

Sistema de apoio a decisões para atendimentos Help Desk

Pool Patrik Oliveira de Oliveira¹
Eurico Jardim Antunes²

Resumo. O presente artigo tem como objetivo apresentar os resultados do desenvolvimento e implementação de um sistema de apoio a decisão voltado para análise de atendimentos Help Desk. O projeto apresenta o uso de ferramentas para transformações *ETL*, desenvolvimento de Dashboards analíticos e construção de um data warehouse consistente, capaz de unificar bancos de dados de diversos sistemas. O produto final apresentado, é capaz de facilitar a análise de registros de atendimentos realizados em softwares distintos, disponibilizando os dados como informações uteis em um único lugar.

Palavras-chaves: Data Warehouse. Sistema Analítico. Pentaho. Painel.

Help Desk decision support system

Abstract. *The present article aims to present the results of the development and implementation of a decision support system for the analysis of Help Desk calls. The project presents the use of tools for ETL transformations, development of analytical dashboards and construction of a consistent datahouse, capable of unifying databases of different systems. The final product presented is able to facilitate the analysis of records of calls made in different software, making the data as useful information in a single place.*

Key-Words: Data Warehouse. Analytical System. Pentaho. Dashboard.

1. INTRODUÇÃO:

Quando se trabalha com um volume expressivo de clientes, o acompanhamento de tratativas bem-sucedidas e fracassadas passa a ser uma tarefa bastante custosa em questão de tempo. Mesmo que muitas vezes o gestor organize suas tarefas a fim de acompanhar os resultados de todos os atendimentos realizados, ainda assim a margem de falha humana será muito grande, e crescerá de acordo com o desenvolvimento da empresa.

Segundo Eleuterio (2015), para uma boa gestão empresarial é fundamental o constante uso de informações sobre o domínio específico, de forma que torne possível prever demandas, identificar falhas, gerar alertas, melhorar o nível de assertividade nas decisões, e a compreensão sobre a importância e impacto que a boa, ou má gestão destas informações pode causar, torna-se um divisor de águas entre o sucesso ou fracasso nos negócios.

¹ Acadêmico do curso de Sistemas de Informação das Faculdades Integradas de Taquara -Faccat, RS. Email: poololiveira@sou.faccat.br

² Professor do curso de Sistemas de Informação das Faculdades Integradas de Taquara – Faccat, RS. Mestre em Ciências da Computação. E-mail: euricoantunes@faccat.br

É evidente que toda empresa de grande, médio ou pequeno porte precisa de uma boa gestão sobre as informações disponíveis. Com o intuito de prover fácil acesso as informações geradas através de atendimentos *Help Desk*, o projeto dissertará sobre a unificação de base de dados *SQL Server* utilizando alguns conceitos de *Data Warehouse*. Espera-se desta forma prover um ambiente para que as informações possam ser extraídas, tratadas e apresentadas de forma adequada.

O presente trabalho tem como objetivo proporcionar formas rápidas para identificar onde a maior parte dos recursos de atendimento *Help Desk* são consumidos, permitindo assim tomadas de decisões estratégicas que permitam reduzir o tempo de atendimento, aprimorar a distribuição de tarefas e distribuição de recurso de acordo com a demanda de atendimentos, minimizar o retrabalho e elevar o nível de satisfação dos clientes.

2. VISÃO GERAL DOS ATENDIMENTOS

A organização em questão, comercializa um leque de *softwares* voltados para a gestão de Ponto Eletrônico e Controle de Acesso por intermédio de cerca de 3000 distribuidores distribuídos em todo o Brasil. Como parte da estratégia de venda, estes revendedores têm direito a suporte gratuito para manutenção de todos os produtos comercializados.

Atualmente cerca de 3500 chamados são atendidos mensalmente em diversas categorias, tais como, comunicação com equipamentos, integrações, revisões de cálculos, controles de refeitórios, entre outras dezenas de categorias de erros e dúvidas. Para gerenciar corretamente todas as categorias de atendimentos, mais de um *software* é utilizado para registro dos chamados, sendo eles, Zimbra e-mail e OctaDesk para chamados em geral, R.Atendimentos (Sistema para registro de atendimentos telefônico, desenvolvido internamente) e Bugtracker.net para controle de erros e customizações.

Todos os softwares utilizados na gestão de atendimentos executam muito bem suas respectivas funções, no entanto, o peso desta divisão de atendimentos é a fragmentação de informação, onde cada uma das ferramentas usa um banco de dados próprio e independente.

Com os dados fragmentados torna-se inviável um levantamento sobre recorrência de problemas, tempo, demanda de recursos. Para ilustrar a deficiência que existe nas métricas de atendimentos, podemos utilizar um caso real. Em uma determinada versão a atualização de um dos sistemas comercializados foi barrada por alguns antivírus comuns no mercado, como Avast, Kaspersky e *Windows Defender*, de forma que o problema ocorreu com diversos clientes diferentes. Em certo momento um dos Atendentes de Suporte passou a pedir para que os clientes

desativassem o antivírus para realizar a atualização, a partir de então a atualização foi realizada com sucesso desde que o antivírus permanece desativado. Esta solução foi documentada e utilizada por vários atendentes, sem o conhecimento do gestor do setor. Dias mais tarde com relatos de insatisfação dos clientes, foi possível tratar o problema corretamente para que o *software* fosse atualizado sem nenhuma restrição. Essa pequena falha no fluxo gerou várias reclamações dos clientes e perda de credibilidade sobre a ferramenta oferecida, problema como estes poderiam ser facilmente identificados em uma plataforma unificada, com a disponibilidade de um *dashboard* para acompanhamento de categorias de problema recorrentes.

2.1. R.Atendimentos

Sistema desenvolvido internamente, direcionado para registros de ligações telefônicas, capaz de armazenar horário de início e fim dos atendimentos, assim como, categorias, nome do cliente e produto relacionado de forma que todos os dados são gravados através de uma tela para interações rápidas conforme figura 1.

Figura 1- R.Atendimentos – Tela de registro de atendimentos

The image shows a web form titled "Registrar atendimento - Pool". It contains the following elements:

- Dados do atendimento:**
 - Protocolo: 184393
 - Técnico: Teete
 - Revenda: Seculum (dropdown menu)
- Vínculos:**
 - Buttons for "Agenda", "Ligação", and "Email".
- Andamento do atendimento:**
 - Link: "Limpar"
 - Date/Time: 11/09/2018 09:00
 - Status: Normal (dropdown menu)
- Conclusão:**
 - Ponto Seculum 4 (dropdown menu)
 - Cadastro Biometria (dropdown menu)
 - Atendimento regular (dropdown menu)
 - Link: "Selecionar mais sistemas"
 - Checkbox: "Acesso remoto"
- Status:**
 - Radio buttons: "Concluído" (selected), "Vai entrar em contato", "Ligação caiu"
 - Link: "Finalizar"

Fonte: Autor.

2.1.1 Relatório/Lista de atendimentos

Como exibido na figura 2, o *software* R.Atendimentos possui um relatório sobre os chamados realizados, o qual traz uma visão geral dos atendimentos, quase como uma tabela de Excel, que nos mostra dados gerais dos registros.

Figura 2. R.Atendimentos - Lista geral de atendimentos

01/08/2018	Data inicial	10/09/2018	Data final	Usuário	ID	Data	Técnico	Revenda	Início	Fim	Descrição	Conclusão	Acesso remoto	Problemático	Ignoramento	Ligação vinculada	Classificação	Emails	Subcategoria	Subcategoriall
				Guilherme...	181558	01/08/20...	viamir	YF TECN...	08:13	08:17	falha ao a...	Concluído	Não	Atendime...			Suporte	Banco de...	17	
				Eduardo ...	181559	01/08/20...	Crislane	Filmaq	08:15	08:34	Dúvidas s...	Concluído	Não	Atendime...			Suporte	Registro	6	
				Dante	181560	01/08/20...	Antonio	Centema...	08:24	08:26	Iniciar os ...	Concluído	Não	Atendime...			suporte	Instalaçã...	3	
				Mauricio ...	181561	01/08/20...	Henrique	Técnica ...	08:27	08:52	duvida de ...	Concluído	Não	Atendime...			Suporte	Cálculos	4	
				Willian	181562	01/08/20...	Patricia	Tecnoville	08:31	08:32	disse que ...	Vai ligar/...	Não	Atendime...			suporte	Verificar e...	14	
				Guilherme...	181563	01/08/20...	eduardo	Control T...	08:33	08:45	Dúvidas l...	Concluído	Não	Atendime...			Suporte	Gerar bac...	1024	
				Duflander...	181564	01/08/20...	Silva	Ebalmaq	08:37	08:39	Iniciar ac...	Concluído	Sim	Atendime...			Suporte	Acesso R...	18	
				Bruno	181565	01/08/20...	Ruan	Thetis	08:39	09:34	Acesso re...	Concluído	Sim	Atendime...			Suporte	Módulo w...	1033	
				Dante	181566	01/08/20...	Breno	Limetech	08:40	08:45	Erro desc...	Concluído	Não	Atendime...			suporte	Comunica...	2	
				Willian	181567	01/08/20...	Macon	Pontec	08:46	09:05	João, em...	Vai ligar/...	Não	Atendime...			suporte	Banco de...	17	
				Gabriel M...	181569	01/08/20...	Anderson	Centemac	08:51	09:36	Erro ao im...	Concluído	Não	Atendime...			Suporte	Erro	2022	
				Guilherme...	181570	01/08/20...	carol	Technop...	08:57	08:58	cau	Ligação c...	Não	Atendime...			Suporte	Ligação C...	3022	
				Mauricio ...	181571	01/08/20...	Mileni	Ata State...	09:09	09:27	problemas...	Concluído	Não	Atendime...			Suporte	Registro	6	
				Dante	181572	01/08/20...	Mileni	Technop...	09:13	09:17	Como ativ...	Concluído	Não	Atendime...			suporte	Backup R...	3049	
				Guilherme...	181573	01/08/20...	diego	Global Inf...	09:17	09:30	dúvidas sd	Concluído	Não	Atendime...			Suporte	Instalaçã...	3	
				Willian	181574	01/08/20...	Ginaldo	Massistec	09:21	10:06	Rinaldo.D...	Vai ligar/...	Não	Atendime...			suporte	Cálculos	4	
				Dante	181575	01/08/20...	Ricardo	Poi Autom...	09:31	09:33	Sobre o ti...	Concluído	Não	Atendime...			suporte	Verificar e...	14	
				Lucas	181576	01/08/20...	Patrick	Gpr Digita...	09:32	09:39	coleta ag...	Concluído	Não	Atendime...			suporte	Gateway	3031	
				Dante	181577	01/08/20...	Ricardo	Poi Autom...	09:35	09:36	Sobre o ti...	Concluído	Não	Atendime...			suporte	Verificar e...	14	
				Mauricio ...	181578	01/08/20...	robson	Integra	09:37	09:41	para rem...	Concluído	Não	Atendime...			Suporte	Registro	6	
				Guilherme...	181579	01/08/20...	viamir	YF TECN...	09:46	09:49	dúvida im...	Concluído	Não	Atendime...			Suporte	Banco de...	17	
				Bruno	181580	01/08/20...	Ayveron	Aw Ponto...	09:55	10:00	o funcion...	Concluído	Não	Atendime...			Suporte	Configurar...	9	

Fonte: Autor

2.1.2 Relatórios/Lista de atendimentos por revendedor

O R.Atendimentos também permite a visualização de um contador de atendimentos agrupado por vendedor, o relatório realiza uma contagem de atendimentos conforme figura 3.

