

**UNIVERSIDADE FEDERAL DE SANTA CATARINA – UFSC  
CENTRO TECNOLÓGICO – CTC  
CURSO DE SISTEMAS DE INFORMAÇÃO**

**MARCELO MENDONÇA SCHEIDT**

**FERRAMENTA PARA EXTRAÇÃO DE WEBTABLES E CRIAÇÃO  
DE SCRIPTS SQL**

Florianópolis – SC, 2013.

**MARCELO MENDONÇA SCHEIDT**

**FERRAMENTA PARA EXTRAÇÃO DE WEBTABLES E CRIAÇÃO  
DE SCRIPTS SQL**

Relatório Final do Trabalho de Conclusão de Curso como requisito para obtenção do grau em Bacharel em Sistemas de Informação pela Universidade Federal de Santa Catarina – UFSC.

Orientador(a): Carina Friedrich Dorneles

Florianópolis – SC, 2013.

Scheidt, Marcelo Mendonça

Ferramenta para Extração de WebTables e Criação de scripts SQL / Marcelo Mendonça Scheidt. Florianópolis. Centro Tecnológico – CTC – Universidade Federal de Santa Catarina – UFSC, 2013, 48 f.

Tipo de Trabalho: Relatório Final do Trabalho de Conclusão de Curso para graduação em Sistemas de Informação.

Extração, Modelo Relacional, WebTables.

**MARCELO MENDONÇA SCHEIDT**

**FERRAMENTA PARA EXTRAÇÃO DE WEBTABLES E CRIAÇÃO  
DE SCRIPTS SQL**

Relatório Final do Trabalho de Conclusão de Curso como requisito para obtenção do grau em Bacharel em Sistemas de Informação pela Universidade Federal de Santa Catarina – UFSC; Centro Tecnológico – CTC.

Orientador(a): Dr<sup>a</sup>. Carina Friedrich Dorneles.

Banca Examinadora

Dr<sup>a</sup>. Patrícia Vilain.

Dr. Ronaldo dos Santos Mello.

Dedico este trabalho a minha família pelo apoio em todos os momentos difíceis e pelas grandes felicidades que me trazem.

*Para cada esforço disciplinado há uma  
retribuição múltipla.*

- Jim Rohn.

## LISTA DE ABREVIACOES

CSS	<i>Cascading Style Sheets</i>
DOM	<i>Document Object Model</i>
EBNF	<i>Extended Backus–Naur Form</i>
HTML	<i>HiperText Markup Language</i>
IDE	<i>Integrated Development Environment</i>
IETF	<i>Internet Engineering Task Force</i>
RFC	<i>Request for Comment</i>
SGBD	Sistema Gerenciador de Banco de Dados
SQL	<i>Structured Query Language</i>
URL	<i>Uniform Resource Locator</i>

## LISTA DE FIGURAS

Figura 1 - Tabela de formatação .....	12
Figura 2 - Juros da poupança em 2013 .....	12
Figura 3 - Tabela Tradicional.....	12
Figura 4 - Taxonomia dos tipos de tabelas .....	14
Figura 5 - Formulário de autenticação.....	17
Figura 6 - Distribuição das tabelas pela Internet .....	18
Figura 7 - Quadro comparativo .....	20
Figura 8 - Funcionamento de um <i>crawler</i> .....	21
Figura 9 - Utilização padrão do <i>robots.txt</i> .....	23
Figura 10 - Utilização com mais de um <i>User-agent</i> .....	23
Figura 11 - Interface de navegação.....	25
Figura 12 - Fluxo de extração.....	27
Figura 13 - Arquitetura de camadas.....	28
Figura 14 - Algoritmo do <i>WebCrawler</i> .....	29
Figura 15 - Algoritmo geral de extração.....	30
Figura 16 - Algoritmo de criação do cabeçalho.....	30
Figura 17 - Algoritmo para extração dos dados.....	31
Figura 18 - Geração do <i>create table</i> .....	32
Figura 19 - Geração dos <i>inserts</i> .....	32
Figura 20 - Diagrama geral de classes.....	34
Figura 21 - Resultado geral para cabeçalhos .....	39
Figura 22 - Precisão e Revocação para cabeçalhos .....	39
Figura 23 - Medida F para cabeçalhos.....	40
Figura 24 - Resultado geral para dados .....	40
Figura 25 - Precisão e Revocação para dados .....	40
Figura 26 - Medida F para dados.....	41



## LISTA DE TABELAS

Tabela 1 - Países por habitantes.....	15
Tabela 2 - Cinco maiores estádios brasileiros em 2012.....	15
Tabela 3 - Especificação da Nikon D90 .....	15
Tabela 4 - Matriz de prioridade .....	16
Tabela 5 - Estados brasileiros .....	17
Tabela 6 - Exemplo para inferência por taxonomia.....	24
Tabela 7 - Tabela comparativa .....	26
Tabela 8 - Configuração do <i>crawler</i> para experimentos.....	37
Tabela 9 - Tabela comparativa com WT2SQL .....	43
Tabela 10 - Tabela com cabeçalho em negrito.....	43
Tabela 11 - Tabela com sufixos repetidos.....	44
Tabela 12 - Tabela de sufixos normalizada.....	44

## LISTA DE EQUAÇÕES

Equação 1 - Equação da precisão do cabeçalho .....	38
Equação 2 - Equação da revocação do cabeçalho .....	38
Equação 3 - Equação da medida F do cabeçalho.....	38
Equação 4 - Equação da precisão da inserção .....	38
Equação 5 - Equação da revocação da inserção .....	38
Equação 6 - Equação da medida F da inserção .....	38

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>12</b>
<b>2</b>	<b>FUNDAMENTAÇÃO.....</b>	<b>14</b>
2.1	CLASSIFICAÇÃO DAS <i>WEB-TABLES</i> .....	14
2.2	MECANISMOS DE EXTRAÇÃO.....	18
2.3	<i>CRAWLER</i> .....	20
2.3.1	O Padrão do Arquivo robots.txt.....	22
<b>3</b>	<b>TRABALHOS RELACIONADOS .....</b>	<b>24</b>
3.1	OCTOPUS .....	24
3.2	PROBASE .....	24
3.3	EXTENSÕES E COMPLEMENTOS.....	25
3.4	WEB PAGE TABLE EXTRACTOR .....	25
3.5	TABELA COMPARATIVA .....	26
<b>4</b>	<b>WT2SQL .....</b>	<b>27</b>
4.1	VISÃO GERAL.....	27
4.1.1	Arquitetura.....	27
4.2	DESENVOLVIMENTO .....	33
4.2.1	Detalhamento dos Componentes .....	34
<b>5</b>	<b>EXPERIMENTOS.....</b>	<b>37</b>
5.1	MÉTRICAS E VARIÁVEIS.....	37
5.2	RESULTADOS .....	39
<b>6</b>	<b>CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS.....</b>	<b>42</b>
	<b>REFERÊNCIAS .....</b>	<b>45</b>
	<b>APÊNDICE – TABELA EXEMPLO E ARQUIVO SQL .....</b>	<b>47</b>

## RESUMO

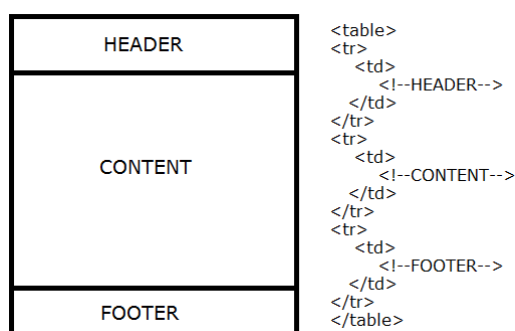
A Internet é composta por inúmeras representações de dados, dentre elas as *WebTables* contidas nas tags de páginas HTML. Essas estruturas de dados são de extrema importância para a utilização diária e o entendimento de diversas informações por elas descritas. É por causa disso que vários estudos são feitos com o objetivo de automatizar a extração de dados dessas tabelas com o objetivo de gerar bases de conhecimento, possibilitar o acesso preciso a informações, etc. Contudo, existem dois principais problemas para o processo de extração de tabelas na Internet: o reconhecimento de tabelas de dados e o reconhecimento da estrutura da tabela. Neste contexto o trabalho foca na definição de uma heurística para a resolução desses problemas e o desenvolvimento de um método para a extração e posterior inserção dos dados em um banco de dados relacional.

