

Sistema para Avaliação de Indicadores voltado para Times de Desenvolvimento de Produtos de Software

*Relatório submetido à Universidade Federal de Santa Catarina
como requisito para a aprovação na disciplina
DAS 5511: Projeto de Fim de Curso*

Tiago Nunes Resmini

Florianópolis, setembro de 2016

Sistema para Avaliação de Indicadores voltado para Times de Desenvolvimento de Produtos de Software

Tiago Nunes Resmini

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. Leandro Buss Becker

Assinatura do Orientador

Banca Examinadora:

Diego Pereira
Orientador na Empresa

Prof. Leandro Buss Becker
Orientador no Curso

Rodrigo Castelan Carlsson
Avaliador

Antonio Adalberto Duarte Junior
Filipe Odozynksi da Silva
Debatedores

Agradecimentos

Agradeço primeiramente aos mais próximos: família e namorada. Beto, Roseli e Filipe, que me auxiliaram e me apoiaram de toda forma imaginável não somente durante todo o período de minha graduação mas também durante todo o período que antecede esta fase, que me preparou para os desafios que enfrentaria numa universidade, cursando uma engenharia; e Roberta que acompanhou de perto e vivenciou comigo todos os desafios enfrentados em 6 anos de curso.

Também gostaria de agradecer aos meus colegas de classe, pela amizade construída, que sempre ultrapassou a sala de aula, e pelos dias, noites e madrugadas de estudo que passamos juntos nas disciplinas mais complicadas, sempre compartilhando as dores e aprendizados trazidos pela graduação.

Devo minha gratidão também ao NEO Empresarial, grupo de estágio do qual fiz parte durante 3 anos da minha graduação, e que contribuiu em grande parte da minha formação técnica e pessoal que adquiri. Tenho certeza de que todos os aprendizados que obtive durante estes anos serão de grande valor ao longo de toda a minha vida profissional.

Por fim, mas não menos importante, agradeço à Resultados Digitais, principalmente ao meu orientador e mentor na empresa, Diego, pela oportunidade que recebi de trabalhar e ajudar uma empresa tão grandiosa a evoluir e pelo aprendizado que continuo obtendo dia após dia na empresa.

Resumo

A gestão de produtos de *Software as a Service* (SaaS) depende muito de dados para apoiar as decisões que são tomadas para definir o produto em questão, desde decisões de curto prazo como qual será a próxima funcionalidade desenvolvida, até decisões de longo prazo como a visão de futuro do produto. À medida que o número de usuários cresce, fica cada vez mais difícil entrevistar uma amostra significativa deles para encontrar suas maiores dificuldades e problemas e a intuição de um gerente de produto precisará cada vez mais de dados para se provar correta. Isso torna a acessibilidade a dados vital para a sobrevivência do produto no mercado: sem os dados certos, o gerente de produto tem menos conhecimento para tomar a decisão correta e pode direcionar o produto a rumos divergentes daqueles que os usuários necessitam.

Não apenas a acessibilidade, mas também a facilidade com que um gerente de produto analisa um dado influencia diretamente na quantidade de embasamento que ele terá para tomar suas decisões. Porém, ao mesmo passo que estes dados se tornam cada vez mais importantes quando um produto cresce, a infraestrutura de dados e serviços cresce e torna mais difícil acessar estes dados.

Este é um dos problemas que a empresa Resultados Digitais enfrenta em seu estágio atual de produto. Diante disso, este trabalho de PFC trata do estudo e aprofundamento da organização de dados do produto da Resultados Digitais, em consequência das dificuldades enfrentadas pelos gerentes de produto da empresa causada por falta de acesso a dados e traz uma proposta de solução para o problema, juntamente com a implementação da prova de conceito para alguns casos de uso.

Abstract

Software as a Service (SaaS) product management depends a lot on data to support decisions that change the short and long-term vision and goals of the product. As the number of users grow, it gets harder for a product manager to get the users' pain points by interviewing them, and intuition will need more and more data to prove itself right. This makes data accessibility vital to the survival of the product in the market: without data, product managers have less knowledge of users to support his or her decisions, which can drive the product to a divergent road from the one users actually need.

Not only accessibility but also readiness of data for a product manager to analyze influences directly on how much knowledge he has to make right decisions. Even so, as data becomes more and more important following the product growth, the architecture of data and services becomes larger and tighter, making it harder to access data.

This is one of the problems Resultados Digitais is facing in the current state of its product and this work brings a deep understanding of the problem on the sight of the company's product management area, along with a proposal of solution for the problem and its proof of concept.

Sumário:

Agradecimentos.....	4
Resumo	5
Abstract	6
Sumário:	7
Simbologia:.....	8
Capítulo 1: Introdução	9
Capítulo 2: Empresa Resultados Digitais	12
Capítulo 3: Definição do Problema e Especificação da Solução	18
Capítulo 4: Desenvolvimento da Nova Base de Dados	27
Capítulo 5: Desenvolvimento de Relatórios de Suporte	37
Capítulo 6: Resultados	46
Capítulo 7: Conclusões e Perspectivas	50
Bibliografia:.....	52

Simbologia:

BI – Business Intelligence

CPC – Cost per Click

DW – Data Warehouse

EMR – Elastic Map Reduce

ETL – Extract, Transform and Load

KPI – Key Performance Indicator

LP – Landing Page

PM – Product Manager

SaaS– Software as a Service

SQL – Strutuctured Query Language

SSOT – Single Source of Truth

UI – User Interface

UX – User Experience

Capítulo 1: Introdução

A evolução da tecnologia digital tanto em dispositivos móveis quanto em serviços de *software* disponíveis via internet fizeram com que a maneira que se desenvolve produtos dentro da nossa sociedade mudasse. Cada vez mais vemos produtos digitais sendo inseridos no mercado ou produtos não digitais sendo digitalizados.

Um dos grandes avanços conceituais que a internet trouxe foi a ideia de que um indivíduo não precisa possuir todos os recursos necessários para executar uma ação (ou serviço), mas que podem existir organizações que centralizam recursos para uma grande população e dividem eles por métodos mais simples e baratos [5]. Uber, Amazon Web Services e Spotify são exemplos de grandes empresas da atualidade que utilizam essa abordagem de centralização e divisão de recursos para fornecer serviços relativamente baratos e de grande qualidade à sociedade.

Desta forma de gestão de produtos e serviços surgiu o modelo SaaS (*Software as a Service*) [5], no qual o consumidor paga mensalmente num modelo de assinatura por um serviço, o qual é apoiado por um produto digital (*software*) e uma equipe de gestão e suporte do mesmo. Este modelo, junto à expansão da internet, trouxe uma grande diferenciação na maneira com que os produtos de *software* são desenvolvidos: antes as soluções eram únicas, construídas de maneira específica para a necessidade do cliente, resultando em altos preços quanto maior fosse a solução e sua customização. Porém, com a fácil distribuição que a internet permite, hoje é possível desenvolver apenas um produto (com algumas customizações, ou “planos”) e replicar a venda do mesmo a diversos consumidores, tornando os preços muito mais acessíveis, principalmente em modelos SaaS, onde o cliente paga pelo produto e serviço mensalmente, em vez de ter um preço estabelecido por um pacote, o qual depois de pago enfraquece o poder de barganha do consumidor com a empresa.

Como a forma de desenvolver produtos digitais mudou, a gestão destes produtos também está acompanhando estas mudanças [1]. O desenvolvimento de uma solução única de *software* se assemelha muito à gestão de projetos com o

método cascata, onde os requisitos são extensivamente especificados para que não haja divergência de solução, um plano de execução é traçado e o gerente do projeto se compromete a uma entrega, num prazo, e gerencia sua equipe e tempo para garantir que prazo, escopo, custo e qualidade sejam entregues conforme acordado [4].

Porém, em soluções de *software* que são quase únicas para milhares de clientes, surge um papel muito importante para que seja possível inovar e melhorar o produto constantemente: o Gerente de Produto (*Product Manager - PM*). O papel do PM é ser o *brand man* do produto [1]: entender seus clientes, entender seu produto e seus concorrentes, entender o mercado no qual está inserido e entender o negócio da sua empresa para que consiga tomar as decisões corretas na hora de priorizar o que será desenvolvido no produto.

Para desenvolver o produto certo (objetivo final do PM), ele precisa responder diversas perguntas. Alguns exemplos são: O que os clientes precisam no *software*? Qual problema eles enfrentam atualmente? Será que eles gostaram dessa funcionalidade nova? Qual funcionalidade eles usam mais? Quais usam menos? Quais acham difícil utilizar? Quão complexo seria para implementar esta melhoria? Para quais concorrentes se está perdendo clientes e por quê? O que o CEO espera do produto? [6].

Muitas destas não são simples de responder, mas em produtos digitais existe a vantagem de se poder coletar informações dos usuários muito facilmente, apenas instrumentando as interações que os mesmos têm com o *software*. Por exemplo: para saber qual a funcionalidade mais utilizada, basta eu monitorar dentro do meu serviço de *software* quantas pessoas acessaram aquela parte do produto. E como este exemplo, existe uma grande variedade de fontes de dados que um PM pode (e deve) consultar para que sempre tenha o máximo de informações para tomar as decisões corretas.

Esta grande variedade traz consigo um problema: o tempo para tomar uma decisão muitas vezes é limitado para que seja possível extrair todas as informações a respeito dele, portanto é muito importante saber exatamente quais são os dados mais importantes que devem ser analisados, que darão mais certeza de uma hipótese ou comportamento. Exemplo: para saber se a funcionalidade de álbuns de

fotos do Facebook é fácil de utilizar, o PM deve monitorar se as pessoas clicam no ícone de ajuda, se elas clicam para criar um álbum e desistem no caminho, ou se existem muitas pessoas enviando dúvidas de como criar um álbum? São três opções diferentes de analisar o mesmo problema, e podem existir muitas outras para as quais não haverá tempo para analisar. Porém como saber qual informação é mais valiosa para a tomada de decisão?

Outra dificuldade é que muitas vezes essas informações valiosas não são acessíveis ao PM, o qual depende muitas vezes de habilidades e tempo que não possui disponível para que consiga uma informação relevante. Para que suas decisões sejam sempre baseadas em dados, é contraditório colocar diversas barreiras entre uma informação relevante e a pessoa que precisa acessá-la [7]. Porém, como muitas informações estão disponíveis nas mesmas fontes de dados que o *software* utiliza, esta situação ocorre com frequência, fazendo com que o PM demore ou até mesmo não consiga obter uma informação que precisa para realizar seu trabalho de maneira adequada.

Na empresa Resultados Digitais não é diferente. A empresa oferece um SaaS e conta, no momento da escrita deste documento, com mais de 10 times de desenvolvimento de produto e mais de 4000 clientes. São 10 *Product Managers* buscando informações de bases de dados de mais de 4000 usuários, todos os dias, com uma certa frequência. Deste cenário, surgem várias complicações relacionadas à tomada de decisão para chegar ao principal objetivo da gestão de produto: construir o produto certo, para as pessoas certas, de acordo com o mercado em que está inserido.

Dado o cenário da empresa, objetivo deste trabalho é mapear e aprofundar o entendimento dos problemas relacionados às métricas que direcionam a tomada de decisão da gestão de produto realizada por estes 10 *Product Managers*, assim como propor soluções para os mesmos.

Capítulo 2: Empresa Resultados Digitais

A empresa Resultados Digitais (<http://resultadosdigitais.com.br/>) é uma *startup* de Florianópolis fundada em 2011 por cinco ex-alunos da Universidade Federal de Santa Catarina, dois deles do curso de Engenharia de Controle e Automação. Hoje, quase 5 anos depois de sua fundação, conta com cerca de 350 colaboradores e 4500 clientes.

A empresa proporciona uma solução SaaS para Marketing Digital, uma demanda que vêm crescendo muito nos últimos anos devido ao crescimento da internet, das redes sociais e da expansão dos métodos de captação e análise de clientes via web. A própria metodologia aplicada aos clientes desde o momento que assinam a solução da Resultados Digitais é a mesma que a empresa sempre aplicou e aplica até hoje, focada em *Inbound Marketing*.

2.1: Inbound Marketing

O *Inbound Marketing* é uma forma divergente à forma clássica de fazer Marketing que é chamada de *Outbound Marketing*. Nela, em vez de oferecer diretamente um produto ou serviço a uma pessoa (como em propagandas de televisão e *outdoors*, por exemplo), a empresa deve conquistar o interesse das pessoas em seu produto por meio da divulgação de conteúdo relacionado ao seu negócio, pedindo a permissão de se comunicar com estas pessoas a fim de criar um relacionamento duradouro em que eventualmente elas se tornam clientes [9].

