maneira que seja possível monitorar possíveis falhas nos clientes através de mudanças inesperadas de comportamento. Sendo assim, será possível detectar falhas que antes não eram detectadas e ainda diminuir a quantidade de falsos-positivos uma vez que o comportamento do cliente é levado em consideração.

² Disponível em <<https://github.com/etsy/skyline>>

³ Site da empresa <<https://www.etsy.com/>>

4 Ferramentas para detecção de falhas

As ferramentas aplicadas no problema são puramente estatísticas, onde padrões são obtidos e comparados com a situação atual para detecção de falhas. Além disso, uma análise do problema é necessária para decidir como aplicar tais ferramentas de forma eficiente, sem agravar o problema de detecção.

Foram utilizadas basicamente três ferramentas no algoritmo para analisar o histórico de dados do cliente. São elas: média, média móvel e desvio padrão. A análise do problema e a solução proposta são apresentadas nos capítulos seguintes.

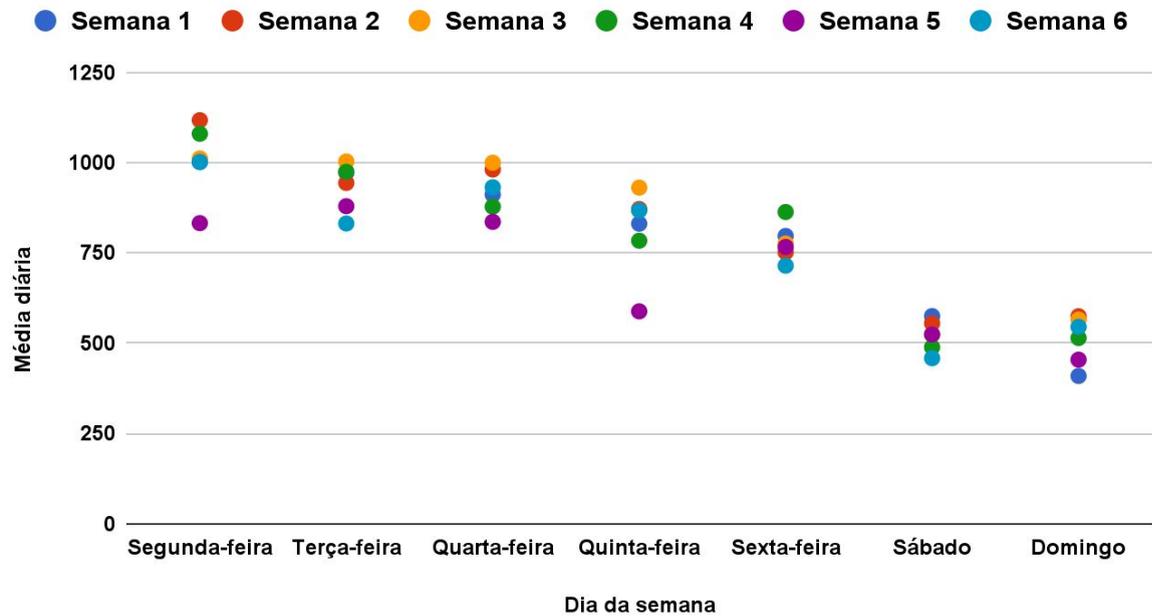
4.1 Análise do problema

Com base nas ferramentas apresentadas, se faz necessário uma análise do problema de modo a definir como essas ferramentas foram aplicadas. Foram realizadas análises de comportamento de acesso dos clientes com o objetivo de se traçar, da melhor forma possível, um padrão de acesso.

Assim como já apresentado na figura 3, o comportamento de acessos varia entre os dias de semana e finais de semana - é o que salienta a figura 8. A figura 8 apresenta a distribuição de acessos (média diária) ao longo de 6 semanas para cada dia da semana. Percebe-se, a partir da figura, que o padrão de acesso é semelhante para os dias da semana no decorrer das semanas anteriores, mas não para a semana como um todo.