**Palavras chaves:** Extração, Modelo Relacional, *WebTables*.

# 1 INTRODUÇÃO

De acordo com Lin *et. al.* (2010) a *Web* é modelada e formada principalmente por dados não estruturados (textos, imagens, etc.), porém pesquisas recentes mostram que existe uma grande quantidade de dados estruturados na *Web* (CAFARELLA *et al.*, 2008). Essa quantidade continua crescendo conforme as pessoas percebem a importância dos dados estruturados para a resolução de problemas diários, como por exemplo: (a) a apresentação de informações de forma objetiva, (b) construção de bases de conhecimento, (c) possibilidade de definição de consultas mais estruturadas na *Web* e conseqüentemente acesso mais preciso às informações. (LIMAYE; SARAWAGI; CHAKRABARTI, 2010).

Uma das formas de solucionar esses problemas é pela extração das tabelas da Internet (*WebTables*) para modelos padronizados e normalizados. Contudo, o processo de extração tem dois principais problemas: (i) diferenciar uma tabela de dados das tabelas de formatação e (ii) o reconhecimento de qual estrutura a tabela de dados está apresentada. Um exemplo do primeiro ponto é mostrado nas Figura 1 e Figura 2. A primeira é uma tabela de formatação e serve apenas para organizar elementos na tela, enquanto a segunda é uma tabela de dados. Para o segundo ponto, a Figura 2 está apresentada em uma tabela matricial, o que torna a extração de dados diferente da Figura 3 que tem seus dados apresentados tradicionalmente.



**Figura 1 - Tabela de formatação**

	JAN	FEV	MAR	ABR	MAI
2013	0,5000	0,5000	0,5000	0,5000	0,5000
<b>ACUMULADO</b>	<b>0,5000</b>	<b>1,0025</b>	<b>1,5075</b>	<b>2,0151</b>	<b>2,5251</b>
2012 MP 567/12	0,4134	0,4134	0,4134	0,4134	0,4134
ACU. MP 567/12	0,4134	0,8285	1,2453	1,6639	2,0842

**Figura 2 - Juros da poupança em 2013**  
Fonte: Portal Brasil (2013)

Modelo*	Valor da carta de crédito	Parcela **
Buggy	22.860,00	452,88
UNO MILLE 1.0 FIRE	25.668,00	508,50
CELTA LIFE 1.0	27.490,00	544,61
PALIO EX 1.0	30.106,00	596,42
GOL CITY 1.0 MI FLEX 2p	28.800,00	570,55
GOL CITY 1.6	32.275,00	639,39
SIENA EX 1.0	33.468,00	663,02
KA XR 1.6	37.145,00	735,87
CORSA HATCH PR 1.0	39.066,00	773,92
CLIO SEDAN FLEX 1.6 16V	42.491,00	841,78

**Figura 3 - Tabela Tradicional**

Com essa grande quantidade de dados estruturados presentes na Internet, diversos trabalhos são realizados para estudar e verificar as melhores formas de extrair e utilizar os dados, tal como o Octopus que é um sistema de extração e integração de tabelas (CAFARELLA; HALEVY; KHOUSSAINOVA, 2009). Essa e outras ferramentas são analisadas em um capítulo posterior.

O trabalho tem como objetivo a definição de heurísticas para a resolução dos dois problemas do processo de extração de tabelas. Além disso, o desenvolvimento de uma ferramenta para realizar a extração de *WebTables* para um banco de dados relacional de forma automática, facilitando o uso das informações contidas em tabelas pela Internet. Essa automatização abrange desde a navegação das páginas, extração de dados e geração de comandos SQL para criar (*create table*) e inserir (*insert into*) os dados que representam a tabela extraída.

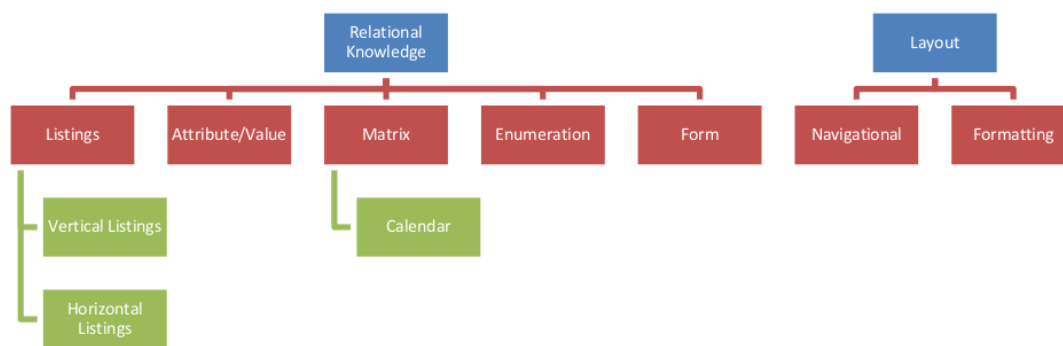
O trabalho é dividido em seis capítulos. Em seguida, o capítulo de fundamentação teórica descreve os principais conceitos e tópicos importantes para o melhor entendimento do projeto. O terceiro capítulo consiste nos trabalhos relacionados, onde são descritas ferramentas que possuem alguma relação com o estudo. O quarto descreve projeto propriamente dito, onde são abordados pontos de sua estrutura e desenvolvimento. Em seguida, o quinto capítulo apresenta os resultados dos experimentos com a ferramenta proposta. Por fim, a conclusão, onde os resultados alcançados com o desenvolvimento da ferramenta são revistos de acordo com o objetivo do trabalho.

## 2 FUNDAMENTAÇÃO

Neste capítulo são descritos definições e conceitos necessários para o melhor entendimento do projeto.

### 2.1 CLASSIFICAÇÃO DAS WEB-TABLES

De acordo com Crestan e Pantel (2011), as tabelas da Internet dividem-se em dois grandes grupos (Tabelas Relacionais e de Layout) que depois são classificados conforme suas características. Isso pode ser visto na Figura 4 abaixo.



**Figura 4 - Taxonomia dos tipos de tabelas**  
Fonte: Crestan; Pantel (2011).

### Tabelas Relacionais

São consideradas tabelas relacionais as que possuem a tripla  $(P, S, O)$ , sendo  $P$  o predicado,  $S$  o sujeito da relação e  $O$  o objeto. Essa tripla caracteriza um *schema* (CRESTAN; PANTEL, 2011). Contudo, as *WebTables* podem ser modeladas e apresentadas de formas diferentes.