A metodologia, popularizada em 2009 com o lançamento do livro [8] “Inbound Marketing: seja encontrado usando o Google, a mídia social e os blogs”, de Brian Halligan e Dharmesh Shah, se baseia em 4 ações principais: Atrair, Converter, Vender e Encantar. Aliada ao Marketing Digital, a metodologia é potencializada pelas ferramentas de divulgação que hoje são disponíveis: email, websites, ferramentas de busca, blogs e muitas outras. A Figura 1 ilustra um esquema do funcionamento e algumas ferramentas do *Inbound Marketing*.

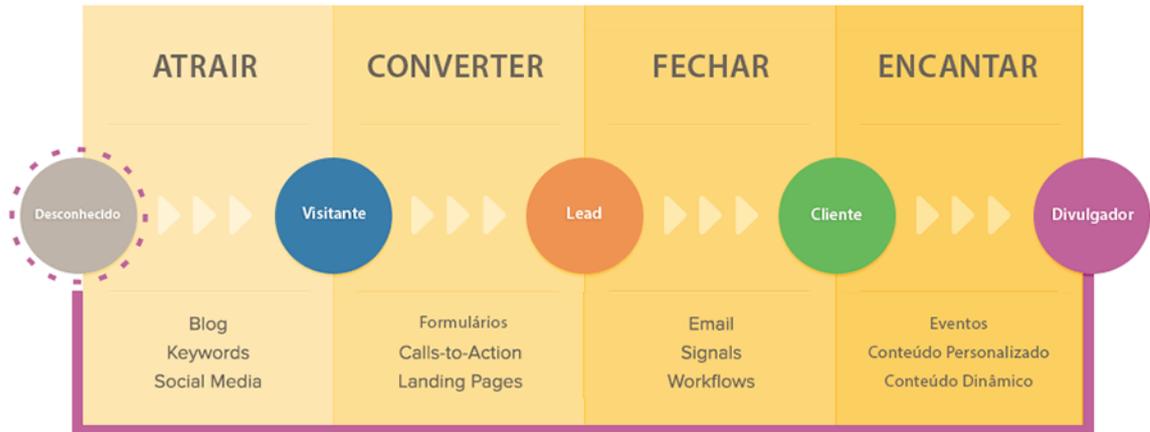


Figura 1 - Esquema demonstrativo do Inbound Marketing

Outra grande vantagem de utilizar o Marketing como aliado é a disponibilidade de informações para tomada de decisão. Há uma frase muito conhecida no Marketing: “Metade do dinheiro que gasto com propagandas é desperdiçado, o problema é que não sei qual metade!” [12]. Este é um problema clássico do Marketing tradicional: não se sabe o quanto uma campanha, um anúncio, uma propaganda traz de fato de retorno em vendas, apenas são feitas apostas de campanhas diferentes e observa-se o resultado. Já no Marketing Digital, todas as métricas de retorno de um anúncio são facilmente extraídas e revisadas, facilitando tanto a tomada de decisão do anunciante como do meio de divulgação (o Google, por exemplo, revisa frequentemente o preço para anunciar uma palavra-chave em sua ferramenta de busca, de acordo com o volume de buscas daquela palavra).

Desta forma, até hoje, a Resultados Digitais adquire seus clientes: produzindo materiais educativos sobre Marketing Digital e *Inbound Marketing*, divulgando estes por email, website, blog, redes sociais e anúncios pagos, captando *Leads* (pessoas interessadas, que fornecem suas informações em troca do conteúdo), relacionando-se com estes Leads para conhecê-los melhor e posteriormente vendendo para estes Leads.

Porém, à medida que o volume de pessoas interessadas num negócio aumenta, fica cada vez mais difícil de gerenciar todas as ações de marketing. A Resultados Digitais, por exemplo, recebe (por meio de formulários em páginas da internet) mais de 30 mil Leads em um mês. Emails precisam ser disparados para

mais milhares de contatos [10]. A empresa precisa saber de forma automática quais destes 30 mil contatos são pessoas com potencial de compra, sem precisar pesquisar em todas as suas campanhas o que sabe sobre cada um destes contatos. As análises precisam ser feitas num lugar só para fazer sentido, do contrário os dados serão duplicados ou inconsistentes.

Pensando nisso é que a Resultados Digitais criou seu SaaS: uma plataforma de Marketing Digital onde é possível realizar todas as ações de marketing que o cliente precisa e que centraliza todas as informações para facilitar o gerenciamento e aumentar a produtividade do usuário.

2.2: RD Station

Como qualquer SaaS, o negócio é focado no seu produto. E no caso da Resultados Digitais, esse produto é um software web (acessível em www.rdstation.com.br).

Aqui será feita uma breve descrição do software, que visa apenas esclarecer alguns pontos chave para o entendimento do problema referente a este projeto, o qual será descrito no capítulo a seguir, e reforçar a importância e valor do mesmo dentro do negócio da Resultados Digitais.

O RD Station traz uma solução completa em marketing digital, onde o usuário tem ferramentas para criar, monitorar, analisar e disparar todo tipo de ação relacionada ao *Inbound Marketing*. O próprio menu do software é baseado nas 4 ações principais do método: Atrair, Converter, Relacionar e Analisar. À seguir, descreve-se todas as suas funcionalidades no momento em que este documento foi escrito (Julho de 2016).

No menu Atrair, o usuário pode publicar postagens em mídias sociais (Facebook, LinkedIn, Twitter), as quais ele conecta previamente com o software, e tem uma visualização centralizada de todas as suas postagens em cada uma das redes sociais. Ainda neste menu ele pode também monitorar termos (ou palavras-chave) buscados na ferramenta de buscas do Google e ter uma estimativa do volume de buscas daquela palavra-chave, do custo por clique (CPC) médio na ferramenta de anúncios do Google (AdWords, na qual é possível pagar para aparecer entre os primeiros resultados de uma busca), da concorrência (número de anunciantes para aquela palavra-chave) e qual o ranking de seu próprio website em

relação àquele termo, ou seja, em qual posição o website aparece quando alguém pesquisa aquele termo.

No menu Converter, é possível criar componentes web com campos de pergunta que visitantes de um site podem preencher para deixar suas informações ao dono do site (neste caso, o usuário do RD Station). Para isso, existem duas ferramentas, uma de criação de *Landing Pages* e uma de criação de formulários.

Landing Pages são páginas “isoladas” do website que tem como objetivo único descrever uma oferta ao visitante para que o mesmo realize uma ação, seja ela clicar em um botão ou preencher um formulário, o que normalmente tem valor para a organização.

A ferramenta de criação de *Landing Pages* do RD Station permite que, o usuário crie uma página específica como descrito acima, sem precisar escrever códigos web, alterando via interface apenas o conteúdo interno (textos, cores, imagens, componentes de vídeo, formulários) de alguns modelos presentes na ferramenta.

A ferramenta de criação de formulário é parecida, porém no lugar de criar uma página da web e publicá-la (deixa-la disponível em um endereço eletrônico), por meio de uma interface parecida, o usuário cria apenas o formulário de uma página e a ferramenta gera um código que pode ser incorporado em uma outra página.

À medida que os visitantes do website do usuário do RD Station preenchem estes formulários (tanto os presentes nas *Landing Pages* quanto os incorporados em outras páginas), eles se tornam *Leads* ou contatos do usuário que podem ser utilizados nas ferramentas do menu Relacionar. Neste menu, é possível visualizar, importar, editar, classificar e excluir *Leads* da lista do usuário, assim como enviar e-mails personalizados a estes contatos de forma automatizada.

As ferramentas de gerenciamento de *Leads* permitem que o usuário possa visualizar cada um dos contatos, suas informações e todas as interações com seu negócio, como preenchimento de algum formulário, recebimento e abertura de e-mails e visita à alguma página do site, por exemplo. Além disso, é possível classificar e segmentar estes *Leads* em diferentes grupos, facilitando o gerenciamento de uma base de contatos muito grande e permitindo que as ações

de relacionamento sejam bem segmentadas, como enviar ofertas de desconto para contatos que já estão próximos de um momento de compra e materiais educativos para contatos que descobriram há pouco tempo o negócio do usuário.

A ferramenta de e-mails conta com dois tipos de disparos; um deles é feito pelo próprio usuário, que pode, assim como nas *Landing Pages*, criar um email numa estrutura (e não simplesmente em texto simples como os editores comuns de email) apenas via interface, editando as informações e de um modelo presente na ferramenta e disparar este e-mail para grupos selecionados de contatos da sua base, incluindo variáveis no corpo do email, como o nome da pessoa que está recebendo por exemplo.

A outra forma é por meio da ferramenta de Automação de Marketing, a qual permite a criação de um fluxo lógico de condições e ações dentro do software. Por exemplo, é possível configurar um email de boas-vindas para quando um contato entra na base (preenche um formulário pela primeira vez). Além disso, a ferramenta de Automação permite também que tarefas de classificação dos *Leads* sejam automatizadas. Por exemplo, quando um contato preenche um formulário de pedido de orçamento, o usuário deseja que este contato seja classificado como uma Oportunidade de negócio, assim, ele não precisará manualmente classificar cada um dos contatos que preencheram tal formulário.

Por fim, o menu Analisar traz diferentes tipos de relatórios que podem ser visualizados pelo usuário. O primeiro exhibe o alcance digital do usuário, somando sua base de e-mails e seus seguidores em redes sociais. O segundo traz um relatório comparativo entre períodos de tempo (dias, semanas, meses) dos visitantes, *Leads* e clientes do usuário, assim como a distinção entre Canais de Aquisição (a forma com que estes visitantes, *Leads* e cliente encontram o site do cliente, dividido por exemplo entre ferramentas de busca, referências de outros sites, mídias sociais e e-mails). O terceiro relatório mostra quais páginas do site do cliente são mais acessadas, em ordem decrescente, juntamente ao tempo médio de um visitante na página e à Taxa de Rejeição da página (porcentagem das visitas que começam e terminam naquela página, em vez de se direcionarem para uma outra página antes de sair do site).

Com isso, o RD Station permite que todo o fluxo do *Inbound Marketing* seja construído em cima dele, facilitando ao usuário o gerenciamento de todas as ações de Marketing por meio da centralização de informações e integração de dados entre várias ferramentas, como formulários e email, por exemplo.

Capítulo 3: Definição do Problema e Especificação da Solução

Para gerenciar um software de tal tamanho (mais de 20 funcionalidades e mais de 4000 clientes), a Resultados Digitais vem expandindo a arquitetura da nuvem e seu time de desenvolvedores, analistas de qualidade, designers e gerentes de produto. Hoje, cada grupo de funcionalidades do produto é responsabilidade de um time diferente, totalizando 12 times de produto quando somamos os times de operação, performance e sistemas internos.

Assim, com uma estrutura tanto de software quanto de pessoas crescendo muito rapidamente, a empresa precisa de um sistema de métricas e indicadores-chave de desempenho (*Key Performance Indicators* - KPIs) muito bem definido para que os times tenham como embasar suas decisões de produto em dados à medida que o número de usuários cresce e fica cada vez mais difícil identificar as demandas mais críticas de uma funcionalidade (muito comumente identificadas por meio de entrevistas, quando ainda existem poucos usuários no software), além da gestão da empresa conseguir acompanhar e monitorar o desempenho dos times.

Por isso, após longos estudos e discussões, a empresa definiu três KPIs que os times de produto deveriam monitorar em suas funcionalidades, baseados na metodologia HEART (*Happiness, Engagement, Adoption, Retention, Task Success*), desenvolvido pela Google [14]:

- Adoção: o percentual de usuários que adotam o uso da funcionalidade (utilizam ela por completo pela primeira vez);
- Retenção: o percentual de usuários que utilizam a funcionalidade de forma recorrente, no período ideal de uso;
- Sucesso do cliente: o quanto a funcionalidade está de fato ajudando (dentro de seu escopo de atuação) os usuários a obterem os resultados esperados como clientes.

Porém, infelizmente, não há como definir uma regra padronizada de como devem ser metrificados estes KPIs de maneira exata. Por exemplo: a recorrência

ideal de posts em mídias sociais é diferente da recorrência de materiais educativos (normalmente lançados em uma *Landing Page*), portanto, o KPI de retenção destas duas funcionalidades será medido de maneira diferente. A adoção da ferramenta de email é facilmente medida: quando um usuário envia uma campanha de e-mails para seus contatos, ele adota a ferramenta, porém, a adoção de uma das ferramentas de análise é mais subjetiva, pois apenas olhando um relatório o usuário pode já ter adotado a funcionalidade. E assim como estes, existem muitos outros casos em que cada funcionalidade tem seu KPI definido de maneira específica.