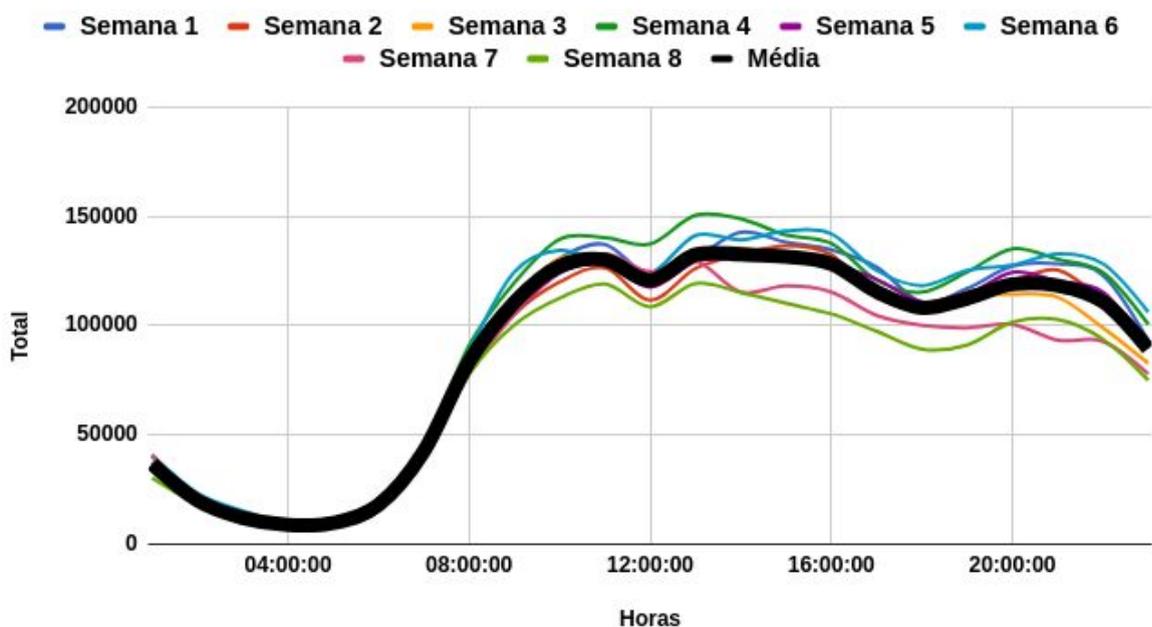
Figura 8 - Visualizações de página por dia da semana.

Visualizações de página



A figura 8 sugere um padrão de comportamento baseado no dia da semana para o cliente. Essa avaliação é reforçada com a figura 9, que explora, para outro cliente, o padrão de acesso durante 8 semanas para apenas um dia da semana, onde fica mais evidente que tal padrão se repete.

Figura 9 - Comportamento de acesso por hora para 8 semanas.

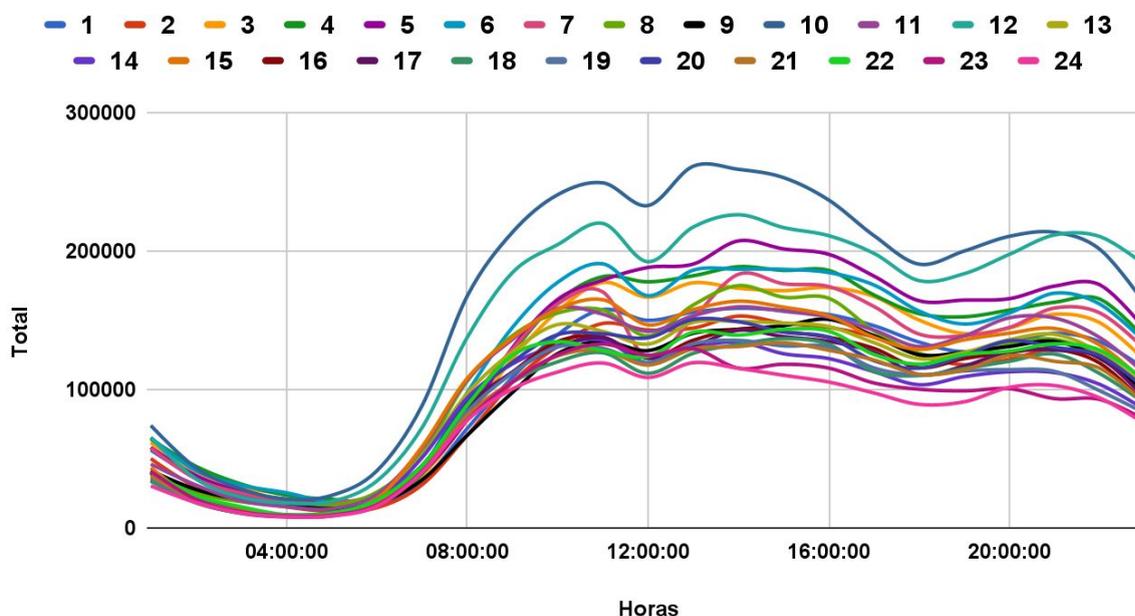
Visualizações de página

Com base nessa análise, podemos usar o histórico do cliente para um dia da semana, ao longo das semanas, para definir seu comportamento. Sendo assim, dado um dia da semana, segunda-feira, basta comparar o comportamento dos eventos gerados nesse dia com as segundas-feiras anteriores.

