CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CURSO DE ENGENHARIA DE COMPUTAÇÃO

# FERRAMENTA PARA PROCESSAMENTO E INTEGRAÇÃO DE DADOS GOVERNAMENTAIS ABERTOS

**PEDRO MAGALHÃES BERNARDO**

Orientador: Ismael Santana Silva
Centro Federal de Educação Tecnológica de Minas Gerais

Coorientadores: Glívia Angélica Rodrigues Barbosa e Flávio Roberto dos Santos Coutinho
Centro Federal de Educação Tecnológica de Minas Gerais

BELO HORIZONTE
OUTUBRO DE 2016

**PEDRO MAGALHÃES BERNARDO**

# FERRAMENTA PARA PROCESSAMENTO E INTEGRAÇÃO DE DADOS GOVERNAMENTAIS ABERTOS

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Ismael Santana Silva
Centro Federal de Educação Tecnológica de Minas Gerais

Coorientadores: Glívia Angélica Rodrigues Barbosa e Flávio Roberto dos Santos Coutinho
Centro Federal de Educação Tecnológica de Minas Gerais

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CURSO DE ENGENHARIA DE COMPUTAÇÃO
BELO HORIZONTE
OUTUBRO DE 2016

**Centro Federal de Educação Tecnológica de Minas Gerais**

Curso de Engenharia de Computação

Avaliação do Trabalho de Conclusão de Curso

Aluno: Pedro Magalhães Bernardo

Título do trabalho: Ferramenta para processamento e integração de dados governamentais abertos

Data da defesa: 10/11/2016

Horário: 10:30

Local da defesa: Sala 101 - Prédio 17 (DECOM) - CEFET-MG Campus II

O presente Trabalho de Conclusão de Curso foi avaliado pela seguinte banca:

Professor Ismael Santana Silva - Orientador
Departamento de Computação
Centro Federal de Educação Tecnológica de Minas Gerais

Professora Glívia Angélica Rodrigues Barbosa - Corientadora
Departamento de Computação
Centro Federal de Educação Tecnológica de Minas Gerais

Professor Flávio Roberto dos Santos Coutinho - Corientador
Departamento de Computação
Centro Federal de Educação Tecnológica de Minas Gerais

Professor Evandrino Gomes Barros - Membro da banca de avaliação
Departamento de Computação
Centro Federal de Educação Tecnológica de Minas Gerais

Professor João Fernando Machry Sarubbi - Membro da banca de avaliação
Departamento de Computação
Centro Federal de Educação Tecnológica de Minas Gerais

Dedico este trabalho aos meus pais, minhas irmãs e à Francesca. Vocês tornaram esse momento possível.

# Agradecimentos

Quero agradecer à minha família por sempre estarem ao meu lado e me apoiarem. Mãe, você nunca mediu esforços e sem seu apoio incondicional não teria chegado onde cheguei. Pai, por me ensinar o valor do conhecimento e da educação. Talita e Luana pela companhia ao longo de tantos anos.

Aos meus amigos pessoais, do CEFET e da Universidade de Manchester, por tantos momentos inesquecíveis que fizeram tudo valer a pena.

À Francesca, por sempre estar ao meu lado, mesmo estando tão longe, e me apoiar em tudo que eu faço.

Aos meus orientadores, Ismael, Glívia e Flávio, pela paciência, incentivo e apoio que tornaram este trabalho possível.

E a todos professores do curso de Engenharia de Computação do CEFET-MG, que tanto fizeram pelo meu desenvolvimento acadêmico e profissional.

*“Não há nada mais certo que nossos próprios erros. Vale mais fazer e arrepender, que não fazer e arrepender.”* (Nicolau Maquiavel)

# Resumo

A crescente demanda por transparência levou os governos a disponibilizarem, na Internet, dados que são de interesse da sociedade, são os chamados dados governamentais abertos. No entanto, para as pessoas interessadas, o acesso a essas bases não é suficiente para fazer uso das mesmas, a falta de conhecimento técnico pode ser um empecilho. Isso ocorre pois esses dados são heterogêneos, disponíveis em diversos formatos, em grande volume e nem sempre de fácil entendimento para as pessoas interessadas. Essas características dificultam a integração desses dados, o que limita a capacidade de manipulação, combinação e análise dos mesmos. Um dos desafios gerados por esse contexto é referente a demanda por uma infraestrutura capaz de processar e integrar esses dados, viabilizando a exploração e análise dessas bases. Motivado por este cenário, este trabalho propõe o WikiOlapBase, uma ferramenta colaborativa para a integração de dados abertos, que viabiliza a análise, cruzamento e visualização desse tipo de dado. Para alcançar esse objetivo foi realizada uma revisão de abordagens para processamento, armazenamento e integração de dados. Esse levantamento identificou plataformas semelhantes, além de diferentes técnicas e tecnologias que viabilizam a criação desse tipo de ferramenta. A partir dessa revisão foram definidos os requisitos e a arquitetura do WikiOlapBase. Com essas decisões tomadas a ferramenta foi implementada. Posteriormente foi feita a avaliação de usabilidade da plataforma, para avaliar sua adequação ao uso sob a perspectiva dos usuários. Os resultados mostram a aceitação da ferramenta por parte dos usuários, bem como sua adequação ao uso.

**Palavras-chave**: Integração de dados. Dados abertos. Big Data. Software colaborativo.

# Lista de Figuras


Figura 1 – Possível modelo para banco de dados de famílias de colunas . . . . . 5
Figura 2 – Metodologia utilizada no trabalho . . . . . 12
Figura 3 – Interação entre componentes do MVC . . . . . 14
Figura 4 – Arquitetura do WikiOlapBase . . . . . 16
Figura 5 – Interface de instruções do WOB - Passo 1 . . . . . 17
Figura 6 – Interface de instruções do WOB - Passo 2 . . . . . 17
Figura 7 – Interface de instruções do WOB - Passo 3 . . . . . 18
Figura 8 – Interface de instruções do WOB - Passo 4 . . . . . 18
Figura 9 – Interface para envio de arquivo do WOB . . . . . 19
Figura 10 – Interface para preenchimento de informações básicas do WOB . . . . . 19
Figura 11 – Interface para preenchimento de tags do WOB . . . . . 20
Figura 12 – Interface para indentificação de hierarquias do WOB . . . . . 21
Figura 13 – Percentual de conclusão das tarefas pelos usuários . . . . . 25
Figura 14 – Grau de adequação do WOB por princípio de usabilidade e colaboração na visão dos usuários - Cenário C1 . . . . . 28
Figura 15 – Grau de adequação do WOB por princípio de usabilidade e colaboração na visão dos usuários - Cenário C2 . . . . . 28
Figura 16 – Grau de adequação do WOB por princípio de usabilidade e colaboração na visão dos usuários - Cenário C3 . . . . . 29

# Lista de Tabelas

Tabela 1 – Relação de tempo decorrido em minutos para cada uma das tarefas em cada teste de usabilidade . . . . . 26

# Lista de Quadros

Quadro 1 – Comparação entre os sistemas encontrados na literatura . . . . . 10
Quadro 2 – Requisitos do WOB . . . . . 13
Quadro 3 – Comparação entre os sistemas encontrados na literatura e o WikiOlapBase 22

# Lista de Abreviaturas e Siglas

| | |
|---|---|
| TIC | Tecnologias de Comunicação e Informação |
| DGA | Dados Governamentais Abertos |
| WOB | WikiOlapBase |
| SGBD | Sistema de Gerenciamento de Banco de Dados |
| NoSQL | *Not Only SQL* |
| API | *Application Programming Interface* |
| XML | *Extensible Markup Language* |
| JSON | *JavaScript Object Notation* |
| BSON | *Binary JSON* |
| RDF | *Resource Description Framework* |
| W3C | *World Wide Web Consortium* |
| OLAP | *Online Analytical Processing* |
| HDFS | *Hadoop File System* |
| SOAP | *Simple Object Access Protocol* |
| REST | *Representational State Transfer* |
| HTTP | *Hypertext Transfer Protocol* |
| SPARQL | *Sparql Protocol and RDF Query Language* |
| URI | *Uniform Resource Identifier* |
| MVC | *Model-View-Controller* |
| SoC | *Separation of Concerns* |
| DRY | *Don't Repeat Yourself* |
| HTML | *HyperText Markup Language* |
| CSS | *Cascading Style Sheets* |

# Sumário

**1 – Introdução** . . . . . **1**

**2 – Fundamentação Teórica** . . . . . **3**
2.1 Big Data e NoSQL . . . . . 3
2.2 Hadoop e Spark . . . . . 5
2.3 *Web Services* . . . . . 6

**3 – Trabalhos Relacionados** . . . . . **8**

**4 – Metodologia** . . . . . **12**

**5 – Desenvolvimento** . . . . . **13**
5.1 Levantamento de Requisitos . . . . . 13
5.2 Arquitetura do WikiOlapBase . . . . . 14
5.3 WikiOlapBase . . . . . 16

**6 – Avaliação do WikiOlapBase** . . . . . **23**
6.1 Metodologia de Avaliação . . . . . 23
6.2 Discussão dos Resultados . . . . . 25

**7 – Conclusões e Trabalhos Futuros** . . . . . **31**
7.1 Trabalhos Futuros . . . . . 31

**Referências** . . . . . **32**

**Apêndices** **35**

**APÊNDICE A – Artefatos de Avaliação** . . . . . **36**

**APÊNDICE B – Lista de Melhorias** . . . . . **46**

# 1 Introdução

Com a crescente demanda popular por mais transparência das ações governamentais, novas políticas de publicidade dessas ações vêm sendo implementadas. Segundo Vaz et al. (2010) as tecnologias de comunicação e informação (TICs) pemitiram potencializar essa transparência, um processo que se deu em três iniciativas, conforme descrito a seguir.

Inicialmente, os governos passaram a publicar informações de forma limitada em seus *websites*, ou seja, decidiam o que e como seria visualizado. Em um segundo momento, visando viabilizar a interação entre os usuários e bases de dados governamentais, a segunda iniciativa de transparência consistiu em permitir a realização de consultas para cruzamento e filtros dos dados, o que favoreceu o processo de análise das informações. Essas iniciativas eram limitadas, pois não permitiam a obtenção dos dados sem tratamentos, em seu formato original. Surgiu assim o conceito de dados governamentais abertos (DGA), nos quais, além de disponibilizar consultas e relatórios, o governo disponibiliza seus dados em estado bruto (i.e., sem pré-processamento), o que permite sua livre manipulação, processamento e análise (VAZ et al., 2010).

Em meio a esse contexto, foi criada no Brasil a Lei de Acesso à Informação (Lei nº 12.527/2011), que permite a qualquer cidadão a obtenção de dados e informações de qualquer entidade pública. Além disso, essa lei prevê a chamada “Transparência Ativa”, que determina que os órgãos públicos se antecipem aos pedidos e publiquem seus dados na Internet. Com isso, foi criado o Portal Brasileiro de Dados Abertos, no qual o governo federal disponibiliza dados, em estado bruto, que são de interesse público.

No entanto, para a maioria das pessoas interessadas, a disponibilidade de acesso a essas bases de dados não é suficiente para fazer uso das mesmas, a falta de conhecimento técnico, na maioria dos casos, se torna um empecilho (GRAVES; HENDLER, 2013). Isso ocorre porque, na maioria das vezes, os dados são heterogêneos, disponíveis em diversos formatos, em grande volume e nem sempre de fácil entendimento para as pessoas interessadas. Essas características dificultam a integração entre esses dados, o que limita a capacidade de manipulação, combinação e análise dos mesmos (HOXHA; BRAHAJ, 2011). Ou seja, a forma como atualmente esses dados estão disponibilizados, não permite a obtenção de informações relevantes sem o uso de ferramentas computacionais que auxiliem no processamento, na visualização e análise desses dados (VAZ et al., 2010).

Esse contexto gera dois desafios: o primeiro é referente a demanda por uma infraestrutura capaz de processar e integrar os DGA, viabilizando a exploração e análise dessas bases. O segundo é referente a demanda por uma ferramenta, alimentada por essa infraestrutura, capaz de gerar análises e visualizações sem a necessidade de conhecimento

técnico do usuário (GRAVES; HENDLER, 2013).

Motivado por esses desafios, este trabalho visa propor o WikiOlapBase, uma ferramenta colaborativa que seja capaz de processar e integrar dados abertos. Isso significa gerar uma nova base de dados integrada, que seja mantida pelos usuários interessados no processamento, na visualização e análise desses dados. O objetivo dessa ferramenta é prover uma infraestrutura base para outras, de modo a viabilizar a análise e visualização de grandes volumes de dados, mesmo por pessoas sem conhecimento técnico na área de Computação.

Para alcançar o objetivo proposto, este trabalho foi dividido em duas fases, subdivididas em etapas. A primeira fase consistiu na revisão da literatura acerca de abordagens para processamento, armazenamento e integração de dados no cenário dos DGA, bem como na definição dos requisitos da ferramenta. Na segunda fase a arquitetura da ferramenta foi proposta, seguida do seu desenvolvimento e avaliação.

Em termos de resultados, o Teste de Usabilidade conduzido demonstrou que o WikiOlapBase é uma ferramenta colaborativa, adequada ao uso, que auxilia no processo de integração de dados abertos. Além disso, esse trabalho apresenta um levantamento de ferramentas similares. Isso pode auxiliar tanto na escolha de qual ferramenta adotar, quanto na definição da arquitetura para pré-pocessamento e integração colaborativa de dados.

A principal contribuição prática deste trabalho é a criação de uma ferramenta colaborativa para a integração de dados abertos, cujo código fonte está disponível ao público. Além disso, outra contribuição, em termos práticos, é a revisão e utilização de soluções emergentes para processamento e integração de grandes volumes de dados, um dos grandes desafios técnicos enfrentados pela área de *Big Data* (JAGADISH et al., 2014). Em termos científicos, este trabalho contribuiu no avanço de tecnologias e abordagens para processamento e integração de dados de forma colaborativa, delimitando as vantagens e desvantagens de cada um.

Este trabalho se encontra dividido da seguinte forma: o Capítulo 2 apresenta conceitos e tecnologias fundamentais para o entendimento do trabalho; o Capítulo 3 explicita outras ferramentas existentes que viabilizam a integração, visualização e análise de dados abertos; o Capítulo 4 indica a metodologia utilizada no desenvolvimento deste trabalho; o Capítulo 5 apresenta o desenvolvimento da ferramenta proposta, desde o levantamento de requisitos, passando pela arquitetura até a apresentação da ferramenta em si; o Capítulo 6 apresenta a metodologia e resultados gerados a partir da avaliação de usabilidade do WikiOlapBase, por fim, o Capítulo 7 apresenta as conclusões geradas e trabalhos futuros propostos.

