Pontifícia Universidade Católica
do Rio de Janeiro

Fernando Antonio Dantas Gomes Pinto

Extração e Triplificação de Atos Públicos

INF2102 - Projeto Final de Programação.

:

Rio de Janeiro
Julho 2020

Pontifícia Universidade Católica
do Rio de Janeiro

Fernando Antonio Dantas Gomes Pinto

Extração e Triplificação de Atos Públicos

INF2102 - Projeto Final de Programação.

Prof. Marcos Kalinowski
Departamento de Informática – PUC-Rio

Prof. Edward Hermann Haeusler
Departamento de Informática – PUC-Rio

Rio de Janeiro, Julho 8th, 2020

Sumário

Lista de Figuras

Lista de Tabelas

List of Algorithms

Lista de Códigos

, .

1
Introdução

(?) Dado o rápido desenvolvimento de novas tecnologias de apoio à informação, o volume de dados gerados tem também crescido de forma expressiva. Essa grande oferta de dados tem ocasionado uma revolução, nos meios tecnológicos e sociais, por novos modelos de gestão da informação. Um exemplo desse modelo é a participação da sociedade nas decisões governamentais com base na análise de dados públicos disponíveis da Internet. A participação social no acesso a estes dados já era prevista desde a constituição de 1988, mas limitado às ferramentas de TICs disponíveis a época. De acordo com a Constituição Federativa do Brasil (2), no seu Art. 5, inciso XXXIII, é dito que todo órgão público é obrigado a ceder informações geradas por suas atividades a indivíduo com interesse particular ou coletivo, sob pena de responsabilidade àqueles que não o cumprirem dentro dos prazos exigidos por lei, respeitando aquelas informações que sejam classificadas como imprescindível à segurança da sociedade e do Estado. Apenas em 2011, dada a grande demanda social por transparência nestes dados governamentais, o governo federal elaborou a Lei de Acesso à Informação (3), provocando debates sobre o tratamento destas informações. Hoje, com o decreto 7.724 de 2012 (BRASIL, 2012), que regulamenta a LAI, o governo federal exige que os entes governamentais publiquem na Internet um conjunto mínimo de informações governamentais. Sabe-se que muitas informações ainda não estão disponíveis para acesso online, por ainda não contar com um mecanismo que automatize o processo de extração de certas informação. Uma outra forma de promover transparência é a extração e publicação dos atos públicos contidos nos Diários Oficiais. A publicidade dos atos públicos em Diários Oficiais é interpretada como um ato legal obrigatório nas esferas públicas, além do fato motivador aos princípios legais da transparência.

Sendo assim, este trabalho pretende apresentar o *DO2RDF*, uma ferramenta que tem como proposta a abertura e transparência de dados públicos a partir dos atos publicados em diários oficiais. Geralmente, estes documentos são disponibilizados no formato PDF, um formato fechado, o que dificulta o manuseio das informações produzidas por entes públicos.

O escopo inicial deste projeto ficou restrito aos processos de movimentação de servidores em cargos comissionados, funções gratificadas e empregos de confiança, dos últimos 7 anos.

Esta ferramenta, além da proposta de tornar os atos públicos mais transparentes para a sociedade, está fortemente apoiada sobre os conceitos

da Web Semântica e dados conectados. Nosso objetivo é permitir que pessoas ou máquinas tenham a oportunidade de fazer consultas a estes conjuntos de dados de forma estrutura na web. E para isso, *DO2RDF* foi especificada e desenvolvida sobre as principais pilhas de protocolo da Web Semântica. Neste caso, a ferramenta expõe os dados no modelo*RDF/XML*.

O *RDF* (*Resource Description Framework*) é um modelo de dados (metadados) utilizado para representação de informação de modo a facilitar a busca por recursos na Web e que faz parte da pilha de protocolos definidos pela w3c para a Web Semântica, Figura 1.1.

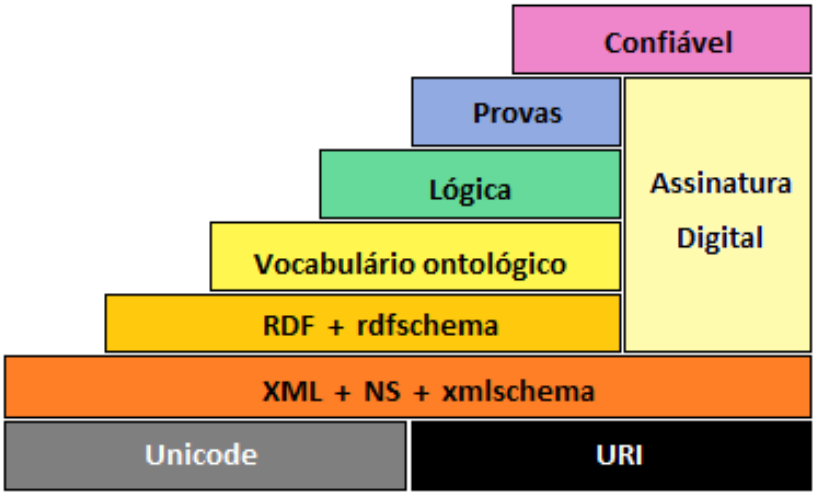


Figure 1.1: Camadas de protocolos da Web Semântica.

Segundo (5), *RDF* extende a estrutura de *Links* da Web ao usar *URIs* para nomear o relacionamento entre os recursos na web, disponíveis na forma de uma tripla de elementos <*sujeito*>, <*predicado*> e <*objeto*>. Com esse simples modelo, o *RDF* permite que dados estruturados e semiestruturados sejam combinados, expostos e compartilhados por diferentes tecnologias.

Esse modelo *RDF* forma um grafo (Grafo *RDF*) direcionado e rotulado onde as extremidades (nós) representam recursos (Sujeito ou Objeto) que são relacionados por um predicado (aresta) (6). Recursos podem ser vistos como qualquer informação disponível, por exemplo, um documento, uma pessoa, etc. Neste caso, a cada recurso é atribuido um elemento identificador *IRI* (*Internationalized Resource Identifier*).

Inicialmente, a proposta do padrão *RDF* não definiu formalmente uma semântica(7)(9). Hoje, para expressar a semântica através do modelo *RDF*, é necessário a definição de um complemento ao *RDF*. Algumas propostas foram desenvolvidas como, por exemplo, *RDF-Schema* (*RDF-S*). *RDF-S* é a especificação que define classes, propriedades e seus relacionamentos, que podem ser usados para descrever triplas (6). Na prática, o esquema *RDF* é uma camada para definição de ontologias. Esta ontologia é composta por duas partes: uma que define as relações entre classes de objetos e outra parte lógica, composta por um conjunto de regras de inferência para os dados.