As tabelas relacionais são classificadas em: (a) *Vertical Listing*; (b) *Horizontal Listing*; (c) *Attribute/Value*; (d) *Matrix*; (e) *Calendar*; (f) *Enumeration*; e (g) *Form*. Cada uma delas é descrita a seguir (CRESTAN; PANTEL, 2011).

- *Vertical Listings*

Essas tabelas listam um ou mais atributos para uma série de entidades similares. Por exemplo, a Tabela 1, uma lista de países e sua população (um por linha). O grande desafio em extrair informações deste tipo é a identificação de qual coluna da lista é o

sujeito e quais são os predicados. Normalmente a primeira coluna é o sujeito e as seguintes são os predicados. Os objetos podem ser extraídos diretamente dos valores das células.

País	Habitantes (em 2010)
<b>China</b>	1 321 852 000
<b>Índia</b>	1 189 172 906
<b>Estados Unidos</b>	313 232 044
<b>Indonésia</b>	245 613 043
<b>Brasil</b>	196 741 680

Tabela 1 - Países por habitantes.

- *Horizontal Listings*

Esta categoria apresenta o sujeito em uma linha e os predicados em linhas subsequentes. Semelhante a categoria *Vertical Listings*, a dificuldade aqui é identificar qual linha possui o sujeito e quais possuem os predicados. Assim como a categoria acima, o objeto pode ser extraído diretamente do valor da célula. Isto pode ser visto na Tabela 2.

<b>Estádio</b>	Maracanã	Mané Garrincha	Morumbi	Castelão	Minerão
<b>Capacidade</b>	78 838	70 000	66 795	64 846	62 170
<b>Cidade</b>	Rio de Janeiro	Brasília	São Paulo	Fortaleza	Belo Horizonte

Tabela 2 - Cinco maiores estádios brasileiros em 2012.

- *Attribute/Value*

É um caso especial de *Horizontal Listings* e *Vertical Listings*. O que a distingue das demais é o fato que o sujeito não está contido na tabela, mas sim no corpo do documento. Esta categoria é normalmente utilizada em uma tabela de fato sobre determinada entidade. Por exemplo, a Tabela 3 contendo especificações sobre um produto, porém não contém o nome do produto. O desafio neste caso é justamente a detecção do sujeito fora da tabela.

#### Especificação da Nikon D90

<b>Fabricante</b>	Nikon
<b>Categoria</b>	DSLR
<b>Peso</b>	620 gramas
<b>Resolução</b>	12.3 megapixels

Tabela 3 - Especificação da Nikon D90



- *Matrix*

Matrizes possuem o mesmo tipo de dado para cada célula na intersecção entre linha e coluna. A primeira linha e a primeira coluna normalmente possuem o sujeito, porém o objeto é inferido somente por meio da combinação dos dois sujeitos. Já o predicado não é frequentemente encontrado nessas tabelas. Para exemplificar, a Tabela 4 a seguir representa uma matriz de prioridades sobre as características de um produto para diferentes clientes.

	Cliente 1	Cliente 2	Cliente 3
Demora da entrega do produto	3	5	4
Validade do produto	5	5	5
Garantia do Produto	4	5	1
Suporte e instalação	3	2	2

**Tabela 4 - Matriz de prioridade**

- *Calendar*

Um caso específico de matriz, diferente apenas pela semântica. Em calendários, o sujeito é uma data e o predicado é uma relação genérica. Por exemplo, as datas marcadas para a apresentação da orquestra local. Neste caso, o predicado seria a data marcada e o objeto o valor da célula. A dificuldade nesta categoria é descobrir os predicados, que muitas vezes não se encontram na tabela.

- *Enumeration*

Representam uma série de objetos que possuem a mesma relação semântica. A Tabela 5 é um exemplo e contém os nove primeiros estados brasileiros em ordem alfabética. O desafio é descobrir o predicado, que nem sempre é listado explicitamente na tabela ou na própria página. O sujeito da tripla é o valor da célula e o objeto é normalmente o cabeçalho.

Acre
Alagoas
Amapá
Amazonas
Bahia
Ceará
Distrito Federal
Espírito Santo
Goiás

Tabela 5 - Estados brasileiros

- *Form*

Este tipo é caracterizado por campos de entrada ou seleção. Um caso típico é um formulário de autenticação com os campos de usuário e senha organizados dentro de uma tabela (Figura 5). Os predicados são os rótulos do formulário e o objeto é aquilo que é requisitado ao usuário. Tipicamente não há sujeito para a tripla semântica neste caso.

Welcome! Please log-in.

Login	
Login:	<input type="text"/>
Password:	<input type="password"/>

Figura 5 - Formulário de autenticação

## Tabelas de Layout

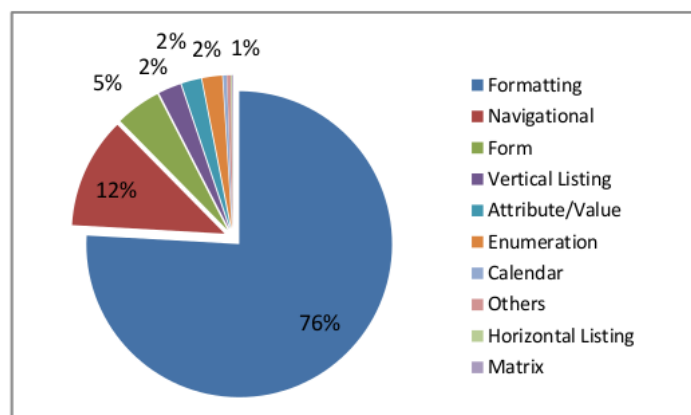
Apesar de serem desconsideradas para a extração de dados neste trabalho, as tabelas de layout possuem os dois grupos:

- *Navigational*

São aquelas utilizadas para organizar os links de navegação da página. Não há uma relação clara entre as células, exceto pela navegação interna ou externa à página.

- *Formatting*

Possui apenas a função de organizar visualmente os elementos da página. Contudo representam a maior parcela das tabelas encontradas na Internet, conforme mostra a Figura 6.



**Figura 6 - Distribuição das tabelas pela Internet**

Fonte: Crestan; Pantel (2011).

Com base nas pesquisas de Crestan e Pantel (2011) é possível constatar que mais de 80% das tabelas da Internet são da categoria layout, enquanto as demais são as tabelas relacionais.

## 2.2 MECANISMOS DE EXTRAÇÃO

Conforme Miao *et. al.* (2009), um consenso sobre qual método deve ser utilizado no processo de identificação e extração de dados ainda não foi alcançado, justamente porque as tabelas, como visto anteriormente, não respeitam nenhum padrão de formatação e estilo, ou segundo Limaye, Sarawagi, Chakrabarti (2010, p. 1338) as *WebTables* “[...] não aderem uniformemente a nenhum esquema”.

Existem seis principais técnicas de extração de dados, sejam eles estruturados ou não. Esta seção tem o intuito de descrever essas técnicas de acordo com Laender *et. al.* (2002).

### Linguagens Declarativas

Esta categoria inclui ferramentas que propõem novas linguagens para substituir outras tradicionais. O funcionamento da maioria delas é a utilização de uma gramática baseada na especificação EBNF<sup>1</sup>. Para cada página, um conjunto de regras é definido. Cada uma define a estrutura de um símbolo não terminal da gramática, encerrando em um símbolo terminal ou outro não terminal.

