

CLAUDIO BONEL

# Metodologia e Engenharia de Requisitos para projetos de Business Intelligence

Análise de dados  
levada a sério



Metodologia e Engenharia de Requisitos para Projetos de  
Business Intelligence

# Metodologia e Engenharia de Requisitos para projetos de Business Intelligence

Análise de dados levada a sério

Claudio Bonel  
Rio de Janeiro  
2021

# Metodologia e Engenharia de Requisitos para Projetos de Business Intelligence

## Dados Internacionais de Catalogação na Publicação (CIP) (Câmara Brasileira do Livro, SP, Brasil)

Bonel, Claudio

Metodologia e engenharia de requisitos para projetos de business intelligence [livro eletrônico] : análise de dados levada a sério / Claudio Bonel. -- 1. ed. -- Rio de Janeiro : Ed. do Autor, 2021.

PDF

Bibliografia

ISBN 978-65-00-17761-9

1. Business intelligence 2. Engenharia de software 3. Inteligência competitiva (Administração) 4. Projetos - Metodologia 5. Sistemas de informação gerencial I. Título.

21-57356

CDD-658.4038

### Índices para catálogo sistemático:

1. Business intelligence : Administração de empresas  
658.4038

Maria Alice Ferreira - Bibliotecária - CRB-8/7964



# Metodologia e Engenharia de Requisitos para Projetos de Business Intelligence

Dedico este trabalho à eterna memória do meu amado pai. Jamais o decepcionarei.

## AGRADECIMENTOS

À minha amada mãe, pelo seu empenho em educar a mim e à minha irmã, nos fazendo entender o real significado da Educação.

Aos professores e professoras, pelo aprendizado constante que tive e tenho.

Aos autores que são minha referência, pelo conteúdo disponibilizado.

“A educação é um processo social,  
é desenvolvimento. Não é a  
preparação para a vida, é a própria  
vida.” (John Dewey, 1933)

## Sumário

Introdução.....	14
Etapas de um Projeto de <i>Business Intelligence</i> .....	19
Descobrir.....	31
Definir.....	32
Desenvolver e validar.....	32
Entregar .....	35
Descobrir – Planejamento e Requisitos da Solução... 37	
Planejamento da Solução. A Engenharia de	
Requisitos .....	37
Requisitos da solução .....	59
Requisitos Funcionais .....	60
Requisitos Não Funcionais.....	68
Matriz de Necessidades .....	71
Definir – Design da Solução .....	94

# Metodologia e Engenharia de Requisitos para Projetos de Business Intelligence

Requisitos Técnicos .....	94
Arquitetura de dados.....	107
Integrações de Dados .....	122
ETL.....	123
ELT .....	123
Governança de Dados .....	127
Processos-chave da Governança de dados .....	129
Time de Governança de Dados.....	139
Lei Geral de Proteção de Dados (LGPD). Lei 13.709 .....	142
Modelo de dados.....	156
Camada de apresentação .....	181
Casos de Teste .....	187
Segurança da Informação .....	198
Desenvolvimento e Validação .....	203
Entrega .....	207



# Metodologia e Engenharia de Requisitos para Projetos de Business Intelligence

Operação .....	207
Manutenção da Solução .....	209
Conclusão.....	214
Referências .....	217
Sobre Claudio Bonel.....	220

## **Figuras**

<b>Figura 1 - Etapas de um projeto de Business Intelligence .....</b>	<b>19</b>
<b>Figura 2 – Princípios da metodologia ágil .....</b>	<b>22</b>
<b>Figura 3 – Dinâmica de uma metodologia ágil .....</b>	<b>23</b>
<b>Figura 4 – Solução de Business Intelligence, dividida em pequenas entregas .....</b>	<b>24</b>
<b>Figura 5 – Releases de um projeto de Business Intelligence .....</b>	<b>28</b>
<b>Figura 6 – Ciclo de desenvolvimento dos módulos e seus releases .....</b>	<b>29</b>
<b>Figura 7 – Etapas da Engenharia de Requisitos.....</b>	<b>39</b>
<b>Figura 8 - Processo para levantamento de requisitos</b>	<b>53</b>
<b>Figura 9 – Modelo da Matriz de Necessidades, proposto por Ralph Kimball .....</b>	<b>75</b>
<b>Figura 10 – Fases do levantamento de requisitos para uma solução de Business Intelligence.....</b>	<b>76</b>
<b>Figura 11 – Análise da descrição do requisito (RF01)</b>	<b>77</b>

<b>Figura 12 – Primeira versão da matriz de necessidades</b>	78
<b>Figura 13 – Análise da Regra (RF01)</b>	79
<b>Figura 14 – Segunda versão da matriz de necessidades</b>	80
<b>Figura 15 – Análise da granularidade (RF01)</b>	81
<b>Figura 16 – Terceira versão da matriz de necessidades</b>	82
<b>Figura 17 – Análise da descrição do requisito (RF02)</b>	83
<b>Figura 18 – Análise da Regra (RF02)</b>	84
<b>Figura 19 – Quarta versão da matriz de necessidades</b>	85
<b>Figura 20 – Análise da granularidade (RF02)</b>	85
<b>Figura 21 – Análise da descrição do requisito (RF03)</b>	86
<b>Figura 22 – Quinta versão da matriz de necessidades</b>	87
<b>Figura 23 – Análise da Regra (RF03)</b>	88
<b>Figura 24 – Sexta versão da matriz de necessidades</b>	89
<b>Figura 25 – Análise da granularidade (RF03)</b>	90

<b>Figura 26 – Sétima versão da matriz de necessidades</b>	91
<b>Figura 27 – Versão final da análise dos três primeiros requisitos</b>	91
<b>Figura 28 – Matriz de necessidades para aderência a tecnologia</b>	101
<b>Figura 29 – Arquitetura de dados proposta por Kimball</b>	110
<b>Figura 30 - Arquitetura de dados proposta por Inmon</b>	114
<b>Figura 31 – Arquitetura de baixa complexidade</b>	118
<b>Figura 32 - Arquitetura de média complexidade</b>	120
<b>Figura 33 - Arquitetura do cenário proposto</b>	122
<b>Figura 34 – Áreas de conhecimento da Governança de dados</b>	130
<b>Figura 35 - Hierarquia do time de governança de dados</b>	141
<b>Figura 36 – Modelo Estrela (Star Join) proposto por Inmon</b>	163

<b>Figura 37</b> – Exemplo de possibilidades de análises um cubo.....	168
<b>Figura 38</b> – Modelo de dados estrela.....	172
<b>Figura 39</b> – Modelo de dados com meta por cidade	173
<b>Figura 40</b> – Ilustração do Relacionamento entre tabelas .....	177
<b>Figura 41</b> – Relacionamento cardinal .....	179
<b>Figura 42</b> – Relacionamento cardinal 1:* (um para muitos).....	180
<b>Figura 43</b> – Fluxo de processos para levantamento de requisitos da camada de apresentação .....	183
<b>Figura 44</b> – Croqui para desenvolvimento de dashboard.....	184
<b>Figura 45</b> – Processos do teste de software .....	190
<b>Figura 46</b> – Fluxo do processo de testes pelo usuário .....	195
<b>Figura 47</b> – Casos de teste em marcos do desenvolvimento.....	204

## **Tabelas**

<b>Tabela 1</b> – Modelo para mapeamento de desafios ....	42
<b>Tabela 2</b> – Modelo para levantamento de Requisitos Funcionais. ....	63
<b>Tabela 3</b> – Modelo para levantamento de requisitos não funcionais .....	70
<b>Tabela 4</b> – Modelo para levantamento de Requisitos Técnicos, associados aos requisitos funcionais .....	102
<b>Tabela 5</b> - Modelo para levantamento de Requisitos Técnicos, associados aos requisitos não funcionais	104
<b>Tabela 6</b> – Modelo para levantamento de requisitos de extração de dados .....	155
<b>Tabela 7</b> – Exemplo da tabela de vendas .....	158
<b>Tabela 8</b> – Modelo de dados do exemplo .....	160
<b>Tabela 9</b> – Modelo para mapeamento dos cenários de teste .....	196
<b>Tabela 10</b> – Modelo para mapeamento dos níveis de acesso .....	199

## Introdução

No livro *How to solve almost any problem*, encontramos um diálogo com Patrick Meehan, que na época era Presidente e Diretor de Pesquisa da *Gartner's CIO Research*, onde fora perguntado: “*Quando os projetos de business intelligence carecem de inteligência?*”. O Sr. Meehan respondeu da seguinte forma:

Se você não fizer as perguntas certas, o BI não é uma bola de cristal que sai respondendo. O pessoal de TI precisa parar de abordar o BI como um fornecedor ou solução de engenharia, ou como uma ferramenta. Eles precisam entender em que negócio estão. Eles estão no negócio de informação e comunicação. (BARKER, 2012, p.141)

As primeiras linhas da resposta de Meehan, se completam com um relatório dos relatórios do Gartner <sup>1</sup> “70% to 80% dos projetos corporativos de *business intelligence* falham” (BARKER, 2012, p.141).

Diante do exposto e alinhado com minha experiência no mercado de consultoria de *business intelligence*, provendo soluções para grandes empresas do mundo corporativo, decidi trazer esta visão prática de como entregar soluções de inteligência de negócio que, de fato, geram resultados para as empresas.

Assim como o Sr. Meehan destacou: “O BI não é uma bola de cristal”, portanto este livro vai te apresentar como obter as respostas que você precisa, para entregar sua solução de *Business Intelligence*, alinhada com os desafios da Empresa.

---

<sup>1</sup> Gartner é o maior instituto de pesquisa do mundo, com o objetivo de fornecer *insights* sobre o negócio, bem como sugerir tecnologias e fornecedores para auxiliar tomadores de decisão em empresas



Atualmente, analisar dados, gerar e acompanhar indicadores para suportar tomadas de decisão, tornou-se uma premissa básica para que você possa conduzir seu negócio, seja qual for. Micro, Pequeno, Médio ou Grande. Analisar dados é essencial.

Você confecciona bolos na cozinha da sua casa e vende?

Você é um/a contador/a e atende clientes do seu bairro?

Você tem uma farmácia?

Você é gerente de um supermercado?

Você é gerente de um departamento em uma grande empresa?

Você é executivo/a de uma multinacional?

Você é presidente de uma Organização Não Governamental (ONG)?

Você é síndico/a?

Você coleta dados públicos para gerar informações para o seu bairro?

Você é Professor/a e precisa fazer a gestão do seu conteúdo e de seus/suas alunos/as?

Você precisa fazer a gestão financeiras dos seus gastos pessoais?

Para todos esses casos e milhares de outros, qualquer que seja, que envolva alguma tomada de decisão, por mais simples que for, você precisará analisar dados para tomar decisões mais assertivas.

A análise de dados é um conceito que, através de uma metodologia pode ser implantado por qualquer pessoa, independente de sua formação profissional, basta seguir as boas práticas apresentadas que servirão, tanto para pequenas soluções, quanto para soluções enormes.

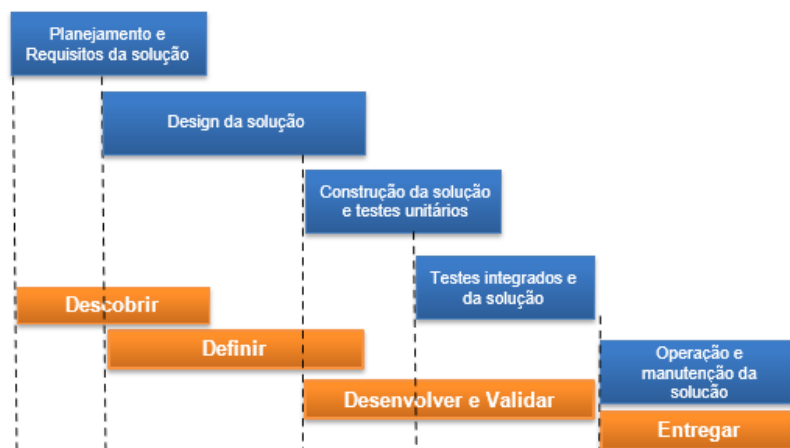
Esta metodologia tem por objetivo apoiar de forma simples e organizada todos os requisitos necessários para qualquer solução de *Business Intelligence*,

independente da complexidade e da ferramenta a ser utilizada, baseando em princípios da Engenharia de *Software* ágil, Engenharia de Requisitos e de *Design Thinking*, porém de fácil acesso a todos, ou seja, de simples entendimento, também para profissionais que não são da área de Tecnologia da Informação.

## Etapas de um Projeto de *Business Intelligence*

As etapas de um projeto de *Business Intelligence*, tem por objetivo descrever cada ação que deverá ser executada e cumprida para que no final você tenha uma solução que, de fato, resolva os desafios dos/as usuário/as.

**Figura 1** - Etapas de um projeto de *Business Intelligence*



Fonte: Autoria própria

A metodologia para desenvolvimento de uma solução, proposta pela Engenharia de *Software* clássica vai nos apresentar que uma fase de projeto só poderá iniciar uma vez que a fase anterior finalizar. Porém com o passar dos anos e a dinâmica de mercado, a engenharia de *software* também foi se adaptando para que pudesse estar alinhada com esse dinamismo.

Para entender melhor, imagine que eu faça um levantamento de requisitos com você, no dia 28 de janeiro de 2021, pois bem, executei todas as melhores práticas da Engenharia de *Software* e de Requisitos clássica e, de acordo com suas necessidades apresentadas, o prazo para desenvolvimento será de 60 (sessenta) dias, onde eu desenvolverei todo o projeto sem interação com você, seguindo exatamente o que está no documento de requisitos, até que eu possa disponibilizar o software para sua validação no dia 21 de março de 2021.

Algumas reflexões que eu te convido a fazer, diante deste cenário:

1. Será que houveram ajustes nos requisitos mapeados, durante os 60 (sessante) dias?
2. Será que essa solução não poderia ser dividida em módulos, com pequenas entregas e validações, onde, uma vez unificados e validados, formariam a software?
3. Quantas reuniões com a sua gestão foram realizadas? E destas reuniões, novas informações foram solicitadas?

Apenas três perguntas para você refletir e responder mentalmente, porém provavelmente você respondeu “Sim” em pelo menos uma destas reflexões mas, creio que você respondeu “Sim” em todas, o mercado de trabalho é dinâmico todas as reflexões expostas são possíveis se tornarem realidade (Se não for a sua realidade neste momento).

Por conta deste cenário atual mercadológico, a Engenharia de *Software* tem recomendado uma metodologia ágil, de forma que seja possível alinhar-se ao máximo as demandas do mercado.

**Figura 2 – Princípios da metodologia ágil**

Princípios	Descrição
Envolvimento do cliente	Os clientes devem estar intimamente envolvidos no processo de desenvolvimento. Seu papel é fornecer e priorizar novos requisitos do sistema e avaliar suas iterações.
Entrega incremental	O software é desenvolvido em incrementos com o cliente, especificando os requisitos para serem incluídos em cada um.
Pessoas, não processos	As habilidades da equipe de desenvolvimento devem ser reconhecidas e exploradas. Membros da equipe devem desenvolver suas próprias maneiras de trabalhar, sem processos prescritivos.
Aceitar as mudanças	Deve-se ter em mente que os requisitos do sistema vão mudar. Por isso, projete o sistema de maneira a acomodar essas mudanças.
Manter a simplicidade	Focalize a simplicidade, tanto do software a ser desenvolvido quanto do processo de desenvolvimento. Sempre que possível, trabalhe ativamente para eliminar a complexidade do sistema.

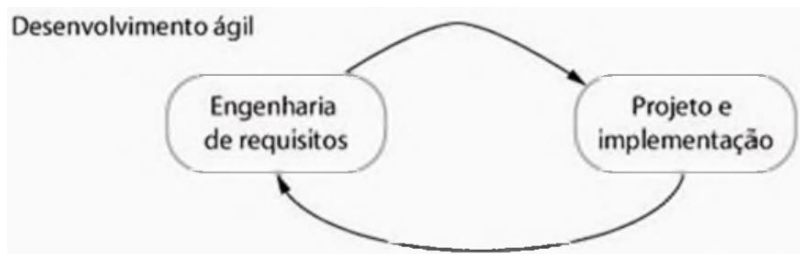
Fonte: Princípios dos métodos ágeis (SOMMERVILLE, 2011, p. 40)

Nas etapas de um projeto de business intelligence, apresentada anteriormente, perceba que elas começam, mesmo antes de se finalizar a etapa anterior, justamente para te salientar da importância desse alinhamento de expectativa constante, sempre verificando se há algo da etapa anterior que precisa ser ajustado.

Todos os princípios apresentados por Sommerville (2011) são importantes, mas gostaria de destacar dois deles: **Envolvimento do cliente e Entrega incremental**, pois são os princípios base.

Particionar o seu *software* e dividi-lo em pequenas entregas, vai facilitar o seu trabalho de mapeamento de desafios e requisitos, bem como terá um maior envolvimento do/a usuário/o, onde ele realizará validações (Veremos em casos de teste) com mais constância, com prazos menores e com isso, você terá mais possibilidades de acomodar e negociar qualquer tipo de alteração.

**Figura 3 – Dinâmica de uma metodologia ágil**



Fonte: Desenvolvimento ágil (SOMMERVILLE, 2011, p. 43)



Como é possível perceber, há uma conversa constante entre a Engenharia de Requisitos e o desenvolvimento do projeto e pelo fato de você dividir o projeto inteiro em pequenas fases, a tendência é que esse processo seja natural e transparente.

**Figura 4** – Solução de Business Intelligence, dividida em pequenas entregas



Fonte: Autoria própria

Ressalta-se que é possível ter N entregas, desde que façam sentido e que, de fato, gere um resultado para o/a usuário/a.

Porém, ágil não quer dizer “fazer de qualquer jeito”. Minha experiência desde 2010 em Engenharia de Requisitos mostra que é preciso que você realize um levantamento de requisitos inicial e tenha um documento de partida, pois será o seu direcionamento e, sempre que houver alterações, este documento deve ser versionado com as devidas especificações.

Isto posto, ressalta-se que a Engenharia de Requisitos e suas documentações continuam sendo essenciais e de extrema importância para o sucesso do seu *software* de *Business Intelligence*.

Uma das metodologias ágeis que gosto de trabalhar é conhecida por *Extreme Programming*, ou simplesmente XP. Vou explicar esta metodologia da forma que eu trabalho em soluções de *Business Intelligence*.

Consiste em levantar os requisitos com os/as usuários/as e a partir destes requisitos há uma divisão de *releases*, em português *lançamentos*. Cada

lançamento é uma parte do todo e que gerará resultado para o/a usuário/a.

Lembre-se que para dividir o software em partes, estas precisam fazer sentido e prover entregas que possam ser utilizadas imediatamente. Definidos os *releases*, inicia-se o desenvolvimento, os testes e a liberação do *release* para consumo.

Imagine que ao levantar os requisitos de um *software* de *business intelligence* para uma empresa, foi identificado que será necessário analisar vendas, comportamento de clientes e acompanhamento dos fornecedores.

Perceba que somente o mapeamento das macro necessidades já te dá um direcionamento sobre o planejamento de *releases*, logo você já tem três partes do seu *software* de *Business Intelligence*, que eu chamo de módulo:

- Vendas
- Clientes

- Fornecedores

Uma vez mapeado, identifique as prioridades junto com seu cliente. Imagine que a prioridade de desenvolvimento seja:

1. Clientes
2. Vendas
3. Fornecedores

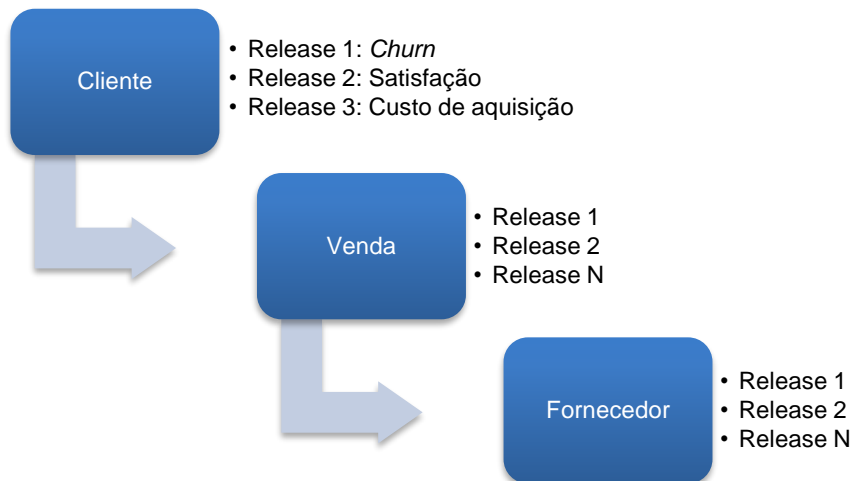
Feito isso, você iniciará a Engenharia de Requisitos na busca das definições para informações relacionadas a clientes, bem como a priorização destes. Imagine que se chegou ao seguinte:

1. *Churn* (perda) de clientes
2. Satisfação dos clientes (*Customer Satisfaction Score*)
3. Custo de aquisição de clientes

Desta forma, você já tem três *releases* relacionados ao módulo de cliente, ou seja, três entregas que agregarão

valor ao negócio, gerarão resultados e suportarão a tomadas de decisão.

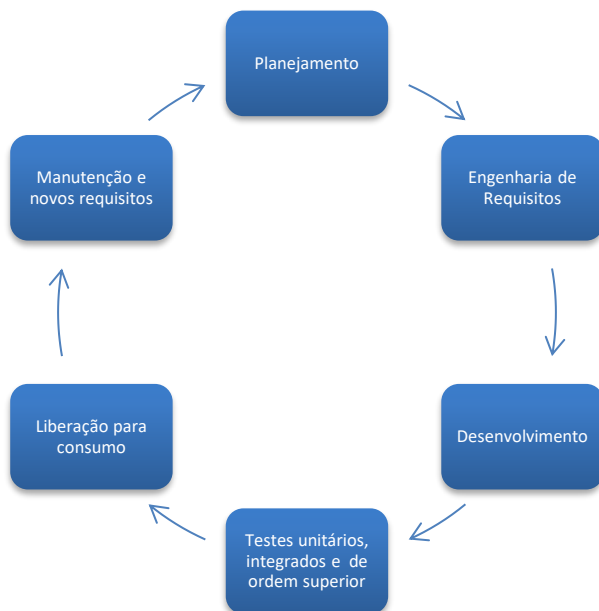
**Figura 5 – Releases de um projeto de Business Intelligence**



Fonte: Autoria própria

Cada *release*, de cada módulo do seu software de business intelligence passará necessariamente pelas etapas essenciais da Engenharia de Requisitos, que no decorrer do projeto se tornará um ciclo natural e transparente.