Sendo assim, este trabalho está dividido nas seguintes seções: Capítulo 1, esta introdução, apresentamos a nossa motivação da pesquisa. O Capítulo 2, a proposta da pesquisa com maiores detalhes sobre a ferramenta *DO2RDF*. O Capítulo 3, apresentamos os resultados gerados pela ferramenta: artefatos e consultas *SPARQL* em uma base *RDF*, e finalmente o Capítulo 4, onde faremos uma breve conclusão da pesquisa e trabalhos futuros.

2
Proposta

Nossa proposta é a extração de dados de atos públicos publicados no Diário Oficial Municipal (DOM) da Prefeitura do Rio de Janeiro. O escopo ficou restrito às publicações que envolvem nomeações, exonerações e designações para cargos comissionados, funções gratificadas e empregos públicos (este último das empresas públicas), dos últimos 7 anos de governo. Dada a grande dificuldade de processamento de arquivos PDF, a ideia é montar uma base histórica de movimentação de pessoal para estes cargos de livre nomeação, o que além agilizar consultas dará maior transparência aos setores de auditoria e, consequentemente, a sociedade.

O primeiro desafio foi capturar todos os diários oficiais a partir de 1 º de janeiro de 2013, primeiro dia da gestão do atual prefeito. Para este atividade, e outras deste projeto, foram utilizadas a linguagem *Python*, versão 3, como *backend* da ferramenta de extração e geração das triplas *RDF*.

O código apresentado no código 1, representa o que foi desenvolvido como primeira etapa deste projeto. Utilizando o método *requests* do *Python* foram baixados e tratados 896 diários oficiais. Nesta etapa, foram descartados diários (edições normais e suplementares) com apenas capas de publicação.

Code 1: Método *Request* para download dos diários.

```
1 count = 0
2 while count < 5000:
3     print(count)
4     filename = Path('pdf/'+str(count)+'.'+'pdf')
5     print(filename)
6     url = 'http://doweb.rio.rj.gov.br/portal/edicoes/download/'+
          str(count)
7     print(url)
8     response = requests.get(url)
9     filename.write_bytes(response.content)
10    count = count + 1
```

De posse destes documentos, a segunda etapa foi construir o extrator de informações contidas em cada diário. Neste processo, foi utilizada a biblioteca *RE*, do Python, para implementar técnicas de extração de dados, com um conjunto de expressões regulares, de forma automática nos documentos PDF. Naturalmente, algoritmos de expressões regulares tendem a ter seu comportamento guloso e extrair dados, de documentos PDF, tornou-se uma atividade de grande custo, dado o grande número de padrões de publicações presentes nos

diários. Esta falta de padronização fez com que fosse definida uma arquitetura de sistema capaz de interpretar cada padrão de forma única, evitando assim sobreposição de padrões.

Para este projeto, como mencionado, foram tratadas apenas as informações de movimentação de pessoal nos cargos de livre nomeação. Cargos de livre nomeação são cargos que não necessitam de concurso público para seu efetivo exercício, cabendo a gestor a escolha dos profissionais que devem ocupar tal cargo. Inicialmente, estamos coletando as seguintes informações dos diários:

- Número do diário;
- Data do diário;
- Tipo do diário (normal ou suplementar);
- Governo;
- Número da portaria de publicação;
- Data da portaria de publicação;
- Gestor da portaria (secretaria);
- Data do efetivo ato;
- Tipo do ato (resolução, portaria ou decreto);
- Tipo da movimentação (nomear, exonerar, designar e dispensar);
- Matrícula;
- Nome do servidor;
- Cargo Efetivo;
- Cargo Comissionado;
- Função Gratificada;
- Tipo do cargo.

Definido estes objetos de extração, foi desenvolvido os extratores de padrão utilizando técnicas de expressões regulares aplicadas a cada documento PDF. Como podemos ver no Código 2, para o caso do tipo de movimentação "Dispensar", atribuído às "funções gratificadas" e "empregos de confiança", foram identificados 8 (oito) formas diferentes de formalizar o ato no diário, consequentemente definimos também 8 (oito) padrões de expressões.

Até o presente momento da escrita deste trabalho, foram identificados 33 (trinta e três) diferentes padrões de atos.