Além do problema de definição dos KPIs específicos de cada funcionalidade, existe também o problema de instrumentação e metrificação destes KPIs. Nos exemplos colocados acima, para medir a adoção da ferramenta de email é possível extrair quais clientes já enviaram pelo menos uma campanha de emails do banco de dados que registra todas as campanhas. Já para medir a adoção de uma ferramenta de análise, é necessário registrar um evento quando um usuário vê uma página, onde será possível consultar quais clientes já executaram aquela ação pelo menos uma vez.

Somando a ambiguidade na definição e instrumentação de um KPI, a independência e autonomia dos times em relação aos demais, a cultura ágil instalada dentro do time de produto como um todo e a falta de experiência do time neste tipo de métrica com o grande volume de dados e estruturas que teria de ser estudado e trabalhado, vários problemas foram surgindo quando iniciou-se a extração e medição dos KPIs dentro dos times.

A proposta inicial deste projeto foi de estudar junto aos times quais seriam as métricas que cada um poderia extrair ou já extraía dos bancos de dados e desenvolver uma solução simplificada de sistema de armazenamento e visualização dos dados. Os problemas mapeados foram divididos em quatro grandes áreas distintas, que sozinhas já representam problemas relativamente grandes de serem resolvidos, mas que se mostram igualmente motivadoras de uma solução grande a ser estudada.

3.1: Definição da métrica, fonte de dados e interface

O primeiro problema surge devido às diferentes formas de definição das métricas de cada uma das funcionalidades, principalmente pelo fato de algumas

delas serem medidas por meio de eventos (ações específicas realizadas por um usuário) como é o caso das ferramentas de análises, onde o critério para utilização da funcionalidade é apenas visualizar a página de um relatório, e outras por meio de registros criados num banco de dados, como no caso da funcionalidade de email, na qual o critério de utilização seria medido com o número de registros de campanhas de email contidas no banco de dados.

Para efetuar a medição de ações do usuário (eventos), previamente a este trabalho foi investido tempo de todo o time de desenvolvimento na inserção de códigos que enviam estes registros de eventos dentro dos códigos que executam as funcionalidades e também orçamento da área para implementação de uma ferramenta que possui uma infraestrutura adequada para receber este tipo de registro, armazená-lo e mostrá-lo de diferentes formas. A Figura 2 mostra um exemplo de uma métrica de análise sendo medida pelo evento de visualização da funcionalidade de “Páginas mais acessadas”, citada anteriormente na descrição produto, dentro do Menu Analisar.

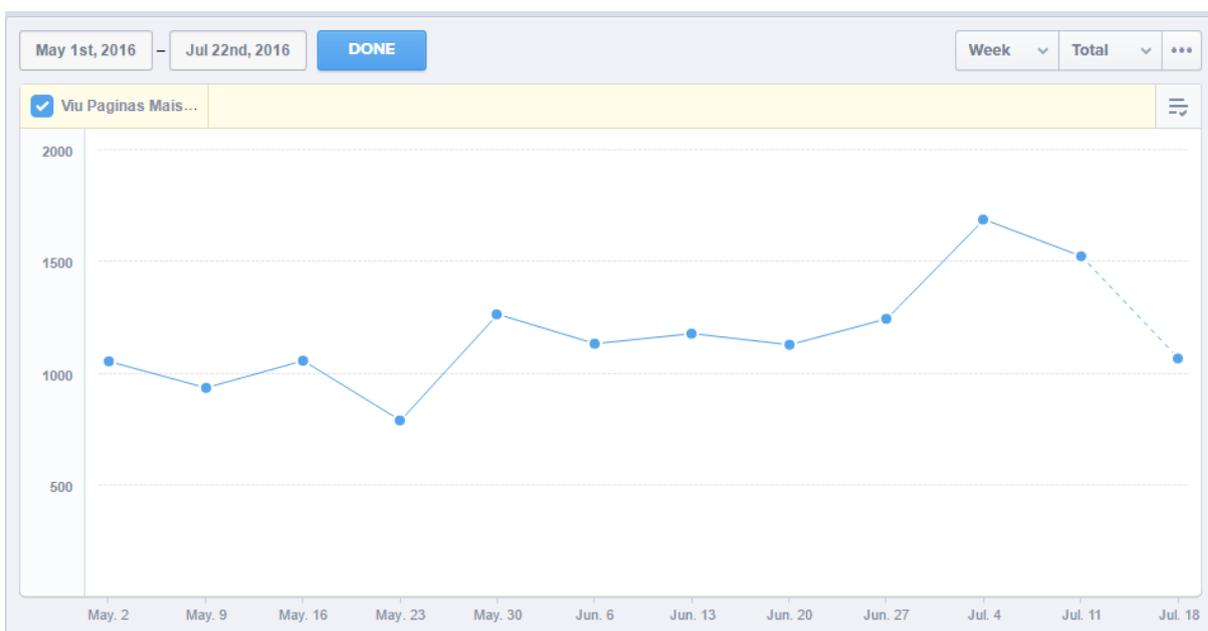


Figura 2 - Acompanhamento do evento de visualização de "Páginas mais acessadas"

Assim, para as medições feitas através de eventos há uma ferramenta robusta para armazenar e visualizar os dados, porém que não foi implementada da maneira adequada. E do outro lado, há bancos de dados “crus” (sem nenhum tipo

de tratamento), com diversas informações espalhadas de toda a operação do software, estruturadas de forma adequada à operação e não à análise, sem nenhum tipo de visualização acoplada (resultando em exportações de tabelas para planilhas onde era montados os filtros e gráficos).

Além disso, devido ao tamanho da infraestrutura de operação (detalhada na seção 3.3:), existe mais de um banco de dados onde as informações de operação estão armazenadas, portanto para alguns KPIs seria necessário o cruzamento de dados de duas ou três diferentes fontes de dados. Como por exemplo no caso da funcionalidade de emails, onde as informações das campanhas estão em um banco, os registros de performance (como emails entregues e emails abertos) estão em outro banco e, ainda, as informações das contas dos clientes se encontram em um terceiro banco, resultando em três extrações diferentes a serem realizadas e o trabalho manual de centralizar os dados de três planilhas em uma só, para então fazer a análise.

3.2: *Single Source of Truth*

Um problema que surge com a multiplicidade das fontes de informações e armazenamento é a divergência de dados quando consultas em diferentes bases de dados são feitas. As principais surgiram em alguns padrões:

3.2.1: Número de usuários ativos:

O conceito de usuário ativo é muito importante, pois define a totalidade de usuários que cada parte do produto pode atingir (normalmente este número é o denominador de uma equação de porcentagem de uso, por exemplo, na retenção de uma funcionalidade). Porém, este conceito é ambíguo, existem contas de usuários internos da empresa, contas de teste, contas de demonstração, contas em fase *trial* (potenciais clientes testando a ferramenta, durante um processo de venda), contas que já cancelaram o serviço, contas que estão em processo de cancelamento, contas bloqueadas devido a atrasos no pagamento e ainda contas de agências de marketing parceiras da Resultados Digitais, que recebem esta conta mas ainda não utilizam o software, apenas revendem ele para terceiros.

Excluindo a parte conceitual do problema, que pode ser resolvida com um consenso geral entre os gerentes de produto, seria necessário ter todas estas

informações disponíveis na hora de fazer uma medição de indicador, para ser possível filtrar todas as contas que não fazem parte do alcance máximo de usuários que seriam medidos.

Aqui surge o primeiro problema: a disposição destas informações na fonte de dados que seria extraído o dado. A fonte que possui todas essas informações é a base de dados do financeiro da empresa, onde ficam todas as informações de cada conta e das cobranças que são feitas aos clientes, porém todas as informações de utilização (os registros de operação do software) estão em uma outra base de dados. Além disso, essa base possui acesso restrito, pois tem informações financeiras sensíveis que não devem ser compartilhadas para a empresa.

3.2.2: Duplicidade de dados

Devido à restrição de dados da base que contém as informações das contas, quando o software de medição de ações do usuário via eventos foi implementado também foram feitas algumas integrações para que estes dados de contas também fossem acessíveis por esta fonte de dados.

Porém, as informações não estão igualmente apresentadas, gerando confusão no momento de extrair os dados e divergência de dados quando comparamos o que teoricamente seria a mesma informação nas duas fontes.

Segundo a teoria de *Single Source of Truth* [13], os dados devem ser armazenados apenas uma vez e apenas referenciadas em outras instâncias, para que sempre haja integridade de todos os dados independente de qual ferramenta ou caminho este dado tome para ser analisado. O comportamento que acontece nos bancos e ferramentas da Resultados Digitais divergia desta teoria, gerando exatamente a falta de integridade dos dados que são utilizados para tomadas de decisão importantes.

O fato de existirem duas bases de armazenamento (uma orientada a registros de eventos e outra orientada a registros virtuais criados pelos usuários) gerava duplicidade de dados nos KPIs, assim, cada time poderia medir o que teoricamente seria o mesmo dado, mas que resultava em dois números diferentes. Por exemplo: se registramos via evento todas as vezes que um usuário aperta no botão de “Enviar uma campanha de emails” e contamos no banco de dados de registros todos os registros de campanhas de email enviadas pelo mesmo usuário, esperamos o

mesmo resultado, porém como são dois armazenamentos diferentes, não existe certeza de onde o dado pode ter divergido, se um evento não foi registrado, se uma propriedade do banco de dados não foi alterada, se os filtros de usuários ativos estão divergentes nas duas fontes. Existem muitas fontes de erro, que tornam inviável a validação de cada KPI a cada momento em que ele fosse analisado.

3.3: Ineficiência da arquitetura atual para análises

Quando se começou a extrair algumas métricas dos times, alguns pontos críticos começaram a aparecer durante a execução das *queries* que rodavam e extraíam os dados necessários. Mesmo trabalhando para que a consulta fosse a mais eficiente possível, algumas delas ainda tomavam muito tempo devido a alguns fatores:

1. Quantidade de registros nas tabelas;
2. Falta da existência de índices estruturados para o cruzamento de informações de tabelas;
3. *Overhead* de cruzamentos de tabelas, devido à forma com que o banco está estruturado.

Estes problemas não seriam facilmente contornados pois a arquitetura toda do sistema é montada em cima deste banco de dados e uma alteração nela não seria simples. Para entender melhor todos os problemas que foram apresentados até o momento, foi feito um esboço de como está estruturada a arquitetura de todos os dados e serviços que o RD Station opera, conforme ilustrado na Figura 3.

As partes importantes do esquema mostrado na Figura 3 para o presente trabalho são as fontes de dados, representadas pelos cilindros. Nota-se a diversidade das mesmas, onde informações de Landing Pages encontram-se parte em um local fora da arquitetura do web app que é o RD Station e parte encontra-se dentro do banco de dados da aplicação (Postgresql Heroku). Além disso, todos os eventos de email são registrados em uma fonte de dados externa (MongoDB), junto à *timeline* de Leads (registros de todos os eventos de email, formulário, visita ao site e edição de informação que qualquer Lead de qualquer cliente), formando bases com entradas na ordem de bilhões e tamanho na ordem de Terabytes.