**Figura 6** – Ciclo de desenvolvimento dos módulos e seus releases



Fonte: Autoria própria

Como estamos falando somente de um determinado *release*, conseqüentemente teremos mais entregas e este ciclo se repetirá constantemente, até o momento que você nem perceberá.

Um ponto de atenção a se observar é que a partir do segundo *release*, este ciclo tem que ser pensado de forma incremental, de modo que todos os *releases* desenvolvidos entreguem o módulo; e todos os módulos entreguem o *software* de *business intelligence*.

Quando falamos no “pensar de forma incremental”, é para você reutilizar ao máximo tudo o que foi desenvolvido e integrar todas as informações.

Apresentaremos um breve resumo sobre cada etapa de um projeto de *Business Intelligence* para, posteriormente, detalharmos cada uma delas, lembrando que tudo o que for abordado nas páginas deste livro, recomenda-se utilizar dentro da metodologia ágil.

## **Descobrir**

É o estágio onde é definido o direcionamento da solução, ou seja:

- O que a solução deve resolver?
  - A etapa de Planejamento e Requisitos da solução deve responder a esta pergunta. E aqui daremos ênfase a Engenharia de Requisitos
- Identificar:
  - Tema norteador e objetivos
  - Requisitos
  - Público
- Nesta etapa também será definida a Matriz de Necessidades da solução.



## **Definir**

É o estágio onde é definido como a solução será desenvolvida e deve responder a seguinte pergunta:

- Como a solução resolverá os desafios mapeados no estágio “Descobrir”?
  - A Etapa de design da solução deve responder essa pergunta.

## **Desenvolver e validar**

É o estágio onde, baseado no anterior, a construção da solução é realizada, testada pelos desenvolvedores e validada pelos/as usuários/as e time de Tecnologia da Informação (TI) dos seus clientes. As perguntas que devem ser respondidas são?

- Os *releases*, módulos e o *software* estão sendo desenvolvidos dentro das melhores práticas, alinhada com os requisitos e *design* levantados e documentados no estágio “Descobrir”?
  - A etapa “Construção da solução e testes unitários” deve responder a esta pergunta.
    - Resposta “Sim”: Melhor dos mundos, vamos em frente!
    - Resposta “Não”: Deve-se entender o porquê, identificar a(s) inconsistência(s) com os requisitos e design e corrigi-los.
- A solução construída resolve os desafios e atende aos requisitos técnicos, mapeados nos estágios “Descobrir” e “Definir”?
  - A etapa “Testes integrados e de ordem superior” é a responsável por esta pergunta e deve ser respondida pelos usuários-chave e pelo time de TI do seu

cliente. Em projetos esta etapa é conhecida como “Go/No Go”, ou seja, vai para operação ou não?

- Resposta “Sim”: Melhor dos mundos, vamos em frente!
- Resposta “Não”: Deve-se entender o porquê, identificar a(s) inconsistências com os requisitos e *design* e:
  - Se for um requisito mapeado e alinhado, deve-se corrigir a inconsistência.
  - Se for um requisito novo, ou seja, está sendo informado somente agora, deve-se negociar o entendimento e esforço, para decidir se justifica a sua implantação neste momento ou em um momento posterior.

## **Entregar**

É o estágio onde, após aprovação da solução construída na etapa “Testes integrados e de ordem superior”, tem por objetivo 2 tarefas:

1. Colocar a solução desenvolvida e aprovada em operação/produção, é o chamado *Go Live!* E a partir deste momento todos/as os/as usuários/as poderão utilizá-la.
2. Realizar a manutenção da solução, que em um primeiro momento deve assistir aos usuários na utilização – chamado de operação (ou produção) assistida – ou seja, auxiliá-los em qualquer dúvida ou problema que seja encontrado durante a utilização. Em um segundo momento entra o suporte (ou sustentação) da solução, que é possível garantir que a solução funcione a médio-longo prazo e prover melhorias a mesma, com a

implantação de novos requisitos e ajustes de  
requisitos antigos.

## **Descobrir – Planejamento e Requisitos da Solução**

Neste estágio, tem-se como principal desafio a descoberta das principais necessidades dos/as usuários/as, mais conhecido como desafios, dores, sonhos, anseios, ou seja, àquilo que a solução deve resolver.

### **Planejamento da Solução. A Engenharia de Requisitos**

A Engenharia de Requisitos está contida na Engenharia de *Software* e é parte importante e premissa essencial para o desenvolvimento de qualquer tipo de *software* e uma solução de *Business Intelligence* é um *software*, cujo o objetivo é prover informações para tomadas de decisão.

Segundo o dicionário Aurélio, Engenharia significa:

Aplicação de conhecimentos científicos e empíricos e certas habilitações específicas à criação de estruturas, dispositivos e processos para converter recursos naturais em formas adequadas ao atendimento das necessidades humanas. (FERREIRA, 2019)

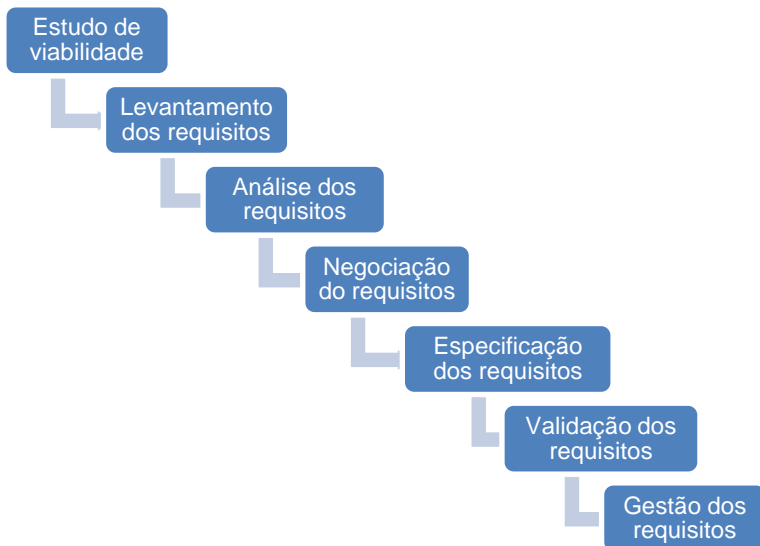
Ao analisar o significado, percebe-se que a engenharia também está diretamente ligada a processos que impactarão as necessidades humanas estas, em um projeto de *Business Intelligence*, são representadas pelos requisitos.

Portanto, no decorrer das páginas deste livro, teremos uma melhor compreensão dos processos essenciais para um mapeamento assertivo dos requisitos da solução de *Business Intelligence*, focando em três tipos de requisitos essenciais:

- Requisitos Funcionais
- Requisitos Não Funcionais
- Requisitos Técnicos

A Engenharia de Requisitos está dividida em algumas etapas e são estas que direcionarão a sua caminhada, para construção da solução de Business Intelligence, por este motivo encontra-se na etapa “Descobrir” e é essencial, afinal uma solução de sucesso é aquela que atende aos requisitos do(s) usuário(s).

**Figura 7 – Etapas da Engenharia de Requisitos**



Fonte: Etapas de Engenharia de Requisitos (OLIVEIRA e DI  
NUNO, [entre 2010 e 2016], p. 30-32)



A primeira etapa é o **Estudo de viabilidade**, que basicamente consiste em responder se o *software* que você pretende desenvolver está alinhado com os objetivos da empresa, se a tecnologia que será utilizada estiver dentro do escopo de custo que a empresa suporta e, no caso de uma solução de *Business Intelligence*, se é possível integrar com as fontes de dados de origem, ou seja, aquelas que serão o insumo para sua solução.

Você pode buscar mais informações sobre a viabilidade, porém estas três são essenciais e qualquer negativa pode tornar o projeto inviável.

Além disso, no **estudo de viabilidade** é preciso identificar alguns pontos-chave, que dentro de uma solução de *Business Intelligence*, chamamos de “desafios”, que podem ser problemas e/ou oportunidades.

Neste processo de identificação de desafios, você pode utilizar algumas das perguntas abaixo, segundo Oliveira e Di Nuno ([entre 2010 e 2016], p. 30)

1. O que te levou a desejar este projeto?
2. Quais os principais desafios com esta iniciativa?
3. Se hoje você pudesse ter a solução perfeita, tivesse todo o investimento disponível, o que esta solução deveria prover? Quais as prioridades?
4. Quais os desafios que você enfrenta no seu processo atual, tanto relacionados a problemas, quanto a oportunidades?
5. Quais os problemas que você enfrenta para tomar decisões e que necessitam de uma solução urgente?
6. Quais os problemas que você enfrenta pelo fato de você não ter essa solução?
7. Como a solução contribuiria com os processos de tomada de decisão?

8. Como a solução contribuirá para os objetivos da empresa?

É possível que você identifique inúmeros desafios, portanto é importante que você registre as respostas, independente de quantas pessoas entrevistou.

Lembre-se! “Menos é mais” e Mínimo Produto Viável. Busque delimitar e dividir suas entregas, os chamados *releases* e módulos, conforme já exposto anteriormente.

Possivelmente você terá que auxiliar seu/sua usuário/a na resposta destas perguntas, portanto dê exemplos, utilize a sua experiência e utilize o modelo a seguir para registrar os resultados.

**Tabela 1** – Modelo para mapeamento de desafios

Usuário-chave	Prioridade	Desafio	Data
---------------	------------	---------	------

Metodologia e Engenharia de Requisitos para Projetos de  
Business Intelligence

<b>João</b>	D01	Ranking das lojas com mais vendas/m²	21/02/2018
<b>João</b>	D02	Produtos que menos batem suas metas de vendas	21/02/2018
<b>Maria</b>	D04	Saber quais vendedores bateram suas metas em cada região	22/02/2018
<b>Maria</b>	D04	Identificar área da loja com maior circulação de pessoas	22/02/2018
<b>Maria</b>	D05	Produtos que não batem metas	22/02/2018

Fonte: Autoria própria

A segunda etapa vai tratar do **levantamento dos requisitos** e que consiste em obter as informações necessárias para o desenvolvimento de seus *releases* e, conseqüentemente, dos módulos e por fim, o *software*. Trata-se do que precisa ser respondido para que os desafios mapeados seja superados.

Aqui você entrevistará os/as usuários/as que entendem das regras do negócio, ou seja, a partir de uma desafio, como seria a resposta - na maioria das vezes, as respostas - para suportar uma decisão de modo a resolver este desafio.

Entenda de uma vez por todas, uma solução desalinhada com os requisitos, possivelmente será deixada de lado. Portanto, esse é um trabalho de suma importância. É tão importante, que neste livro, você entrará em detalhes de como relizar entrevistas e registrar todos os requisitos responsáveis por prover respostas aos desafios.

Dando sequência, chegamos a terceira etapa, chamada de **Análise dos Requisitos**. Muito frequentemente, caminha junto com a etapa anterior (Eu trabalho dessa forma), pois consiste em obter uma compreensão além do requisito levantado, por exemplo, se você está realizando uma solução para uma empresa de recursos humanos, convém você entender como o mercado de recursos humanos trabalha, seus principais indicadores, métricas, de modo a analisar o requisito e poder apoiar pró-ativamente o/a usuário/a.

Além disso, nesta etapa, você deve realizar a priorização de cada requisito, dos mais importantes aos menos importantes, obedecendo o processo de dependência destes requisitos. Desta forma, você terá o direcionamento desde o ponto zero (ponto inicial), bem como estará alinhado com a estratégia da empresa.

Ainda nesta etapa, pode ser que você precise gerir conflitos de requisitos (acontece com frequência). Trata-se de duas áreas distintas apresentarem o mesmo

requisito, porém com regras de concepção diferentes. Cabe a você identificar e junto com as áreas envolvidas resolver esse conflito. É importante este alinhamento de expectativas com todos/as os/as envolvidos/as.

Por fim e não menos importante, uma análise geral, de modo a garantir que não há inconsistências, falta de informação ou dúvidas. É uma etapa muito importante para o sucesso das suas soluções.

Feito isso, podemos avançar para quarta etapa, chamada de **negociação dos requisitos**. Nesta etapa, por conta do seu envolvimento no projeto, a expectativa é que você já tenha vencido algumas fases da curva de aprendizado e esteja mais familiarizado com os desafios da solução que está desenvolvendo, logo é esperado que você, após a análise dos requisitos, possa identificar se neste conjunto existe alguns que **NÃO** estão alinhados com os desafios da empresa, se existe requisito que não está associado ao/a usuário/a, ou seja,

inconsistências (mapeadas na etapa anterior) que precisarão ser negociadas.

Trata-se de uma negociação, caso você tenha encontrado estas inconsistências, evidenci-as e negocie com as partes envolvidas.

Reforço, mais uma vez, as expectativas de todos os envolvidos devem, necessariamente, estar alinhadas. Todos precisam saber o que farão, bem como o que receberão como resultado.

Uma vez que todos estão alinhados, chegamos a quinta etapa, chamada de **especificação dos requisitos**. Trata-se de documentar tudo o que foi mapeado, cenários para testar os requisitos, plano de segurança, tudo de forma bem organizada, linguagem fácil e simplificada, pois é um documento que servirá de apoio para você, para seus/suas usuários/as, para o time de suporte dos *releases*, módulos e soluções desenvolvidas (falaremos mais sobre isso) e para futuros profissionais que poderão atualizar sua solução.



Os modelos que estão dispostos neste livro, te apoiarão nesse processo de documentação. Lembre-se! Menos e Mais!

Pode ser que a empresa possua um modelo de documentação, se este for o seu caso, você adaptará o seu mapeamento. Pela minha experiência, normalmente a Empresa deixa o modelo de documentação por conta de quem está implantando a solução, se esse for o seu caso, seja simples e objetivo.

Uma vez documentado, é chegada a hora da sexta etapa, a **validação dos requisitos**. Neste ponto, você, os/as usuários/as e todas as partes envolvidas na solução, deverão realizar a validação final, antes de iniciar o desenvolvimento. É o alinhamento de expectativa para que o desenvolvimento possa ser iniciado.

Aqui tem ponto importante, que a Engenharia de Requisitos clássica não considera, até por conta de se tratar uma matéria antiga. Porém, atualmente e como já

exposto anteriormente, há de se entender que requisitos podem sofrer mudanças em curto espaço de tempo (Sim! Isso é frequente!) e, você precisa ser flexível e usar de sua experiência, para negociar mais uma vez.

Essa negociação consiste em avaliar o impacto da mudança, tanto no custo, quanto no prazo, quanto na prioridade do requisito, de modo que seja possível incluí-lo no desenvolvimento.

Pode acontecer de ser um ajuste muito impactante, porém muito importante e, impactar na priorização dos requisitos. O contrário também é uma verdade, ou seja, um ajuste de baixo (ou nenhum) impacto.

O importante nesta etapa é garantir que os requisitos que foram levantados, de fato, proverão respostas aos desafios mapeados.

A última etapa, na realidade, não é o fim da Engenharia de Requisitos, visto que se trata de uma etapa cíclica, chamada **Gestão dos requisitos**. Como o próprio nome

sugere, você precisará gerir cada requisito, monitorar se está sendo desenvolvido de acordo com o mapeado, se há algum problema que impeça de desenvolver o requisito, ser proativo no sentido de se comunicar com o/a usuário/a para verificar se há algum ajuste a ser feito (se houver, negocie conforme já explanado).

Lembre-se de versionar a sua especificação, para garantir rastreabilidade as alterações que foram sendo realizadas nos requisitos e para que haja um histórico que possa ser consultado em caso de dúvidas.

Como disse é uma etapa cíclica e precisa ser monitorada constantemente, sempre que possível voltando ao mapeamento dos requisitos e analisando-os, de forma a minimizar as inconsistências.

Engenharia de requisitos é essencial para o sucesso de qualquer desenvolvimento de *software*. Jamais inicie o desenvolvimento sem antes lograr êxito no estágio “Descobrir”.

Para dar uma agilidade ao seu projeto, lembre-se de dividi-lo em *releases* e módulos. Quando se trata de uma solução grande, com muitos departamentos envolvidos, diversos requisitos, se você esperar o mapeamento dos requisitos de todas as áreas, para validá-los e, conseqüentemente, iniciar o desenvolvimento, você poderá (e vai) ter um problema relacionado a agilidade, que poderá (e vai) impactar o seu cronograma de projeto, bem como acontecerá de você desenvolver requisitos desatualizados.

Portanto, quando fizer o estudo de viabilidade, faça uma classificação prioritária das áreas e levantamento dos requisitos conforme essa prioridade, isso te dará mais agilidade.

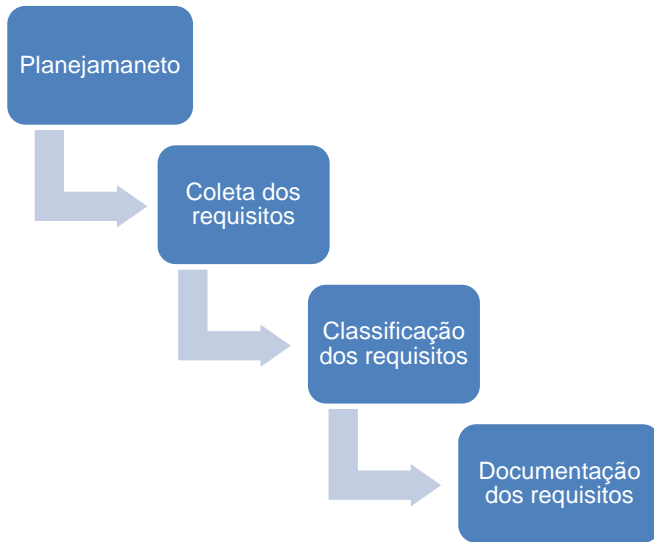
Outro fator que te dará agilidade, trata-se de conforme for validando um grupo de requisitos, vá avançando para o estágio “Definir”. Por este motivo que na Figura 1, você vai observar que o estágio “Definir” se inicia antes do término do estágio “Descobrir”.

É preciso ter muita atenção na gestão dos requisitos, de modo que você vá avançando por etapas, mas que não deixe de validar nenhum deles. Esta forma de trabalho te dará muito mais agilidade, mas também será necessário muito mais gestão.

Quem dita a cadência da Engenharia de Requisitos são as etapas de **Levantamento e Análise de requisitos** que, como mencionado, podem ser conduzidas ao mesmo tempo (inclusive eu te sugiro fazer dessa forma), para facilitar a compreensão pois, é ela quem vai te apresentar todos os requisitos essenciais para sua solução de Business Intelligence funcionar.

Para isso, existe um processo que você deve se atentar, para garantir um levantamento de requisitos assertivo, confiável, seguro e com a devida rastreabilidade. A figura a seguir ilustra esse processo e logo após explicaremos cada etapa, visando uma melhor compreensão.

**Figura 8** - Processos para levantamento de requisitos



Fonte: Autoria Própria

Uma vez que você segue este processo, seu trabalho fica muito mais elucidativo e te auxiliará na etapa seguinte, se for necessário, a negociação dos requisitos.

A primeira fase deste processo, te apresenta o **Planejamento**. Toda e qualquer atividade em um desenvolvimento de *software*, convém realizar um planejamento, por mais simples que seja, pois planejar-se é como utilizar um GPS (Sistema de Posicionamento Global) para chegar a um destino.

Nesta fase, você precisará:

- Identificar os/as usuários/as que você levantará os requisitos. São facilmente identificáveis, normalmente já estão envolvidos no projeto.
- Criar uma agenda com cada usuário/a, de forma que se crie um compromisso.
- Definir como será a coleta dos requisitos. Normalmente, utiliza-se a entrevista, porém pode ser através de um questionário online. Falaremos mais sobre tipos de coleta quando chegarmos nesta fase.
- Independentemente do tipo de coleta escolhido, defina as perguntas que realizará. Quando estiver

realizando o levantamento de requisitos, você já passou pelo estudo de viabilidade, já conhece os desafios que a solução precisa resolver, portanto tome-os como seu direcionamento para definir as perguntas, por exemplo:

- Desafio: Produtos que menos batem suas metas de vendas.
- Perguntas:
  - Atualmente, como que você identifica se um produto bateu a meta ou não?
  - Qual cálculo você utiliza?
  - Existe uma classificação de atingimento de metas? Até 70% é péssimo. Entre 71% e 90% é regular. Entre 91% e 99% é aceitável? Algo do tipo?
  - Quando avalia o Produto, está avaliando o SKU (Unidade de manutenção de estoque)?



- Você também avalia a meta de toda a hierarquia de produto?

Perceba que em apenas um desafio, podemos abrir algumas perguntas, o resultado destas perguntas você vai preencher um documento de levantamento de requisitos, documento este que vou te apresentar um modelo, mais adiante.

Definido seu planejamento e agenda alinhada com todos as pessoas envolvidas no levantamento de requisitos, chega-se a hora da **Coleta dos Requisitos**. Existem duas formas clássicas de se coletar:

- Entrevista: É a mais tradicional. É a que eu recomendo, pois contato direto com a pessoa vai ajudar a elucidação dos requisitos.
- Questionário: Pouco usado. Não recomendo, pois se torna muito frio, visto que você não tem contato com a pessoa. O questionário é enviado e a pessoa responde sozinha, porém já tive

experiências ruins, com respostas vagas, demora para responder, o que impacta o prazo e dúvidas.

Neste caso, minha sugestão sempre será que você siga pelas entrevistas pessoais, mesmo que seja remotamente.

Note que você terá perguntas abertas, ou seja, aquelas que o/a usuário/a tem de explicar, apresentar como ele trabalha. E teremos perguntas fechadas, ou seja, aquelas que requerem uma resposta objetiva, sem explicações.

Normalmente, eu sugiro começar por perguntas abertas, pois conforme o/a usuário/a vai explanando, podem surgir novos questionamentos e podem responder a outras perguntas que você mapeou. Além disso, a tendência é que se torne uma reunião mais leve, pois tanto a pessoa sendo entrevistada, quanto você, começam a se sentir mais à vontade.

Algumas vezes, você se deparará com um/a usuário/a mais tímido/a, que fala menos. Nesse caso, convém começar pelas perguntas fechadas, de modo a fazer com que a pessoa vá se soltando aos poucos.

Ponto muito importante a se atentar é que a entrevista **NÃO** deve ser mecanizada, conduza com leveza, suavidade, faça as perguntas como se fosse um papo e não um inquérito policial. O seu trabalho é fazer com que o/a usuário/a se sinta confortável, explique todos os requisitos essenciais e saia da reunião com o sentimento de que passou muito rápido.

