

UNIVERSIDADE DE PASSO FUNDO
INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM
COMPUTAÇÃO APLICADA

**Desenvolvimento da interface e
automatização da coleta de dados na
Open Smart City View**

Fernanda Rigo

Passo Fundo

2018

UNIVERSIDADE DE PASSO FUNDO
INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

**DESENVOLVIMENTO DA INTERFACE
E AUTOMATIZAÇÃO DA COLETA DE
DADOS NA OPEN SMART CITY VIEW**

Fernanda Rigo

Dissertação apresentada como requisito parcial
à obtenção do grau de Mestre em Computação
Aplicada na Universidade de Passo Fundo.

Orientador: Prof. Dr. Roberto dos Santos Rabello

Passo Fundo

2018

CIP – Catalogação na Publicação

R572d Rigo, Fernanda
Desenvolvimento da interface e automatização da
coleta de dados na Open Smart City View / Fernanda
Rigo. – 2018.
60 f. : il. color. ; 30 cm.

Orientador: Prof. Dr. Roberto dos Santos Rabello.
Dissertação (Mestrado em Computação Aplicada) –
Universidade de Passo Fundo, 2018.

1. Computação. 2. Web - Rastreamento. 3. Informações
eletrônicas governamentais 4. Mineração de uso da web.
5. OSCV. I. Rabello, Roberto dos Santos, orientador. II.
Título.

CDU: 004

Catalogação: Bibliotecária Jucelei Rodrigues Domingues - CRB 10/1569

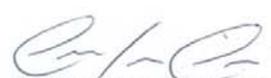
**ATA DE DEFESA DO
TRABALHO DE CONCLUSÃO DE CURSO DO ACADÊMICO**

FERNANDA RIGO

Aos dois dias do mês de março do ano de dois mil e dezoito, às 14 horas, realizou-se, no Instituto de Ciências Exatas e Geociências, prédio B5, da Universidade de Passo Fundo, a sessão pública de defesa do Trabalho de Conclusão de Curso "**Desenvolvimento da Interface e automatização da coleta de dados na Open Smart City View**", de autoria de Fernanda Rigo, acadêmica do Curso de Mestrado em Computação Aplicada do Programa de Pós-Graduação em Computação Aplicada – PPGCA/UPF. Segundo as informações prestadas pelo Conselho de Pós-Graduação e constantes nos arquivos da Secretaria do PPGCA, a aluna preencheu os requisitos necessários para submeter seu trabalho à avaliação. A banca examinadora foi composta pelos doutores Roberto dos Santos Rabello, Cristiano Roberto Cervi e Carina Friedrich Dorneles. Concluídos os trabalhos de apresentação e arguição, a banca examinadora considerou a candidata AVINDA M. Foi concedido o prazo de até quarenta e cinco (45) dias, conforme Regimento do PPGCA, para a acadêmica apresentar ao Conselho de Pós-Graduação o trabalho em sua redação definitiva, a fim de que sejam feitos os encaminhamentos necessários à emissão do Diploma de Mestre em Computação Aplicada. Para constar, foi lavrada a presente ata, que vai assinada pelos membros da banca examinadora e pela Coordenação do PPGCA.



Prof. Dr. Roberto dos Santos Rabello - UPF
Presidente da Banca Examinadora
(Orientador)



Prof. Dr. Cristiano Roberto Cervi - UPF
(Avaliador Interno)



Profa. Dra. Carine Friedrich Dorneles - UFSC
(Avaliador Externo)



Prof. Dr. Rafael Rieder
Coordenador do PPGCA

AGRADECIMENTOS

Primeiramente agradeço a Deus pela vida, pelo amparo nos momentos de dificuldade, pelos obstáculos propostos e oportunidades de crescimento, tanto como pessoa, moral e espiritual. Sou eternamente grata.

A minha família, por estar sempre ao meu lado, pelo incentivo, amor e apoio incondicional. Por aceitarem sempre minhas decisões e aventuras em busca dos meus sonhos, servindo de suporte nas horas de fraqueza. Vocês são o bem mais precioso que possuo.

Em especial, agradeço ao meu orientador Dr. Roberto dos Santos Rabello, pelos ensinamentos, paciência e pela condução das atividades durante este percurso para o findar deste trabalho, sempre motivando para a melhor realização da pesquisa. Seu apoio e incentivo foi de suma importância para a conclusão de mais esta etapa.

Ao Diego Lusa, por projetar a arquitetura OSCV e me permitir dar continuidade ao seu projeto de conclusão de curso de mestrado. Muito obrigada pelos inúmeros auxílios prestados, pelas palavras de apoio e incentivo.

A todos os docentes do PPGCA da Universidade de Passo Fundo, funcionários, colegas e amigos que estiveram comigo neste período, trocando conhecimento, experiências e nos fortalecendo a cada dia.

DESENVOLVIMENTO DA INTERFACE E AUTOMATIZAÇÃO DA COLETA DE DADOS NA OPEN SMART CITY VIEW

RESUMO

Com o aumento da população das cidades nas últimas décadas, vários são os desafios encontrados em âmbito governamental. Mas o cidadão mostra-se um elemento importante para o processo de construção de uma cidade inteligente, a partir das informações e conhecimento disponibilizados pelos dados governamentais abertos. Se estes dados forem manipulados e minerados adequadamente, servem para tornar a população mais empoderada e participativa na tomada de decisões. Assim, projetou-se e desenvolveu-se a arquitetura Open Smart City View (OSCV) como sendo um modelo tecnológico funcional de coleta, processamento e apresentação de informações relevantes, a partir de dados coletados nos portais de transparência governamental [1]. Para complementar ainda mais as funcionalidades presentes na ferramenta, desenvolveu-se um processo de *web crawler* para realizar a coleta de dados automaticamente nas fontes e portais. Com isso, mantendo sempre o *data warehouse* com informações atualizadas. Para avaliar o desenvolvimento do processo de extração, realizou-se teste funcional com auxílio do *framework* JUnit para obter a veracidade do correto funcionamento dos métodos implementados. Também, desenvolveu-se uma interface web clara e sucinta para a visualização das informações coletadas e armazenadas no *data warehouse* pela ferramenta. A interface é utilizada pelo usuário para realizar consultas e pesquisas de assuntos relacionados à sociedade, propiciando assim um melhor engajamento dos cidadãos com questões da sua comunidade. Através do método de avaliação SUS pode-se avaliar a usabilidade da interface projetada, obtendo após os testes realizados, resultado com pontuação satisfatória.

Palavras-Chave: dados governamentais abertos, *web crawler*, OSCV.

DESENVOLVIMENTO DA INTERFACE E AUTOMATIZAÇÃO DA COLETA DE DADOS NA OPEN SMART CITY VIEW

ABSTRACT

Along with the increase of cities' population during the last decades, several challenges are found in governmental scope. The citizen is an important element in the process of building an intelligent city, based on the information and knowledge provided by open governmental data. If these data are handled and mined properly, they serve to empower the population and its part in decision making. Thus, the Open Smart City View (OSCV) architecture was designed and developed as a functional technological model for collecting, processing and presenting relevant information, from data collected in government transparency portals [1]. To further complement the functionalities present in the tool, a web crawler process was developed to perform the data collection automatically in the sources and portals. With this, the data warehouse is constantly updated with information. To evaluate the development of the extraction process, a functional test was performed using the JUnit framework to obtain the correct functioning of the implemented methods. Also, a clear and concise web interface has been developed for the visualization of the information collected and stored in the data warehouse by the tool. The interface is used to carry out queries and research on issues related to society, providing a better engagement of citizens with issues of their community. Through the SUS evaluation method, it is possible to evaluate the usability of the projected interface obtaining, after the tests performed, a result with a satisfactory score.

Keywords: open government data, web crawler, OSCV.

LISTA DE FIGURAS

Figura 1.	Sinalização mundo mais urbano [5].	20
Figura 2.	Brasil mais urbano [5].	20
Figura 3.	Eixos de atuação e sistemas de uma cidade inteligente [13].	22
Figura 4.	Modelo simplificado das camadas funcionais da arquitetura OSCV [1].	26
Figura 5.	Camada Coleta e Processamento [1].	28
Figura 6.	Diagrama conceitual da Camada Web [1].	29
Figura 7.	Seção de critérios de filtragem - Interface de apresentação de informações [1].	32
Figura 8.	Seção de renderização - Interface de apresentação de informações [1].	32
Figura 9.	Camada Dispositivos Interação [1].	33
Figura 10.	Fluxo de um Crawler [23].	34
Figura 11.	Interface MuniMex 1.0 [28].	39
Figura 12.	Fluxo ilustrativo da metodologia abordada.	41
Figura 13.	Camada de Coleta e Processamento OSCV. Adaptada [1]	42
Figura 14.	Interface de listagem das configurações da extração de dados.	44
Figura 15.	Gerenciar automatização na coleta de dados.	44
Figura 16.	Sequência de comandos responsáveis pela extração de arquivos.	45
Figura 17.	Visão simplificada do algoritmo.	46
Figura 18.	Fluxo de processamento da <i>Thread</i>	46
Figura 19.	Interface de Log de processamento.	47
Figura 20.	Exemplo de e-mail de notificação.	47
Figura 21.	Resultado teste JUnit.	49
Figura 22.	Interface login.	51
Figura 23.	Menu ambiente administrativo.	52
Figura 24.	Listagem de apresentação de fato.	52
Figura 25.	Interface de visualização e edição dos fatos.	52
Figura 26.	Interface de perspectivas associadas.	53
Figura 27.	Listagem de Jobs.	53
Figura 28.	Visualização de Jobs.	54
Figura 29.	Interface <i>frontend</i>	55
Figura 30.	Interface <i>frontend</i> - pesquisa.	55
Figura 31.	Apresentação de gráficos.	56
Figura 32.	Interface <i>frontend</i> responsiva.	57

Figura 33.	Padrão Resposta SUS.	58
Figura 34.	Questionário SUS.	58
Figura 35.	Score SUS [41].	59
Figura 36.	Gráfico de resultado por questão.	60

LISTA DE SIGLAS

ABNT – Associação Brasileira de Normas Técnicas

UPF – Universidade de Passo Fundo

OSCV – Open Smart City View

UPF – Universidade de Passo Fundo

IBGE – Instituto Brasileiro de Geografia e Estatística

IBM – *International Business Machines*

TIC – Tecnologias da Informação e Comunicação

ETL – *Extract, Transform and Load*

PDI – *Pentaho Data Integrator*

ROLAP – *Relational Online Analytical Processing*

OLAP – *Online Analytical Processing*

PNG – *Portable Network Graphics*

HTTP – *Hypertext Transfer Protocol*

TCP – *Transmission Control Protocol*

IP – *Internet Protocol*

SMTP – *Simple Mail Transfer Protocol*

FTP – *File Transfer Protocol*

API – *Application Programming Interface*

IWB – *Information Workbench*

INEGI – Instituto Mexicano de Estatística e Geografia

SUS – *System Usability Scale*

JSF – *Java Server Faces*

MVC – Modelo-Visão-Controlador

CSS – *Cascading Style Sheets*

HTML – *HyperText Markup Language*

SQL – *Structured Query Language*

IMSS – Instituto Mexicano de Segurança Social

ISSSTE – Instituto de Segurança Social dos Funcionários do Governo Federal

SUMÁRIO

1	INTRODUÇÃO	17
2	REVISÃO DA LITERATURA	19
2.1	DADOS ABERTOS E PORTAIS DE TRANSPARÊNCIA	22
2.2	OPEN SMART CITY VIEW	26
2.2.1	Fontes de Dados	26
2.2.2	Coleta e Processamento	27
2.2.3	Camada Web	29
2.2.3.1	Visão <i>backend</i>	31
2.2.3.2	Visão <i>frontend</i>	31
2.2.4	Dispositivos de Interação	33
2.3	TÉCNICAS DE EXTRAÇÃO DE DADOS	33
2.4	TRABALHOS RELACIONADOS	35
3	PROCESSO DESENVOLVIMENTO DA INTERFACE E AUTOMATIZAÇÃO DA COLETA DE DADOS NA OPEN SMART CITY VIEW	41
3.1	AUTOMATIZAÇÃO DO PROCESSO DE COLETA DE DADOS	42
3.1.1	Teste do Processo	47
3.2	DESENVOLVIMENTO DA INTERFACE WEB	50
3.2.1	Visão funcional <i>backend</i>	51
3.2.2	Visão funcional <i>frontend</i>	54
3.2.3	Avaliação da interface web	56
4	ANÁLISE E DISCUSSÃO DE RESULTADOS	59
5	CONSIDERAÇÕES FINAIS	61
5.1	PUBLICAÇÕES	62
	REFERÊNCIAS	63
	APÊNDICE A – Termo de consentimento	67
	APÊNDICE B – Questionário de avaliação	69

1. INTRODUÇÃO

As cidades evoluíram e vêm evoluindo exponencialmente. Nesse sentido, estudos e pesquisas apontam dados de crescimento populacional significativo se comparado a décadas passadas, onde a maioria da população vivia na área rural. Em 1900, cerca de 200 milhões de pessoas habitavam as cidades. Já nos dias atuais, este número aumentou mais de 500%, ou seja, estima-se que as cidades estejam com mais de 3.5 bilhões de habitantes. Com o aumento populacional e o inchaço nas grandes áreas urbanas, vieram junto vários problemas a serem enfrentados em diversos setores como saúde, transporte, educação, segurança, mobilidade, entre outros. Perante este cenário, as cidades precisam adequar-se e adaptar-se para sanar suas demandas, fazendo-se uso de ferramentas e tecnologias para auxiliar na prestação de serviços e produtos mais eficientes. Em decorrência disso, surgem as cidades inteligentes, quais utilizam-se das tecnologias para a resolução dos problemas encontrados, até mesmo desafios enfrentados no dia a dia da população.

Além das tecnologias, outro fator que auxilia para a amenização e resolução dos problemas e necessidades do povo é a participação do cidadão nos assuntos e decisões de âmbito governamental. Mas para que a população se torne mais participativa, precisa se empoderar e conhecer as informações que se encontram em poder oficial, sendo este um dos benefícios e objetivos dos dados governamentais abertos.

Os dados governamentais abertos são informações geradas pelo governo, e disponíveis aos cidadãos, para que estes se beneficiem dos dados na tomada de decisões e obtenção de respostas conclusivas para os assuntos relevantes pesquisados. Os dados disponibilizados nos portais de transparência governamentais na maioria das vezes são apresentados sem mineração, e de forma que dificulta o entendimento e mensuração da informação desejada. Por este motivo há a exploração e utilização de ferramentas que auxiliam na mineração e filtragem de assuntos desejados.

Assim sendo, a arquitetura OSCV foi projetada e desenvolvida por um ex-aluno do curso de Pós-graduação em Computação Aplicada, da Universidade de Passo Fundo, Ms. Diego Lusa, para dispor de condicionamento ao cidadão de conhecer e avaliar os assuntos de seu município e localidade, através de apresentações simplificadas das informações coletadas dos dados governamentais abertos. Essa arquitetura é um modelo tecnológico funcional de coleta, processamento e apresentação de dados, que se divide em 4 camadas de operação. Cada uma delas é responsável por um processo de funcionamento. Pelo potencial enquanto ferramenta analítica, é possível alinhar algumas evoluções para o melhor cumprimento de seus objetivos quando projetado.

O desafio do trabalho proposto é desenvolver alguns aspectos considerados relevantes para a evolução da ferramenta Open Smart City View, os quais não foram abordados na primeira versão da arquitetura, versão 1.0, pois não era o foco de abordagem naquele momento. Sendo um dos aspectos a automatização no processo de coleta dos conjuntos de dados disponíveis, nos portais de transparência governamentais, o qual foi realizado de forma manual na versão 1.0. O objetivo de

tornar o processo de coleta automatizado para evitar possíveis falhas humana ao realizar a atividade de coleta, garantindo com isso dados atualizados para serem disponibilizados a população.

Outro desafio é o desenvolvimento de uma interface simples, interativa, de fácil manuseio para que o usuário possa melhor realizar suas pesquisas e visualizações, contudo, responsiva para atender às diversas formas de acesso. Assim sendo, para concretizar essa evolução, realizou-se a coleta de dados de forma automatizada nas fontes públicas com dados específicos do município de Passo Fundo, para permitir o empoderamento do cidadão através do conhecimento das informações públicas apresentadas em uma forma clara e intuitiva pela interface.

A organização do presente trabalho deu-se da seguinte forma: no capítulo 2, apresenta-se a revisão da literatura sobre a evolução urbana com o passar dos anos e a necessidade de adequação das cidades para as demandas que imergem, com uma breve explanação sobre dados governamentais abertos e portais de transparência. No mesmo capítulo, aborda-se ainda sobre a arquitetura OSCV, suas camadas e funcionalidades, além de uma abordagem sobre automação na coleta de dados através da utilização de *crawler*. Também, apresenta-se nessa mesma seção alguns trabalhos relacionados à extração e apresentação de dados públicos. No capítulo 3, descreve-se a metodologia utilizada para o desenvolvimento da automatização do processo de coleta de dados e a realização do teste funcional sobre o processo de automatização. Contudo, apresenta-se o desenvolvimento da interface *web* sobre a visão *backend* e *frontend*, e o resultado da avaliação de usabilidade aplicado sobre a interface. Por fim, explana-se a respeito dos resultados obtidos e as considerações finais sobre o trabalho elaborado.

2. REVISÃO DA LITERATURA

Com o passar dos anos, uma parte considerável dos indivíduos que viviam no campo passaram a migrar e viver nas grandes cidades. Essas pessoas almejavam novas oportunidades e conquistas que as áreas urbanas têm oferecido aos cidadãos. Entre as oportunidades estão: maior variedade de emprego, fácil acesso a saúde, educação, entretenimento e cultura. Além de muitos outros, esses são alguns dos benefícios almejados pelos indivíduos migrantes [2].

O crescimento das cidades é um processo irrefreável e irreversível. De acordo com o Programa das Nações Unidas para Assentamentos Humanos (UN-Habitat), a população urbana foi multiplicada por cinco entre 1950 e 2011 no mundo todo. Foi em 2007 que, pela primeira vez na história da humanidade, o número de pessoas vivendo em cidades ultrapassou a quantidade daquelas que se encontravam no campo [3].

O cenário atual mostra-se favorável e comprova estas mudanças. Segundo dados das Organizações das Nações Unidas, entre 2007 e 2050, o número de habitantes terá crescimento de 3,1 bilhões de pessoas, que se dispersarão por inúmeras cidades e 26 mega cidades [4]. Conforme estudos da empresa brasileira de consultoria em cenários prospectivos Macroplan Prospectiva Estratégia Gestão, há indicativo de que a urbanização e desenvolvimento estão positivamente relacionados e de que 2/3 do crescimento mundial até 2025 virá de 600 cidades. Assim, cada vez mais a competitividade de uma nação irá refletir a competitividade combinada de suas cidades [5].