Code 2: *Scripts* com expressões regulares para atos "Dispensar".

```
1       def atos_dispensar(self, buffer_local, Detalhe):
2           servidor = []
3           dispensar_pattern1 = re.compile(u'[\.|\s]*Dispensar[,|\s
                ]*a\s*pedido[,|\s]*(?P<nome>[A-ZÉÁÍÓÚÇÃÊÔÕÀÜ\s]+)[,|\
                s]*(?P<cargoEfetivo>[A-ZÉÁÍÓÚÇÃÊÔÕÀÜa-záêéóíçãâôú\-\s
                ]+),\s*matrícula\s*(?P<matricula>[0-9\./-]+),\s*com\s
                *eficácia\s*a\s*contar\s*de\s*(?P<dia>[0-9]+).*\s*de\
                s*(?P<mes>[J|j]aneiro|[F|f]evereiro|[M|m]arço|[A|a]
                bril|[M|m]aio|[J|j]unho|[J|j]ulho|[A|a]gosto|[S|s]
                etembro|[O|o]utubro|[N|n]ovembro|[D|d]ezembro)\s*de\s
                *(?P<ano>[0-9]+),\s*da\s*[F|f]unção\s*[G|g]ratificada
                \s*(de)*\s*(?P<cargo>[A-ZÉÁÍÓÚÇÃÊÔÕÀÜa-záêéóíçãâôú\-\
                s]+),\s*símbolo\s*(?P<simbolo>[A-Z\-0-9\/\s]+),')
4           dispensar_pattern2 = re.compile(u'[\.|\s]*Dispensar[,|\s
                ]*a\s*pedido[,|\s]*(?P<nome>[A-ZÉÁÍÓÚÇÃÊÔÕÀÜ\s]+)[,|\
                s]*(?P<cargoEfetivo>[A-ZÉÁÍÓÚÇÃÊÔÕÀÜa-záêéóíçãâôú\-\s
                ]+),\s*matrícula\s*(?P<matricula>[0-9\.\/-]+),\s*da\s
                *[F|f]unção\s*[G|g]ratificada\s*de\s*(?P<cargo>[A-ZÉÁ
                ÍÓÚÇÃÊÔÕÀÜa-záêéóíçãâôú\-\s]+),\s*símbolo\s*(?P<
                simbolo>[A-Z\-0-9\/\s]+),')
5           dispensar_pattern3 = re.compile(u'[\.|\s]*Dispensar[,|\s
                ]*(?P<nome>[A-ZÉÁÍÓÚÇÃÊÔÕÀÜ\s]+),*\s*matrícula\s*(?P<
                matricula>[0-9\.\/-]+),\s*da\s*[F|f]unção\s*de\s*(?P<
                cargo>[A-ZÉÁÍÓÚÇÃÊÔÕÀÜa-záêéóíçãâôú\-\s]+),*\s*sí
                mbolo\s*(?P<simbolo>[A-Z\-0-9\s\/]+),')
6           dispensar_pattern4 = re.compile(u'[\.|\s]*Dispensar[,|\s
                ]*(?P<nome>[A-ZÉÁÍÓÚÇÃÊÔÕÀÜ\s]+)[\s|,]+com[\s|,]+
                validade[\s|,]+de[\s|,]+(?P<dia>[0-9]+).*\s*de\s*(?P<
                mes>[J|j]aneiro|[F|f]evereiro|[M|m]arço|[A|a]bril|[M|
                m]aio|[J|j]unho|[J|j]ulho|[A|a]gosto|[S|s]etembro|[O|
                o]utubro|[N|n]ovembro|[D|d]ezembro)\s*de\s*(?P<ano
                >[0-9]+)[\s|,]+(?P<cargoEfetivo>[A-ZÉÁÍÓÚÇÃÊÔÕÀÜa-záê
                éóíçãâôú\-\s]+)[\s|,]+matrícula\s*(?P<matricula
                >[0-9\.\/-]+)[\s|,]+da\s*[F|f]unção\s*[G|g]ratificada
                \s*(de)*\s*(?P<cargo>[A-ZÉÁÍÓÚÇÃÊÔÕÀÜa-záêéóíçãâôú\-\
                s]+)[\s|,]+símbolo[\s|,]+(?P<simbolo>[A-Z\-0-9\/\s]+)
                ,')
7           dispensar_pattern5 = re.compile(u'[\.|\s]*Dispensar[,|\s
                ]*(?P<nome>[A-ZÉÁÍÓÚÇÃÊÔÕÀÜ\s]+)[\s|,]+(?P<
                cargoEfetivo>[A-ZÉÁÍÓÚÇÃÊÔÕÀÜa-záêéóíçãâôú\-\s]+)[\s
                |,]+matrícula\s*(?P<matricula>[0-9\.\/-]+)[\s|,]*[\s
                |,]*com\s*eficácia\s*a\s*contar\s*de\s*(?P<dia
                >[0-9]+).*\s*de\s*(?P<mes>[J|j]aneiro|[F|f]evereiro|[
                M|m]arço|[A|a]bril|[M|m]aio|[J|j]unho|[J|j]ulho|[A|a]
                gosto|[S|s]etembro|[O|o]utubro|[N|n]ovembro|[D|d]
```

```
        ezembro)\s*de\s*(?P<ano>[0-9]+)[\s|,]+da\s*[F|f]unção
        \s*[G|g]ratificada\s*(de)*\s*(?P<cargo>[A-ZÉÁÍÓÚÇÃÊÔÕ
        ÀÜa-záêéóíçãâôú\-\s]+)[\s|,]+sím[-|\n]*bolo[\s|,]+(?P
        <simbolo>[A-Z\-0-9\/\s]+),')
8       dispensar_pattern6 = re.compile(u'[\.|\s]*Dispensar[,|\s
        ]*(?P<nome>[A-ZÉÁÍÓÚÇÃÊÔÕÀÜ\s]+)[,|\s]*matrícula\s*(?
        P<matricula>[0-9\.\/-]+)[,|\s]*da\s*[F|f]unção\s*de\s
        *[C|c]onfiança\s*de\s*(?P<cargo>[A-ZÉÁÍÓÚÇÃÊÔÕÀÜa-záê
        éóíçãâôú\-\s]+)[,|\s]*código\s*(?P<simbolo>[A-Z\-0-9\
        s\/]+),')
9       dispensar_pattern7 = re.compile(u'[\.|\s]*Dispensar[,|\s
        ]*(?P<nome>[A-ZÉÁÍÓÚÇÃÊÔÕÀÜ\s]+)[,|\s]*registro\s*(N|
        n)*.\s*[0-9]+[,|\s]*com(\s)*(a)*(\s)*validade(\s)*(a)
        *(\s)*partir(\s)*de(\s)*(?P<dia>[0-9]+).*(\s)*de(\s)
        *(?P<mes>[J|j]aneiro|[F|f]evereiro|[M|m]arço|[A|a]
        bril|[M|m]aio|[J|j]unho|[J|j]ulho|[A|a]gosto|[S|s]
        etembro|[O|o]utubro|[N|n]ovembro|[D|d]ezembro)\s*de\s
        *(?P<ano>[0-9]+)[,|\s]*do\s*[E|e]mprego\s*de\s*[C|c]
        onfiança\s*de\s*(?P<cargo>[A-ZÉÁÍÓÚÇÃÊÔÕÀÜa-záêéóíçãâ
        ôú\-\s]+)[,|\s]*[C|c]*ategoria\s*(?P<simbolo>[A-Z
        -0-9\ â  \-\/\s]+),')
10      dispensar_pattern8 = re.compile(u'[\.|\s]*Dispensar[,|\s
        ]*(?P<nome>[A-ZÉÁÍÓÚÇÃÊÔÕÀÜ\s]+)[,|\s]*(?P<
        cargoEfetivo>[A-ZÉÁÍÓÚÇÃÊÔÕÀÜa-záêéóíçãâôú\-\s]+)[,|\
        s]+matrícula\s*(?P<matricula>[0-9\ â  \.\/-]+)[,|\s]*
        com\s*a*\s*validade\s*a\s*partir\s*de\s*(?P<dia
        >[0-9]+).*\s*de\s*(?P<mes>[J|j]aneiro|[F|f]evereiro|[
        M|m]arço|[A|a]bril|[M|m]aio|[J|j]unho|[J|j]ulho|[A|a]
        gosto|[S|s]etembro|[O|o]utubro|[N|n]ovem[-|\n]*bro|[D
        |d]ezembro)\s*de\s*(?P<ano>[0-9]+)[,|\s]*da\s*[F|f]un
        ção\s*[G|g]ratificada\s*(de)*\s*(?P<cargo>[A-ZÉÁÍÓÚÇÃ
        ÊÔÕÀÜa-záêéóíçãâôú\-\s]+),\s*(código)*\s*[0-9]*[,|\s
        ]*[S|s]ím[-|\n]*bolo[,|\s]*(?P<simbolo>[A-Z\ â
        \-0-9\/\s]+),')
11
12      disp1 = dispensar_pattern1.search(buffer_local)
13      disp2 = dispensar_pattern2.search(buffer_local)
14      disp3 = dispensar_pattern3.search(buffer_local)
15      disp4 = dispensar_pattern4.search(buffer_local)
16      disp5 = dispensar_pattern5.search(buffer_local)
17      disp6 = dispensar_pattern6.search(buffer_local)
18      disp7 = dispensar_pattern7.search(buffer_local)
19      disp8 = dispensar_pattern8.search(buffer_local)
```