Lembre-se que cada usuário/a representa um determinado grupo de requisitos, portanto mantenha o foco, evite perguntas relacionadas a outro/a usuário/a. Somente as faça se você identificou que de fato há uma correlação, afora isso, evite. **Mantenha o FOCO.**

Ainda na coleta dos requisitos, já avance para a próxima etapa que é a **Classificação dos Requisitos**. Busque obter informações sobre a prioridade dos requisitos, ou

seja, uma ordenação de implantação dos requisitos. Identifique também se algum requisito não ficou claro o suficiente, sane suas dúvidas no momento da entrevista, procure finalizar a entrevista com todas as perguntas respondidas. Porém, fique tranquilo se por algum motivo, surgirem dúvidas posteriores, afinal a **gestão dos requisitos** te dará oportunidade de saná-las.

Esforce-se para realizar um levantamento de requisitos em, no máximo, 60 minutos. Lembre-se que o/a usuário/a tem suas rotinas diárias na Empresa.

Uma vez encerrado o levantamento, você deverá **documentar os requisitos**. E para isso, vou te apresentar cada tipo de requisito e um modelo de preenchimento que vai apoiar o seu trabalho.

## **Requisitos da solução**

Ainda no estágio “Descobrir”, após a sua primeira entrevista, entendimento e alinhamento das necessidades do projeto é necessário transformá-las em requisitos funcionais, ou seja, tudo aquilo que o sistema precisa fazer para de fato funcionar para prover respostas que suportarão tomadas de decisão para os desafios mapeados.

O Levantamento de requisitos que você realizou com cada usuário/a, será registrado como Requisito Funcional.

## **Requisitos Funcionais**

Como dito anteriormente e reforçado neste momento, os requisitos funcionais serão coletados em entrevista e/ou questionário com os/as usuários/as e a principal finalidade destes requisitos, como o próprio nome sugere, é identificar quais os requisitos necessários para

suportar, na exata medida, tomadas de decisão relacionadas aos desafios que foram mapeados. Ou seja, são extremamente importantes, pois sem estes requisitos, sua solução será inutilizada.

Perguntas importantes e que facilitarão sua vida, tomando como premissa o cenário que estamos utilizando neste livro:

- Como você precisa analisar o “*Ranking* das lojas com mais vendas/m<sup>2</sup>”?
- Qual o procedimento para identificar o “*Ranking* das lojas com mais vendas/m<sup>2</sup>”?
- Qual periodicidade de análise dos “Produtos que não batem metas”?
- Qual o indicador que responde sobre os “Produtos que não batem metas”?
- Qual a regra de negócio (Cálculo) do indicador que calcula os “Produtos que não batem metas”?
- Como é feita a identificação da “área da loja com maior circulação de pessoas”?

- Como saberemos se a loja vende muito ou pouco, através do “*Ranking* das lojas com mais vendas/m<sup>2</sup>”? Como montar o semáforo de vendas/m<sup>2</sup>?
- Existe um indicador de atingimento de vendas?
- Como é calculado o indicador que diz se as lojas bateram suas metas de vendas?
- Qual a regra de negócio (cálculo) do indicador de atingimento de metas de vendas?
- Um vendedor pode vender em mais de uma região?
- Qual o indicador que analisa se o vendedor bateu sua meta de venda?
- Como esse indicador é calculado?

Uma vez que estas perguntas (e outras) forem respondidas, basta preencher o quadro abaixo, associando o Requisito funcional ao Desafio (Tabela 1.0).

**Tabela 2 – Modelo para levantamento de Requisitos Funcionais.**

Cod Nec	Cod Req	Requisito	Regra	Granularidade
D01	<b>RF01</b>	Possibilitar o cálculo das vendas/m <sup>2</sup> loja a loja	Total de vendas por loja / M2 da respectiva loja	Mensal e Cidades
D01	<b>RF02</b>	Possibilitar o ranqueamento das lojas pelo indicador Venda/m <sup>2</sup>	Ranquear lojas. Considerar todos os produtos.	Mensal e Cidades
D02	<b>RF03</b>	Possibilitar a análise de atingimento de vendas	(Total Venda por Produto/Meta Produto)*100	Mensal e Loja



Metodologia e Engenharia de Requisitos para Projetos de Business Intelligence

		(R\$) por produto, segundo suas metas.		
D02	<b>RF04</b>	Possibilitar o ranqueamento do atingimento de vendas (R\$) por produto, segundo suas metas.	Ranquear Produtos considerando a regra do RF03	Loja, Mensal
D03	<b>RF05</b>	Possibilitar a análise de atingimento de vendas (R\$) por vendedor, segundo suas metas.	$(\text{Total Venda por Vendedor/Meta Vendedor}) * 100$	Mensal

## Metodologia e Engenharia de Requisitos para Projetos de Business Intelligence

D03	<b>RF06</b>	Possibilitar o ranqueamento dos vendedores pelo indicador % atingimento dentro da sua Região	Ranquear vendedores, segundo seu atingimento.	Mensal
D03	<b>RF07</b>	Possibilitar o ranqueamento dos vendedores pelo total vendido em relação a todos os vendedores do Brasil	Ranquear vendedores, segundo seu volume de vendas	Tempo e Loja

## Metodologia e Engenharia de Requisitos para Projetos de Business Intelligence

D01	<b>RF08</b>	Possibilitar a análise de crescimento (ou Redução) de vendas/m² de cada loja, entre o ano atual e ano anterior, mês a mês.	Vendas/m² do mês do ano atual – Vendas/m² do mês do ano anterior	Cidades, loja e sku (stock keeping unit)
D02	<b>RF09</b>	Disponibilizar um dashboard de “Heróis e Vilões” por produto vendido, considerando 20 produtos mais vendidos para Heróis e os 20 menos	<p>Criar um dashboard com 2 gráficos de barras:</p> <p>Heróis: considerar os 20 melhores produtos com total valor vendido</p>	Mensal, cidades, loja

# Metodologia e Engenharia de Requisitos para Projetos de Business Intelligence

		vendidos para Vilões.	Vilões: considerar os 20 piores produtos com total valor vendido (desconsiderar produtos inativos e sazonais)	
D03	<b>RF10</b>	Disponibilizar uma análise de vendedor x produto	<p>Criar um Dashboard com uma tabela, com as colunas:</p> <p>1 – Vendedor</p> <p>2 – Produto</p>	Mensal, Cidades, loja

			3 – Valor total vendido	
--	--	--	----------------------------	--

Fonte: Autoria própria

Quando chegarmos em matriz de necessidades, entenderemos para que serve cada coluna preenchida e como utilizar essas informações.

## Requisitos Não Funcionais

Tão importantes quanto os requisitos funcionais, os não funcionais têm por objetivo dizer como que a solução deve se comportar no relacionamento e interação do/a usuário/a com a tecnologia, com a sua solução, como por exemplo: desempenho, usabilidade, confiabilidade,

segurança, disponibilidade, manutenção, tecnologias envolvidas entre outros.

Os requisitos não funcionais podem inviabilizar a utilização de sua solução, mesmo que todos os requisitos funcionais estejam sendo atendidos. Por exemplo:

A solução que você acabou de desenvolver, precisa ser acessada pelos/as gestores/as de todas as unidades, porém cada gestor/a só poderá analisar os dados da sua unidade, ou seja, você precisa implantar um processo de segurança das informações, para garantir o nível de acesso, segundo o perfil.

Acontece que a versão da tecnologia que você utilizou para desenvolver, não dá suporte à implantação de segurança e você não conseguirá realizar este processo.

Logo, sua solução não atende a um requisito não funcional e, por este motivo pode inviabilizar o seu uso.

Para identificar os requisitos não funcionais, você também se utilizará das entrevistas e/ou questionários e uma vez identificados, preencha a tabela a seguir:

**Tabela 3 – Modelo para levantamento de requisitos não funcionais**

Cód.	Req. Não Funcional	Obs.
RNF01	Usabilidade	Os dashboards devem funcionar em tablets e smartphones android e ios.
RNF02	Usabilidade	Os usuários só poderão utilizar a solução após treinamento de 8 horas.

<b>RNF03</b>	Confiabilidade	A solução deve ter alta disponibilidade. Expectativa de 99% do tempo.
<b>RNF04</b>	Segurança	Os usuários somente terão acesso as informações de suas respectivas lojas.
<b>RNF05</b>	Segurança	Diretores e presidente terão acesso a todas as informações, sem restrições

Fonte: Autoria própria

## Matriz de Necessidades

Após as reuniões para definição das necessidades, dos requisitos funcionais e não funcionais, chega-se o momento de “desenhar” a sua matriz de necessidades.



A matriz de necessidades de um projeto de *business intelligence* utilizado por mim, nos projetos que atuo e apresentada aqui segue a dinâmica apresentada por Ralph Kimball (2002), que diz que a Matriz de Necessidades é uma ferramenta usada para criar, documentar e comunicar a arquitetura do negócio, bem como definir o modelo de dados.

Aqui neste livro, trataremos apenas de uma parte da Matriz de Necessidades, pois é um assunto extremamente extenso e que caberia em um único livro. Para o propósito deste, o que será explanado aqui te atenderá perfeitamente e gerará um resultado incrível.

Como já dito anteriormente, a matriz de necessidades te ajudará na definição do modelo de dados (veremos mais detalhadamente a frente), por isso que neste momento é importante contextualizarmos alguns conceitos.

- Métrica: É tudo aquilo que é mensurável, por exemplo:

- Km
  - Kg
  - M<sup>2</sup>
  - M<sup>3</sup>
  - R\$
  - Quantidade
  - Volume
  - % de atingimento de vendas
  - Vendas/M<sup>2</sup>
  - Entre outras diversas medidas.
- Dimensão: É tudo aquilo que interferirá diretamente na medida, por exemplo:
    - % de atingimento de vendas por produto
      - Televisão teve 90% de atingimento de vendas
      - Cafeteira teve 105% de atingimento de vendas
      - Liquidificador teve 60% de atingimento de venda

- Perceba que a métrica é a mesma (% de atingimento de vendas), porém o seu valor varia de acordo com o tipo de produto, logo “Produto” é uma dimensão
- Vendas/m<sup>2</sup> por loja
  - Loja 1 realizou R\$ 1000/m<sup>2</sup> em vendas
  - Loja 2 realizou R\$ 800/m<sup>2</sup> em vendas
  - Perceba que a métrica é a mesma (Venda/m<sup>2</sup>), porém o seu valor varia de acordo com a Loja, logo “Loja” é uma dimensão
- Fato: É o cruzamento da dimensão com a métrica.
  - Televisão teve 90% de atingimento de vendas, na Loja 1
  - Cafeteira realizou R\$ 100/m<sup>2</sup>, na Loja 2
  - O Fato é a verdade apresentada pelos dados.

Entendido o que são métricas, dimensões e fato, partiremos para a Matriz de Necessidades, seguindo o modelo de Ralph Kimball (2002), conforme ilustra a figura a seguir:

**Figura 9** – Modelo da Matriz de Necessidades, proposto por Ralph Kimball

	Date	Customer	Product	Rate Plan	Sales Channel	Service Line #	Switch	Vendor	GL Account	Organization	Employee	Service Call Type	Service Call Status
Customer Billing	X	X	X	X	X	X							
Call Detail Traffic	X	X	X	X		X	X						
Purchasing	X		X					X	X	X			
Distributor Inventory	X		X		X			X		X			
Channel Sales	X	X	X	X	X	X		X					
Service Calls	X	X	X							X	X	X	X
Repair Items	X	X	X							X	X	X	X

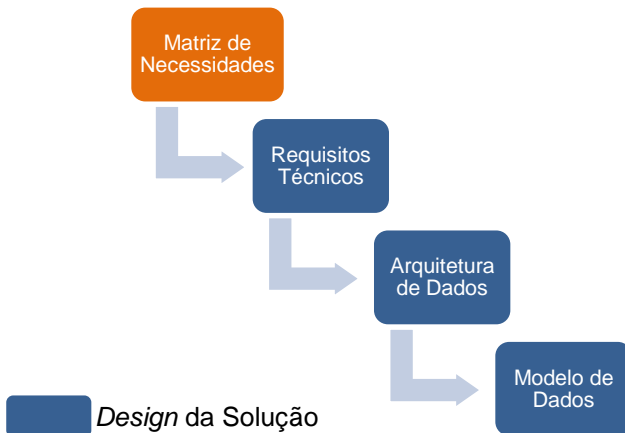
Fonte: The Data Warehouse Toolkit (KIMBALL, 2002, p.219)

A proposta deste modelo é muito simples e vai te ajudar no estágio de *Design* da Solução. Observe que o modelo apresenta linhas, onde estão descritas as métricas; colunas, onde estão descritas as dimensões; e as células, que serão marcadas quando o cruzamento da

dimensão com a métrica fizer sentido e, consequentemente representar um fato.

A Matriz de Necessidades deverá ser preenchida a “quatro mãos”, ou seja, entre você e o/a usuário/a, de forma a garantir que todos os cruzamentos essenciais para o seu projeto, estejam qualificados. Lembrando que a Matriz de Necessidades será a base para o Design da Solução e do Modelo de Dados.

**Figura 10** – Fases do levantamento de requisitos para uma solução de *Business Intelligence*



Fonte: Autoria própria

Recomenda-se que você utilize qualquer tecnologia de planilhas eletrônicas, como o *Microsoft Excel*, *Google Planilhas*, *OpenOffice Calc*.

A Matriz de Necessidades utiliza como insumo os requisitos funcionais, que já foram apresentados anteriormente.

O Primeiro passo é abrir a sua tabela de requisitos funcionais. Veja o primeiro requisito funcional:

**Figura 11** – Análise da descrição do requisito (RF01)

D01	RF01	Possibilitar o cálculo das vendas/m <sup>2</sup> loja a loja	Total de vendas por loja / M2 da respectiva loja	Mensal e Cidades
-----	------	--	--	------------------

Fonte: Autoria própria

Na descrição do requisito, é possível identificar alguns pontos relevantes para nossa Matriz:

- Métricas:
  - Vendas

- M<sup>2</sup>
- Dimensão
  - Loja

Com essas informações, iniciaremos o desenvolvimento da Matriz de necessidades, seguindo o modelo proposto por Ralph Kimball (2002).

**Figura 12** – Primeira versão da matriz de necessidades

	Loja
Vendas	X
m <sup>2</sup>	X

Fonte: Autoria própria

Pronto, já temos a primeira versão da nossa Matriz de Necessidades.

Agora, vamos analisar o texto da coluna “Regra”, do mesmo requisito, conforme imagem a seguir:

**Figura 13 – Análise da Regra (RF01)**

D01	RF01	Possibilitar o cálculo das vendas/m <sup>2</sup> loja a loja	Total de vendas por loja / M2 da respectiva loja	Mensal e Cidades
-----	------	--	--	------------------

Fonte: Autoria própria

Neste texto, encontramos a regra de cálculo, a regra de negócio, ou seja, uma métrica. Também apresenta a dimensão “Loja”, porém esta já se encontra na nossa matriz.

- Métrica:
  - Total de Vendas/m<sup>2</sup>

Alguns autores não colocam a regra do negócio na matriz de necessidades. Eu gosto de colocar, por 2 motivos:

1. A matriz de necessidade é uma documentação oficial do projeto, logo quanto mais completa for, melhor será, principalmente em um futuro, onde



seja necessário realizar alterações, exclusões, inclusões etc.

2. A matriz de necessidade é apresentada e entregue para os/as usuários/as, profissionais que muitas vezes (a grande maioria) não são da área de Tecnologia da Informação, logo uma matriz bem detalhada facilitará o entendimento.

Desta forma, adicionaremos a matriz de necessidades a métrica apresentada na coluna Regra, e o resultado será o seguinte:

**Figura 14** – Segunda versão da matriz de necessidades

	Loja
Vendas	X
m <sup>2</sup>	X
Total de Vendas/m <sup>2</sup>	X

Fonte: Autoria própria

Lembrando de “marcar um X” no cruzamento com a dimensão “Loja”, pois a regra diz que este cálculo deverá ser apresentado por Loja.

Seguindo com a análise do requisito, chegamos a coluna “Granularidade” e essa apresenta o seguinte texto:

**Figura 15 – Análise da granularidade (RF01)**

D01	RF01	Possibilitar o cálculo das vendas/m <sup>2</sup> loja a loja	Total de vendas por loja / M2 da respectiva loja	Mensal e Cidades
-----	------	--	--	------------------

Fonte: autoria própria

Primeiro, vamos contextualizar “Granularidade”, que se trata simplesmente do menor grão que métricas podem ser analisadas. No nosso exemplo temos duas granularidades representadas: Mensal e Cidades. Isso quer dizer que quando a medida Total de Vendas/m<sup>2</sup> for analisada no tempo, deve-se considerar o Mês, como

seu menor grão; e quando analisada por Região, deve-se considerar a Cidade como seu menor grão.

Logo, é possível identificar mais duas dimensões:

- Tempo (Para possibilitar a análise mensal)
- Região (Para possibilitar a análise regional)

Identificado isso, vamos a preenchimento da matriz de necessidades, que terá como resultado:

**Figura 16** – Terceira versão da matriz de necessidades

	Loja	Tempo (Grão: Mensal)	Região (Grão: Cidade)
Vendas	X	X	X
m <sup>2</sup>	X	X	X
Total de Vendas/m <sup>2</sup>	X	X	X

Fonte: Autoria própria

Perceba, que com apenas 1 (um) requisito funcional, nossa matriz de necessidades já começa a ganhar volume.

Vamos analisar o seguinte requisito funcional, seguindo os mesmos passos anteriores, visando uma melhor compreensão de como se estruturar uma matriz de necessidades.

**Figura 17** – Análise da descrição do requisito (RF02)

D01	<b>RF02</b>	Possibilitar o ranqueamento das lojas pelo indicador Venda/m <sup>2</sup>	Ranquear lojas. Considerar todos os produtos.	Mensal e Cidades
-----	-------------	---	--	------------------

Fonte: Autoria própria

Na descrição do requisito, é possível identificar uma métrica e uma dimensão:

- Métrica
  - Venda/m<sup>2</sup>
- Dimensão
  - Loja

Tanto a métrica, quanto a dimensão já estão em nossa matriz de necessidades, logo não há ajustes a serem feitos e partimos para o texto da coluna “Regra”:

**Figura 18 – Análise da Regra (RF02)**

D01	RF02	Possibilitar o ranqueamento das lojas pelo indicador Venda/m <sup>2</sup>	Ranquear lojas. Considerar todos os produtos.	Mensal e Cidades
-----	------	---	--	------------------

Fonte: Autoria própria

Nesta coluna, é possível identificar a regra do negócio, ou seja, um ranqueamento a partir da medida “Venda/m<sup>2</sup>” e duas dimensões:

- Lojas
- Produtos

A dimensão “Loja” já se encontra na nossa matriz de necessidades, precisaremos adicionar somente a

dimensão “Produto” e o resultado será conforme a imagem a seguir:

**Figura 19** – Quarta versão da matriz de necessidades

	Loja	Tempo (Grão: Mensal)	Região (Grão: Cidade)	Produto
Vendas	X	X	X	X
m <sup>2</sup>	X	X	X	X
Total de Vendas/m <sup>2</sup>	X	X	X	X

Fonte: Autoria própria

Agora vamos analisar o texto da coluna “Granularidade”, para identificar possíveis ajustes na matriz:

**Figura 20** – Análise da granularidade (RF02)

D01	<b>RF02</b>	Possibilitar o ranqueamento das lojas pelo indicador Venda/m <sup>2</sup>	Ranquear lojas. Considerar todos os produtos.	Mensal e Cidades
-----	-------------	---	--	------------------

Fonte: Autoria própria

Ao analisar o texto, concluímos que a dimensão “Tempo” e a dimensão “Região”, que representam estas granularidades, já estão na matriz.

Vamos fazer mais um requisito funcional juntos, para vermos algumas variações que podem acontecer na matriz de necessidades:

**Figura 21** – Análise da descrição do requisito (RF03)

D02	<b>RF03</b>	Possibilitar a análise de atingimento de vendas (R\$) por produto, segundo suas metas.	(Total Venda por Produto/Meta Produto)*100	Mensal e Loja
-----	-------------	--	--	---------------

Fonte: Autoria própria

A analisar o texto da coluna “Descrição”, podemos identificar o seguinte:

- Métricas
  - Vendas
  - Metas
- Dimensão
  - Produto

A métrica “Vendas” e a dimensão “Produto” já estão em nossa matriz, neste caso adicionaremos a métrica “Metas”, para obtermos o seguinte resultado:

**Figura 22 –** Quinta versão da matriz de necessidades

	Loja	Tempo (Grão: Mensal)	Região (Grão: Cidade)	Produto
Vendas	X	X	X	X
m <sup>2</sup>	X	X	X	
Total de Vendas/m <sup>2</sup>	X	X	X	
Metas				X

Fonte: Autoria própria

Repare que marcamos o cruzamento de “Vendas” e “Metas” com a dimensão “Produto”. Caso esteja se



perguntando o motivo, isso foi feito pois o texto só faz referência ao cruzamento de “Vendas” e “Metas” com “Produto”. Pode ser que analisando outros detalhes dos requisitos, outros cruzamentos sejam marcados, portanto vamos seguir na análise deste requisito:

**Figura 23 – Análise da Regra (RF03)**

D02	RF03	Possibilitar a análise de atingimento de vendas (R\$) por produto, segundo suas metas.	(Total Venda por Produto/Meta Produto)*100	Mensal e Loja
-----	------	--	--	---------------

Fonte: Autoria própria

Ao analisar o texto da coluna “Regra”, identificamos o seguinte:

- Métrica
  - (Total de Venda/Meta)\*100

- Dimensão
  - Produto

A dimensão “Produto” já se encontra na nossa matriz. Será necessário a inclusão da métrica, para obtermos o seguinte resultado:

**Figura 24 – Sexta versão da matriz de necessidades**

	Loja	Tempo (Grão: Mensal)	Região (Grão: Cidade)	Produto
Vendas	X	X	X	X
m²	X	X	X	
Total de Vendas/m²	X	X	X	
Metas				X
(Total de Venda/Meta)*100				X

Fonte: Autoria própria

E por fim, vamos analisar o texto da coluna granularidade, conforme representado e destacado na imagem a seguir:

**Figura 25 – Análise da granularidade (RF03)**

D02	RF03	Possibilitar a análise de atingimento de vendas (R\$) por produto, segundo suas metas.	(Total Venda por Produto/Meta Produto)*100	Mensal e Loja
-----	------	--	--	---------------

Fonte: Autoria própria

Podemos observar duas dimensões: Tempo e Loja que, por sua vez, já se encontram na nossa matriz, porém há um ponto de atenção a se observar, pois se há a necessidade de se exibir métricas nestas granularidades, então será necessário marcar os cruzamentos destas dimensões com a métrica em questão.