Figura 3. R.Atendimentos - Lista de atendimentos por revendedor

Revenda	Quantidade de atendimentos
Ebalmaq	44
Copemaq	41
Afimaq Com. E Serviços	39
Feira Ponto Ltda	38
Timotec	37
Soltech	35
Centemaq - So	26
Assistec Relogios	24
Sistemporto	24
CHF Tecnologia	23
Centemac	22
Dalmec	22
Abril (Rb Systems)	22
Lanna Variedades	22
Audicom	21
Renatex Automação	21
Rep Acesso	21
Lponto (Setponto)	20
Limetech	20
Cap Informatica	19
Acesso Ponto	19
A Hora Certa Parabana	19
Ciga	19
Control Tech	19
Apolo Copy	18
Gpr Digital - Loja do Ponto	18
Inova Tecnologia	17

Fonte: Autor

2.2 Bugtracker.net

Aplicação para registros de *bugs* gratuita, de código aberto baseado em *interface web*(IFDEFINED, 2018). Além de registros de erros e bugs, a ferramenta foi adaptada para registrar pedidos de alterações nos sistemas comercializados, todos os atendimentos registrados nessa ferramenta possuem descrição da solicitação e *software/projeto* a qual esse pertence.

2.2.1 Lista de bugs registrados

Essa ferramenta possui consultas por listagens de *bugs*, consultas que podem inclusive ser construídas a través de scripts *SQL Server*.

Figura 4. Bugtracker.net - Lista geral de bugs e alterações registrados

sel	id	desc	reported by	reported on	last updated by	last updated on	project	organization	category	priority	assigned to
<input type="checkbox"/>	101458	Contas e editar contas acompanham a tela caso tenha tela de rolagem no Ponto Web	SupRuan	2018-10-22 2:24 PM	Diego	2018-10-23 11:10 AM	Ponto Web	Secullum	Erro	Normal	SupRuan
<input type="checkbox"/>	101457	Setas ficam sobrepostas as opções das batidas no Cartão Ponto do Ponto Web	SupRuan	2018-10-22 2:15 PM	Diego	2018-10-23 11:30 AM	Ponto Web	Secullum	Erro	Normal	SupRuan
<input type="checkbox"/>	101443	Erro "Referência de objeto não definida para uma instância de um objeto" ao concluir as configurações especiais	SupMarcelo	2018-10-11 10:45 AM	SupMarcelo	2018-10-23 3:34 PM	Acesso.NET	Secullum	Erro	Alta	SupMarcelo
<input type="checkbox"/>	101411	Problema na tela de configuração de layout.	SupGabriel	2018-09-27 1:54 PM	GuilhermeA	2018-10-18 11:10 AM	Ponto Web	Secullum	Erro	Normal	SupGabriel
<input type="checkbox"/>	101395	Listagem detalhada e Listagem geral de Lançamento de eventos não calcula o total geral	SupRuan	2018-09-19 12:43 PM	SupRuan	2018-10-09 4:00 PM	Ponto 4	Secullum	Erro	Normal	SupRuan
<input type="checkbox"/>	101354	versão 1.83.1 não tem mais a tela de	SupLincon	2018-08-27 4:40 PM	SupLincon	2018-09-06 8:47 AM	Ponto 4	Secullum	Erro	Normal	SupLincon

Fonte: Autor

2.3 Zimbra e Octa Desk

Até meados de 2018, a empresa foco fazia uso de servidores de e-mail rodando sobre o serviços Zimbra, a aplicação utilizada era bastante semelhante a maioria das ferramentas para gerenciamento de e-mails mais conhecidas como Outlook, Opera Mail ou Thunderbird. A ferramenta realizava um bom gerenciamento de *e-mails*, no entanto, apesar de uma API de integração baseada em Json, ainda era extremamente difícil identificar dados como, assunto, categoria, atendente responsável e data de conclusão do chamado.

No decorrer do levantamento do projeto para este artigo, ficou clara a necessidade de uma ferramenta direcionada para atendimento *Help Desk*, uma vez que mesmo unificando bases de dados e montando um modelo analítico, não seria possível a coleta de informações úteis sobre os atendimentos realizados em um dos canais mais importantes de atendimento, os *e-mails*. Com o acompanhamento da diretoria e área técnica responsável, houve o consentimento para buscas de uma nova ferramenta voltada para atendimentos e otimização de tarefas. A busca consistiu em encontrar uma ferramenta capaz de gerenciar e-mails em forma de chamados, armazenando data e hora de início e fim de cada atendimento, bem como categorias de software e técnico responsável por este chamado.

Softwares como Zendesk, FreshDesk e MoviDesk foram estudados, no entanto devido as funcionalidades oferecidas, custos e o fato de ser uma empresa nacional, o sistema Octadesk mostrou o melhor custo benefício para investimento. Apesar desta ser uma mudança significativa no ambiente da empresa foco, para não desviar o projeto de seu objetivo, não foram levantadas mais informações sobre a ferramenta escolhida no decorrer do artigo.

2.4 Análise manual.

Com as ferramentas citadas, o acompanhamento dos indicadores é realizado de forma desordenada, o gestor precisa gerar várias listas de atendimentos em ferramentas diferentes e em seguida cruzar as informações manualmente em planilhas.

Listas disponíveis:

- Lista de e-mails recebidos em períodos específicos;
- Lista de ligações recebidas em períodos específicos;
- Lista de ligações recebidas por *software* em períodos específicos;
- Lista de ligações recebidas por categoria de atendimento;

Para gerar todas estas listas e agrupá-las em planilhas, o gestor precisa em média de 3 horas para organizar tudo e gerar informações úteis para a equipe. Devido a isso, os relatórios são gerados de forma mensal e em alguns casos sequer são montados devido a demanda de tempo.

3. SISTEMA ANALÍTICO PROPOSTO PARA ATENDIMENTOS DE SUPORTE

No cenário descrito anteriormente, fica evidente que o levantamento de informações úteis é precário e trabalhoso, a tomada de decisões torna-se tardia, atuando sob informações defasadas, na maioria das vezes os problemas chegam a níveis elevados de insatisfação do cliente final por falta de ferramentas ágeis que auxiliem na tomada de decisões. O gestor do departamento de Suporte Técnico, precisa garantir que todos os clientes estão sendo atendido da melhor forma possível, tomando medidas preventivas baseadas nas informações disponíveis.

Para atender essa demanda, propõe-se um sistema analítico, uma maneira de apresentar informação facilmente interpretável por um tomador de decisão (Turban et al, 2009). O sistema proposto reúne informações dos softwares R.Atendimento e Bugtracker.net, possibilitando visualizações em forma de *dashboards* sobre os atendimentos realizados, erros reportados e customizações de *software*. O sistema trabalha com uma tabela para tratamento de dados, a qual

permite importação de dados de diversos outros sistemas, possibilitando o crescimento da ferramenta conforme necessidade e disponibilidade de dados.

Como o *software* Octadesk encontra-se em período de implementação, este não foi integrado ao modelo analítico proposto, pois dependendo das configurações utilizadas na ferramenta, os campos do banco de dados podem variar, retornando informações não úteis para o projeto. Ainda assim, entende-se que, se faz necessário o início de esforços para implantação de uma ferramenta analítica sobre o domínio da empresa e sobre seus serviços de forma rápida, clara e objetiva, proporcionando um ambiente capaz de incorporar dados de qualquer outro sistema no futuro. O uso adequado da informação tem como objetivo proporcionar melhor controle em termos de previsibilidade, consistência, certeza e inclusive a gestão de pessoas, recursos e outros fatores importantes que possam afetar o relacionamento com os clientes (PRATES, 2013).

3.1 Dashboards

Conforme Vieira (2017), o uso de *dashboards* em meio a realidade empresarial é um método eficaz para acompanhamento de resultados e disseminação de informações para acompanhamento do fluxo geral de demandas de trabalho.

Para atender a necessidade da empresa, serão criados dois *dashboards*, contemplando os seguintes itens:

- Quantia de atendimento no período
- Tempo gasto por técnico de acordo com a categoria dos atendimento e período
- Quantia de atendimentos categorizados como erros ou bugs
- Total de customizações solicitadas por período.
- Comparativo de quantia de demanda de atendimentos entre dois períodos.

Com o desenvolvimento destes *dashboards*, espera-se ir de encontro com a proposta de Eleuterio, o qual esclarece que, além de apoiar as tomadas de decisão, a disseminação de informações, de forma concreta podem colaborar com a formação de equipes consistentes e bem integradas, servindo como um fator de motivação para todos os colaboradores (Eleuterio,2015).