Uma pesquisa socioeconômica realizada pela Macroplan, em 2013, apresentou resultados e expectativas de oito décadas, entre os anos de 1950 e 2030. Onde, no fim da oitava década pesquisada, estima-se que 5 milhões de pessoas estarão habitando as cidades. E, neste mesmo período, um pouco mais de 3 milhões será o número aproximado da população rural, conforme apresenta o Figura 1 [5].

Segundo Censo do IBGE, em 2010, a taxa de urbanização no Brasil alcançava o índice de 84%. Sendo esta taxa a percentagem de população que vivia nas cidades em relação a população total do território [5]. A Figura 2, ilustra o percentual de população habitando os centros urbanos, nas quatro regiões do Brasil, no período de 1940 a 2010. Observa-se um aumento significativo no índice em todas as regiões brasileiras, no intervalo de 1940 e 2000, se comparado com o período de 2000 a 2010.

Com a grande aglomeração de pessoas vivendo nas cidades, vários são os obstáculos e dificuldades encontradas, tanto de ordem ambiental, como social e governamental. Além disso, deve-se levar em consideração os benefícios, conhecimentos e desenvolvimentos que estas pessoas podem proporcionar, com a colaboração na disseminação de conhecimentos e experiências para ampliar e melhorar os indicadores nas cidades.

Os centros urbanos têm de estar em constante aprimoramento e evolução para dar conta das demandas e necessidades que esses habitantes proporcionarão. Com a elevada migração de habitantes para as cidades, os governos enfrentam diversos desafios a frente desse crescimento muitas

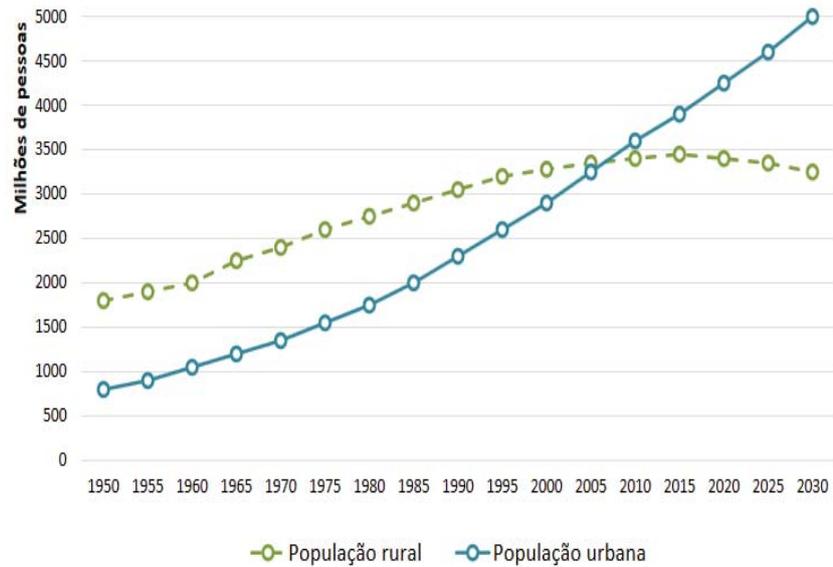


Figura 1. Sinalização mundo mais urbano [5].

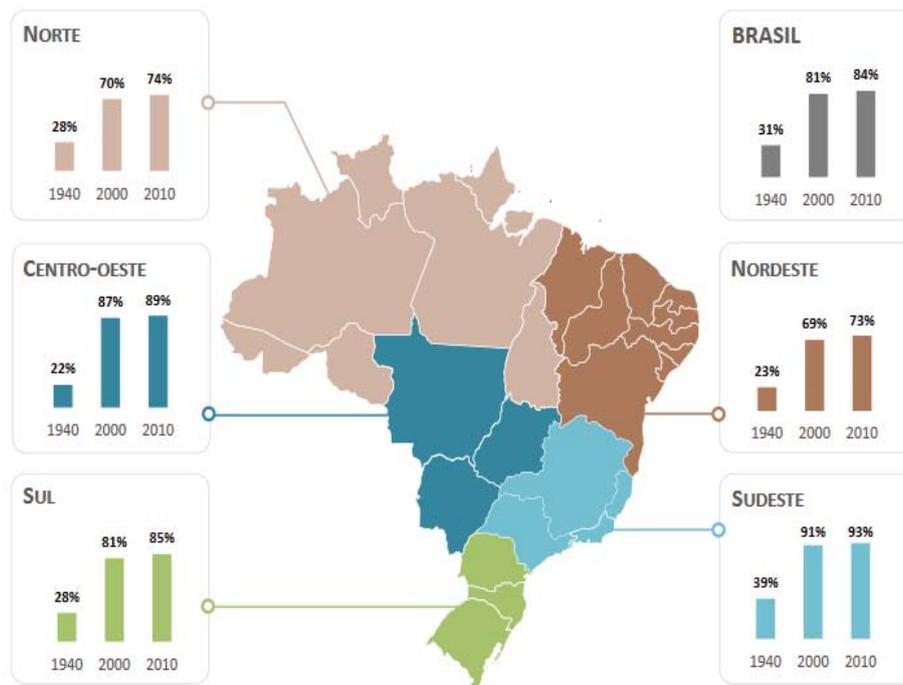


Figura 2. Brasil mais urbano [5].

vezes desordenado. Com isso, torna-se essencial para a construção de uma cidade competitiva o desenvolvimento em todos os setores: infraestrutura, tecnologias, serviços, governo, recursos naturais, entre outros [6]. Conforme dados estatísticos, o enorme crescimento populacional das grandes cidades tendem a aumentar ainda mais, de forma alarmante, nos próximos anos [7]. Quanto mais pessoas há vivendo na sociedade, cada vez mais os produtos, serviços e ferramentas terão de ser eficientes e bem preparados, proporcionando, assim, mais desafios na solução de problemas legados sem que haja prejuízos para o meio ambiente.

Por volta de 1990 pesquisadores apostavam na origem das “Cidades Digitais” como aliadas no auxílio de demandas que estavam surgindo devido ao crescimento populacional. Eram focalizadas principalmente na conectividade e no governo eletrônico local. Atualmente emerge o conceito de “Cidades Inteligentes” (*Smart Cities*), as quais se utilizam das tecnologias a serviço do ser humano e do meio ambiente [8].

Todas as cidades inteligentes são também cidades digitais, mas nem todas as cidades digitais são inteligentes. A diferença entre cidades digitais e cidades inteligentes é a capacidade de resolução de problemas com o uso de tecnologias inteligentes, sendo que a capacidade marcante das cidades digitais está na oferta de serviços através da comunicação digital. A partir dessa perspectiva, uma cidade inteligente ou um espaço inteligente se refere a um ambiente físico no qual as tecnologias de comunicação e de informação se embutem nos objetos físicos e nos ambientes nos quais vivemos, viajamos e trabalhamos [9].

Em decorrência das evoluções, surgiu a necessidade de soluções que convirjam para iniciativa de “Cidades Inteligentes”. Ou seja, aquelas que utilizam a tecnologia para oferecer recursos que podem ajudar a solucionar ou minimizar os problemas urbanos em seus domínios específicos. Cidades inteligentes [10] é um termo que se menciona em centros universitários, empresas e governos para descrição de cidades que, por um lado, são cada vez mais compostas e monitoradas por computação pervasiva e ubíqua e, por outro, na inovação, criatividade e empreendedorismo, sendo analisado por especialistas a presença na economia e governança.

A *World Foundation for Smart Communities* define que as cidades inteligentes devem ser baseadas em um crescimento inteligente e planejado por meio das TICs. Um espaço e comunidade inteligente utiliza tecnologia da informação para transformar e auxiliar no convívio com a sociedade e no trabalho de uma forma significativa e fundamental, em vez de seguir uma forma incremental [11].

Já Caragliu, menciona que uma cidade inteligente é o investimento em capital humano-social que incentiva a utilização de tecnologias avançadas de TIC como viabilizador para um crescimento econômico sustentável e uma melhora na qualidade de vida, uma boa gestão de recursos naturais e energéticos, com participação atuante do governo e estado [12].

Diversas são as expectativas para que as cidades se tornem realmente inteligentes. Muitos são os estudos em andamento a fim de avaliar as mais diferenciadas tecnologias e analisar seus resultados. A integração e planejamento conjunto dos setores envolvidos com o público privado, sociedade e instituições de pesquisa, é um diferencial para que todos possam usufruir dos benefícios e garantir um crescimento sustentável e inevitável dos grandes centros urbanos [6].

A IBM intitula-se uma das primeiras empresas a se preocupar e definir cidades inteligentes. Descreve que cidade é um seguimento interconectado de sistemas e, além disso, cidades mais inteligentes norteiam o crescimento econômico sustentável e a prosperidade para seus cidadãos. Os órgãos públicos contém as ferramentas para analisar dados, com isso melhorar decisões, prever problemas, resolvê-los de forma proativa e coordenar recursos para operar efetivamente [13]. A IBM ainda define em três eixos básicos os sistemas que fazem parte de uma cidade inteligente: planejamento e gerenciamento, infraestrutura e humano, conforme ilustrado na Figura 3.



Figura 3. Eixos de atuação e sistemas de uma cidade inteligente [13].

A participação do cidadão é fundamental para o processo de inovações tecnológicas, sociais, culturais e econômicas. As tecnologias tornam possível a construção de serviços para que o cidadão, a partir de dados que se encontram em poder público, consigam respostas para várias questões de seu interesse no cotidiano, como saber as melhores oportunidades de emprego, o caminho mais curto e mais seguro da sua casa até seu trabalho, interesses e preocupações em outras áreas como a qualidade do ar que se respira e meio ambiente [14]. Se mesclado as tecnologias e o potencial das informações em poder governamental, tem-se uma ferramenta que pode auxiliar e orientar indivíduos nessa sociedade competitiva, assim como empresas e indústrias no desenvolvimento das cidades. Os dados governamentais abertos são um canal para se utilizar dessas informações, criando-se mais valores e impactos nas diversas áreas da sociedade.

2.1 DADOS ABERTOS E PORTAIS DE TRANSPARÊNCIA

Dados governamentais abertos são dados produzidos pelo governo e colocados à disposição das pessoas de forma a tornar possível, além da leitura e acompanhamento, também a reutilização em projetos e aplicativos. No cruzamento das informações com outros dados de diferentes fontes gera-se potenciais visualizações [14].

Os dados são considerados abertos quando qualquer pessoa pode utilizar, reutilizar e redistribuir livremente, sujeito apenas a exigência de creditar a sua autoria e compartilhar pela mesma licença [2]. A definição e significado dos termos utilizados se resumem em alguns pontos [14]:

- Disponibilidade e acesso: o dado precisa estar disponível por inteiro e por um custo razoável de reprodução, preferencialmente por meio de *download* na internet, além da obrigatoriedade de estar em um formato conveniente e modificável.
- Reuso e redistribuição: o dado precisa ser fornecido em condições que permitam reutilização e redistribuição, além de permitir a reprodução, ou seja, o cruzamento com outros conjuntos de dados.
- Participação universal: todas as pessoas interessadas podem usar, reutilizar e redistribuir, não havendo discriminação contra áreas de atuação, pessoas ou grupos. Não permitidas restrições como “não comercial”, que impedem o uso comercial e restrições de uso para certos fins, como “somente educacional”.

Os dados, ao serem publicados desta maneira, se tornam fonte de subsídio para pesquisadores, empresas, comunidade de Tecnologia da Informação e gestores públicos. A disponibilização dos dados de maneira aberta está prevista na Lei nº 12.527, de 18 de novembro de 2011, que regula o Acesso a Informação [15].

Dados abertos, especialmente os governamentais, são um excelente recurso ainda muito pouco explorado. Muitos indivíduos e organizações coletam uma ampla gama de dados de diferentes tipos para executar suas tarefas. O governo é particularmente importante nesse contexto, sendo por causa da quantidade e da centralidade dos dados que o possui em seu poder, a partir das coletas, quanto pelo fato de que tais dados são públicos, um direito garantido no artigo 5º da Constituição Federal Brasileira [14].

Em 2007, um grupo de especialistas denominado OpenGovData desenvolveu os oito princípios dos dados governamentais abertos [16]. Estes princípios estabelecem que os dados governamentais abertos devem ser:

- Completos: todos os dados públicos deverão ser disponíveis, onde o dado não está sujeito a limitações válidas de privacidade, segurança ou controle de acesso;
- Primários: os dados são apresentados tais como os coletados na fonte, com o maior nível possível de granularidade e sem agregação ou modificação;
- Atuais: os dados são disponibilizados tão rapidamente quanto necessário à preservação do seu valor;
- Acessíveis: os dados são disponibilizados para o maior alcance possível de usuários e para o maior conjunto possível de finalidades;
- Compreensíveis por máquinas: os dados são razoavelmente estruturados de modo a possibilitar processamento automatizado;
- Não discriminatórios: os dados são disponíveis para todos, sem exigência de requerimento ou cadastro;

- Não proprietários: Os dados são disponíveis em formato que nenhuma entidade detenha controle exclusivo;
- Livres de licenças: os dados não estão sujeitos a nenhuma restrição de direito autoral, patente, propriedade intelectual ou segredo industrial. Restrições sensatas relacionadas à privacidade, segurança e privilégios de acesso são permitidas.

Já em 2009, Eaves [17] apresentou as três leis dos dados governamentais abertos:

- Se o dado não for encontrado e indexado na web, ele não existe;
- Se não estiver aberto e disponível em formato compreensível por máquina, ele não pode ser aproveitado;
- Se algum dispositivo legal não permitir sua replicação, ele é inútil.

Há várias áreas de atuação para as quais são importantes dados abertos, e também vários grupos de organizações e indivíduos que podem se beneficiar dessa disponibilidade, inclusive o próprio governo. É possível notar um grande número de atividades e áreas que os dados abertos são valáveis [14]. Entre essas áreas estão:

- Transparência e controle democrático;
- Participação popular;
- Empoderamento dos cidadãos;
- Melhores ou novos produtos e serviços privados;
- Inovação;
- Melhora na eficiência de serviços governamentais;
- Melhora na efetividade de serviços governamentais;
- Medição do impacto das políticas;
- Conhecimento novo a partir da combinação de fontes de dados e padrões.

A área de transparência, segundo Vaz [18], é uma contribuição dos dados governamentais abertos, criando melhores possibilidades de controle social das ações governamentais. Os dados abertos como agente de transparência se devem às possibilidades de tornar os dados governamentais acessíveis a todos, eliminando as restrições referentes à tecnologia, legislação e acessibilidade, para garantir acesso sem restrição e utilização dos dados públicos pelos cidadãos.

Já transparência para Angélico [19], é a característica de governos, empresas, organizações e indivíduos em serem abertos em relação a informações sobre planos, regras, processos e ações.

Também ressalta a transparência como um meio, onde o grau de utilidade da transparência depende da sua contribuição para um sistema de prestação de contas efetivo, ou seja, está ligado à preocupação de disponibilizar informações que sejam úteis e que possibilitem inferências. Assim, a preocupação principal não deve ser somente a publicação de informações, mas também a veracidade dos dados para que o usuário possa fazer uso destas para a obtenção de conclusões dos assuntos desejados.

Para que as informações sejam disponibilizadas aos cidadãos, existe um canal onde pode-se adquirir e acompanhar os dados disponíveis pelos órgãos públicos. Esse canal é denominado Portal de Transparência. De acordo com a Controladoria Geral da União, o objetivo do portal é aumentar a transparência da gestão pública, permitindo que o cidadão acompanhe como está sendo conduzido a administração e governância do espaço onde habita, interagindo e fiscalizando [20].

A abertura dos dados governamentais à população em geral, beneficia e auxilia também ao empoderamento do cidadão para a construção de cidades mais inteligentes e participativas. A partir dos conhecimentos impulsionados ao cidadão, o processo de construção e inovação do ambiente urbano se alavanca fortemente, de forma ativa e colaborativa para a melhoria da cidade [21].

Inicia-se assim um método de fornecer conhecimento em conjunto para a população, a partir do momento que os dados governamentais passam a ser dispostos à sociedade. Onde novas ideias, técnicas e práticas elaboradas são melhoradas e o cidadão é o beneficiado com os resultados gerados. Além disso, as autoridades governamentais ao adquirirem uma abordagem mais centrada no cidadão, auxiliam a permanecerem as atividades governamentais de acordo com os interesses da população, tornando-se algo fundamental a qualquer cidade inteligente [21].

Dados governamentais abertos possui relação com o papel do cidadão inteligente no ambiente urbano, pois um cidadão que procura produzir e utilizar as informações para conhecimento do espaço onde convive, propondo soluções criativas e inovadoras para sua cidade [22]. O papel do cidadão inteligente é estar sempre próximo do poder público, monitorando e obtendo conhecimento do seu direito nos processos decisórios.

Para que o cidadão se torne cada vez mais influente nos processos decisórios dos órgãos públicos, alguns fatores precisam ser revistos. Pois a gama de informações e dados que são disponibilizados pelo governo, através dos portais de transparência, são vários e abrangem diversas áreas. Esses conteúdos podem ser utilizados no processo de melhoria dos serviços, na infraestrutura das cidades e capacitando o cidadão em assuntos referentes à sociedade. Porém, estes dados são disponibilizados de um modo que dificulta a interpretação das informações para assuntos de interesse consultados, muitas vezes pela forma com que os dados são apresentados. Como o volume de informação no poder do órgão público é elevada, e a forma a ser disponibilizada para o cidadão não estar minerada, dificulta o acesso e a compreensão dos dados, tornando-se assim irrelevante e inacessível para a sociedade. Por este motivo há exploração de ferramentas para auxiliar na filtragem e visualização dos dados abertos.

2.2 OPEN SMART CITY VIEW

A arquitetura Open Smart City View (OSCV), foi projetada para ser um modelo tecnológico funcional de coleta, processamento e apresentação de informações relevantes acerca de um determinado município. Os dados de entrada da OSCV são única e exclusivamente conjunto de dados governamentais abertos, coletados de diversas fontes, segundo diretrizes que atendam as necessidades de informação elencadas [1].

A arquitetura OSCV divide-se em quatro camadas funcionais de operação, sendo: Fontes de Dados, Coleta e Processamento, Web e Dispositivos de Interação (Figura 4).