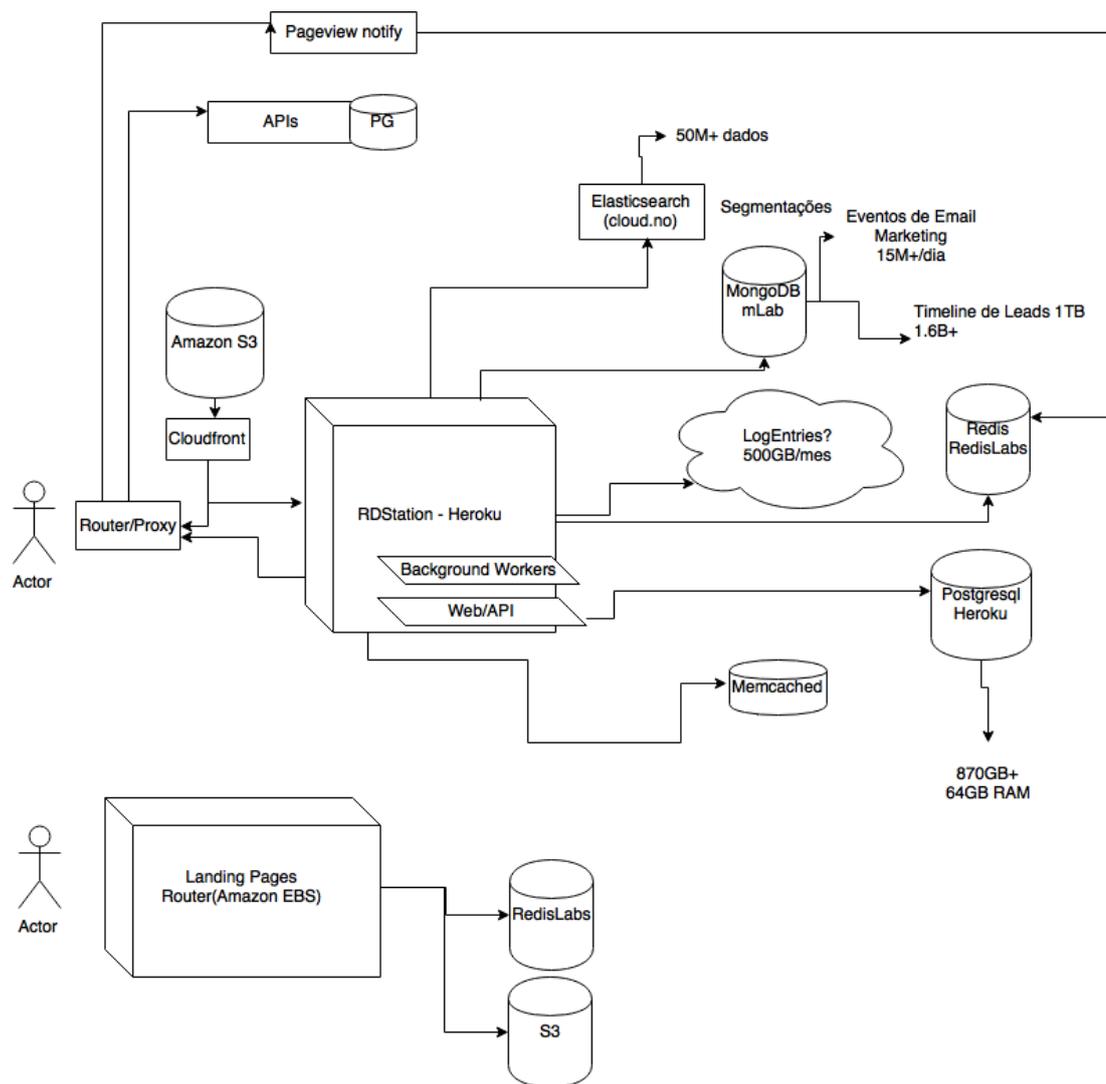


Figura 3 - Arquitetura de operação do RD Station

Fora a arquitetura de serviço do RD Station mostrada acima, temos também a arquitetura de serviço do software que faz o controle financeiro de todos os clientes, citado anteriormente, chamado de Contas. Nele, como comentado, interessa ao projeto apenas a base de dados que contém as informações e propriedades das contas dos clientes, para execução dos filtros quando se quer encontrar o número de usuários ativos e algumas segmentações básicas (plano de pagamento, por exemplo).

Por fim, temos também a arquitetura de registro de eventos, também fora desta que foi explicitada acima, onde temos uma base de dados inacessível aos

gerentes de produto, podendo apenas ser consultada via interface do software implementado para visualização das informações de eventos.

Este resumo das arquiteturas de serviços e dados por si já resume também a maioria dos problemas mapeados neste projeto. A grande quantidade de dados, a distribuição dos mesmos entre diferentes arquiteturas juntamente à duplicidade e conseqüente divergência de informações em consultas são os maiores motivadores para o início de uma solução. Além disso, toda arquitetura apresentada (excluindo a base de dados de eventos) é de onde a própria aplicação, o produto que os clientes utilizam quase diariamente, retira os dados, portanto cada consulta que é feita para analisar uma informação “concorre” com um cliente para consumir o mesmo recurso, o dado registrado no banco.

3.4: Métricas ainda não exploradas

Além dos problemas com a extração das métricas definidas e mapeadas previamente, havia também outros indicadores que não tinham como ser coletados na estrutura atual do RD Station.

Principalmente os KPIs de sucesso das funcionalidades, pois muitas vezes estes indicadores não são métricas medidas na própria funcionalidade. Adoção e retenção são métricas de uso, então podem ser medidas diretamente com números produzidos na própria área do software, porém a métrica que mede se a funcionalidade entrega o valor pelo qual foi pensada pode não estar diretamente ligada a uma métrica produzida pela funcionalidade. Um exemplo disso são as funcionalidades de tráfego do RD Station, das quais nenhuma mede ou altera diretamente o número de visitas no site de um cliente, apenas o ajuda a promover conteúdos e páginas, porém a métrica de sucesso é justamente o número de visitas do site. O problema neste caso é de onde seria tirada o dado, visto que não era guardado em nenhuma das tabelas das funcionalidades de tráfego.

Outra métrica que os times de produto ainda não trabalham é a medição quantitativa de indicadores de suporte. Em diversos processos de gestão de produto, o suporte é utilizado amplamente para exploração de problemas que usuários enfrentam, pois é um canal direto de comunicação de clientes que estão com alguma dificuldade no uso do software, ou gostariam de sugerir uma melhoria de funcionalidade, ou ainda uma funcionalidade nova. O problema é que os

gerentes de produto ainda subutilizam este canal para gerar informações valiosas sobre seus produtos. Apenas nestes casos de exploração ou casos de um problema muito claro e visível devido à recorrência de tickets é que um gerente de produto utiliza uma informação vinda deste canal para tomar uma decisão no seu produto.

Acredita-se que uma análise do suporte com maior frequência traga *insights* importantes aos times de produto, resumidos em quatro objetivos principais:

1. Evitar que problemas grandes sejam resolvidos de forma reativa, como normalmente eram, uma vez que apenas eram acionados quando havia uma grande quantidade de clientes reportando um problema, mas sim de forma proativa, observando frequentemente as tendências de crescimento de tickets em cada funcionalidade;
2. Auxiliar a medição do impacto de lançamentos de novas funcionalidades ou melhorias, observando o número de clientes com problemas na utilização da nova funcionalidade;
3. Categorizar de forma adequada os tickets dos clientes, a fim de facilitar a segmentação e análise dos mesmo por um gerente de produto;
4. Traçar correlações de problemas sofridos pelos clientes observados no suporte com ações críticas para a empresa (principalmente o cancelamento do serviço).

Por fim, como as métricas de produto estavam em fase de implementação, os times focaram primeiramente em instrumentar disponibilizar a medição dos KPIs, sem se preocupar neste primeiro momento com análises mais profundas que podem ser feitas. Quando se observa uma queda de um KPI, o gerente de produto precisa investigar o porquê da queda, normalmente “quebrando” a métrica em mais um nível, subdividindo ela em vários grupos para tentar encontrar padrões de comportamento que tenham influenciado a queda.

Por isso, apesar de poder medir alguns KPIs, ainda não é possível (ou muito difícil, trabalhoso ou demorado) por meio de análises quantitativas entender quais fatores e comportamentos atuam para que a métrica esteja no patamar observado.

Capítulo 4: Desenvolvimento da Nova Base de Dados

4.1: Proposta inicial

Observando todos os problemas descritos no capítulo anterior, apesar de nem todos serem aparentes no primeiro momento, a proposta inicial deste projeto (anteprojeto) foi definida nos seguintes passos:

1. Estudo e definição com os times de quais informações e indicadores são importantes para a tomada de decisão e planejamento de cada time.
2. Mapeamento de todos os indicadores e informações com as respectivas bases de dados.
3. Escolha da ferramenta que fará as extrações e visualizações, definindo também frequência e método de atualização.
4. Execução das extrações.
5. Interface e amostragem dos dados.
6. Implementação da ferramenta.

A primeira etapa ocorreu como esperado: durante cerca de três semanas diversas conversas foram marcadas com os gerentes de produto de cada time para documentar como cada um extraía suas métricas (não apenas os KPIs) das bases de dados e também definir quais seriam as métricas ainda não extraídas.

Neste ponto alguns dos problemas mencionados no Capítulo 3: começaram a aparecer. A heterogeneidade de ferramentas de consulta e de fontes de dados já aparentava aumentar a complexidade do projeto, juntamente com a falta de coerência dos dados entre estas fontes.

Então iniciou-se a segunda etapa, onde se procurou mapear todos os indicadores de todos os times em todas as fontes de dados, mesmo aqueles que já eram coletados, a fim de encontrar a base mais confiável e com mais dados disponíveis. Aqui surgiram os problemas de métricas que eram medidas via eventos e outras medidas via registros e estados, configurando dependências de duas

estruturas diferentes. Também foram observadas algumas consultas inviáveis de serem executadas pelos PMS, devido a problemas de performance derivados do tamanho das tabelas.

Observando estes problemas, antes de partir para a terceira etapa, percebeu-se que uma nova estrutura de dados precisaria ser desenhada para que pudéssemos contornar todos os problemas encontrados e continuássemos com a proposta de facilitar a medição das métricas pelos gerentes de produto.

Ao mesmo tempo, o time responsável por performance da ferramenta de email do RD Station enfrentava problemas semelhantes a estes, porém com o objetivo de implementar uma melhoria para a funcionalidade. Neste ponto, os dois projetos convergiram para o mesmo ponto: a resolução do problema de escalabilidade do banco de dados e SSOT.

Devido à urgência maior e à experiência com sistemas de informação, arquiteturas de dados e de serviços de extração presente no time de desenvolvimento, o mesmo desenvolveu toda a proposta de uma nova arquitetura unificada de dados, da qual extrairíamos todas as informações de análises de dados do RD Station, tanto para alimentar de forma mais eficiente algumas de nossas funcionalidades presentes no software, quanto para facilitar (e em alguns casos possibilitar) a extração de métricas de produto para os PMs, resolvendo, assim, o problema de SSOT. Esta proposta é ilustrada na Figura 4.

A proposta traz uma estrutura baseada em um *Data Warehouse* populado com eventos de diferentes “Clientes”, sejam eles usuários ou aplicações. Deste DW seriam executados diversos EMRs (Elastic Map Reduce - <https://aws.amazon.com/pt/emr/>), processando e agregando os dados em *schemes* SQL construídos para a finalidade que cada time definir. Na Figura 4 temos um exemplo de *schemes* de três times (Mushin, Dispatch e Vikings são os nomes internos dos times) com suas respectivas finalidades.

Por se tratar de uma solução de grande porte, que tomará tempo a ser implementada e precisa ser muito bem pensada para não criar legados, o projeto de implementação da mesma foi separado deste. Portanto, o escopo deste projeto abrange:

1. Mapeamento das métricas de produto que cada PM deve analisar;

2. Criação de um *scheme* SQL enxuto e otimizado que contém os dados necessários para extrair as métricas de produto;
3. População destas tabelas com amostras dos dados disponíveis hoje (estratégia de ETL);
4. Prova de conceito da melhoria de performance e integridade das análises de produto.