Em suas pesquisas Eckerson (2003) descreve diversos benefícios ao proporcionar uma visão unificada sobre o ambiente de trabalho como um todo, entre elas se destacam economia de tempo, elaboração de estratégias de negócio, precisão sob análise de informações e

assertividade na tomada de decisões táticas. Itens que podem ser atendidos com o uso do painel proposto.

4. FERMENTAS E TECNOLOGIAS

4.1 Pentaho Data Integration (PDI)

O Pentaho *Data Integration* é o componente da suíte Pentaho usado para criar processos de extração, transformação e carga. Com o Pentaho *Data Integration* é possível fazer inúmeras operações de Integração de Dados, como por exemplo, Migração de dados, Movimentação de grandes volumes de dados, Transformação de dados, Limpeza de dados, Conformidade de dados (PENTAHO 2016).

O Pentaho *Data Integration* foi eleito como ferramenta para realizar a coleta, gerenciamento e persistência dos dados, a ferramenta foi escolhida levando em consideração seu baixo custo para implementação, enquanto outras ferramentas como Serviços de Integração do *SQL Server* e *IBM InfoSphere DataStage* necessitam de altos investimentos, o Pentaho possui uma versão gratuita, chamada de *Community Edition*, a qual disponibiliza todos os recursos existentes na versão comercial. Além de possuir licença gratuita e de código aberto, o Pentaho *Data Integration* possui *interface* simples, com diversos componentes que minimizam ao máximo o desenvolvimento com linhas de programação. Outro ponto importante na escolha da ferramenta foi a vasta comunidade, com infindáveis vídeos, tutoriais e manuais disponíveis na internet.

Na implementação apresentada no artigo, o PDI (Pentaho *Data Integration*) foi utilizado para construir diversos componentes, que chamaremos de “Transformações”, executando inúmeras tarefas para transformação de dados, com auxílio de algumas linhas de código em linguagem *SQL*, os quais servem como complemento na criação de componentes específicos dentro da ferramenta.

4.2 Pentaho BI Services

Ferramenta utilizada nas primeiras versões entregues para os usuários. O Pentaho *BI Server* é responsável por toda a camada servidora da plataforma do Pentaho, responsável por fornecer a arquitetura e infraestrutura necessária para a construção de soluções para *business intelligence*, como os *dashboards* utilizados neste projeto (PENTAHO. 2016).

4.3 SQL Server 2016

O *SQL Server* 2016 conta com diversas funcionalidades que garantem a segurança e persistência dos dados, gerenciando o acesso, a persistência, a manipulação e a organização dos dados de forma efetiva, disponibilizando inclusive um *interface* fácil e amigável através do *Studio SQL*(Microsoft, 2018).

Além das funcionalidades disponíveis no SGBD escolhido, levou-se em consideração a disponibilidade de servidores já implementados na organização e chaves de registro já adquiridas para trabalhar com *SQL Server*, estando disponível tanto em ambiente local quando em ambientes *Web*, como o *Azure*.

4.4 ClicData

Plataforma de *Business Intelligence* em nuvem, incluindo um criador de painéis de negócios, ferramentas extensivas e integração com diversos tipos de arquivos e base de dados (CLICDATA, 2018).

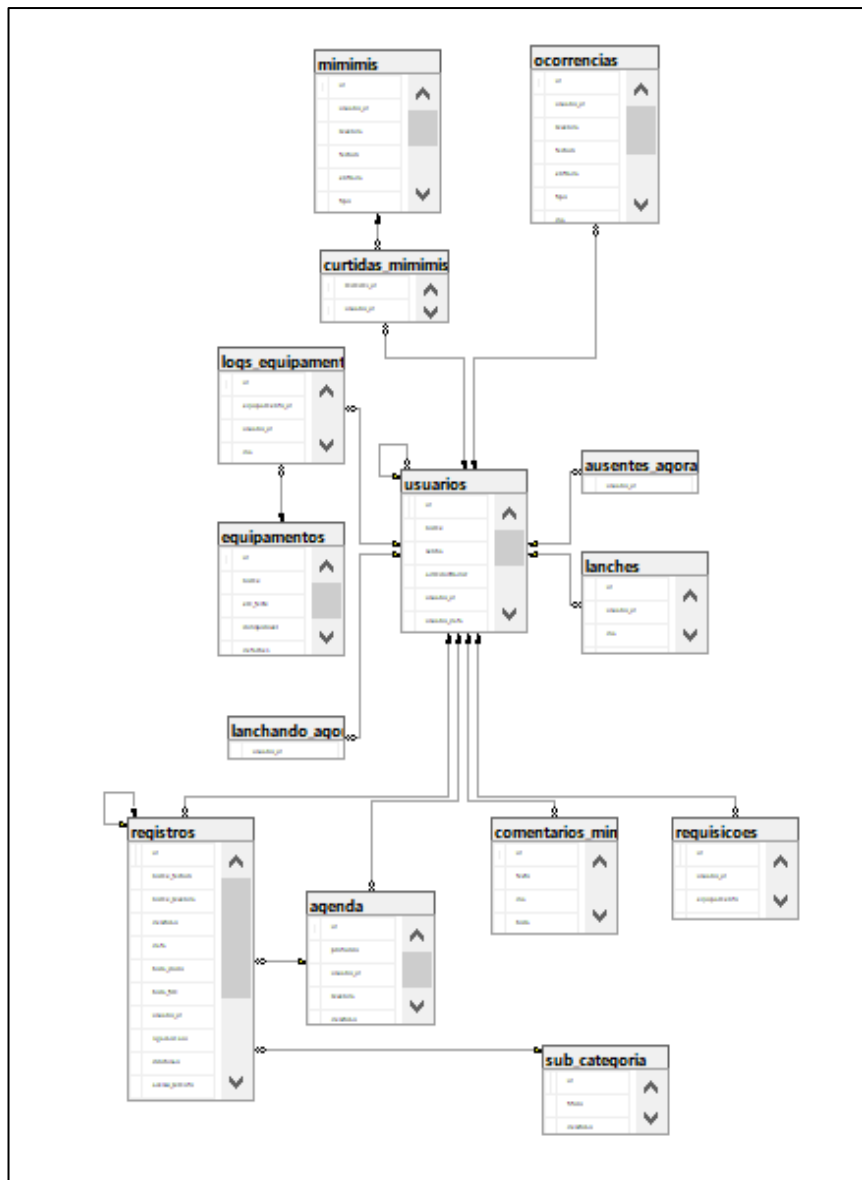
Dadas algumas limitações encontradas para uso da ferramenta Pentaho *BI Server* nos servidores testados, como, alto consumo de memória *RAM* e processamento, buscou-se uma alternativa que pudesse entregar indicadores de forma amigável, atribuindo recursos de usabilidade, além de consumir menos recursos dos servidores.

5. BANCO DE DADOS

5.1 Modelo ER R.Atendimentos

O banco de dados do R.Atendimentos, possui uma estrutura e normalizada com diversas tabelas vinculadas a tabela de usuários, entre outras tabelas utilizada para gravar dados que não dependem de usuários.

Figura 5. Modelo ER R.Atendimentos

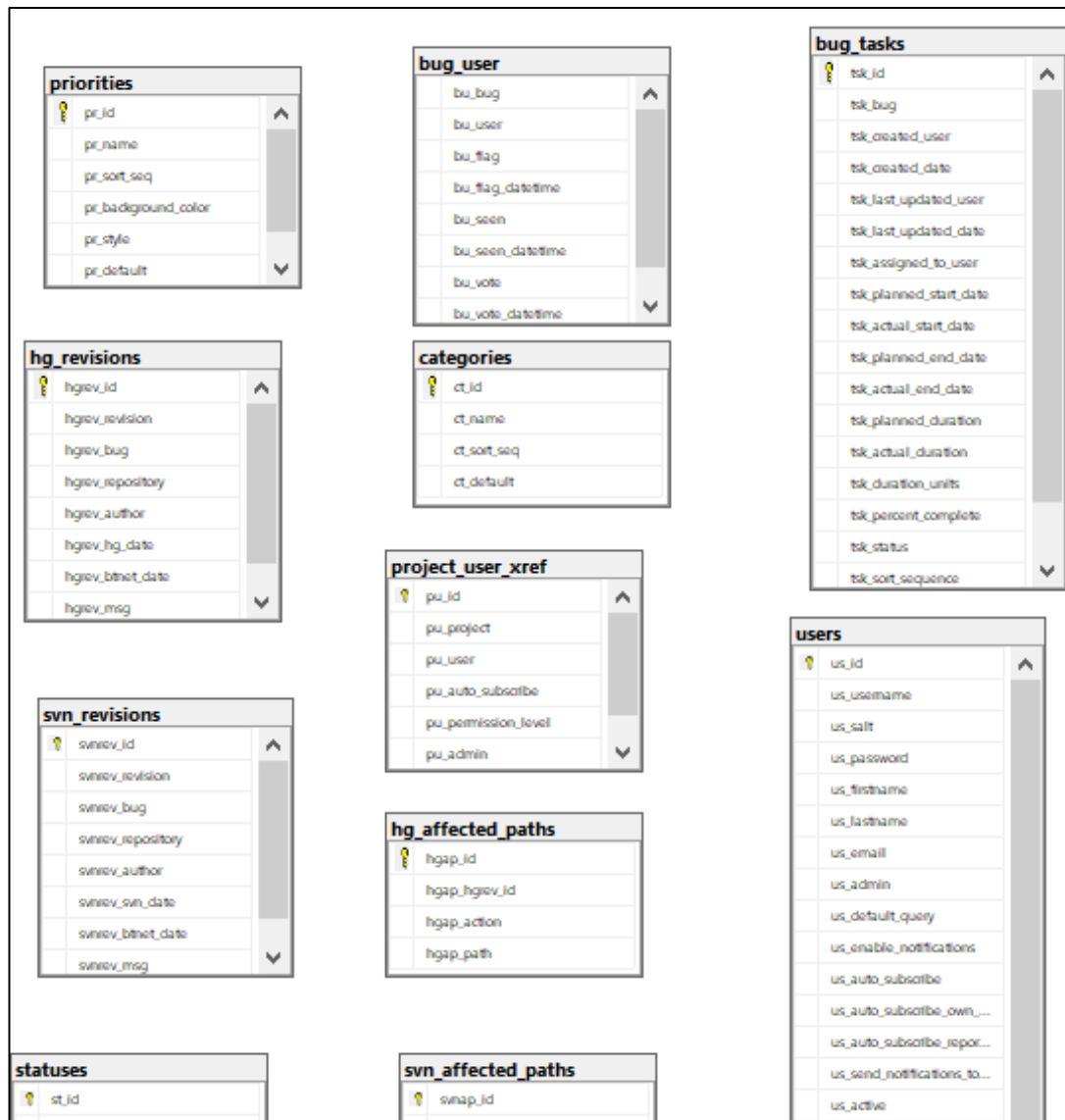


Fonte: Autor

5.2 Modelo ER Bugtracker.net

O Bugtracker.net por ser uma ferramenta mais antiga, possui um banco de dados mais complexo, suas tabelas não possuem relacionamentos, no caso da figura 6, podemos observar que campos de diversa tabelas referenciam a tabela “users”, no entanto não há um relacionamento através de chaves estrangeiras.