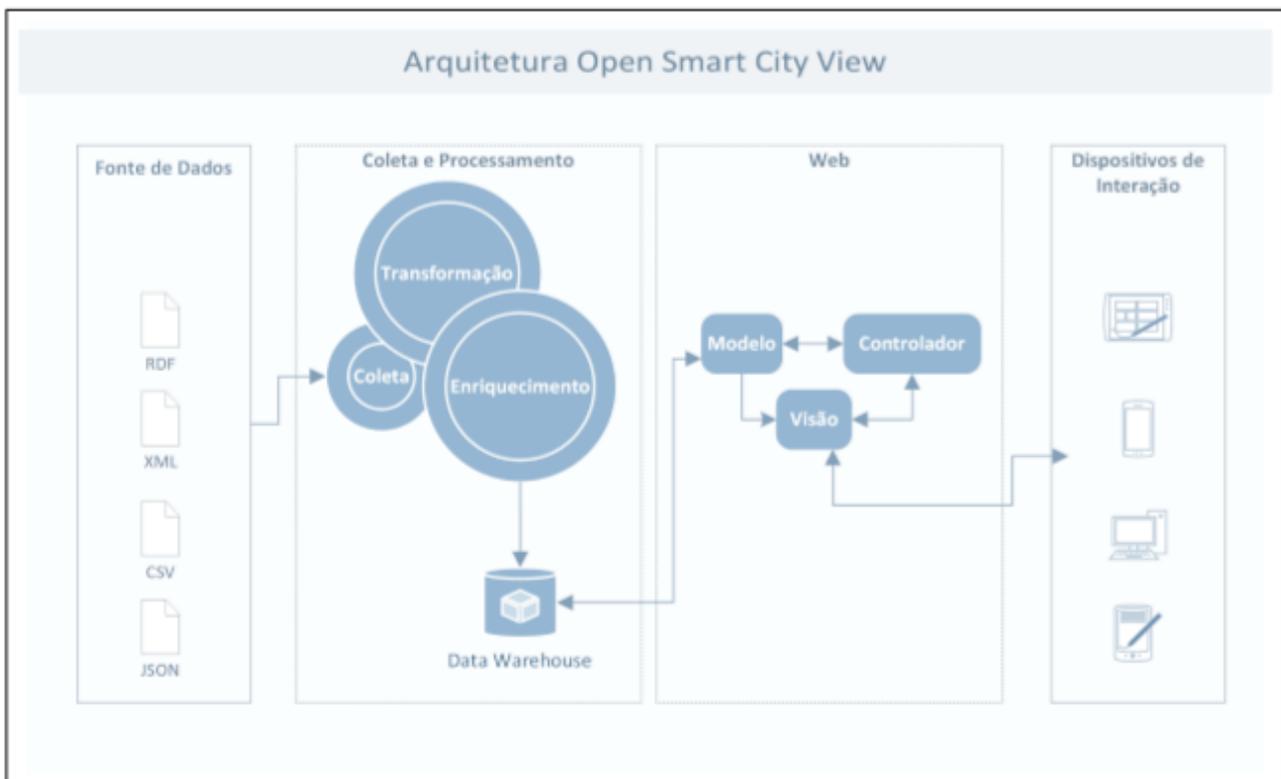


Figura 4. Modelo simplificado das camadas funcionais da arquitetura OSCV [1].

A seguir será melhor explanado a respeito da função de cada camada, segundo [1]:

2.2.1 Fontes de Dados

A camada de Fontes de Dados é a primeira da arquitetura OSCV, com função de identificar e mapear as fontes de dados governamentais abertos de interesse. Esses dados devem ser exclusivamente arquivos legíveis por máquina. Na versão 1.0 da ferramenta a captura desses documentos era realizada de forma manual, tornando-se de inteira responsabilidade do administrador da plataforma essa atividade de coleta.

Para coletar os dados de uma fonte específica, é preciso realizar um processo de análise prévia, no qual deve-se identificar as fontes e os conjuntos de dados que melhor atendem as necessi-

dades das informações desejadas. Além disso, é preciso analisar a estrutura dos conjuntos de dados, identificando as variáveis e observações contidas, o nível de granularidade dos registros e a periodicidade da atualização dos mesmos [1]. O conceito de granularidade se refere ao nível de detalhamento de um fato específico no *data warehouse*.

Arquivos com granularidade baixa, ou seja, com alto grau de detalhamento das observações disponíveis, e com alta frequência de atualização, por exemplo, são mais adequados do que arquivos com baixa frequência de atualização e granularidade alta. Níveis mais altos de granularidade conflitam com a premissa básica da arquitetura proposta, na qual se realiza a coleta e extração de conjuntos de dados em nível municipal [1].

Para a integração dos dados governamentais abertos com a camada de Coleta e Processamento requer-se um processo específico, pois cada conjunto de arquivos é composto com estrutura e metadados particulares, que precisam ser processados exclusivamente, identificando e documentando as características estruturais, nos scripts de integração [1].

Após o decreto da Lei de Transparência, muitas entidades públicas tornaram-se obrigadas a disponibilizar em seus portais institucionais dados referentes as atividades realizadas. Por não seguir um protocolo de disponibilização dos dados, um único dado pode ser encontrado em diferentes granularidades e portais governamentais diferentes ao mesmo tempo [1].

Pela vasta opção de fontes e conjuntos de dados disponíveis, o pesquisador e desenvolvedor da arquitetura OSCV [1], optou por escolher um subconjunto de fontes de dados para realizar testes e avaliação da estrutura funcional da arquitetura em seus aspectos internos. O subconjunto utilizado para o experimento se delimitou em algumas categorias distintas, as quais conduziram as atividades.

A integração dos dados elencados foi realizada a partir de scripts ETL (*Extract, Transform and Load*) desenvolvidos especificamente para o consumo. Estes scripts, assim como a interface de carga, são responsabilidade da camada Coleta e Processamento, que será descrita a seguir.

2.2.2 Coleta e Processamento

Nesta camada, os conjuntos de dados são coletados de seus locais de origem passando por um pipeline de operações para extração, transformação e carga dos registros no *data warehouse* da arquitetura, realizando um processo conhecido como ETL (*Extract, Transform and Load*). O diagrama da camada de Coleta e Processamento está ilustrada na Figura 5.

A tipificação dos arquivos, disponibilizados publicamente através dos portais governamentais, devem ser compreensíveis por máquinas. Já o formato dos arquivos são encontrados de vários tipos, sendo XML, HTML, JSON, CSV, RDF, XLS, TXT, os quais terão de ser tratados especificamente de acordo com suas características estruturais pelos scripts ETL.

A camada Coleta e Processamento, segundo a arquitetura OSCV, tem seu principal objetivo alimentar o *data warehouse*, inserindo novos registros nas tabelas fato (tabela que armazena os valores detalhados de medidas, ou fatos) e dimensão (contém o nome específico de cada membro da

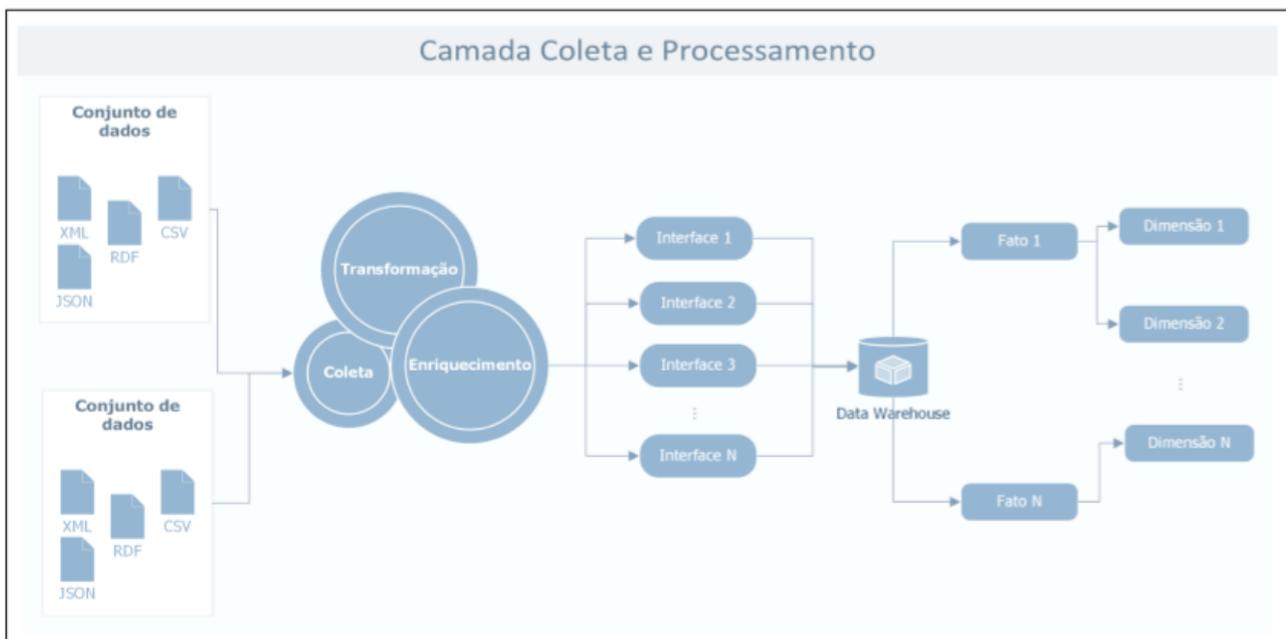


Figura 5. Camada Coleta e Processamento [1].

dimensão, ou seja, atributo). Um *data warehouse* é definido por armazenar os dados em um formato multidimensional, sendo essa modelagem de inúmeras dimensões para a compreensão estruturada dos dados denominados fatos e dimensões, que compõem a arquitetura fundamental de armazenamento de dados além das ligações lógicas criadas entre os dados, sendo denominadas tabelas fato e tabelas dimensão. Por meio da relação entre fatos e dimensões torna-se possível manipular os dados através de formatos abstratos chamadas cubos (até três dimensões) ou hipercubos (acima de três dimensões) [1]. Sendo assim, os scripts ETL realizam a leitura dos conjuntos de dados abertos das fontes de interesse em seu estado bruto, executam atividades de saneamento e transformação e, por fim, armazenam os dados resultantes nas tabelas de interface.

Para processamento dos arquivos da OSCV utilizou-se a ferramenta *Pentaho Data Integrator* (PDI), em sua versão *community*, também conhecido como Kettle. Através do desenvolvimento de scripts, os arquivos são processados utilizando representações gráficas de componentes, os quais, quando postos em conjunto e sob determinada ordem, executam as atividades de ETL definidas pelo desenvolvedor [1].

No contexto de script ETL, os componentes classificam-se em quatro grupos fundamentais: *transformations*, *jobs*, *hops* e *steps*. As *transformation* são os elementos que atuam diretamente no processamento dos dados, sendo por meio delas que o processo de ETL é realmente realizado. Uma *transformation* é composta por um ou mais *steps*, cada um executando uma parte específica da tarefa de ETL. Os *hops* ligam os diferentes *steps* que compõem uma *transformation*, os quais atuam como túnel por onde os dados trafegam de um *step* para outro no fluxo de ETL [1].

Para os scripts da arquitetura OSCV definiu-se uma estrutura baseada em tabelas de interface e *stored procedures*, com intuito de evitar erros de carga no *data warehouse* da arquitetura, e de modo a garantir a qualidade dos dados armazenados. Outra importante responsabilidade da es-

truturação utilizada refere-se à simplificação do processo de construção dos scripts ETL, que possui funcionalidade de reduzir o tempo necessário para desenvolver uma carga específica de dados. O *data warehouse* da arquitetura OSCV caracteriza-se estruturalmente por ser do tipo ROLAP, ou seja, está implementado sobre um banco de dados relacional, mantido pelo Sistema Gerenciador de Banco de Dados PostgreSQL4. A medida que novas demandas de informação forem surgindo, ou mesmo, novas fontes de interesse forem sendo adicionadas, existe a probabilidade de acréscimo de tabelas fato ou mesmo de tabelas dimensão no *data warehouse* [1].

O correto funcionamento da camada de Coleta e Processamento mostra-se importante no contexto da arquitetura OSCV, pois seus processos alimentam a arquitetura como um todo, caso contrário, a próxima camada designada Web não apresentaria condições de fornecer ao cidadão qualquer informação de valor.

2.2.3 Camada Web

A Camada Web tem a função de fornecer serviços de acesso e consumo para os dados armazenados, lhe conferindo características de um *software* OLAP (*Online Analytical Processing*). Na arquitetura OSCV, essa camada situa-se como uma interface analítica entre os dados mantidos no *data warehouse* e os diversos dispositivos de interação que se alimentarão dessas informações. A Figura 6 ilustra conceitualmente a Camada Web.

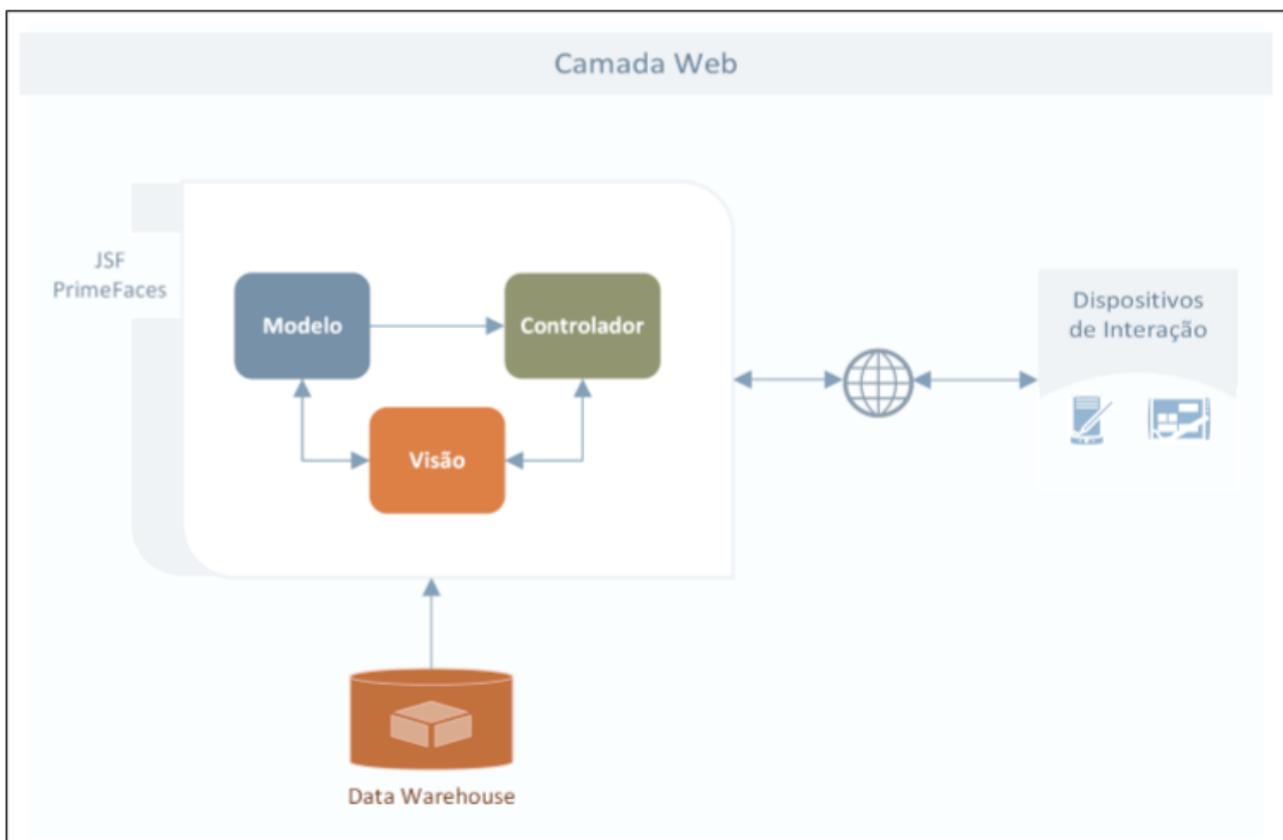


Figura 6. Diagrama conceitual da Camada Web [1].

Alguns dos desafios da arquitetura OSCV é dispor de informações sem necessitar do conhecimento prévio do real funcionamento da plataforma, além de oferecer ao cidadão informações, as quais julgue relevante, de uma forma mais adequada para compreensão e simplicidade na apresentação. Interfaces funcionais e objetivas têm maior probabilidade de alcançar sua finalidade se comparadas à interfaces poluídas e superdimensionadas. Almejar pelo simples mostra-se a melhor escolha, pois há uma grande variabilidade de usuários interagindo com a arquitetura OSCV [1].

Deste modo, prezando pela simplicidade, o autor [1] optou por organizar as informações evidenciadas ao usuário em três diferentes conceitos, que são:

- Apresentação de Fatos: são informações de natureza dinâmica na arquitetura OSCV, na qual refere-se ao grupo de perspectivas que fazem uso de uma tabela fato em particular para apresentar ao cidadão informações relevantes. O administrador do sistema possui o controle das informações de apresentação de fatos, com permissão de adicionar novos, alterar, ou mesmo remover os desnecessários.
- Perspectivas: concebe-se como uma observação particular feita sobre informações de uma tabela fato. Refere-se a questão a ser respondida através de consulta ao *data warehouse* e possui relação com as apresentações de fato na proporção de 1 para N, ou seja, uma apresentação de fatos deve ter uma ou muitas perspectivas associadas. Assim como as apresentações de fato, as perspectivas também são mantidas pelo administrador do sistema.
- Perspectivas relacionadas: proporcionam a conexão de informações na arquitetura OSCV. Diferentes fatos podem ter uma relação implícita estabelecida, e somente torna-se visível quando as observações de ambos os fatos são analisadas de forma conjunta. Há alguns critérios a obedecer, como as perspectivas relacionadas não são obrigatórias, assim como podem expor uma correlação real entre as informações ou uma simples casualidade, no qual o usuário é quem realiza a compreensão. Além disso, uma perspectiva pode possuir no mínimo 0 (zero) e no máximo N perspectivas relacionadas.

Os conceitos descritos são de conhecimento exclusivo do administrador da plataforma, o qual possui compreensão mais detalhado das relações. O usuário final, como consumidor de informações, não tem necessidade qualquer de conhecer os conceitos para interagir com a camada Web. Isso é possível devido a separação da Camada Web em duas visões funcionais separadas, projetadas para os diferentes tipos de usuário da plataforma. A visão funcional designada *frontend*, integra as interfaces de acesso público, desenvolvidas e planejadas especificamente para interação com o usuário final. Já a visão funcional de *backend* contempla o conjunto de interfaces projetadas também com especificidade para interação com os administradores da plataforma. Sendo assim, as visões de *frontend* e *backend* possuem cada uma suas próprias características e funcionam dentro de um mesmo contexto, de forma complementar. O que as tornam diferentes é a abstração de conceitos técnicos. Enquanto na visão de *frontend* eles são encapsulados em sua totalidade, na visão de *backend* se fazem presentes, visto que deles depende o bom funcionamento da plataforma como um todo [1].

2.2.3.1 Visão *backend*

A visão *backend* é imperceptível ao cidadão, onde a correta configuração dos metadados permite ao cidadão interagir e visualizar com os dados armazenados no *data warehouse*. Já os conceitos de apresentação de fato, perspectivas e perspectivas relacionadas precisam ser de conhecimento do administrador da arquitetura. Contudo, a configuração incorreta impede o cidadão de visualizar os dados de interesse. Pela importância de conter dados administrativos, têm acesso restringido por meio de autenticação via usuário e senha [1].

A configuração dos metadados inicia pelas apresentações dos fatos, conforme conceito de interação da OSCV, logo após a configuração das perspectivas associadas a apresentação. Por último, e opcional, as perspectivas relacionadas a perspectiva principal.