<sup>1</sup> EBNF é um acrônimo para *Extended Backus-Naur Form* que representa uma notação para expressar linguagens livres de contexto.

Fonte: <https://www.cl.cam.ac.uk/~mgk25/iso-14977.pdf>

### **Análise de estruturas HTML**

Essas ferramentas, antes de extrair os dados, transformam a página em uma árvore HTML hierarquizada (DOM) e depois realizam operações de busca percorrendo a árvore através de seus nodos e extraindo informações de suas folhas. A ferramenta desenvolvida por este projeto encaixa-se nesta categoria.

### **Processamento de Linguagens Naturais**

Essa técnica baseia-se na utilização de filtros, marcação léxica e semântica para construir relações entre frases e sentenças do texto. Por meio dessas relações, regras podem ser derivadas para decidir qual parte do documento é importante e deve ser extraída.

### **Aprendizado de Máquina**

Elas representam programas que, inicialmente, devem ter um conjunto de treinamento para aprender quais estruturas de dados são relevantes para a extração e quais não são.

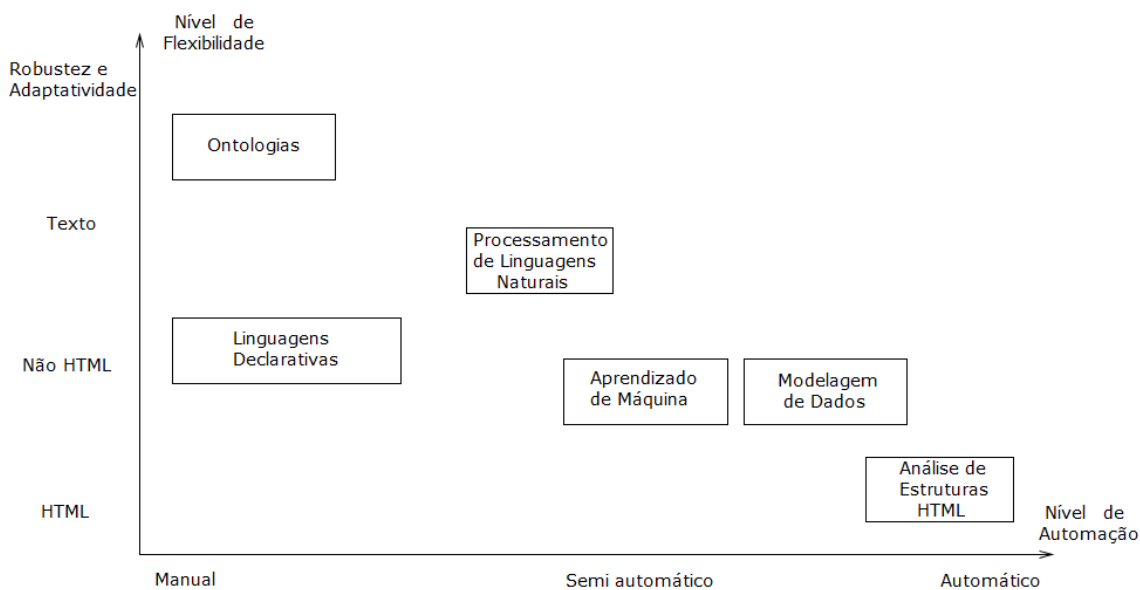
### **Modelagem de Dados**

As ferramentas desta categoria orientam-se pela busca na *Web* por uma determinada estrutura fornecida inicialmente, como uma tabela ou lista.

### **Ontologias**

Por fim, diferente das técnicas apresentadas acima, que utilizam a estrutura em que os dados estão dispostos no documento para realizar a extração, este método realiza um processo de extração baseando-se somente nos dados em si e em uma ontologia para localizar constantes no documento e construir objetos a partir delas. A ontologia utilizada deve ser previamente construída, descrevendo quais são os dados de interesse, suas relações, aparência léxica, contexto, suas palavras-chaves e sinônimos.

Os autores realizaram ainda uma comparação entre as técnicas (resumido na Figura 7), onde no eixo X está representado o grau de automação e no eixo Y o grau de flexibilidade (LAENDER *et. al.*, 2002).



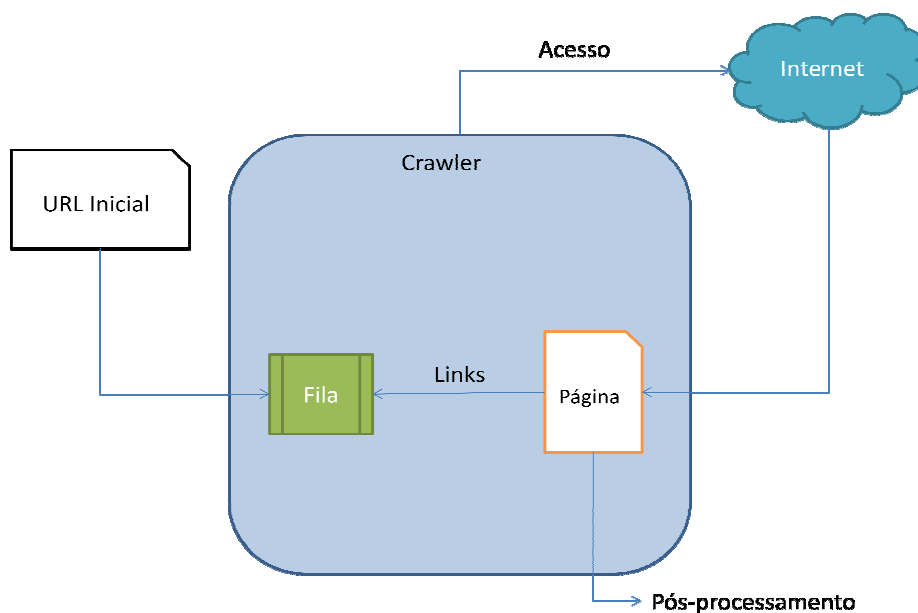
**Figura 7 - Quadro comparativo**  
 Fonte: Laender; Ribeiro-Neto; Silva (2002).

Com base neste quadro, pode-se notar um paralelo entre as Ontologias e a Análise de estruturas HTML. Quanto mais automática a ferramenta, menos flexível ela é. Desse mesmo modo, quanto menor o grau de automação, mais o grau de flexibilidade.

### 2.3 CRAWLER

A utilização de *crawlers* na Internet é bem difundida e apesar de ter muitos sinônimos (*spider*, *robot*, *bot*, etc.) seu funcionamento é relativamente simples. Como por exemplo, o *Googlebot* que realiza a indexação das páginas para o motor de busca da Google (GOOGLE, 2012).

Segundo Cho (2002) o processo de um *crawler* se resume em recuperar uma URL de uma fila, acessar a página referente a essa URL, recolher os links dessa página acessada e adicionar na fila. Durante esse procedimento, o *crawler* salva a página acessada (em disco ou em memória) para um pós-processamento. Esse processo é repetido até que alguma condição de parada ocorra, como por exemplo, o número de páginas acessadas. A Figura 8 ilustra este funcionamento.



**Figura 8 - Funcionamento de um crawler**

Fonte: Adaptado de Cho (2002).

O pós-processamento pode variar de acordo com o objetivo de cada *crawler*, como visto acima, o *Googlebot* realiza a indexação das páginas, porém o *crawler* criado para o projeto tem o objetivo de extrair as tabelas das páginas e salvá-las em arquivos.