A figura 10 extrapola os limites da figura 9, considerando 24 semanas de dados para um mesmo dia de semana. Fica evidente que a dispersão dos dados é muito maior do que considerando apenas 8 semanas, que é o caso da figura 9. Isso serve como premissa para determinar o comportamento do cliente, ou seja, podemos utilizar apenas 2 meses de dados para determinar o comportamento do mesmo, eliminando assim uma alta dispersão causada pela sazonalidade.

Figura 10 - Comportamento de acessos por hora para 24 semanas.

Visualizações de página

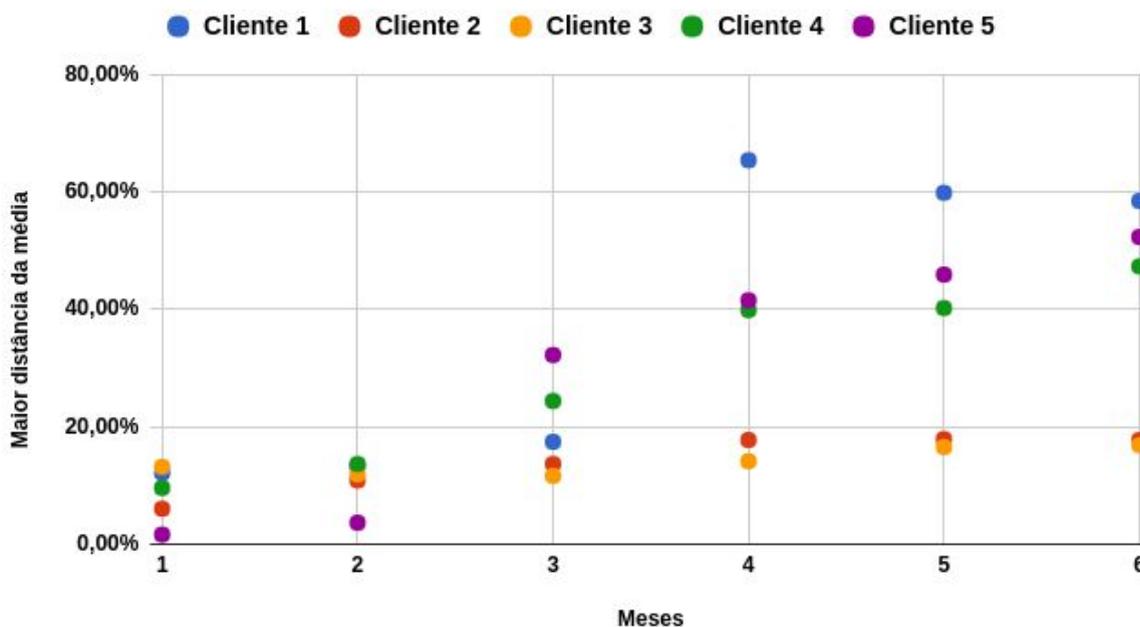


O efeito da sazonalidade é bastante evidente no comércio eletrônico, ou seja, determinadas épocas do ano vendem mais que outras dependendo do tipo de produto comercializado. Por exemplo, lojas que vendem material escolar possuem um aumento no número de acessos e vendas nos primeiros meses do ano. Já lojas de roupas para praia acabam tendo um maior número de acessos próximo e durante o verão.

A figura 11 apresenta uma análise de 5 lojas de nichos diferentes e seus comportamentos nos 6 primeiros meses do ano. O gráfico mostra a maior variação em relação a média para um acumulado de meses de dados para um dia da semana. Por exemplo, para o cliente 1, se considerarmos 5 meses de dados a maior variação em relação a média desse período em visualizações de página chega a 60%.

Figura 11 - Variação em relação a média para 5 clientes nos primeiros 6 meses do ano.