Ressaltando a similaridade dos atributos tanto para perspectivas quanto perspectivas relacionadas, onde os conceitos são implementados pela mesma classe. Além disso, a característica dos atributos de informação de perspectivas possui relação com as parametrizações de geração de gráficos de plano cartesiano. Portanto, ao especificar os eixos X e Y, está sendo feita referência respectivamente ao eixo de abscissas e eixo de ordenadas [1].

2.2.3.2 Visão *frontend*

Já a visão *frontend* se responsabiliza pela interação com o usuário final, provendo informações por meio da apresentação gráfica dos dados armazenados no *data warehouse*, no qual o usuário obtém os resultados através de critérios de filtragem. Na visão *frontend*, a apresentação das informações mostra-se importante, pois é por meio dela que o cidadão pode interagir com os dados armazenados na arquitetura, através das visualizações geradas a partir das buscas realizadas. Esta interface apresenta-se dividida em duas seções distintas, sendo a seção superior a qual contém os critérios de filtragem, utilizada especificamente pelo usuário para determinar as características de sua busca e a seção central, que corresponde a área de renderização e ilustração dos resultados da pesquisa do usuário [1].

A Figura 7 ilustra a seção superior, na qual as perspectivas são exibidas ao usuário em formato de árvore, sendo o primeiro nível reservado para as perspectivas principais e o segundo para as relacionadas. Neste menu, a escolha de uma perspectiva relacionada impõe, de modo automático, a escolha da perspectiva principal correspondente. Já o oposto não é verdadeiro. Ou seja, a escolha de uma perspectiva principal não obriga a escolha de nenhuma relacionada [1].

Após a realização do preenchimento dos campos de pesquisa e a solicitação do processamento da mesma, uma sequência de operações é executada internamente para construir a visualização gráfica dos dados. Inicialmente valida-se os dados de entrada do usuário. Após, cada uma das perspectivas selecionadas é convertida em uma representação gráfica correspondente, conforme parametrização informada pelo administrador da arquitetura na visão de *backend*. A Figura 8 ilustra a representação gráfica referente a pesquisa.

Escolha os critérios de sua pesquisa

Estado: RIO GRANDE DO SUL

Cidade: PASSO FUNDO

Intervalo de tempo: 2005 - 2015

Assunto: Segurança

Visualizações disponíveis:

- Homicídios dolosos
- Evolução da renda per capita
- Evolução dos latrocínios
- Violência contra a mulher
- Relação de tipos de crime

Apresentar

Figura 7. Seção de critérios de filtragem - Interface de apresentação de informações [1].

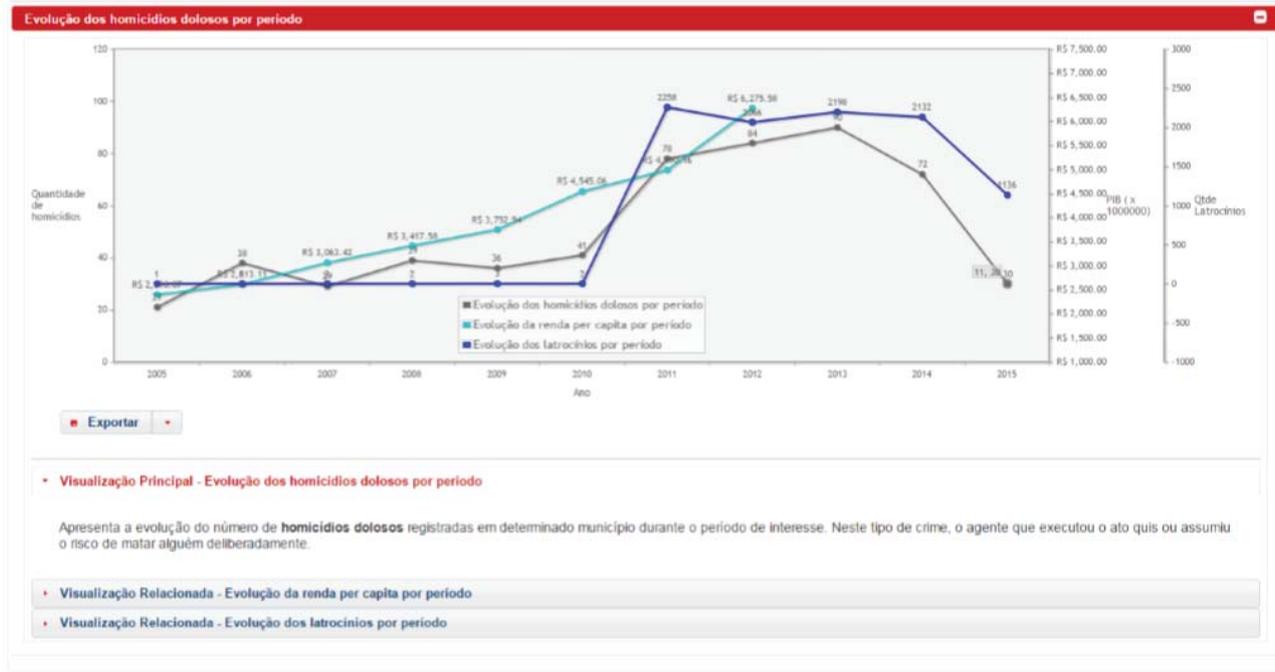


Figura 8. Seção de renderização - Interface de apresentação de informações [1].

Todo gráfico apresentado acompanha um conjunto de informações auxiliares para facilitar sua interpretação. Em formato de título, consta o nome da perspectiva principal, já na parte inferior encontra-se a descrição de cada uma das perspectivas utilizadas para compor a visualização. Além disso, contem o botão "Exportar", através do qual o usuário tem disponível a opção de exportar o gráfico apresentado em tela em formato PNG, com possibilidades também de salvar em arquivo, nos formatos XML e CSV, os dados que foram utilizados na composição das séries do gráfico. O usuário tem a possibilidade de escolher uma ou mais perspectivas acerca de um determinado assunto e a seção de renderização poderá conter inúmeras visualizações, sempre o mesmo número de perspectivas principais selecionadas [1].

As visões de *frontend* e *backend* foram elaboradas para estarem disponíveis para acesso através da Internet, onde qualquer dispositivo dispondo de um *Web Browser* adequadamente instalado, encontra-se capaz de acessá-las. Com amplo universo de dispositivos, a camada de Dispositivos de Interação é responsável por tratá-los.

2.2.4 Dispositivos de Interação

O cidadão inteligente é reconhecido por sua capacidade de conectar-se com a cidade, interagindo por meio dos diversos canais de comunicação disponíveis. Sendo assim, um dos objetivos da arquitetura Open Smart City View é se tornar uma importante ferramenta de empoderamento do cidadão. Com isso, sua amplitude de cobertura estende-se a todos os canais de comunicação tecnologicamente disponíveis no contexto urbano [1].

É preciso considerar que um extenso universo de dispositivos tecnológicos poderá vir a interagir diretamente com a arquitetura. Embora inicialmente restrita a dispositivos que dispõem de *Web Browser* instalado, espera-se que à medida que a ferramenta evoluir, outros dispositivos possam comunicar-se com a arquitetura. Figura 9 apresenta a Camada Dispositivos Interação [1].



Figura 9. Camada Dispositivos Interação [1].

O cidadão é o eixo fundamental para a camada de Dispositivos de Interação. Assim, essa camada tem o objetivo de mapear os diferentes dispositivos que o cidadão pode ou poderá utilizar para usufruir das informações disponíveis na OSCV. Outra importante função desempenhada pela camada Dispositivos de Interação diz respeito às interfaces de integração com outros sistemas, onde os mecanismos de integração permitem que outros *softwares* possam fazer uso dos dados armazenados no seu *data warehouse* [1].

2.3 TÉCNICAS DE EXTRAÇÃO DE DADOS

A automatização na coleta de dados é uma das atividades desenvolvidas para a evolução da ferramenta OSCV. Esse processo necessita de um método automatizado de indexação de conteúdo, pois os dados governamentais abertos apresentam um grande volume de dados disponíveis, e por

este motivo analisou-se o processo de automatização na coleta com o auxílio de ferramentas específicas para um correto funcionamento. Os *web crawlers*, são técnicas bastante utilizadas no setor de automação e indexação de dados, pois realizam tarefas de agentes de *softwares* utilizados para especializar pesquisas, capturas e extração de dados. O processo que um *web crawler* executa é conhecido como *web crawling* ou *spidering*. Estes dados geralmente são trafegados em protocolos HTTP acompanhados com o modelo TCP/IP, mas não se restringe especificamente a este protocolo. Pode-se trabalhar também com SMTP, FTP, entre outros. A maioria dos *robots* desenvolvidos são voltados para a web tradicional, onde documentos HTML conectados por *hyperlinks*, tags e palavras-chave são usados para rastrear e atender necessidades específicas, como download de imagens e arquivos, coleta de conteúdos específicos, e também pode ser desenvolvido para agregar dados de diversas fontes da internet, podendo persistir essas informações em banco de dados [23].

Muitos sites motores de buscas utilizam os *crawlers* para manter suas bases de dados atualizadas. Além disso, são utilizados para tarefas de manutenção automatizada de portais e sites, validação de código HTML e mineração de informações. A estrutura básica de funcionamento de um *crawler* contém fluxos que são formados em torno de uma requisição de conteúdo, extração da informação, processo de indexação e persistência, como ilustrado na Figura 10 [23].

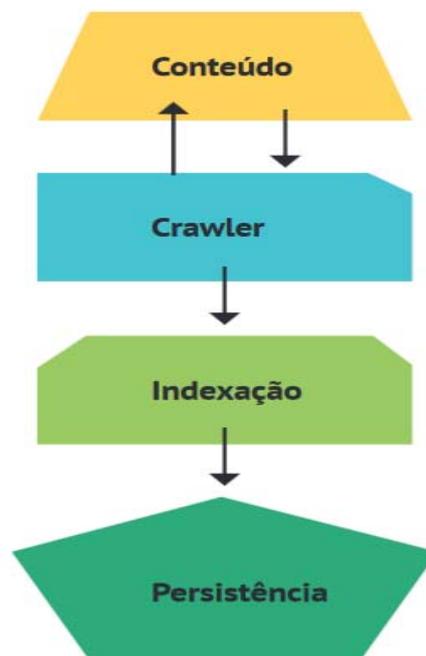


Figura 10. Fluxo de um Crawler [23].

O fluxo de requisição e extração fazem parte de qualquer processo de *crawling*, independente do objetivo e tecnologia utilizada. A requisição, como já mencionado, pode atuar em diversos tipos de fontes, independente do protocolo utilizado. Ao realizar a requisição à fonte de dados obtêm-se o conteúdo desejado, o qual será utilizado para a extração das informações buscadas. Um método bastante utilizado na extração de dados é a aplicação de expressões regulares. Sendo esta aplicada corretamente, a sua eficiência passa a ser determinante no processo do *crawler*. Além de tornar fle-

xível a localização de sequências de caracteres e conjuntos de informações, as expressões regulares podem ser exploradas para a validação dos dados [23].

Para um melhor funcionamento do *crawler*, alguns requisitos devem ser levados em consideração [24], como:

- Flexibilidade: possibilidade de utilização e reutilização de um mesmo *crawler* em vários cenários;
- Baixo custo e alto desempenho: o sistema deve realizar várias consultas e escalas a páginas por segundo e essa execução com um baixo custo computacional, pois o uso eficiente de acesso a *hardware* mantém um alto desempenho do sistema;
- Robustez: o sistema deve ser capaz de identificar e ignorar processos com falhas, como *crawlers* são utilizados para volumes grandes de consultas e informações, mesmo identificando problema não pode perder as informações já identificadas, nesse caso apenas ignorar as que estão com falha;
- Etiqueta e Controle de Velocidade: é importante seguir um padrão para o fornecimento dos contatos para os *crawlers* e seguir um método de supervisionamento das atividades de rastreamento, além de permitir o controle da velocidade de acesso de maneira distinta;
- Gerenciamento e reconfigurabilidade: necessidade de uma interface apropriada para realizar os monitoramentos de velocidade, estatísticas e tamanho dos principais conjuntos de dados. Além de monitorar, o administrador deve ter permissão para ajustes na velocidade, controle dos componentes e do sistema.

Os *web crawlers* podem ser desenvolvidos em qualquer linguagem de programação, no qual pode-se usufruir de bibliotecas e APIs *open source* para realizar a extração, comunicação e conexão com páginas web necessárias para a coleta dos dados, como Crawler4j, HtmlUnit e JSoup entre outros. Essas APIs permitem a execução de comandos e ações nos sites, da mesma forma que o usuário realizaria no navegador como, por exemplo, invocando uma página web, clicando em botões, links entre outras ações. A HtmlUnit é utilizada para modelar documento HTML, fornecendo bibliotecas responsáveis pela chamada de páginas, com suporte JavaScript e possibilidade de trabalhar com componentes AJAX complexas, conhecida como navegador sem interface para programas desenvolvidos em java [25].

2.4 TRABALHOS RELACIONADOS

Os trabalhos relacionados neste capítulo, seguiram a busca e seleção dos estudos relevantes no resultado da pesquisa sistemática, elaborada em uma estratégia de palavras-chave pertinentes ao contexto. As palavras-chave foram agrupadas seguindo a lógica de busca: (A1 OR A2 OR A3) AND (B1 OR B2), onde a Tabela1 mostra os termos de pesquisa que fazem parte da busca.

Tabela 1. Argumentos da pesquisa revisão sistemática

A1. Extraction	B1. Open government data
A2. Collect	B2. Transparency portals
A3. Interface	

As pesquisas foram realizadas nas bases: Springer, ACM, ScienceDirect, considerando buscas específicas a título, resumo e palavras-chave, seguindo a funcionalidade de cada indexador. Além disso, foram catalogados os artigos escritos em português e inglês, com ano de publicação a partir de 2010.

Através da pesquisa elaborada, observou-se na literatura vigente vários estudos referentes à disponibilização dos dados governamentais, assim como a forma com que os dados são apresentados para conhecimento da população. Segundo Böhm, a quantidade de dados disponibilizados na web pelos órgãos públicos aumentou consideravelmente nos últimos anos. Junto com isso, a qualidade, integridade e consistência dos dados passam a ser um fator que difere amplamente. Indivíduos tem dificuldade de minerar os dados relevantes a serem abordados, pois as informações são disponibilizadas em formatos de arquivos diversificados. Com objetivo de fornecer dados estruturados e recursos unificados, foi criado o GovWILD (*Government Web Data Integration for Linked Data*) no qual integra e interconecta dados governamentais, criando links entre os conjuntos de dados existentes, bem como fornecendo um aplicativo web que permite explorar os dados resultantes [26].

Os dados de integração do GovWILD são específicos dados financeiros e de fundos públicos, em geral dos Estados Unidos e União Européia, onde através de funcionalidades de clusterização e scripts de execução que foram realizados os processos de integração. Para a integração, os dados foram submetidos a 5 etapas:

- Preparação: inicialmente foram selecionadas e identificadas as fontes de dados relevantes;
- Mapeamento: desenvolvido os scripts de depuração específicos da fonte e superar a heterogeneidade dos esquemas. O esquema é simples e, portanto, fácil de expandir se novas fontes não puderem ser mapeadas para entidades existentes;
- Identificação de entidades: é identificar em tempo real as diferentes entidades nas fontes de dados;
- Fusão de dados: integração das fontes de dados já extraídas;
- Pós-processamento: preparação para apresentação das informações na interface web.

Após a conclusão das etapas e integração realizada, a interface web foi desenvolvida utilizando a plataforma *Information Workbench* (IWB) para condicionar o usuário a visualizar as relações entre pessoas, entidades do governo e indústria [26].

Também preocupado com a integração das informações e com o grande desperdício de tempo em analisar, importar e exportar dados, até mesmo para os desenvolvedores criarem as in-

interfaces de visualização das informações obtidas, que Seligman projetou o OpenII. O OpenII é um conjunto de ferramentas responsável pela integração dos dados abertos, no qual foi utilizado para testes da ferramenta, altas escalas de dados governamentais dos Estados Unidos [27].

A ferramenta é constituída de uma plataforma escalável e extensível, onde outras podem se conectar a ela para obter parte de sua funcionalidade. A arquitetura OpenII é composta inicialmente por um repositório compartilhado, que implementa um modelo de entidade de relacionamento neutro e estendido para esquemas e mapeamentos próprios. Implementado baseado em banco de dados relacional, no qual é possível inicialmente utilizar no Postgres ou Derby. Além do repositório, é possível incluir uma variedade de importadores e exportadores, sendo XML Schema, SQL DDL, OWL e planilhas do Excel. A ferramenta funciona trocando informações a partir de *web services* desenvolvido em Java e suas APIs, a interface baseia-se na integração Workbench, que foi usado pela primeira vez para integrar uma ferramenta de mapeamento de esquema, sendo que os repositórios podem se conectar entre si [27].

A ferramenta OpenII teve aplicação de sucesso em várias tarefas do governo dos Estados Unidos, como auxílio no mapeamento entre os modelos de dados de algumas organizações com infraestruturas críticas, como os aeroportos e usinas de energia. Além disso, a Força Aérea dos EUA está usufruindo também para combinar não apenas esquemas, mas grandes coleções de códigos e suas definições. Algumas organizações estão realizando experimentos para visualizar as relações entre centenas de esquemas em seus depósitos existentes. Outras organizações estão avaliando como o OpenII pode acelerar a adoção de padrões de troca de dados, tornando mais fácil para sistemas existentes criar mensagens conforme padrão da arquitetura criada para OpenII [27].

Já Coria [28], curioso em investigar a política pública e fenômenos econômicos e sociais dos municípios mexicanos, utilizou os dados adquiridos no censo de 2010, realizado pelo Instituto Mexicano de Estatística e Geografia (INEGI), para elaborar o MuniMex 1.0., o qual é uma interface básica e simples para seleção e exportação de dados sociodemográficos mexicanos. Os dados do recenseamento foram interpretados por 191 variáveis, correspondentes a 2.456 municípios, no qual uma coleção de variáveis percentuais foram calculadas e inseridas no banco de dados, com isso proporcionando análises mais críticas dos dados e possibilitando a exportação dos resultados em arquivos. O formato de arquivo onde podem ser exportados os dados são: Excel e CSV, servindo ainda para integração com ferramentas estatísticas e até mesmo para fins de conhecimento da população.