Como resultado da execução deste algoritmo, até o presente momento, foram extraídos 14.826 atos públicos. Para facilitar o processo de análise

dos dados extraídos foi gerado um arquivo de auditória (Figura 2.1) com as principais afirmações processadas. Este passo é necessário para garantir a qualidade da informação antes de ser inserida no banco de dados. Em em segundo momento, quando a ferramenta já estiver homologada este passo poderá ser reavaliado.

```
(PUC-RIO/TECMF)   ::PROCESSAMENTO DO DIÁRIO:: ANO: 31 No.: 000226 TIPO: NORMAL      * RIO DE JANEIRO * ARQUIVO: 3672.PDF SEQ.: 0010

(PADRAO 2.3)  RESOLUÇÕES  2284 21/02/2018 DESIGNAR  10/116630-5  REGINA COELI VIEIRA FERNANDES                    01/02/2018 COORDENADOR PEDAGÓGICO
(PADRAO 2.3)  RESOLUÇÕES  2287 21/02/2018 DESIGNAR  10/231922-6  SABRINA MARTINS DA ROCHA                         09/01/2018 DIRETOR-ADJUNTO
(PADRAO 2.3)  RESOLUÇÕES  2290 21/02/2018 DESIGNAR  10/266309-4  JACQUELINE CARDELLY JOORIS                       01/02/2018 COORDENADOR PEDAGÓGICO
(PADRAO 2.3)  RESOLUÇÕES  2292 21/02/2018 DESIGNAR  10/264537-2  CAROLINE GONZALEZ VIVAS                          01/02/2018 COORDENADOR PEDAGÓGICO
(PADRAO 2.3)  RESOLUÇÕES  2293 21/02/2018 DESIGNAR  10/177378-7  ANA LÚCIA OLIVEIRA DO COUTO                      01/02/2018 COORDENADOR PEDAGÓGICO
(PADRAO 2.3)  RESOLUÇÕES  2294 21/02/2018 DESIGNAR  10/241890-3  MÔNICA JANETE CARVALHO TEIXEIRA                  01/02/2018 COORDENADOR PEDAGÓGICO
(PADRAO 2.3)  RESOLUÇÕES  2302 21/02/2018 DESIGNAR  10/286152-4  FERNANDA DA SILVA CALDEIRA VASCONCELOS           01/02/2018 COORDENADOR PEDAGÓGICO
(PADRAO 2.3)  RESOLUÇÕES  2304 21/02/2018 DESIGNAR  10/171387-4  ANDRÉA MARIA RANGEL MARQUES                      01/02/2018 COORDENADOR PEDAGÓGICO
(PADRAO 2.3)  RESOLUÇÕES  2305 21/02/2018 DESIGNAR  10/299736-9  CARLA AMORIM BIDIRÁ ALVES                        01/02/2018 COORDENADOR PEDAGÓGICO
(PADRAO 2.3)  RESOLUÇÕES  2306 21/02/2018 DESIGNAR  10/275922-3  JOICE MIRANDA TOLENTINO MENDES                   01/02/2018 COORDENADOR PEDAGÓGICO
(PADRAO 2.4)  RESOLUÇÕES  2285 21/02/2018 DESIGNAR  10/259392-9  TERESA CRISTINA BORGES FERREIRA                  XX/XX/XXXX COORDENADOR PEDAGÓGICO
(PADRAO 2.4)  RESOLUÇÕES  2286 21/02/2018 DESIGNAR  10/285513-8  ERNANE CARRANO JANN                              XX/XX/XXXX COORDENADOR PEDAGÓGICO
(PADRAO 2.4)  RESOLUÇÕES  2289 21/02/2018 DESIGNAR  10/260465-0  ALESSANDRA RIBEIRO CAMPOS DE LIMA                XX/XX/XXXX COORDENADOR PEDAGÓGICO
(PADRAO 2.4)  RESOLUÇÕES  2298 21/02/2018 DESIGNAR  10/127986-8  LÚCIA SOBRAL ROCHA SILVEIRA                       XX/XX/XXXX COORDENADOR PEDAGÓGICO
(PADRAO 2.6)  RESOLUÇÕES  2281 21/02/2018 DISPENSAR 12/235713-5  SILVIA DE JESUS PEREIRA                          XX/XX/XXXX COORDENADOR PEDAGÓGICO
(PADRAO 2.6)  RESOLUÇÕES  2282 21/02/2018 DISPENSAR 12/172270-1  SONIA MARIA DA SILVA MÁXIMO                      XX/XX/XXXX DIRETOR-ADJUNTO

(PUC-RIO/TECMF)   ::PROCESSAMENTO DO DIÁRIO:: ANO: 33 No.: 000079 TIPO: NORMAL      * RIO DE JANEIRO * ARQUIVO: 4207.PDF SEQ.: 0011


(PUC-RIO/TECMF)   ::PROCESSAMENTO DO DIÁRIO:: ANO: 32 No.: 000007 TIPO: NORMAL      * RIO DE JANEIRO * ARQUIVO: 3707.PDF SEQ.: 0012

(PADRAO 1.16) RESOLUÇÃO   1825 22/03/2018 EXONERAR  60/210987-4  MARCELO MOREIRA PESSOA                           19/03/2018 DIRETOR I
(PADRAO 1.4)  RESOLUÇÃO   1827 22/03/2018 NOMEAR    60/260220-9  CLÁUDIO MARCIO FERREIRA DA SILVA                 19/03/2018 GERENTE I
(PADRAO 1.16) RESOLUÇÃO   1828 22/03/2018 EXONERAR  60/260220-9  CLÁUDIO MARCIO FERREIRA DA SILVA                 19/03/2018 ASSESSOR I
(PADRAO 1.3)  RESOLUÇÃO   1830 22/03/2018 NOMEAR    XXXXXXXXXXX  RONALDO BRAZIL PEREIRA                           20/02/2018 GERENTE II
(PADRAO 1.4)  RESOLUÇÃO   1831 22/03/2018 NOMEAR    60/292047-8  MANUELITO DE SOUSA REIS JUNIOR                   01/02/2018 ASSISTENTE I
(PADRAO 1.18) RESOLUÇÃO   1835 22/03/2018 EXONERAR  11/215102-5  ELIAS SILVA DE OLIVEIRA                          12/03/2018 COORDENADOR II
(PADRAO 1.18) RESOLUÇÃO   1837 22/03/2018 EXONERAR  11/237115-1  CARMEM LUCIA PEREIRA LOPES                       14/03/2018 GERENTE II
```

Figure 2.1: Porção do arquivo de auditoria com as informações extraídas dos diários.