Fazendo isso, obteremos o resultado, conforme ilustra a imagem a seguir:

**Figura 26** – Sétima versão da matriz de necessidades

	Loja	Tempo (Grão: Mensal)	Região (Grão: Cidade)	Produto
Vendas	X	X	X	X
m <sup>2</sup>	X	X	X	
Total de Vendas/m <sup>2</sup>	X	X	X	
Metas	X	X		X
(Total de Venda/Meta)*100	X	X		X

Fonte: Autoria própria

Ao finalizar todos os requisitos funcionais, a nossa Matriz de Necessidades ficará da seguinte forma:

**Figura 27** – Versão final da análise dos três primeiros requisitos

	Loja	Tempo (Grão: Mensal)	Região (Grão: Cidade)	Produto (Grão: SKU)	Vendedor
Vendas	X	X	X	X	X
m <sup>2</sup>	X	X	X		
Total de Vendas/m <sup>2</sup>	X	X	X		
Metas	X	X		X	X
(Total de Venda/Meta)*100	X	X		X	X
Vendas/m <sup>2</sup> (ano atual) – Vendas/m <sup>2</sup> (ano anterior)	X	X	X	X	

Fonte: Autoria própria

Requisitos funcionais que são referentes a criação de gráficos, tabelas ou qualquer outro objeto visual, não entrará na matriz de necessidade.

Os Requisitos não funcionais também não devem estar descritos na matriz de necessidade.

Apenas certifique-se de que o objeto visual que está sendo pedido contém métricas e dimensões que estão presentes na Matriz de Necessidade.

- Se estiverem, ótimo.
- Se não estiverem, informe ao/a usuário/a para que vocês possam encontrar uma solução.

Uma vez finalizada a Matriz de Necessidades, perceba que nas colunas, nós temos as dimensões; nas linhas, nós temos as métricas e nas células, nós temos o fato (cruzamentos entre dimensões e fatos). Estas informações serão necessárias para o estágio de “Definir - Design da Solução”.

É claro que, sempre que houver dúvidas, retorne ao seu usuário-chave e converse com o mesmo, objetivando saná-la.

## **Definir – Design da Solução**

No estágio “Definir” iniciaremos um trabalho de extrema importância e sucesso para o seu projeto, trata-se de realizar a aderência das necessidades identificadas no estágio “Descobrir” – Requisitos funcionais, não funcionais e Matriz de Necessidades – com a sua solução proposta.

Neste estágio vamos detalhar quais as premissas para sua solução funcionar corretamente e vamos alinhar como a tecnologia vai prover solução para as necessidades dos seus clientes.

## **Requisitos Técnicos**

São os requisitos necessários para que os requisitos funcionais e não funcionais possam ser implantados, ou seja como aderir o negócio a tecnologia.

Neste ponto, as entrevistas acontecerão com o time de Tecnologia da Informação (TI) do seu cliente, porém você já deve saber as premissas de tecnologia para sua solução funcionar, por exemplo:

Vamos imaginar que você utilizará o *Power BI* para implantar sua solução. Será necessário saber que para o *Power BI Desktop* funcionar, você precisará de um computador com:

- Sistema operacional Windows
- 8 gb de ram
- 10 gb de espaço em disco
- Acesso a internet para consumir dados da nuvem

Caso você tenha que realizar a instalação do Power BI Report Server, você deve saber que as premissas são:

- ***Hardware***



- Disco Rígido (HD)
  - Instalação do *Power BI Report Server*. 1GB
  - Base de dados de relatórios: Volumetria suficiente para armazenar os dados utilizados na PoC.
- Memória
  - Mínimo: 1GB
  - Recomendado pela Microsoft: 4GB
- Velocidade do Processador
  - Mínimo: 1.4GHz
  - Recomendado pela *Microsoft*. 2.0 GHz ou superior
- Tipo do processador
  - x64: AMD Opteron, AMD Athlon 64, Intel Xeon com suporte a Intel EM64T, Intel Pentium IV com suporte a EM64T ou superiores

- ***Somente há suporte para a instalação do Power BI Report Server em processadores x64.***

- **Software**

- .Net Framework
  - Versão 4.6
  - Windows 10 e Windows Server 2012 R2 requerem a atualização **KB2919355** antes da instalação do .NET Framework 4.6.
- Sistema Operacional
  - Windows Server 2016 Datacenter
  - Windows Server 2016 Standard
  - Windows Server 2012 R2 Datacenter
  - Windows Server 2012 R2 Standard
  - Windows Server 2012 R2 Essentials

- Windows Server 2012 R2 Foundation
- Windows Server 2012 Datacenter
- Windows Server 2012 Standard
- Windows Server 2012 Essentials
- Windows Server 2012 Foundation
- Windows 10 Home
- Windows 10 Professional
- Windows 10 Enterprise
- Windows 8.1
- Windows 8.1 Pro
- Windows 8.1 Enterprise
- Windows 8
- Windows 8 Pro
- Windows 8 Enterprise
- Banco de dados para armazenamento dos relatórios do *Power BI Report Server*. O *SQL Server* é usado para hospedar os bancos de dados do *Power BI Report*

*Server*. A instância de Banco de Dados do  
SQL Server **pode ser local ou remota**.

- SQL Server 2017
- SQL Server 2016
- SQL Server 2014
- SQL Server 2012
- SQL Server 2008 R2
- SQL Server 2008

Acima listei alguns exemplos, mas poderíamos acrescentar vários, por exemplo se o seu banco de dados ficar na nuvem, você precisará mapear alguns pontos importantes:

- Qual fornecedor de nuvem? *Microsoft Azure, Google Cloud, Amazon Web Services*, outro?
- Quais serviços você utilizará?
- Serão somente serviços ou precisará de uma infraestrutura em nuvem?
- Como será a integração dos dados locais com a nuvem?

É importante que você conheça e esteja familiarizado com a tecnologia que implantará, pois uma vez que você informa as premissas para sua tecnologia funcionar corretamente, é necessário identificar como você atenderá os requisitos funcionais e não funcionais já identificados.

Para os requisitos técnicos, utilizaremos a Matriz de Necessidades como insumo, de modo a identificar junto a TI do cliente a origem das dimensões e métricas, mapeando assim todas as fontes de origem de cada dado.

Mais um trabalho que será feito a “quatro mãos”, ou seja, entre você e o profissional de TI do seu cliente. Com a Matriz de Necessidades em mãos, você pedirá para que este profissional te informe a localização de cada métrica e dimensão:

**Figura 28 – Matriz de necessidades para aderência a tecnologia**

	Loja	Tempo (Grão: Mensal)	Região (Grão: Cidade)	Produto (Grão: SKU)	Vendedor
Vendas	X	X	X	X	X
m²	X	X	X		
Total de Vendas/m²	X	X	X		
Metas	X	X		X	X
(Total de Venda/Meta)*100	X	X		X	X
Vendas/m² (ano atual) – Vendas/m² (ano anterior)	X	X	X	X	

Fonte: Autoria própria

As informações coletadas deverão ser preenchidas na tabela abaixo. Esta tabela é um modelo feito por mim, porém fique à vontade para organizá-la da melhor forma. Eu preencho primeiro as métricas e, em seguida, as dimensões.

As métricas que representam “Regras”, você deve apenas se atentar na composição da regra, ou seja, identificar as métricas que compõem o cálculo, para garantir o mapeamento de todas.

**Tabela 4 – Modelo para levantamento de Requisitos  
Técnicos, associados aos requisitos funcionais**

Req Téc	Origem	Local	Objetos	Colunas
<b>RT01</b>	Banco de dados SQL Server	Servidor: SRV01	TB_Movimentacao	VLR_Venda
<b>RT02</b>	Banco de dados SQL Server	Servidor: SRV01	TB_Loja	M2
<b>RT03</b>	Planilha Excel	//srv02/financeiro/	metas.xlsx (Aba: DefinicaoMetas)	VLR_Meta
<b>RT04</b>	Banco de dados SQL Server	Servidor: SRV01	TB_Loja	Todas
<b>RT05</b>	Banco de dados SQL Server	Servidor: SRV01	TB_Movimentacao	Dt_Movimentacao

## Metodologia e Engenharia de Requisitos para Projetos de Business Intelligence

<b>RT06</b>	Banco de dados SQL Server	Servidor: SRV01	TB_Localidade	Todas
<b>RT07</b>	Banco de dados SQL Server	Servidor: SRV01	TB_Produto	Todas
<b>RT08</b>	Banco de dados SQL Server	Servidor: SRV01	TB_Vendedor	Todas

Fonte: Autoria própria

Note que no exemplo acima fica claro que o trabalho neste ponto é identificar de onde os dados são originários, ou seja, o ponto zero do dado, de modo que este seja o seu insumo, para a construção das métricas, indicadores e dimensões que atenderão seus requisitos funcionais, baseados na Matriz de Necessidades e,



consequentemente os desafios e oportunidades do/a usuário/a.

Agora, precisamos buscar aderência tecnológica para os requisitos não funcionais.

Deve-se atentar para o modelo de levantamento de requisitos não funcionais e, ainda junto com o time de TI do seu cliente, necessita-se buscar como resolvê-los.

Na verdade, como prover aderência da tecnologia com o requisito, utilizando a tecnologia disponível ou se necessário indicando se haverá necessidade de aquisição.

Lembre-se! Menos é mais! Esforce-se para utilizar a tecnologia já disponível em seu cliente, apenas sugira aquisições, se realmente elas forem imprescindíveis.

Para fazer isto, preencha o quadro abaixo, dando sequência aos requisitos técnicos já mapeados.

**Tabela 5 - Modelo para levantamento de Requisitos Técnicos, associados aos requisitos não funcionais**

## Metodologia e Engenharia de Requisitos para Projetos de Business Intelligence

Req Téc	Req N Func	Solução
<b>RT10</b>	RNF01	<p>Celulares e tablets IOS a partir da versão 9</p> <p>Celulares e tablets Android a partir da versão 8</p>
<b>RT11</b>	RNF02	<p>Após a fase de homologação, os usuários deverão participar do treinamento de 8 horas sobre Business Intelligence e mais 16 horas sobre Power BI.</p> <p>Cada turma deverá ter no máximo 15 profissionais.</p>
<b>RT12</b>	RNF03	<p>Existirão 2 servidores de produção, onde 1 será a réplica do outro.</p> <p>Os servidores estarão em endereços diferentes.</p>

## Metodologia e Engenharia de Requisitos para Projetos de Business Intelligence

		<p>Todo dia, às 02:00 AM haverá uma cópia fiel do servidor 1 para o servidor 2.</p> <p>Caso ocorra algum problema no servidor 1, o servidor 2 deverá ser ativado automaticamente.</p> <p>Além disso, durante o dia haverá 3 backups: 06:00 AM, 14:00 PM, 00:00 AM.</p>
<b>RT13</b>	RNF04	<p>Serão criados datamarts no SQL Server por loja, onde somente os usuários nomeados daquela loja terão acesso de leitura.</p> <p>O Power BI será controlado pelo Azure Active Directory e os usuários somente terão acesso de leitura as informações de suas respectivas lojas.</p>

		Diretores e presidente estarão no grupo “executivos” e terão acesso total aos dados, sem restrições.
--	--	--

Fonte: Autoria Própria

## Arquitetura de dados

A Arquitetura de dados irá dizer qual o caminho que o dado percorrerá até chegar ao dashboard e aqui o apoio será a Tabela de Requisitos Técnicos, afinal você já mapeou todas as fontes de dados, basta apenas apresentar o caminho.

Aqui não existe receita de bolo e cada caso deve ser avaliado, conforme os requisitos já identificados. Por favor, não gere complexidade onde não há necessidade. O que quero dizer é que a melhor arquitetura é aquela

que atenderá e resolverá as necessidades de seu usuário ou cliente.

Com a arquitetura bem desenhada e definida, é possível garantir o rastreamento do dado, bem como qualidade, segurança, confiabilidade e, um dos pontos mais importantes, garantir uma central única da verdade, ou seja, o dado que se encontra nesta arquitetura é o dado de maior confiabilidade, garantindo assim, a Governança (Gestão) do dado.

Antes de detalhar alguns tipos de arquiteturas e depois definirmos a nossa, precisamos contextualizar alguns pontos essenciais para o sucesso.

Quando fazemos o mapeamento dos requisitos técnicos, identificamos as fontes de dados das métricas e dimensões, por muitas vezes são fontes distintas, por exemplo: Banco de Dados SQL Server e Excel.

Neste caso, recomenda-se unificar em uma “Central Única da Informação”, esta por sua vez, é conhecida por

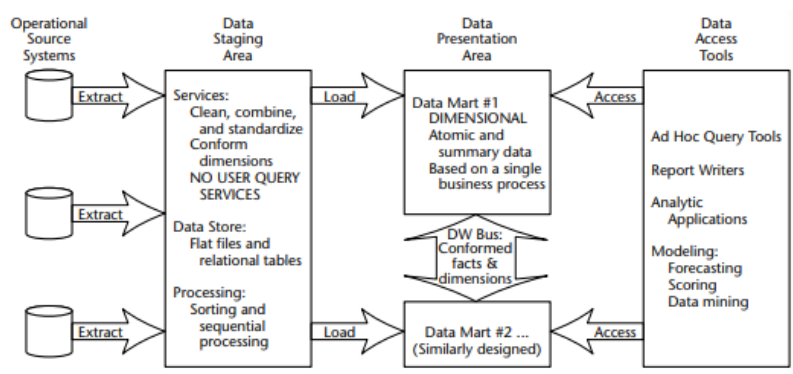
*Data Warehouse*. E, para isso fundamentaremos nos dois principais autores sobre ao assunto: Ralph Kimball (2002) e Bill Inmon (2005).

Kimball, vai nos apresentar alguns objetivos do Data Warehouse, são eles:

- O *data warehouse* deve tornar as informações de uma organização facilmente acessíveis,
- O *data warehouse* deve apresentar as informações da organização de forma consistente,
- O *data warehouse* deve ser adaptável e resiliente às mudanças,
- O *data warehouse* deve ser seguro e proteger nossos ativos de informação,
- O *data warehouse* deve servir como base para uma melhor tomada de decisão,
- A corporação deve aceitar o *data warehouse*, se ele for considerado bem sucedido, como sua fonte de informação.

Como proposição de arquitetura de dados, para se construir um *Data Warehouse*, Kimball (2002) vai apresentar o seguinte:

**Figura 29 – Arquitetura de dados proposta por Kimball**



Fonte: Elementos básicos de uma data warehouse (KIMBALL, 2002, p. 7)

Vamos entender as etapas propostas na arquitetura de Kimball:

- Sistemas Operacionais de Origem: Trata-se das fontes de dados mapeadas no levantamento de requisitos. Elas serão o insumo para construção de seu *Data Warehouse*.
- Extração: É o processo que indica que os dados mapeados no levantamento de requisitos serão extraídos de cada uma de suas fontes de dados de origem e serão tratados (Transformados) na etapa posterior.
- Área de Estágio de Dados: É um ambiente, que pode ser um banco de dados, onde os dados serão tratados, por exemplo: Se extraímos dados por dia e o requisito deseja que o período seja por mês, é neste momento que proveremos essa transformação, para posterior carga na próxima etapa.
- Área de apresentação dos dados: Uma vez que todas as transformações são realizadas, os dados são carregados aqui, que podem ser bancos de dados ou tecnologias para específicas para este



tipo de armazenamento. Esta camada divide-se em 2 momentos:

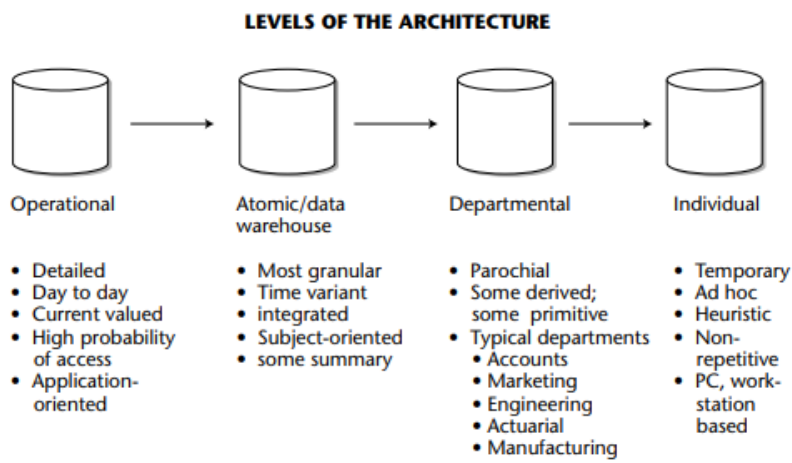
- *Data Marts*: É onde são carregados os dados separados assunto. Por exemplo: Dados relacionados a marketing serão carregados em um Data Mart de Marketing. Dados relacionados a Recursos Humanos serão carregados em um Data Mart de Recursos Humanos.
- *Data Warehouse*: Que se trata da junção de todos os dados, provenientes de todos os Data Marts. Todos os assuntos ficam interligados dentro do Data Warehouse
- Ferramentas de acesso a dados: Trata-se de como os dados que estão nos *Data Marts* e *Data Warehouse* serão consultados, visualizados. Por exemplo, a criação de um *Dashboard* em *Power BI*.

Inmon (2005) nos traz uma abordagem semelhante, guardada uma pequena diferença na abordagem da construção da arquitetura. Ele também vai nos apresentar alguns pontos importantes sobre o *Data Warehouse*:

- O *data warehouse* contém dados integrados, oriundos dos sistemas operacionais,
- O *data warehouse* armazenará o histórico dos dados,
- O *data warehouse* armazenará dados derivados dos dados primitivos, oriundos dos sistemas transacionais,
- O *data warehouse* armazenará os dados de todos os departamentos da Empresa.

Como proposição para a construção da arquitetura de dados, Inmon (2005) vai nos apresentar sua abordagem, conforme imagem a seguir:

**Figura 30** - Arquitetura de dados proposta por Inmon



Fonte: Níveis da arquitetura (INMON, 2005, p. 16)

Perceba que, apesar do desenho ser diferente, a abordagem de Inmon (2005) é bem semelhante a de Kimball (2002), onde:

- Operacional: São os dados que foram mapeados no levantamento de requisitos e apesar de não estar representado graficamente. A “Seta” informa que há uma extração, transformação e

carga de dados, seguindo o mesmo processo apresentado por Kimball (2002).

- *Data warehouse*: Os dados são carregados diretamente no *Data Warehouse*, de forma integrada. Normalmente utilizam-se bancos de dados ou tecnologia própria para isso.
- Departamental (*Data Mart*): Os dados são divididos por assuntos departamentais, ou seja, dados de *Marketing*, torna-se o *Data Mart* de *Marketing*. Dados de Recursos Humanos, torna-se o *Data Mart* de Recursos Humanos. Normalmente utilizam-se bancos de dados ou tecnologia própria para isso.
- Individual (Acesso ao Dado): Trata-se de como os dados que estão nos *Data Marts* serão consultados, visualizados. Por exemplo, a criação de um Dashboard em Power BI.

Ao analisar as duas propostas, percebemos a diferença em um ponto específico. Kimball (2005) sugere que os dados, uma vez que já foram transformados, sejam carregados nos *data marts* departamentais e só depois serem carregados no *Data Warehouse*. Para Inmon (2002), este processo é inverso, pois a carga dos dados, já transformados, é feita diretamente no *data warehouse* e, conforme necessidade vão se criando os *data marts* departamentais.

Observamos também que esta é a única diferença na abordagem dos autores, pois de resto o diálogo entre eles é visível, principalmente no ponto sobre a extração dos dados, de diversas fontes, que foram levantados nos requisitos técnicos e sua integração no *data warehouse* e *data mart*, nos fazendo apreciar a importância de uma central única da informação, onde os dados estão integrados em um único ambiente.

Vale ressaltar que as abordagens de Kimball (2002) e Inmon (2005) devem ser fortemente consideradas em

soluções de *Business Intelligence* Corporativas, que envolvem os departamentos da Empresa, para garantir uma informação consistente, confiável e segura, de forma a suportar a tomada de decisão.

Com o que já foi explanado no livro sobre metodologia ágil, percebe-se que a abordagem proposta por Kimball (2002) pode ser mais fácil de se aderir ao desenvolvimento por *releases* e módulos, onde cada módulo desenvolvido seria um *data mart*. Desta forma seriam construídos primeiro, até o desenvolvimento de todos os módulos e, conseqüentemente o carregamento do *data warehouse*.

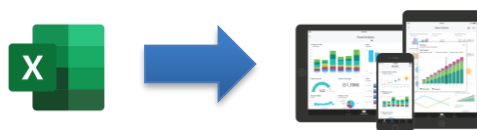
Não podemos esquecer, que nem todas as soluções de *business intelligence* são corporativas, que envolvem inúmeros departamentos. Atualmente, com a difusão massiva da importância de se analisar dados, empresas menores e individuais também terão suas soluções de *business intelligence* e possivelmente uma complexidade menor, não sendo necessário uma

abordagem tão ampla, quanto a apresentada pelos autores. Porém, os conceitos apresentados sempre serão utilizados, independentemente da complexidade de sua solução.

A partir deste entendimento, apresentamos três níveis de complexidade para uma arquitetura de dados, onde o primeiro nível é o de **baixa complexidade**.

Uma arquitetura de baixa complexidade é quando existe apenas uma fonte de dado, normalmente em planilha ou arquivo de texto, com baixa frequência de atualização dos dados, pouca ou nenhuma transformação de dados. Normalmente, encontramos este tipo de arquitetura em Empreendedores Individuais, pequenos comércios locais, autônomos etc.

**Figura 31** – Arquitetura de baixa complexidade



Fonte: Autoria Própria

Como podemos observar, uma arquitetura simples que envolve apenas a criação de uma *dashboard*, baseado em dados de uma planilha em *Excel*.

Lembre-se! Menos é mais! Não coloque complexidade onde não existe complexidade!