Os dados do INEGI abrangeram 14 categorias de análise:

- Identificação geográfica: identificação e nomes geográficos;
- População: quantidade de habitantes por segmentos de gênero e idade;
- Fecundidade: média de filhos;
- Migração: população que vive no mesmo estado por segmentos de gênero e idade, nos anos 2005 e 2010;
- População indígena: população que fala um idioma indígena por segmentos de gênero e idade;

- Incapacidade: população que apresenta deficiência motora, visual, auditiva ou mental;
- Educação: população por nível de escolaridade por segmentos de gênero e idade;
- Características econômicas: relação de empregados e desempregados. Ou seja, população economicamente ativa ou inativa, considerando segmentos de gênero e idade;
- Serviços de saúde: relação de população com serviços de saúde do Instituto Mexicano de Segurança Social (IMSS), do Instituto de Segurança Social dos Funcionários do Governo Federal (ISSSTE), dos 32 institutos estaduais de segurança social ou do Seguro Popular;
- Estado civil: população solteira, casada, divorciado ou separado;
- Religião: população com ou sem religião, considerando católico, cristão em geral, entre outros;
- Famílias do recenseamento: estes são os agrupamentos familiares que vivem em uma casa;
- Características da habitação: número de casas (ou famílias) que possuem serviços, infraestrutura ou bens, tais como: água potável, esgoto, piso de terra ou barro, eletricidade, telefone, internet, etc;
- Tamanho da localidade: o INEGI classifica entre 01 e 14 pontos as localidades de acordo com a sua quantidade de habitantes.

Atualmente, o MuniMex está disponível na língua espanhola, mas uma versão está em estruturação na língua inglesa. A interface MuniMex foi desenvolvida na linguagem Java e compatível com banco de dados Mysql [28]. A figura 11 ilustra a interface.

Conforme Figura 11 a interface está subdividida em áreas como segue abaixo:

1. escopo e Seleção de granularidade;
2. seleção de tabela de categorias;
 - (a) tabela descrição;
3. seleção de campo;
 - (a) descrição do atributo;
4. exibição de dados detalhados;
 - (a) exibição de totais.

Conforme pesquisas e trabalhos apresentados neste capítulo, pode-se observar a preocupação dos autores em utilizar os dados disponibilizados livremente, da melhor forma, para tornar o cidadão empoderado dessas informações. Os estudos apontam também as diversas formas desenvolvidas para resolver o problema da diversificação na forma que os órgãos públicos utilizam para

The screenshot shows the MuniMex 1.0 interface. At the top, there is a menu bar with 'Archivo' and 'Ayuda'. Below it, a selection area (1) contains radio buttons for 'Nacional por estados', 'Nacional por municipios' (selected), 'Estatad', and 'Municipal'. The main area is divided into several sections:

- Estados:** A dropdown menu showing 'Aguascalientes'.
- Municipios:** A dropdown menu showing 'Aguascalientes'.
- Tabla:** A list of tables (2) including 'caract_economicas', 'caract_educativas', 'discapacidad', 'fecundidad', 'hogares_censales', 'idengeografica', 'migración', 'poblacion', 'poblacion_indigena', 'religion', 'servicios_salud', and 'situacion_conyugal'.
- Campos:** A list of fields (3) including 'PDESOCUP', 'PDESOCUP_F', 'PDESOCUP_F_PORCENT', 'PDESOCUP_M', 'PDESOCUP_M_PORCENT', 'PDESOCUP_PORCENT', 'PEA', 'PEA_F', 'PEA_F_PORCENT', 'PEA_M', 'PEA_M_PORCENT', and 'PEA_PORCENT'.
- Información:** A section with two text boxes:
 - Descripción de la Tabla:** 'caract_economicas: Características económicas.' (2a)
 - Descripción del Campo:** 'PDESOCUP_F_PORCENT: Porcentaje de Población femenina desocupada.' (3a)
- Datos:** A data table (4) with columns: 'CIUDAD', 'MUN', 'NOM_ENT', 'NOM_MUN', and 'PDESOCUP'. The table contains 20 rows of data for various municipalities in Aguascalientes and Baja California. Below the table is a 'Totales' section (4a) with columns: 'PDESOCUP', 'PDESOCU', and 'PDESOCU.'. The totals are: PDESOCUP: 031369.0, PDESOCU: 433488.0, PDESOCU.: 0.4.

At the bottom right, there are three buttons: 'Eliminar columna', 'Exportar a .xls', and 'Exportar a .csv'.

Figura 11. Interface MuniMex 1.0 [28].

disponibilizar as informações, de ordem obrigatória, nos portais de transparência. As ferramentas tecnológicas são grandes aliadas para tornar o cidadão cada vez mais próximo dessas informações e da realidade da sua comunidade e região. Na Tabela 2 pode-se visualizar as principais características de cada trabalho mencionado acima, assim como, uma breve descrição da OSCV. Onde é possível identificar que assim como os demais trabalhos, a OSCV possui uma interface web para disponibilizar as informações, porém na OSCV é possível configurar a coleta e visualização para várias áreas de pesquisa, não se limitando a áreas específicas. No capítulo seguinte será abordada a metodologia utilizada no referente trabalho.

Tabela 2. Tabela comparativa de trabalhos relacionados

GovWILD	OpenII	MuniMex 1.0	OSCV
Integra e interconecta dados governamentais, especificamente dados financeiros e de fundos públicos dos Estados Unidos e União Européia.	Conjunto de ferramentas responsável pela integração dos dados abertos. Possui uma plataforma escalável e extensível conectada a um repositório compartilhado.	Através do censo de 2010 o autor criou uma interface básica e simples para seleção e exportação de dados socio-demográficos mexicanos.	Possibilita a configuração e integração de dados governamentais de qualquer área, sendo que os arquivos disponibilizados nos portais de transparência sejam legíveis por máquina. Coletando as informações periodicamente e armazenando em um data warehouse.
Possui uma interface web para visualização das informações.	Possui uma interface baseada na integração Workbench.	Possui uma interface web para visualização das informações.	Possui uma interface web para visualização das informações recentemente atualizadas.
Apresenta cruzamento das informações de várias áreas.	Apresenta cruzamento das informações de várias áreas.	Não apresenta cruzamento das informações de várias áreas.	Apresenta cruzamento das informações de várias áreas.

3. PROCESSO DESENVOLVIMENTO DA INTERFACE E AUTOMATIZAÇÃO DA COLETA DE DADOS NA OPEN SMART CITY VIEW

Este capítulo apresenta a metodologia utilizada para o desenvolvimento do processo de automatização da coleta de dados para a ferramenta OSCV, assim como a elaboração da interface web para apresentação das informações coletadas ao usuário final. A metodologia de desenvolvimento do trabalho proposto foi realizada basicamente em 3 fases, como pode ser observado na Figura 12.

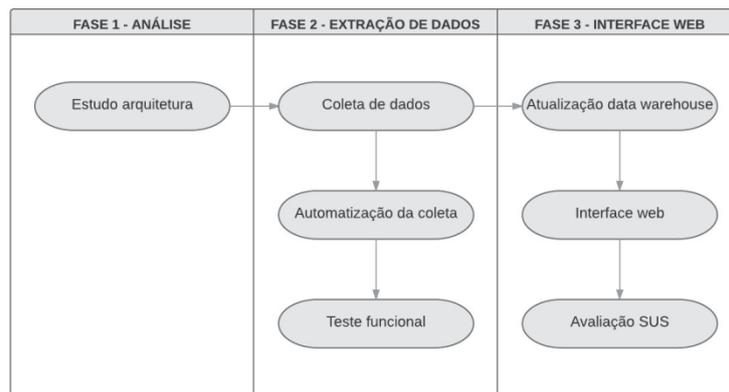


Figura 12. Fluxo ilustrativo da metodologia abordada.

Na fase inicial, designada de análise, realizou-se um período de estudo e observação da arquitetura OSCV, para maior compreensão das suas camadas, funcionalidades e processos. Após o devido conhecimento da arquitetura. Já na segunda fase, analisou-se mais especificamente a camada de Coleta e Processamento, no que diz respeito à coleta de dados nos portais de transparência, que era realizado manualmente. Esse processo de coleta era operacionalizado da seguinte forma: primeiramente identificava-se a fonte de dados a ser extraído o arquivo. Ou seja, o portal de transparência desejado, após navegava-se pelo portal até encontrar a fonte de interesse e, em seguida, realizava-se filtros de pesquisa até localizar os dados desejados. Por fim, realizava-se o *download* do arquivo em formato compreensível por máquina. Com base no processo de coleta dos arquivos realizado manualmente, é que foi projetada a extração dos dados na fonte de forma automatizada, desenvolvendo um *web crawler* para capturar os arquivos na origem e realizar o *download* do mesmo.

Na terceira e última fase, analisou-se o funcionamento das tarefas de Transformação e Enriquecimento do *data warehouse*, tarefas essas que fazem parte da camada de Coleta e Processamento, no qual esse processo passa por uma linha de extração, transformação e carga dos registros no *data warehouse*, processo esse conhecido como ETL (*Extract, Transform and Load*). Com os dados armazenados no banco de dados, projetou-se uma interface web de fácil acesso e entendimento para apresentar as informações armazenadas de forma clara ao usuário final. A interface divide-se em ambiente *backend*, designado para operações administrativas e de configuração da ferramenta, e ambiente *frontend*, a qual estará disponível para interação com o usuário, disponibilizando os dados armazenados no banco de dados para que este possa ter acesso as informações. Por último, efetuou-

se a avaliação da interface com o método de avaliação SUS (*System Usability Scale*), permitindo a análise dos resultados obtidos para possível evolução da ferramenta. Nas seções seguintes será descrito mais detalhadamente o desenvolvimento das fases.

3.1 AUTOMATIZAÇÃO DO PROCESSO DE COLETA DE DADOS

Essa atividade faz parte da segunda fase de elaboração do respectivo trabalho, a automação da extração dos dados à partir dos portais de transparência é a contribuição de evolução na camada de Coleta e Processamento da ferramenta OSCV, pois na sua primeira versão esse processo de extração e coleta dos arquivos nos portais governamentais eram realizados manualmente (Figura 13). Esse processo depende sempre do esforço e disponibilidade de tempo do administrador da ferramenta, para realizar as tarefas de busca e captura dos arquivos. Sendo esta uma atividade repetitiva de se executar, pois as tarefas eram realizadas várias vezes e sempre da mesma forma. Contudo, havendo o risco de ocorrer uma falha humana e a atividade não ter êxito de conclusão.

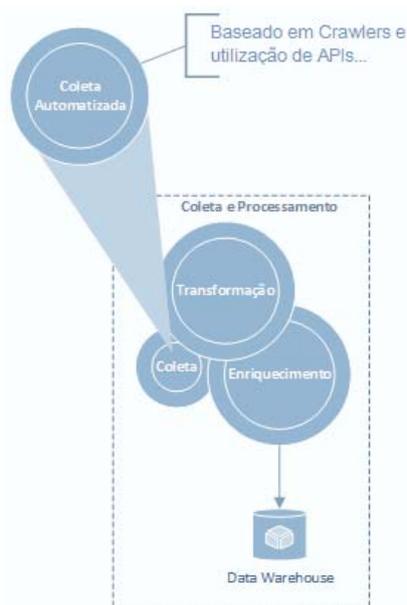


Figura 13. Camada de Coleta e Processamento OSCV. Adaptada [1]

A partir disso, surgiu a necessidade de automação do processo de extração dos arquivos, onde projetou-se e desenvolveu-se um módulo complementar a ferramenta OSCV. Módulo este composto por *Threads* que executam atividades desde a consulta aos portais abertos de transparência até a extração dos arquivos desejados. Assim como os demais métodos responsáveis pelo funcionamento da ferramenta, esse módulo foi desenvolvido na linguagem Java. Optou-se por esta linguagem por ser não apenas uma linguagem, mas sim uma plataforma de desenvolvimento, podendo desenvolver aplicação para desktop, mobile, web, entre outros. É uma linguagem multiplataforma muito reconhecida no mercado de desenvolvimento de aplicações, possui uma vasta gama de *frameworks* que visam auxiliar o trabalho do desenvolver, além de suas *APIs open sources*.

A implementação do módulo de extração baseou-se no processo de *crawling*, pelo motivo de que inúmeras empresas renomadas no mercado (Google, Yahoo, Bing, entre outras), fazem uso de ferramentas e técnicas baseadas em *web crawlers* para realizar as tarefas de automação dos processos de indexação e mineração dos conteúdos. A maioria dos processos de *crawling* são desenvolvidos e voltados para a web tradicional, onde documentos HTML conectados por *hyperlinks*, tags e palavras chaves são usados para rastrear e atender necessidades específicas, como *download* de imagens e arquivos, coleta de conteúdos específicos, além de agregar dados de diversas fontes da internet [23].

Basicamente, o processo de extração e coleta de dados implementou-se com bibliotecas e APIs *open source* do Java, mais especificamente o HtmlUnit, na qual permite que suas funções manipulem comandos e ações nos portais e sites da mesma forma que o usuário realizaria no navegador como, por exemplo, acesso a uma página web, cliques em botões, links entre outras ações, porém sem a utilização de *browser* de navegação gráfico. HtmlUnit foi criado por Mike Bowler do Gargoyle e lançado sob a licença Apache. Tem suporte ao JavaScript, o que está em constante aperfeiçoamento e funciona até mesmo com bibliotecas AJAX mais complexas, simulando o Chrome, o Firefox ou o Internet Explorer dependendo da configuração usada. Desde então, recebeu muitas contribuições de outros desenvolvedores, por ser livre [29].

Mas comandos precisam ser fornecidos para a realização das funcionalidades da biblioteca HtmlUnit. Com isso algumas tarefas precisaram ser realizadas na ferramenta para a correta execução da biblioteca. Primeiramente, é fundamental a identificação prévia dos endereços web nos quais os arquivos devem ser coletados, para que o administrador responsável pela ferramenta ou qualquer pessoa capacitada a realizar as tarefas administrativas, e com cadastro habilitado, realize a configuração dos comandos a serem executados pela aplicação. A configuração é realizada no ambiente de acesso restrito para administrador, na interface web desenvolvida exclusivamente para a OSCV, seu desenvolvimento será abordado no próximo capítulo. A seguir, serão descritos os passos necessários para o cadastramento dos comandos, onde com usuário e senha cadastrados o responsável tem permissão de acessar o portal da ferramenta OSCV, no menu “Login” para realizar a entrada no ambiente administrativo da ferramenta. Após acessar o menu “Extração de Dados”, disponível na parte direita superior da interface, para visualizar as configurações já cadastradas e possível adição de novos cadastros. A figura 14 ilustra a lista de configurações já cadastradas para a realização da extração da coleta de dados.

Para realizar o novo cadastro, deve-se apenas clicar no botão “Adicionar”, o qual direcionará para a referida interface de cadastro. Nessa interface, encontram-se os campos que precisam ser cadastrados para a devida extração de arquivos. Estes campos são:

- Descrição: o qual é um nome descritivo da configuração;
- Endereço: contém o endereço do portal de transparência de onde será realizada a extração;
- URL salvar arquivo: o caminho no servidor de onde o arquivo extraído será armazenado;
- Nome arquivo: nome que o arquivo receberá após ser realizado o *upload* no servidor;

Open Smart City View Apresentação dos fatos Jobs Configuração Extração de Dados Sair

Automatização de dados Adicionar

5 resultados por página

	Código	Descrição	Endereço	Ativa
 	1	Seguranca	http://dados.fee.tche.br/	true
 	872	Saude_Internacoes Hosp_NroInterAno	http://dados.fee.tche.br/	true
 	873	Saude_Internacoes Hosp_TaxaMortAno	http://dados.fee.tche.br/	true
 	874	Saude_NroLeitos_total	http://dados.fee.tche.br/	true
 	875	Frota	http://dados.fee.tche.br/	false

Mostrando de 1 até 5 de 5 registros Anterior 1 Próximo

Figura 14. Interface de listagem das configurações da extração de dados.

- Job execução: contém as opções de escolha dos Jobs, sendo estes Jobs responsáveis pelos scripts ETL das interfaces de carga do *data warehouse*;
- Ativa: opção de tornar ativa ou não a configuração de extração;

Esses campos são melhor visualizados na Figura 15.

Gerenciar automatização na coleta de dados [Saude_NroLeitos_total]

Descrição*

Saude_NroLeitos_total

Endereço*

http://dados.fee.tche.br/

Url salvar arquivo*

\\home\upf\arquivo_download\saude_nroLeitosTotal

Nome arquivo*

saude.zip

Job execução

LOAD_SAUDE_FEE

Ativo*

Voltar Salvar

Figura 15. Gerenciar automatização na coleta de dados.

Além desses campos, é fundamental configurar também os scripts de comandos para o processo de execução do HtmlUnit, assim como o tipo e a sequência de execução. Para a configuração dos comandos é necessário cadastrar conforme segue ilustração da Figura 16.

Sequência de comandos para realizar o download de arquivos

Adicionar

10 resultados por página

Comandos	Tipo	Sequência
//a[@title='Abrir Pasta: Saúde']	xpath	1
//a[@title='Abrir Pasta: Número de Leitos Hospitalares']	xpath	2
//a[@title='Selecionar Variável: Total']	xpath	3
unidadegeo Municípios	select	4
//*[@onclick='check_all(true)']	xpath	5
//a[@title='Baixar em formato CSV']	xpath	6

Mostrando de 1 até 6 de 6 registros

Anterior 1 Próximo

Figura 16. Sequência de comandos responsáveis pela extração de arquivos.

Após a configuração, o processo é de total responsabilidade da classe “ColetaAutomatizadaJob”, que é a *Thread* de execução do processamento das informações cadastradas e do algoritmo de *crawler*, na qual realiza a busca e a coleta do arquivo na fonte. Essa *Thread* possui um agendamento de execução controlada pela API Quartz do Java, que executa de tempos em tempos a chamada para a *Thread*. Inicialmente configurou-se para execução uma vez por semana, com flexibilidade de configuração no período de tempo desejado pelo administrador.

O Quartz é uma biblioteca bastante utilizada para agendamento de tarefas em aplicações Java, e utilizado nos mais diferentes tipos e tamanhos de aplicações. É formada de três componentes principais: o *Scheduler* é o núcleo do Quartz e responsável pelo controle das *Triggers* e Jobs que serão executados. Ou seja, é nele que se pode encontrar as *Triggers* e o Jobs que o sistema está usando ou que ainda irá usar; as *Triggers* que são os objetos que contém a lógica da execução, são os objetos que possuem os horários, quantidade de execução de um determinado Job; e os Jobs que são as classes responsáveis pela execução de uma determinada lógica. Em outras palavras, que contém as instruções que serão executadas de tempos em tempos ou em um determinado horário.

O fluxo de atividades realizadas pela *Thread* “ColetaAutomatizadaJob” é composto por algumas etapas, sendo a primeira a execução do método que contém o algoritmo baseado em *crawler*, onde acessa a url do portal de transparência cadastrado e após executa os comandos configurados para captura de arquivos. Esta etapa é de responsabilidade de realização da API Htmlunit, para garantir o correto processamento executa cada comando na ordem de sequência cadastrada até a realização de download do arquivo. A Figura 17 ilustra a visão simplificada do algoritmo.