Uma vez que os *crawlers* percorrem a Internet automaticamente, deve existir uma maneira de regradar o acesso deles às páginas da Internet, pois muitas vezes esse acesso é indesejado ou prejudica o funcionamento do *site*. Para evitar isso, em junho de 1994, após um consenso de profissionais liderados por Martijn Koster, criou-se um documento que especifica como indicar a um *crawler* quais páginas ele pode ou não acessar dentro de um determinado domínio por meio de um arquivo hospedado na raiz do domínio (ROBOTSTXT, 2012).

Em 1997, houve uma tentativa de oficializar o documento, porém não se obteve sucesso. Até hoje não existe uma documentação oficial sobre o padrão, como um RFC<sup>2</sup>. Contudo, o padrão é amplamente aceito e utilizado pela Internet.

<sup>2</sup> RFC é um acrônimo para *Request for Comment*. São documentos lançados pelo IETF (*Internet Engineering Task Force*) que descrevem os padrões que cada protocolo da Internet deve seguir. Fonte: <https://www.ietf.org>

### 2.3.1 O Padrão do Arquivo robots.txt

De acordo com o consenso, o padrão criado para o documento deve seguir os seguintes requisitos:

- O nome do arquivo deve estar de acordo com as normas de nomes de arquivos para todos os sistemas operacionais;
- A extensão do arquivo não deve exigir configuração extra para o servidor;
- O nome do arquivo deve representar seu propósito e ser fácil de lembrar;
- A frequência de colisão do nome do arquivo deve ser mínima;
- A localização deste arquivo deve ficar na raiz do servidor, para facilitar o acesso e não prejudicar o desempenho do *site*, por exemplo, *http://www.ufsc.br/robots.txt*.

O arquivo consiste em um ou mais registros separados por uma ou mais quebras de linha (determinadas pelos caracteres CR e NL), sendo que cada linha deve seguir a seguinte forma (ROBOTSTXT, 2012):

*<campo>:<espaço\_opcional><valor><espaço\_opcional>*

O *<campo>* pode possuir dois valores básicos que são descritos a seguir:

- *User-agent*: O *<valor>* para este campo representa o nome do *crawler* que terá a política de acesso descrita. Se o valor for um asterisco (\*), então as regras subsequentes ao campo são todos os *crawlers* que não possuem regras específicas.
- *Disallow*: O *<valor>* deste campo descreve uma URL que não é visível para o *crawler*. Se o valor for vazio, significa que não existem restrições e se o valor for uma barra (/), significa que todas as páginas são restritas.

O consenso ainda descreve que comentários podem ser incluídos por meio do caractere # no início do comentário, indicando que tudo a partir daquele ponto poderá ser descartado por programas. A utilização desse padrão é exemplificada a seguir.

Conforme a Figura 9, todos os *crawlers* (representado pelo \* no atributo *User-agent*), não possuem permissão para acessar os diretórios e páginas listados pelos atributos *disallow*. Já na Figura 10, pode-se notar que o *crawler* com nome de *cybermapper* possui acesso irrestrito ao domínio, enquanto os demais não podem acessar o diretório */cyberworld/map/*.

```
# robots.txt for http://www.example.com/  
  
User-agent: *  
Disallow: /cyberworld/map/ # This is an infinite virtual URL space  
Disallow: /tmp/ # these will soon disappear  
Disallow: /foo.html
```

**Figura 9 - Utilização padrão do *robots.txt***  
Fonte: Robotstxt (2012).

```
# robots.txt for http://www.example.com/  
  
User-agent: *  
Disallow: /cyberworld/map/ # This is an infinite virtual URL space  
  
# Cybermapper knows where to go.  
User-agent: cybermapper  
Disallow:
```

**Figura 10 - Utilização com mais de um *User-agent***  
Fonte: Robotstxt (2012).



### 3 TRABALHOS RELACIONADOS

Neste capítulo, são apresentados extratores de tabelas, que possuem alguma semelhança com o trabalho desenvolvido.

#### 3.1 OCTOPUS

O Octopus (CAFARELLA; HALEVY; KHOUSSAINOVA, 2009) é um sistema que combina operações de busca, extração, limpeza e integração de dados na Internet. Todas as operações são divididas em três etapas básicas: (i) Busca, (ii) Contexto e (iii) Extensão. A busca é realizada com base em palavras-chaves fornecidas pelo usuário e o resultado é composto pelas tabelas encontradas apresentados de forma ranqueada de acordo com sua relevância.

Em seguida, na etapa do contexto, o usuário pode escolher uma determinada tabela-resultado apresentada na etapa anterior e adicionar informações relativas ao contexto em que a tabela está inserida. Por fim, na etapa de extensão, é possível adicionar colunas na tabela com base em outras tabelas do resultado inicial (CAFARELLA; HALEVY; KHOUSSAINOVA, 2009).

#### 3.2 PROBASE

A proposta da ferramenta Probase (WANG *et al.*; 2012) é utilizar uma taxonomia de conceitos para auxiliar a identificação dos cabeçalhos das tabelas e por sua vez facilitar a extração de suas informações. A cada tabela extraída, a taxonomia é alimentada com novos conceitos. Para exemplificar o processo, considere o conceito de *politician* relacionado aos conceitos de *political party* e *assumed office*. Dessa maneira, para a Tabela 6 abaixo a ferramenta percorrerá cada linha e cada coluna medindo o grau de confiança para eleger um cabeçalho.

Name	Birthdate	Political Party	Assumed Office
Barack Obama	4 Aug 1961	<i>Democratic</i>	2009
Arnold Schwarzenegger	30 Jul 1947	<i>Republican</i>	2003

**Tabela 6 - Exemplo para inferência por taxonomia**

Fonte: Adaptado de Wang *et. al.*, (2012).

Após isso, a primeira linha é eleita, uma vez os dados da linha relacionam-se na taxonomia, depois os conceitos de *name* e *birthdate* são adicionados à taxonomia relacionando-se aos conceitos utilizados. Por fim, o processo passa para a etapa de extração de dados percorrendo as linhas subsequentes.

### 3.3 EXTENSÕES E COMPLEMENTOS

O *Web Table Extractor* (WTE, 2013) é uma extensão disponível para o Internet Explorer e o Firefox e possui funcionamento manual – quando o usuário acessa uma página, ele pode selecionar a área de uma tabela e solicitar a extração. Após isso, os dados ficam disponíveis na área de transferência e podem ser colados em arquivos de texto ou planilhas eletrônicas.

Nesta mesma categoria, existe o *TableTools2* (TABLETOOLS2, 2013) que é um complemento para o Firefox que, além de copiar os dados, possui várias funções: organizar e filtrar, gerar gráficos, dentre outras. Todas as funções estão disponíveis após o usuário selecionar a área de dados e requisitar que o complemento copie a tabela. Assim como o anterior, os dados ficam disponíveis na área de transferência.

### 3.4 WEB PAGE TABLE EXTRACTOR

O *Web Page Table Extractor* – WPTE (OOLUTION TECHNOLOGIES, 2013) é um programa gratuito para extração de tabelas *Web* desenvolvido pela Oolution Technologies. Seu funcionamento acontece manualmente junto com a navegação do usuário. A cada página navegada o usuário escolhe se quer ou não realizar um *parse* do conteúdo HTML. Se alguma tabela for encontrada, o programa possibilita sua extração. Porém, o resultado da extração salva apenas o código HTML da tabela em um arquivo separado. A Figura 11 ilustra a interface de navegação da ferramenta.