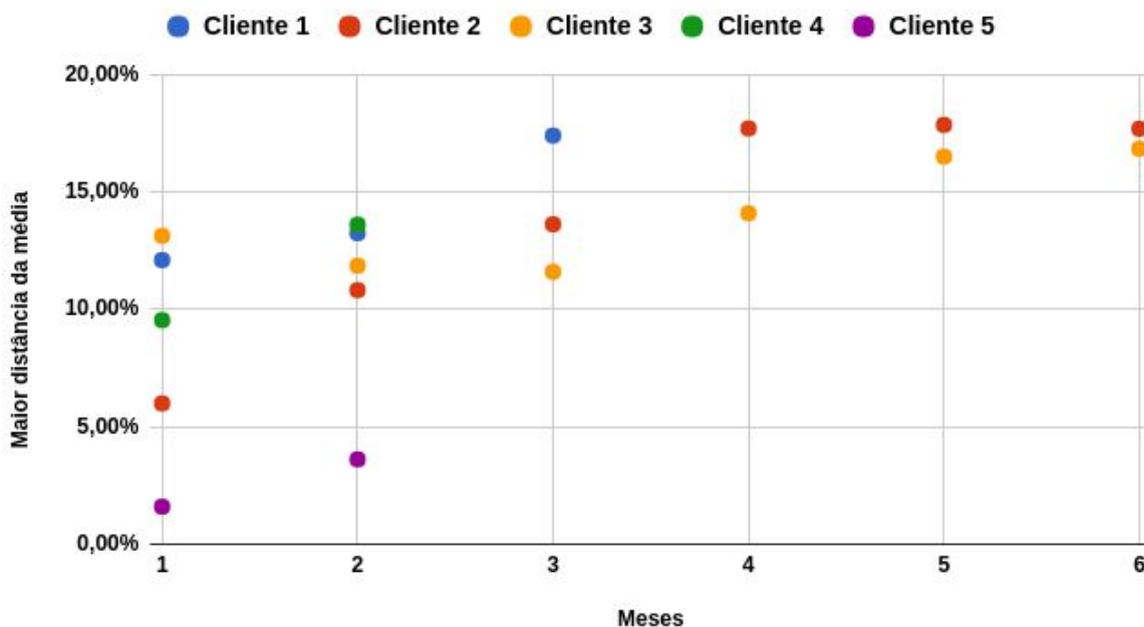
Visualizações de página



A figura 11 complementa o que foi apresentado na figura 10, onde foi apresentado uma dispersão dos dados ao considerar mais de 2 meses. A figura 12 é equivalente a figura 11, porém ampliada para analisar uma faixa mais restrita de valores, até 20% em relação a média. Isso reforça o fato que utilizar muitos dados pode resultar em uma análise equivocada do momento no qual o cliente se encontra.

Figura 12 - Variação em relação a média para 5 clientes nos primeiros 6 meses do ano (Ampliada).

Visualizações de página



Portanto, utilizar 2 meses de dados é o suficiente para caracterizar e analisar um cliente, evitando comportamentos derivados da sazonalidade dos meses anteriores e também uma sensibilidade muito alta no caso de apenas 1 mês de dados, onde qualquer variação terá um impacto muito grande.

4.2 Solução proposta

Com base nas ferramentas propostas e na análise anterior, foi projetada uma nova ferramenta para detectar possíveis falhas na coleta de dados com base no comportamento do cliente. A ferramenta compara o comportamento atual do cliente com seu comportamento passado para inferir sobre possíveis problemas.

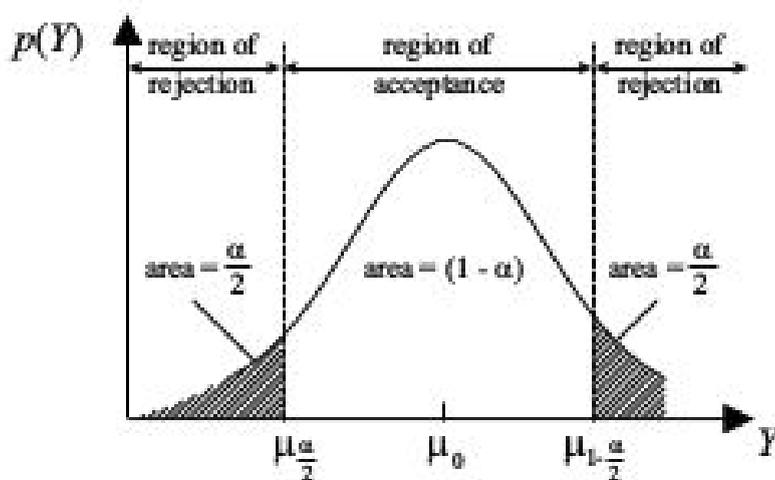
Os eventos coletados são analisados separadamente, ou seja, a mesma análise é aplicada em cada um dos eventos de *PageView*, *ProductView*, *CartView* e *Transaction*. Para cada um deles é analisado o comportamento passado do cliente.

A técnica consiste em analisar os últimos 2 meses de dados coletados e com base na média diária (para apenas um dia da semana) obtém-se a média desse período e o desvio padrão. Uma janela calcula essa média para cada dia

individualmente. Semelhante ao que é mostrado na figura 9, onde é traçada a média das médias dos últimos 2 meses.

O resultado de tal análise é semelhante ao apresentado por [9] na figura 13, onde a partir da média é obtida uma distribuição com uma região de interesse e uma região de rejeição. No presente contexto, na região de rejeição seria detectada uma possível falha e um alerta enviado para os responsáveis.

Figura 13 - Distribuição ao entorno da média com região de rejeição.