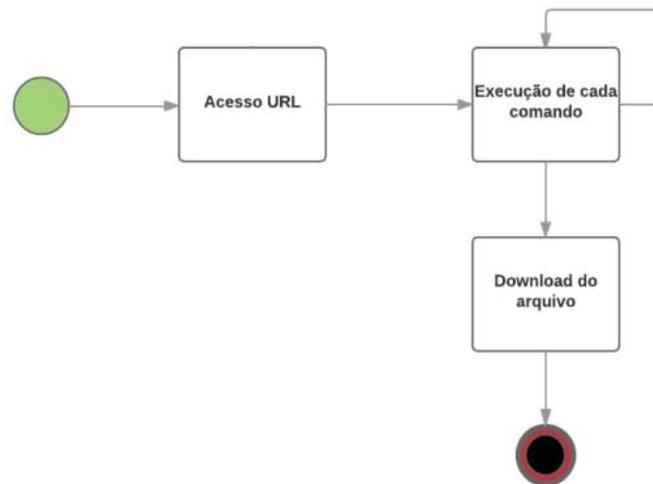


Figura 17. Visão simplificada do algoritmo.

A etapa seguinte é a descompactação do arquivo que foi realizado *download*, caso o mesmo seja disponibilizado pelo portal de transparência de forma compactada. A próxima etapa é de comparação de arquivos, onde verifica se o arquivo que foi realizado *download* contém exatamente os mesmos dados que o último arquivo processado, ou seja, compara-se igualdade dos dados dos arquivos. É utilizado um diretório temporário apenas para armazenar o último arquivo processado pelo script ETL. Caso o novo arquivo não apresentar igualdade nos dados, exclui-se o arquivo que se encontra no diretório temporário e move-se o novo arquivo para este diretório. E a última etapa é a chamada da execução do script ETL, o qual alimentará o *data warehouse* com os dados do novo arquivo. Esse fluxo é melhor visualizado na Figura 18.

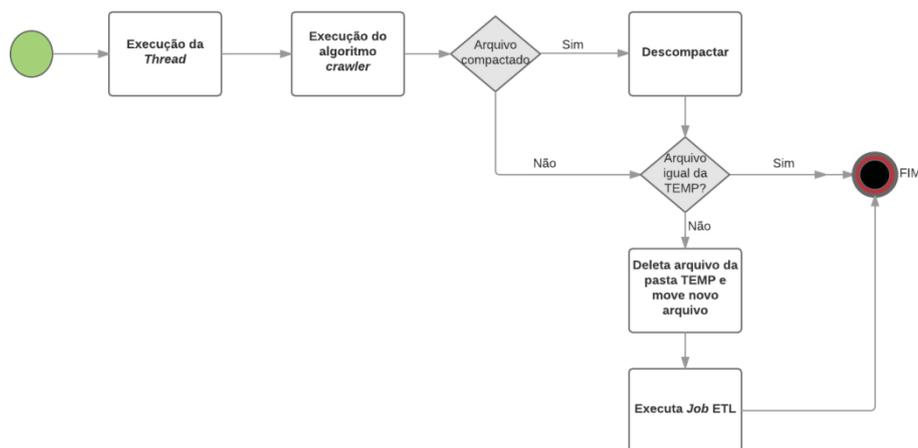


Figura 18. Fluxo de processamento da *Thread*.

Para garantir que as etapas estão sendo realizadas corretamente e que o *data warehouse* está recebendo carga atualizada de informações, para cada processo realizado é gerado log de controle e acompanhamento, onde o administrador tem acesso a cada atividade de *download* realizada(Figura 19).

Open Smart City View Apresentação dos fatos Jobs Extração de Dados - Sair

Log de processamento

5 resultados por página

Código	Mensagem	Data	Status
977	Download de arquivo: saude.zip realizado: http://dados.fee.tche.br/	2017-11-15 23:35:09.63	SUCES
976	Download de arquivo: saude.zip realizado: http://dados.fee.tche.br/	2017-11-15 23:34:55.167	SUCES
975	Download de arquivo: saude.zip realizado: http://dados.fee.tche.br/	2017-11-15 23:34:40.356	SUCES
974	Download de arquivo: seguranca.zip realizado: http://dados.fee.tche.br/	2017-11-15 23:34:25.487	SUCES
973	Download de arquivo: saude.zip realizado: http://dados.fee.tche.br/	2017-11-15 23:32:35.131	SUCES

Mostrando de 1 até 5 de 28 registros

Anterior 1 2 3 4 5 6 Próximo

Figura 19. Interface de Log de processamento.

Caso alguma falha ocorra no processamento das etapas, como por exemplo a não localização do arquivo desejado ou mudança no formato do arquivo, no qual implicaria a não atualização do *data warehouse*, entre outras exceções, a aplicação gera log de erro para que o responsável tenha conhecimento das ocorrências e possa intervir com a resolução do problema. Além de logs, envia-se e-mail de notificação para o administrador com descrição do erro ocorrido (Figura 20). O administrador tem a opção de habilitar ou não o recebimento de notificações da aplicação por e-mail.

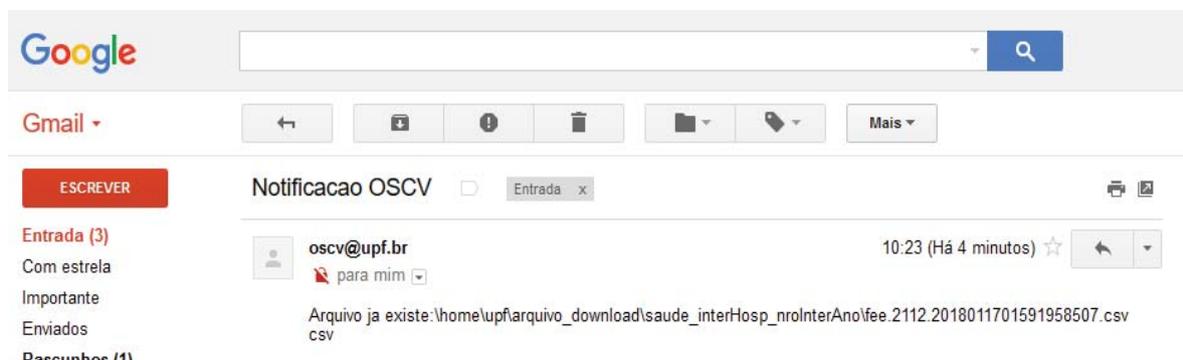


Figura 20. Exemplo de e-mail de notificação.

Seguinte à implementação do módulo de extração, realizou-se o processo de teste funcional para demonstrar o correto funcionamento dos métodos desenvolvidos para a elaboração da automação na coleta de dados. O próximo tópico conterá a descrição do teste realizado.

3.1.1 Teste do Processo

Além do planejamento e desenvolvimento de uma aplicação ou de um *software* o teste é fundamental, pois é através dos processos de testes que busca-se avaliar se as funcionalidades estão trabalhando de acordo com as especificações e requisitos do projeto, garantindo que o *software* atinja o nível de qualidade esperada pelo público de maior interesse [30].

A literatura apresenta várias técnicas de testes, sendo que as mais abordadas e mencionadas são:

- Teste Funcional: também conhecido de teste Caixa Preta, são os testes concentrados de acordo com os requisitos funcionais do *software*. Sem necessidade de conhecimento sobre a operação interna do programa, o analista concentra-se nas funções que o *software* contemplará. Baseado na especificação, determina-se as saídas que são esperadas para representar o funcionamento com êxito [31].
- Teste Estrutural: mencionada na literatura também como teste Caixa Branca, os testes são desenvolvidos a partir dos conceitos de implementação do *software* e da estrutura desenvolvida [31]. Além de possibilitar avaliar a estrutura interna do programa, sua codificação, módulos, implementação e execução por partes do programa e de seus componentes elementares, sendo que estas informações também são utilizadas para a criação dos casos de testes [32];
- Teste Baseado em Falha e Erro: são baseados a partir dos problemas de inconsistência de informações mais comuns que os desenvolvedores enfrentam no processo de desenvolvimento, assim são utilizados para a criação dos casos de testes [33];
- Teste de Regressão: são testes executados para garantir a funcionalidade de um programa já testado, mas que sofreu novas implementações e alterações. Garantindo com isso a qualidade do produto após manutenção.

Para o processo de teste abordado na metodologia de automatização da coleta de dados, optou-se pela técnica de Teste Funcional, por apresentar alinhamento com o objetivo de análise de processo de extração de dados. Como o processo contém métodos separadamente, é possível obter uma avaliação parcial. Assim como uma avaliação global do processo de automatização da coleta de dados, a qual é executada paralelamente à ferramenta OSCV. Além disso, Koscianski [34] descreve ser uma técnica em sua concepção relativamente simples se comparado a outras técnicas. O teste funcional é particularmente útil para revelar problemas tais como:

- Funções incorretas ou omitidas;
- Erros de comportamento ou desempenho;
- Erros de iniciação e término.

Segundo Sommerville, as atividades de testes podem muitas vezes se tornar cansativas e trabalhosas, dificultando assim a execução dos testes de forma adequada para a análise de qualidade. Com o objetivo de melhorar a qualidade da análise e o tempo de execução dos testes, foram criados os testes automatizados, que tornam a execução dos testes de uma forma mais rápida e com maior análise do *software* [32].

Com isso, utilizou-se o *framework* JUnit para realizar os requeridos testes no módulo de extração na coleta de dados da ferramenta OSCV. O JUnit facilita o desenvolvimento e execução

de testes unitários em código Java, pois herda algumas características da própria linguagem [35]. Devido à sua grande versatilidade, pode ser utilizado em conjunto com outras ferramentas para criar testes funcionais automatizados. Assim como no código Java, no JUnit também é criada classes e métodos que executam casos de teste de pequena, média ou grande complexidade. Essas classes ficam encarregadas de testar o *software* através de comandos comparativos para observar o resultado final, se os dados de saída são iguais aos valores esperados pelo caso de teste.

Com auxílio do JUnit, realizou-se o teste unitário para os métodos que compõem o processo de extração automatizada na coleta de dados. Este teste unitário foi gerado para se ter a veracidade dos resultados esperados para cada método e a certeza de integridade de cada função:

- Algoritmo de *crawler*: a execução do algoritmo responsável por acesso ao portal de transparência, localização do arquivo e *download* do mesmo, apresentou sucesso de execução para o processo de uma configuração;
- Descompactação de arquivo: para descompactar o arquivo no qual foi realizado *download*, o teste unitário também foi realizado sem apresentar adversidade;
- Comparação dos arquivos: o método de comparação entre o arquivo já integrado no *data warehouse* e o novo arquivo obteve resultado bem sucedido de execução;
- Job ETL: a chamada para o processamento do Job ETL, responsável pela alimentação e atualização do *data warehouse* realizou a atividade correspondente de execução do Job ETL.

Na Figura 21 pode-se observar os resultados de sucesso obtidos para cada teste unitário com o JUnit descrito. E através dos resultados a análise de conclusão é correta a afirmar que todos os processos foram realizados com sucesso. Sendo cada método responsável por um processo específico descrito acima.



Figura 21. Resultado teste JUnit.

Uma das características mais interessantes na utilização do teste unitário é que o mesmo pode ser realizado com código particionado. Ou seja, pode-se testar funções separadamente ou realizar testes a cada nova implementação no código fonte, assim como descrito acima. Após a obtenção

do resultado do teste, ilustrando a atualização das informações no *data warehouse* com êxito. A próxima etapa é a apresentação dessas informações para o usuário de uma forma simples, onde será descrito no próximo capítulo o desenvolvimento da interface web.

3.2 DESENVOLVIMENTO DA INTERFACE WEB

Após a atualização dos dados coletados para o *data warehouse*, é indispensável a apresentação das informações para que os usuários possam realizar as devidas consultas, provendo novos conhecimentos e fortalecendo ainda mais o seu empoderamento com informações atualizadas de seu município ou região. Com isso, projetou-se uma interface web para que o usuário pudesse fazer uso da mesma para a realização de pesquisas condizentes com assuntos do seu interesse e cotidiano, de uma forma clara e simples, sem a necessidade de conhecimentos técnicos para conseguir conduzir essa atividade. Além disso, acoplado na mesma interface web, dispõem de um link para acesso a interface administrativa da ferramenta, onde o administrador tem permissão para realizar configurações e manutenções da ferramenta OSCV.

Para o desenvolvimento da interface, foram utilizadas tecnologias concordantes com a demanda observada, fazendo-se uso da tecnologia *Java Server Faces*(JSF), na qual é um *framework* para aplicativo Web que simplifica o designer da interface com o usuário e separa ainda mais a apresentação de um aplicativo Web da sua lógica de negócio [36]. Já na visão da Oracle Corporation [37], JSF refere-se a um conjunto de APIs para apresentação de componentes visuais e controle dos seus respectivos estados, eventos, validação de entrada do usuário, definição de navegação, internacionalização e acessibilidade. Estas APIs constituem um arcabouço arquitetural que permite uma clara separação entre a lógica da aplicação e a sua apresentação.

Já Martin descreve que o JSF integra a implementação do padrão Modelo-Visão-Controlador, comumente conhecido pelo acrônimo MVC e com total relação ao termo de *design patterns*. Esse padrão de arquitetura possui forte apelo pela separação entre as camadas de uma aplicação, em especial, na separação dos modelos de dados criados para representar a percepção do mundo real da sua apresentação final na interface com o usuário, por meio de componentes gráficos. Assim, para atender a adequada desacoplação entre modelo e apresentação, o padrão MVC designa o desenvolvimento da aplicação em três camadas distintas: camada de modelo, de visão e de controle [38].

Além disso, o JSF permite o gerenciamento das aplicações, controlando seu ciclo de vida e outros aspectos gerais como alta disponibilidade, segurança e gerenciamento de memória. A plataforma básica de gerenciamento de soluções web é conhecida como *web container*, pelo motivo do JSF ser incorporado pela Java EE, permite que a execução da aplicação seja realizada por qualquer *web container* Java conhecido no mercado, como GlassFish da Oracle, JBoss AS e WebLogic, entre outros.

Junto ao JSF integrou-se outras bibliotecas *open source*. Uma delas é o *framework* Bootstrap, com objetivo de tornar a interface com designer responsivo e mais amigável com o contexto. Bootstrap usa CSS tradicional, mas seu código fonte utiliza os dois pré-processadores CSS mais po-

pulares, Less e Sass. Além de ser facilmente integrado com o JSF, possui uma extensa e específica documentação para elementos HTML e CSS [39]. Optou-se pela utilização das duas tecnologias, por ser muito afamado pelo mercado de desenvolvimento web, em diferentes níveis de aplicações.

Além disso, foram utilizados também *framework* para controle de sessões e conexões com banco de dados. Utilizaram-se as bibliotecas Hibernate e Spring, cada uma com suas funções específicas de persistência de dados e controle de transações, facilitando e reduzindo significativamente o tempo gasto de desenvolvimento. O Hibernate também é responsável pelo mapeamento das classes para as tabelas do banco de dados, além dos tipos de dados para tipos SQL, fornecendo recursos de consulta e recuperação de dados [40].

A interface web é dividida em duas visões funcionais: *backend* e *frontend*. A visão funcional *backend* possui a interface projetada especificamente para administração da ferramenta, na qual um administrador habilitado para realizar esta função possui acesso a interface *backend*. Já a visão funcional *frontend*, contém a interface de acesso irrestrito ao público em geral, onde através de pesquisas obtêm-se as informações armazenadas no *data warehouse*, designada de *frontend*.

3.2.1 Visão funcional *backend*

A visão *backend* diferente da visão *frontend*, é formada por processos mais técnicos que exigem um conhecimento prévio do administrador da ferramenta sobre a arquitetura e funcionalidades da ferramenta OSCV para realizar as devidas configurações e manutenção da mesma. A visão *backend* encontra-se acoplada ao contexto da interface *frontend*, onde através de um link “Login”, localizado no canto superior direito da interface, é permitido realizar o acesso para o ambiente administrativo, conforme Figura 22.

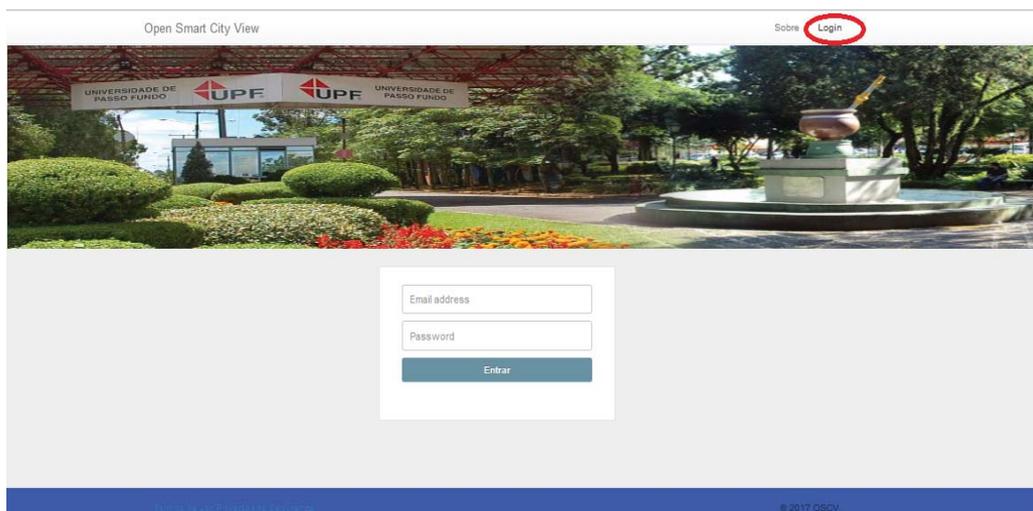


Figura 22. Interface login.

Com usuário e senha cadastrados, é possível acessar o ambiente administrativo e realizar a configuração e visualização dos fatos, perspectivas e perspectivas relacionadas, mencionados no capítulo 2.2.3. Além da adição e configuração dos Jobs, responsáveis pela execução do processo ETL.

E a possibilidade de configuração do processo de automatização na coleta de dados, descrito no capítulo 3.1. Essas opções de acesso encontram-se na barra menu, disponível no ambiente administrativo, conforme pode ser visualizado na Figura 23.



Figura 23. Menu ambiente administrativo.

Ao acessar o menu "Apresentação dos fatos", visualiza-se a listagem dos fatos já configurados, além da possibilidade de adição de novos fatos através da opção "Adicionar", que se localiza na parte superior direita da tabela. A Figura 24 ilustra a listagem de apresentação de fato configurada na ferramenta OSCV.

	Código	Nome	Tabela Fato	Dica	Situação
	1	Segurança	fact_security	Dados sobre crimes por município e período	Ativo
	2050	Crimes de Transito	fact_traffic_accident	Apresenta a relação de crimes de transito	Ativo
	2	Frota	fact_vehicle_fleet	Dados sobre frota de veiculos	Ativo
	2100	Produto Interno Bruto	fact_pib	Produto Interno Bruto dos municípios	Ativo
	2000	Demografia	fact_demography	Dados sobre crescimento populacional	Ativo

Mostrando de 1 até 5 de 7 registros

Figura 24. Listagem de apresentação de fato.