Figura 11 - Interface de navegação

### 3.5 TABELA COMPARATIVA

A Tabela 7 apresenta as principais características das ferramentas descritas neste capítulo de forma comparativa. Nela é possível verificar que as duas primeiras são de funcionamento automático, ou seja, contam com um *crawler*, enquanto as demais são de funcionamento manual. Note que ambas as ferramentas automáticas exigem um conhecimento anterior ao processo de extração. Para o aprendizado de máquina é necessário um conjunto de treino e para a geração de ontologias é necessário uma base de conhecimento.

<b>Ferramenta</b>	<b>Automação</b>	<b>Resultado da Extração</b>	<b>Retro alimentação</b>	<b>Integração de dados</b>	<b>Técnica</b>
Octopus	Sim	Tabelas formatadas	Não	Sim	Aprendizado de máquina
Probase	Sim	Tabelas formatadas	Sim	Não	Ontologia
Web Table Extractor	Não	Área de transferência	Não	Não	Cópia de dados
TableTools2	Não	Área de transferência	Não	Não	Cópia de dados
Web Page Table Extractor	Não	Arquivo HTML	Não	Não	Cópia de dados

**Tabela 7 - Tabela comparativa**

## 4 WT2SQL

Neste capítulo, é tratado o desenvolvimento e aplicação do projeto *WebTable2SQL* (WT2SQL). Primeiramente é descrita a arquitetura de uma forma geral, passando pelo fluxo de informação dentro da ferramenta e suas camadas. Posteriormente é detalhada a organização dos módulos dentro de cada camada e suas funções.

Como visto no Capítulo 2, existem diversas formas de representação de *WebTables*, então, neste trabalho, são consideradas para extração apenas as estruturas categorizadas como tabelas relacionais, levando em conta estruturas complexas, como tabelas aninhadas, além disso, são descartadas do processo de extração as tabelas de layout descritas na Seção 2.1 e a tabela do tipo *Form e Matrix* descrita na mesma Seção.

### 4.1 VISÃO GERAL

Esta seção ilustra as camadas e a arquitetura utilizada no desenvolvimento do projeto, assim como o fluxo de informação dentro delas.

Com base no funcionamento de um *crawler* visto no Capítulo 2, foi criado um fluxo que o *crawler* deste trabalho deve seguir para cumprir as etapas necessárias para a extração.

Neste caso, as etapas são:

- a. Navegação por páginas da Internet;
- b. Reconhecimento e extração;
- c. Escrita de arquivos em disco.

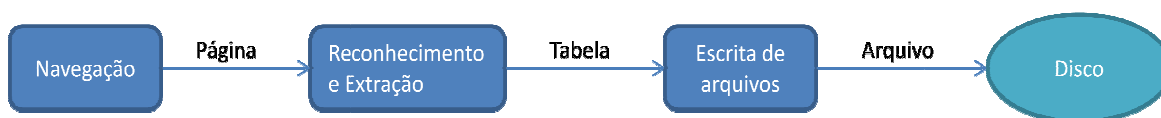


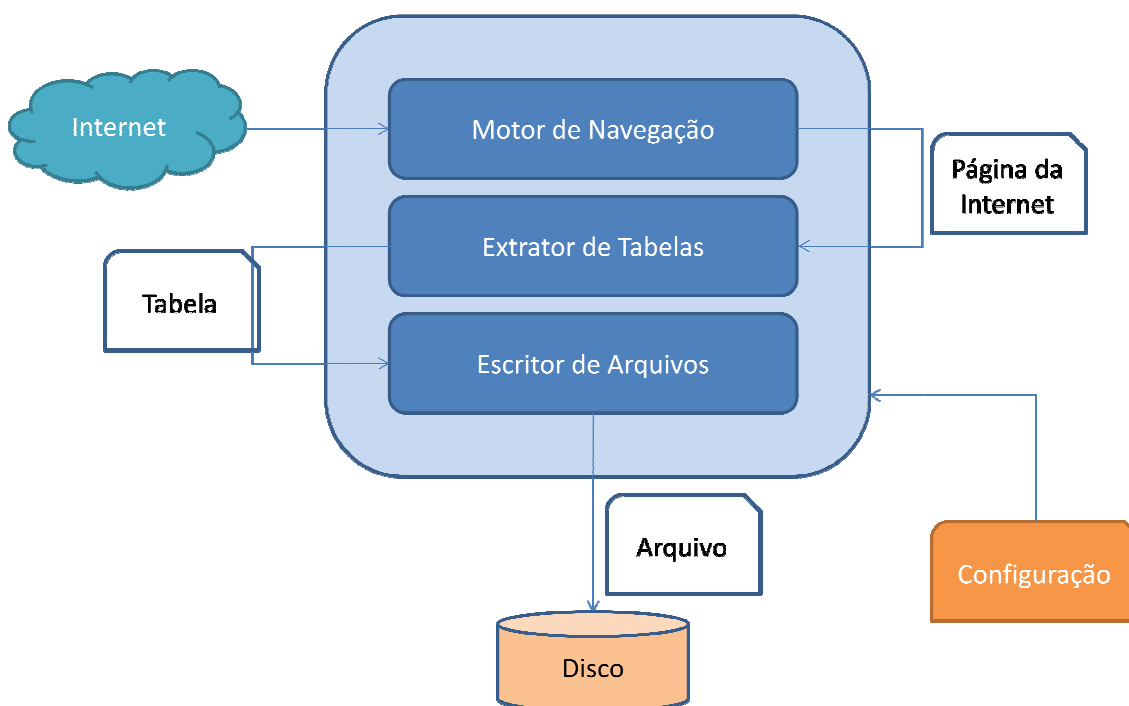
Figura 12 - Fluxo de extração

Na Figura 12, as arestas representam o fluxo de dados e as caixas os módulos da ferramenta. Por fim, o círculo representa o disco onde os arquivos são salvos.

#### 4.1.1 Arquitetura

A arquitetura modelada teve como objetivo abrigar cada uma das operações citadas anteriormente, assim como cada um dos seus módulos. Desta forma, foi criada uma

camada para cada etapa onde o produto resultante de uma é a informação necessária de entrada para a próxima etapa. Isto pode ser visto na Figura 13.



**Figura 13 - Arquitetura de camadas**

Foi por meio da arquitetura ilustrada na Figura 13 que os componentes foram criados, cada qual com sua função bem definida, evitando retrabalho e sobreposição de funções dentro do programa. Os componentes são descritos abaixo.

### **Motor de Navegação** (*WebCrawler*)

Este componente possui as características de um *crawler* descritas no Capítulo 2. Ele navega pela *Web* e busca os links para as próximas páginas até uma condição de parada ser satisfeita. O resultado de cada acesso é passado para a camada de extração para verificar se é possível extrair uma tabela da página.