Um ponto a se observar no arquivo de auditoria é a qualidade das informações publicadas nos diários. Muitas datas estavam com ausência de informação e poucos casos com problemas na sua tipografia. Assim, para este momento do projeto, foi definido uma data "fictícia" para compor esta informação. No código 3 apresenta uma expressão regular que trata esta informação.

Code 3: Expressão regular para tratar datas ausentes ou inconsistentes

```
1    def converteData(sef, dates):
2          date_pattern = re.compile(u'(?P<data>[X]+|00/|0000)')
3          date = date_pattern.search(dates)
4          if (date):
5            vlr = '01/01/1900'
6          else: vlr = dates
7          return vlr
```

Para resumir esta atividade a Tabela 2.1 apresenta uma síntese da movimentação de pessoal na prefeitura do Rio de Janeiro:

Quantidade	**Tipo da ação**
5.311	Nomear
1.994	Designar
5.698	Exonerar
1.823	Dispensar

Table 2.1: Quantitativo de tipos de atos extraídos nos diários.

3
Documentação e Testes

Neste capítulo iremos apresentar a documentação do DO2RDF com base em artefatos UML (*Unified Modeling Language*), concentrados nos diagramas de classe, módulo e componentes. Na sequência iremos apresentar um plano de testes e sua execução para validar a qualidade do modelo proposto.

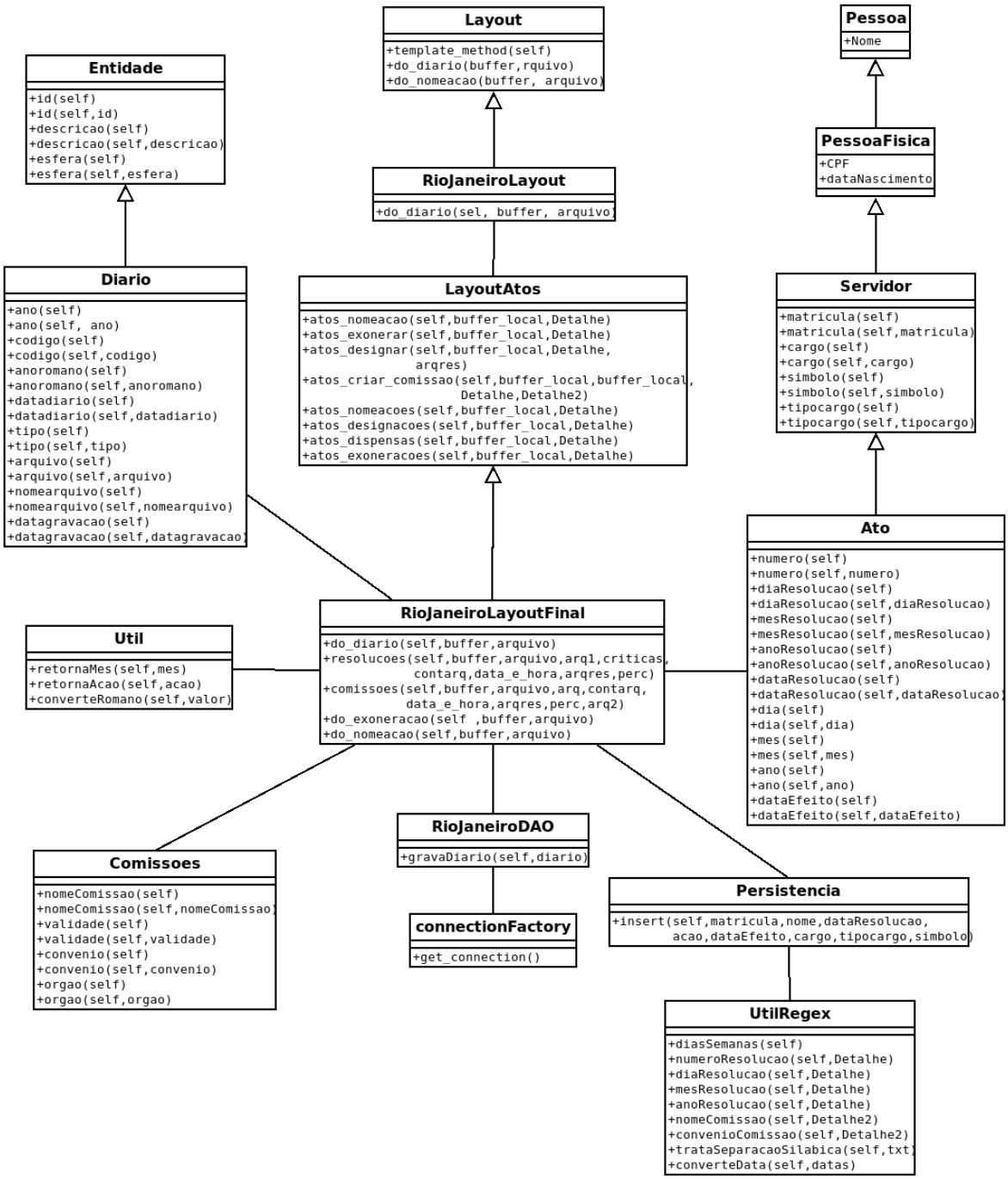


Figure 3.1: Diagrama de Classes do DO2RDF.

4
Resultados

Neste capítulo iremos apresentar os resultados do processo de extração das informações contidas nos diários. Neste etapa, iremos apresentar o passo de triplificação em dados *RDF/XML* para ser executado em um ambiente de consultas *SPARQL*.

Como apresentando na seção anterior, de posse das informações previamente verificadas, foi realizada uma carga em um banco de dados *PostgreSQL* versão 12.1 em ambiente Ubuntu GNU/Linux. Estas informações são então tratadas por um módulo que normaliza os dados. A tabela de dados ficou assim organizada e apresentada na Figura 4.1:

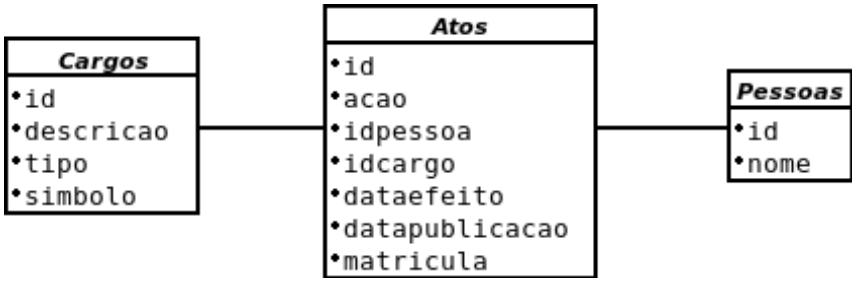


Figure 4.1: Modelo do banco de dados.