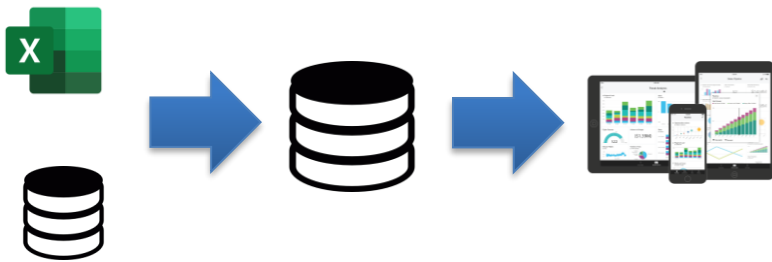
O segundo nível de arquitetura de dados é o de **média complexidade**. Neste nível já é possível ter mais de uma fonte de dado, pode existir uma tabela de banco de dados, uma atualização de dados mais frequente, apresentam transformações de dados importantes, porém ainda não se trata de uma solução corporativa, que envolva vários departamentos.

Normalmente, encontramos este nível de complexidade em micro empresas, que possuem um volume de vendas frequente, diário. Algumas médias empresas que já começam a implantar suas soluções de *business intelligence*, já percebem que a complexidade aumentará com o tempo, portanto já necessitam criar uma central única da informação. Também já há a



necessidade de compartilhamento da informação com um nível de segurança.

**Figura 32** - Arquitetura de média complexidade



Fonte: Autoria Própria

Percebe-se que já há a necessidade de um banco de dados para integrar as fontes de dados, principalmente pensando na escala de crescimento dessa solução.

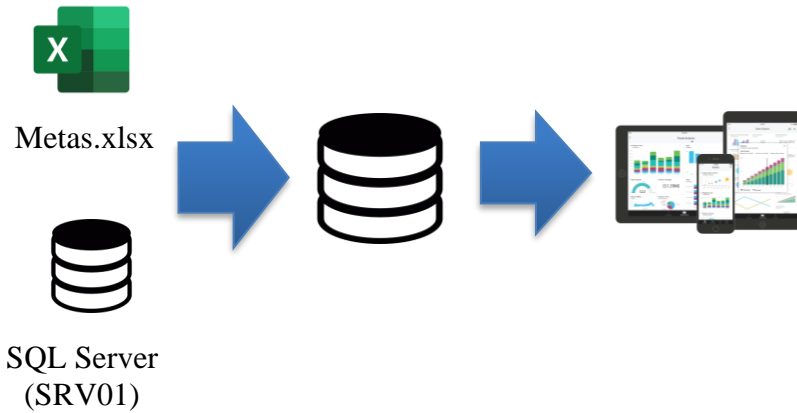
Por fim, temos o nível de arquitetura de dados de **alta complexidade**. Este nível está voltado para empresas que possuem vários departamentos, volume alto de dados, diversas fontes de dados, diversas regras de

negócio e transformações de dados. Normalmente, este nível é encontrado em grandes companhias e até em médias, que já possuem uma maturidade maior no quesito análise de dados.

Para estes cenários a recomendação é utilizar as abordagens apresentadas por Kimball (2002) e Inmon (2005), pois garantirão a confiabilidade, integridade, rastreabilidade, escalabilidade, manutenibilidade, performance e segurança para sua solução de Business Intelligence.

No nosso caso, ou seja, utilizando o levantamento de requisitos técnicos, podemos dizer que se trata de uma arquitetura de dados de média complexidade, onde o desenho da arquitetura de dados seria conforme a imagem a seguir:

**Figura 33 - Arquitetura do cenário proposto**



Fonte: Autoria Própria

## Integrações de Dados

Conforme observamos nas Figuras 3 e 4, que apresentam as abordagens de arquitetura de dados de Kimball (2002) e Inmon (2005), respectivamente, é possível identificar que em ambas há o processo de integração de dados, que você também vai reconhecer pelos nomes: ETL ou ELT.

## **ETL**

*Extract, Transform and Load.* Em português, Extrair, Transformar e Carregar. Que consiste em extrair os dados das fontes de origem (de acordo com a Matriz de Necessidades), transformá-las em uma área de estágio e carregar o resultado dentro do *Data Warehouse* ou *Data Mart*, dependendo da abordagem que utilizará.

## **ELT**

*Extract, Load and Transform.* Em português, Extrair, transformar e Carregar. Que consiste em extrair os dados das fontes de origem (de acordo com a Matriz de Necessidades), carregá-los no *Data Warehouse* ou *Data Mart*, dependendo da abordagem que utilizará, e transformá-las neste destino.

Atualmente, as principais tecnologias de integração de dados, trabalham como ELT, visto que a performance de transformação dos dados no destino é superior, se

comparada quando feita no meio do processo, conforme o ETL.

É importante destacar que, para ser considerado um processo de ETL ou ELT, faz-se necessário alta disponibilidade, automação, performance e estarorquestrando dados corporativos, ou seja, dados que permeiam toda empresa.

Reforço este ponto, para não confundirmos “Tratamento de Dados” com “Integração de Dados”. São processos diferentes.

Tratamento de dados você pode realizar no Excel, no *Power BI*, no *Google Planilhas*. Conecte-se a uma ou mais fontes de dados, carregue-os em uma dessas ferramentas e realize do tratamento LOCAL dos dados.

Quando falamos local, significa que estes estão disponíveis somente na sua estação de trabalho e para compartilhá-los você precisará enviar por e-mail, salvar

em um drive na nuvem ou ainda publicar em um serviço da web.

Veja, quando escrevo este livro, em 2020/21, as tecnologias citadas são apenas tecnologias de Tratamento de dados, pode ser que em um futuro todas elas evoluam e se tornem tecnologias de integração de dados.

Já, a Integração de dados, além de ser possível prover tratamento aos dados, também é possível disponibilizá-la corporativamente em um servidor, por exemplo, para ser executada automaticamente e atender a TODA corporação e não somente uma estação local.

Um aspecto importante do ambiente da arquitetura é a integração de dados que ocorre em todo o processo. Os dados trafegam a partir do ambiente operacional até serem carregados no ambiente do Data Warehouse. (INMON, 2005, p. 18)

Como podemos observar nas palavras de Inmon (2005), a integração de dados é a responsável por orquestrar ou,

com o perdão do trocadilho, arquitetar todo o processo responsável pela carga dos dados no *Data Warehouse* ou *Data Mart*. Reforçando o que dissemos sobre a diferença de tratamento e integração de dados.

Tanto negócio, quanto TI possuem muito interesse na cadeia integração de valor [...] vários projetos que envolvem Data Warehouse focam no gerenciamento necessário para entender o cliente, a partir de uma perspectiva de ponta a ponta. Obviamente, isso requer uma habilidade de analisar as informações através dos processos corporativos, como cotações, pedidos, faturamentos, pagamentos, atendimento ao cliente. (KIMBALL, 2002, p. 76-77)

Assim como Inmon (2005), Kimball (2002) vai destacar a importância de uma integração de dados que atravesse todos os departamentos da Empresa, formando uma “cadeia integrada de valor” (KIMBALL, 2002, p. 76), garantindo assim uma gestão completa do cliente.

Nos tempos atuais, onde o volume, a variedade e a velocidade em que os dados chegam é muito alta,

precisamos destacar a importância de uma governança de dados e a integração deles é que será o grande aliado da governança.

## **Governança de Dados**

Atualmente, o Dado é considerado um ativo da empresa, dado – com o perdão da redundância – a sua importância para a corporação, assim como os demais ativos, porém com uma vantagem, pois o dado pode ser compartilhado, trabalhado, tratado, transformado.

O Dado é o grande responsável pelo sucesso – e muitas vezes, pelo fracasso – de uma empresa, pois é nele que os tomadores de decisão encontrarão oportunidades e ameaças, sejam elas externas ou internas e, partir daí, planejarem as ações para enfrentar cada uma delas.

Observe que vários tipos de elementos (imóveis, veículos, pessoas, etc.) já são considerados ativos organizacionais e



habitam os balancetes das empresas. Algumas empresas já pensam em colocar os seus dados neste mesmo patamar. Isso Mesmo. Pelo valor intrínseco que os dados possuem e que, potencialmente, podem oferecer, cada vez mais esses elementos estão subindo nessa escala de valoração e caminham nessa direção. (BARBIERI, 2019, p. 36)

O Professor Carlos Barbieri (2019) destaca a importância do dado, principalmente por conta do seu valor intrínseco. E a palavra intrínseco, segundo o dicionário Aurélio, apresenta dois significados:

1. Que está por dentro de alguma coisa ou pessoa e lhe é próprio, íntimo
2. Inseparavelmente ligado a uma coisa ou pessoa.

Percebam a riqueza que o dado oferece a uma empresa, pois se combinarmos os dois significados, trazendo para a realidade corporativa, obteremos que o dado está Inseparavelmente ligado ao negócio de uma empresa e por conta disso está por dentro de tudo que acontece na empresa. Neste ponto, eu reforço que todas pessoas da

empresa devem estar inseridas na governança de dados, desde o/a profissional que está na ponta realizando os cadastros, inseridos dados nos sistemas transacionais até o acionistas.

Conhece aquele ditado que diz: “Contra fatos, não há argumentos”? Os dados são os insumos dos fatos e nada melhor do que tomá-los como base, para suportar decisões de negócio.

### **Processos-chave da Governança de dados**

Segundo o DMBok (*Data Management Book of Knowledge*), que é a bíblia da Governança de Dados e desenvolvido pela DAMA.org (*Data Management Association*), para se adequar as melhores práticas da Governança de Dados, faz-se necessário o cumprimento de alguns processos-chave, que no DMBok são conhecidos como áreas de conhecimento.

No Guia DAMA-DMBOK2, estamos enfatizando "áreas de conhecimento" em vez de "funções". Uma área de conhecimento é uma categoria de especialização. Pode ser composto de um ou mais tópicos, que será tratado em seções separadas. (HENDERSON *et al.*, 2014, p. 8)

O Guia do DMBOK, nos apresenta dez áreas de conhecimento:

**Figura 34 – Áreas de conhecimento da Governança de dados**



Fonte: Roda das áreas de conhecimento (HENDERSON *et al.*, 2014, p. 9)

A partir da imagem anterior, podemos identificar o quanto é importante os temas abordados aqui neste livro, pois são a sustentação de uma Governança de Dados.

O Guia do DMBOK vai definir cada área de conhecimento desta roda, porém ele começa apresentando a definição de Governança de Dados.

Planejamento, supervisão e controle sobre o gerenciamento de dados e o uso de dados e recursos. Embora entendamos que a governança abrange 'Processos', não 'coisas', o termo comum para governança de dados é gerenciamento de dados. (HENDERSON *et al.*, 2014, p. 10)

Nessa definição apresentada, já podemos perceber que a governança de dados é uma gestão de processos, obviamente com a participação de pessoas, de modo a garantir uma solução de Business Intelligence íntegra e confiável, de ponta a ponta.

Após a definição de Governança de dados, o guia do DMBoK, nos apresenta a definição de cada área de conhecimento:

### **Arquitetura de dados**

“A estrutura geral de dados e recursos relacionados a dados como uma parte integrante da arquitetura corporativa.” (HENDERSON *et al.*, 2014, p. 10)

Perceba que a definição do guia DMBoK “dialoga” com a nossa definição, bem como com a definição de Kimball (2002) e Inmon (2005), afirmando que a arquitetura necessita ser corporativa.

### **Modelagem e Projeto de Dados**

“Análise, projeto, construção, teste e manutenção”  
(HENDERSON *et al.*, 2014, p. 10)

Ainda falaremos posteriormente sobre modelagem de dados, mas perceba que o sugerido pelo DMBOK está perfeitamente alinhado com a metodologia proposta por este livro.

## **Armazenamento e operações de dados**

“Implantação de armazenamento de ativos de dados físicos estruturados” (HENDERSON *et al.*, 2014, p. 10)

Mais um ponto que já comentamos, sobre a importância de se ter uma estrutura que seja possível armazenar os dados.

## **Segurança de dados**

“Garantindo privacidade, confidencialidade e acesso apropriado” (HENDERSON *et al.*, 2014, p. 10)

A segurança é de suma importância para qualquer solução que envolva dados. Aqui neste livro trataremos sobre este assunto, inclusive apresentando um modelo para mapeamento da segurança.

### **Integração de dados e interoperabilidade**

“Aquisição, extração, transformação, movimento, entrega, replicação, federação, virtualização e suporte operacional” (HENDERSON *et al.*, 2014, p. 10)

Mais uma área de conhecimento que conversa bastante com o exposto no livro, bem como com as definições de Kimball (2002) e Inmon (2005). Reafirmando a

importância desse processo em uma solução de  
*Business Intelligence*.

## **Documentos e Conteúdo**

“Armazenamento, proteção e habilitação de acesso aos dados encontrados em fontes não estruturadas, e disponibilizando esses dados para integração” (HENDERSON *et al.*, 2014, p. 10)

Fontes não estruturadas são dados que não se encontram em bases de dados, por exemplo: Excel, Google Planilhas, Bloco de Notas etc.

Esse processo vem sustentar o que dissemos anteriormente, inclusive representamos em nossa arquitetura de dados, onde estes arquivos não estruturados devem ser integrados e armazenados em



bancos de dados, correlacionando-se com as demais fontes.

## **Referência e dados mestre**

“Gerenciamento dos dados compartilhados para reduzir a redundância e garantir melhor qualidade por meio da definição padronizada e do uso dos dados.”  
(HENDERSON *et al.*, 2014, p. 10)

Quando se fala do gerenciamento dos dados compartilhados, se trata de representar o dado que é consumido por dois ou mais departamentos. Quando você utiliza uma tecnologia para integrar dados localmente e, esses dados, também são utilizados por outro departamento, corre o risco de você e o profissional do outro departamento, proverem tratamentos diferentes ao dado e, consequentemente obterem resultados diferentes.

Mais um processo que dialoga comigo, Kimball (2002) e Inmon (2005), afirmando que dados corporativos necessitam de uma arquitetura de dados e, para que esta exista, há a necessidade de um processo de integração de dados.

### ***Data Warehousing & Business Intelligence***

“Gerenciamento analítico do processamento de dados, garantindo o acesso a dados que suportem a tomada decisão, através de relatórios e análises” (HENDERSON *et al.*, 2014, p. 10)

Este livro é sobre isso. Porém, perceba quantas áreas de conhecimento, quantos processo são necessários para se garantir uma tomada de decisão assertiva e confiável.

## **Metadados**

“Coleta, categorização, manutenção, integração, controle, gerenciamento e entrega dos metadados” (HENDERSON *et al.*, 2014, p. 10)

Primeiro é importante contextualizar “metadado”, que nada mais é do que o dado sobre o dado. Por exemplo: Loja é uma dimensão, que dentro dela estão contidas todas as lojas (Loja 1, Loja 2, Loja 3 ... Loja N). A Loja 1 é um dado da dimensão Loja. O m<sup>2</sup> da Loja 1 é um metadado da Loja 1. O m<sup>2</sup> da Loja 3 é um metadado da Loja 3.

São os atributos do dado. Como destaca na definição do DMBOK. É uma categorização.

## **Qualidade de dados**

“Definir, monitorar, manter a integridade dos dados e melhorar a qualidade dos dados” (HENDERSON *et al.*, 2014, p. 10)

Falaremos um pouco mais adiante sobre o tema, mas de momento entenda que qualidade de dados é o resultado de um trabalho que aplica uma metodologia séria, para uma solução de *Business Intelligence*. Que executa um levantamento de requisitos de alto nível e o aplica com profissionalismo tudo o que foi mapeado.

É o que este livro está apresentando a você e o fazendo refletir.

## **Time de Governança de Dados**

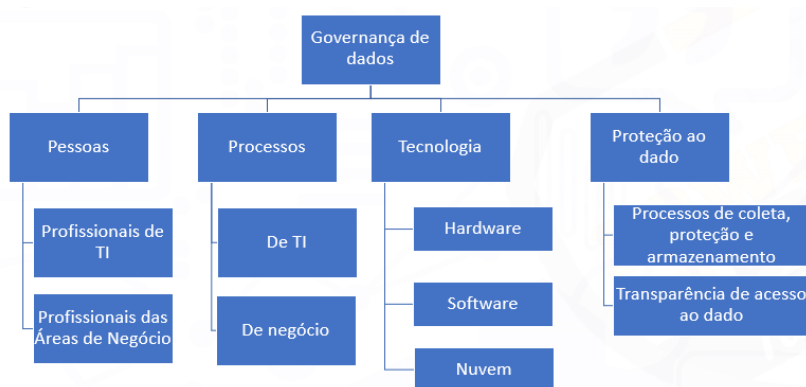
Para garantir que estes processos-chave (áreas de conhecimento) logrem êxito, faz-se necessário que haja um time de profissionais (Tecnologia da Informação e

Negócio) responsáveis por garantir que o Dado esteja íntegro, confiável e seguro para ser utilizado em suas análises e dashboards.

É importante destacar aqui 2 coisas:

1. Um time de governança de dados é necessário quando estamos falando de grandes empresas, com volumes de dados altíssimos. Se sua empresa é micro ou você é um empreendedor individual, você também será responsável pela Governança do seu Dado.
2. É importante começar, mesmo que você e sua Empresa não tenham orçamento para destacar um time exclusivo de Governança de Dados. Neste caso, envolvam profissionais já existentes na Empresa e com esse perfil, para que estes, além das atividades que exercem, também sejam envolvidos na Governança de Dados

**Figura 35** - Hierarquia do time de governança de dados



Fonte: Autoria Própria

Essa hierarquia foi definida e é extremamente defendida por mim, obviamente, fundamentando-me nos autores já apresentados até o momento, de modo que seja possível perceber a importância da integração entre as áreas e profissionais de negócio com os de Tecnologia da Informação.

Os profissionais e a área de tecnologia da informação garantirão suporte tecnológico para sustentar os processos-chave da Governança de Dados.

Os profissionais e a área de negócios serão os responsáveis por apoiar o levantamento de requisitos, para informar sobre os processos e regras de negócio, além de homologar os dados que estão sendo disponibilizados, pois são estes que realmente conhecem a qualidade do dado.

Temos também um pilar de “Proteção ao Dado”, onde é de extrema importância, principalmente com a sanção da Lei Geral de Proteção ao Dado e que será nosso próximo tema.

## **Lei Geral de Proteção de Dados (LGPD). Lei 13.709**

Esta Lei dispõe sobre o tratamento de dados pessoais, inclusive nos meios digitais, por pessoa natural ou por pessoa jurídica de direito público ou privado, com o objetivo de proteger os direitos fundamentais de liberdade e de privacidade e o livre desenvolvimento da personalidade da pessoa natural. (BRASIL, 2018, Art. 1)

Acima podemos analisar o artigo 1 da LGPD que foi sancionada em 14 de agosto de 2018 e diz que toda e qualquer empresa que coletar dados pessoais, por qualquer meio deverá seguir o que está escrito nesta lei.

Segundo a LGPD, no artigo 5 pode-se encontrar o que são considerados dados pessoais, são eles:

- **I - Dado pessoal:** informação relacionada a pessoa natural identificada ou identificável;
- **II - Dado pessoal sensível:** dado pessoal sobre origem racial ou étnica, convicção religiosa, opinião política, filiação a sindicato ou a organização de caráter religioso, filosófico ou político, dado referente à saúde ou à vida sexual, dado genético ou biométrico, quando vinculado a uma pessoa natural;
- **III - Dado anonimizado:** dado relativo ao titular que não possa ser identificado, considerando a utilização de meios técnicos razoáveis e disponíveis na ocasião de seu tratamento;



Isto posto a lei determina que para que uma empresa possa capturar o dado de alguém, deve haver consentimento desta pessoa e, entende-se por consentimento:

XII - consentimento: manifestação livre, informada e inequívoca pela qual o titular concorda com o tratamento de seus dados pessoais para uma finalidade determinada. (BRASIL, 2018, Art. 5)

Para que você entenda melhor, vamos discorrer um pouco mais sobre consentimento, onde no artigo 8º menciona que: “O consentimento deverá ser fornecido por escrito ou por outro meio que demonstre a manifestação de vontade do titular.” (BRASIL, 2018, Art. 8)

Isso quer dizer que a pessoa tem que ter total conhecimento que a Empresa está capturando seu dado, bem como saber o motivo da captura do dado, conforme o inciso 4º, do artigo 8: “O consentimento deverá referir-se a finalidades determinadas, e as autorizações

genéricas para o tratamento de dados pessoais serão nulas.” (BRASIL, 2018, Art. 8)

Repare, então, que para que a captura do dado pessoal seja realizada, deve-se haver uma relação clara e objetiva entre o consentimento da pessoa natural e a intenção de captura do dado da Empresa em questão.

E a Empresa que é a responsável por guardar as provas desta relação, caso contrário, se acionada, poderá ser multada.

Ainda em se tratando de consentimento, a qualquer momento este poderá ser revogado pela pessoa, segundo o inciso 5º, artigo 8:

O consentimento pode ser revogado a qualquer momento mediante manifestação expressa do titular, por procedimento gratuito e facilitado, ratificados os tratamentos realizados sob amparo do consentimento anteriormente manifestado enquanto não houver requerimento de eliminação, nos termos do inciso VI do caput do art. 18 desta Lei. (BRASIL, 2018, Art. 8)

Atente-se para o trecho “...por procedimento gratuito e facilitado...” (BRASIL, 2018). É um ponto importante e que haverá mais dedicação por parte das empresas, pois para atender a este inciso integralmente, será necessário a criação de alguma interface, como um portal web, por exemplo, onde qualquer pessoa poderá consultar os dados que Empresa capturou, bem como o que esta Empresa está fazendo com estes dados e ter a opção de exclusão do dado pelo titular do mesmo.

Isso tudo deverá ser de forma transparente, simples, fácil e gratuita. Quando falamos sobre este acesso, no artigo 9, pode-se observar as informações ao qual o titular do dado deverá ter acesso:

O titular tem direito ao acesso facilitado às informações sobre o tratamento de seus dados, que deverão ser disponibilizadas de forma clara, adequada e ostensiva acerca de, entre outras características previstas em regulamentação para o atendimento do princípio do livre acesso:

I - finalidade específica do tratamento;

II - forma e duração do tratamento, observados os segredos comercial e industrial;

III - identificação do controlador;

IV - informações de contato do controlador;

V - informações acerca do uso compartilhado de dados pelo controlador e a finalidade;

VI - responsabilidades dos agentes que realizarão o tratamento; e

VII - direitos do titular, com menção explícita aos direitos contidos no art. 18 desta Lei. (BRASIL, 2018, Art. 9)

Por este motivo, as Empresas terão que dispor de um bom planejamento, bem como investimento, para atender a estes requisitos dispostos na lei.