# 2 Fundamentação Teórica

Neste capítulo são discutidos conceitos fundamentais para o melhor entendimento deste trabalho. A primeira seção define o conceito de *Big Data*, e discute como esse cenário impulsionou o desenvolvimento de soluções para armazenamento, e gerenciamento de grandes volumes de dados. Também serão analisados novos modelos de dados e sistemas de gerenciamento de banco de dados (SGBDs), que surgiram como alternativa para os modelos convencionais, no contexto de *Big Data*. A seção 2.2 fala sobre ferramentas auxiliares, que podem ser utilizadas para processar grandes volumes de dados, especificamente serão apresentadas as plataformas Hadoop e Spark. Por fim, a seção 2.3, trata sobre *web services*, soluções para comunicação entre aplicações, que viabilizam, por exemplo, o acesso a dados que uma ferramenta disponibiliza.

## 2.1 Big Data e NoSQL

A evolução tecnológica viabilizou um grande aumento na velocidade e quantidade de dados que são gerados diariamente. Esses dados são produzidos em transações *online*, redes sociais, dispositivos móveis, sensores, registros governamentais, entre outros. Esse fenômeno ficou conhecido como *Big Data* (SAGIROGLU; SINANC, 2013).

O termo *Big Data* é caracterizado por três componentes: variedade, volume e velocidade. O primeiro componente se refere a variedade de fontes e tipos dos dados, em geral eles aparecem em três tipos: estruturados, não estruturados e semiestruturados. O segundo componente de *Big Data*, o volume, se refere a grande escala na quantidade de dados que são adquiridos, geralmente passando da marca dos *terabytes*. O último elemento, velocidade, se refere ao fato da geração dos dados estar acontecendo permanentemente. Devido a esses três fatores surge uma dificuldade inerente no armazenamento, gerenciamento e análise de dados no contexto de *Big Data* (SAGIROGLU; SINANC, 2013).

Por exemplo, no âmbito de armazenamento, um desafio consiste na modelagem para otimizar o processamento ao qual esses dados serão submetidos. Isso porque, desde os anos 80, o modelo de dados relacional tem dominado o mercado em diversas implementações de SGBDs. No entanto, o uso de banco de dados relacional gera diversos problemas, no contexto de *Big Data*, devido a questões como escalabilidade e limitações no armazenamento, não sendo, portanto adequado para esse cenário (MONIRUZZAMAN; HOSSAIN, 2013). Assim, novas alternativas que atendessem a novos requisitos de escalabilidade e disponibilidade se fez necessário, de modo a viabilizar que empresas e governos possam fazer uso do potencial de *Big Data* (DIANA; GEROSA, 2010).

Como solução para esses requisitos surgiram os bancos de dados NoSQL (*Not Only SQL*). Em geral esses bancos compartilham as seguintes características: não relacional, distribuído, escalável, sem esquema ou com esquemas flexíveis, suporte a replicação nativo e acesso através de interfaces de programação de aplicativos (APIs) (DIANA; GEROSA, 2010). Existem diversos modelos de banco de dados NoSQL, em seguida serão discutidos os tipos mais comuns, são eles: chave-valor, orientado a documentos, grafos e família de colunas (DIANA; GEROSA, 2010).

Bancos de dados do tipo chave-valor armazenam elementos com uma chave identificadora, e seu respectivo valor, em tabelas conhecidas como *hash tables*. Esses valores podem ser textos comuns ou estruturas, como listas e conjuntos. São ideais para respostas a requisições rápidas, uma vez que a busca é realizada apenas pelos valores das chaves. Os SGBDs não relacionais abertos, do tipo chave-valor, mais conhecidos são o Voldemort, do LinkedIn, e o Redis (MONIRUZZAMAN; HOSSAIN, 2013).

Já os bancos de dados orientados a documentos armazenam uma coleção de atributos e seus valores, estes últimos podendo ser multivalorados. Em geral não possuem estrutura fixa, ou seja, diferentes documentos podem ter diferentes estruturas, o que os torna uma escolha apropriada para armazenamento de dados semiestruturados (DIANA; GEROSA, 2010). Geralmente são codificados em formatos padrão, como *Extensible Markup Language* (XML), *JavaScript Object Notation* (JSON) ou *Binary JSON* (BSON). Podem ser utilizados, por exemplo, para armazenamento e gerenciamento de representações não normalizadas de entidades, além disso, as buscas nesse modelo podem ser feitas tanto por atributos quanto por valores. Os SGBDs desse tipo mais utilizados são o MongoDB e o CouchDB (MONIRUZZAMAN; HOSSAIN, 2013).

Na modelagem baseada em grafos os dados são representados como grafos dirigidos. Além disso, as operações sobre os dados fazem uso dos conceitos referentes à grafos, como: caminhos, vizinhos e sub-grafos (DIANA; GEROSA, 2010). Em geral esse tipo de modelagem é útil quando se existe interesse tanto no relacionamento, quanto no dado em si. Os SGBDs mais utilizados são o Neo4j, InfoGrid e AllegroGraph (MONIRUZZAMAN; HOSSAIN, 2013). Vale ressaltar também o SGBD Jena, que implementa uma API para criação e manipulação de grafos no formato *Resource Description Framework* (RDF), modelo proposto pela *World Wide Web Consortium* (W3C) para troca de dados na internet (MCBRIDE, 2001).

Finalmente, para os casos em que se deseja otimizar a leitura, podem ser usados os bancos de dados de famílias de colunas. Os bancos de dados relacionais convencionais armazenam os dados em linhas, ou seja, todas as informações referentes a uma entidade são armazenadas juntas, já no caso do armazenamento colunar, um registro passa a ser armazenado em colunas separadas. A Figura 1 mostra uma possível modelagem para esse caso. Esse tipo de armazenamento possui algumas vantagens, como, por exemplo, a

compressão dos dados e a velocidade das operações de leitura. Essa última característica torna os bancos de dados de famílias de colunas ideais para processamento analítico *online* (OLAP), quando se deseja uma leitura rápida. São exemplos de SGBDs nessa categoria o Cassandra e o HBase (DIANA; GEROSA, 2010).

Figura 1 – Possível modelo para banco de dados de famílias de colunas

| Coluna Nome | |
|---|---|
| Nome | Key |
| Lucas | 100,101 |
| Gabriela | 102 |
| Maria | 103 |

| Coluna Idade | |
|---|---|
| Idade | Key |
| 15 | 100 |
| 20 | 101 |
| 30 | 103 |

| Tabela | | |
|---|---|---|
| Key | Pessoa | Idade |
| 100 | Lucas | 15 |
| 101 | Lucas | 20 |
| 102 | Gabriela | |
| 103 | Maria | 30 |

Fonte: O Autor

Cada modelo de dados possui suas vantagens e desvantagens, e a melhor escolha depende do que se deseja alcançar. Apesar das modelagens e SGBDs não relacionais possuirem claras vantagens, elas ainda possuem certas limitações. Por exemplo, a maioria das implementações NoSQL não suportam as operações *join* e *order by* (POKORNY, 2013). Nesse contexto surgem outras plataformas, para auxiliar no processamento de grandes volumes de dados, e que possuem uma comunicação natural com bancos de dados NoSQL. Na próxima seção são apresentadas duas dessas plataformas, o Hadoop e o Spark.

## 2.2 Hadoop e Spark

O Apache Hadoop[1], também conhecido apenas como Hadoop, é um projeto de código aberto mantido pela Apache Software Foundation, que desenvolve softwares para processamento distribuído de grandes volumes de dados. O ecossistema do Hadoop é composto por quatro módulos:(1) *Hadoop Common*, (2) *Hadoop Distributed File System (HDFS)*, (3) *Hadoop YARN* e (4) *Hadoop Map Reduce* (KUMAR et al., 2014).

O primeiro, *Hadoop Common*, é um conjunto de utilitários que suporta os outros módulos do Hadoop. O HDFS é um sistema de arquivos distribuído que permite o armazenamento de um grande volume de dados em diversos nodos de um *cluster*. As grandes vantagens do HDFS são: a portabilidade, capacidade de armazenamento, o custo-benefício e a tolerância a falhas (KUMAR et al., 2014).

[1] http://hadoop.apache.org/

O YARN é um *framework* para agendamento de tarefas e gerenciamento de recursos em *cluster*. Por fim, o *Hadoop MapReduce* é um método para distribuir tarefas em múltiplos nodos, o que permite o processamento paralelo e distribuído, tolerante a falhas e de fácil abstração. Como o próprio nome sugere, ele é baseado no *MapReduce*, um modelo de programação e *framework* introduzido pelo Google (KUMAR et al., 2014). A grande desvantagem do Hadoop é que seu processamento ocorre em disco, o que limita sua velocidade (SHORO; SOOMRO, 2015).

Já o Apache Spark[2], ou simplesmente Spark, é uma ferramenta de propósito geral para processamento em *cluster*, que realiza operações em memória. Esta característica permite o aceleramento da análise de dados, tornando mais rápido, tanto as operações de escrita quanto o processamento de dados. Em alguns casos o Spark pode ser até cem vezes mais rápido que o Hadoop.

Além disso, o Spark possui APIs para diversas linguagens de programação, como Python, Java, Scala e R. Outra vantagem, é a existência de diversas ferramentas de alto nível que auxiliam no processamento de dados, como por exemplo a M Lib, uma biblioteca para aprendizado de máquina e o Spark SQL, uma biblioteca que permite, entre outras coisas, a realização de operações como *join* e *order by* em qualquer conjunto de dados, mesmo aqueles originários de bancos de dados NoSQL (SHORO; SOOMRO, 2015).

Assim, dada uma fonte de dados NoSQL é possível utilizar o Hadoop ou Spark para acessar, processar e até mesmo alimentar essa fonte. Por fim, deve-se permitir o acesso aos dados já processados que são armazenados, isso pode ser realizado utilizando *web services*.

## 2.3 *Web Services*

A troca de informações entre aplicações distribuídas na web é feita através de protocolos de comunicação (SCHEPKE et al., 2010). Estes protocolos permitem, entre outras operações, a recuperação de dados de aplicações que possuem esse acesso liberado. Serão discutidos duas das formas de comunicação, conhecidas como *web services*, mais utilizadas, o *Simple Object Access Protocol (SOAP)*, e o *Representational State Transfer (REST)* (LIMA, 2012).

O SOAP é um protocolo adotado pela W3C, que permite invocar aplicações remotas independente de linguagem de programação e plataforma. O protocolo é baseado em XML, e utiliza o *Hypertext Transfer Protocol (HTTP)* para transporte da mensagem. Uma mensagem SOAP é composta por três elementos: (1) envelope, (2) cabeçalho e (3) corpo (SUDA, 2003).

[2] http://spark.apache.org/

O envelope SOAP é o recipiente que armazena os outros elementos da mensagem, como o cabeçalho e o corpo. O cabeçalho é um elemento opcional, que contém informações adicionais, como por exemplo, se a mensagem deve ser processada por um nó intermediário antes de chegar ao ponto final da aplicação. O corpo SOAP é um elemento obrigatório, que armazena os dados da mensagem transportada. No caso de uma mensagem de requisição, o corpo pode conter o método a ser chamado e os parâmetros de entrada e saída do método, já para uma mensagem de resposta o corpo contém o resultado (dados), gerado pelo método chamado (SUDA, 2003).

O modelo REST foi definido por Fielding (2000), que buscou as melhores práticas de arquiteturas de *web services* já existentes e compôs uma nova arquitetura que as reunissem em apenas uma. Essa arquitetura reúne as melhores práticas no que se refere a: (1) cliente/servidor, (2) sistemas de camadas, (3) cache e (4) sem estado (FIELDING, 2000).

No modelo REST são definidos dois papéis, cliente e servidor. O servidor oferece uma série de serviços do qual o cliente faz uso. Ao receber uma requisição do cliente o servidor decide o que fazer com ela, aceitar a requisição ou rejeitá-la (FIELDING, 2000).

Outra característica do modelo REST é a divisão em camadas. O sistema é dividido de forma que uma camada inferior conhece apenas a interface da camada superior, isso melhora a escalabilidade do sistema, mas adiciona uma sobrecarga no tratamento dos dados, o que pode ser combatido utilizando cache (FIELDING, 2000).

O cache evita que dados, que já tenham sido enviados anteriormente, sejam reenviados, isso melhora a eficiência, escalabilidade e performance dos servidores. Finalmente, o sistema não possui estado, ou seja, as informações para atender uma requisição estão contidas nela mesma (FIELDING, 2000).

Diferentemente do SOAP, os *web services* REST não possuem um formato padrão para envio de mensagens, os mais comuns são o JSON e XML. Além disso, a arquitetura REST utiliza o protocolo HTTP e seus métodos para manipulação de recursos. O termo recursos se refere a qualquer estrutura que pode ser armazenada em um computador. Esses métodos permitem, entre outras operações, a exclusão, atualização, inserção e recuperação de recursos (LIMA, 2012).

A desvantagem do serviço SOAP é o uso limitado do protocolo HTTP, já que utilizam um único método para realizar múltiplas operações, enquanto serviços REST utilizam todos os métodos disponíveis. Outra desvantagem do SOAP é a falta de flexibilidade na definição do formato das mensagens. No entanto, embora os serviços REST sejam mais flexíveis isso também pode ser um problema, já que em serviços flexíveis a interoperabilidade pode ficar prejudicada. Não existe serviço melhor que o outro, a escolha deve ser feita de acordo com o contexto (LIMA, 2012).

# 3 Trabalhos Relacionados

Na literatura são encontrados trabalhos referentes a arquitetura de sistemas de visualização, integração e análise de dados, no contexto dos DGA. Por exemplo, Graves e Hendler (2013) mostram como o uso de visualizações podem beneficiar a população, que não possui conhecimento técnico, no contexto de DGA. Além disso, os autores demonstram a necessidade de criar mecanismos de exploração para navegar por dados e metadados, e quais as funcionalidades uma ferramenta deve contemplar para facilitar a criação de visualizações.

Para isso são relatadas três fases em que problemas relacionados a visualizações aparecem, como tratar esses problemas, e a apresentação de um protótipo que os resolvem. A primeira fase é a de criação, quando ocorre o processamento dos dados que serão usados. Nessa fase deve-se resolver questões sobre como tratar dados em formatos diversos, e como combinar dados de diferentes bases. A segunda é a fase de exploração, o maior problema nessa fase é a falta de informação que garante a qualidade da visualização, como a origem dos dados e o histórico de alterações feitas no conjunto de interesse. A última é a fase de adaptar a visualização para gerar outros conhecimentos. Nessa fase, devem ser tratados problemas referentes a capacidade de modificação dos dados implícitos na visualização, por exemplo, utilizar a média de uma métrica no lugar da mediana (GRAVES; HENDLER, 2013).