Uma observação é que para cada ato de nomeação, por exemplo, uma pessoa poderá ter uma nova matrícula associada. Este comportamento motivou esta modelagem das informações sobre matrículas. Quanto as outras informações não persistidas no banco é justificado por ser um ambiente de desenvolvimento e testes. Para uma versão final, em um ambiente com maior disponibilidade física de hardware, as demais informações farão parte do esquema.

Com o banco de dados devidamente "populado", iniciamos o processo de triplificação das informações. Para esta etapa foi utilizada a biblioteca *RDFLib* (8) para *Python*. A biblioteca possui interfaces que tornam simples e facilitam a implementação dos nós *RDF*. Como opção, ela inclui analisadores e serializadores para *RDF/XML*, *N3*, *NTriples*, *N-Quads*, *Turtle*, *TriX*, *RDFa* e *Microdata*. Ela implementa uma interface *Graph* ao qual podemos armazenar informações dos grafos em memória ou de armazenamento persistente. Também com ela é possível executar consultas e atualizações *SPARQL*. Neste trabalho optamos por fazer a serialização dos dados no formato *RDF/XML* pois os demais formatos (*N3*, *NTriples* e *Turtle*) não foram possíveis de carga no ambiente do *AllegroGraph*. Um passo importante do projeto foi quanto a ontologia

a ser utilizada na definição dos significados aos conteúdos do Diário Oficial. Para este momento, como prova básica de conceito, optamos por definir uma ontologia adaptada e genérica em FOAF (4).

O código 4 apresenta kernel do módulo de serialização dos dados do diário oficial e no código 5 podemos observar uma porção do arquivo *DO2RDF.rdf*, resultado do processo de serialização do *RDFLib*, para carga no *AllegroGraph*.

Code 4: Processo de serialização em *RDF/XML*

```
1  for row in funcionarios_records:
2
3      seqa+=1
4      idP = seqa
5      idP = BNode()
6
7      store.add((idP, RDF.type, FOAF.Funcionarios))
8      store.add((idP, FOAF.matricula, Literal(row[0].strip())))
9      store.add((idP, FOAF.nome, Literal(row[1].strip())))
10     store.add((idP, FOAF.dataPublicacao, Literal(row[2])))
11     store.add((idP, FOAF.acao, Literal(row[3].strip())))
12     store.add((idP, FOAF.dataEfeito, Literal(row[4])))
13     store.add((idP, FOAF.cargo, Literal(row[5].strip())))
14     store.add((idP, FOAF.tipoCargo, Literal(row[6].strip())))
15     store.add((idP, FOAF.simbolo, Literal(row[7].strip())))
16
17     # Serialize the store as RDF/XML to the file DO2RDF.rdf.
18     store.serialize("RDF/DO2RDF.rdf", format="pretty-xml",
           max_depth=3)
19     print('RDF Serializations:', seqa, 'De', size)
```

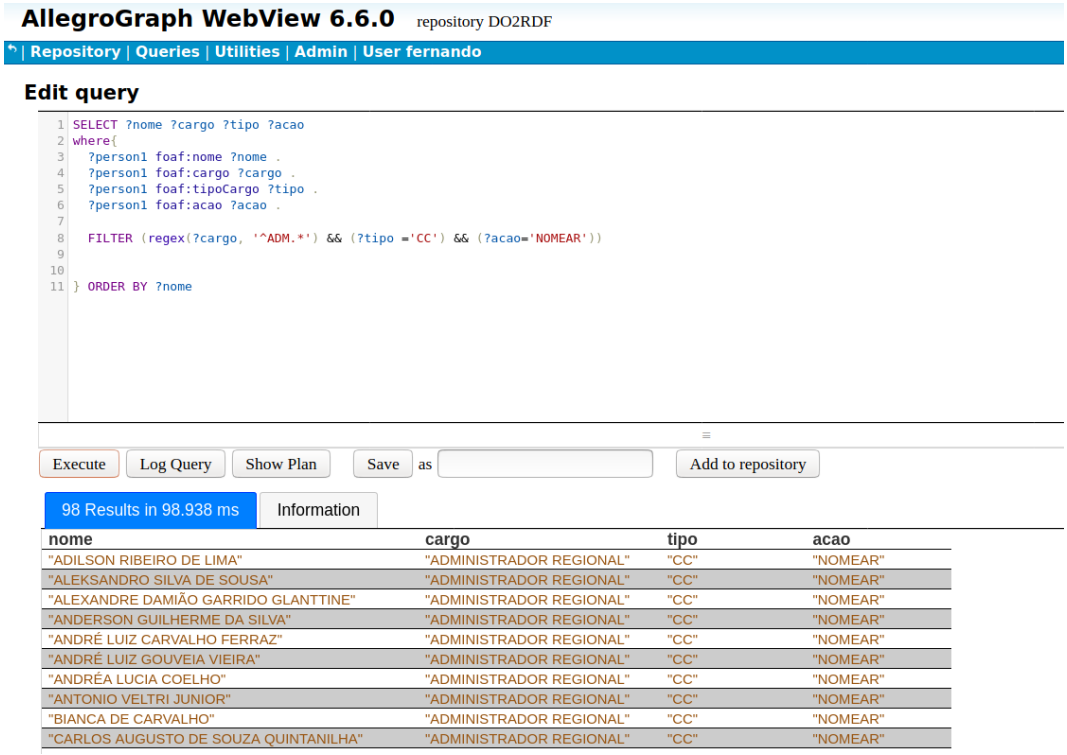
Nesta etapa, o módulo gerador apresentou lentidão para a serialização dos 14.826 registros de atos. Sua completa geração está levando em torno de 26 horas neste ambiente virtual (Intel core i7, 8 GB de memória RAM). A solução foi particionar sua execução em lotes de 1.000 registros, que levou pouco mais de 2 horas no total.

Code 5: Processo de serialização em *RDF/XML*

```
1
2 <?xml version="1.0" encoding="utf-8"?>
3 <rdf:RDF
4   xmlns:foaf="http://xmlns.com/foaf/0.1/"
5   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
6 >
7   <foaf:Funcionario rdf:nodeID="
        N7654fa3513984ea89c0ade256a971939">
8     <foaf:acao>NOMEAR</foaf:acao>
9     <foaf:dataPublicacao>2017-01-19</foaf:dataPublicacao>
10    <foaf:tipoCargo>CC</foaf:tipoCargo>
11    <foaf:simbolo>DAS-06</foaf:simbolo>
12    <foaf:dataEfeito>2017-01-01</foaf:dataEfeito>
13    <foaf:matricula>60/210917-1</foaf:matricula>
14    <foaf:cargo>ASSISTENTE I</foaf:cargo>
15    <foaf:nome>MARIA CRISTINA DOS SANTOS BASTOS</foaf:nome>
16  </foaf:Funcionario>
17  <foaf:Funcionario rdf:nodeID="
        Nde2047713a3f431e91f6f2c4471025ee">
18    <foaf:simbolo>DAS-10</foaf:simbolo>
19    <foaf:dataEfeito>1900-01-01</foaf:dataEfeito>
20    <foaf:nome>RICARDO VIEIRA SILVA</foaf:nome>
21    <foaf:cargo>SUBSECRETÁRIO</foaf:cargo>
22    <foaf:matricula>11/087325-7</foaf:matricula>
23    <foaf:acao>NOMEAR</foaf:acao>
24    <foaf:dataPublicacao>2019-02-21</foaf:dataPublicacao>
25    <foaf:tipoCargo>CC</foaf:tipoCargo>
26  </foaf:Funcionario>
```

Para executar as consultas *SPARQL* foi instalado o *AllegroGraph* na versão 6.6.0 (1), em ambiente virtual com Ubuntu GNU/Linux. Na sequência, foi criado o repositório *DO2RDF* para receber a carga do arquivo *DR2RDF.rdf*.