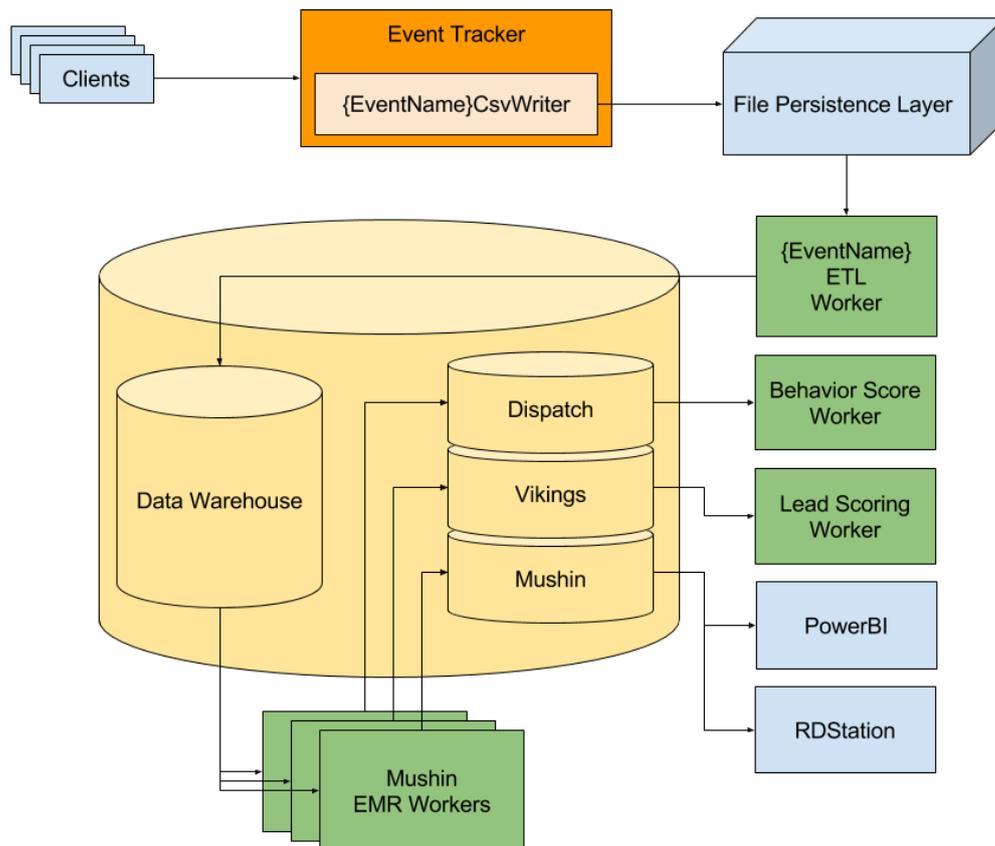


Figura 4 - Proposta de arquitetura de dados do RD Station para análises e otimização de funcionalidades

4.2: Proposta Final

Seguindo o escopo final acordado deste projeto, a seguir são descritos os desenvolvimentos de cada uma das quatro etapas.

4.2.1: Mapeamento das métricas de produto

Esta primeira etapa já havia sido realizada durante o estudo prévio do projeto, antes mesmo da identificação de que a nova estrutura de BI seria necessária. Alguns times ainda precisavam definir mais precisamente suas métricas, mas grande parte delas foram mapeadas.

Para avançar à próxima etapa, seria necessário selecionar uma parte das métricas (preferencialmente as mais problemáticas), com as quais seria desenvolvida uma proposta de banco de dados que permitisse a fácil extração, integridade dos dados e desacoplamento das bases de dados que operam o software. Com isso, seria possível desenvolver a prova de conceito das melhorias propostas pela nova arquitetura de BI.

Na Tabela 1 são mostradas as métricas-foco dessa prova de conceito e as motivações por trás de cada uma.

Métrica	Motivação
Número de contas ativas na base de clientes	Hoje o critério para discussão deste número é ambíguo, além de ser extraído de duas bases diferentes
Performance das campanhas de email	Apesar de já serem medidas, não é possível fazer consultas sob demanda, visto que a medição é feita via scripts que rodam semanalmente
Número de Leads gerados nas Landing Pages (KPI de sucesso)	Um dos KPIs de sucesso ainda não mensurados devido ao custo das consultas
Engajamento e adoção das funcionalidades de tráfego	Mesmo problema do primeiro item: ambiguidade no número de contas que fazem parte do escopo todo de usuários
Quantidade de visitantes separados por fonte de tráfego (KPI de sucesso)	Outro KPI de sucesso não mensurado devido ao custo das consultas e estrutura de apresentação dos dados no banco

Tabela 1 - Motivação da escolha das métricas da Prova de Conceito

4.2.2: Criação de um *scheme* SQL enxuto e otimizado

Partindo destas métricas escolhidas, foi desenvolvido um esquema de banco de dados relacional no qual os dados já teriam a granularidade adequada para a análise a ser feita, na frequência definida para cada métrica.

Assim seria possível ter independências das tabelas e relações atuais nos bancos de dados e desenvolver o esquema mais otimizado possível, o que permitiria uma escalabilidade muito maior.

O primeiro passo foi mapear as entidades e relações presentes em cada uma das métricas e montar um esquemático com todas elas, para utilizar um algoritmo de modelagem de bancos de dados relacionais. Após essa primeira iteração, alguns ajustes e refinamentos manuais foram feitos, para que fossem incluídos casos “específicos” de medições não previstas num cenário recorrente e, ainda, possibilitasse a segmentação de alguns dados para melhor entendimento das métricas e validação de hipóteses formadas no processo de gestão de produto. A proposta final de banco de dados ficou conforme exibido na Figura 5.

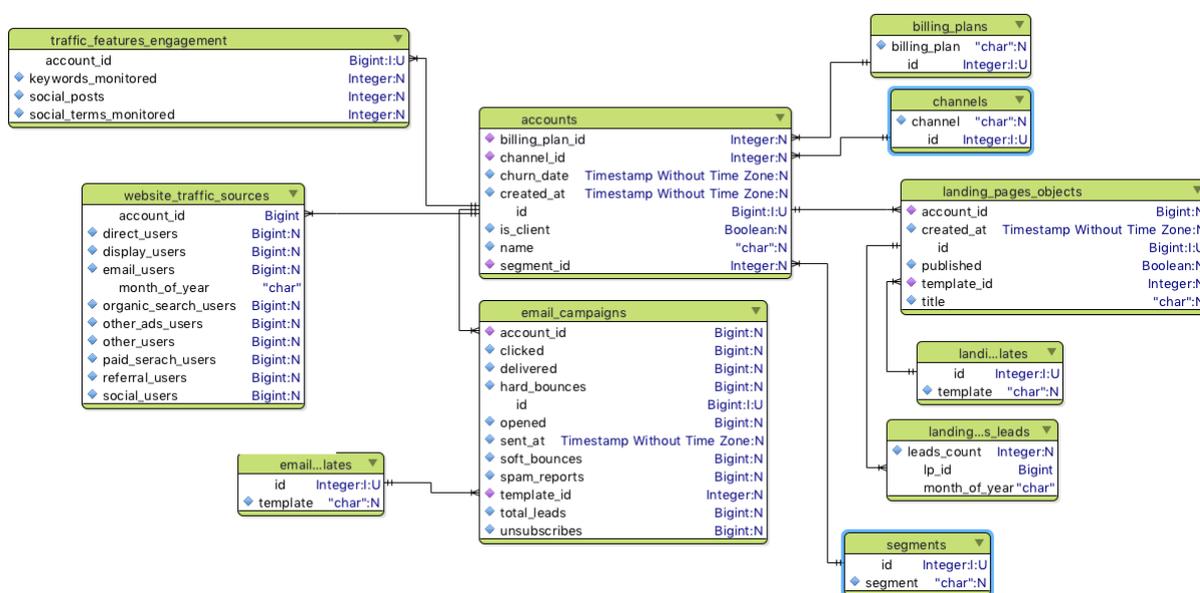


Figura 5 - Diagrama de entidades e relações proposto para a prova de conceito

4.2.3: Estratégia de ETL

Com este banco contendo as tabelas adequadas para fazer todas as análises necessárias dentro do escopo das métricas apresentadas em 4.2.1:, seria necessário criar este novo banco de dados, hospedá-lo em um local acessível e desenvolver uma estratégia de população de amostras de dados.

O banco foi criado utilizando PostgreSQL (<https://www.postgresql.org/>) localmente em uma máquina, a fim de facilitar a criação das tabelas (via interface gráfica do software pgAdmin III - <https://www.pgadmin.org/>) e inserção dos dados. Mas foi em seguida migrado para um banco de dados de teste criado no Heroku (<https://www.heroku.com/> - plataforma de serviços web que hospeda o RD Station, vide Figura 3), hospedando os dados na Amazon WebServices (<https://aws.amazon.com/>) e tornando fácil a integração com interfaces de consultas e *dashboards*.

Para a migração de dados, foi necessário primeiramente extrair dados crus dos bancos de dados atuais, em casos trabalhá-los em planilhas ou scripts para que o formato se adequasse àquele proposto no novo banco (isso foi necessário principalmente pois foi tomada a decisão consciente de não estruturar o banco pensando em como a migração seria feita, alvejando apenas a otimização da organização de dados).

Vale também ressaltar que no casos de extrações muito custosas, apenas algumas amostras de dados foram retiradas, pois o objetivo seria apenas provar o conceito da nova base de dados e não tornar possível (ainda) a medição histórica de todas as métricas. Os casos mais críticos como número de Leads gerados em Landing Pages, Visitas por fonte de tráfego e performance de campanhas de email foram limitados a dois meses de dados e algumas contas de clientes específicas.

Houve um problema durante a extração dos dados das campanhas de email que impossibilitou o carregamento dos dados antes que a base de dados fosse migrada, portanto estes dados ficaram hospedados apenas na base local.

4.2.4: Prova de conceito

Após ter a base de dados criada e devidamente hospedada, com acesso aberto a aplicações web, escolheu-se uma ferramenta *open-source* para

amostragem dos dados chamada Metabase (<http://www.metabase.com/>). Ela possui integração direta com o Heroku, o que facilitou a instalação da mesma e integração com o banco de dados criado.

A seguir, foram desenvolvidas as *queries* que extrairiam os dados deste novo banco e alimentariam essa ferramenta para que ele disponibilizasse via tabelas, indicadores numéricos e gráficos de todas as métricas de produto (dentre as escolhidas) em um ou mais *Dashboards*.

A Tabela 2 mostra a prova de conceito feita com todas as métricas pensadas para todas as tabelas e as figuras 6 e 7 ilustram dois *dashboards* (criados no Metabase) para dois times diferentes, ilustrando como seria parte da entrega final do projeto.

Outro ponto que vale citar é justamente a facilidade de executar cruzamentos entre tabelas, uma vez que todas estão simplificadas e unificadas em um único esquema de banco de dados. Para filtrar todas as métricas da Tabela 2 apenas por usuários ativos, agora basta utilizar os índices e chaves estrangeiras com a tabela de contas de clientes, eliminando ambiguidades de consultas e a necessidade de cruzamento de tabelas em planilhas.

Além do fácil acesso a dados “extras”, com a finalidade de segmentar as métricas e entender melhor o comportamento delas, como por exemplo no caso de dividir a adoção de uma ferramenta em diferentes segmentos de clientes (coluna *segments* da tabela *accounts*), ou mesmo avaliar a diferença na geração de leads de Landing Pages que utilizam o *template* 1 com as que utilizam o *template* 2. São diversos ganhos no que diz respeito à análise e construção de correlações que aparecem à medida que os dados são mais facilmente acessados e a gestão de produto como um todo passa ser mais *data-driven*.

Métrica	Query
Número de contas ativas na base de clientes	<code>SELECT count(id) FROM accounts a WHERE is_client = TRUE</code>
Adoção da funcionalidade de email	<code>SELECT count(distinct(account_id)) FROM email_campaigns ec</code>

Retenção da funcionalidade de email	<i>SELECT count(distinct(account_id)) FROM email_campaigns ec WHERE (sent_at BETWEEN now() AND (now() – interval '30 days'))</i>
Perfomance de entregabilidade de campanhas de email	<i>SELECT delivered / total_leads FROM email_campaigns</i>
Performance de abertura de campanhas de email (KPI de sucesso)	<i>SELECT opened / total_leads FROM email_campaigns</i>
Performance de clique de campanhas de email (KPI de sucesso)	<i>SELECT clicked / total_leads FROM email_campaigns</i>
Adoção da funcionalidade de Landing Pages	<i>SELECT count(distinct(account_id)) FROM landing_pages_objects lpo WHERE published = TRUE</i>
Retenção da funcionalidade de Landing Pages	<i>SELECT count(distinct(account_id)) FROM landing_pages_objects lpo WHERE published = TRUE AND (created_at BETWEEN now() AND (now() – interval '30 days'))</i>
Número de Leads gerados nas Landing Pages (KPI de sucesso)	<i>SELECT account_id, SUM(leads_count) FROM landing_pages_objects lpo join landing_pages_leads lpl on lpo.id = lpl.lp_id WHERE month_of_year = to_char(now(), 'yyyy-mm')</i>
Adoção da funcionalidade de Monitoramento Social	<i>SELECT count(distinct(account_id)) FROM traffic_features_engagement WHERE social_terms > 0</i>
Adoção da funcionalidade de Postagem Social	<i>SELECT count(distinct(account_id)) FROM traffic_features_engagement WHERE social_posts > 0</i>

Adoção do Painel de Palavras-Chave	<pre>SELECT count(distinct(account_id)) FROM traffic_features_engagement WHERE keywords > 0</pre>
Quantidade de visitantes separados por fonte de tráfego (KPI de sucesso)	<pre>SELECT account_id, (direct_users + email_users + referral_users + social_users + organic_search_users + paid_search_users + other_users + other_advertising_users + display_users) FROM website_traffic_sources WHERE month_of_year = to_char(now(), 'yyyy-mm')</pre>