O protótipo proposto por Graves e Hendler (2013) utiliza os princípios de *linked data*, para integrar os dados e torná-los legíveis tanto para humanos quanto para máquinas. Além disso, possui uma interface gráfica que permite a criação de visualizações.

Também seguindo os princípios de *linked data*, Hoxha e Brahaj (2011), propõem a utilização de tecnologias da web semântica para realizar a integração entre dados de diferentes organizações governamentais. Os autores propõem uma abordagem composta por três módulos, o primeiro é responsável por modelar uma ontologia e converter os dados não processados, utilizando o formato do RDF. O segundo é uma interface para consulta a essa base de conhecimento, composto por uma interface gráfica e mecanismo para consultas utilizando o *Sparql Protocol and RDF Query Language* (SPARQL). O terceiro módulo é uma ferramenta de visualização, que faz uso dessa interface de consulta. Finalmente, os autores sugerem uma implementação composta por quatro etapas: processamento dos dados, agregação da informação, visualização gráfica e contribuição da comunidade.

Assim como os trabalhos anteriores, Ding et al. (2010), vêm trabalhando em uma iniciativa para integrar os dados do Data.gov[1] (página web mantida pelo governo dos

[1] http://www.data.gov/

Estados Unidos da América, no qual dados referentes ao governo são disponibilizados) utilizando os princípios de *linked data*. Os autores mostram como as tecnologias de web semântica são utilizadas para converter e integrar esses dados. Para isso quatro problemas são tratados: (1) como tornar os dados capazes de fazer parte da nuvem do *linked data*, (2) como conectar esses dados entre si e com fontes externas, (3) como tornar esses dados utilizáveis para usuários e desenvolvedores e (4) como preservar o histórico de dados (DING et al., 2010).

A solução apresentada por Ding et al. (2010) começa tratando das transformações necessárias para adequar os dados, isso é feito através da conversão para o formato RDF. Após isso, os dados são enriquecidos através de processos semiautomáticos, nos quais os valores semânticos são associados a identificadores uniformes de recursos (URIs) que possuem relevância, para isso é utilizado o Semantic MediaWiki[2], que permite que usuários colaborem na edição de conteúdos semânticos. Esses dados são disponibilizados através de um *webservice* SPARQL, que permite a integração dos dados com APIs convencionais, como a Google Visualization API. Por fim é discutida a importância de metadados, que devem permitir avaliar o histórico dos dados.

Fora do contexto dos DGAs também existem trabalhos que tratam da arquitetura de sistemas de visualização. Viegas et al. (2007) discutem o desenho e desenvolvimento do ManyEyes, um *website* no qual usuários podem enviar dados, criar visualizações interativas e discutir tais visualizações. O objetivo é permitir a colaboração e análise de dados de forma social.

Segundo os autores, as decisões de design envolvem três aspectos; (1) a visualização da informação, (2) a coleta de dados e manipulação por parte dos usuários, e (3) a colaboração de forma social. O site incentiva os usuários a disponibilizarem os metadados e oferece suporte para dados no formato de tabelas e texto não estruturado (VIEGAS et al., 2007). Por fim os autores discutem os tipos de visualização disponibilizados, como é feito o mapeamento dos dados para as visualizações, e os aspectos sociais da ferramenta.

Trabalhos como o realizado por Tang et al. (2004), abordam decisões de projeto, para a arquitetura de sistemas de visualização de dados. Para isso, Tang et al. (2004) discutem os desafios enfrentados no contexto do Rivet[3], um ambiente para desenvolvimento de visualizações. Inicialmente são discutidos três aspectos fundamentais: (1) o modelo de dados, (2) a forma de envio, ou seja, como os dados podem ser importados para a ferramenta, e (3) as capacidades de transformação que devem existir.

Em relação ao modelo de dados é discutido as vantagens e desvantagens do modelo relacional, comumente utilizado na implementação de sistemas. Para a forma de envio, os autores discutem a importância de qualquer dado importado parecer igualmente expressivo

[2] https://www.semantic-mediawiki.org/
[3] https://graphics.stanford.edu/projects/rivet/

para os usuários, ou seja, independente do formato a informação armazenada deve ser a mesma. Para isso eles disponibilizam diversas formas de se importar dados, passando por conversores CSV até drivers para conexão com banco de dados SQL. Quando discutindo os tipos de transformação, os autores mostram algumas opções que parecem ser comuns no processo de análise, e, portanto, essenciais em um sistema de visualização, como agregação e contagem, ordenação e filtros (TANG et al., 2004). Além desses aspectos também são tratadas questões como a importância de metadados, a modularização na arquitetura e pôr fim a forma de especificação para gerar as visualizações.

Para melhor compreender as ferramentas apresentadas, o Quadro 1 sintetiza as principais características das mesmas.

Quadro 1 – Comparação entre os sistemas encontrados na literatura

| Referência | Modelo de Dados | Forma de acesso aos dados | Formato de importação dos dados | Importação de dados por usuários | Acesso a base de dados de outros usuários | Disponibilização de metadados | Cruzamento entre dados[4] |
|---|---|---|---|---|---|---|---|
| OpenData-Vis - Graves e Hendler (2013) | Linked Data | Interface gráfica | Não especificado | Não especificado | Não especificado | Sim | Não |
| Hoxha e Brahaj (2011) | Linked Data | Interface gráfica e consultas SPARQL | Diferentes formatos (XML, CSV, texto) | Não | Não | Sim | Não |
| Data-GovWiki - Ding et al. (2010) | Linked Data | Web-service SPARQL | CSV | Não | Não | Sim | Não |
| Many Eyes - Viegas et al. (2007) | Tabela e texto não estruturado | Interface gráfica | Texto separado por tabulação. | Sim | Sim | Sim | Não |
| Rivet - Tang et al. (2004) | Relacional | API REST | CSV, MDX e conexões SQL | Sim | Não | Sim | Não |

Fonte: O Autor

[4] Utilizando dados enviados por outros usuários

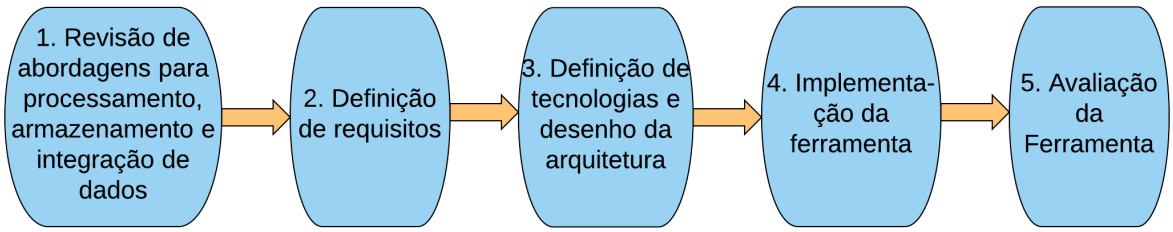
Embora os trabalhos apresentados contemplem informações sobre a infraestrutura que suporta os sistemas de visualização de dados, os autores não detalham os projetos das arquiteturas. Também é possível notar que, embora enalteçam a importância do aspecto social na análise de dados governamentais abertos, não propõem alternativas concretas que permitam a colaboração nas fases mais elementares do processo de análise, por exemplo, durante a inserção ou o gerenciamento dos dados.

A ferramenta aqui proposta busca dar suporte a sistemas de visualização colaborativos, e estender a capacidade de colaboração para além da geração e análise de visualizações. Ou seja, permitir, de forma colaborativa, o processamento e integração de dados distribuídos, e estabelecer relacionamento entre esses dados.

# 4 Metodologia

A metodologia utilizada na realização desse trabalho consiste em 6 etapas, como mostrado na Figura 2.

Figura 2 – Metodologia utilizada no trabalho




Fonte: O Autor

Como demonstrado na Figura 2, o primeiro passo da metodologia consistiu em revisar abordagens já existentes para o processamento, armazenamento e integração de dados abertos. Essa etapa levantou quais as melhores práticas e alternativas a serem aplicadas no contexto dos DGAs. Para isso foi realizado a revisão da literatura acerca do tema, o que deu origem ao capitulo 3.

A segunda etapa consistiu no levantamento dos requisitos funcionais e não funcionais da ferramenta, com o objetivo de delimitar o escopo da mesma. Para isso foi realizada uma reunião, com especialistas na área, e também foi levado em consideração a revisão da literatura feita no passo anterior. Como produto dessa etapa foi gerado um *backlog* com as funcionalidades definidas.

A partir dos requisitos levantados anteriormente, foram definidas as tecnologias e o desenho da arquitetura que possibilitaram o desenvolvimento da ferramenta. Nesse processo se destaca a definição do modelo de dados, a forma de acesso e importação dos dados, e quais tipos de metadados seriam especificados pelos usuários. Isso foi feito com base na revisão das abordagens, realizada no primeiro passo. Após a definição dos requisitos, das tecnologias e feito o desenho da arquitetura, foi realizada a implementação da ferramenta.

Com o término do desenvolvimento a ferramenta foi avaliada segundo a perspectiva do usuário, através de um Teste de Usabilidade (BARBOSA; SILVA, 2010). Essa análise traçou as vantagens e limitações da ferramenta proposta a partir desse ponto de vista.

# 5 Desenvolvimento

Neste capítulo serão apresentadas todas as etapas referentes ao desenvolvimento da ferramenta WikiOlapBase (WOB), que permite a integração de dados abertos de forma colaborativa. Na seção 5.1 serão explicitados os requisitos levantados para a ferramenta. Posteriormente, na seção 5.2, será mostrada a arquitetura proposta para o WOB. Finalmente, na seção 5.3, a ferramenta será apresentada, ainda nessa seção será apresentada uma análise comparativa entre o WOB e as ferramentas encontradas na literatura.

## 5.1 Levantamento de Requisitos

Na etapa de Levantamento de Requisitos foram definidas as funcionalidades e características do software proposto. Esse levantamento ocorreu a partir da revisão da literatura e através de uma reunião de *brainstorming*, no dia 29 de abril de 2016, com três especialistas que possuem mais de oito anos de experiência na área de processamento e análise de dados. O resultado gerado foi a lista de requisitos, mostrada no Quadro 2.

Quadro 2 – Requisitos do WOB

| Identificador | Requisito |
|---|---|
| RF_1 | A ferramenta deve permitir a importação de dados, de forma a manter o significado dos dados originais |
| RF_2 | A ferramenta deve ser capaz de converter diferentes formatos para o modelo de dados definido. |
| RF_3 | A ferramenta deve permitir aos usuários o acesso aos dados presentes no banco de dados integrado da ferramenta. |
| RF_4 | A ferramenta deve permitir a definição de metadados que se relacionam com um determinado conjunto de dados. |
| RF_5 | A ferramenta deve ser capaz de estabelecer relacionamento entre conjunto de dados diferentes. |
| RF_6 | A ferramenta deve aceitar arquivos compactados |
| RF_7 | A ferramenta deve possibilitar a divisão dos conjuntos de dados em múltiplos arquivos para envio. |
| RF_8 | A ferramenta deve disponibilizar uma interface para que outras aplicações acessem os dados presentes na base de dados integrada. |
| RNF_1 | A ferramenta deve ser capaz de armazenar dados em larga escala |
| RNF_2 | A ferramenta deve otimizar o tempo de consulta aos dados. |

Fonte: O Autor

A partir dos requisitos levantados foi possível escolher as tecnologias, bem como propor uma arquitetura, para o desenvolvimento da solução. Na seção a seguir essas escolhas são expostas e justificadas.

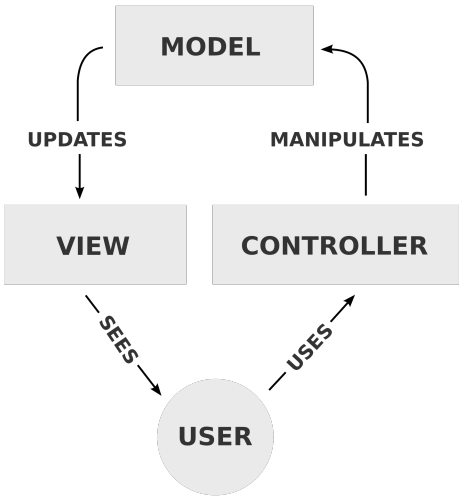
## 5.2 Arquitetura do WikiOlapBase

A arquitetura do WOB foi especificada de modo a definir: (1) a linguagem de programação utilizada, (2) o modelo de dados e os SGBDs utilizados, (3) a forma de acesso aos dados, e qualquer outra decisão de projeto. Essas decisões foram tomadas levando em consideração a revisão bibliográfica e os requisitos da ferramenta.

Para o desenvolvimento da ferramenta foi utilizado o padrão de arquitetura *Model-View-Controller (MVC)*. Nesse padrão, o modelo de dados, a interface do usuário e lógica de controle são separados em três componentes: (1) o modelo, que representa a estrutura de dados e regras de negócio da aplicação, (2) a *view*, que apresenta o modelo para o usuário e (3) o controlador, que interpreta a entrada do usuário e se comunica com o modelo para realizar as mudanças necessárias (PLEKHANOVA, 2009).

Essa separação de conceitos, do inglês *separation of concerns (SoC)*, permite o desenvolvimento e teste de cada componente de forma independente, o que facilita e agiliza o desenvolvimento. Isso também facilita a evolução das funcionalidades de aplicações web, o que justifica a escolha do padrão MVC para o desenvolvimento da ferramenta proposta (GUPTA et al., 2012). A Figura 3 mostra a interação entre os componentes no padrão MVC.

Figura 3 – Interação entre componentes do MVC




Fonte: Wikipedia (2016)

Para a codificação da ferramenta foi utilizada a linguagem de programação Python, através do *framework* Django. Python é uma linguagem de programação popular, que possui suporte para integração com outras linguagens e ferramentas, além de uma variedade de bibliotecas. Já o Django, é um *framework* de código aberto que busca automatizar ao máximo o desenvolvimento, aderindo ao princípio "não repita a si mesmo", do inglês *don't repeat yourself (DRY)* (PLEKHANOVA, 2009). Também vale ressaltar que para a interface

do usuário foram usadas as linguagens *HyperText Markup Language* (HTML), *Cascading Style Sheets (CSS)* e JavaScript.

A arquitetura aqui proposta busca funcionar como infraestrutura base para ferramentas de visualização de grandes volumes de dados que fazem uso de operações OLAP, e portanto, o modelo de dados escolhido deve viabilizar isso. Dessa forma, foi escolhido o modelo de família de colunas, pois ele otimiza esse tipo de operações, que tipicamente envolvem consultas complexas em grandes porções de dados (SORJONEN, 2012).