Diferente de um *crawler* comum que envia todas as páginas para o pós-processamento, o motor de navegação proposto realiza um pequeno filtro e somente são enviadas as páginas que contiverem pelo menos uma *tag* `<table>`. Além disso, a ferramenta proposta conta com um controle de acesso para respeitar os domínios que possuem o arquivo *robots.txt*. O processo de navegação é realizado por meio do algoritmo apresentado na Figura 14.

```

while not stopCondition do // stop condition can be an empty queue or a limit pages to visit.
begin
  url <- next url from queue

  if can visit url then // check in robots.txt if crawler can visit the url.
  begin
    page <- visit url // visit and retrieve the page.

    if page has table then
    begin
      extract page // send to extraction.
    end

    queue <- links from page // retrieve links from page and add to queue.
  end
end
end

```

Figura 14 - Algoritmo do *WebCrawler*

### Extrator de Tabelas (*TableExtractor*)

Esta camada contém as regras necessárias para realizar a identificação e extração das tabelas da página. Para cada página recebida o extrator percorre todas as tabelas disponíveis e verifica se é possível extrair ou não os dados. A verificação é feita com base nos parâmetros passados pelo arquivo de configuração e outras definições pré-estabelecidas. Essa verificação é a heurística utilizada para separar tabelas de dados de tabelas de formatação. Seus parâmetros são:

- O número de linhas e colunas da tabela deve ser no mínimo o informado no arquivo de configuração.
- Tabelas que possuem listas (*tags* `<ul>`, `<ol>` e `<li>`) não são extraídas. Isto porque listas e sublistas são dados complexos e sua extração exigiria uma heurística à parte, o que não é foco deste trabalho.
- Tabelas que possuem elementos de formulários (botões, campos de entrada de texto, seleção, etc.) não são extraídas.

Após a verificação acima, o processo continua com a extração procurando o cabeçalho da tabela para criar os rótulos das colunas e depois o corpo da tabela para criar os dados – esses dois processos são detalhados posteriormente. Por fim, a tabela extraída é enviada para a escrita em disco. A Figura 15 ilustra o procedimento de extração.

```

page <- received from web crawler
while page has tables do // repeat the process while page has tables
begin
  htmlTable <- get next html table from page

  if htmlTable is extractable then // check if html table is extractable
begin
  table <- new table object // create a new table object
  table <- add table header from htmlTable // adds to the object the header
  table <- add table body from htmlTable // add to the object the body
  write table // writes the object in disk

end

end
end

```

**Figura 15 - Algoritmo geral de extração**

O procedimento de criação dos rótulos das colunas, que representa a heurística de reconhecimento de estruturas, envolve a identificação do cabeçalho da *WebTable* e a extração deles para gerar os comandos de *create table*. Isso é feito primeiramente verificando se a tabela HTML possui a *tag <th>* (*table header*). Quando isso ocorrer, os valores dos rótulos serão os mesmos à dessas *tags*. Porém, quando a tabela não apresentar nenhuma *tag <th>*, os rótulos são criados com base na quantidade de linhas e colunas da tabela. Quando houver mais linhas do que colunas, a primeira linha é eleita o cabeçalho, se não, a primeira coluna é eleita. A Figura 16 apresenta o algoritmo do processo de reconhecimento de rótulos. O processo de descoberta de cabeçalhos pode ser reutilizado ao longo da extração diversas vezes devido a tabelas aninhadas.

```

if htmlTable has tag th then // check if html table has tags th
begin
  header <- get th from htmlTable // retrieve tags th from html table
end

else
begin
  row <- get first row of htmlTable
  column <- get first column of htmlTable

  if numberOfRows >= numberOfColumns then // check sizes of rows and columns
begin
  header <- first row // the first row is elected header
end

else
begin
  header <- first column // the first column is elected header
end

end
end

```

**Figura 16 - Algoritmo de criação do cabeçalho**

Com base na criação dos rótulos das colunas é possível identificar a direção da tabela, se é vertical ou horizontal. Com essa informação a extração dos dados inicia-se. Relembrando que uma tabela considerada vertical é aquela que apresenta os dados verticalmente, ou seja, cada linha é um objeto novo. Dessa maneira, os dados são extraídos por linha. Se a tabela for horizontal, seus dados estão expostos a cada coluna, então os dados são extraídos respeitando essa ordem. Se durante a extração de dados, novos itens de cabeçalhos forem encontrados, caracterizando uma tabela aninhada, o cabeçalho da tabela é feito levando em consideração estes itens. O processo de extração de dados é apresentado na Figura 17.

```
if table is vertical then // check if table direction is vertical
begin
  for each row in htmlTable do
  begin
    table <- add next row from htmlTable
  end
end
else
begin
  for each column in htmlTable do
  begin
    table <- add next column from htmlTable
  end
end
end
```

Figura 17 - Algoritmo para extração dos dados

Tanto na criação dos rótulos como na extração dos dados, são realizados diversos tratamentos:

- Separação de células que possuem o atributo *colspan* e *rowspan*;
- Marcação de células vazias para a escrita NULL no arquivo SQL;
- Criação de colunas novas quando tabelas aninhadas forem analisadas;
- Adição da palavra *image* para células que contiverem apenas imagens.

Por fim, a funcionalidade de tratar células multi-valoradas que é opcional e ativada a partir do arquivo de configuração. Esta função realiza o produto cartesiano entre todos os valores da célula com o restante da linha. Se a linha possuir mais de uma célula com essa característica, isso é feito para todas as células.

A quebra de linha (*tag <br>* e suas variações) é considerada como separador de valor dentro de uma célula. Com isso, o processo pode se tornar lento. Considere uma tabela



de cinco linhas e cinco colunas, onde cada célula possui três valores diferentes separados por uma quebra de linha. Se o produto cartesiano for aplicado, cada linha irá virar 243 novas linhas. Considerando cinco linhas da tabela original, resulta em um total de 1.215 linhas.

### Escritor de Arquivos (*SqlGenerator*)

Uma vez que a camada de extração termina seu processo, uma tabela é enviada para este componente para ser escrita em disco. Neste processo é gerado o *script* de criação da tabela a partir do cabeçalho. Para cada rótulo de coluna, uma nova coluna é adicionada no *script* com o tipo TEXT conforme executado pelo algoritmo descrito na Figura 18.

```
file <- new file object // creates a new file object
file <- generate and add table name // generate a table name
file <- add date and url // add to file url and generation date

for each label in table header do
  begin
    file <- add label to create statement
  end
```

Figura 18 - Geração do *create table*

Os comandos de inserção (*insert into*) são gerados a partir dos dados. Cada linha resultará em uma tupla a ser inserida sem a criação de chave primária. Logo depois o arquivo é salvo em disco e o processo recomeça para a próxima tabela. A Figura 19 apresenta o algoritmo descrito e no Apêndice é apresentado uma tabela e seu arquivo SQL correspondente.

```
for each row in table do
  begin
    file <- generate and add insert statement
  end

write file
```

Figura 19 - Geração dos *inserts*

### Configuração

Como citado ao longo deste capítulo, foi necessária a criação de um arquivo de configuração para lidar com as variáveis do processo. Esse arquivo armazena os seguintes parâmetros para a execução:

- Número máximo de páginas: Representa o número máximo de páginas que o *crawler* deve navegar.
- URL inicial: A semente que o *crawler* irá utilizar para iniciar a navegação na Internet.
- Número mínimo de linhas: Quantidade mínima de linhas que uma tabela deve possuir para entrar no processo de extração.
- Número mínimo de colunas: Mesmo significado que a anterior, porém aplicado ao número de colunas.
- Caminho da pasta: Localização da pasta em que serão salvos os arquivos gerados pelo programa, assim como o arquivo de Log.
- Versão do navegador: Como algumas páginas são reconhecidas de forma diferente para determinados navegadores, essa opção permite escolher entre o Firefox e o Internet Explorer para ser utilizado na navegação.
- Tempo de desconexão: Representa o tempo (em milissegundos) que o *crawler* deve aguardar por uma resposta de uma página.
- Células multi-valoradas: Uma opção de sim (*true*) ou não (*false*), sinalizando para o programa se essas células devem ser tratadas ou não. Essa escolha existe pois pode impactar no desempenho do programa e nem sempre o usuário necessita deste tratamento.