Tabela 2 - Queries para extração das métricas da Prova de Conceito

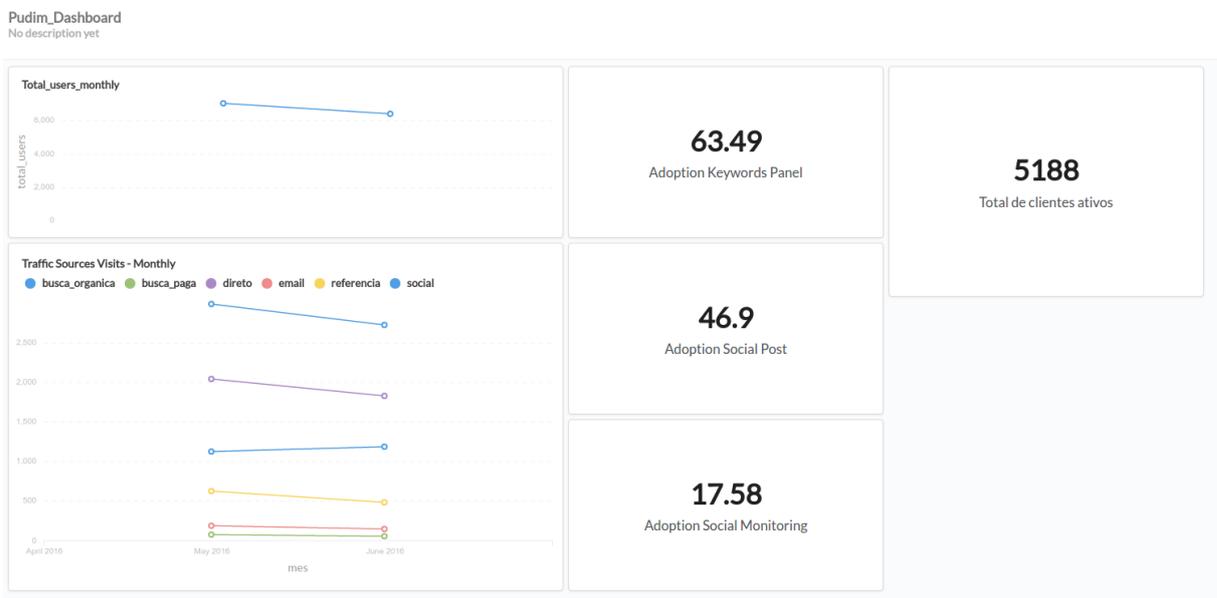


Figura 6 – Dashboard modelo para o time das funcionalidades de Tráfego do RD Station

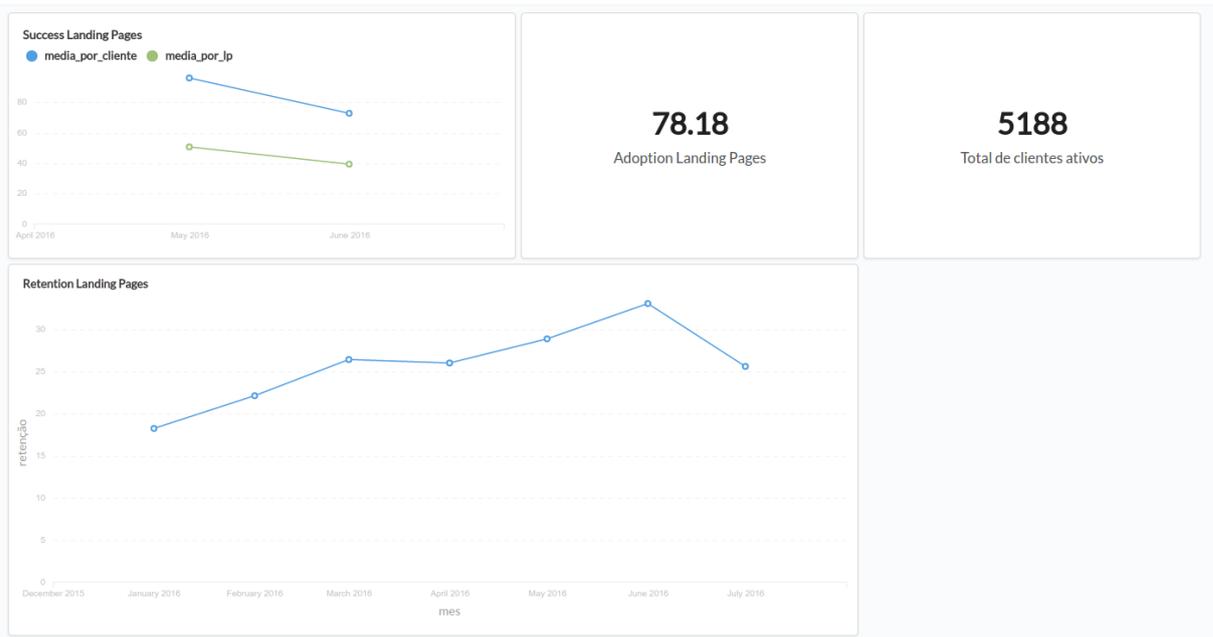


Figura 7 - Dashboard modelo para o time da funcionalidade de Landing Pages do RD Station

Capítulo 5: Desenvolvimento de Relatórios de Suporte

O desenvolvimento dos indicadores e relatórios recorrentes de suporte também começou com uma longa fase de estudos, mas por um motivo distinto: no lugar de indefinições em *como* extrair certos indicadores de bases de dados, o problema estava em descobrir o *quê* deveria ser extraído.

Para isso, primeiro estudou-se o que havia de informação disponível nos atendimentos de suporte via tickets por email. A ferramenta utilizada pela Resultados Digitais é o software Zendesk (<https://www.zendesk.com/>), uma plataforma *helpdesk* que permite organizar, agrupar e armazenar os atendimentos de forma mais eficiente e que também permita justamente o que se buscava durante esta parte do projeto: analisar os dados dos atendimentos.

5.1: Zendesk

A ferramenta disponibiliza a criação de campos de categoria para que cada ticket atendido seja classificado e posteriormente analisado. Nesse aspecto, a empresa já categorizava seus tickets nos seguintes agrupamentos:

1. Produto/Serviço: diferencia para qual serviço ou produto da empresa o cliente está solicitando o atendimento.
2. Funcionalidade: diferencia de qual funcionalidade do RD Station se trata o atendimento.
3. Tipo de suporte: diferencia qual o problema reportado pelo cliente, dividido em:
 - a. Bug: a ação desejada existe, mas não funciona.
 - b. Performance: a ação desejada existe, mas é demorada ou não executa no tempo desejado.
 - c. UX/UI (*User Experience / User Interface*): a ação desejada existe, funciona, não tem problemas de performance (tempo de espera), mas o cliente não consegue realiza-la.

- d. Educação: a ação desejada existe, o cliente consegue fazer mas tem dúvidas conceituais sobre a funcionalidade
 - e. Comunicação: a ação desejada existe, mas o cliente não sabe
 - f. Funcionalidade inexistente: a ação desejada não é possível de ser executada pelo usuário na versão atual do software
 - g. Tarefa Manual: tipo de ação específica que não é possível realizar pelo software, mas o suporte do RD Station realiza manualmente (estas tarefas estão mapeadas em um processo interno da empresa)
4. Prioridade: categoria interna para classificar a prioridade do atendimento, normalmente atrelada ao tempo de resposta do ticket (quanto mais tempo sem uma resposta, maior a prioridade)

O Zendesk também disponibiliza uma ferramenta de análises bastante flexível dentro do próprio software, baseada na plataforma GoodData (<http://www.gooddata.com/>), que permite montar diversos tipos de relatórios, tabelas e gráficos por meio de consultas num banco bem estruturado.

Com isso em mãos, o estudo e *benchmarking* de métricas de suporte para desenvolvimento de produto foi iniciado.

5.2: Estudo e *benchmarking* de métricas de suporte

A base toda de estudos foi feita através da busca de experiências práticas de outras empresas com o mesmo problema: como transformar milhares de tickets que citam os mais diversos tipos de problemas em *inputs* reais ao produto da empresa. Este método é bastante utilizado na Resultados Digitais pois muitas empresas com características similares (principalmente empresas SaaS norte-americanas) costumam compartilhar seus aprendizados em blogs ou livros eletrônicos (ebooks).

Entendendo melhor o problema, observa-se que a dificuldade de integrar a área de suporte com a gestão de produto está em alguns pontos:

1. Para qualquer tipo de decisão, gerentes de produto baseiam-se muito em dados, pois com uma base grande de clientes é muito arriscado decidir por um rumo baseado apenas em feedbacks qualitativos, sem

uma amostra estatística minimamente suficiente para que uma hipótese de problema seja validada [15];

2. Os times que trabalham diretamente com clientes tendem a responder rapidamente às demandas dos clientes, portanto mesmo que tomem o cuidado de não transformar em um problema uma solicitação de apenas um cliente, é comum que se prendam apenas ao aspecto “frequência” de um problema (o que é ótimo, em alguns casos), deixando de lado outros aspectos igualmente relevantes a uma decisão de produto [15] [16], como impacto ao usuário, quantidade de usuários afetados, impacto no negócio da empresa, entre outros;
3. Por lidar com diversas informações o tempo todo, excluindo casos em que já há uma hipótese de problema que está sendo validada, um gerente de produto não terá disponibilidade recorrente de correr atrás de informações extras, como parar para analisar tickets de suporte a fim de encontrar padrões que possam servir de *input* qualitativo para seu produto;
4. Mesmo que dispusessem de tempo, para ter a informação da maneira mais correta, eles deveriam analisar (ler) todos os tickets de clientes, o que se torna inviável à medida que o número de clientes aumenta em grande escala num *SaaS*.

Também se observa que as únicas métricas que se pode medir dos atendimentos são: o número de tickets abertos por cliente e o tempo médio de resposta (tempo até que o ticket seja fechado). Portanto a parte mais importante da metrificação destes tickets está na verdade na classificação [15] [20] dos mesmos para uma posterior visualização de um problema crítico e entendimento profundo do mesmo (encontrar a causa raiz do problema a ser resolvido, e não apenas o que é reportado pelos clientes).

Para isso facilitar este processo, existem duas ações observadas em todos os *benchmarkings*: o estudo e melhoria da classificação e entendimento do problema dentro do próprio time de suporte, para que quando conversem com gerentes de produto, tragam informações mais estruturadas e validadas por dados, e a inserção de pessoas de produto (e até mesmo executivos) dentro do time do

suporte por um período de tempo [17] (*All-Hands Support*), para que todas as pessoas atendam clientes pelo menos algumas vezes por mês e passem a entender melhor quais problemas eles enfrentam.

5.3: Proposta de processo

A partir destas duas “macro” soluções (melhoria da classificação e inserção de pessoas de todas as áreas no atendimento), decidiu-se por adaptar e validar qual delas se encaixaria melhor na Resultados Digitais.

Apesar de ser uma solução interessante, a ideia de criar um processo para trazer os times de produto (principalmente os gerentes de produto) para o atendimento no suporte foi descartada pois indiretamente este tipo de ação já acontece na empresa num nível considerado adequado.

O time de suporte pertence à área de Sucesso do Cliente, e possui um time não-técnico, que recebe todas as solicitações e está habilitado e treinado a responder dúvidas e resolver problemas recorrentes de clientes, porém quando o problema necessita de uma investigação técnica, ele é repassado ao time especializado na funcionalidade que está ligada ao problema. Assim, todos os dias há pelo menos um desenvolvedor de cada um dos times de produto respondendo a solicitações técnicas de clientes e todos os tickets relacionados a melhorias de funcionalidades ou funcionalidades que não existem no software são também repassados aos gerentes de produto.

Mesmo assim, grande parte dos tickets não trata de um problema técnico e acaba sendo resolvida pelo time de suporte da empresa, e são estes tickets que os gerentes de produto não acessam recorrentemente e, portanto, necessitariam de um novo processo para garantir que esta fonte de informação está sendo coletada.