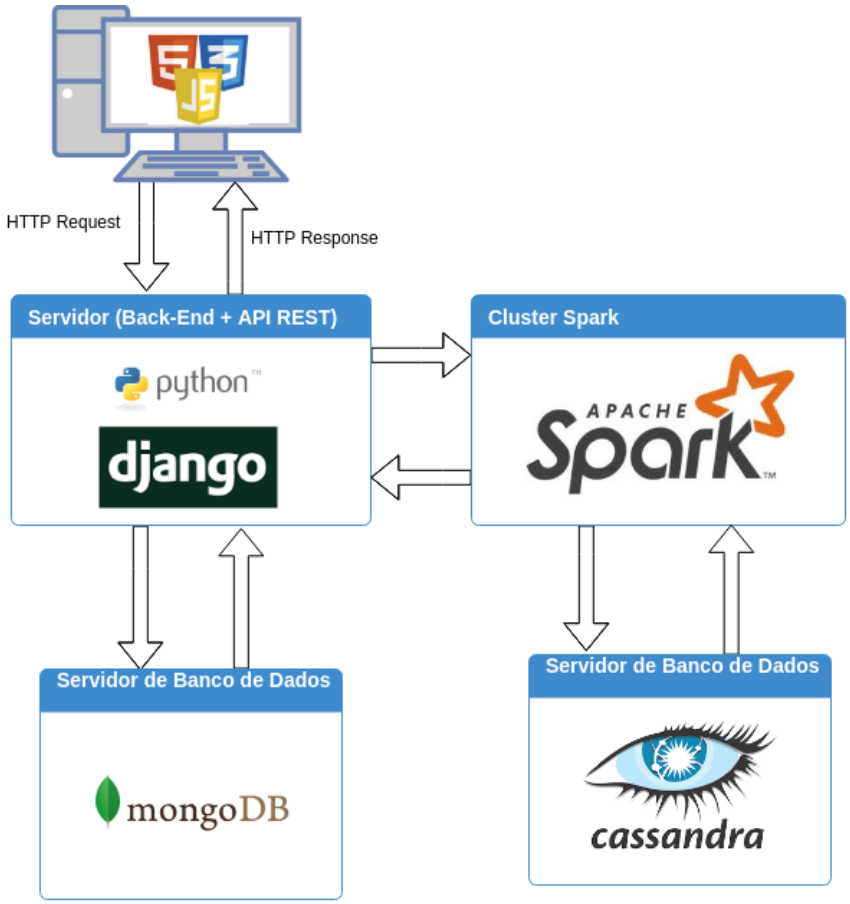
Comparativamente, bancos de dados relacionais, orientados por linha, precisam manipular uma grande quantidade de itens para selecionar os dados necessários para responder a uma consulta, o que torna operações de leitura lentas, e portanto não indicadas quando se deixa realizar uma operação OLAP (SORJONEN, 2012). Já bancos de dados orientados por coluna, acessam apenas os itens necessários para responder uma consulta, o que torna as operações de leitura mais rápidas. Além disso, o modelo de família de colunas é mais escalável, o que no contexto de grandes volumes de dados é uma característica desejada (MONIRUZZAMAN; HOSSAIN, 2013).

Além dos conjuntos de dados que serão armazenados pela ferramenta, também foi necessário registrar os metadados referentes a esses conjuntos, de modo que seja possível caracterizá-los. Metadados são comumente definidos como dados sobre dados, no entanto, vão além dessa definição. Eles permitem ao usuário, ou um computador, procurar e gerenciar informações, definir as regras para uma estrutura de dados e integrar dados de diferentes fontes (TURNER, 2002). Desse modo, para o armazenamento dos metadados, foi utilizado o modelo orientado a documentos. Esse modelo não possui estrutura definida, o que o torna uma boa escolha para armazenamento de metadados (DIANA; GEROSA, 2010).

O SGBD utilizado para implementar e gerenciar o modelo de família de colunas foi o Cassandra, já o modelo orientado a documentos foi implementado no MongoDB. Segundo o site db-engines.com (2016), esses dois SGBDs estão entre os dez mais utilizados, sendo os primeiros colocados em suas categorias. Isso demonstra a popularidade e aceitação da comunidade em relação a essas ferramentas, o que justifica suas escolhas. Devido as limitações dos bancos de dados NoSQL, já explorada anteriormente neste trabalho (na Seção 2.1), também foi utilizada a plataforma Spark para realizar uma interface com o Cassandra. A utilização do Spark permite a realização de operações mais complexas sobre os dados, além de tornar ainda mais rápida a leitura e escrita de dados sobre o Cassandra (KOLACZKOWSKI, 2014).

Finalmente, para o acesso aos dados, foi disponibilizado uma API REST, devido a sua simplicidade e adequação natural a web (MALESHKOVA et al., 2010). A Figura 4 apresenta o diagrama da arquitetura implementada.

Figura 4 – Arquitetura do WikiOlapBase




Fonte: O Autor

## 5.3 WikiOlapBase

A partir da arquitetura proposta na seção 5.2 e nos requisitos definidos na seção 5.1 a ferramenta WikiOlapBase foi implementada. O desenvolvimento foi iniciado no dia 2 de junho de 2016 e a primeira versão funcional foi finalizada no dia 22 de setembro de 2016, demorando, portanto, 3 meses e 20 dias.

A ferramenta proposta possui dois módulos: o primeiro é responsável por receber, caracterizar e integrar os conjuntos de dados enviados pelos usuários, o segundo permite o acesso a esse repositório de dados integrados por meio de uma API REST. O primeiro módulo é composto por uma série de interfaces, nas quais os usuários preenchem os metadados referente a base de dados que está sendo enviada. O conjunto de dados é então processado e armazenado no Cassandra, já os metadados são armazenados no MongoDB.

Para realizar o armazenamento no Cassandra, é utilizada uma API do Spark, o que torna esse processo mais rápido e eficaz (KOLACZKOWSKI, 2014). Já a API REST possui diversos métodos que podem ser acessados para realizar operações como: recuperação

de dados, recuperação de metadados e cruzamento entre diferentes bases de dados. Para que o usuário possa recuperar e pré-processar os dados foi utilizada uma API do Spark, que realiza essas operações de uma forma mais rápida (KOLACZKOWSKI, 2014). A seguir serão detalhados os principais passos do fluxo de execução da ferramenta.

O fluxo de execução principal do WOB é composto por quatro passos, para facilitar o aprendizado do usuário foi elaborada uma interface que explica esses passos, conforme demonstrado nas Figuras 5, 6, 7 e 8.

Figura 5 – Interface de instruções do WOB - Passo 1


Fonte: O Autor

Figura 6 – Interface de instruções do WOB - Passo 2


Fonte: O Autor

Figura 7 – Interface de instruções do WOB - Passo 3

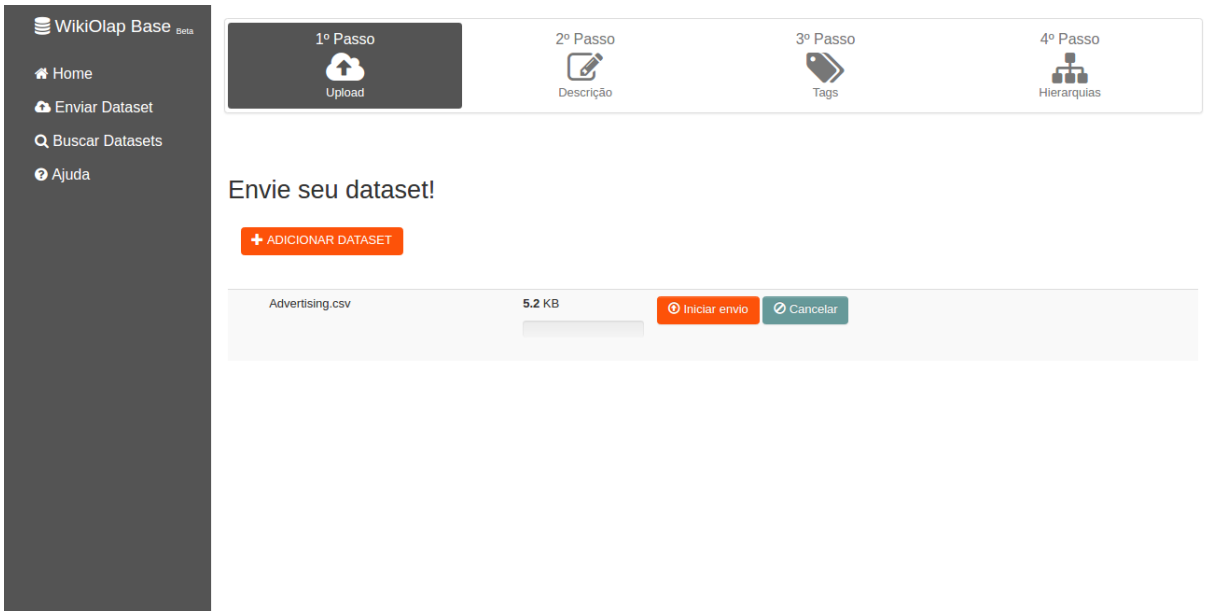

Fonte: O Autor

Figura 8 – Interface de instruções do WOB - Passo 4


Fonte: O Autor

O primeiro passo de execução engloba a seleção e envio do conjunto de dados desejado, vale ressaltar que nessa primeira versão só são aceitos arquivos no formato CSV, a Figura 9 mostra a interface elaborada para essas ações. A partir dos dados enviados o usuário deve preencher os metadados correspondentes, esse procedimento engloba os três passos de execução seguintes, embora seja sugerida uma sequência, o usuário pode realizar essa fase na ordem que desejar.

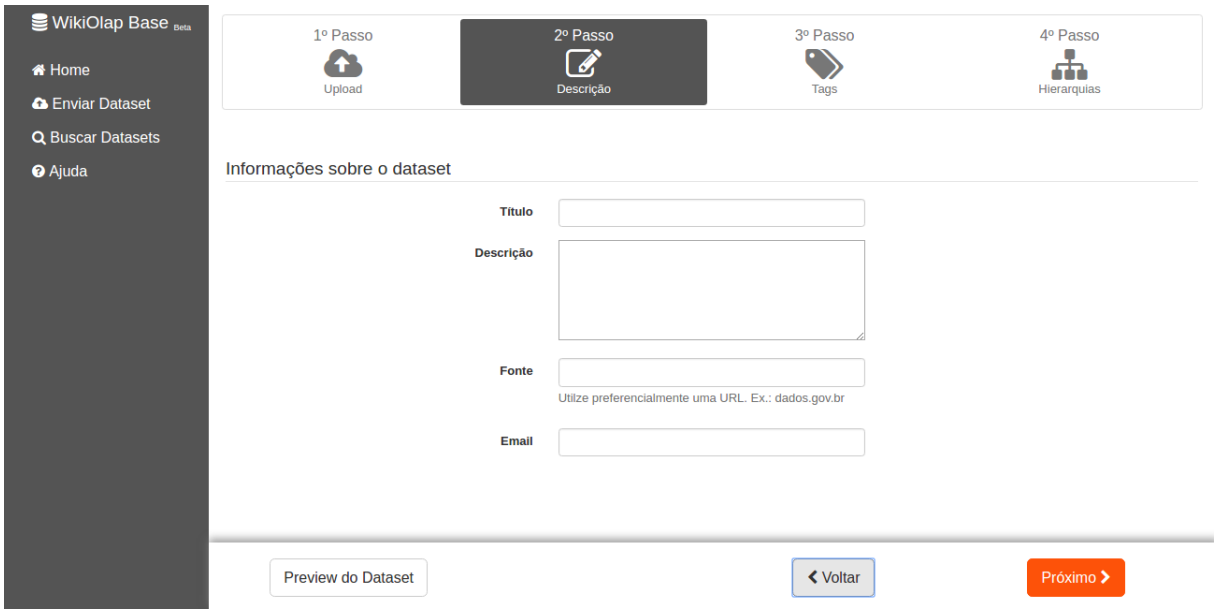
Figura 9 – Interface para envio de arquivo do WOB




Fonte: O Autor

Seguindo a sequência sugerida, primeiro deve ser preenchido informações básicas do conjunto de dados, como fonte, título e descrição. Essas informações permitem a indexação dentro do repositório, possibilitando, posteriormente, que outros usuários possam buscar esses dados, a interface pode ser vista na Figura 10.

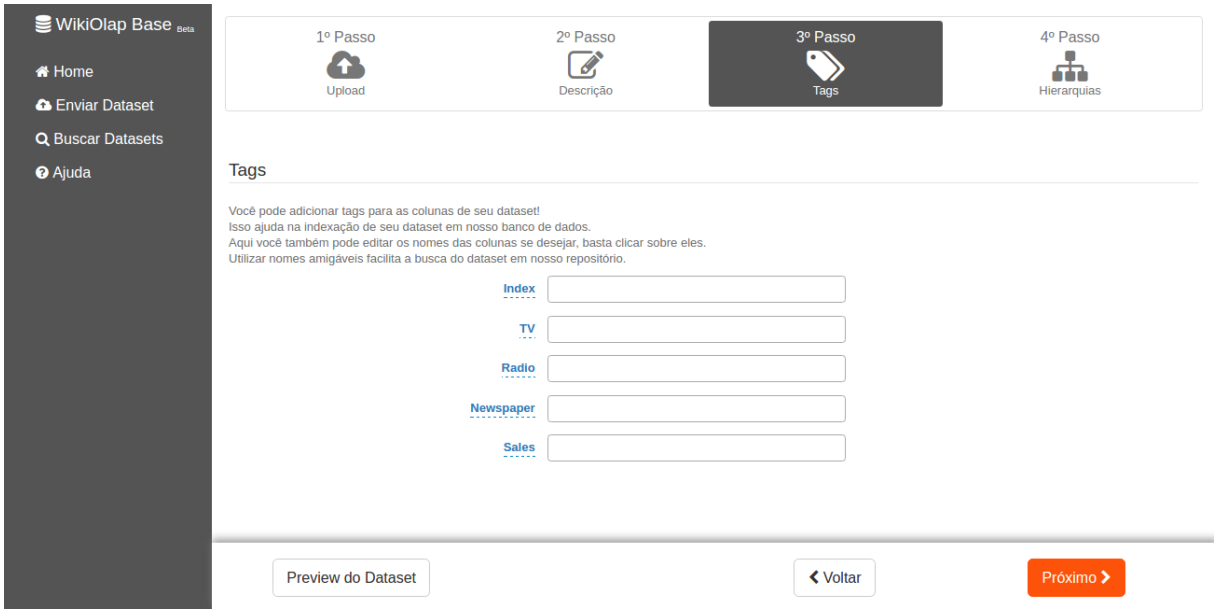
Figura 10 – Interface para preenchimento de informações básicas do WOB




Fonte: O Autor

Logo depois, o usuário pode adicionar *tags* às colunas do conjunto de dados. Isso, além de ajudar na indexação desses dados, também viabiliza o cruzamento entre conjuntos diferentes, pois permite a descoberta de conjuntos de dados que possuem atributos em comum. Nesse ponto o usuário também pode renomear as colunas, se assim o desejar, essa interface é mostrada na Figura 11.