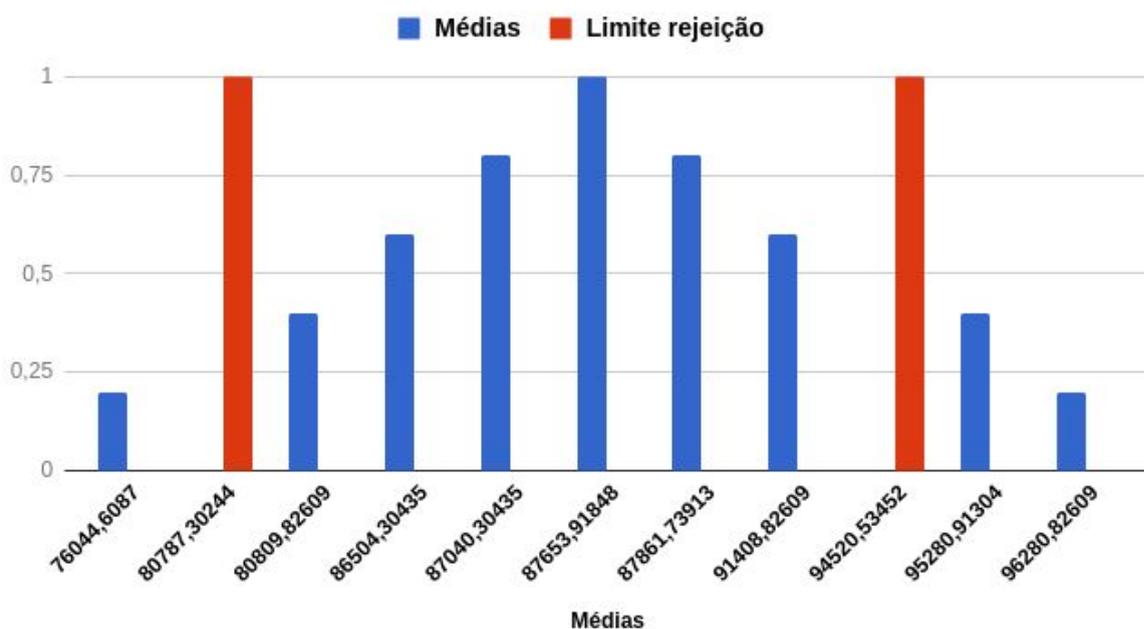


Fonte - [9]

Utilizando-se de dois desvios padrões, de modo a não deixar o sistema muito sensível, a análise aplicada a figura 9 é apresentada na figura 14. São consideradas as médias diárias, uma vez que a ferramenta analisa os dados diariamente.

Figura 14 - Distribuição de médias e limite de dois desvios padrões de rejeição.

Médias de visualização de página



Caso a média calculada esteja entre as duas barras de limite de rejeição, nada precisa ser feito. Caso contrário, um alerta é disparado e a situação analisada individualmente. Tal processo é repetido diariamente e os cálculos são refeitos para os dois meses anteriores.

5 Resultados

Os resultados obtidos são comparados com a ferramenta atual para uma análise mais completa. Foi usada uma base de 20 clientes para validar o algoritmo desenvolvido, baseada nas falhas detectadas pela ferramenta atual e clientes atualmente em situação crítica - se o sistema atual detectou problema, tal cliente foi adicionado na base de testes da nova ferramenta.

Conforme foi apresentado, a ferramenta atual possui uma limitação de detecção para diminuição do número de eventos e para aumento no número de eventos, funcionando bem apenas para gráfico zerado. Além disso, existe uma quantidade de falsos-positivos excessiva detectada por esta ferramenta.

A ferramenta desenvolvida visa atacar esses pontos de forma a detectar outros tipos de falha e ainda diminuir a quantidade de falsos-positivos, uma vez que o comportamento do cliente é considerado - a ferramenta atual não considera aspectos de sazonalidade e comportamento do cliente. A tabela 12 apresenta um comparativo para aplicação da nova ferramenta durante 15 dias (primeiros 15 dias após sua conclusão). Vale ressaltar que a quantidade de alertas é cumulativa, ou seja, caso uma falha ocorra e no dia seguinte ela ainda não tenha sido corrigida, um novo alerta é gerado.