Figura 6. Modelo ER Bugtracker.net



Fonte: Autor

5.3 Data Warehouse

Conforme Kimball (1998), um *Data Warehouse* é um banco de dados relacional, que é modelado para facilitar as consultas, permitindo ao usuário visualizar uma grande massa de dados sob diferentes ângulos.

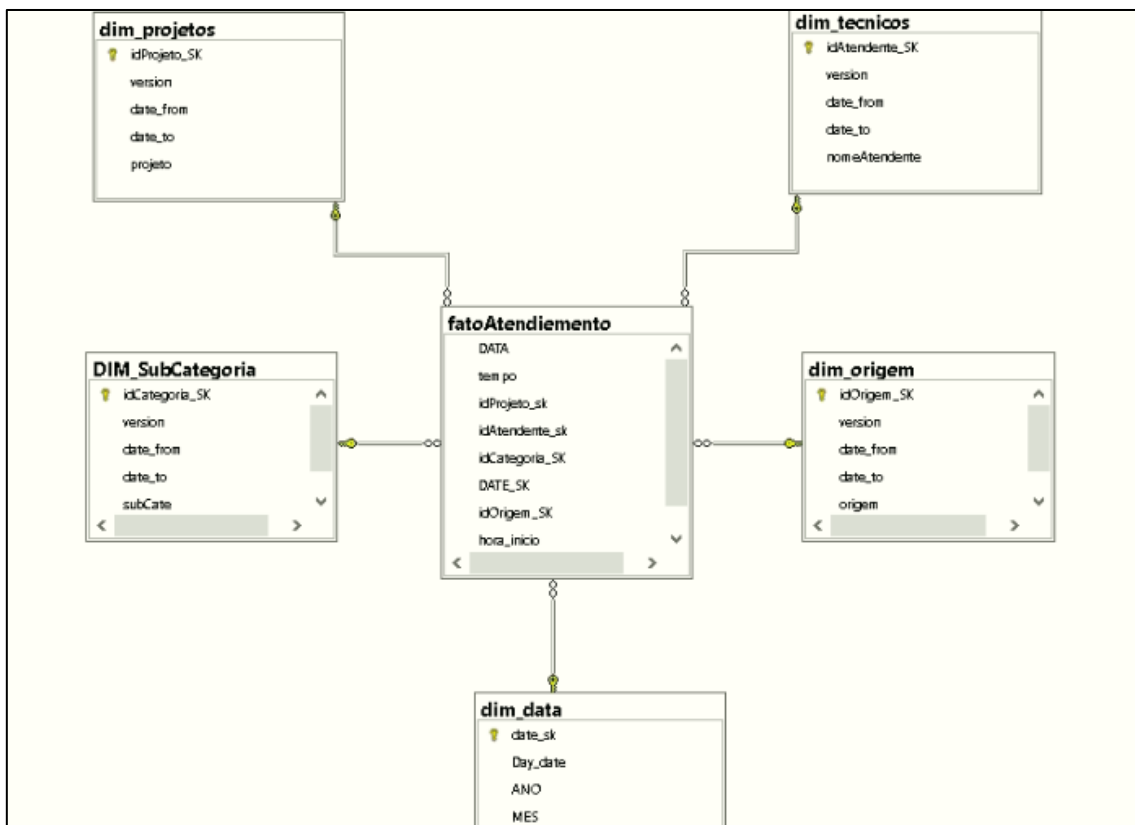
Para criar uma base de dados com registros históricos, inalteráveis pelos processos transacionais, optou-se então por utilizar a modelagem estrela ou *Star Schema*, conforme Machado explica em seu livro (MACHADO, 2013). Para trabalhar com este modelo, é necessário utilizar o conceito de dimensões e fatos, onde dimensões são tabelas que armazenam dados sobre os atendimentos realizados, como por exemplo, nome do técnico, dia do

atendimento e projeto envolvido, os dados contidos nessas dimensões servem como chaves estrangeiras para a tabela fato. Por sua vez a tabela fato contém apenas ligação entre todas as dimensões, ligações realizadas através de chaves estrangeiras, permitindo que o fato “Atendimento” possa ser exibido de várias perspectivas, assim como, tempo, responsável e assunto.

O Modelo proposto conta com 5 dimensões, e uma única tabela fato para relacionamento dos registros, conforme diagrama da figura 7. De acordo com Kimball (1998), dimensões podem ser definidas como dados que compõem um fato, e para escolher quais informações devem compor uma dimensão, é importante que a informação seja útil para responder algumas questões, como por exemplo, “quem atendeu?”, “quando atendeu?”, “porque atendeu?”.

Podemos dizer então que, a modelo estrela é composto de uma tabela fato que será composta de chaves estrangeiras de todas as dimensões criadas, e que as tabelas de dimensões são os dados que compõem uma perspectiva sobre a visualização da informação.

Figura 7. Modelo ER Data Warehouse



Fonte: Autor

5.4 Tabela de tratamento de dados

A tabela de tratamento de dados, é a tabela que recebe dados das bases transacionais, neste projeto a tabela recebe dados de bases de dois sistemas, o R.Atendimentos e o Bugtracker.net, no entanto, como o intuito desta tabela é tratar as informações antes que estas sejam alocadas no *data warehouse*, a tabela permite a inserção de inúmeras fontes de dados, possibilitando o crescimento do sistema analítico conforme amadurecimento do produto.

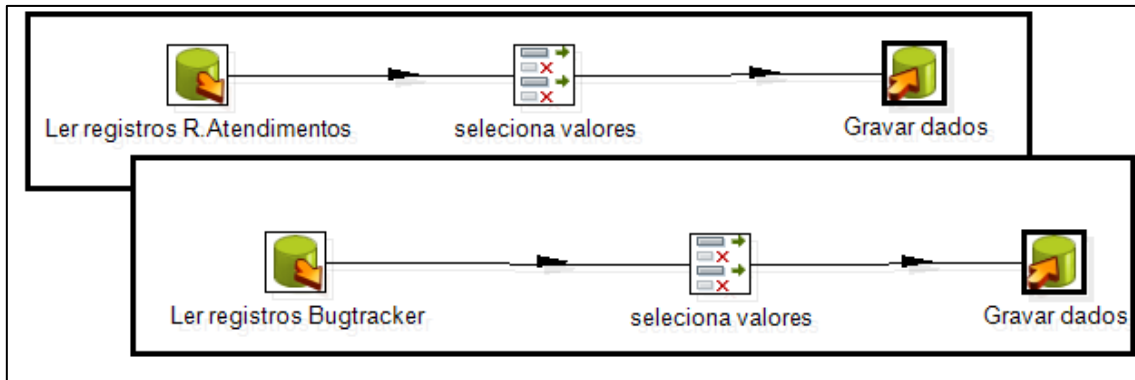
A tabela de tratamento de dados é trabalhada em duas partes, com as Transformações “Ler dados” e “Trata dados”, as quais realizam o processo *ETL* (*Extract, Transform and Load*) ou Extração, Transformação e Carga dos Dados de bases distintas, para uma base de dados centralizada, ou mais de uma quando necessário. No caso deste projeto, os dados são carregados em uma tabela no *data warehouse*, com o objetivo de tornar a carga *ETL* menos custosa em questões de consumos de recursos de processamento, tempo para implementação e manutenção. Tendo em vista que em grande parte das implementações de *data warehouse*, o processo *ETL* pode consumir cerca de 80% dos esforços empregados (INMON, 1997). Espera-se que a rotina criada para tratamento de dados facilite manutenções necessárias, pois estará abstraída dos demais componentes.

A transformação “Ler dados” (Figura 8), busca registros nos bancos transacionais e os armazena na tabela “Tratamento”. Na figura 9, podemos identificar alguns componentes que realizam o tratamento de dados, neste momento é determinado qual o último registro lido nos respectivos bancos de dados, assim como, executa o tratamento e exclusão de informações não úteis. O componente “Trata dados” roda diversos *updates* na tabela de tratamento, removendo campos nulos, definindo e ajustando valores incorretos nos registros, tudo com a finalidade preparar a tabela para importação no *data warehouse*. Ainda na figura 9, há um componente com o nome “Guardar último registro” esse componente guarda o último registro lido de cada uma das fontes de dados, evitando que o mesmo registro seja lido em cargas futura, o que reduz o consumo de recursos do servidor e tempo de execução. Este processo foi pensando para que a quantidade de dados processadas seja menor, tornando então a atualização do *data warehouse* mais rápida.

É importante ressaltar que, a principal função desta transformação é garantir o menor número de registros nulos possível, como no *data warehouse* todos os registros são lidos e comparados antes de serem gravados, a existência de campos nulos pode gerar demora excessiva no processo. Para tratar os dados vindos do R.Atendimentos e Bugtracker.net, não houve necessidade de exclusão de registros, os dados são apenas tratados com *updates* para que

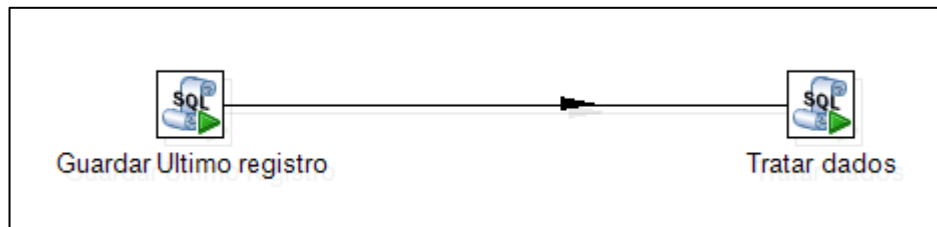
possam ser carregados no *data warehouse*. Na figura 10 podemos ver alguns exemplos de *updates* que são executados.