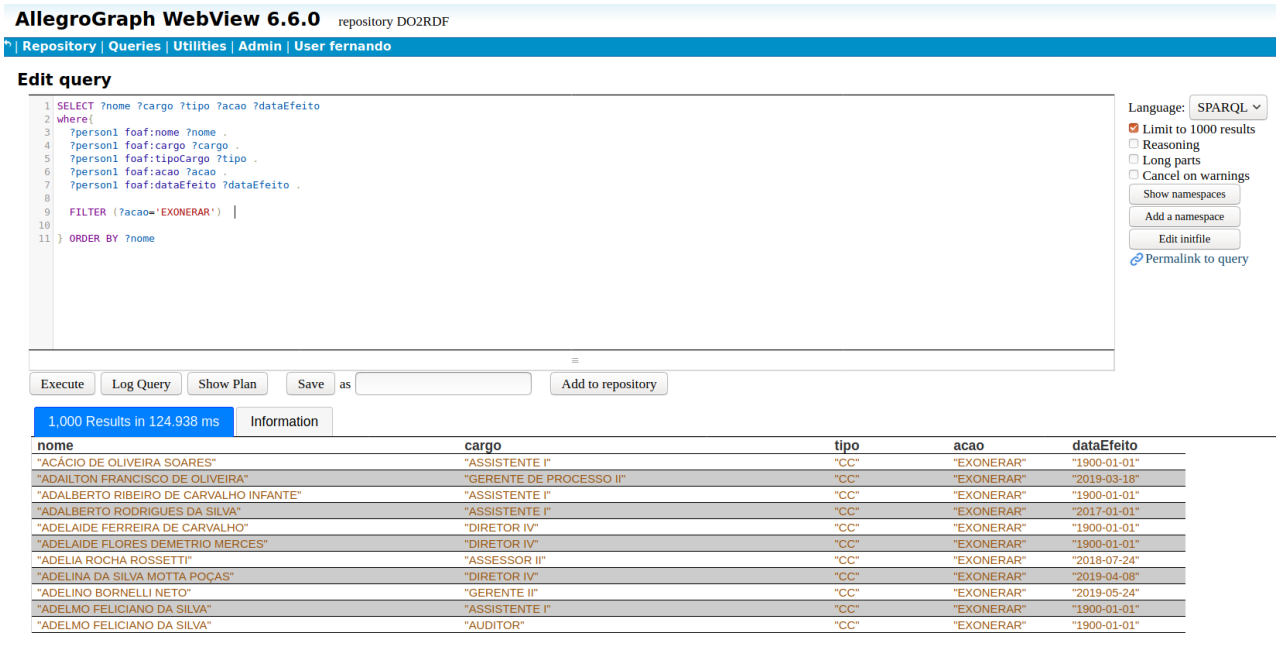
Podemos observar na Figura 4.2 o resultado de uma simples consulta *SPARQL* para os casos de nomeação para todos os cargos que iniciam com a expressão "ADM".



nome	cargo	tipo	acao
"ADILSON RIBEIRO DE LIMA"	"ADMINISTRADOR REGIONAL"	"CC"	"NOMEAR"
"ALEKSANDRO SILVA DE SOUSA"	"ADMINISTRADOR REGIONAL"	"CC"	"NOMEAR"
"ALEXANDRE DAMIÃO GARRIDO GLANTTINE"	"ADMINISTRADOR REGIONAL"	"CC"	"NOMEAR"
"ANDERSON GUILHERME DA SILVA"	"ADMINISTRADOR REGIONAL"	"CC"	"NOMEAR"
"ANDRÉ LUIZ CARVALHO FERRAZ"	"ADMINISTRADOR REGIONAL"	"CC"	"NOMEAR"
"ANDRÉ LUIZ GOUVEIA VIEIRA"	"ADMINISTRADOR REGIONAL"	"CC"	"NOMEAR"
"ANDRÉA LUCIA COELHO"	"ADMINISTRADOR REGIONAL"	"CC"	"NOMEAR"
"ANTONIO VELTRI JUNIOR"	"ADMINISTRADOR REGIONAL"	"CC"	"NOMEAR"
"BIANCA DE CARVALHO"	"ADMINISTRADOR REGIONAL"	"CC"	"NOMEAR"
"CARLOS AUGUSTO DE SOUZA QUINTANILHA"	"ADMINISTRADOR REGIONAL"	"CC"	"NOMEAR"

Figure 4.2: Consulta das nomeações para cargos com iniciais "ADM".

Para concluir, na Figura 4.2, podemos observar o resultado de uma consulta para os casos todas os registros de exonerações.



nome	cargo	tipo	acao	dataEfeito
"ACÁCIO DE OLIVEIRA SOARES"	"ASSISTENTE I"	"CC"	"EXONERAR"	"1900-01-01"
"ADAILTON FRANCISCO DE OLIVEIRA"	"GERENTE DE PROCESSO II"	"CC"	"EXONERAR"	"2019-03-18"
"ADALBERTO RIBEIRO DE CARVALHO INFANTE"	"ASSISTENTE I"	"CC"	"EXONERAR"	"1900-01-01"
"ADALBERTO RODRIGUES DA SILVA"	"ASSISTENTE I"	"CC"	"EXONERAR"	"2017-01-01"
"ADELAIDE FERREIRA DE CARVALHO"	"DIRETOR IV"	"CC"	"EXONERAR"	"1900-01-01"
"ADELAIDE FLORES DEMETRIO MERCES"	"DIRETOR IV"	"CC"	"EXONERAR"	"1900-01-01"
"ADELIA ROCHA ROSSETTI"	"ASSESSOR II"	"CC"	"EXONERAR"	"2018-07-24"
"ADELINA DA SILVA MOTTA POÇAS"	"DIRETOR IV"	"CC"	"EXONERAR"	"2019-04-08"
"ADELINO BORNELLI NETO"	"GERENTE II"	"CC"	"EXONERAR"	"2019-05-24"
"ADELMO FELICIANO DA SILVA"	"ASSISTENTE I"	"CC"	"EXONERAR"	"1900-01-01"
"ADELMO FELICIANO DA SILVA"	"AUDITOR"	"CC"	"EXONERAR"	"1900-01-01"

Figure 4.3: Consulta de exonerações.