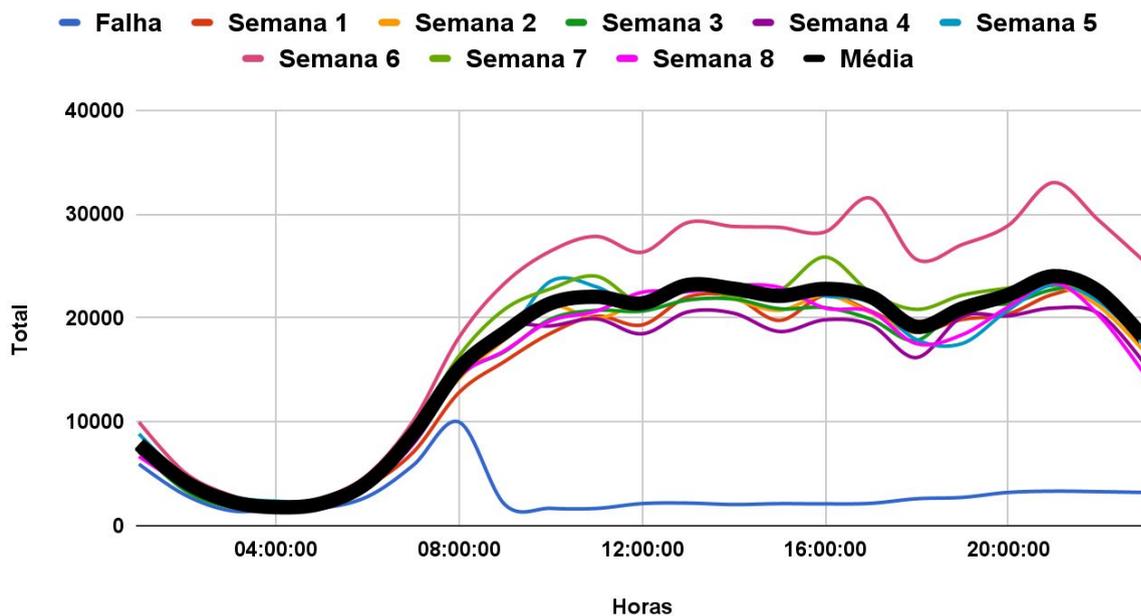
Tabela 12 - Comparação de 15 dias entre ferramentas.

Ferramenta	Alertas reais	Alertas falsos	Alertas totais
Atual (<i>Skyline</i>)	40 (30,8%)	90 (69,2%)	130
Nova	45 (84,9%)	8 (15,1%)	53

Como apresentado na tabela 12, a atual ferramenta possui uma quantidade de falsos-positivos maior que alertas reais, o que foi melhorado com a nova ferramenta. A ferramenta nova também detectou falhas que a ferramenta antiga não detectou no mesmo período. A figura 15 apresenta um tipo de falha não capturado pela ferramenta atual, onde há uma queda na quantidade de eventos coletados devido a uma falha de integração com o cliente.

Figura 15 - Exemplo de falha com queda na captura de eventos.

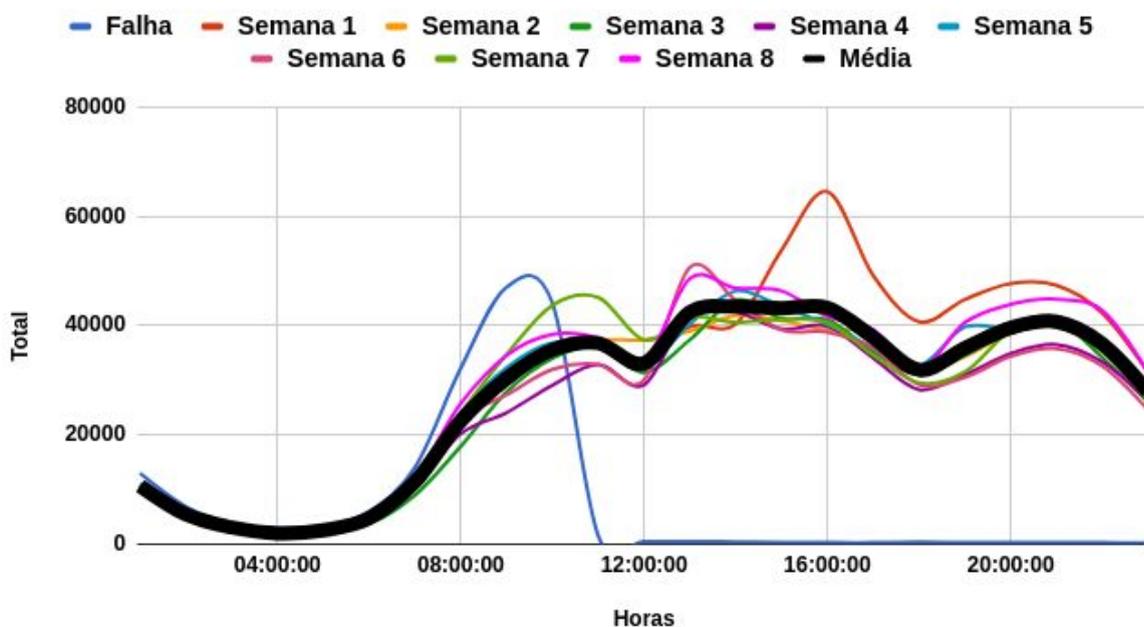
Visualizações de página



A figura 16 apresenta um alerta detectado pelas duas ferramentas, ou seja, uma queda abrupta no gráfico, que acaba zerando a captura de eventos. Nesse caso específico, foi removido o *script* de integração por parte do cliente.