Figura 8. Transformação Ler Registros



Fonte: Autor

Figura 9. Transformação para tratamento de dados



Fonte: Autor

Figura 10. Exemplos de scripts usado para tratar dados

```
update tratamento set subCate = 'Nao Identificado' where subCate is null
update tratamento set origem = 'bugtracker' where id_bug is not null
update tratamento set origem = 'rAtendimentos' where id_registro is not null
update tratamento set hora_inicio = '00:01' where hora_inicio is null and origem = 'buftracker'
update tratamento set hora_fim = '00:01' where hora_fim is null and origem = 'buftracker'
update tratamento set tempo = 0 where tempo is null and origem = 'buftracker'
```

Fonte: Autor

5.5 Dimensões

Para criar as tabelas das dimensões, é executado um *scripts SQL*, o qual cria os campos das tabelas e também dois índices, um deles utilizado para comparar o nome dos dados já importados e outra para controle da chave primária, abaixo script da criação da dimensão projetos, que pode ser utilizado como exemplo para criação das demais dimensões (figura 11).

Figura 11. Exemplos de scripts usado para criar tabelas de dimensões

```

-----CRIAR TABELA dim_projetos-----
CREATE TABLE dim_projetos
(
    idprojeto_sk BIGINT PRIMARY KEY,
    version      INT,
    date_from   DATETIME,
    date_to     DATETIME,
    projeto     VARCHAR(50));
GO
CREATE INDEX idx_dim_projetos_lookup
ON dim_projetos ( projeto );
Go
CREATE INDEX idx_dim_projetos_tk
ON dim_projetos ( idprojeto_sk );
GO

```

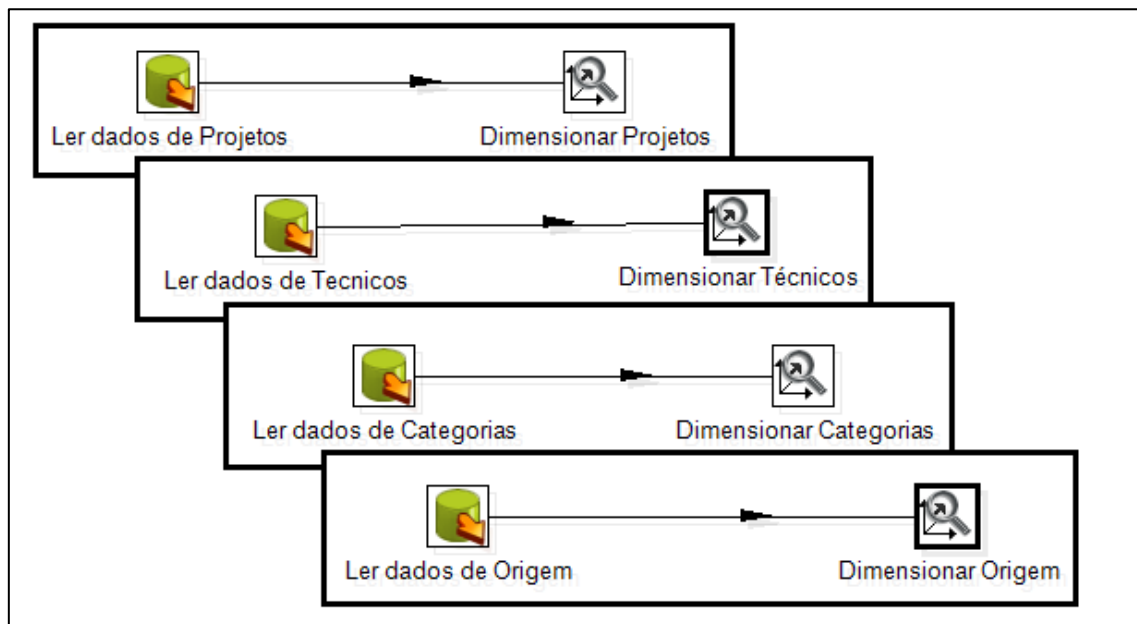
Fonte: Autor

5.5.1 Popular dimensões

Após criar as dimensões do *data warehouse*, foram criadas transformações no *PDI* (*Pentaho Data Integration*), que fazem a leitura da tabela de tratamento de dados mencionada na seção 5.4, conforme figura 12 que ilustra 4 transformações utilizadas. A cada carga de dados para o *data warehouse*, todas as 4 transformações são executadas, lendo os dados e comparando se já existe alguma referência para eles nas suas respectivas dimensões.

A carga dos dados nas tabelas de dimensões é realizada após todo o processo *ETL* visto na tabela de tratamento, desta forma evita-se que o mesmo dado seja carregado mais de uma vez na carga de cada uma das dimensões. Utilizando a tabela de tratamento de dados, as informações são buscadas em bases externas uma única vez, assim como, são tratadas uma única vez, reduzindo a quantia de dados trafegados, processamento utilizado e tempo necessário para realizar a atualização das dimensões.

Figura 12. Transformações para dimensões



Fonte: Autor

Componentes Ler Dados (*Input table*): É o tipo de componente de entrada de dados, responsável pela leitura dos dados que estão disponíveis na tabela “Tratamento”. Este componente tem em sua configuração o comando *SQL* “*select * from tratamento*” realizando a leitura de todos os registros da tabela.

Componentes Dimensionar (*Lookup Dimension*): Componente para atualização e inserção de valores nas tabelas de dimensões. Este componente cria a chave primária da dimensão, filtra os campos que serão gravados, compara os dados da tabela e cria versionamento dos registros.

Ao executar estas transformações, as tabelas de dimensões estarão carregadas com os dados disponíveis na tabela tratamento.

5.5.2. Dimensão Data

A dimensão de data, possui um tratamento diferente das demais dimensões. Como esta tabela é carregada apenas na criação do *data warehouse*, foi criado um *script* que cria a tabela e carrega datas de 2008 até 2030, desta forma o sistema carregará dados históricos entre estas datas, contando que até 2030 existirão diversas manutenções no banco ou até mesmo a construção de um novo Data Warehouse. Para criar os registros da tabela, foram criados

diversas variáveis, entre elas, Data início e Data fim, as quais são utilizadas para definir o intervalo de datas serão geradas, os demais campos da tabela são utilizados para gravar dias do mês, feriado dias da semana e demais dados que podem ser utilizados para visualizações da dimensão data.

Figura 13. Variáveis utilizada para definir registros a serem gravados

```

declare @dataInicial date, @dataFinal date, @data date,
        @ano smallint, @mes smallint, @dia smallint,
        @diaSemana smallint, @diaUtil char(1), @fimSemana char(1),
        @feriado char(1), @preFeriado char(1), @posFeriado char(1),
        @nomeFeriado varchar(30), @nomeDiaSemana varchar(15),
        @nomeDiaSemanaAbrev char(3), @nomeMes varchar(15),
        @nomeMesAbrev char(3), @bimestre smallint, @trimestre smallint,
        @nrSemanaMes smallint, @estacaoAno varchar(15),
        @dataPorExtenso varchar(50)

--informe aqui o período para o qual deseja criar os dados
set @dataInicial = '01/01/2008'
set @dataFinal = '31/12/2030'

while @dataInicial <= @dataFinal
begin
    set @data = @dataInicial
    set @ano = year(@data)
    set @mes = month(@data)
    set @dia = day(@data)
    set @diaSemana = datepart(weekday,@data)

    if @diaSemana in (1,7)
    set @fimSemana = 'S'
    else set @fimSemana = 'N'

```

Fonte: Autor

Figura 14. Exemplo de script utilizado para definir meses do ano

```

set @nomeMes = case
when @mes = 1 then 'janeiro'
when @mes = 2 then 'fevereiro'
when @mes = 3 then 'março'
when @mes = 4 then 'abril'
when @mes = 5 then 'maio'
when @mes = 6 then 'junho'
when @mes = 7 then 'julho'
when @mes = 8 then 'agosto'
when @mes = 9 then 'setembro'
when @mes = 10 then 'outubro'
when @mes = 11 then 'novembro'
when @mes = 12 then 'dezembro' end

```

Fonte: Autor

5.6 Tabela FATO

Para criar a tabela fato, foi criado um *script* bastante simples. O *script* cria a tabela fatoAtendimento, contendo chaves estrangeiras de todas as dimensões e uma coluna chamada tempo, que armazena o tempo de cada registro de atendimento. O tempo de atendimento é calculado através de uma função criado no banco de dados que calcula a diferença entre o horário de início do atendimento e horário de finalização, conforme figura 16. Na figura 17, podemos visualizar os componentes utilizados para criar a transformação que trata essa tabela.