A Lei ainda trata, detalhadamente, sobre a segurança dos dados pessoais, disposto no artigo 46:

Os agentes de tratamento devem adotar medidas de segurança, técnicas e administrativas aptas a proteger os dados pessoais de acessos não autorizados e de situações acidentais ou ilícitas de

destruição, perda, alteração, comunicação ou qualquer forma de tratamento inadequado ou ilícito. (BRASIL, 2019, Art. 46)

Caso o dado pessoal seja violado ou algum problema de segurança seja identificado, este deverá ser comunicado a autoridade nacional, conforme descreve o artigo 48:

O controlador deverá comunicar à autoridade nacional e ao titular a ocorrência de incidente de segurança que possa acarretar risco ou dano relevante aos titulares. (BRASIL, 2018, Art. 48)

Uma vez que a autoridade nacional tem a ciência do fato, recomendará as possíveis ações a Empresa e, havendo prejuízo ao titular, a Empresa poderá receber uma multa.

A Lei prevê a criação de um órgão regulador de proteção ao dado, que esta chama de Autoridade Nacional, este Órgão se chamará ANPD-Autoridade Nacional de Proteção de Dados, conforme descreve o capítulo IX da LGPD.

Fica criada, sem aumento de despesa, a Autoridade Nacional de Proteção de Dados - ANPD, órgão da administração pública federal, integrante da Presidência da República. (BRASIL, 2018, Art. 55-A)

A ANPD terá várias competências e que estão dispostas no artigo 55-J:

Compete à ANPD:

I - zelar pela proteção dos dados pessoais;

II - editar normas e procedimentos sobre a proteção de dados pessoais;

III - deliberar, na esfera administrativa, sobre a interpretação desta Lei, suas competências e os casos omissos;

IV - requisitar informações, a qualquer momento, aos controladores e operadores de dados pessoais que realizem operações de tratamento de dados pessoais;

V - implementar mecanismos simplificados, inclusive por meio eletrônico, para o registro de reclamações sobre o tratamento de dados pessoais em desconformidade com esta Lei;

VI - fiscalizar e aplicar sanções na hipótese de tratamento de dados realizado em descumprimento à legislação, mediante processo administrativo que assegure o contraditório, a ampla defesa e o direito de recurso;

VII - comunicar às autoridades competentes as infrações penais das quais tiver conhecimento;

VIII - comunicar aos órgãos de controle interno o descumprimento do disposto nesta Lei praticado por órgãos e entidades da administração pública federal;

IX - difundir na sociedade o conhecimento sobre as normas e as políticas públicas de proteção de dados pessoais e sobre as medidas de segurança;

X - estimular a adoção de padrões para serviços e produtos que facilitem o exercício de controle e proteção dos titulares sobre seus dados pessoais, consideradas as especificidades das atividades e o porte dos controladores;

XI - elaborar estudos sobre as práticas nacionais e internacionais de proteção de dados pessoais e privacidade;

XII - promover ações de cooperação com autoridades de proteção de dados

peçoais de outros países, de natureza internacional ou transnacional;

XIII - realizar consultas públicas para colher sugestões sobre temas de relevante interesse público na área de atuação da ANPD;

XIV - realizar, previamente à edição de resoluções, a oitiva de entidades ou órgãos da administração pública que sejam responsáveis pela regulação de setores específicos da atividade econômica;

XV - articular-se com as autoridades reguladoras públicas para exercer suas competências em setores específicos de atividades econômicas e governamentais sujeitas à regulação; e

XVI - elaborar relatórios de gestão anuais acerca de suas atividades. (BRASIL, 2018, Art. 55-J)

A LGPD ainda cria um artigo dedicado a boas práticas e governança dos dados pessoais, mencionando como que este dado deverá ser manipulado, uma vez que foi capturado. Este artigo é o 50 e diz o seguinte:

Os controladores e operadores, no âmbito de suas competências, pelo tratamento de



dados pessoais, individualmente ou por meio de associações, poderão formular regras de boas práticas e de governança que estabeleçam as condições de organização, o regime de funcionamento, os procedimentos, incluindo reclamações e petições de titulares, as normas de segurança, os padrões técnicos, as obrigações específicas para os diversos envolvidos no tratamento, as ações educativas, os mecanismos internos de supervisão e de mitigação de riscos e outros aspectos relacionados ao tratamento de dados pessoais.(BRASIL, 2018, Art. 50)

Este ponto é interessante, pois como foi visto em “Time da Governança de dados”, tem-se um pilar de Proteção ao Dado e este terá um responsável, chamado Encarregado (ou controlador), onde o objetivo é desenvolver o seguinte programa de Governança em privacidade, conforme descrito no artigo 50:

I - implementar programa de governança em privacidade que, no mínimo:

a) demonstre o comprometimento do controlador em adotar processos e políticas internas que assegurem o cumprimento, de forma abrangente, de

normas e boas práticas relativas à proteção de dados pessoais;

b) seja aplicável a todo o conjunto de dados pessoais que estejam sob seu controle, independentemente do modo como se realizou sua coleta;

c) seja adaptado à estrutura, à escala e ao volume de suas operações, bem como à sensibilidade dos dados tratados;

d) estabeleça políticas e salvaguardas adequadas com base em processo de avaliação sistemática de impactos e riscos à privacidade;

e) tenha o objetivo de estabelecer relação de confiança com o titular, por meio de atuação transparente e que assegure mecanismos de participação do titular;

f) esteja integrado a sua estrutura geral de governança e estabeleça e aplique mecanismos de supervisão internos e externos;

g) conte com planos de resposta a incidentes e remediação; e

h) seja atualizado constantemente com base em informações obtidas a partir de monitoramento contínuo e avaliações periódicas; (BRASIL, 2018, Art. 50)

Além disso deverá garantir a integração, se necessário, com os demais dados da Empresa.

A LGPD vem com objetivo de garantir privacidade a dados pessoais e deve se ajustar, principalmente, quando estiver em vigor e se deparar com situações reais.

Integração e governança de dados é muito mais do que foi apresentado, porém achei importante trazer estes conceitos, mesmo que de forma sintetizada, para que você entenda a importância de uma integração de dados e o impacto que ela exerce em sua solução de Business Intelligence.

Para você fazer o mapeamento da sua integração de dados, utilize a Matriz de Necessidades e os Requisitos técnicos.

**Tabela 6** – Modelo para levantamento de requisitos de  
extração de dados

Pacote	Regra	Destino
<b>PKG01</b>	VLR_Venda, COD_Loja, DT_Movimentacao, COD_Cidade, COD_SKU, COD_Vendedor.	TB_MOVIMENTACAO
<b>PKG02</b>	COD_Loja, DSC_Loja, M2_Loja, ENDERECO_Loja, BAIRRO_Loja, CIDADE_Loja, UF_Loja, CEP_Loja, VLR_Latitude, VLR_Longitude	TB_LOJA
<b>PKG03</b>	DT_Movimentacao, COD_MESANO	TB_PERIODO
<b>PKG04</b>	COD_MESANO, VLR_Meta	TB_META

<b>PKG05</b>	COD_SKU, DSC_SKU, FAMILIA_SKU, DEPARTAMENTO_SKU	TB_PRODUTO
<b>PKG06</b>	COD_Vendedor, NM_Vendedor, CPF_Vendedor, ENDERECO_Vendedor, BAIRRO_Vendedor, CIDADE_Vendedor UF_Vendedor, CEP_Vendedor, DT_Admissao, DT_Desligamento	TB_VENDEDOR

Fonte: Autoria Própria

## Modelo de dados

Como já citado anteriormente, em uma arquitetura de dados, encontraremos o *Data Warehouse* este, por sua vez, será constituído por tabelas de banco de dados,

tabelas estas que estão mapeadas na matriz de necessidades e detalhadas nos pacotes de extração de dados.

Para que estas tabelas se comuniquem umas com as outras, ou seja, para que seja possível relacionar dados de lojas, com produtos e vendas, faz-se necessário um modelo de dados, no qual será responsável por prover a relação entre os dados das diversas tabelas.

Vamos trazer um exemplo prático do dia a dia: Imagina que você vai a uma loja e efetua a compra de uma calça, uma camisa e um tênis. Vamos atentar para os processos que acontecem:

Processo de compra:

- a. Você escolhe os produtos
- b. Encaminha ao caixa
- c. Caixa registra os produtos
- d. Você efetua o pagamento
- e. Recebe a NF e seus produtos

f. Finaliza a compra

Neste simples processo, quando a pessoa do caixa registra seus produtos, esses dados estão sendo inseridos em uma tabela de banco de dados, responsável por armazenar as vendas. Ficando da seguinte forma:

**Tabela 7 – Exemplo da tabela de vendas**

Data_Compra	CPF_Cliente	COD_Vendedor	Cod_Produto	Qtde	Vlr_Produto	Vlr_Desconto	Vlr_Impostos	Vlr_Total
25-01-21	000.111.222.-00	99999	123	1	100	0	20	80
25-01-21	000.111.222.-00	99999	345	1	50	0	10	40
25-01-21	000.111.222.-00	99999	678	1	80	0	16	64

Fonte: Autoria própria

Agora, imagine que este processo de compra aconteça inúmeras vezes, durante os dias, meses e anos. Ou seja, constantemente dados são inseridos nesta tabela, registrando as sucessivas vendas.

Em paralelo a este processo, existe um outro, chamado processo gerencial, responsável pela gestão da loja, onde o gestor necessita tomar algumas decisões e para

isso algumas perguntas precisam de respostas, como por exemplo:

- a) Quais os três produtos que mais venderam na semana?
- b) Quais os três produtos que menos venderam neste mês?
- c) Qual o melhor vendedor da semana?
- d) Qual o cliente que mais comprou neste mês?
- e) Qual a evolução do meu volume de vendas no ano de 2020?
- f) Qual o valor vendido nos últimos quinze dias?
- g) Quais os vendedores não bateram a meta mensal?

Enfim, citei algumas perguntas, mas certamente cada gestor necessitará de diversas respostas, para tomar decisões de forma a conduzir o negócio da melhor forma possível.

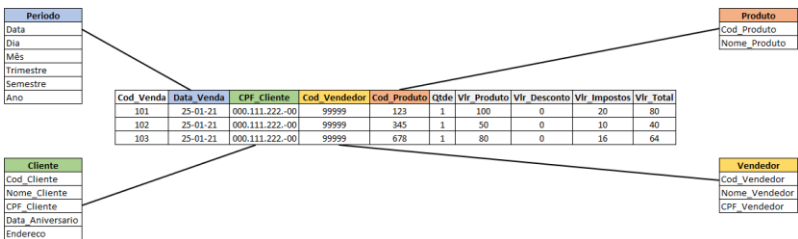
Para que estas perguntas (e outras) sejam respondidas, se faz necessário um modelo de dados capaz de



sustentá-las, de forma que seja possível prover as informações de produtos, vendedores, períodos, clientes etc.

Estas informações acerca dos dados serão as tabelas do modelo de dados que, para responder as perguntas hipotéticas listadas acima, ficariam modeladas da seguinte forma:

**Tabela 8 – Modelo de dados do exemplo**



Fonte: Autoria própria

Perceba que orbitam ao redor da tabela com o registro das vendas, as demais tabelas com os dados inerentes as respostas que serão fornecidas ao gestor da loja, em seu processo gerencial de tomada de decisão.

Desta forma, entenda que a tabela que possui as MÉTRICAS de vendas é o FATO; as tabelas que a orbitam são as DIMENSÕES.

Este modelo de dados está mapeado na sua matriz de necessidades, ela será o insumo para a desenvolvimento.

O modelo de dados clássico, utilizado em um *Data Warehouse* e *Data Marts* é o Modelo Dimensional, também chamado de *star schema* (KIMBALL, 2002) ou *star join* (INMON, 2005), traduzindo para o português, modelo estrela, pois a tabela fato está no meio do modelo e as dimensões nas pontas, lembrando uma estrela.

Modelo dimensional é uma metodologia para modelagem lógica de dados que visa desempenho de consulta e facilidade de uso começando a partir de um conjunto de eventos de aferição dos dados. No banco de dados, uma tabela de fatos é construída geralmente com um registro para cada medição. Esta tabela de fatos é rodeada por um conjunto de tabelas de dimensão que descrevem precisamente o

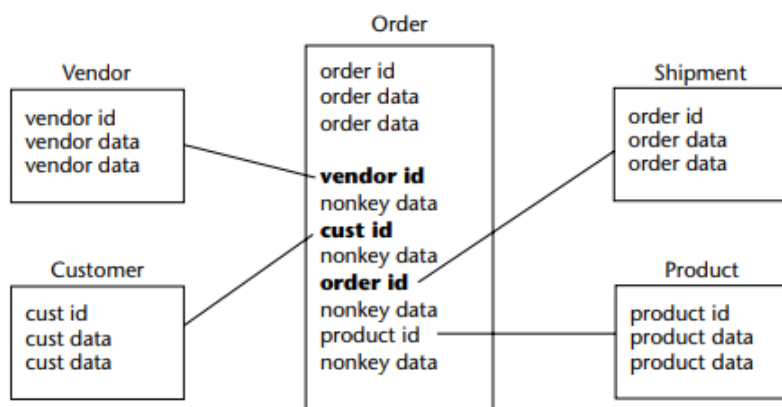
que é conhecido no contexto de cada registro de medição. Por causa da estrutura característica de um modelo dimensional, é freqüentemente chamado de esquema em estrela. (KIMBALL, 2002, p. 399)

Este tipo de modelagem tornou-se a principal busca em projetos de business intelligence, por ser de fácil compreensão, baixa manutenção, facilmente escalável e que possibilita consultas rápidas, ágeis e de fácil assimilação pelos/as usuários/as. Por este motivo que os requisitos do usuário, conforme constam na matriz de necessidades, bem como na sua documentação de requisitos é que serão o insumo para a criação do seu modelo de dados.

O benefício de criar modelos estrela é agilizar os dados para o processamento. Interligando os dados e criando redundância seletiva, o modelo simplifica e agiliza os dados para acesso e análise, que é exatamente o que é necessário para as consultas [...] A grande vantagem do modelo multidimensional é sua eficiência de acesso. Quando projetado

corretamente, o modelo estrela é muito eficiente na entrega de dados para o usuário final. Para tornar a entrega de informações eficiente, os requisitos do usuário final devem ser reunidos e assimilados. São o cerne da definição de como será a estrutura multidimensional. Uma vez que os requisitos do usuário final são compreendidos, serão usados para o desenvolvimento da estrutura ideal do modelo estrela. (INMON, 2005, p. 129-130-362)

**Figura 36 – Modelo Estrela (*Star Join*) proposto por Inmon**



Fonte: Um simples modelo estrela (INMON, 2005, p. 129)

Aqui é a hora de criar o(s) modelo(s) de dados, que será responsável por gerar o dado necessário para você construir o seu dashboard. É necessário que você tenha algum conhecimento de Business Intelligence, caso não tenha, peça a ajuda de um especialista.

É importante analisar que ambos os autores trazem abordagem semelhantes com relação ao modelo estrela, principalmente no sentido de desempenho, agilidade e facilidade de consulta pelo/a usuário/a. E isto é um fator preponderante de sucesso do seu projeto, a facilidade e agilidade com que o usuário conseguirá realizar consultas para obter respostas aos seus requisitos de negócio.

Neste ponto, eu ousou acrescentar ao tema, a metáfora do **CUBO**, nome popular ao modelo estrela que é entregue ao usuário final, que visa facilitar ainda mais o acesso pelo/a usuário/a, fazendo com que o seu modelo estrela seja também uma camada semântica, de forma

a utilizar uma nomenclatura de fácil entendimento pelo/a usuário/a, por exemplo:

- Ao invés de DSC\_Produto, utilize Descrição do Produto.
- Ao invés de NM\_Cliente, utilize Nome do Cliente.
- Ao invés de Vlr\_Total, utilize Valor Total.

E assim por diante. Os nomes dos campos devem ficar somente para utilização pelos/as analistas de dados, quando for disponibilizar para o/a usuário/a, faça com que seja de simples entendimento e alinhado com a terminologia que o/a mesmo/a utiliza.

O nome CUBO surge pelo fato de que um modelo dimensional ou, no nosso caso, multidimensional – pois apresenta duas ou mais dimensões – possibilita a análise de uma determinada informação sobre vários pontos de vista, várias perspectivas. Recomenda-se a construção de cubos em *Data Marts*, justamente pelo fato de serem entregues para os/as usuários/as. No

exemplo que citamos, a venda poderá ser analisada por produto e esta mesma venda poderá ser analisada por vendedor.

subconjunto de dados altamente inter-relacionados organizado para permitir aos usuários combinar quaisquer atributos em um cubo (p.ex., lojas, produtos, clientes, fornecedores) com quaisquer métricas no cubo (p.ex., vendas, lucro, unidades, idade). (TURBAN *et al.*, 2009, p. 241-242)

Perceba que Turban *et al.* (2009) ainda reforça a possibilidade de se analisar todas as métricas existentes na tabela fato, com todas as dimensões existentes em seu cubo. Neste ponto, é preciso ter cuidado pois esta funcionalidade pode (e vai) abrir possibilidade de se analisar uma métrica ou um indicador com alguma dimensão que não faça sentido, por exemplo:

Imagine o que na sua empresa existam vendedores com exclusividade em produtos – isso é comum quando se requer um vendedor técnico - Maria só vende parafusos

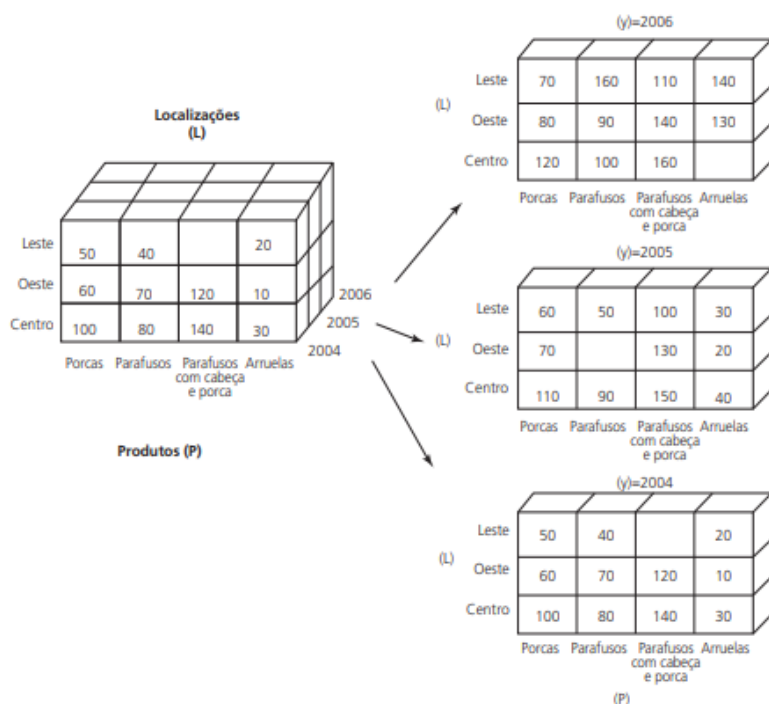
e João só vende pregos. Observa-se então, uma dimensão de Vendedor (Maria e João), uma dimensão de Produto (Parafuso e Prego) e uma fato (Valor da Venda).

Como o cubo fornece a possibilidade de se analisar qualquer dimensão com qualquer métrica, será possível realizar uma análise sobre o valor das vendas de parafusos feitas por João e o valor das vendas de pregos feitas por Maria. O que resultará em valores nulos. É importante capacitar o/a usuário/a sobre isto e orientá-lo como proceder nestes casos. Uma simples supressão ou filtro de valores em branco já resolveria.

Destaca-se aqui a importância da matriz de necessidades, pois ela vai te apresentar, na exata medida, todos as possíveis inter-relações entre dimensões e métricas, conforme ilustra a imagem a seguir:



**Figura 37 – Exemplo de possibilidades de análises em um cubo.**



Fonte: Visões e análise do cubo (TURBAN *et al.*, 2009, p. 119)

Lembre-se que o Cubo é um modelo de dados estrela e que pode ser desenvolvido em qualquer banco de dados relacional ou utilizar tecnologias próprias para isso, como

por exemplo o *Oracle Essbase* ou *SQL Server Analysis Services*.

A estrutura de um cubo de dados, como já mencionada, pode utilizar tabelas de um banco de dados relacional, bastando implementar a arquitetura e modelagem corretas. Outra opção, normalmente utilizada em arquitetura de dados corporativa é utilizar tecnologias específicas (como já citado), porém deve-se atentar a estrutura desta, visto que existem tecnologias que trabalham utilizando armazenamento em disco e outras que trabalham utilizando memória.

O cubo em uma estrutura que envolve armazenamento, por exemplo, *Oracle Essbase* e *SQL Server Analysis Services Multidimensional*, necessitarão de uma infraestrutura que suporte a escala de volume de dados, de crescimento. Estas tecnologias armazenam em disco todos os dados, referentes a todas as inter-relações entre as dimensões e métricas, independente se faça ou não sentido.

Elas trabalham desta forma, pois visam a agilidade na consulta aos dados, visto que todos eles já estão disponíveis, todos as métricas e indicadores já estão com os seus resultados calculados, ou seja, a consulta será instantânea, muito rápida.

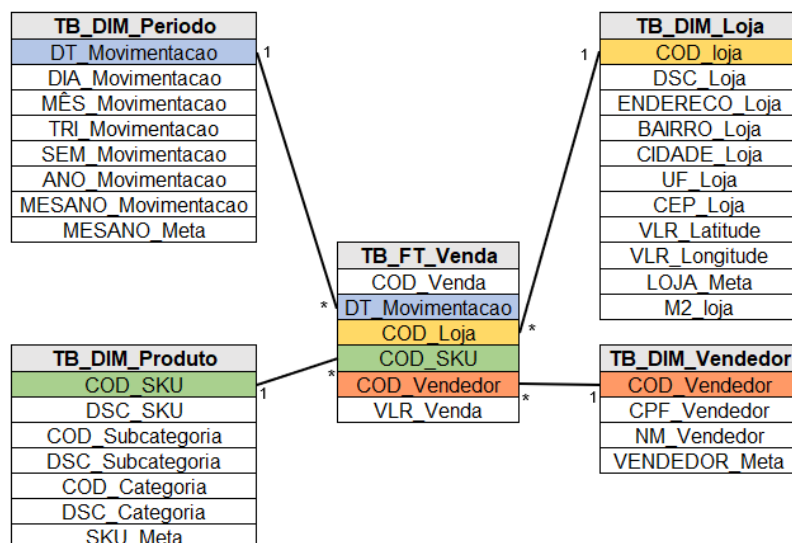
Já estruturas que utilizam tecnologias baseadas em memória (*in-memory*), não armazenam os dados, os cubos ficam conectados a sua fonte de origem e quando o/a usuário/a realiza uma consulta, o cubo se utilizará de memória e processamento para buscar a informação, calculá-la (se necessário) e apresentar na tela. Todas as operações no cubo são realizadas com o suporte da memória. Neste caso, faz-se necessário que a sua estrutura garanta escalabilidade de memória que suporte o crescimento do volume de informações e transações. Atualmente, as grandes empresas optam por esta estrutura, utilizam serviços em nuvem, como o *Azure Analysis Services*.