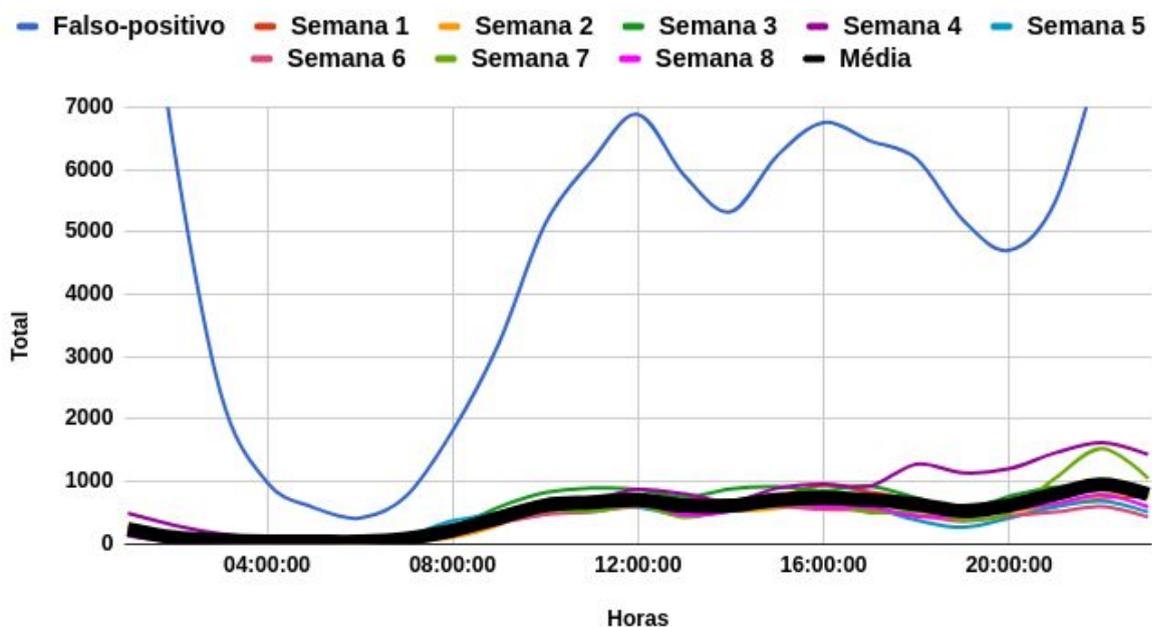
Figura 16 - Falha detectada pelas duas ferramentas.

Visualizações de página



Apesar dos resultados positivos, ainda foram detectados falsos-positivos, que são pontos críticos para ferramentas de detecção de falhas (colocam em xeque a credibilidade dos alertas). A figura 17 apresenta um falso-positivo detectado. A detecção se deu pelo fato de um aumento na captura de eventos, porém a razão para tal comportamento foi uma promoção, semelhante a *Black Friday*, feita pelo cliente.

Figura 17 - Falso-positivo detectado por promoção na página do cliente.

Visualizações de página

Mesmo com a limitação em alguns pontos da ferramenta, como mostrado na figura 17, a avaliação geral da mesma é positiva, visto que problemas antes não detectados passaram a ser detectados e a quantidade de falsos-positivos diminuiu consideravelmente.

6 Conclusão

O presente trabalho fez análise de um processo de captura de eventos de páginas de clientes da empresa Linx. Os eventos coletados servem como insumos para as soluções oferecidas pela empresa (personalização de comércio eletrônico).

Os eventos são gerados para cada página visitada por um usuário, sendo que no caso de páginas de produto, carrinho e confirmação de compra, há geração de eventos extras, que são analisados separadamente, dada a importância dos insumos dessas páginas.

Pôde-se perceber que diversos problemas podem gerar falhas no processo de captura desses eventos. Assim, se faz necessário detectar tais falhas logo que possível, evitando um comprometimento das soluções contratadas pelo cliente.

Para analisar possíveis falhas nesse processo, foi aplicada a metodologia FMEA (Análise dos Modos de Falha e seus Efeitos), expondo as vulnerabilidades do processo. Com base no resultado do FMEA, ficou evidente que um dos principais pontos de detecção de falhas na captura de eventos é a análise dos dados do cliente.