Figura 15. Script de criação da tabela fato

```
CREATE TABLE fatoAtendimento
(
  "DATA" DATE
, tempo DECIMAL(5,2)
, idProjeto_sk bigint FOREIGN KEY REFERENCES dim_projetos(idProjeto_sk)
, idAtendente_sk bigint FOREIGN KEY REFERENCES dim_tecnicos(idAtendente_sk)
, idCategoria_SK bigint FOREIGN KEY REFERENCES dim_subcategoria(idCategoria_sk)
, DATE_SK int FOREIGN KEY REFERENCES im_data(DATE_SK)
, idOrigem_SK bigint FOREIGN KEY REFERENCES dim_origem(idorigem_sk)
, hora_inicio VARCHAR(5)
)
;
```

Fonte: Autor

Figura 16. Script de criação da função tempo

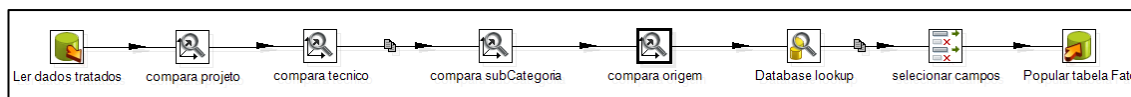
```
CREATE FUNCTION [dbo].[tempo](@hora1 varchar(5),@hora2 varchar(5) )
RETURNS VARCHAR(1000)
BEGIN
declare @horas1 int, @minutos1 int, @horas2 int, @minutos2 int

set @horas1 = (substring (@hora1, 1, 2)) -- horas
set @minutos1= (substring (@hora1, 4, 2)) -- minutos
set @horas2 = (substring (@hora2, 1, 2)) -- horas
set @minutos2= (substring (@hora2, 4, 2)) -- minutos

RETURN( ((@horas2 * 60) + @minutos2) - ((@horas1 * 60) + @minutos1))
END
GO
```

Fonte: Autor

Figura 17. Transformação Popular Fato



Fonte: Autor

Ler Dados (*Input Table*): Componente responsável por ler os dados da tabela de tratamentos.

Compara (projeto, técnico, subCategoria, origem): este é o mesmo componente utilizado nas transformações das dimensões, o *Lookup Dimension*, porém nesta transformação estes módulos são responsáveis por ler os campos da tabela de tratamento e comparar com o *index* das tabelas de dimensões já criadas.

Database lookup: Utilizado para comparar as referências de datas entre a tabela fato e a dimensão Data.

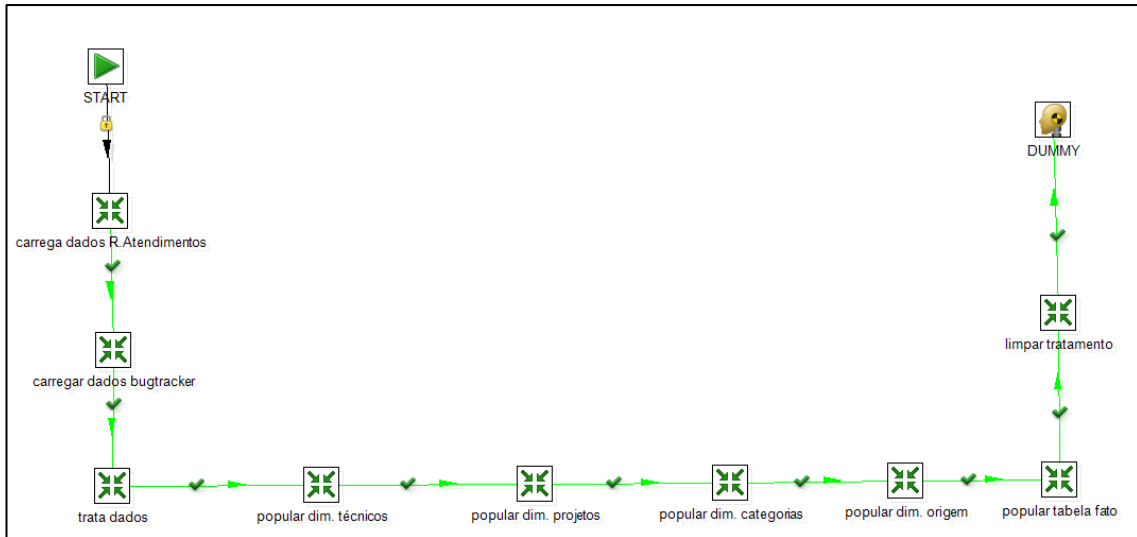
Selecionar dados: Funciona de forma bem simples, é apenas uma seleção de quais campos serão salvos na tabela fato.

Popular tabela fato (*Output table*): Componente responsável por gravar os dados finais na tabela fato.

6. CONTROLE DE CARGA

Neste projeto, o controle de carga é responsável por ler dados das bases transacionais, popular a tabela de tratamento de dados, tratar os dados desta tabela, popular as dimensões, popular a tabela fatoAtendimento e por fim, limpar a tabela de tratamento de dados. Todas estas etapas são reunidas em um processo chamado de *Job* dentro do *PDI*, este processo é responsável por executar cada uma das transformações existentes (figura 18).

Figura 18. Job - Controle de carga



Fonte: Autor

7. AUTOMATIZAÇÃO

Para que todas as transformações sejam executadas automaticamente de forma periódica, é necessário salvar o arquivo do *Job* configurado, em seguida criar um *script* que execute o *Job* criado através do *Kitchen*, programa que pode executar tarefas projetadas no *PDI* (PENTAHO, 2016). Na figura 19, é possível visualizar o *script* utilizado, onde nas últimas 4 linhas são habilitadas configurações para gravar logs de execução do *Job*. Após a execução deste *script* o *data warehouse* estará atualizado com os dados dos bancos de dados transacionais.

Figura 19. Bat - Script para execução do Job

```

1 @echo off
2
3 TITLE ExecutarJob
4 SET currentdir=%~dp0
5 SET kitchen=E:\pentaho\biserver-ce-4.8.0-stable\data-integration\Kitchen.bat
6 SET logfile="%currentdir%log.txt"
7
8 echo. >> %logfile%
9 echo. >> %logfile%
10
11 "%kitchen%" /file:"%currentdir%job.kjb" /level:Basic >> %logfile%
```

Fonte: Autor

Figura 20. Exemplo de log da execução do JOB

```

DEBUG: Using JAVA_HOME
DEBUG: _PENTAHO_JAVA_HOME=C:\Program Files\Java\java-se-7u75-ri
DEBUG: _PENTAHO_JAVA=C:\Program Files\Java\java-se-7u75-ri\bin\java.exe
INFO 03-11 13:46:59,651 - Kitchen - Logging is at level : Log básico
INFO 03-11 13:46:59,651 - Kitchen - Start of run.
INFO 03-11 13:46:59,949 - job - Início da execução do job
INFO 03-11 13:46:59,955 - job - Starting entry [carrega dados R.Atendimentos]
INFO 03-11 13:46:59,960 - carrega dados R.Atendimentos - Loading transformation from XML file
[file:///E:/pentaho/Nova pasta/trasformações/testes.ktr]
INFO 03-11 13:47:00,320 - Ler dados R.Atendimentos - Expedindo início para transformação [Ler dados
R.Atendimentos]
.
.
.
.
.
.
.
|
INFO 03-11 13:48:36,688 - job - Job execution finished
INFO 03-11 13:48:36,689 - Kitchen - Finished!
INFO 03-11 13:48:36,689 - Kitchen - Start=2018/11/03 13:46:59.651, Stop=2018/11/03 13:48:36.689
INFO 03-11 13:48:36,689 - Kitchen - Processing ended after 1 minutes and 37 seconds (97 seconds total).

```

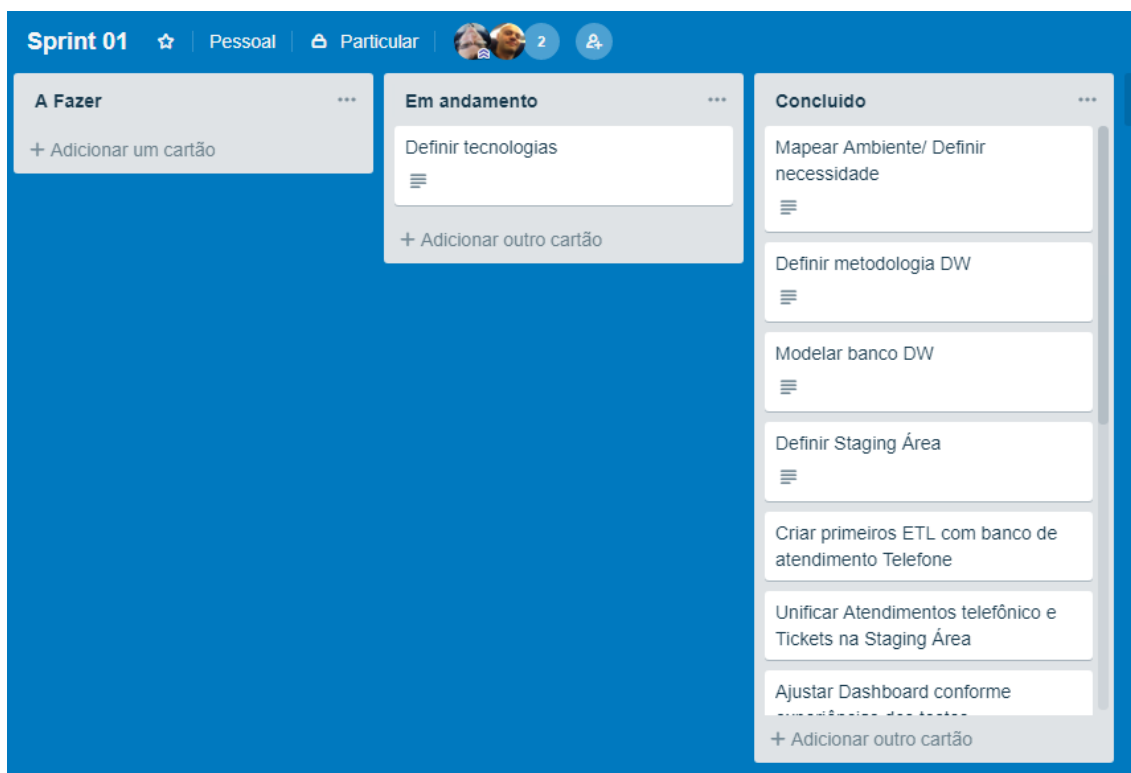
Fonte: Autor

8. METODOLOGIA

A fim de desenvolver uma ferramenta eficaz, foi necessário um acompanhamento muito próximo de alguns usuários, que puderam determinar como a informação poderia ser apresentada de forma mais útil, então a cada incremento, houve um curto período de avaliação. Assumindo características de um processo incremental, o projeto foi dividido em *sprints*, sendo o primeiro o ciclo mais longo, com 30 dias de duração, voltado para o estudo das bases de dados existentes, modelagem de dados e construção do *backend* responsável por todo o tratamento dos dados, sem gerar entregas.