Figura 11 – Interface para preenchimento de tags do WOB




Fonte: O Autor

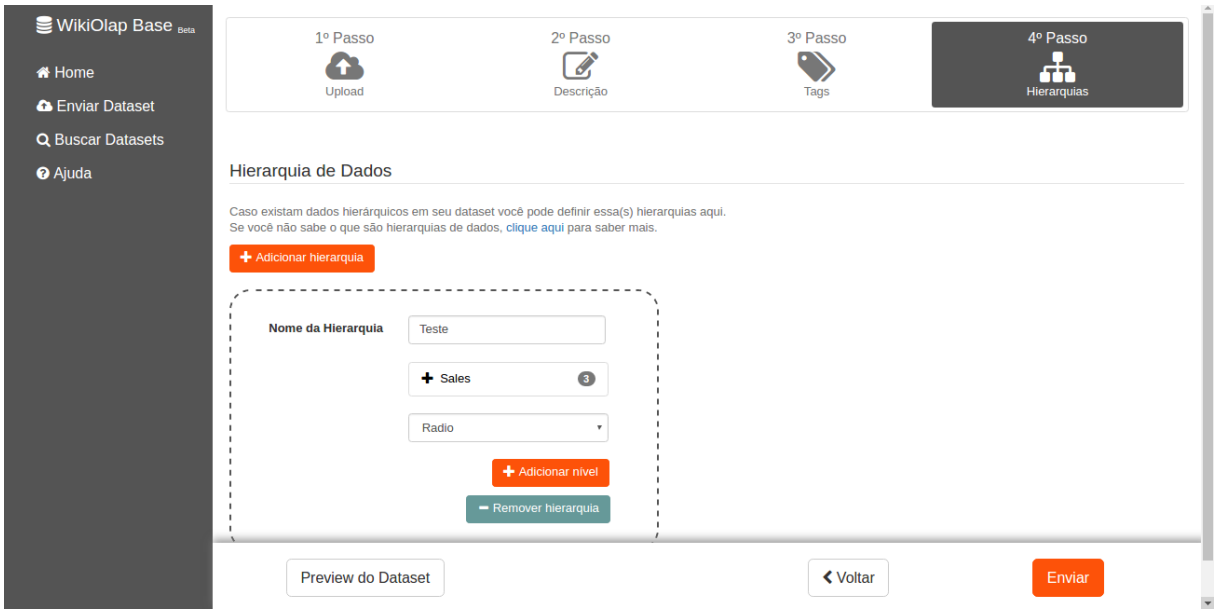
Por fim, o usuário pode identificar hierarquias de dados dentro do conjunto enviado. Essa informação pode ser utilizada na geração de visualizações que utilizam operações OLAP como *drill down* e *drill up*, a interface pode ser vista na Figura 12. Além disso também é disponibilizado ao usuário um *preview* de seu conjunto de dados, desse modo ele pode verificar se não ocorreu um erro ao enviar seu arquivo.

Essa sequência de ações, realizadas pelo usuário da ferramenta, viabiliza a integração entre o conjunto de dados enviado por ele e os que já existem no repositório. Além disso, o preenchimento consciente dos metadados faz parte do aspecto colaborativo da ferramenta, já que isso possibilita a reutilização dos conjuntos enviados por qualquer usuário que assim o desejar.

Finalmente, para acessar o repositório do WikiOlapBase, e realizar operações em cima dos conjuntos de dados disponíveis, os usuários podem utilizar a API REST que foi desenvolvida, sua documentação[1] já se encontra disponível. Embora a ferramenta em si ainda não esteja disponível para o público em geral, o código fonte[2] já é de domínio público.

[1] http://docs.wikiolapapi.apiary.io/
[2] https://github.com/pedromb/wikiolapbase

Figura 12 – Interface para indentificação de hierarquias do WOB




Fonte: O Autor

Percebe-se que o WikiOlapBase possui algumas características que o diferenciam das outras ferramentas mostradas no capítulo 3. O Quadro 3 mostra um comparativo entre as ferramentas e o WOB.

Quadro 3 – Comparação entre os sistemas encontrados na literatura e o WikiOlapBase

| Referência | Modelo de Dados | Forma de acesso aos dados | Formato de importação dos dados | Importação de dados por usuários | Acesso a base de dados de outros usuários | Disponibilização de metadados | Cruzamento entre dados[3] |
|---|---|---|---|---|---|---|---|
| OpenData-Vis - Graves e Hendler (2013) | Linked Data | Interface gráfica | Não especificado | Não especificado | Não especificado | Sim | Não |
| Hoxha e Brahaj (2011) | Linked Data | Interface gráfica e consultas SPARQL | Diferentes formatos (XML, CSV, texto) | Não | Não | Sim | Não |
| Data-GovWiki - Ding et al. (2010) | Linked Data | Web-service SPARQL | CSV | Não | Não | Sim | Não |
| Many Eyes - Viegas et al. (2007) | Tabela e texto não estruturado | Interface gráfica | Texto separado por tabulação. | Sim | Sim | Sim | Não |
| Rivet - Tang et al. (2004) | Relacional | API REST | CSV, MDX e conexões SQL | Sim | Não | Sim | Não |
| WikiOlap-Base | Família de colunas | API REST | CSV | Sim | Sim | Sim | Sim |

Fonte: O Autor

Pode-se observar que o WOB possui dois grandes diferenciais, o primeiro é o aspecto colaborativo, já que todo gerenciamento da base de dados é feita pelos próprios usuários. O segundo diferencial é a possibilidade de relacionar conjuntos de dados que estão disponíveis no repositório. Já um aspecto a ser melhorado pelo WOB é a disponibilidade de importação de dados em diferentes formatos, já que na versão inicial só está disponível o formato CSV.

Após o desenvolvimento da ferramenta foi realizada de uma avaliação de usabilidade, com o objetivo de determinar se a ferramenta está adequada ao uso por parte do público alvo. O capítulo seguinte mostra a metodologia utilizada e os resultados alcançados nessa avaliação.

[3] Utilizando dados enviados por outros usuários

# 6 Avaliação do WikiOlapBase

Terminado o desenvolvimento do WikiOlapBase foi realizada a avaliação da ferramenta, segundo a perspectiva dos usuários. Dois fatores são importantes na ferramenta desenvolvida, primeiro que ela seja adequada a utilização por parte do público alvo e segundo que ela permita a colaboração entre usuários (BARBOSA; SILVA, 2010). Neste capítulo são discutidos a metodologia e os resultados da avaliação do WOB na visão de seus usuários.

## 6.1 Metodologia de Avaliação

Com intuito de avaliar a adequação de uso da ferramenta WOB, foi realizado um Teste de Usabilidade. Este teste consiste em um método de avaliação de interface que, além dos avaliadores, envolve a participação de usuários e prevê as seguintes fases: preparação, execução e análise (BARBOSA; SILVA, 2010).

A fase de preparação é subdividida nas etapas de: (1) determinação dos objetivos do teste; (2) definição das tarefas que serão executadas; (3) seleção dos participantes; (4) considerações sobre os aspectos éticos; e (5) execução do teste piloto. Essas etapas geram artefatos que são posteriormente utilizados durante o passo de execução do Teste de Usabilidade. Dentre esses artefatos, incluem-se o *Script* para apresentação do sistema, os cenários de descrição das tarefas, o questionário de seleção dos participantes, o questionário pré-teste e o formulário de consentimento (BARBOSA; SILVA, 2010).

É importante ressaltar que, além das tarefas que serão executadas pelos usuários, são definidas as métricas de usabilidade que serão observadas em cada execução. Para cada medida, são definidos os limites mínimos aceitáveis, os limites máximos possíveis e o valor almejado de usabilidade para cada métrica (BARBOSA; SILVA, 2010).

A execução representa a fase em que ocorre a avaliação do sistema sob a perspectiva dos usuários. O avaliador conduz essa fase, efetuando as etapas de: (1) recebimento do usuário; (2) apresentação do sistema, conforme o *Script* preparado; (3) consentimento formal dos usuários, utilizando para isso o termo de consentimento; (4) questionamento pré-teste, utilizando o questionário preparado; (5) observação das tarefas executadas pelos usuários e (6) a entrevista ou questionário pós-teste (BARBOSA; SILVA, 2010).

Já na terceira fase do método, os dados coletados pelo avaliador são analisados. Nessa fase ocorre a verificação de cada uma das medidas de usabilidade, observadas durante a fase de execução, relacionando-as aos valores almejados durante a preparação. Nesse passo também são classificadas as gravidades dos problemas encontrados e

possivelmente são discutidas as hipóteses relacionadas às causas dos problemas encontrados. Todos estes passos são posteriormente relatados em um relatório final do Teste de Usabilidade (BARBOSA; SILVA, 2010).

Após elucidado a forma de condução do Teste de Usabilidade, é possível relatar como esse método foi conduzido para a avaliação da ferramenta WOB. Na fase de preparação, após estabelecido o objetivo do teste (i.e., avaliar a usabilidade e os mecanismos de colaboração do WOB), foram elaborados os artefatos que seriam utilizados durante as avaliações. São eles: o *Script* da avaliação, o termo de consentimento de participação, os cenários de descrição das tarefas, a ficha de controle da avaliação e o questionário referente ao grau de adequação à usabilidade. Esses artefatos podem ser visualizados no Apêndice A.

Em relação às tarefas, é importante ressaltar que foram considerados os principais cenários de interação com o WOB, conforme segue: (T1) aprender a utilizar a ferramenta a partir das instruções presentes na seção de ajuda, (T2) enviar um conjunto de dados no formato CSV, (T3) observar o *preview* do conjunto de dados, (T4) preencher as informações básicas referentes ao conjunto de dados, (T5) definir *tags* para as colunas do arquivo enviado e renomeá-las, (T6) definir uma hierarquia de dados dentro do conjunto enviado, (T7) enviar os metadados, (T8) verificar se o conjunto de dados foi incluído no repositório utilizando a ferramenta de busca, (T9) utilizar a API disponível para recuperar os dados que foram enviados e gerar visualizações e (T10) utilizar a API para cruzar dois conjuntos de dados distintos e gerar visualizações.

A partir dessas tarefas foram definidos três cenários diferentes que envolvem uma ou mais delas, da seguinte forma: (C1) que envolve enviar um conjunto de dados e gerar uma visualização a partir do mesmo, para isso é necessário realizar as tarefas de T1 a T9; (C2) no qual deve-se enviar um conjunto de dados e fazer o cruzamento do mesmo com outro conjunto já existente no repositório para gerar uma visualização, para isso é preciso realizar as tarefas de T1 a T8, e T10; (C3) que envolve utilizar dois conjuntos de dados já existentes no repositório e gerar uma visualização a partir deles, logo é necessário realizar as tarefas T8 e T10.

A fase de execução do teste de usabilidade do WOB contou com a participação de 6 usuários. Desses, 5 possuem formação na área de computação (Engenharia de Computação ou Sistemas de Informação), o último possui formação em Engenharia Mecânica, porém todos atuam na área de desenvolvimento de software. Além disso, 4 possuem alguma experiência com um dos temas: análise de dados, visualização de dados ou mineração de dados. É importante ressaltar que essa quantidade de usuários se justifica, pois, segundo Nielsen (2000), testes de usabilidade devem ser executados por 3 a 5 usuários.

Para cada um dos cenários propostos foram realizados testes com dois usuários diferentes. Para cada tarefa executada por um usuário, o avaliador considerava o tempo

gasto em sua execução e, além disso, observava e anotava como a tarefa era concluída (i.e., concluída sem erro, concluída com erro ou não concluída). Não era permitido, ao longo da execução, que o avaliador respondesse a perguntas referentes a interface, ou a alguma funcionalidade do WOB. Este tipo de pergunta seria respondida somente no período após cada tarefa, quando também seriam discutidas as dúvidas, dificuldades e sugestões dos usuários.

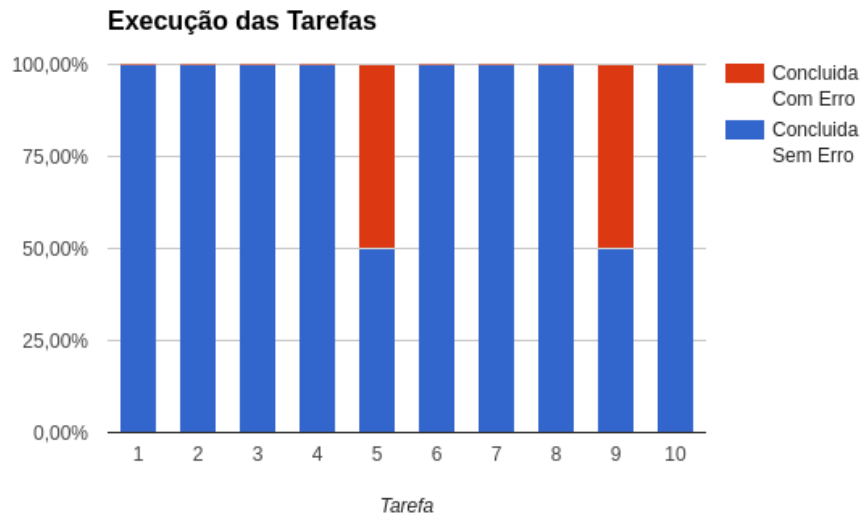
Os testes de usabilidade, com os 6 usuários, ocorreram em um período de 3 dias, entre 27 de setembro de 2016 e 29 de setembro de 2016 sendo que cada um deles teve duração máxima de uma hora.

A partir dos dados obtidos, os resultados foram analisados de forma a caracterizar os indicadores de conclusão das tarefas pelos usuários; o tempo médio decorrido para cada uma das tarefas bem como o tempo médio total (i.e., tempo médio de conclusão de um cenário); e o grau de adequação do WOB aos princípios de usabilidade e colaboração. Através dessas medidas foi possível caracterizar a usabilidade e colaboração do WOB na perspectiva de seus usuários. Os resultados obtidos são discutidos a seguir.

## 6.2 Discussão dos Resultados

Em relação à execução das tarefas, o gráfico da Figura 13 demonstra o percentual de conclusão de cada tarefa.