Com o intuito de melhorar a análise dos dados dos clientes foi desenvolvida uma ferramenta de análise estatística que usa o histórico de dados dos clientes como insumo de sua análise. A ferramenta utiliza os dados diários dos clientes para inferir seu comportamento baseado no dia da semana.

Os resultados obtidos com o uso da nova ferramenta foram comparados com os resultados obtidos pela ferramenta antiga, para o mesmo período. A comparação mostrou uma melhora significativa na detecção das possíveis falhas e uma redução na quantidade de alertas falsos gerados.

Apesar da ferramenta proposta apresentar um resultado melhor do que a ferramenta atual, a mesma apresenta pontos de falha que geram alertas falsos. Um dos principais alertas falsos detectados pela nova ferramenta são promoções feitas pelo cliente, que acabam gerando um volume maior de visitas e, conseqüentemente, de eventos.

Uma possível melhoria para a ferramenta desenvolvida é uma análise próxima ao tempo real, de modo que as falhas sejam detectadas o quanto antes (as mais críticas, como remoção de script, por exemplo) e não somente após um dia. Tal análise pode usar o comportamento do cliente ao longo do dia para inferir sobre possíveis problemas, além disso considerar variações mais abruptas na coleta, evitando uma sensibilidade muito alta.

O cruzamento de informações coletadas por outros processos (*Google Analytics*⁴, por exemplo) também pode auxiliar no processo de detecção de falhas, uma vez que duas fontes de dados para o mesmo cliente devem apresentar o mesmo comportamento. Assim, caso haja uma divergência entre as duas fontes, um alerta deve ser gerado e o problema analisado.

⁴ Disponível em <<https://www.google.com/intl/pt/analytics/>>

Referências

- [1] Chaordic, Tagueando as páginas: Todas as páginas. Disponível em: <<http://docs.chaordic.com.br/docs/tagging-all>>. Acesso em 18 de junho de 2017.
- [2] Chaordic, Tagueando as páginas: Página de produto. Disponível em: <<http://docs.chaordic.com.br/docs/tagging-product>>. Acesso em 18 de junho de 2017.
- [3] Chaordic, Tagueando as páginas: Página de produto. Disponível em: <<http://docs.chaordic.com.br/v5/docs/tagging-product>>. Acesso em 18 de junho de 2017.
- [4] Chaordic, Tagueando as páginas: Página de carrinho. Disponível em: <<http://docs.chaordic.com.br/docs/tagging-cart>>. Acesso em 18 de junho de 2017.
- [5] Chaordic, Tagueando as páginas: Página de confirmação de compra. Disponível em: <<http://docs.chaordic.com.br/docs/tagging-confirmation>>. Acesso em 18 de junho de 2017.
- [6] TOLEDO, José; AMARAL, Daniel. **FMEA: Análise do tipo e efeito de falha**. São Paulo. Grupo de estudos e pesquisa em qualidade - UFSCar. 12p.
- [7] SAKURADA, Eduardo. **As técnicas de Análise do Modos de Falhas e seus Efeitos e Análise da Árvore de Falhas no desenvolvimento e na avaliação de produtos**. 2001. 145 folhas. Dissertação (Mestrado em Engenharia Mecânica) - Universidade Federal de Santa Catarina, Florianópolis.
- [8] BEN-DAYA, M.; RAOUF, A. **A revised failure mode and effect analysis model**. International Journal of Quality & Reliability Management, v. 13, n. 1, p. 43-47, 1996.
- [9] ISERMANN, Rolf. **Fault-diagnosis systems: an introduction from fault detection to fault tolerance**. Springer Science & Business Media, 2006.

ANEXO A - Tabela FMEA Completa

A tabela 13, abaixo, é uma representação completa do quadro FMEA preenchido para o caso de visualizações de páginas. O mesmo quadro pode ser replicado para páginas de produto, carrinho e confirmação de compra. A mesma é adaptada a partir da tabela exemplo apresentada por [7]. As colunas de ações recomendadas, responsabilidade e resultados não são exploradas.

Tabela 13 - Quadro FMEA completo para PageView.

Componente	Função	Modo potencial de falha	Efeito	Severidade	Causas	Ocorrência	Controles	Deteção	NPR			
Captura de eventos	Visualização de páginas (PageView)	Remoção de script	Sem captura de eventos	7	Atualização de conteúdo/ <i>lyout</i>	3	Bot	2	42			
										Histórico de dados	3	63
		Lint	8	280								
					Code Review	6	210					
		QA	2	70								
					Histórico de dados	7	140					
		Lint	8	160								
					Code Review	6	120					
		QA	2	40								
Bot	2				8							
		Histórico do	10	40								