Após fim do primeiro ciclo, passou-se a realizar entregas mais ágeis, realizadas a cada 10 dias, apresentado incremento de funcionalidades já utilizáveis pelo usuário, dando então abertura para validação sobre as funcionalidades implementadas.

Cada ciclo foi controlado por um quadro de *KanBan* específico para a interação planejada, os quadros servem de guia para conclusão de todas as tarefas necessárias (figura 21). Conforme BOEG (2012) o quadro de *Kanban*, é uma das principais ferramentas para auxílio em metodologias ágeis de desenvolvimento, controlado facilmente e de forma visual a tarefas pendentes, em andamento e realizadas, onde cada tarefa é associada a um cartão que se move entre as colunas do quadro conforme andamento das tarefas.

Figura 21. Kanban - primeiro ciclo

Fonte: Autor

9. ENTREGAS REALIZADAS

Entre as versões 1.0 e 1.5, o usuário interagiu com a *interface* gera através do Pentaho *BI Server*, manipulando relatórios analíticos (figura 22) e *dashboards* (figura 23). No entanto, o Pentaho *BI Server* demonstrou a necessidade de recursos excessivos de memória *RAM*, necessitando de cerca de 3GB de memória dedicada, quando em execução prolongada. Desta forma iniciou-se estudos com a plataforma Clicdata, como alternativa.

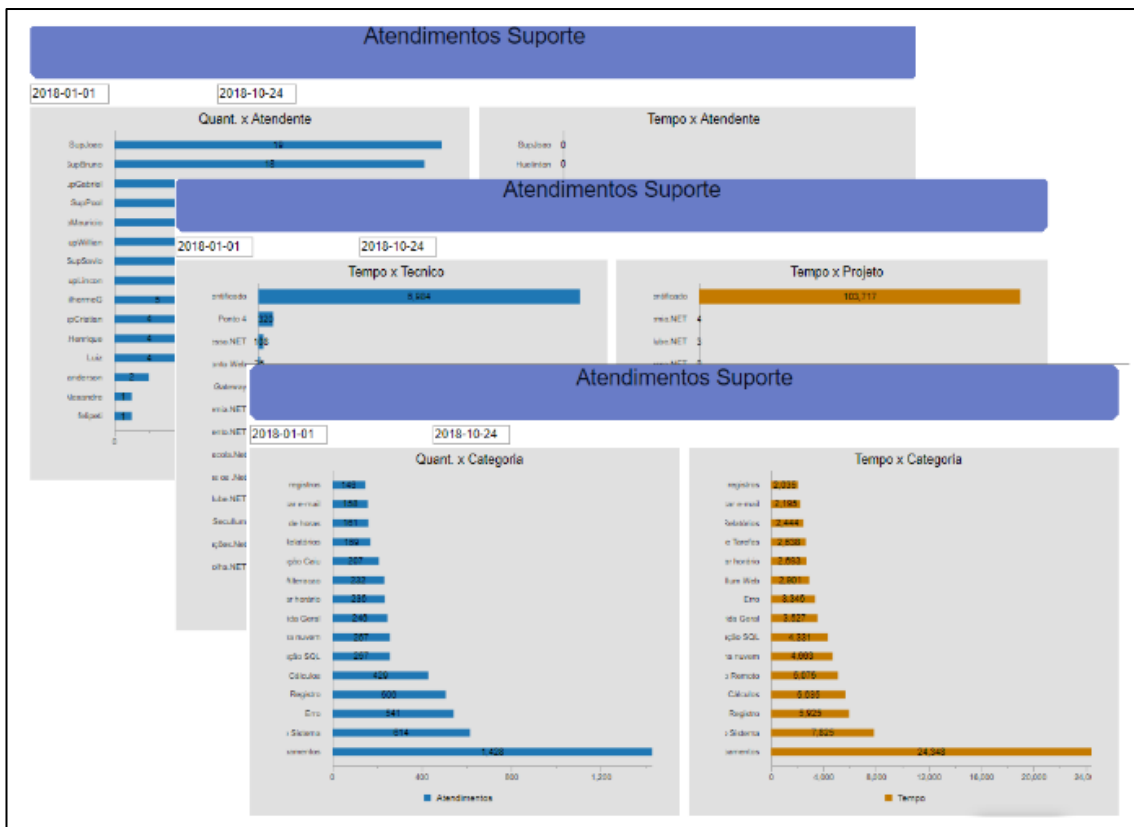
A partir da versão 1.6, a *interface* é apresentada diretamente na plataforma Clicdata, de forma mais amigável e totalmente *Web*, conforme pode ser observado na figura 23.

Figura 22. Relatório Analítico utilizando Pentaho BI

dim_tecnicos	DIM_DATA	DIM_PROJETO	DIM_SUBCATEGORIA	dim_Origem	Measures
All dim_tecnicos	All DIM_DATAs	All DIM_PROJETOs	All DIM_SUBCATEGORIAS	All dim_Origems	TEMPO 74.943
Alessandra	All DIM_DATAs	All DIM_PROJETOs	All DIM_SUBCATEGORIAS	All dim_Origems	752
Augusto Barnaske	All DIM_DATAs	All DIM_PROJETOs	All DIM_SUBCATEGORIAS	All dim_Origems	546
Bruno	All DIM_DATAs	All DIM_PROJETOs	All DIM_SUBCATEGORIAS	All dim_Origems	711
ComLeandro	All DIM_DATAs	All DIM_PROJETOs	All DIM_SUBCATEGORIAS	All dim_Origems	60
ComRodrigo	All DIM_DATAs	All DIM_PROJETOs	All DIM_SUBCATEGORIAS	All dim_Origems	128
Cristian	All DIM_DATAs	All DIM_PROJETOs	All DIM_SUBCATEGORIAS	All dim_Origems	8.471
Daniel	All DIM_DATAs	All DIM_PROJETOs	All DIM_SUBCATEGORIAS	All dim_Origems	206
Dante	All DIM_DATAs	All DIM_PROJETOs	All DIM_SUBCATEGORIAS	All dim_Origems	2.611
DevMateus	All DIM_DATAs	All DIM_PROJETOs	All DIM_SUBCATEGORIAS	All dim_Origems	637
Diego	All DIM_DATAs	All DIM_PROJETOs	All DIM_SUBCATEGORIAS	All dim_Origems	163
Douglas	All DIM_DATAs	All DIM_PROJETOs	All DIM_SUBCATEGORIAS	All dim_Origems	7.429
Douglas S.	All DIM_DATAs	All DIM_PROJETOs	All DIM_SUBCATEGORIAS	All dim_Origems	1.401
	2015	All DIM_PROJETOs	All DIM_SUBCATEGORIAS	All dim_Origems	18
	2017	All DIM_PROJETOs	All DIM_SUBCATEGORIAS	All dim_Origems	683
	9	All DIM_PROJETOs	All DIM_SUBCATEGORIAS	All dim_Origems	309
		Alteração	All dim_Origems	All dim_Origems	1
		Backup	All dim_Origems	All dim_Origems	8
		Banco de Dados Corrompido	All dim_Origems	All dim_Origems	1

Fonte: Autor

Figura 23. Dashboards utilizando Pentaho BI Server



Fonte: Autor

9.1 Entrega final, utilizando CliData

Para que o Clicdata receba os dados necessários para exibir o painel da figuras 26 e 27, foi necessário enviar os dados para a plataforma *web* da ferramenta, tarefa realizada através de um agente que comunica com o banco de dados e envia as informações para a plataforma, o agente pode ser encontrado no *site* clicdata.com e para enviar os dados necessários, alguns *selects* são executados, como por exemplo o *select* exibido na figura 24.

Figura 24. Exemplo de *select* que alimenta o ClicData

```
SELECT nomeAtendente,
       data,
       tempo,
       subCate,
       projeto,
       origem
FROM fatoAtendimento
INNER JOIN dim_data t
  ON t.date_sk = fatoAtendimento.Date_sk
INNER JOIN dim_tecnicos
  ON dim_tecnicos.idAtendente_SK = fatoAtendimento.idAtendente_sk
JOIN DIM_SubCategoria
  ON DIM_SubCategoria.idCategoria_SK = fatoAtendimento.idCategoria_SK
```

Fonte: Autor

Em uma plataforma bastante simplificada, foi realizada a modelagem dos painéis com os dados carregados na plataforma, onde no exemplo abaixo (figura 25), *Drilldown Columns* podem ser interpretadas como as dimensões e *Values Columns* são contadores e medições.