Figura 13 – Percentual de conclusão das tarefas pelos usuários




Fonte: O Autor

Através desse gráfico é possível observar que todas as tarefas foram concluídas pelos usuários, sendo que apenas 20% foi concluída com erro, sendo a tarefa T5 concluída

com erro por 2 usuários, e a tarefa T9 concluída com erro por 1 usuário.

A Tabela 1, por sua vez, apresenta o tempo decorrido para a execução de cada uma das tarefas pelos usuários e a média do tempo gasto pelos mesmos. Também são exibidos os tempos totais de execução das tarefas por cada usuário.

Tabela 1 – Relação de tempo decorrido em minutos para cada uma das tarefas em cada teste de usabilidade

| Tarefa | U1 | U2 | U3 | U4 | U5 | U6 | Total de Usuários | Tempo Médio |
|---|---|---|---|---|---|---|---|---|
| T1 | 01:07 | 03:09 | 01:56 | 01:20 | 00:00 | 00:00 | 4 | 01:53 |
| T2 | 00:18 | 00:31 | 00:27 | 00:20 | 00:00 | 00:00 | 4 | 00:24 |
| T3 | 01:18 | 00:40 | 00:29 | 00:15 | 00:00 | 00:00 | 4 | 00:40 |
| T4 | 01:22 | 01:01 | 00:51 | 01:38 | 00:00 | 00:00 | 4 | 01:13 |
| T5 | 02:23 | 00:36 | 01:18 | 02:01 | 00:00 | 00:00 | 4 | 01:34 |
| T6 | 02:06 | 00:41 | 02:07 | 00:57 | 00:00 | 00:00 | 4 | 01:27 |
| T7 | 00:24 | 00:04 | 00:10 | 00:05 | 00:00 | 00:00 | 4 | 00:10 |
| T8 | 00:52 | 00:35 | 01:13 | 01:29 | 01:11 | 00:54 | 6 | 01:02 |
| T9 | 03:18 | 03:04 | 00:00 | 00:00 | 00:00 | 00:00 | 2 | 03:11 |
| T10 | 00:00 | 00:00 | 03:35 | 03:52 | 04:33 | 04:13 | 4 | 04:03 |
| Total | 13:08 | 10:21 | 12:06 | 11:57 | 05:44 | 05:07 | - | - |

Fonte: O Autor

Os usuários U1 e U2 executaram o cenário C1, enquanto os usuários U3 e U4 executaram o cenário C2 e os usuários U5 e U6 executaram o cenário C3. Assim, também foi possível calcular o tempo médio por cenário. Verificou-se que C1 possui um tempo médio de execução de 11 minutos e 44 segundos, enquanto o tempo médio para concluir C2 é 12 minutos e 01 segundo, já o tempo médio para concluir C3 foi de 05 minutos e 25 segundos.

A tarefa T1 apresentou um tempo de execução similar entre a maioria dos usuários. Essa tarefa envolve acessar a página de instruções da ferramenta, para aprender sobre seu funcionamento. Embora essa tarefa não tenha gerado dificuldades ou dúvidas, foi possível perceber que a maioria dos usuários prefere aprender a utilizar a ferramenta durante a própria execução. Isso explica o tempo discrepante que ocorreu, pois um usuário se mostrou mais interessado em entender as instruções de como utilizar a ferramenta antes de utilizá-la de fato. Já a tarefa T2 foi executada sem maiores problemas e sem grande variação de tempo entre os usuários.

A tarefa T3 apresentou uma pequena variação nos tempos de execução entre os usuários. Nessa tarefa os usuários deveriam verificar o conjunto de dados enviado a partir da funcionalidade “*preview*”. Durante a avaliação, 3 usuários reportaram dificuldades em localizar essa funcionalidade e, além disso, apresentaram sugestões de melhorias para a

visibilidade dessa função. Embora esse tenha sido considerado um problema cosmético, as sugestões apontadas pelos usuários serão implementadas na próxima versão do WOB.

Já na execução da tarefa T4 não existiu nenhuma dúvida ou dificuldade. Uma tarefa que apresentou grande variação entre os tempos de execução foi a T5, além disso ela foi concluída com erro por 2 usuários. Nessa tarefa os usuários deveriam inserir *tags* para cada coluna do conjunto de dados enviado e renomear suas colunas. Os erros ocorreram por dois motivos: (1) não ficou claro a possibilidade de editar os nomes das colunas e a forma de fazer isso, e (2) no processo para se adicionar uma *tag* é mostrada uma lista com *tags* já utilizadas anteriormente, não ficou claro, segundo os usuários, que era possível adicionar uma nova *tag* que não existia nessa lista. Esse problemas serão corrigidos em versões futuras da ferramenta para melhorar a usabilidade do sistema.

A tarefa T6, embora tenha sido concluída sem erros por todos usuários, também apresentou problemas, o que fica claro com a variação entre os tempos de execução. Nessa tarefa deveria ser especificada uma hierarquia de dados dentro do conjunto de dados enviado. Dois usuários reportaram a falta de *feedback* ao se definir a hierarquia, ou seja, não era possível saber se a hierarquia havia sido salva ou não. Esse problema também será corrigido em uma futura versão da ferramenta.

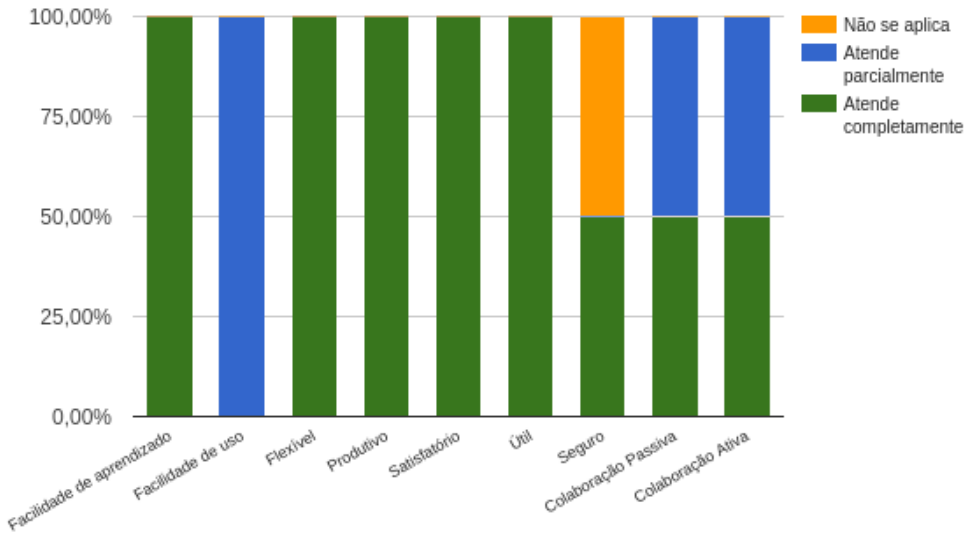
Na execução das tarefas T7, T8 e T10 não ocorreram problemas. Já a tarefa T9 foi concluída com erro por um usuário. Nessa tarefa os usuários deveriam utilizar a API disponível para recuperar o conjunto de dados enviado, e a partir disso gerar uma visualização. Uma possível explicação, para o erro cometido nessa tarefa, é a falta de experiência do usuário na utilização de *web services*. Vale ressaltar que, embora o objetivo da ferramenta seja permitir sua utilização, mesmo por parte de usuários inexperientes, o projeto prevê uma segunda fase na qual será desenvolvida uma ferramenta de visualização de dados que irá consumir a API desenvolvida. Logo, isso não deverá ser um problema para usuários menos experientes no futuro, pois o acesso direto a API não será necessário.

Conforme mencionado anteriormente, depois de executadas as tarefas, cada usuário avaliou a ferramenta sob a perspectiva dos 07 princípios de usabilidade (NIELSEN, 1994), além dos 02 princípios de colaboração definidos especificamente para essa ferramenta, são eles: (1) colaboração passiva e (2) colaboração ativa. Para que seja realizada uma análise mais assertiva, as respostas foram separadas de acordo com o cenário para o qual cada usuário foi submetido, dessa forma a Figura 14 sumariza as respostas para os usuários que realizaram o cenário C1, a Figura 15 para os usuários que realizaram o cenário C2 e a Figura 16 para os usuários que realizaram o cenário C3

O termo colaboração passiva se refere a utilizar um conjunto de dados já existente, ou seja, o grau com que o sistema permite que usuários utilizem conjuntos de dados que foram enviados por outros usuários, já o termo colaboração ativa se refere a enviar um conjunto de dados para outra pessoa utilizar, ou seja, o grau com que o sistema incentiva

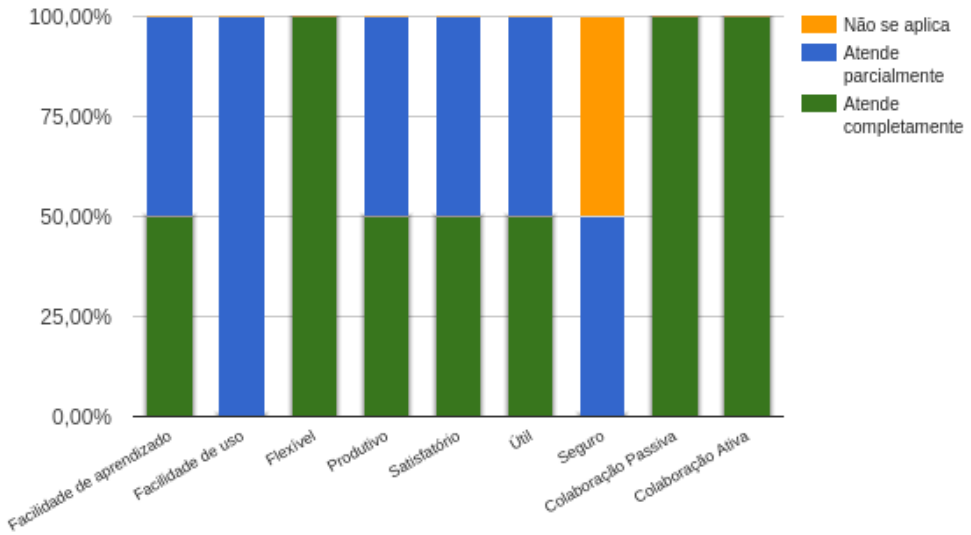
a colaboração entre usuários, mostrando que ao enviar um conjunto de dados ele estará disponível para qualquer um utilizar.

Figura 14 – Grau de adequação do WOB por princípio de usabilidade e colaboração na visão dos usuários - Cenário C1



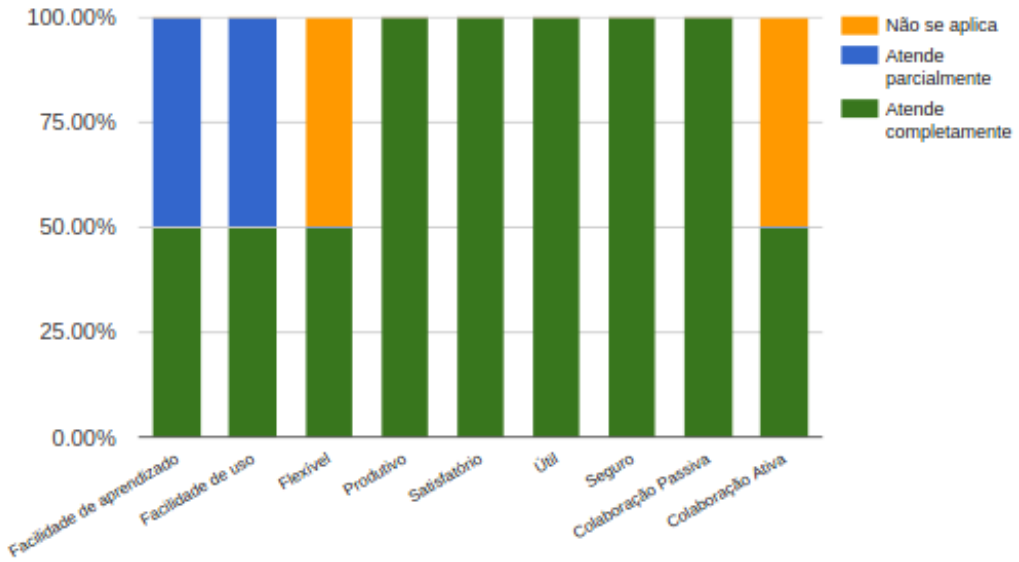

Fonte: O Autor

Figura 15 – Grau de adequação do WOB por princípio de usabilidade e colaboração na visão dos usuários - Cenário C2




Fonte: O Autor

Figura 16 – Grau de adequação do WOB por princípio de usabilidade e colaboração na visão dos usuários - Cenário C3




Fonte: O Autor

Através dos dados apresentados é possível observar que, nos três cenários, nenhum princípio foi julgado como “não atende”, na perspectiva dos usuários, ou seja, para todos os usuários todos os princípios são atendidos ou não são aplicáveis. Foi possível perceber que para 66.67% dos usuários o WOB atende completamente o princípio de facilidade de aprendizado, no entanto, para apenas 16.67% o princípio de facilidade de uso é atendido completamente. Isso mostra que na visão desses usuários, embora, inicialmente, o WOB não apresente uma fácil utilização, aprender como utilizá-lo é uma tarefa simples.

Pode-se notar também, que três princípios foram julgados como não aplicáveis. No cenário C1 e C2 foi o princípio “Seguro”, já no cenário C3 foram os princípios “Flexível” e “Colaboração Ativa”. Os usuários julgaram que o princípio “Seguro” não se aplica pois a ferramenta é aberta, permitindo sua utilização por qualquer pessoa. Já para os casos dos princípios “Flexível” e “Colaboração Ativa” os usuários julgaram que estes não se aplicam devido ao cenário que foram expostos. O cenário C3 envolve apenas a busca por dois conjuntos de dados já existentes no repositório e a utilização da API para o cruzamento desses dados, não envolvendo, portanto, caminhos alternativos ou o envio de conjuntos de dados para a utilização por outros usuários, isso explica a interpretação dos usuários ao responderem que os princípios não se aplicam.

Embora existam melhorias a serem implementadas, tanto que foram identificadas durante o teste, quanto aquelas que já estavam planejadas para futuras versões (ver lista de melhorias no Apêndice B), conclui-se que a ferramenta WOB é adequada ao uso. Isso é

corroborado pela fala dos usuários ao longo dos testes, que aprovaram a idéia por trás da ferramenta bem como seu fluxo de execução. O capítulo seguinte conclui este trabalho, e propõe trabalhos futuros, a serem realizados, a partir do que foi desenvolvido até aqui.