Então, foi iniciada uma proposta de como, quando e o quê analisar quantitativamente nos atendimentos de suporte. O primeiro passo foi rever um estudo já feito na empresa que descrevia quais informações um gerente de produto deveria conseguir extrair deste tipo de análise e refinar a proposta. Ao final, chegou-se aos seguintes pontos (todos, exceto o último, relacionados a cada funcionalidade em específico):

1. Tendências de problemas e correlação com a recorrência de uso
 - a. Observação do número de tickets abertos por cliente nos últimos meses
 - b. Observação da correlação entre o uso da funcionalidade e a quantidade de problemas reportados por clientes
2. Efetividade e problemas com lançamentos
 - a. Observação do número de tickets abertos por cliente ao lançar uma melhoria de funcionalidade, onde espera-se uma diminuição do número
 - b. Observação do mesmo número em uma funcionalidade nova, identificando o quão problemática ela pode ser
3. Correlação com cancelamentos do serviço
 - a. Por quais problemas de quais funcionalidades um cliente que cancela o serviço passa
4. Feedbacks qualitativos por amostras
 - a. Coletas de feedbacks e/ou problemas críticos reportados por clientes, escolhidos pelo time de suporte
5. Recorrência de Tarefas Manuais
 - a. Observar quais das tarefas manuais que a empresa disponibiliza via suporte estão sendo mais requisitadas, para que possivelmente se tornem funcionalidades do produto
6. Comparativo entre funcionalidades
 - a. Observação de quais funcionalidades estão gerando mais problemas resolvidos via suporte

O segundo passo foi novamente estudar como outras empresas faziam este tipo de análise, mas o que se observou foi que, na verdade, a categorização é o principal fator para retirada de informações quantitativas do suporte e ela depende muito do produto e da estrutura da empresa, portanto cada um dos *benchmarks* feitos utilizava uma estrutura diferente. Mesmo assim, observou-se o padrão de algumas categorias que apareciam mais, como Bugs, Usabilidade e Sugestões.

Por isso, olhando para as categorias já registradas no suporte da Resultados Digitais (citadas na seção 5.1:) e pensando no esforço de uma reestruturação de processo e treinamento de todas as pessoas que estão aptas a atender tickets, resolveu-se não alterar esta estrutura pois, apenas utilizando as categorias Funcionalidade e Tipo de Suporte, já é possível retirar as informações de maneira satisfatória. A única mudança proposta na análise (e não no processo) seria de consolidar as categorias de UX/UI, Comunicação e Educação em uma categoria apenas, chamada de Usabilidade. Esta mudança tem duas principais motivações: a primeira e principal é a ambiguidade de preenchimento por se tratarem de problemas similares, a qual gera desconfiança na análise, quando feita de forma separada, e a segunda é o fato de que para analisar o produto como um todo, as três categorias representam uma funcionalidade que existe e funciona, mas que o usuário não consegue utilizar sozinho.

Outra melhoria proposta ao processo foi feita após as tentativas de execução de relatórios pela ferramenta Zendesk (mostradas ao final deste capítulo). Trata-se da integração dos dados da base de dados de contas (citada no Capítulo 3:), para que fosse possível realizar a correlação entre os tickets e os cancelamentos de serviços e também segmentar melhor os tickets por características de clientes (plano de funcionalidades, por exemplo).

Ao fim, as métricas definidas foram as seguintes:

1. Tickets criados/ nº de clientes, separados por funcionalidade e agrupados por mês e tipo de suporte (abrange a observação de tendências, de tarefas manuais e de lançamentos)
2. Tempo médio de resolução dos tickets, separados por funcionalidade e agrupados por mês e tipo de suporte
3. Gráfico de correlação entre Tickets criados/nº de clientes x KPI de retenção, para cada funcionalidade
4. Diagrama de pareto de todos os tickets, separados por funcionalidade
5. Cópias de textos escritos por clientes em tickets, em problemas críticos ou clientes insatisfeitos, expondo feedbacks qualitativos para reforçar os *outputs* das métricas

A ferramenta de coleta é o próprio Zendesk, visto que ela já possui uma ferramenta de análises interna ao software. Porém, para fazer os devidos ajustes como a junção das categorias, a construção dos gráficos da forma que foi proposta e correlações com informações contidas fora do software, esta primeira proposta (que está em fase de aprovação durante a escrita deste documento) ainda centraliza todas as informações em uma planilha.

As figuras 8, 9, 10 e 11 mostram algumas prévias de análises, métricas e gráficos que já foram coletadas para a prova de conceito e apresentação da proposta em andamento.

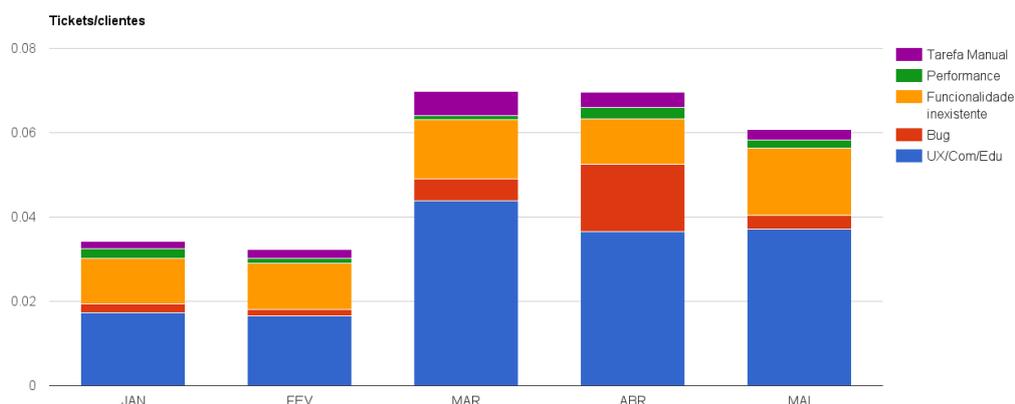


Figura 8 - Gráfico de Tickets/cliente agrupado por mês e por tipo de suporte, para a funcionalidade de Landing Pages

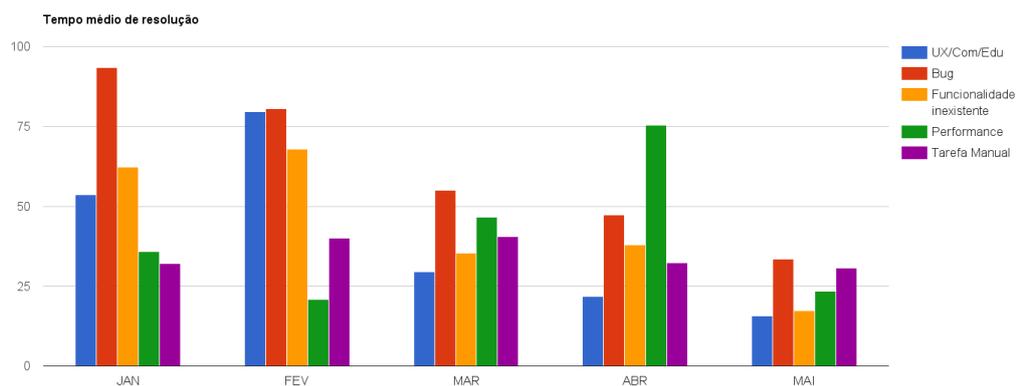


Figura 9 - Gráfico de tempo médio de resolução de tickets agrupado por mês e por tipo de suporte, para a funcionalidade de Landing Pages

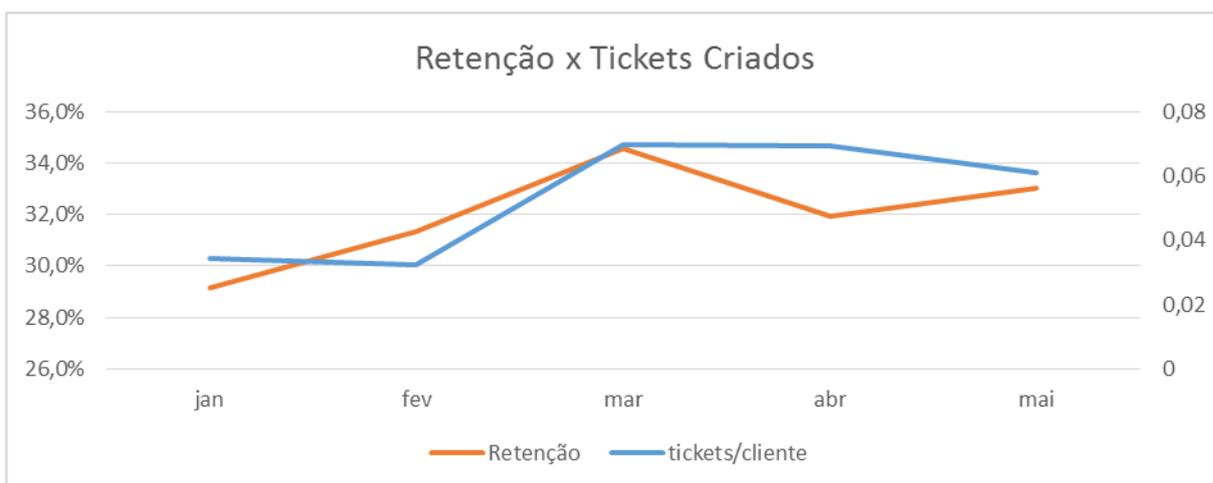


Figura 10 - Evolução de tickets/cliente acompanhada pela retenção da funcionalidade de Landing Pages

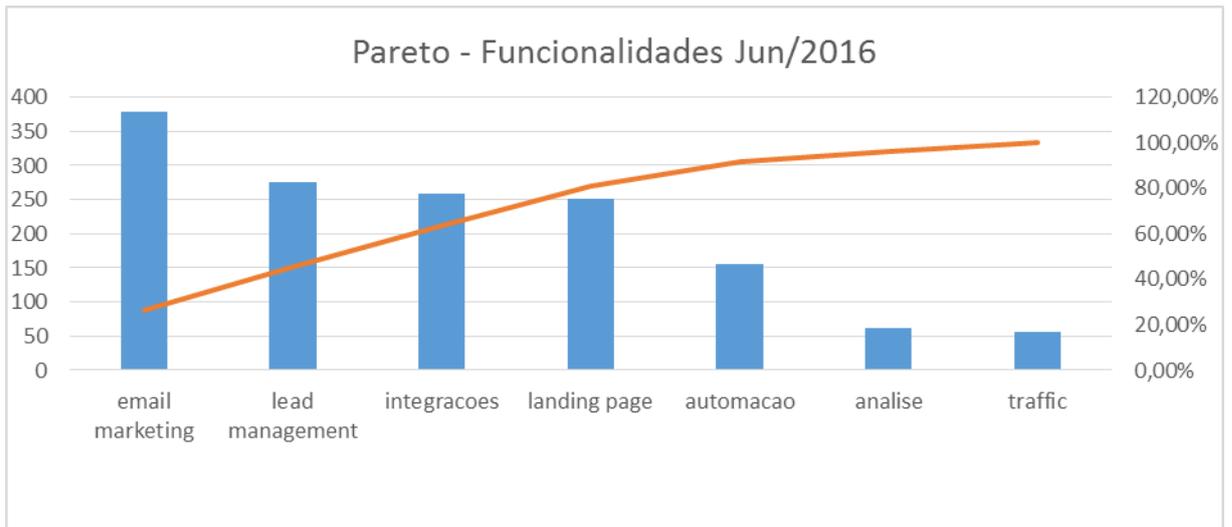


Figura 11 - Diagrama de Pareto dos tickets criados em junho, separados por macro-funcionalidade do RD Station

Capítulo 6: Resultados

Os resultados alcançados com a base de dados são mais “teóricos” que “práticos”, visto que não temos a solução completa do problema de banco de dados implementada, pois ela precisa ser melhor estruturada e ainda tomará tempo para chegar na maturidade desejada (como exposto no Capítulo 4:). Já a proposta de KPIs do suporte, está num estágio de aprovação com os times de produto e suporte da empresa e poderá em seguida ser posta em prática para validação dos resultados.

Dado este cenário, os resultados aqui apresentados são discussões conceituais entre ganhos e perdas das soluções construídas, que servem como prova de conceito para soluções que serão ainda refinadas antes de iniciarem sua implementação real. Assim, são apresentados em ordem: um problema, a solução proposta para ele neste projeto, uma discussão sobre a solução e os próximos passos.

6.1: Definição da métrica, fonte de dados e interface

6.1.1: Solução

Planilha contendo as métricas analisadas atualmente pelos gerentes de produto e os KPIs de adoção, retenção e sucesso de cada funcionalidade.