Figura 25. Exemplo de configuração do Dashboard

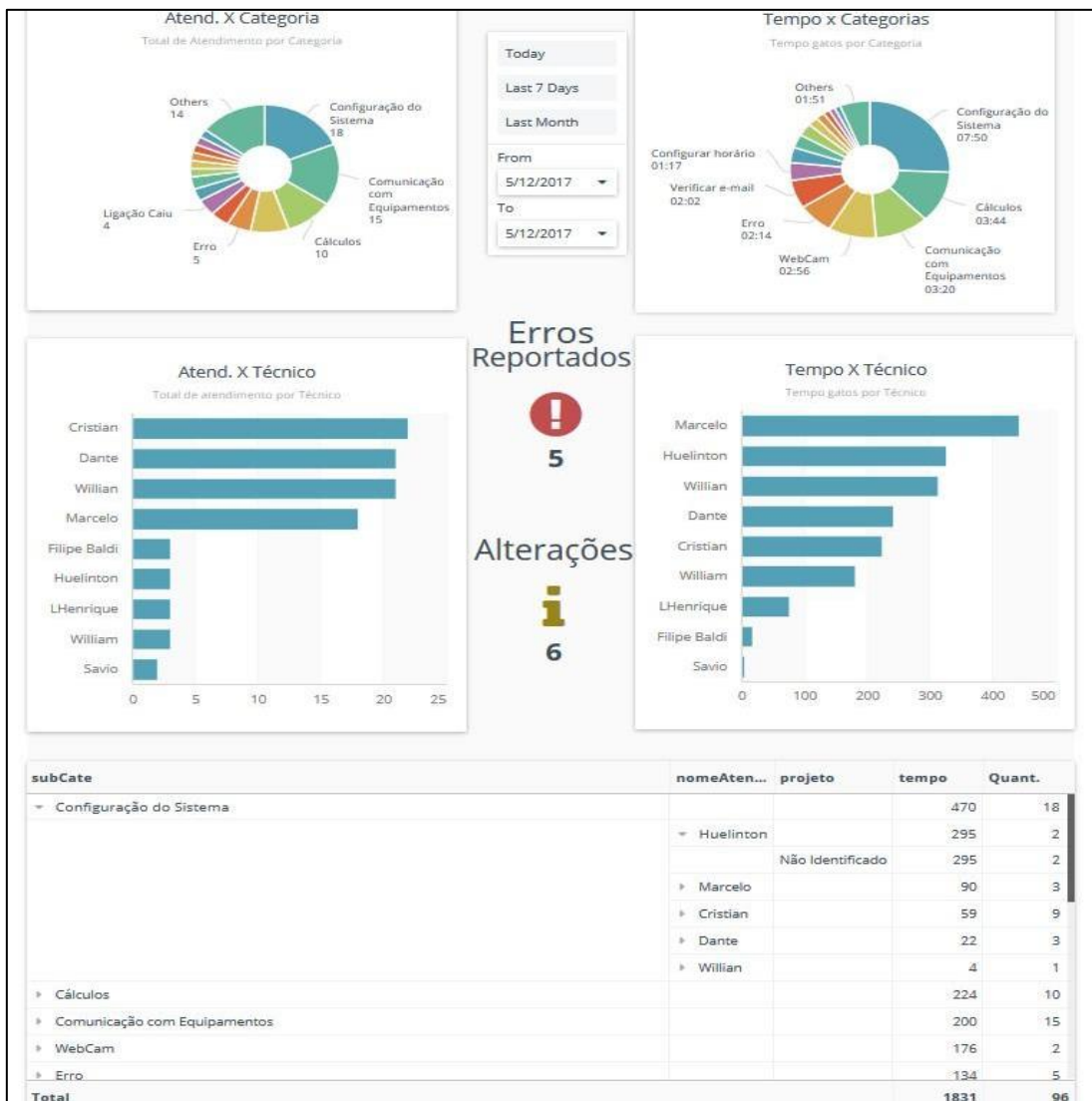
subCate	tempo	Quant.
Configuração do Sistema	943	30
Clicdata	468	20
Comunicação com Equipamentos	400	30
Insticam	322	4
Etis	208	10
Verificar e-mail	244	6
Configurar horário	154	16
Acesso Remoto	132	2
Instalador do sistema	116	4
Instalação SQL	108	6
Instalação com sistema	76	4
Total	3662	102

Fonte: Autor

Na versão final para este projeto, apresentada na figura 26 e 27, pode-se observar a evolução na apresentação das informações para tomada de decisões, onde em uma única tela é possível encontrar respostas para diversas dúvidas pertinentes aos atendimentos, como por exemplo:

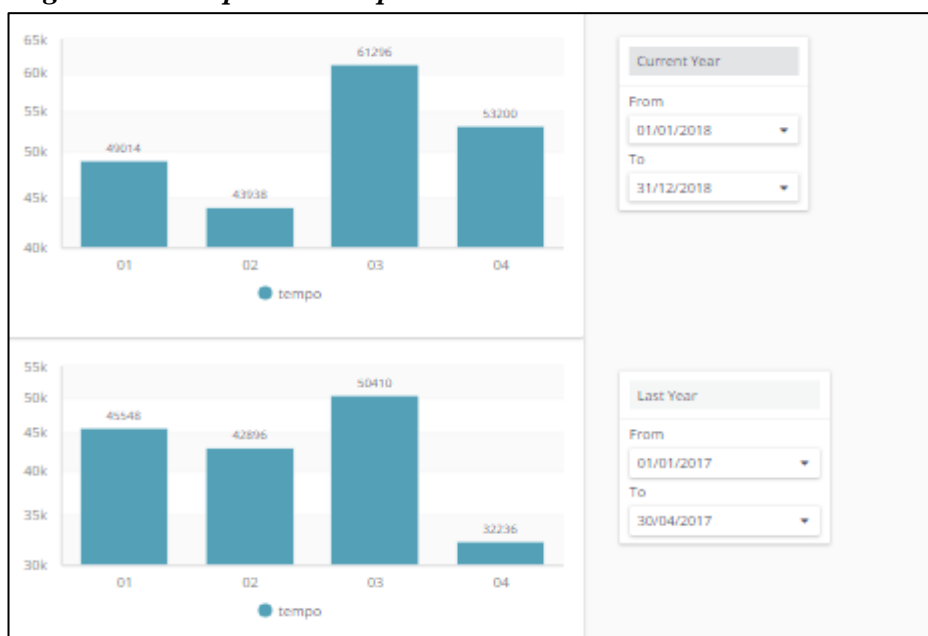
- Quais os colaboradores mais atuantes no período e quais atividades foram realizadas?
- Qual tipo de atendimento tem consumido mais recursos da equipe?
- Qual tipo de atendimento gera maior número de chamados?
- Dentre os atendimentos do período, quanto foram reportados como erro?
- Qual período possui maior demanda de atendimentos?

Figura 26. Dashboards e Relatório analítico utilizando ClicData



Fonte: Autor

Figura 27. Comparativo de períodos



Fonte: Autor

Percebe-se a diferença na facilidade de visualização destas informações e a propriedade estabelecida com a solução viabilizando facilmente o subsídio para tomada de decisão quando comparado com os relatórios apresentados na seção 2.0.

10. Resultados

Ao final do projeto, conclui-se que o objetivo do trabalho foi atingido. O fato de existirem dados de fácil acesso para análise e acompanhamento, instigou o departamento de suporte técnico a realizar reuniões periódicas a fim de tentar entender picos de atendimentos, sobrecargas de atendimentos para grupos de pessoas e tempo gasto para algumas categorias de atendimentos. As reuniões periódicas serviram também como uma forma de integrar todos os participantes do departamento do suporte e permitir que estes sejam mais participativos na organização do setor.

Observando a sobrecarga dos atendimentos, houve a implementação de rodízios periódicos para atendimentos telefônicos, de forma que cada grupo de atendentes fica 30 dias realizando atendimentos no canal telefônico, após esse período, outro grupo assume a função.

11. Conclusões e futuro.

No desenvolvimento do projeto, pode-se identificar e comprovar a eficácia da ferramenta Pentaho *Data Integration* no tratamento de dados, criação de rotinas *ETL* e junção de múltiplas bases de dados. No entanto, também foi possível constatar que a ferramenta Pentaho *BI Server*, não se adequou corretamente ao ambiente ao qual foi exposta, uma vez que esta ferramenta demanda grande consumo de memória RAM e processamento, apresentando consumo crescente conforme o volume de dados carregados. Encontra-se então os serviços da plataforma ClicData como alternativa ao Pentaho *BI Server*, devido ao processamento desta plataforma ocorrer todo em ambiente *WEB*, não há sobrecarga de processamento ou estouro de consumos de recursos dos servidores internos, apresentando ainda uma *interface* mais amigável para o usuário final. Quanto a utilização de ferramentas analíticas e *dashboards*, foi possível entender a importância do tratamento dos dados coletados em ambiente empresarial, bem como a divulgação de informações de forma fácil e rápida pode auxiliar nas mudanças comportamentais de todos envolvidos, instigando a busca por soluções de problemas que até então não estavam evidentes.

A implementação do projeto para este artigo foi concluída na versão 2.0, no entanto, estima-se que o produto continue crescendo conforme necessidade da organização, incluindo novas fontes de dados, como o Octadesk que não pode ser incluso na implementação, e até mesmo buscar dados para análise de outros setores.

Notou-se que, as informações atualizadas periodicamente agilizam de forma significativa a tomada de decisões, o que abre espaço para questionarmos quais melhorias podem ser encontradas caso os dados sejam atualizados em tempo real nos dashboards, permitindo a visualização em instantânea sobre os atendimentos. Para estudo desta possibilidade, reserva-se pesquisas sobre o tratamento de dados sobre o fluxo de *Data Stream*, onde os dados são trabalhados em tempo real.

12. REFERÊNCIAS

CLIDATA. Solutions Disponível em: < <https://www.clicdata.com/solutions/>> Acesso em:25 de out. 2018.

ECKERSON, W. *Smart Companies in the 21 Century: the secrets of creating successful business intelligent solutions*. Seattle, WA: The Data Warehousing Institute, 2003.

ELEUTERIO, Marco Antonio. Sistemas de Informações Gerenciais na Atualidade: 1. ed. Curitiba: Editora Intersaberes, 2015.

IFDEFINED. About BugTracker.NET. Disponível em:
<<https://ifdefined.com/bugtrackernet.html>> Acesso em 20 ago 2018.

INMON, W. H. Como construir o Data Warehouse. 2. ed. Rio de Janeiro: Campus, 1997.

KIMBALL, Ralph. Data Warehouse Toolkit, São Paulo, SP: Makron Books do Brasil Editora Ltda, 1998.

MACHADO, Felipe Nery Rodrigues. Tecnologia e Projeto de Data Warehouse 6. ed. São Paulo: Editora Erica, 2013.

MICROSOFT .Notas de Versão do SQL Server 2016. Disponível em:
<<https://docs.microsoft.com/pt-br/sql/sql-server/sql-server-2016-release-notes?view=sql-server-2016>> Acesso em 20 set 2018.

PENTAHO .Pentaho: Install Pentaho Software. Disponível em:
<<https://help.pentaho.com/Documentation/5.1/0F0>> Acesso em jul de 2018.

PRATES, Gláucia Aparecida. Tecnologia da informação em pequenas empresas: fatores de êxito, restrições e benefícios. Disponível em:
<http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1415-6552004000200002>. Acesso em 10 ago. 2018.

TURBAN, E., Sharda, R., Aronson, J. E., & King, D. (2009). Business Intelligence: um enfoque gerencial para a inteligência do negócio. Bookman.

VIEIRA, Maria Inês Paraíso, Monitorização do desempenho através de Dashboards. Disponível em: <https://comun.rcaap.pt/bitstream/10400.26/21147/1/Maria_Vieira.pdf> acesso em 09 out. 2018