# 7 Conclusões e Trabalhos Futuros

Este trabalho foi desenvolvido com o objetivo de criar o WikiOlapBase, uma ferramenta colaborativa que permite a integração de dados abertos. A metodologia empregada para este projeto consistiu nas etapas de levantamento de ferramentas similares existentes na literatura, levantamento dos requisitos do sistema, elaboração de uma arquitetura que satisfizesse os requisitos, desenvolvimento e teste da ferramenta e avaliação da mesma sob a perspectiva dos usuários.

Os resultados encontrados, a partir do Teste de Usabilidade, demonstram que o WOB é uma ferramenta útil, satisfatória e adequada ao uso, que permite a integração de dados abertos de forma colaborativa. Além disso, o levantamento feito das ferramentas similares, presente neste trabalho, também é relevante, pois aborda uma análise comparativa, que permite explorar diferentes abordagens e técnicas, para a criação de ferramentas para integração de dados.

Assim, este trabalho apresenta contribuições tanto práticas quanto científicas. Como contribuição prática, a ferramenta permitiu adicionar elementos de colaboração no processo de integração de dados, algo que ainda era limitado em outras ferramentas similares. Além disso, outra contribuição prática, é a disponibilização do código fonte[1] gerado. Em termos científicos este trabalho delimitou as vantagens e desvantagens de diferentes abordagens para o processamento, integração e armazenamento de dados, contribuindo para o avanço na discussão destes temas.

## 7.1 Trabalhos Futuros

Como trabalho futuro é proposto a evolução da ferramenta, uma lista de melhorias foi gerada e pode ser acessada no Apêndice B. Além disso outros tipos de teste podem, e devem ser realizados para avaliar o desempenho da ferramenta, como por exemplo a geração de um *benchmarking* para comparar a abordagem proposta com outras.

Também vale ressaltar que o projeto já prevê uma segunda fase, na qual deve ser desenvolvida uma ferramenta de visualização de dados, que consome a API REST disponibilizada neste trabalho. Finalmente, devido a natureza distribuída da maioria das tecnologias utilizadas no desenvolvimento dessa ferramenta, a infraestrutura necessária, e a melhor maneira de disponibilizá-la para o público, também deve ser estudada.

[1] https://github.com/pedromb/wikiolapbase

# Referências


BARBOSA, S.; SILVA, B. S. D. **Interação Humano-Computador**. São Paulo: Elsevier, 2010. Citado 3 vezes nas páginas 12, 23 e 24.

DIANA, M. D.; GEROSA, M. A. Nosql na web 2.0: Um estudo comparativo de bancos não-relacionais para armazenamento de dados na web 2.0. 2010. Citado 4 vezes nas páginas 3, 4, 5 e 15.

DING, L. et al. Data-gov wiki: Towards linking government data. In: . [S.l.: s.n.], 2010. Citado 4 vezes nas páginas 8, 9, 10 e 22.

FIELDING, R. T. **Architectural styles and the design of network-based software architectures**. Tese (Doutorado) — University of California, Irvine, 2000. Citado na página 7.

GRAVES, A.; HENDLER, J. Visualization tools for open government data. In: **Proceedings of the 14th Annual International Conference on Digital Government Research**. New York, NY, USA: ACM, 2013. (dg.o '13), p. 136–145. ISBN 978-1-4503-2057-3. Disponível em: <http://doi.acm.org/10.1145/2479724.2479746>. Citado 5 vezes nas páginas 1, 2, 8, 10 e 22.

GUPTA, P. et al. Utilizing asp.net mvc in web development courses. **J. Comput. Sci. Coll.**, Consortium for Computing Sciences in Colleges, USA, v. 27, n. 3, p. 10–14, jan. 2012. ISSN 1937-4771. Disponível em: <http://dl.acm.org/citation.cfm?id=2038772.2038778>. Citado na página 14.

HOXHA, J.; BRAHAJ, A. Open government data on the web: A semantic approach. In: IEEE. **Emerging Intelligent Data and Web Technologies (EIDWT), 2011 International Conference on**. [S.l.], 2011. p. 107–113. Citado 4 vezes nas páginas 1, 8, 10 e 22.

JAGADISH, H. V. et al. Big data and its technical challenges. **Commun. ACM**, ACM, New York, NY, USA, v. 57, n. 7, p. 86–94, jul. 2014. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/2611567>. Citado na página 2.

KOLACZKOWSKI, P. Lightning fast cluster computing with spark and cassandra. Apresentação no evento CodeMesh-London, disponível em https://www.infoq.com/presentations/spark-cassandra, acesso em 18-Outubro-2016. 2014. Citado 3 vezes nas páginas 15, 16 e 17.

KUMAR, R. et al. Apache hadoop, nosql and newsql solutions of big data. **International Journal of Advance Foundation and Research in Science & Engineering (IJAFRSE)**, v. 1, n. 6, p. 28–36, 2014. Citado 2 vezes nas páginas 5 e 6.

LIMA, J. **Web Services (SOAP x REST)**. 2012. Trabalho de Conclusão de Curso para a Faculdade de Tecnologia de São Paulo. Citado 2 vezes nas páginas 6 e 7.

MALESHKOVA, M. et al. Investigating web apis on the world wide web. In: **Web Services (ECOWS), 2010 IEEE 8th European Conference on**. [S.l.: s.n.], 2010. p. 107–114. Citado na página 15.

MCBRIDE, B. Jena: Implementing the rdf model and syntax specification. In: CEUR-WS. ORG. **Proceedings of the Second International Conference on Semantic Web-Volume 40**. [S.l.], 2001. p. 23–28. Citado na página 4.

MONIRUZZAMAN, A.; HOSSAIN, S. A. Nosql database: New era of databases for big data analytics-classification, characteristics and comparison. **arXiv preprint arXiv:1307.0191**, 2013. Citado 3 vezes nas páginas 3, 4 e 15.

NIELSEN, J. Usability inspection methods. In: ACM. **Conference companion on Human factors in computing systems**. [S.l.], 1994. p. 413–414. Citado na página 27.

NIELSEN, J. **Why You Only Need to Test with 5 Users**. 2000. [Online; acesso 23-Outubro-2016]. Disponível em: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>. Citado na página 24.

PLEKHANOVA, J. Evaluating web development frameworks: Django, ruby on rails and cakephp. **Institute for Business and Information Technology**, 2009. Citado na página 14.

POKORNY, J. Nosql databases: a step to database scalability in web environment. **International Journal of Web Information Systems**, Emerald Group Publishing Limited, v. 9, n. 1, p. 69–82, 2013. Citado na página 5.

SAGIROGLU, S.; SINANC, D. Big data: A review. In: IEEE. **Collaboration Technologies and Systems (CTS), 2013 International Conference on**. [S.l.], 2013. p. 42–47. Citado na página 3.

SCHEPKE, C. et al. **Avaliação de Desempenho de SOAP sobre HTTP, SMTP e BEEP**. 2010. Citado na página 6.

SHORO, A. G.; SOOMRO, T. R. Big data analysis: Apache spark perspective. **Global Journal of Computer Science and Technology**, v. 15, n. 1, 2015. Citado na página 6.

SORJONEN, S. Olap query performance in column orientde databases (december 2012). In: . [S.l.: s.n.], 2012. Citado na página 15.

SUDA, B. **SOAP Web Services**. Dissertação (Mestrado) — University of Edinburgh, 2003. Citado 2 vezes nas páginas 6 e 7.

TANG, D. et al. Design choices when architecting visualizations. **Information Visualization**, Palgrave Macmillan, v. 3, n. 2, p. 65–79, jun. 2004. ISSN 1473-8716. Disponível em: <http://dx.doi.org/10.1057/palgrave.ivs.9500067>. Citado 3 vezes nas páginas 9, 10 e 22.

TURNER, T. What is metadata. **Kaleidoscope**, v. 10, n. 7, p. 1–3, 2002. Citado na página 15.

VAZ, J. C. et al. Dados governamentais abertos e seus impactos sobre os conceitos e práticas de transparência no brasil. **Cadernos ppg-au/ufba**, v. 9, n. 1, 2010. Citado na página 1.

VIEGAS, F. B. et al. Manyeyes: A site for visualization at internet scale. **IEEE Transactions on Visualization and Computer Graphics**, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 13, n. 6, p. 1121–1128, nov. 2007. ISSN 1077-2626. Disponível em: <http://dx.doi.org/10.1109/TVCG.2007.70577>. Citado 3 vezes nas páginas 9, 10 e 22.

WIKIPEDIA. **Model–view–controller — Wikipedia, The Free Encyclopedia**. 2016. [Online; acesso 18-Outubro-2016]. Disponível em: <https://en.wikipedia.org/wiki/Model-view-controller>. Citado na página 14.

# Apêndices

# APÊNDICE A – Artefatos de Avaliação

**SCRIPT DA AVALIAÇÃO**

**Script para a avaliação de Usabilidade do WikiOlapBase**

- **Recepção do participante:**
  - Boas vindas e agradecimento ao participante

- **Explicar sobre o Sistema:**
  - Apresentar o WikiOlapBase (seu objetivo) e suas funcionalidades básicas:
    - Integração de dados abertos;
    - Colaboração entre usuários;
    - Acesso a dados integrados para geração de análises e visualizações;

- **Explicar o Objetivo da pesquisa:**
  - Avaliar a usabilidade do WikiOlapBase.
  - Explicitar que o que será avaliado é o sistema e não o usuário.

- **Explicar sobre a Realização da avaliação**:
  - Explicar sobre a sala de teste (gravação da interação através da interface e câmera filmadora do ambiente), sobre os observadores e sobre o anonimato da pesquisa

  - Explicar os passos do teste, sobre:
    - Leitura e assinatura do termo de consentimento
    - Duração máxima do teste (1 hora)
    - Realização das tarefas pelo usuário
      - Dizer que os observadores não poderão responder perguntas relacionadas ao sistema que está sendo avaliado;
      - Dizer que o usuário deverá dizer em voz alta tudo que está pensando e/ou fazendo durante a execução das tarefas.
    - Entrevista pós-teste
      *"A entrevista pós-teste permite que se obtenha 2 tipos de dados distintos: (1) explicações sobre ações observadas durante a execução de tarefas; (2) aspectos relacionados à experiência do participante e sua satisfação com o sistema."*

  - Reforçar que o objetivo é avaliar o sistema (não o usuário)

  - Tirar todas as dúvidas do usuário antes de iniciar o teste.

**EXECUÇÃO DOS TESTES (Apenas para o avaliador)**

**Sobre a Execução dos testes**

- **Leitura/assinatura do termo de consentimento**
  - Pedir ao usuário para ler e (se desejar) assinar o termo de consentimento

- **Início da avaliação**:
  - Observar o usuário e anotar pontos relevantes e julgar necessário no "Formulário de Observação"

- **Término da avaliação**:
  - Realizar a entrevista Pós-Teste;
  - Agradecer a participação voluntária do participante

**TERMO DE CONSENTIMENTO DE PARTICIPAÇÃO DO USUÁRIO**

O termo de consentimento é exigido por lei (Resolução 196/96 do Conselho Nacional de Saúde – disponível em: *http://conselho.saude.gov.br/resolucoes/1996/Reso196.doc*). O objetivo do mesmo é deixar claro para o participante como os dados serão utilizados, garantir seu anonimato, deixar claro que sua participação é voluntária e pode ser interrompida a cada momento, além de disponibilizar o contato dos pesquisadores responsáveis.

**Termo de Consentimento de Participação**

**Título:** Avaliação de Usabilidade do WikiOlapBase
**Data:** Setembro/2016
**Instituição:** DECOM/CEFET-MG
**Avaliadores Responsáveis:**
Pedro Magalhães Bernardo (pedromagalhaesbernardo@gmail.com)
Ismael Silva (ismaelsantana@decom.cefetmg.br)
Glívia Barbosa (gliviabarbosa@decom.cefetmg.br)

**Introdução:** Este Termo de Consentimento contém informações sobre a avaliação indicada acima. Para assegurar que você esteja informado sobre a sua participação nesta pesquisa, pedimos que ouça a leitura deste Termo de Consentimento. Caso tenha alguma dúvida, não hesite em perguntar ao avaliador responsável. Você também deverá assinar o termo do qual receberá uma cópia.

**Objetivo da avaliação:** O objetivo desta avaliação é caracterizar a usabilidade e potencial de colaboração da ferramenta WikiOlapBase, criada para integrar dados abertos de forma colaborativa e permitir o acesso aos dados integrados, de forma a gerar diferentes visualizações e análises.

**Informação geral sobre a avaliação:** Você será solicitado a realizar algumas tarefas simples utilizando o sistema. A realização dessas tarefas será gravada para posterior análise pelos investigadores. Ao fim da execução das tarefas, será realizada uma entrevista sobre sua experiência com o sistema.

**Utilização dos dados coletados:** Os dados coletados durante a avaliação serão utilizados para a análise de usabilidade do WikiOlapBase. Quaisquer dados utilizados para publicação serão apresentados de forma a garantir o anonimato dos participantes da avaliação.

**Privacidade:** Informações que possam identificar os participantes da avaliação não serão divulgadas. O seu nome não aparecerá em nenhum relatório. Caso deseje, poderá solicitar uma cópia dos dados gerados por você.

**Se você decidir não participar na avaliação:** Você é livre para decidir, a qualquer momento, se quer participar ou não nesta avaliação, podendo inclusive interrompê-la se achar necessário.

**Compensação:** A participação nesta avaliação é voluntária, e não será oferecida nenhuma remuneração aos seus participantes.

**Se tiver algum problema ou se tiver outras perguntas:** Se você tiver algum problema que pensa que pode estar relacionado com sua participação nesta avaliação, ou se tiver qualquer pergunta sobre a mesma, poderá entrar em contato com os avaliadores a qualquer momento pelo e-mail pedromagalhaesbernardo@gmail.com

**Novas condições:** Caso deseje, você pode especificar novas condições que devem ser atendidas para que você participe desta avaliação.

**Consentimento Livre e Esclarecido (Acordo Voluntário)**

O documento mencionado acima descrevendo as condições de participação da "Avaliação de Usabilidade do WikiOlapBase" foi explicado. Eu tive a oportunidade de fazer perguntas sobre a avaliação, que foram respondidas satisfatoriamente. Eu estou de acordo em participar como voluntário.