Finalmente podemos apresentar a arquitetura da ferramenta *DO2RDF* (Figura 4.4), onde

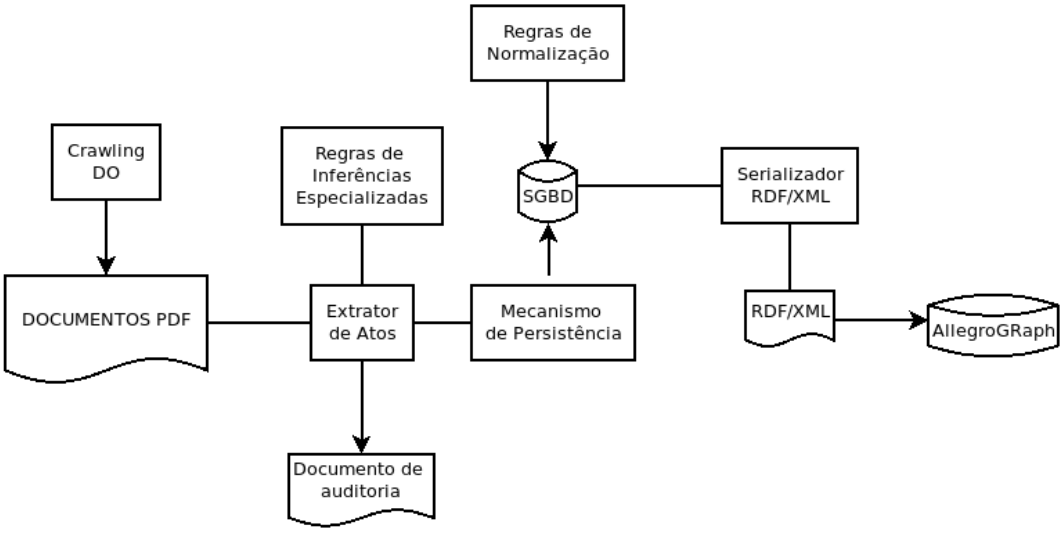


Figure 4.4: Arquitetura da ferramenta *DO2RDF*.

os módulos ficam assim representados:

- **Módulo de *Crawling*** - É responsável por recuperar os Diários Oficiais, de forma automática, do repositório da prefeitura do Rio de Janeiro.
- **Módulo Extrator de Atos** - Conjunto de regras que utiliza expressões regulares para extrair conteúdos dos diários oficiais. Também é responsável por um processo de geração de arquivos de auditoria.
- **Módulo de Regras de Inferências Especializadas** - Módulo que contém as regras gramaticais para cada ente público (governos estaduais, municipais e federal). Opera em conjunto com o Módulo Extrator de Atos.
- **Mecanismo de Persistência de dados** - Responsável por manter uma interface de gravação dos dados extraídos com o Banco de Dados *PostgreSQl*.
- **Módulo de Regras de Normalização** - Responsável pela leitura e aplicação de regras de normalização nos dados previamente armazenados.
- **Módulo de Serializador** - Este módulo faz a leitura do banco de dados e serialização *RDF*. Neste trabalho optamos pelo formato *RDF/XML*.

5
Conclusões e trabalhos futuros

Apresentamos como a transparência pública poderá ser dada por processos automatizados de leitura e extrações de informações de atos públicos disponíveis em Diários Oficiais. De posse dos diários, e definido uma arquitetura de estilos de padrões, o processo de extração utilizando expressões regulares apresentou-se de forma simples e eficiente.

Um outro ponto a se observar foi quanto a utilização da biblioteca *RDFLib* que apresentou ser pouco flexível para criação de ontologias genéricas. Inicialmente tentamos modificar seus *scripts* para que fossem incorporados novos conceitos. Uma solução mais simples foi adaptar a ontologia *FOAF* para nossas necessidades.

Superada esta parte, a serialização em formato *RDF/XML* se tornou bastante eficiente para o propósito da pesquisa. Consultas *SPARQL* foram realizadas para demonstrar a razoabilidade da ferramenta.

Como trabalhos futuros, podemos extrair outras informações públicas como pagamento de diárias, licitações, pagamento de custeios, participações de servidores em comissões remuneradas, etc. Todas estas informações são passíveis de extração bastando definir seus padrões de publicação na forma de expressões regulares. Um outro ponto a ser explorado é a definição de uma ontologia para o diário oficial. Neste trabalho foi utilizado um modelo genérico de ontologia para fins de execução da posposta. Para uma atividade fim, faz-se necessário definir o significado de cada dado presente no modelo *RDF*. Isso facilitará o processo de captura e processamento dos dados armazenados nas triplas *RDF*, por máquinas ou usuários, através de consultas *SPARQL*.

6
Referências bibliográficas

[1] ALLEGROGRAPH. **Allegrograph - The Enterprise Knowledge Graph**. https://allegrograph.com. Acessado: 11-12-2019. 4

[2] BRASIL. **Emenda Constitucional nº 9, de 9 de novembro de 1995**. Diário Oficial [da] República Federativa do Brasil, 59:1966. 1

[3] BRASIL. **Lei nº. 12.527. Lei de Acesso à Informação.** Diário Oficial [da] República Federativa do Brasil. 1

[4] BRICKLEY, D.; MILLER, L.. **FOAF Vocabulary Specification 0.99**. http://www.foaf-project.org/, 2014. Acessado: 03-12-2019. 4

[5] GROUP, R. W.. **Resource Description Framework (RDF)**. https://www.w3.org/RDF/, 2014. Acessado: 11-10-2019. 1

[6] ISOTANI, S.; BITTENCOURT, I.. **Dados Abertos Conectados: em Busca da Web do Conhecimento**. 08 2015. 1

[7] LASSILA, O.; SWICK, R. R.. **Resource Description Framework (RDF) Model and Syntax Specification**. W3c recommendation, W3C, February 1999. 1

[8] TEAM, R.. **RDFLib**. https://rdflib.readthedocs.io/en/stable/#, 2013. Acessado: 13-11-2019. 4

[9] YANG, G.; KIFER, M.. **On the Semantics of Anonymous Identity and Reification**. In: ON THE MOVE TO MEANINGFUL INTERNET SYSTEMS, 2002 - DOA/COOPIS/ODBASE 2002 CONFEDERATED INTERNATIONAL CONFERENCES DOA, COOPIS AND ODBASE 2002, p. 1047–1066, Berlin, Heidelberg, 2002. Springer-Verlag. 1