Os cubos de dados oferecem a oportunidade de obter informações para suporte à decisão de maneira eficiente. A análise de cubo permite às pessoas realizar consultas através da busca de uma série de visualizações de relatórios, usando os recursos [...] como page-by, pivot, classificação, filtro e drill-up/drill-down, que permitem aos usuários fazer “visões parciais” de um cubo de dados ou analisar um cubo usando um simples clique do mouse. (TURBAN *et al.*, 2009, p. 118)

Como já mencionado anteriormente e reforçado neste momento, utilize a matriz de necessidades e o documento de requisitos como seus principais insumos para o desenho do modelo de dados.

O modelo de dados, do *Data Warehouse*, baseado na matriz de necessidades que foi desenvolvida neste livro, ficará conforma imagem a seguir:

**Figura 38 – Modelo de dados estrela**



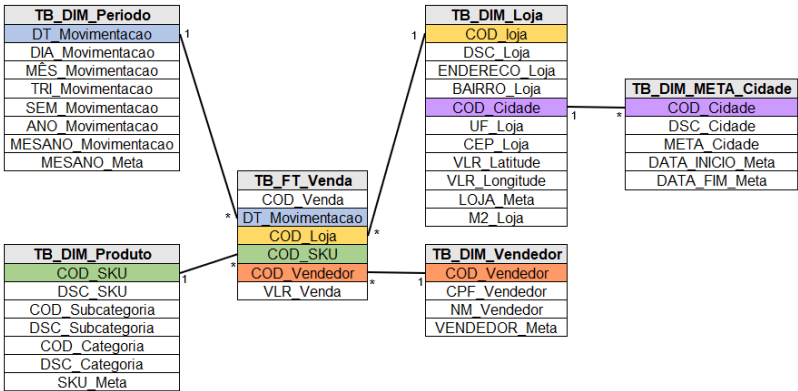
Fonte: Autoria própria

Temos alguns pontos a se observar com relação a este modelo de dados. O **primeiro ponto** a observar é que, mesmo nossa matriz de necessidades apresentando uma dimensão de região, o modelo não a trouxe. Isso é proposital, visto que região já se encontra na

TB\_DIM\_Loja, respeita a granularidade – CIDADE\_Loja - e não há um requisito acerca da análise de metas por Cidade (Vide matriz de necessidades).

Se houvesse um requisito que necessitasse da análise de metas por cidade, este modelo de dados sofreria uma variação, ficando da seguinte forma:

**Figura 39 – Modelo de dados com meta por cidade**



Fonte: Autoria própria

Este modelo é uma variação do estrela, chamado *snow flake*, em português floco de neve, que consiste em criar um relacionamento entre uma dimensão e uma tabela de atributo desta mesma dimensão. Neste caso específico, a meta da cidade, que é diferente da meta da loja, passa a ser um atributo da dimensão loja.

Isso já deixa exposto a importância da Engenharia de Requisitos em sua solução e o desenvolvimento da matriz de necessidades, pois um pequeno detalhe pode alterar a forma como implantará o seu modelo de dados.

O **segundo ponto** a destacar está relacionado as medidas calculadas e indicadores que constam na matriz de necessidades: Total de Vendas/m<sup>2</sup>,  $(\text{Total de Venda/Meta}) \cdot 100$ , Vendas/m<sup>2</sup> (ano atual) – Vendas/m<sup>2</sup> (ano anterior). Como é possível observar no modelo, nós não os descrevemos na tabela fato, pois eu tenho a preferência em calcular as métricas somente na criação dos *Data Marts*, ou seja, no cubo que será

disponibilizado para o/a usuário/a, de forma a respeitar o requisito de cada análise.

Se tivéssemos criado a medida  $(\text{Total de Venda/Meta}) \times 100$  dentro da tabela fato, teríamos muitas informações equivocadas, com análises que não fariam sentido, pois na tabela fato, encontra-se a inter-relação entre todas as dimensões e as metas são por dimensão, consequentemente este cálculo estaria errado.

Porém, é possível fazer isto de forma correta, neste caso específico na tabela fato deveria ter uma coluna com o cálculo, considerando a meta por dimensão, da seguinte forma:

- Coluna: Total de Venda/MetaLoja
  - Regra:  $(\text{Total de Venda/MetaLoja}) \times 100$
- Coluna: Total de Venda/MetaPeriodo
  - Regra:  $(\text{Total de Venda/MetaPeriodo}) \times 100$
- Coluna: Total de Venda/MetaProduto
  - Regra:  $(\text{Total de Venda/MetaProduto}) \times 100$



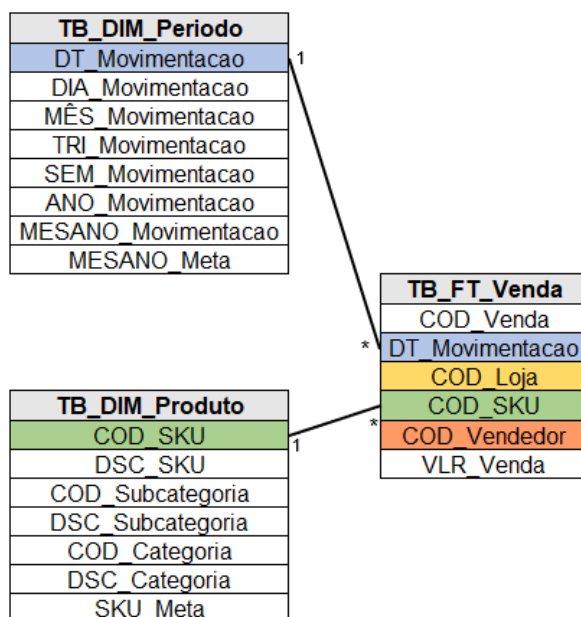
- Coluna: Total de Venda/MetaVendedor
  - Regra:  $(\text{Total de Venda/MetaVend}) * 100$

Eu, particularmente, não gosto e nem recomendo trabalhar desta forma, visto que em soluções mais complexas, é possível que haja vários cenários como esse, logo sua tabela fato ficará muito grande, com muitas regras e, com isso dificultando a manutenção futura, tanto da integração, quanto do modelo de dados, pois imagine se esse cálculo mudar, você terá que ajustá-lo em todos os locais que teve que replicar.

Portanto, a minha recomendação é criar as regras no cubo que será disponibilizado para acesso e consulta pelos/as usuários/as.

Partindo para o **terceiro ponto**, falaremos sobre relacionamentos cardinais. Repare que em cada relação da dimensão com a tabela fato, existe uma linha conectando uma coluna da tabela de dimensão com uma coluna da tabela fato.

**Figura 40** – Ilustração do Relacionamento entre tabelas



Fonte: Autoria própria

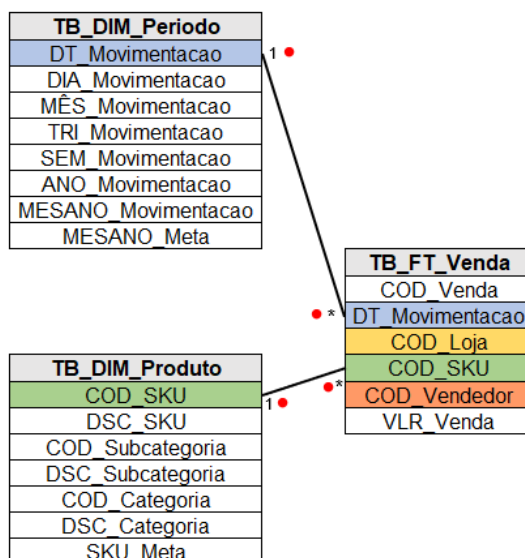
Perceba que existe uma inter-relação física entre a tabela fato e as tabelas de dimensão, através – do que chamamos – de chave de relacionamento. Na figura 17 as chaves de relacionamento estão conectadas e

pintadas da mesma cor. Este relacionamento possibilita obter informações complementares, existentes somente na dimensão.

Por exemplo, para saber o total de vendas da categoria, que um SKU pertence, utilizaremos a métrica VLR\_Venda (TB\_FT\_Vendas), e através da chave de relacionamento (COD\_SKU), será possível obter a categoria, através da coluna DSC\_Categoria (TB\_Dim\_Produto). Se o relacionamento não existisse, não seria possível analisar o total de vendas da categoria.

Ainda em referência aos relacionamentos, é possível observar que existe o número “1”, de um lado do relacionamento e o “\*” do outro lado, conforme ilustra a imagem a seguir:

**Figura 41 – Relacionamento cardinal**



Fonte: Autoria própria

O nome que se dá a este procedimento é cardinalidade, criando assim um relacionamento cardinal, que pode ser de três tipos:

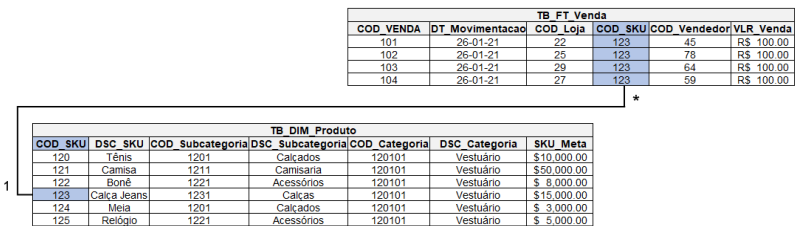
- 1:\* - Leia “um para muitos”
- 1:1 – Leia “um para um”

- \*.\* - Leia “muitos para muitos”

O numeral “1” significa que na tabela não existe repetição da chave de relacionamento. Na tabela TB\_DIM\_Produto, o COD\_SKU é único, nunca se repetirá.

O símbolo “\*”, que significa “muitos”, informa que na tabela a chave do relacionamento se repete frequentemente. Na tabela TB\_FT\_Venda, o COD\_SKU se repetirá.

**Figura 42 – Relacionamento cardinal 1:\* (um para muitos)**



Fonte: Autoria própria

## Camada de apresentação

Em tempos de *Self-service Business Intelligence*, em geral, é a etapa desenvolvida pelo/a usuário/a, por conta das facilidades que as tecnologias de *report* de business intelligence oferecem, como: *Power BI*, *Qlik Sense*, *Tableau*, *Microstrategy*, *Excel* entre outras. Porém é importante entender o real conceito do *Self-Service Business Intelligence*.

O conceito de BI é mesmo. A diferença é que existem 2 atores: Profissional da área de dados e o/a Profissional da área de negócio. Vamos entender os papéis de cada um. O primeiro tem a responsabilidade de garantir que o Dado esteja dentro de uma governança. Ou seja, é o profissional responsável pela “saúde” do dado. Já o/a profissional de negócio será o/a ator/a responsável por desenvolver os dashboards, utilizando os dados disponibilizados pelo/a profissional da área de dados e com a liberdade de criar análises, relatórios e dashboards de acordo com seus requisitos de negócio e necessidades. (BONEL, 2019, p. 34)

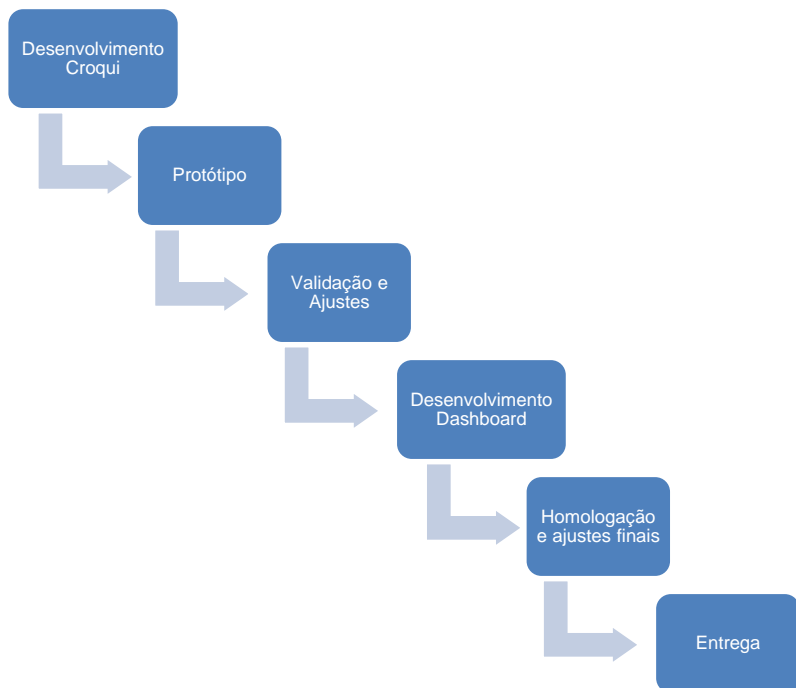
Conforme observado por Bonel (2019), o/a profissional de negócio – entenda o/a usuário/a é quem desenvolverá as análises, relatórios e *dashboards*, afinal é quem está vivenciando os desafios do dia a dia do negócio, porém nem sempre ele/ela sabe como utilizar a tecnologia para criação destes painéis, nem como desenvolver visualizações de dados, dentro das melhores práticas. Caberá a você esta orientação.

Em outros casos, o/a usuário/a ele não terá a possibilidade de desenvolver suas análises ou preferirá demandar esta atividade para você, caso isto aconteça, você terá de realizar um levantamento de requisitos para desenvolvimento de *dashboards*, para encontrar a melhor forma de representar estes dados.

Neste caso, uma boa prática é que o resultado do levantamento de requisitos seja um croqui do dashboard, para que você e o/a usuário/a estejam alinhados acerca do como será a visualização do dado.

Para este levantamento de requisitos existe um processo que você pode seguir para te auxiliar.

**Figura 43** – Fluxo de processos para levantamento de requisitos da camada de apresentação

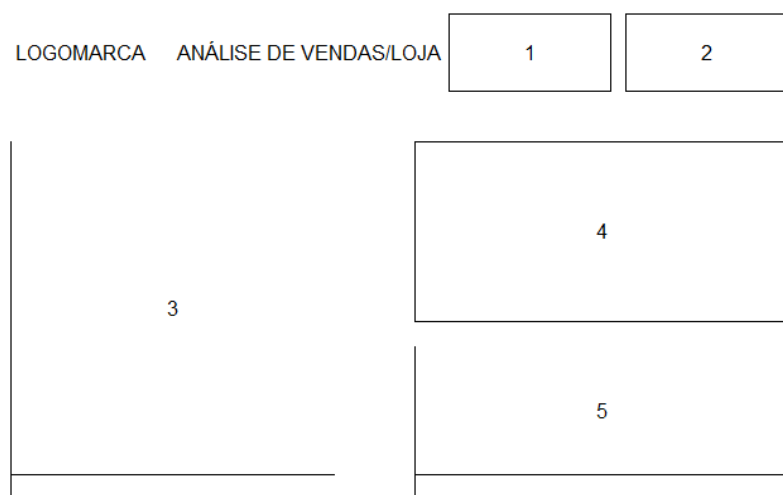


Fonte: Autoria própria



A primeira etapa do fluxo é o **desenvolvimento do croqui**, que será realizado em entrevista com o/a usuário/a, visando chegar a uma primeira versão. Existem tecnologias específicas para isso, mas sugiro utilizar o Excel, Word, Lousa, de forma que fique visível.

**Figura 44** – Croqui para desenvolvimento de dashboard.



Fonte: Autoria Própria

### Legenda:

1. Total Vendido
2. % Atingimento de vendas (Loja)
3. Ranking de Lojas por total vendido
4. Mapa por latitude e longitude das lojas, representadas segundo total de vendas
5. Total vendas no tempo, representado pela hierarquia de datas (Ano -> Mês)

A legenda é importante para que registre quais os objetos visuais e seus dados, deverão ser representados no *dashboard*.

O objetivo do croqui é fazer com que o/a usuário/a possa ter visibilidade de como será o seu *dashboard*, facilitando assim um percentual de assertividade maior, no desenvolvimento do mesmo.

Vencido o primeiro processo do fluxo, com base nas informações coletadas e desenho do croqui, chega o momento de desenvolver o **protótipo**, que se trata de

uma primeira versão do que foi levantado nos requisitos, utilizando a tecnologia de *report de BI*, para ser apresentado ao/a usuário/a, com o objetivo de que ele/ela homologue o que foi especificado no croqui, que é justamente o próximo processo do fluxo, chamado de **validação e ajustes**. Nesta fase o/a usuário/a valida o protótipo e identifica possíveis alterações que deverão ser realizadas por você na próxima etapa do processo, o **desenvolvimento do dashboard**, onde você, em posse das alterações solicitadas, promoverá os ajustes necessários para atendê-las.

Uma vez finalizado, retorna ao/a usuário/a para que ele/ela realize a **homologação final**, baseando-se nos cenários de teste e plano de segurança (veremos mais adiante). Pode ser que sejam solicitados novos ajustes, você os realiza e retorna para homologação. Esta etapa é muito importante e é de responsabilidade do/a usuário/a, pois ele/ela precisará dar o seu aceite ao que

está desenvolvido. A você cabe estar proativo, em contato com o/a usuário/a, para apoiá-lo.

Nem sempre é rápido, visto que o/a usuário/a também possui suas atividades do dia a dia, por isso sua proatividade e comunicação é importante para que esta etapa seja vencida.

Uma vez que o/a usuário/a formaliza que está de acordo com o *dashboard*, o mesmo estará **entregue**, ou seja, está finalizado e poderá ser utilizado no dia a dia do/a usuário/a.

## Casos de Teste

Os casos de teste têm por objetivo garantir que o software foi codificado corretamente, se as integrações funcionam e estão de acordo com o mapeado; e apoiar o/a usuário/a na execução da homologação dos resultados a partir dos requisitos levantados. Um passo

a passo para que, a partir dos requisitos, ele promova os testes em cada um deles, valide o resultado final e identifique se atendem (ou não) os desafios mapeados.

A estratégia de teste de software fornece um roteiro que descreve os passos a serem executados como parte do teste, define quando esses passos são planejados e então executados, e quanto trabalho, tempo e recursos serão necessários. Portanto, qualquer estratégia de teste deve incorporar planejamento dos testes, projeto de casos de teste, execução dos testes, coleta e avaliação dos dados resultantes. (PRESSMAN, 2011, p. 401)

Pressman (2011) ressalta a importância de se promover uma estratégia de teste rígida e que possa garantir resultados qualitativos e confiáveis, porém precisa ser ágil e flexível.

Planeje-se com antecedência, em tempo de levantamento de requisitos. Não espere o fim do desenvolvimento para mapear os casos de teste. Dada a importância desta etapa, muitas empresas possuem

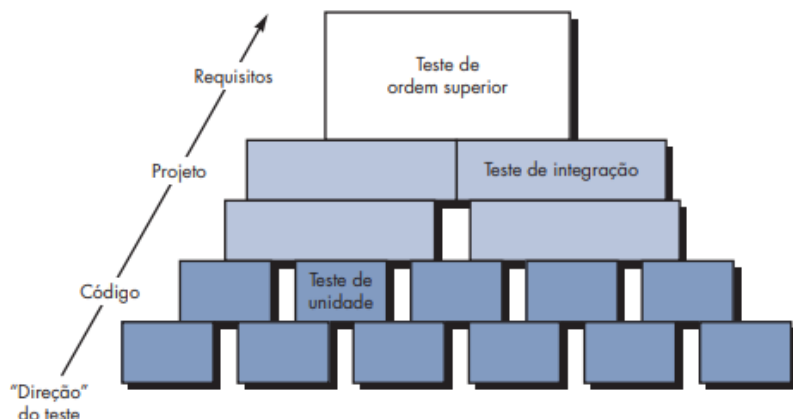
um profissional especializado em testes de software, visto que somente o êxito nela poderá garantir que o software entregou os resultados que suportarão os desafios mapeados e, conseqüentemente, tornar-se-á um software de qualidade.

O processo de teste tem dois objetivos distintos: 1. Demonstrar ao desenvolvedor e ao cliente que o software atende a seus requisitos. Para softwares customizados, isso significa que deve haver pelo menos um teste para cada requisito do documento de requisitos [...] 2. Descobrir situações em que o software se comporta de maneira incorreta, indesejável ou de forma diferente das especificações. Essas são conseqüências de defeitos de software. O teste de defeitos preocupa-se com a eliminação de comportamentos indesejáveis do sistema, tais como panes, interações indesejáveis com outros sistemas, processamentos incorretos e corrupção de dados. (Sommerville, 2011, p. 144)

Sommerville (2011) apresentará uma definição importante, no sentido dos objetivos do teste de software e, que se completa com o que diz Pressman (2011), no

sentido dos resultados esperados a partir dos testes de software. Perceba que ele vai destacar que o teste é importante tanto para o cliente, quanto para desenvolvedor, ambos necessitam ter a clareza de que o software atende aos requisitos mapeados. Este ponto destacado por Sommerville (2011) também conversa com Pressman (2011), a ver pelas etapas de teste de software propostas por este último teórico.

**Figura 45 – Processos do teste de software**



Fonte: Etapas do teste de software (PRESSMAN, 2011, p. 405)

Pressman (2011) apresenta as etapas do teste de software e você deve atentar para cada uma delas, pois seu software somente logrará o êxito esperado, se você vencer estas 3 etapas.

A primeira etapa a ser testada é referente a codificação que foi utilizada no desenvolvimento de *software*, normalmente é chamada de testes de unidade (PRESSMAN, 2011) ou testes unitários (SOMMERVILLE, 2011). É uma etapa que dependerá muito de você, do seu time e da sua proatividade.

Nesta etapa você deverá validar se tudo o que você desenvolveu está funcionando corretamente, ou seja, não apresenta erros e se está coerente com os requisitos mapeados. Certifique-se que tenha testado todos os *releases*, módulos, interfaces, cálculos e qualquer outro objeto desenvolvido, obviamente de acordo com o momento em que o desenvolvimento se encontra.

Uma vez que a testagem unitária ou de unidade foi vencida, deve-se promover os testes integrados, que



tem por objetivo testar a arquitetura implementada e a consistência da mesma, os bancos de dados, os modelos de dados, as integrações de dados e os *dashboards*. Consiste em testar todo o processo, de ponta a ponta, desde a coleta do dado, passando pelas transformações, aplicações de regras, até a exibição no *dashboard* e a segurança da informação (veremos adiante), garantindo a confiabilidade do processo.