Contudo, a edição dos fatos é um fator importante para a consumação da manutenção. A realização da manutenção pode ser feita ao clicar no ícone de edição, na primeira coluna da tabela de listagem dos fatos, com isso, apresentando a interface de edição conforme Figura 25.

Manter apresentação de fato

Nome*
Segurança

Ajuda
Dados sobre crimes por município e período

Esquema*
fact

Descrição
Visualizações relacionadas ao assunto Segurança descrevem informações relevantes ao cidadão referentes a crimes cometidos em determinado município.

Tabela*
fact_security

Ativo*

[Voltar](#) [Salvar](#)

Figura 25. Interface de visualização e edição dos fatos.

Cada fato contém uma ou mais perspectivas associadas, cada perspectiva associada pode ser visualizada ou editada acessando o link de edição, disponível ao lado esquerdo do código identificador da perspectiva na tabela de listagem, conforme ilustrado na Figura 26. Já as perspectivas

relacionadas encontram-se vinculadas com as associadas, sendo opcional a existência de perspectiva relacionada com a sua representante associada. No caso de existência de perspectiva relacionada, sendo possível também editar, adicionar ou visualizar as mesmas.

Código	Nome	Titulo	Tipo de Gráfico
650	Relação de tipos de crime	Relação percentual dos tipos de crime	PIE
904	Violência contra mulher X meses do ano	Violência contra mulher X meses do ano	BAR
3	Violência contra a mulher	Violência contra mulher no município	LINE
902	Violência contra mulher X dias da semana	Violência contra mulher X dias da semana	PIE

Figura 26. Interface de perspectivas associadas.

Além da manutenção dos fatos no ambiente administrativo, é possível configurar também os Jobs responsáveis pela execução dos scripts ETL. Através do menu "Jobs", pode-se acessar a listagem de configuração já cadastradas na ferramenta OSCV (Figura 27) e também realizar manutenção na mesma.

Código	Nome	Diretório de entrada	Nome do arquivo	Situação
1	LOAD_RAPE_CRIMES_SSP	\home\upf\dados\seguranca\dados_rs\violencia_contra_mulher\	^estupros-mulher-dados-aberto.*.csv\$	Ativo
2	LOAD_CITIES_IBGE	\home\upf\dados\gerais\ibge\municipios\legislacao	*.xls	Ativo
3	LOAD_PIB_FEE	\home\upf\dados\economia\fee_dados_abertos\pib	*.csv	Ativo
4	LOAD_VEHICLE_FLEET_BY_TYPE_DENATRAN	\home\upf\dados\transito\denatran\frota\tipo_municipio	*.xls	Ativo
5	LOAD_TRAFFIC_ACCIDENTS_DADOS_RS	\home\upf\dados\transito\dados.rs.gov\acidentes	*.csv	Ativo

Figura 27. Listagem de Jobs.

Cada configuração pode sofrer novas manutenções e alterações, através do ícone de edição contido na primeira coluna da tabela de listagem. A Figura 28 ilustra a interface de edição com os respectivos campos:

- Nome: nome da configuração;

- Arquivo de integração: descrição do nome do arquivo de script, em formato ktr;
- Descrição: informação auxiliar para identificar a funcionalidade da configuração;
- Diretório de origem: caminho url da pasta onde se encontra o arquivo a ser processado pelo script ETL;
- Padrão nomenclatura arquivo: definição em formato regex da nomenclatura do arquivo a ser processado;
- Diretório de destino (Sucesso): caminho url da pasta onde o arquivo será armazenado no caso de sucesso de execução;
- Diretório de destino (Erro): caminho url da pasta onde o arquivo será armazenado no caso de erro de execução;
- Controle: controle de status de processamento da configuração se ativou ou não.

Gerenciar Job

Nome*
LOAD_EDUC_DOCENTE_INF_FEE_INTERFACE

Arquivo de integração*
LOAD_EDUC_DOCENTE_INF_FEE_INTERFACE.ktr

Descrição*
Transformação responsável pela carga do arquivo educacao docente ensino infantil da FEE

Diretório de origem
\home\up\dados\educacao\ensinoinf\docente

Padrão nomenclatura arquivo
*.csv

Diretório de destino (sucesso)
\home\up\dados\educacao\ensinoinf\docente\succes

Diretório de destino (erro)
\home\up\dados\educacao\ensinoinf\docente\error

Controle
Ativo

Voltar

Figura 28. Visualização de Jobs.

3.2.2 Visão funcional *frontend*

Visão *frontend* é responsável por apresentar as informações armazenadas no *data warehouse* para o usuário final da plataforma Open Smart City View. Contempla em sua interface a exibição dos fatos configurados para pesquisa do usuário e disponibilização das informações em formato gráfico, para facilitar e agilizar a compreensão dos resultados retornados para cada pesquisa efetuada.

Na versão atual da ferramenta OSCV está disponível para visualização sete fatos com suas referentes perspectivas, conforme configurado no ambiente administrativo (Figura 29). Havendo possibilidade de inclusão de novos fatos e perspectivas, enriquecendo ainda mais a ferramenta OSCV com novas informações e tornando cada vez mais completa.



Figura 29. Interface *frontend*.

Para que o usuário possa realizar a pesquisa desejada, basta acessar um dos fatos apresentados e selecionar as opções de escolha de critérios para a consulta, conforme Figura 30. O critério de filtragem montado em formato árvore, seguiu-se o elaborado no protótipo da versão anterior da ferramenta OSCV, pela praticidade de ilustração, através da lógica da arquitetura desenvolvida pelo autor. Algumas opções de filtragem que estavam presentes no protótipo foram retiradas nessa versão, como seleção de estado, cidade e intervalo de tempo. Pois o foco deste trabalho é armazenar dados públicos específicos para o município de Passo Fundo, apresentando sempre nas pesquisas dados atualizados, ou seja, dados mais recentes disponibilizados pelos portais de transparência.



Figura 30. Interface *frontend* - pesquisa.

Após realizar a seleção almejada, os dados retornados da busca são apresentados em formato gráfico, contendo descrição e informações específicas referentes aos fatos consultados, assim como, legenda dos dados contidos no gráfico (Figura 31). De acordo com a primeira versão da fer-

ramenta OSCV, é possível salvar os dados que foram utilizados na composição das séries do gráfico em arquivo, nos formatos XML e CSV.

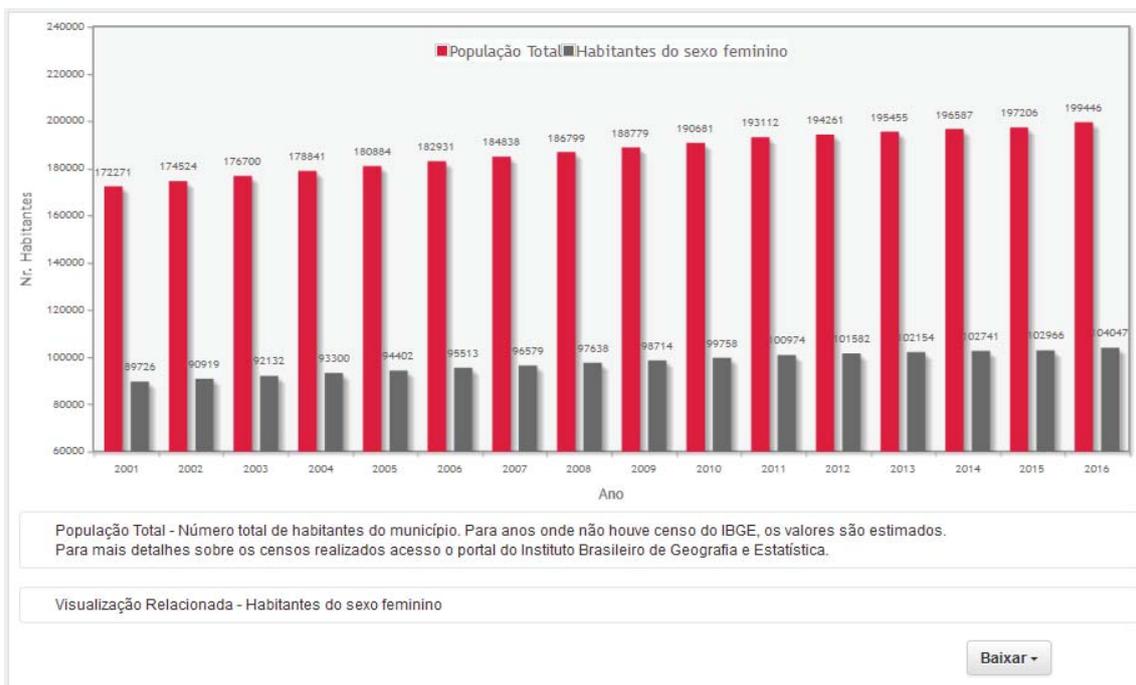


Figura 31. Apresentação de gráficos.

Além da geração dos resultados a serem apresentados através de gráficos para facilitar o entendimento, preocupou-se também nessa versão da ferramenta com a apresentação dos componentes de interface responsivo. A responsividade é um fator relevante para os dias atuais, pois com o grande crescimento do mercado de dispositivos móveis, principalmente tablets e celulares, onde tamanhos e resoluções são cada vez mais variados, e o desenvolvimento das interfaces deve estar de acordo. O designer responsivo, conforme o nome indica, consegue responder ao tamanho da interface para se adequar da melhor forma.

Com isso, tratou-se na ferramenta para que o usuário possa utilizar normalmente o mesmo para realizar suas pesquisas e consultas, sem a necessidade de uso de versões específicas de dispositivos e modelos. A Figura 32, ilustra a interface *frontend* da ferramenta OSCV a partir de um *Smartphone*, com tamanho de interface 5 polegadas.

Um dos objetivos de explorar as informações específicas para o município de Passo Fundo, na versão atual da ferramenta, é a possibilidade de realizar a avaliação da interface web construída para a população municipal. O subcapítulo seguinte abordará a avaliação realizada.

3.2.3 Avaliação da interface web

Após o desenvolvimento da interface web e disponibilização da mesma em um servidor de hospedagem com acesso externo, fornecido pela Universidade de Passo Fundo, foram realizadas avaliações de usabilidade da interface *frontend* com usuários da região de Passo Fundo. Os testes foram



Figura 32. Interface *frontend* responsiva.

aplicados a 18 participantes, entre os dias 20 e 29 de novembro de 2017. Participaram da avaliação pessoas com idade entre 18 e 47 anos, de ambos os sexos e diversas áreas profissionais.

Os participantes eram previamente convidados a colaborar através do preenchimento de uma avaliação a interface. Para alguns deles agendou-se horário e efetuou-se deslocamento até o participante, pela indisponibilidade de locomoção, pelo motivo da significativa importância da sua avaliação e *feedback* ao trabalho desenvolvido. Isso, tendo em vista que se tratava de funcionários públicos municipais, os quais possuem maior conhecimento do domínio abordado no trabalho e que seus pareceres seriam de suma importância para o enriquecimento do projeto desenvolvido. Os demais participantes foram escolhidos aleatoriamente, representando membros da sociedade em geral, como possíveis usuários regulares da ferramenta desenvolvida.

Para cada avaliação realizada, primeiramente era fornecido ao participante um Termo de Consentimento (Apêndice A) e eram realizadas a leitura e assinatura do mesmo. Após, distribuiu-se uma folha (Apêndice B) que possuía uma breve explicação do projeto que estava sendo avaliado e as questões a serem preenchidas, em seguida ocorria a realização do teste de utilização da interface. Para a navegação e utilização da interface foi estipulado um tempo máximo de 10 minutos.

O questionário era composto por 10 perguntas, com objetivo de avaliar a usabilidade da interface. Em cada uma das questões, o avaliador respondeu a partir do seu nível de concordância

seguindo uma escala Likert que varia de 1 a 5, onde 1 representa “Discordo Completamente” e 5 “Concordo Completamente”, conforme ilustrado na Figura 33.

Discordo Completamente 1	2	3	4	Concordo Completamente 5
○	○	○	○	○

Figura 33. Padrão Resposta SUS.

As questões utilizadas para o questionário de avaliação seguiram o método SUS (*System Usability Scale*). Método este testado e consagrado no mercado de avaliação de usabilidade. Essa técnica foi desenvolvida em 1986, por John Brooke, no laboratório da *Digital Equipment Corporation*, no Reino Unido. É um questionário composto por 10 itens e cada um deles com 5 opções de respostas. A pontuação final pode variar de 0 a 100 [41].

O cálculo da pontuação das questões funciona da seguinte forma: os itens ímpares subtrai-se 1 da pontuação que o usuário respondeu. Para os itens pares, subtrai-se da resposta na posição da escala de 5. Para obter a contagem final de cada item, soma-se todos os resultados e multiplica-se por 2,5. Desse modo, o resultado obtido será um índice de satisfação do utilizador [41]. A Figura 34 representa o questionário aplicado na avaliação.

<p>1. Eu acho que gostaria de usar esse sistema com frequência.</p> <p>Discordo Fortemente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Fortemente</p> <p>1 2 3 4 5</p>	<p>6. Eu acho que o sistema apresenta muita inconsistência.</p> <p>Discordo Fortemente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Fortemente</p> <p>1 2 3 4 5</p>
<p>2. Eu acho que o sistema poderia ser menos complexo.</p> <p>Discordo Fortemente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Fortemente</p> <p>1 2 3 4 5</p>	<p>7. Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.</p> <p>Discordo Fortemente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Fortemente</p> <p>1 2 3 4 5</p>
<p>3. Eu achei o sistema fácil de usar.</p> <p>Discordo Fortemente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Fortemente</p> <p>1 2 3 4 5</p>	<p>8. Eu achei o sistema atrapalhado de usar.</p> <p>Discordo Fortemente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Fortemente</p> <p>1 2 3 4 5</p>
<p>4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.</p> <p>Discordo Fortemente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Fortemente</p> <p>1 2 3 4 5</p>	<p>9. Eu me senti confiante ao usar o sistema.</p> <p>Discordo Fortemente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Fortemente</p> <p>1 2 3 4 5</p>
<p>5. Eu acho que as várias funções do sistema estão muito bem integradas.</p> <p>Discordo Fortemente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Fortemente</p> <p>1 2 3 4 5</p>	<p>10. Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.</p> <p>Discordo Fortemente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Fortemente</p> <p>1 2 3 4 5</p>

Figura 34. Questionário SUS.

O questionário SUS auxilia na avaliação dos critérios de efetividade, eficiência e satisfação. Efetividade quando os usuários conseguem completar seus objetivos. Eficiência quando, a partir da métrica, quanto esforço e recursos foram necessários para a realização da tarefa. E satisfação se a experiência obteve resultado satisfatório [41].

4. ANÁLISE E DISCUSSÃO DE RESULTADOS

Na atividade de automatização da coleta de dados, aplicou-se o teste funcional no processo desenvolvido de extração. Com auxílio da biblioteca Junit pode-se testar o processo como um todo, assim como cada método individualmente. Ao finalizar o teste, obteve-se resultado satisfatório na conclusão da execução geral do processo, assim como na execução de cada método. Como pode-se observar no capítulo 3.1.1, todos os testes obtiveram resultado de sucesso, com visualização de sua métrica temporal de execução. Cada método realizou sua função designada:

- Método algoritmo de *crawler*: acessou com sucesso o portal de transparência utilizado para teste, localizou o arquivo desejado e realizou o *download* do mesmo;
- Método de descompactação de arquivo: realizou a atividade de descompactação do arquivo, sem apresentar problemas;
- Método de comparação dos arquivos: a tarefa de comparação entre o novo arquivo, há pouco realizado *download*, e o arquivo já integrado com o *data warehouse*, apresentou sucesso de execução;
- Método job ETL: o processo de chamada ao script ETL também obteve êxito de execução.

Com isso, conclui-se o processo de automatização de extração de dados, como uma tarefa de execução bem sucedida e realizando as especificações propostas.

Já na avaliação da usabilidade da interface *frontend*, aplicou-se o questionário SUS, no qual obteve-se pontuação média de 90,3. Segundo Sauro [41], a média padrão aceitável para o SUS são de 68 pontos, abaixo dessa pontuação obtem-se classificação C, conforme ilustra Figura 35.

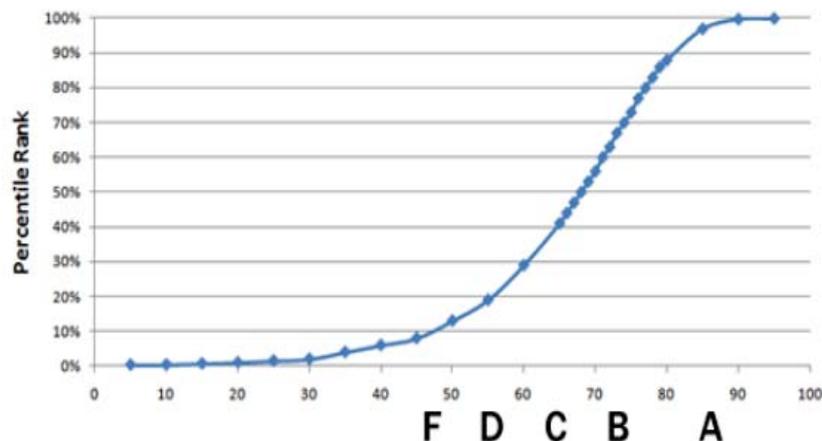


Figura 35. Score SUS [41].

Ao analisar o resultado obtido por questão (Figura 36), verificou-se que nenhuma questão ficou abaixo da média de 68 pontos. Porém, a Questão 1 obteve a média mais baixa entre as questões,

alcançando um grau de satisfação de 4,0 pontos na escala, equivalendo 7,5 na média. Com isso adquirindo conceito B, segundo Sauro [41].

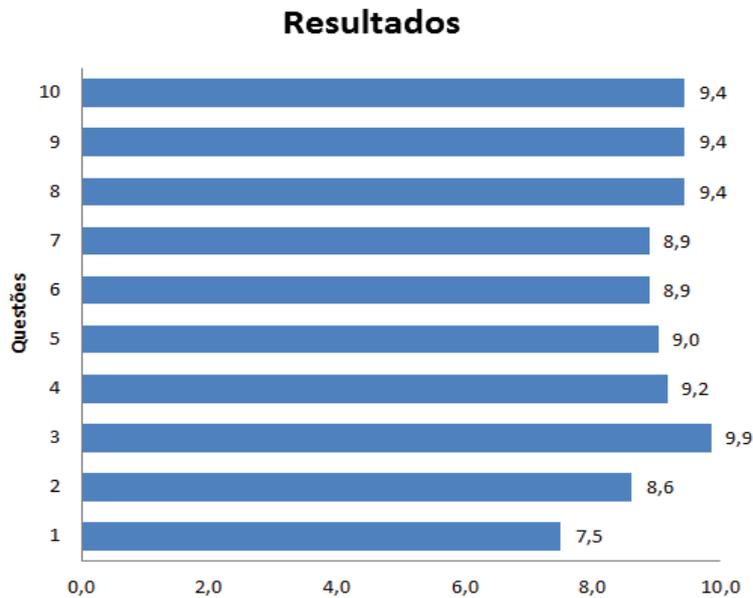


Figura 36. Gráfico de resultado por questão.