Data: _____ _____

Assinatura do participante

_____

Nome do participante

_____

Assinatura do pesquisador

_____

Nome do pesquisador

**TAREFAS A SEREM EXECUTADAS**

- **Cenário 1**

Você ouviu dizer que a Contribuição Provisória sobre Movimentações Financeiras (CPMF) pode voltar. Curioso sobre os efeitos deste imposto resolveu pesquisar mais a fundo e conseguiu um arquivo CSV com a série histórica mostrando a quantia arrecadada através desse imposto desde sua criação em 1997 até 2012. No entanto apenas o arquivo não é suficiente e você resolveu gerar um gráfico para entender melhor os dados. Decidiu então utilizar a ferramenta WikiOlapBase para facilitar seu trabalho. Para isso você seguiu a seguintes tarefas.

| **T1: Tarefa 01** - Aprender a utilizar a ferramenta |
|---|
| Como é sua primeira vez utilizando a WikiOlapBase você deve entrar no site da ferramenta e acessar a página de instruções para aprender como utilizá-la. |

| **T2: Tarefa 02 –** Enviar seu arquivo CSV. |
|---|
| Você deve enviar o arquivo que contém os dados que você deseja integrar com o repositório do WikiOlapBase. |

| **T3: Tarefa 03 –** Ver o preview de seu dataset |
|---|
| Para garantir que você selecionou o arquivo correto você deve utilizar a função de preview da ferramenta para confirmar que o arquivo enviado foi o correto. |

| **T4: Tarefa 04 –** Preencher informações básicas do dataset |
|---|
| Você deve preencher as informações básicas do dataset referente a fonte, título e descrição, além disso deve informar um email de contato. Preencha com as seguintes informações:<br>● Título: CPMF<br>● Descrição: Série histórica da arrecadação com CPMF de 1997 a 2012.<br>● Fonte: http://www.ipeadata.gov.br/<br>● Email: teste@teste.com.br |

| **T5: Tarefa 05 –** Definir tags para as colunas do seu dataset e renomeá-las |
|---|
| Como você deseja que outras pessoas também possam utilizar seu dataset você irá preencher tags para cada coluna de seu dataset e mudar os nomes das colunas para um nome mais agradável. Faça as seguintes alterações<br>● cd_ano -> ano<br>● cd_mes -> mes<br>E adicione as tags:<br>● ano: ano, tempo<br>● mes: mes, tempo<br>● cpmf: cpmf, imposto |

| **T6: Tarefa 06 –** Definir uma hierarquia para seu dataset |
| --- |
| Você deseja fazer um gráfico após terminar de enviar seu arquivo, por isso irá definir uma hierarquia em seu conjunto de dados para utilizar essa informação posteriormente. Defina a seguinte hierarquia:<br>● Nome: Tempo<br>● Hierarquia: ano -> mes |

| **T7: Tarefa 07 –** Enviar os metadados e seu dataset |
| --- |
| Você irá, então, enviar seu dataset e os metadados que foram preenchidos para que eles possam ser salvos no repositório WikiOlapBase |

| **T8: Tarefa 08 –** Verificar se seu dataset foi incluído no repositório |
| --- |
| Para garantir que o dataset foi incluído no repositório você irá utilizar a função de busca do WikiOlapBase para buscar o dataset que acabou de ser enviado. Para isso utilize o termo: cpmf. Anote o tableId pois ele será necessário posteriormente. |

| **T9: Tarefa 09 –** Utilizar a API para recuperar os dados e gerar sua visualização |
| --- |
| Você então irá utilizar a API fornecida pelo WikiOlapBase para recuperar os dados e gerar sua visualização. Para isso acesse a documentação da API (http://docs.wikiolapapi.apiary.io/). A esquerda embaixo de “Reference” estarão os métodos de acesso disponíveis, você irá utilizar o método “Recuperar Dados”. Clicando sobre ele você terá informações de como utilizá-lo, depois clique sobre “Get Data” para verificar o endpoint que deve ser utilizado. Preencha os campos necessários no script disponibilizado e gere sua visualização. |

***Obrigada por sua grande ajuda!***

**TAREFAS A SEREM EXECUTADAS**

- **Cenário 2**

Você ouviu dizer que a Contribuição Provisória sobre Movimentações Financeiras (CPMF) pode voltar, além disso também ouviu comentários que com a CPMF a inflação deve aumentar. Você então decidiu verificar se isso é verdade historicamente. Ao buscar na ferramenta WikiOlapBase percebeu que já existia um conjunto de dados referente a arrecadação com CPMF ao longo dos anos. Resolveu então enviar um conjunto de dados referente a variação do Índice Nacional de Preços ao Consumidor Amplo (IPCA), um índice que indica o aumento ou não da inflação, para a ferramenta WikiOlapBase, a partir do cruzamento entre esses dados você será capaz de verificar se o aumento da arrecadação com a CPMF tem alguma ligação com o aumento da inflação. Para isso você irá realizar as seguintes ferramentas:

| **T1: Tarefa 01** - Aprender a utilizar a ferramenta |
|---|
| Como é sua primeira vez utilizando a WikiOlapBase você deve entrar no site da ferramenta e acessar a página de instruções para aprender como utilizá-la. |

| **T2: Tarefa 02 –** Enviar seu arquivo CSV. |
|---|
| Você deve enviar o arquivo que contém os dados que você deseja integrar com o repositório do WikiOlapBase. |

| **T3: Tarefa 03 –** Ver o preview de seu dataset |
|---|
| Para garantir que você selecionou o arquivo correto você deve utilizar a função de preview da ferramenta para confirmar que o arquivo enviado foi o correto. |

| **T4: Tarefa 04 –** Preencher informações básicas do dataset |
|---|
| Você deve preencher as informações básicas do dataset referente a fonte, título e descrição, além disso deve informar um email de contato. Preencha com as seguintes informações:<br>● Título: IPCA<br>● Descrição: Série histórica da variação do IPCA de 1985 a 2016.<br>● Fonte: http://www.ipeadata.gov.br/<br>● Email: teste@teste.com.br |

| **T5: Tarefa 05 –** Definir tags para as colunas do seu dataset e renomeá-las |
|---|
| Como você deseja que outras pessoas também possam utilizar seu dataset você irá preencher tags para cada coluna de seu dataset e mudar os nomes das colunas para um nome mais agradável. Faça as seguintes alterações<br>● cd_ano -> ano<br>● cd_mes -> mes<br>E adicione as tags:<br>● ano: ano, tempo<br>● mes: mes, tempo<br>● ipca: ipca, inflação |

| **T6: Tarefa 06 –** Definir uma hierarquia para seu dataset |
| --- |
| Você deseja fazer um gráfico após terminar de enviar seu arquivo, por isso irá definir uma hierarquia em seu conjunto de dados para utilizar essa informação posteriormente. Defina a seguinte hierarquia:<br>● Nome: Tempo<br>● Hierarquia: ano -> mes |

| **T7: Tarefa 07 –** Enviar os metadados e seu dataset |
| --- |
| Você irá, então, enviar seu dataset e os metadados que foram preenchidos para que eles possam ser salvos no repositório WikiOlapBase |

| **T8: Tarefa 08 –** Verificar se seu dataset foi incluído no repositório |
| --- |
| Para garantir que o dataset foi incluído no repositório você irá utilizar a função de busca do WikiOlapBase para buscar o dataset que acabou de ser enviado. Para isso utilize o termo: ipca. Anote o tableId pois ele será necessário posteriormente. Além disso verifique o dataset referente a cpmf, para isso utilize o termo: cpmf. Anote também o tableId, pois ele será necessário posteriormente. |

| **T9: Tarefa 09 –** Utilizar a API para cruzar os dados e gerar sua visualização |
| --- |
| Você então irá utilizar a API fornecida pelo WikiOlapBase para recuperar os dados e gerar sua visualização. Para isso acesse a documentação da API (http://docs.wikiolapapi.apiary.io/). A esquerda embaixo de “Reference” estarão os métodos de acesso disponíveis, você irá utilizar o método “Cruzar dados”, clicando sobre ele você terá informações de como utilizá-lo, depois clique sobre “Join Data” para verificar o endpoint que deve ser utilizado. Você irá cruzar os dados pelas colunas mes e ano de cada dataset. |

***Obrigada por sua grande ajuda!***

**TAREFAS A SEREM EXECUTADAS**

- **Cenário 3**

Você ouviu dizer que a Contribuição Provisória sobre Movimentações Financeiras (CPMF) pode voltar, além disso também ouviu comentários que com a CPMF a inflação deve aumentar. Você então decidiu verificar se isso é verdade historicamente. Ao buscar na ferramenta WikiOlapBase percebeu que já existia dois conjuntos de dados referente a arrecadação com CPMF ao longo dos anos e dados referente a variação do IPCA. Decidiu então utilizar a API do WIkiOlapBase para cruzar esses dados e verificar a hipótese levantada a partir do cruzamento entre esses dados. Para isso você irá realizar as seguintes tarefas:

| **T1: Tarefa 01 -** Buscar as informações sobre os datasets na ferramenta |
|---|
| Antes de utilizar a API você precisa verificar os dados de cada conjunto para garantir que eles são suficientes para o que você precisa. Para isso você irá utilizar a funcionalidade de busca do WikiOlapBase. Você deverá realizar a busca para o conjunto de dados da CPMF utilizando o termo: cpmf, e para o conjunto de dados do IPCA utilizando o termo: ipca. Em ambos os casos anote os dados de tableId pois eles serão necessários posteriormente. |

| **T2: Tarefa 02 –** Utilizar a API para cruzar os dados para gerar sua visualização |
|---|
| Você então irá utilizar a API fornecida pelo WikiOlapBase para recuperar os dados e gerar sua visualização. Para isso acesse a documentação da API (http://docs.wikiolapapi.apiary.io/). A esquerda embaixo de “Reference” estarão os métodos de acesso disponíveis, você irá utilizar o método “Cruzar dados”, clicando sobre ele você terá informações de como utilizá-lo, depois clique sobre “Join Data” para verificar o endpoint que deve ser utilizado. Você irá cruzar os dados pelas colunas mes e ano de cada dataset. |

***Obrigada por sua grande ajuda!***

| **CARACTERIZAÇÃO DO USUÁRIO** | | | |
|---|---|---|---|
| **NOME:** | | | |
| **FORMAÇÃO:** | | | **IDADE**: |
| **PROFISSÃO:** | | | |
| **TEMPO DE EXPERIÊNCIA COM ANÁLISE DE DADOS, VISUALIZAÇÃO DE DADOS E INTEGRAÇÃO DE DADOS** | | | |
| **EXECUÇÃO DAS TAREFAS** | | | |
| **TAREFA** | **TAREFA CONCLUÍDA SEM ERRO** | **TAREFA CONCLUÍDA COM ERRO** | **TAREFA NÃO CONCLUÍDA** |
| **T1** | | | |
| **T2** | | | |
| **T3** | | | |
| **T4** | | | |
| **T5** | | | |
| **T6** | | | |
| **T7** | | | |
| **T8** | | | |
| **T9** | | | |
| **MEDIDA DE EFICIÊNCIA (TEMPO GASTO POR TAREFA)** | | | |
| **T1** | | | |
| **T2** | | | |
| **T3** | | | |
| **T4** | | | |
| **T5** | | | |
| **T6** | | | |
| **T7** | | | |
| **T8** | | | |
| **T9** | | | |
| **TEMPO TOTAL** | | | |
| **DIFICULDADES DE USO (RECURSOS E TAREFAS QUE GERARAM PROBLEMAS )** | | | |
| | | | |
| **DÚVIDAS DO USUÁRIO DURANTE A INSPEÇÃO** | | | |
| | | | |

**GRAU DE ADEQUAÇÃO À USABILIDADE (Avaliação Pós Teste)**

**Para cada princípio de Usabilidade, indique o grau de adequação do WikiOlapBase**

**1. Facilidade de aprendizado -** se refere ao tempo e esforço necessários para que os usuários aprendam a utilizar uma determinada porção do sistema com bom nível de competência e desempenho.

[ ] Atende complementa; [ ] Atende parcialmente; [ ] Não Atende; [ ] Não se aplica

**2. Facilidade de uso -** está relacionado não apenas com o esforço cognitivo para interagir com o sistema, mas também com a facilidade de completar a interação sem cometer erros durante este processo.

[ ] Atende complementa; [ ] Atende parcialmente; [ ] Não Atende; [ ] Não se aplica

**3. Flexível -** considera o quanto um sistema é capaz de acomodar caminhos distintos para se atingir um mesmo objetivo, apoiando assim as preferências e modo de trabalho individuais dos usuários.

[ ] Atende complementa; [ ] Atende parcialmente; [ ] Não Atende; [ ] Não se aplica

**4. Produtivo -** analisa se o sistema consegue fazer bem aquilo a que se destina, e se o usuário completa suas tarefas de forma rápida e eficaz.

[ ] Atende complementa; [ ] Atende parcialmente; [ ] Não Atende; [ ] Não se aplica

**5. Satisfatório -** enfatiza a avaliação subjetiva do sistema feita pelo usuário, incluindo suas preferências pessoais e emoções (positivas ou negativas) que possam surgir durante a interação

[ ] Atende complementa; [ ] Atende parcialmente; [ ] Não Atende; [ ] Não se aplica

**6. Útil -** relativo ao conjunto de funcionalidades oferecidas ao sistema para que os usuários realizem suas tarefas.

[ ] Atende complementa; [ ] Atende parcialmente; [ ] Não Atende; [ ] Não se aplica

**7. Seguro -** se refere ao grau de proteção de um sistema contra condições desfavoráveis ou até mesmo perigosas para os usuários, envolvendo desde aspectos de recuperação de condições de erro até impacto no seu trabalho ou sua saúde

[ ] Atende complementa; [ ] Atende parcialmente; [ ] Não Atende; [ ] Não se aplica

**8. Colaboração - Utilizar conjunto de dados já existente -** se refere ao grau com que o sistema permite que usuários utilizem conjuntos de dados que foram enviados por outros usuários.

[ ] Atende complementa; [ ] Atende parcialmente; [ ] Não Atende; [ ] Não se aplica

**9. Colaboração - Enviar um conjunto de dados para outra pessoa utilizar -** se refere ao grau com que o sistema incentiva a colaboração entre usuários, mostrando que ao enviar um conjunto de dados eles estará disponível para qualquer um utilizar.

[ ] Atende complementa; [ ] Atende parcialmente; [ ] Não Atende; [ ] Não se aplica

**10. Outras observações:**

# APÊNDICE B – Lista de Melhorias

1. Criação de um sistema de cadastro e autenticação de usuários.
2. Adicionar suporte a outros formatos de arquivos.
3. Permitir o envio de arquivos compactados.
4. Permitir o envio de múltiplos arquivos.
5. Estender a função de busca para mostrar todos metadados.
6. Incluir ícone na interface de editar nome das colunas para especificar a possibilidade de edição.
7. Estender as funcionalidades da API para permitir outras operações e aplicação de filtros.
8. Utilização de URIs para identifação das *tags* das colunas. Utilizar, por exemplo, o schema.org.