Com relação a consistência da arquitetura, deve-se testar a capacidade da mesma sob estresse, executando diversos processos em paralelo, visando garantir que a mesma está pronta para o dia a dia da corporação, quando vários/as usuários/as estiverem demandando processos ao mesmo tempo.

Na última etapa, chega-se o momento dos testes de ordem superior, popularmente conhecido como a homologação dos requisitos pelo/a usuário/a e, conforme destacou Sommerville (2011), deve-se propor um teste para cada requisito.

Seu papel será o de apoiar o/a usuário/a durante os testes, comunicar-se constantemente, visando facilitar esse trabalho e estar à disposição para sanar dúvidas, registrar os ajustes e consertá-los, se for necessário.

Com relação aos ajustes solicitados pelo/a usuário/a estes, por sua vez, devem ser classificados em:

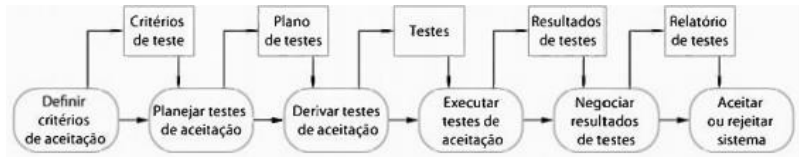
- Erro: O/A usuário/a encontrou um erro ao executar um teste, que impede de seguir testando o software.
- Inconsistência: O/A usuário/a identificou que um requisito implantado está divergente do que foi mapeado em levantamento de requisitos. A inconsistência deve ser evitada, pois no teste de unidade ou unitário, você deve garantir que o que foi implantado está de acordo com o requisito mapeado.
- Alteração: O/A usuário/a testou o requisito, ele está de acordo com o levantamento de requisito, porém ele/ela deseja que seja alterado, ou seja,

um ajuste do requisito inicial. Caso isso aconteça, você deverá especificar este requisito na sua documentação, informar o impacto de custo e prazo e negociá-lo. Poderá ser feito no projeto ou entrará como um novo requisito, que é a próxima classificação.

- Inclusão: O/A usuário/a identificou que há a necessidade de se incluir um novo requisito, que não fora mapeado no levantamento de requisitos. Normalmente, a inclusão de requisito é registrada como uma melhoria e implantada em uma fase posterior. Caso o requisito seja primordial, deve-se especificar na documentação, estimar o impacto, custo, prazo e negociá-lo.

Esta etapa de testes dos requisitos pelo/a usuário/a, deve ser bem planejada e alinhada com todos/as os/as envolvidos/as, de forma a obter o resultado positivo no final.

**Figura 46** – Fluxo do processo de testes pelo/a usuário/a



Fonte: O processo de testes e aceitação (SOMMERVILLE, 2011, p. 160)

Este fluxo, proposto por Sommerville (2011), reflete o que falamos anteriormente.

Na etapa chamada **planos de teste**, vou te sugerir um modelo para que você possa desenvolver junto com o/a usuário/a, conforme detalhamos na imagem que se segue:

**Tabela 9** – Modelo para mapeamento dos cenários de teste

Cod. Cen. Teste	Cod Req Func	Cenário a ser testado	Resultado	Parecer do/a usuário/a
<b>CT01</b>	RF01	Abrir o dashboard de lojas e analisar as vendas/m². Validar com a planilha lojas_vendasm2.xlsx	OK	De acordo com o requisito
<b>CT02</b>	RF02	Abrir o dashboard de lojas e analisar o ranking das vendas/m². Validar com a planilha lojas_vendasm2.xlsx	OK	De acordo com o requisito
<b>CT03</b>	RF03	Abrir o dashboard de Produtos e analisar o	OK	De acordo com o requisito

## Metodologia e Engenharia de Requisitos para Projetos de Business Intelligence

		atingimento dos produtos de acordo com o semáforo. Validar com a planilha gestao_vendasm2.xlsx		
<b>CT04</b>	RF04	Abrir o dashboard de Produtos e analisar o atingimento dos produtos de acordo com o semáforo. Validar com a planilha gestao_vendasm2.xlsx	NÃO OK	O Dashboard de produtos não está abrindo
<b>CT05</b>	RF05	Abrir o dashboard de vendedores e analisar o atingimento das vendas por vendedor. Validar	NÃO OK	O resultado do atingiment o das vendas

		com a planilha gestao_vendedores .xlsx		não está de acordo com o levantame nto de requisito
--	--	--	--	--

Fonte: Autoria Própria

Atente-se para os resultados “NÃO OK” e para o parecer dado pelo/a usuário/a, pois é a partir destes que você classificará o resultado, para seguir com as ações inerentes a cada classificação.

## **Segurança da Informação**

A definição da segurança da informação tem por objetivo garantir a confidencialidade a informação, o acesso adequado a mesma e precisa estar dentro do seu planejamento dos casos de teste. Um *software* de

*business intelligence* contém informações sensíveis e confidenciais a determinadas pessoas e departamentos.

Isto posto, você junto com o/a usuário/a realizará o mapeamento de acesso aos dados, de forma que seja implantado na sua solução e garanta a segurança das informações.

**Tabela 10 – Modelo para mapeamento dos níveis de acesso**

Grupo	Usuários	Acesso	Tipo de Acesso
Financeiro	João e Maria	CC001, CC002	Leitura
Marketing	José e Joca	CC003, CC004, CC007	Leitura
Vendas	Manoel, Manuela, Jorge	CC008, CC010, CC011,	Leitura



		CC020, CC022	
<b>Executivos</b>	Diretor1, Diretor2, Presidente	Todos os CC	Leitura e Escrita

Fonte: Autoria Própria

Este mapeamento deve ser feito em tempo de levantamento de requisitos, porém é muito possível que sofra ajustes no decorrer do desenvolvimento, portanto no momento de implantar a segurança, certifique-se sobre a assertividade deste plano.

Uma vez implantando os níveis de acesso as informações, há a necessidade de testá-los e para isso devem estar mapeados em seus casos de teste. Desta forma, “o teste de segurança tenta verificar se os mecanismos de proteção incorporados ao sistema vão de fato protegê-lo contra acesso indevido.” (PRESSMAN, 2011, p. 419)

A segurança da informação, além de garantir o nível de acesso correspondente a cada pessoa, também promoverá a confiança do/a usuário/a em utilizá-la, ele/ela se sentirá seguro/a na relação com o software, criando uma relação de confiança entre usuário/a e a solução de *business intelligence*.

Falhas de segurança contínuas e que não são solucionadas, podem (e vão) inviabilizar a utilização do software. Por mais que sua solução de *business intelligence* seja confiável, ou seja, atende todos os requisitos e responde a todos os desafios, uma insegurança de acesso as informações pelos/as usuários/as deixará sua solução inviável para uso, e para que ela retorne a utilização, você terá de prover as devidas soluções para sanar os problemas de segurança de acesso.

A confiabilidade e a segurança de um sistema estão relacionadas, mas um sistema confiável pode ser inseguro e vice-versa. O software pode, ainda, comportar-se de tal forma que o resultado

do comportamento do sistema cause um acidente. [...] A chave para garantir a segurança é assegurar que os acidentes não ocorram e/ou que as conseqüências de um acidente sejam mínimas. (SOMMERVILLE, 2011, p. 209-210)

O que Sommerville (2011) destaca é que acidentes por falha na segurança podem acontecer, porém que sejam mínimos e preferencialmente rastreáveis, para que possam ser corrigidos.

Os testes de segurança, conforme apresentado por Pressman (2011) são a chave para garantir que os acidentes não aconteçam. Mas, se por algum motivo acontecer, você tem um registro de todos os níveis de acesso, que ajudará na identificação da falha, ou seja, uma rastreabilidade e, conseqüentemente resolução do problema.

## **Desenvolvimento e Validação**

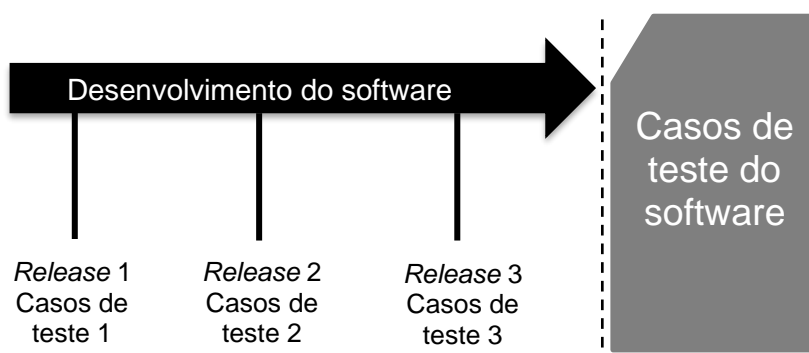
Uma vez finalizado todo o processo de Engenharia de Requisitos e aprovação do que será desenvolvido na sua solução de *business intelligence*, chega-se ao momento de iniciar o desenvolvimento dos *releases*, observando e obedecendo o levantamento de requisitos.

Como percebido, os resultados dos testes podem impactar bastante o seu trabalho, tanto em custo, quanto em prazo, por este motivo, eu sempre sugiro realizar os casos de teste em alguns marcos durante o desenvolvimento do *release*, desta forma você estará mais próximo do dia a dia do/a usuário/a, bem como este último estará mais envolvido no projeto.

Não espere o final do desenvolvimento para testar tudo. Conforme for desenvolvendo módulos, vá executando os casos de teste.

No final de todos os módulos, quando será necessário realizar o teste final, o seu software já estará bem alinhado com os requisitos do/a usuário/a, reduzindo as possibilidades de divergências e aumentando as possibilidades de sucesso.

**Figura 47** – Casos de teste em marcos do desenvolvimento



Fonte: Autoria própria

Como percebido, existem três camadas de testes: unitários, integrados e de ordem superior (Requisitos), que juntas, chamamos de validação. Este último é de responsabilidade do/a usuário/a, pois é ele quem

garantirá que o software desenvolvido está aderente aos requisitos e, conseqüentemente respondendo a todos os desafios.

O mapeamento dos cenários de teste que você realizou no levantamento de requisitos será o roteiro para que o/a usuário/a promova a validação e sua responsabilidade será a de apoiá-lo, registrar os testes em desacordo (Não OK) e negociá-los.

Normalmente esta etapa é chamada de “Go/NoGo”, ou seja é o ponto onde será decidido se a solução está apta (Go) para entrar em operação e ser disponibilizada para todos os seus usuários ou não (NoGo).

O NoGo só acontecerá se algum teste estiver em desacordo e não foi possível negociá-lo. Por este motivo é importante a negociação dos requisitos em desacordo, para que seu projeto não seja impedido de entrar em operação.

Seja flexível, resiliente e esteja aberto ao diálogo, pois pode ser que surja um novo requisito, exatamente neste momento, que é importante e que precisa ser implantado, para entrega do *release*. Por isso, que você está como consultor deste projeto, para negociar e buscar as melhores alternativas que sejam boas para ambos, tanto com relação a custo, tanto com relação a prazo. E lembre-se que você tem um documento de requisitos.

Quando tudo estiver aderente, seu projeto é um *Go*, desta forma se recomenda uma formalização entre as partes, deixando claro que a solução está de acordo com todos os requisitos.

Atualmente existem várias formas jurídicas de se oficializar uma entrega, opte por aquela que dará a segurança necessária para ambos e que seja ágil. Evite burocratizar os processos de oficialização. Isso pode causar um desconforto em todos os envolvidos e isto refletir no projeto.

## **Entrega**

A última etapa do seu projeto é a entrega, que como o próprio nome sugere, uma vez vencida todas as etapas anteriores, é a disponibilização do software para consumo pelo/a usuário/a.

## **Operação**

Se você chegou nesta etapa é porque sua solução foi validada e aprovada pelo/a seu usuário/a (Go), nos testes de Requisitos e está pronta para entrar em operação e ser compartilhada com todos/as que terão acesso.

Esta é uma etapa importante pois, por mais que o software tenha sido exaustivamente testado, é o momento em que ficará à disposição para ser acessado a qualquer momento, no trabalho, em casa, na rua, 24



horas por dia e 7 dias por semana. Garantir que o/a usuário/a esteja capacitado a esta utilização, de forma simples, amigável e ágil. E você será o facilitador, para que essa experiência seja a melhor possível.

Os processos operacionais não devem ser muito restritivos, não devem exigir que as operações sejam realizadas em determinada ordem e o software de sistema não deve depender de um processo específico a ser seguido. Os operadores geralmente melhoram o processo porque sabem o que funciona ou não em uma situação real. (SOMMERVILLE, 2011, p. 197)

Essa visão de Sommerville (2011) é importante, pois além de corroborar com a nossa proposta de metodologia, ele ressalta a capacidade que a pessoa que operará o sistema tem em se reconhecer no processo e, com isso, identificar possíveis melhorias.

Normalmente, uma vez que o software é entregue, existe um trabalho chamado operação assistida, que é

temporário, normalmente nos primeiros dias, dependendo da complexidade, pode ser alguns meses.

Este trabalho tem por responsabilidade, garantir a melhor experiência possível do/a usuário/a, registrar possíveis problemas, que não foram identificados nos testes, registrar novos requisitos e, conseqüentemente negociá-los para que sejam ajustados, portanto devemos ser parceiros e – mais uma vez reforçando – proativos, no sentido de manter o canal de comunicação aberto, no entendimento de todos os pontos identificados.

## **Manutenção da Solução**

Todos os pontos que foram registrados e negociados na operação assistida, serão desenvolvidos na etapa de manutenção da solução, que é uma etapa constante,

que está sempre sendo visitada, visto que um software sempre precisará de alguma manutenção.

Perceba que quando falamos “manutenção”, não quer dizer que o software sempre estará com problemas, com defeitos. Não se trata disso. Esta manutenção, além de corrigir possíveis erros, também está relacionada a atualização do software, para que ele sempre possa atender a dinâmica do mercado.

A manutenção começa quase imediatamente. O software é liberado para os usuários finais, e em alguns dias, os relatos de bugs começam a chegar à organização de engenharia de software. Em algumas semanas, uma classe de usuários indica que o software deve ser mudado para se adaptar às necessidades especiais de seus ambientes. E em alguns meses, outro grupo corporativo, ainda não interessado no software quando foi lançado, agora reconhece que pode lhes trazer alguns benefícios. Eles precisarão de algumas melhorias para fazer o software funcionar em seu mundo. (PRESSMAN, 2011, p. 663)

É possível observar que Pressman (2011) vai nos apresentar algumas possibilidades de manutenção, umas relacionadas a erros, porém outras relacionadas a melhorias. Essa exposição, pela minha experiência profissional, retrata a realidade, pois quanto mais um *software* é utilizado, quanto mais pessoas o utilizam, mais manutenções serão demandadas.

E posso te expor, também por experiência própria, que a manutenção do *software* é um desafio a parte dentro do seu projeto. Afinal, você entregará um *release* e assim que começar a ser utilizado, existirão manutenções, porém, você já está em processo de desenvolvimento de outro *release* (ou outros *releases*). Como fazer tudo ao mesmo tempo, de forma ágil e gerando resultados positivos?

Não há uma receita de bolo, uma resposta definitiva mas, aqui eu te sugiro que conforme seus *releases*, forem sendo disponibilizados, a manutenção deverá ser realizada por um outro time, chamando de sustentação,

para que o time de desenvolvimento continue à disposição dos futuros releases, módulos e do software.

Desta forma, o time de desenvolvimento, assim que entregar o *release*, faz uma passagem de conhecimento, ou algumas vezes, um profissional do time de manutenção participa da fase final, para que este possa atuar assertivamente. Mais uma vez, ressalta-se a importância da documentação que você criou na Engenharia de Requisitos, de forma atualizada, pois esta será a base para apoiar o time de manutenção/suporte.

Inicialmente, o time de manutenção não terá a necessidade de trabalhar *full*, então aloque horas suficientes para este trabalho e vá ajustando, conforme os *releases* e módulos forem disponibilizados.

A recomendação que posso te dar é que você seja transparente com o seu cliente, exponha as vantagens e desvantagens de se ter um time para manutenção e outro para o desenvolvimento, visto que isso aumentará

o custo do seu projeto, porém aumentará a agilidade e, principalmente, a qualidade do *software*.

## **Conclusão**

Após navegarmos nas águas de um projeto de *business intelligence* e “conversarmos” com alguns teóricos renomados no assunto, podemos observar que uma solução de *business intelligence* sustentável e confiável vai além da construção de gráficos em uma tecnologia específica.

Isso fica claro no exposto, que a tecnologia é apenas um facilitador, apenas um meio ou melhor dizendo, é um instrumento, onde possamos implantar as melhores práticas de metodologia e engenharia de requisitos.

Um projeto ágil, capaz de se adaptar as mudanças e ao dinamismo que o mercado impõe, com a segurança e flexibilidade necessárias para gerar informações que suportem às decisões relacionada aos desafios, utilizando a Engenharia de Requisitos como sua aliada, que mesmo com um nome que possa assustar em um

primeiro contato, ficou claro que são processos simples e objetivos que podem ser executados por profissionais com qualquer tipo de formação, bastando apenas adentrar-se nos processos e modelos que foram disponibilizados.

Dividir todo o projeto em partes – *releases* e módulos - é uma estratégia que, na minha experiência, tem gerado excelentes resultados, principalmente, no sentido de gerir a expectativa do/a usuário/a, pois ele estará muito mais presente no desenvolvimento do projeto, mas também no sentido de estar o mais alinhado possível com os requisitos essenciais do projeto. Tendo em mente que o documento de requisitos gerado no levantamento é essencial, pois será nossa bússula no decorrer do projeto, apontando o caminho que devemos seguir e sempre que houver ajustes nessa rota, devemos registrar e negociar, para seguirmos caminhando.

Encerro a minha conclusão, na expectativa de que essas páginas possam ter contribuído com sua caminhada, na



carreira de análise de dados, iluminando sua consciência com o entendimento de que uma solução de *business intelligence* é algo muito sério e precisa ser desenvolvida de forma que a confiança, segurança e alinhamento com os desafios corporativos estejam presentes constantemente.

## Referências

BARBIERI, Carlos. **Governança de Dados – Práticas, conceitos e novos caminhos**. Rio de Janeiro: Alta Books, 2020.

BARKER, Alan. ***How to solve almost any problem***. Edimburgo/RU: Pearson Education Limited, 2012.

BONEL, Claudio. **Power BI – Do zero ao dashboard**. 1. ed. São Paulo: Perse, 2019

BRASIL, **Decreto Lei Nº 13.709, de 14 de agosto de 2018, Lei Geral de Proteção de Dados Pessoais (LGPD)**. Disponível em:  
[http://www.planalto.gov.br/ccivil\\_03/\\_ato2015-2018/2018/lei/L13709.htm](http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm). Acesso em: 13 de julho 2020.

FERREIRA, Aurélio. **Dicionário Aurélio**: 8. ed. Curitiba: Positivo, 2019.

HENDERSON, Deborah *et al.* **DAMA-DMBOK2 Framework**. DAMA International, 2014.

INMON, Willian H. **Building the Data Warehouse**. 4. ed. Indiana/EUA: Wiley Publishing, Inc., 2005.

KIMBALL, Ralph. **The data warehouse toolkit : the complete guide to dimensional modeling**. 2. ed. Nova Iorque/EUA: John Wiley and Sons, Inc., 2002.

OLIVEIRA, Elisamara; DI NUNO, Claudinei. **Engenharia de Software**. Estácio de Sá, [entre 2010 e 2016].

PRESSMAN, Roger S. **Engenharia de software: uma abordagem profissional**. 7. ed. Porto Alegre: AMGH, 2011.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

TURBAN, Efraim *et al.* **Business Intelligence: um  
enfoque gerencial para a inteligência do negócio.**

Porto Alegre: Bookman, 2009.

## **Sobre Claudio Bonel**

Carioca da gema, daqueles que amam praia, mate gelado, biscoito globo; e torcedor do mais querido, América/RJ.

Nasci na Ilha do Governador, Zona Norte do Rio de Janeiro, especialista em Sistemas da Informação, atuando na área de tecnologia da informação desde 1996 e na área de dados desde 2004. Minha carreira foi e é dedicada a buscar soluções de análise de dados sustentáveis e duradouras, focando-me nas melhores práticas da Engenharia de *Software*, aliada a melhores práticas de Business Intelligence.

Atualmente, sou Mestrando em Educação, pela Universidade Católica de Petrópolis/RJ, Professor de nível superior e pós-graduação, em disciplinas relacionadas a análise de dados e, sócio-diretor da

Dadoteca, uma Empresa que presta serviços de consultoria e treinamentos em análises de dados.

Em paralelo, criei um projeto social chamado “Dado Humanizado: Humanizando pessoas, através dos dados”, que consiste em prover transformação social, para a população mais carente, utilizando treinamentos de análises de dados.

A educação é o único caminho para uma sociedade verdadeiramente democrática.

## **Contatos**

E-mail: [claudiobonel@gmail.com](mailto:claudiobonel@gmail.com)

Linked in: <http://br.linkedin.com/in/bonel>

Intagram: @bonelclaudio

**A**nalisar dados, gerar e acompanhar indicadores para suportar tomadas de decisão é uma premissa básica para que você possa conduzir seu negócio, seja qual for. Micro, Pequeno, Médio ou Grande. Você é gerente de um departamento em uma grande empresa? Você é gestor público? Você é executivo/a de uma multinacional? Você é um/a contador/a e atende clientes do seu bairro? Você é dono de uma farmácia? Você é presidente de uma Organização Não Governamental (ONG)? Você é gerente de um supermercado? Você é Professor/a e precisa fazer a gestão do seu conteúdo e de seus/suas alunos/as? Você é síndico/a? Você coleta dados públicos para gerar informações para o seu bairro? Você precisa fazer a gestão financeira dos seus gastos pessoais?

Para todos esses casos e milhares de outros que envolvam alguma tomada de decisão, por mais simples que for, você precisará analisar dados para decisões mais assertivas. A análise de dados é um conceito que, através de uma metodologia pode ser implantado por qualquer pessoa, independentemente de sua formação profissional, basta seguir as boas práticas apresentadas que servirão, tanto para pequenas soluções, quanto para soluções enormes. Esta metodologia tem por objetivo apoiar de forma simples e organizada todos os requisitos necessários para qualquer solução de Business Intelligence, independente da complexidade e da ferramenta a ser utilizada, baseando-se em princípios da Engenharia de Software Ágil, Engenharia de Requisitos e de Design Thinking, de fácil acesso a todos, ou seja, de simples entendimento também para profissionais que não são da área de Tecnologia da Informação.