A Questão 2 avalia a complexidade do sistema. Obteve média equivalente a 8,6 pontos. Portanto, o usuário não encontrou complexidade na interface.

A Questão 3 avalia a relação de facilidade de uso do sistema. Essa questão obteve a pontuação mais alta entre as questões, 9,9 pontos. Conclui-se que o usuário identificou a interface ser de fácil utilização.

A questão 4 verifica a necessidade de conhecimentos técnicos para realizar a atividade proposta. Obteve pontuação de 9,2.

A questão 5 aborda a integração das funções do sistema. Adquiriu-se 9,0 pontos, considerando que as funções da interface estão de acordo.

A Questão 6 verifica a inconsistência do sistema. Para esta questão, a pontuação foi 8,9, demonstrando que o sistema foi eficiente.

A Questão 7 investiga a rápida aprendizagem do usuário perante a ferramenta, constatando através da pontuação 8,9 não haver problema de aprendizagem da ferramenta.

A Questão 8 que examina a complexidade na utilização da interface, a Questão 9 que verifica o quanto o usuário se sentiu confiante e, por último a Questão 10 avalia a necessidade de aprender coisas novas para conseguir utilizar a interface, obtiveram pontuação de 9,4. Considerando não haver complexidade de utilização, o usuário sentiu-se seguro em utilizar a interface, sem a necessidade de novos conhecimentos para conseguir operar as atividades desejadas.

Com isso, a classificação geral adquirida no questionário aplicado para avaliação da usabilidade obteve conceito A. Resultando na não identificação de maiores problemas de usabilidade na interface apresentada.

5. CONSIDERAÇÕES FINAIS

No presente trabalho desenvolveu-se o processo de automatização para a coleta de dados nos portais de transparência, sendo esta uma atividade evolutiva para a camada de Coleta e processamento da ferramenta OSCV. Além do desenvolvimento da interface web para que o usuário possa ter acesso aos dados de âmbito público, que foram coletados e armazenados pela ferramenta. A interface dividiu-se em duas visões: *backend* e *frontend*.

Na visão *backend*, o acesso fica restrito ao administrador da ferramenta, para que possa realizar as configurações e manutenções pertinentes ao correto funcionamento da mesma. Já na visão *frontend*, os dados coletados e armazenados são disponibilizados para que o usuário final possa se tornar empoderado de informações públicas atualizadas.

Além do desenvolvimento do processo de automatização da coleta dos dados, foram realizados testes funcionais para apresentar a veracidade do funcionamento do processo desenvolvido. Com o auxílio da biblioteca JUnit pode-se visualizar o funcionamento de sucesso dos métodos que compõem o processo todo de automatização.

Já a interface web desenvolvida foi avaliada por usuários da região de Passo Fundo através de questionários baseados no método de avaliação de usabilidade SUS, onde pode-se obter o resultado de satisfação do usuário. Obteve-se conceito A de avaliação, não apresentando maiores problemas de usabilidade na interface. Alguns avaliadores relataram ser agradável e facilitador para a compreensão e comparação dos resultados, a forma de apresentação dos dados utilizada, em formato gráfico. Nenhum dos avaliadores necessitou de ajuda para realizar as pesquisas. Após passadas as informações sobre o projeto, todos realizaram as consultas sem apresentar problema. Dois participantes relataram a importância de utilizar a interface para coletar informações que seriam úteis para a sua profissão. Um desses avaliados, funcionário público e agente de trânsito, relatou a importância dos dados apresentados através da interface consultada para a tomada de decisão em alguns assuntos relacionados a sua atividade prestada no órgão público.

Os objetivos propostos foram desenvolvidos, testados e disponibilizados para que usuários da região do município de Passo Fundo possam fazer uso das informações coletadas e disponibilizadas referentes a esta sociedade. Porém, fica como sugestão para trabalhos futuros o enriquecimento do *data warehouse* com novos fatos e perspectivas. Explorando, assim, novas áreas que possam ser de interesse de conhecimento da comunidade abordada.

Outro trabalho futuro é a expansão da configuração de dados para outros municípios. O objetivo do presente trabalho foi explorar os assuntos do município passofundense, em particular. Porém, não existem limitações de cadastro na estrutura da arquitetura, a qual foi projetada para suportar informações de inúmeros municípios concomitantemente. Além disso, torna-se possível a implementação da possibilidade de comparação de dados entre os diferentes municípios, evidenciando ao cidadão de uma forma mais clara as possíveis diferenças e inconsistências nos fatores governamentais, além de facilitar o processo de empoderamento.

Por fim, buscar aprimorar a interatividade da ferramenta, com algoritmos que possuam técnicas de recomendação, auxiliando o cidadão na aquisição de novos conhecimentos a partir de indicação de consultas relacionadas ao seu perfil ou identificação de similaridade entre outros usuários, para a realização da recomendação de assuntos que já foram recomendados para usuários similares.

5.1 PUBLICAÇÕES

No período de desenvolvimento do trabalho, foram desenvolvidas atividades e publicações relacionadas com o tema estudado e abordado, as quais encontram-se elencadas na sequência:

- RIGO, F.; RABELLO, R. S. . Automatização na extração de dados para Open Smart City View. In: 5º Conferência Ibero Americana Computação Aplicada, 2017, Vilamoura, Algarve, Portugal. Ata das Conferências Ibero Americanas. Vilamoura, Algarve, Portugal: IADIS Press, 2017. p. 255-258.
- RIGO, F.; RABELLO, R. S. ; BELLEI, E. A. ; BREZOLIN, F. L. . Automatização da Extração de Dados na Open Smart City View Utilizando Crawling. In: VIII Encontro Anual de Tecnologia da Informação, 2017, Frederico Westphalen. Anais do EATI - Encontro Anual de Tecnologia da Informação e Semana Acadêmica de Tecnologia da Informação. Frederico Westphalen: IF-Far/UFSM, 2017. v. 7. p. 154-157.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] LUSA, D. A. *OPEN SMART CITY VIEW -Uma Solução Computacional para Manipulação e Apresentação de Dados Governamentais Abertos*. Dissertação (Mestrado) — Universidade de Passo Fundo - UPF, Passo Fundo, RS, Brasil, 2016.
- [2] BERST, J. *Smart Cities Readiness Guide*. Redmond, WA, USA, 2013.
- [3] PRIZIBISCZKI, C. *O crescimento urbano é o problema do século*. Universidade de Cambridge/UK: [s.n.], 2013. Disponível em: <<http://www.oeco.org.br/colunas/colunistas-convidados/27229-o-crescimento-urbano-e-o-problema-do-seculo/>>. Acesso em: Mai. 14, 2016.
- [4] ONU, P. d. N. U. *Centro Internacional de Políticas para Crescimento Inclusivo*. 2011. Disponível em: <<https://nacoesunidas.org/onu-no-brasil/ipc-igpnud/>>. Acesso em: Mai. 16, 2016.
- [5] MACROPLAN. *A Gestão das grandes cidades brasileiras: resultados & práticas*. [S.l.], 2013. 174 p.
- [6] AVELAR, R. E. A. Cidades inteligentes: uma abordagem tecnológica. v. 12, 2013. Disponível em: <<http://www.telesintese.com.br/o-que-sao-aas-cidades-inteligentes-e-o-que-as-torna-smart-uma-abordagem-tecnologica-de-redes-inteligentes-de-comunicacao/>>. Acesso em: Jun. 15, 2016.
- [7] NATIONS, U. *World Urbanization Prospects The 2011 Revision*. [S.l.], 2012. 302 p.
- [8] MOUTINHO, J. L. Das Cidade Digitais às Cidades Inteligentes: notas sobre a coevolução das tecnologias de informação e comunicação e do desenvolvimento urbano da europa. 2010. Disponível em: <https://www.academia.edu/5060528/Das_Cidades_Digitais_%C3%A0s_Cidades_Inteligentes_notas_sobre_a_evolu%C3%A7%C3%A3o_das_tecnologias_de_informa%C3%A7%C3%A3o_e_comunica%C3%A7%C3%A3o>. Acesso em: Jun. 10, 2016.
- [9] STEVENTON A. E WRIGHT, S. *Intelligent spaces: The application of pervasive ICT*. [S.l.]: Springer-Verlag New York, 2006. 438 p. ISBN 978-1-84628-429-8.
- [10] NAPHADE G. BANAVAR, C. H. M. Smarter cities and their innovation challenges. v. 44, p. 32 – 39, 2011. ISSN 0018-9162. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5875937>. Acesso em: Mai. 120, 2016.
- [11] DETROZ CRISTIENNEMAGALHAES PEREIRA PAVEZ, A. P. V. D. Cidades sustentáveis, inteligentes e inclusivas: reinvenção das cidades. 2014. Disponível em: <<http://www.sociesc.org.br/reis/index.php/reis/article/view/26/81>>. Acesso em: Jun. 13, 2016.
- [12] CARAGLIU, A.; Del Bo, C.; NIJKAMP, P. Smart Cities in Europe. *Journal of Urban Technology*, v. 18, n. 2, p. 65–82, 2011. ISSN 1063-0732.

- [13] IBM. *Cidades Mais Inteligentes*. 2016. Disponível em: <http://www.ibm.com/smarterplanet/br/pt/smarter_cities/overview/index.html>. Acesso em: Jun. 10, 2016.
- [14] GRAY, J.; DIETRICH, D.; MCNAMARA, T. *Manual dos dados abertos: governo*. [S.l.], 2010. 58 p. Disponível em: <http://www.w3c.br/pub/Materiais/PublicacoesW3C/Manual_Dados_Abertos_WEB.pdf>.
- [15] PLANEJAMENTO, M. do. *Cidadão pode verificar informações com aplicativo de dados governamentais*. 2013. Disponível em: <<http://www.brasil.gov.br/ciencia-e-tecnologia/2013/05/cidadao-pode-verificar-informacoes-com-aplicativo-de-dados-governamentais>>. Acesso em: Mai. 23, 2016.
- [16] LESSIG, L. *The Annotated 8 Principles of Open Government Data*. 2007. Disponível em: <<https://opengovdata.org/>>. Acesso em: Mai. 26, 2016.
- [17] EAVES, D. *The Three Laws of Open Government Data*. 2009. Disponível em: <<https://eaves.ca/2009/09/30/three-law-of-open-government-data/>>. Acesso em: Mai. 26, 2016.
- [18] VAZ MANUELLA MAIA RIBEIRO, R. M. J. C. *Dados Governamentais Abertos e seus impactos sobre os conceitos e práticas de transparência no Brasil*. 2010. Disponível em: <<http://www.portalseer.ufba.br/index.php/ppgau/article/view/5111/3700>>. Acesso em: Jun. 10, 2016.
- [19] ANGÉLICO, F. *Estudo analítico sobre transparência e legitimidade das organizações da sociedade civil brasileira*. 2011. Disponível em: <<http://www.abong.org.br/fundospublicos/destaques/transparencia>>. Acesso em: Jun. 10, 2016.
- [20] MARTINS FABRÍCIO BOECHAT COELHO, F. M. d. M. A. Douglas da S. *A Importância da Implantação do Portal de Transparência Pública na Prefeitura Municipal de Dores do Rio Preto/es*. 2012. Disponível em: <<http://www.aedb.br/seget/arquivos/artigos12/22716555.pdf>>. Acesso em: Jun. 12, 2016.
- [21] SCHUURMAN, D.; BACCARNE, B.; De Marez, L. *Smart Ideas for Smart Cities: Investigating Crowdsourcing for Generating and Selecting Ideas for ICT Innovation in a City Context*. *Journal of theoretical and applied electronic commerce research*, v. 7, n. 3, p. 11–12, 2012. ISSN 0718-1876. Disponível em: <http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-18762012000300006&lng=en&nrm=iso&tlng=en>.
- [22] LEMOS, A. *Cidades inteligentes*. *GV-Executivo*, v. 12, n. 2, p. 46–49, 2013. Disponível em: <<http://bibliotecadigital.fgv.br/ojs/index.php/gvexecutivo/article/viewFile/20720/19454>>. Acesso em: Mai. 10, 2016.
- [23] RIBEIRO, L. C. R. *Web crawler em Java coleta e filtragem de conteúdo da Web*. Disponível em: <http://www.univale.com.br/unisite/mundo-j/artigos/59_Webcrawler.pdf>. Acesso em: Jun. 28, 2016.

- [24] SHKAPENYUK, T. S. V. Design and Implementation of a High-Performance Distributed Web Crawler. 2001. Disponível em: <<http://cis.poly.edu/tr/tr-cis-2001-03.pdf>>. Acesso em: Jun. 28, 2016.
- [25] RODRIGUES, W. A. *Crawler4j – Web Robot em Java*. Disponível em: <<https://javaes.wordpress.com/2013/04/20/crawler4j-web-robot-em-java/>>. Acesso em: Jun. 23, 2016.
- [26] BÖHM, C. et al. *GovWILD: integrating open government data for transparency*. [S.l.], 2012. Disponível em: <<http://dl.acm.org/citation.cfm?id=2188039://dl.acm.org/citation.cfm?doid=2187980.2188039>>.
- [27] SELIGMAN, L. et al. *OpenII: An Open Source Information Integration Toolkit*. [S.l.], 2010. Disponível em: <<https://sci-hub.bz/https://dl.acm.org/citation.cfm?id=1807285CFID=1012694822CFTOKEN=34584215>>.
- [28] CORIA, S. R. et al. *MuniMex 1.0: a Basic Software Interface for Socio-Demographic Data of Mexican Municipalities*. [S.l.], 2014. Disponível em: <<https://sci-hub.bz/https://dl.acm.org/citation.cfm?id=2612782CFID=1012694822CFTOKEN=34584215>>.
- [29] SOFTWARE, G. *HTMLUNIT*. Disponível em: <<http://htmlunit.sourceforge.net/>>. Acesso em: Out. 18, 2017.
- [30] PRESSMAN, R. S. *Engenharia de Software*. Brasil: São Paulo: Pearson Makron Books, 1995.
- [31] PFLEEGER, S. L. *Engenharia de Software: Teoria e Prática*. Brasil: São Paulo: Pearson Prentice Hall Brasil, 2004.
- [32] SOMMERVILLE, I. *Engenharia de Software*. Brasil: São Paulo: Pearson Education Brasil, 2007.
- [33] JINO JOSÉ C. MALDONADO, M. E. D. M. *Introdução ao Teste de Software*. Brasil: São Paulo: Campus Elsevier, 2007.
- [34] KOSCIANSKI, A.; SOARES, M. dos S. *Qualidade de software : aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software*. Brasil: São Paulo: Novatec, 2006.
- [35] JUNIT. Disponível em: <<http://junit.org/junit5/>>. Acesso em: nov. 10, 2017.
- [36] DEITEL, H. D. P. *Java - Como Programar*. Brasil: Pearson Prentice Hall. 1144 p. ISBN 9788576055631.
- [37] CORPORATION, O. *JavaServer Faces Technology Overview*. 2015. Disponível em: <<http://www.oracle.com/technetwork/java/javaee/overview-140548.html>>. Acesso em: set. 03, 2017.
- [38] FOWLER, M. *Model View Controller*. 2015. Disponível em: <<http://martinfowler.com/eaaDev/uiArchs.html#ModelViewController>>. Acesso em: set. 07, 2017.
- [39] DOCUMENTATION. *BOOTSTRAP*. Disponível em: <<http://getbootstrap.com.br/>>. Acesso em: JUN. 05, 2017.

[40] DOCUMENTATION. *HIBERNATE*. Disponível em: <<http://hibernate.org/orm/documentation/5.2/>>. Acesso em: AGO. 21, 2017.

[41] SAURO, J. *Measuring Usability With The System Usability Scale (SUS)*. Disponível em: <<http://www.measuringusability.com/sus.php/>>. Acesso em: NOV. 22, 2017.

APÊNDICE A – TERMO DE CONSENTIMENTO



TERMO DE CONSENTIMENTO

Eu _____ declaro ter sido informado e concordo em participar de um teste de avaliação da interface web da ferramenta Open Smart City View, em participação voluntária, bem como a liberação das informações geradas a partir deste software, para fins acadêmicos e de pesquisa. Este trabalho está sendo desenvolvido pela aluna do Mestrado em Computação Aplicada da Universidade de Passo Fundo, Fernanda Rigo, orientado pelo Prof. Dr Roberto dos Santos Rabello.

Os dados e resultados individuais desta pesquisa estarão sempre sob sigilo ético, não sendo mencionados os nomes dos participantes em nenhuma apresentação oral ou trabalho escrito, que venha a ser publicado.

A participação nesta pesquisa não oferece risco ou prejuízo à pessoa participante. Se no decorrer da pesquisa o(a) participante resolver não mais continuar terá toda a liberdade de o fazer, sem que isso lhe acarrete qualquer prejuízo.

Assinatura

APÊNDICE B – QUESTIONÁRIO DE AVALIAÇÃO

Questionário de avaliação da OSCV

- A avaliação a ser aplicada é um projeto de conclusão no curso de mestrado em Computação Aplicada da UPF, de desenvolvimento da estudante Fernanda Rigo e orientação do prof. Dr. Roberto dos Santos Rabello.
- O trabalho proposto tem como objetivo avaliar a usabilidade da interface web da ferramenta Open Smart City View
- É importante ressaltar que, nesta avaliação o participante não será analisado pelo seu desempenho na execução das atividades.
- Os observadores estarão próximos para fazer possíveis anotações e observações necessárias.
- O tempo estimado é em média de 10 minutos para a sessão de avaliação.

Genero: Feminino Masculino

Idade: _____

Profissão: _____

1. Eu acho que gostaria de usar esse sistema com frequência.

Discordo Concordo
Fortemente 1 2 3 4 5 Fortemente

2. Eu acho que o sistema poderia ser menos complexo.

Discordo Concordo
Fortemente 1 2 3 4 5 Fortemente

3. Eu achei o sistema fácil de usar.

Discordo Concordo
Fortemente 1 2 3 4 5 Fortemente

4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.

Discordo Concordo
Fortemente 1 2 3 4 5 Fortemente

5. Eu acho que as várias funções do sistema estão muito bem integradas.

Discordo Concordo
Fortemente 1 2 3 4 5 Fortemente

6. Eu acho que o sistema apresenta muita inconsistência.

Discordo Concordo
Fortemente 1 2 3 4 5 Fortemente

7. Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.

Discordo Concordo
Fortemente 1 2 3 4 5 Fortemente

8. Eu achei o sistema atrapalhado de usar.

Discordo Concordo
Fortemente 1 2 3 4 5 Fortemente

9. Eu me senti confiante ao usar o sistema.

Discordo Concordo
Fortemente 1 2 3 4 5 Fortemente

10. Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

Discordo Concordo
Fortemente 1 2 3 4 5 Fortemente

Comentários: _____