6.1.2: Discussão

No cenário atual, a planilha serve como centralizadora das informações métricas monitoradas por cada PM, o que é bom para gestão de informação e para apoiar o início de um projeto de migração de arquitetura de informações. Porém as métricas contidas ainda são limitadas à arquitetura de informações atual. Quando migrarmos para uma estrutura mais performática e flexível, é bem provável que surjam outras métricas que devem ser medidas.

6.1.3: Próximos passos

À medida que o projeto proposto (apresentado em 4.2:) avança, deve-se repensar como cada métrica será medida (visto que os registros serão todos de

eventos de usuário, e não mais de registros de software) e quais informações que podem não estar disponíveis hoje poderão ser medidas futuramente.

6.2: Single Source of Truth

6.2.1: Solução

Uma única fonte de dados (registros de eventos), integrada como diversas tabelas, e cada dado é persistido apenas uma vez, em uma tabela.

6.2.2: Discussão

O conceito de SSOT foi validado na nova estrutura de banco de dados proposta. Dois conceitos importantes são observados nela: a existência de um dado específico em apenas um lugar (uma única coluna de uma única tabela) e a facilitação de leitura do critério de um filtro. O exemplo mais claro é a consulta do número de clientes da tabela *accounts*, onde antes era necessário filtrar cinco valores específicos de três colunas diferentes para retirar contas internas, contas de ex-clientes e contas de teste. Toda essa abstração conceitual foi feita durante a migração da base (estratégia de ETL), onde foi criada a coluna *is_client*, que já representa a agregação destes filtros. Novamente vale citar que esta agregação é possível de ser feita apenas neste caso de uso. Internamente, cada uma das três informações aqui abstraídas são valiosas para outras aplicações.

6.2.3: Próximos passos

Primeiramente é necessária a criação da base de dados de eventos e implementação da mesma (inclusão dos códigos para monitorar as ações do usuários). Com essa base que, de fato, conterá o conceito de SSOT, devem ser desenhados os EMRs que transformarão essa base “crua” em *schemes* menores.

6.3: Ineficiência da arquitetura para análise

6.3.1: Solução

A solução desenvolvida neste projeto propõe uma base extremamente simples (11 tabelas), simplificando o entendimento da estrutura dos dados e consequente construção das consultas. Vale ressaltar que ela não contém todos os dados que deveria, visto que foi criada apenas para a prova de conceito, mas sabe-se que, mesmo com todos os dados de todos os times, ela ainda teria um número

bastante reduzido de tabelas (estima-se entre 15 e 20 tabelas) em comparação com a base de operação (cerca de 100 tabelas), o que manteria a simplicidade da mesma.

6.3.2: Discussão

Um ganho importante é o desacoplamento da arquitetura proposta da base de dados que roda o serviço aos clientes, aumentando a performance das consultas dos dois lados (tanto aos clientes quanto à empresa), mas que também impacta no custo de manutenção dos dados e serviços web. Este custo ainda será estimado quando as especificações técnicas das bases e serviços forem feitas.

Ainda falando de eficiência, nesta nova arquitetura são propostas diversas soluções de desacoplamento de tabelas e criação de índices para otimizar as uniões entre tabelas e filtros. O exemplo mais claro é a junção (na arquitetura atual) da tabela de objetos de Landing Page com a tabela de conversões (a segunda maior do banco de dados atual), que é feita pela junção de colunas do tipo *string*. Juntando a tabela com segundo maior número de entradas da base com outra tabela que contém mais de 90 mil registros, cruzando todos estes dados em uma coluna *string* resulta em uma perda gigantesca de eficiência. Na nova arquitetura, a junção de *landing_pages_objects* com *landing_pages_leads* pode ser feita por meio da chave primária da primeira tabela

6.3.3: Próximos passos

Refinar a proposta de *schemes* para cada time a partir das novas métricas que poderão ser medidas com a nova arquitetura.

6.4: Métricas não exploradas

6.4.1: Solução

Apenas o fato de ter uma base mais simples já torna possível obter métricas críticas que antes não eram coletadas (principalmente os KPIs de sucesso de Landing Pages e Tráfego).

Além disso, também foi desenvolvida a nova proposta de métricas e análise do suporte, fonte muito importante de informações que normalmente é pouco utilizada [15].

6.4.2: Discussão

O principal ganho das soluções construídas em relação a novas métricas se encontra nos relatórios de suporte, onde agora existe um novo canal de feedback quantitativo e qualitativo para os gerentes de produto. A proposta ainda precisa ser validada e futuramente refinada, mas apenas o fato de estabelecer um processo para este tipo de análise já traz a esperança de que este canal será cada vez mais utilizado e otimizado, chegando em níveis ótimos como visto em *benchmarkings*.

Na solução da nova arquitetura de dados, o ganho que se espera é a flexibilidade do banco com relações a análises mais profundas e conclusivas, mas que ainda não puderam ser feitas e validadas nesta prova de conceito.

6.4.3: Próximos passos

Implementar um processo de *feedback* contínuo do time de suporte ao time de produto por meio das métricas, análises e relatórios desenvolvidos neste projeto e refinar a proposta ao longo dos próximos meses. Além disso, é possível avaliar a possibilidade de integração da ferramenta de suporte com as bases de dados da empresa, a fim de traçar melhores correlações de dados, conforme sugerido no Capítulo 5:.

6.5: Considerações Finais

De modo geral, o resultado obtido é um maior poder de análise por parte dos gerentes de produto da empresa, desde novas métricas e fontes inexploradas previamente (como o suporte), passando pela acessibilidade dos dados de métricas já conhecidas mas que antes eram problemáticas de serem coletadas, até uma camada acima destas, que representa a confiabilidade dos dados contidos em todas estas métricas, que influencia diretamente na velocidade e assertividade de uma tomada de decisão.

Capítulo 7: Conclusões e Perspectivas

O grande aprendizado que tive durante estes quase seis meses como *Product Manager* me trouxe conhecimentos em horizontes que nunca havia tido contato. Desde a gestão de produto, que foi abordada apenas como introdução deste trabalho, e que se tornará minha primeira profissão agora que finalizo meu estágio, até conhecimentos de serviços web e *big data*.

Estes conhecimentos trouxeram uma clareza maior do tamanho do problema que foi descrito no Anteprojeto deste trabalho, o que inicialmente pensava ter soluções mais simples. Assim, foi necessário repensar o escopo do projeto para que se adequasse a um desafio atingível neste tempo de empresa.

Apesar de não resolver por completo o problema descrito desde o Anteprojeto, o resultado atingido é bastante satisfatório. Seguindo a metodologia de *Lean Startup* (<http://theleanstartup.com/>), de Eric Ries, que a Resultados Digitais adota desde sua fundação, foi possível provar o conceito tanto do problema quanto de uma parte da solução completa que será futuramente desenvolvida para o problema em questão. Este passo é importante pois reduz as incertezas e riscos relacionados a uma solução grande e que leva tempo para ser construída.

A partir de agora, dando sequência ao projeto apresentado no Capítulo 4:, a empresa já tem embasamento prático comprovado em alguns conceitos essenciais para a implementação da solução de BI e tem mais garantia de uma solução que atenda todas as necessidades da empresa e que seja escalável para que no futuro continue suprindo estas necessidades. Além disso, boa parte das tabelas finais que serão produzidas pelos EMR *workers* do projeto já tem seu modelo quase pronto, dado todo o estudo de métricas e KPIs de produto feito e documentado neste projeto, facilitando e agilizando o processo de implementação.

Pensando em aspectos de curto prazo, esta solução já está sendo utilizada por dois times de produto da empresa, auxiliando em análises exploratórias na validação quantitativa de hipóteses. Ainda não é possível realizar a medição das métricas por causa da falta de integração direta da entrada de dados com a nova base que foi implementada, mas como esta base possui dados fiéis de um período

anterior, ela é uma “foto” de como os dados estavam há pouco tempo atrás, sendo perfeitamente utilizável para a validação de hipóteses.

Na parte de KPIs do suporte, acredito que o grande aprendizado ainda está por vir, no momento em que colocarmos a solução em prática (o que não é o caso do primeiro problema, que a empresa já vive na pele). Neste projeto o estudo realizado se mostra interessante para entender o problema e mapear todas as oportunidades que não são aproveitadas nesta nova fonte de informações, mas os impedimentos técnicos e os resultados práticos serão medidos apenas depois de um certo período de implementação, refinamento e adaptação da rotina dos gerentes de produto (inserir esta nova fonte de informações em suas análises técnicas).

Somadas as duas soluções, a empresa espera agora não apenas facilitar o trabalho de um PM, mas também encontrar problemas que os usuários enfrentam com o produto e oportunidades de melhoria de maneira mais simples, rápida e efetiva, otimizando o processo de desenvolvimento de novas funcionalidades e melhorias do *software*.

Bibliografia:

- [1] Saeed Khan. “The Origins of Product Management” part 1, 2, 3. [Online]. Disponível em: <http://onproductmanagement.net/2010/03/08/the-origins-of-product-management-part-1/> - Artigo online.
- [2] Andrea Saez. “The Product Manager’s Ultimate Trio of SaaS Metrics”. [Online]. Disponível em: <https://www.prodpad.com/2015/08/3-important-metrics-saas-product-managers/> - Artigo online.
- [3] Intercom. “Intercom on Product Management”. [Online]. Disponível em: <https://www.intercom.io/books/product-management> - Livro eletrônico.
- [4] Martin Eriksson. “The History and Evolution of Product Management”. [Online]. Disponível em: <http://www.mindtheproduct.com/2015/10/history-evolution-product-management/> - Artigo online.
- [5] Barbara Nelson. “SaaS Product Management”. [Online]. Disponível em: <http://pragmaticmarketing.com//resources/saas-product-management?p=1> - Artigo online.
- [6] Dan Schmidt. “What is Product Management?”. [Online]. Disponível em: <https://www.quora.com/What-is-product-management> - Fórum de discussão online.
- [7] Alicia Shiu. “Thinking about building your own analytics? Don’t.” [Online]. Disponível em: <https://amplitude.com/blog/2016/04/21/thinking-building-analytics-dont/> - Artigo online.
- [8] Brian Halligan e Dharmesh Shah. “Inbound Marketing: seja encontrado usando o Google, a mídia social e os blogs” – Livro físico.
- [9] Vitor Peçanha. “O que é Inbound Marketing?”. [Online]. Disponível em: <http://marketingdeconteudo.com/o-que-e-inbound-marketing/> - Artigo online.
- [10] Resultados Digitais. “Inbound Marketing: Tudo o que você precisa saber”. [Online]. Disponível em: <http://resultadosdigitais.com.br/inbound-marketing/> - Artigo online.

- [11] Unbounce. “What is a Landing Page”. [Online]. Disponível em: <http://unbounce.com/landing-page-articles/what-is-a-landing-page/> - Artigo online.
- [12] Gerald Chait. “Half the money I spend on advertising is wasted; the trouble is I don't know which half”. [Online]. Disponível em: <https://www.b2bmarketing.net/en/resources/blog/half-money-i-spend-advertising-wasted-trouble-i-dont-know-which-half> - Artigo online.
- [13] -. “Single Source of Truth”. [Online]. Disponível em: https://en.wikipedia.org/wiki/Single_source_of_truth - Artigo online.
- [14] Nick Kellingley. “Google’s HEART Framework for Measuring UX”. [Online]. Disponível em: <https://www.interaction-design.org/literature/article/google-s-heart-framework-for-measuring-ux> - Artigo online.
- [15] Heather McCloskey. “What’s the Best Way to Share Customer Feedback with Product?”. [Online]. Disponível em: <https://community.uservoice.com/blog/how-to-share-customer-feedback/> - Artigo e apresentação online.
- [16] Evan Hamilton. “How to Get Product Managers to Listen to Customer Feedback”. [Online]. Disponível em: <https://community.uservoice.com/blog/product-managers-customer-feedback/> - Artigo online.
- [17] Ben Congleton. “All-Hands Support”. [Online]. Disponível em: <https://community.uservoice.com/talks/all-hands-support/> - Apresentação online.
- [18] Kissmetrics. “Customer Analytics”. [Online]. Disponível em: <http://grow.kissmetrics.com/customer-analytics> - Livro eletrônico.
- [19] Janna Bastow. “How Product Managers Can Work With Customer Support”. [Online]. Disponível em: <https://www.prodpad.com/2014/07/how-product-managers-can-work-with-customer-support/> - Artigo online.
- [20] Heather McCloskey. “Building a Feedback Machine”. [Online]. Disponível em: <http://www6.uservoice.com/l/17202/2015-12-08/364xyb> - Livro eletrônico.