## 4.2 DESENVOLVIMENTO

Para o desenvolvimento do programa, foi utilizado a IDE (*Integrated Development Environment*) Eclipse na sua versão *indigo* (ECLIPSE, 2012) e a linguagem de programação Java da Oracle. Essas escolhas foram feitas uma vez que a IDE Eclipse é *open source* e a linguagem Java é amplamente utilizada e possui recursos suficientes para o desenvolvimento (ORACLE, 2012).

Como explicado anteriormente, cada camada tem sua função. Dessa maneira, os componentes internos foram desenvolvidos respeitando essa separação, resultando em pacotes de classes diferentes e relacionamentos entre eles. A Figura 20 ilustra o diagrama geral de classes com as principais relações.

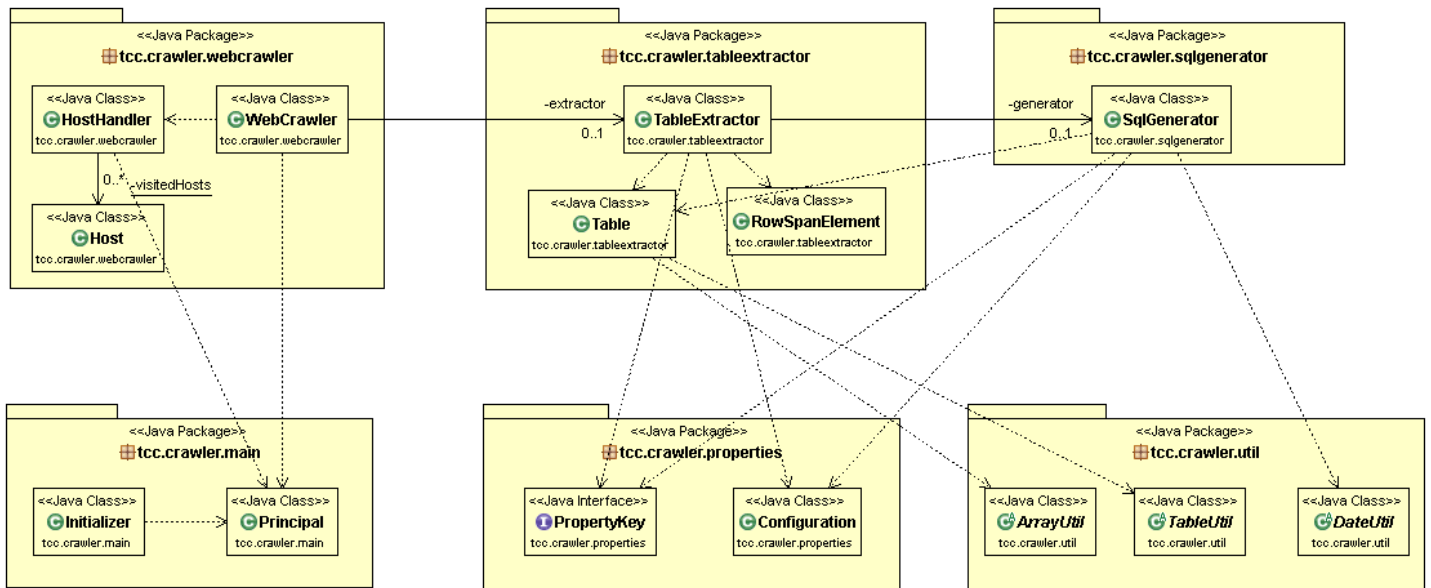


Figura 20 - Diagrama geral de classes

#### 4.2.1 Detalhamento dos Componentes

Como pode ser visto, além dos três pacotes superiores, que representam o fluxo de extração, existem outros criados com a finalidade de auxiliar o desenvolvimento e separação das partes. O detalhamento a seguir inicia-se pelos pacotes auxiliares.

##### **Pacote *Main***

Contêm duas classes responsáveis com as funções seguintes:

*Initializer*, responsável em iniciar e finalizar o programa, e a *Principal*, que realiza uma série de verificações ao iniciar e as operações de *log*, como mostrar e escrever mensagem sobre a execução.

##### **Pacote *Properties***

Também contêm duas classes, sendo a *Configuration* responsável em carregar e gerenciar o arquivo de configuração e a *PropertyKey* possui os valores das chaves para as propriedades dentro do arquivo.

##### **Pacote *Util***

As classes existentes nesse pacote auxiliam operações específicas que são necessárias em diversas partes do programa. O *ArrayUtil* realiza o produto cartesiano entre células multi-valoradas de uma tabela e foi posta nessa classe pois as operações envolvidas

eram demasiadamente grandes para ficar junto às outras classes. A classe *DateUtil* possui operações que envolvem data e hora que são utilizadas por todo o programa. E por fim, a classe *TableUtil* executa operações de verificação em uma tabela.

### **Pacote *WebCrawler***

Nesta camada está implementado o *crawler* para a visitação de páginas e a descoberta de links. Suas classes e funções são:

O *WebCrawler* é a principal classe deste pacote e possui diversas funções realizadas na navegação e descoberta de *links*. Utiliza a semente passada para iniciar a navegação na Internet e apenas encerra quando o valor de páginas atingiu o configurado ou quando a lista de links termina.

O *HostHandler* é o controlador de domínios. Responsável em guardar os domínios já acessados e carregar as configurações do arquivo *robots.txt* para novos domínios. A classe mantém uma lista de *Host*.

*Host*, por sua vez, é a classe que representa um domínio em si (*www.ufsc.br*), com informações de nome, configurações de acesso e proibições.

### **Pacote *TableExtractor***

A classe *TableExtractor* é onde todo o processo de reconhecimento e extração da tabelas acontece. Primeiramente, é verificado se a página contém tabelas que respeitam as configurações. Após isso, é iniciado o processo de reconhecimento e extração. Primeiro é extraído o cabeçalho da tabela e por fim seus valores. Tudo isso é armazenado em um objeto da classe *Table*.

A classe *Table* representa, como visto acima, uma tabela extraída, com informações de nome, colunas, linhas e valores dos dados.

Já a classe *RowSpanElement* é uma caso específico e é utilizada quando alguma célula da tabela possui o atributo *rowspan*. Neste caso, esta classe armazena informações de valor do atributo, índice e conteúdo da célula.

### **Pacote *SqlGenerator***

Por fim, a classe *SqlGenerator* é responsável em receber tabelas extraídas e escrevê-las em disco. Para isso, a classe cria um arquivo adicionando informações de data e hora e depois gera o script de criação e inserção da tabela percorrendo o objeto *Table* recebido.

Além disso, o programa utiliza a biblioteca HTMLUnit (HTMLUNIT, 2012) para realizar as operações de acesso a páginas e busca na árvore DOM de cada uma delas. Escolheu-se por utilizar essa biblioteca por estar sob a *Apache License v2.0* e ser *open source* (APACHE, 2012).

## 5 EXPERIMENTOS

Na seção que segue são apresentados os experimentos realizados para avaliar a ferramenta proposta por este trabalho. Para isso dois aspectos devem ser verificados:

1. **Header:** Identificação e extração da estrutura da tabela, levando em consideração o número de linhas e colunas, assim como os rótulos de cabeçalho da tabela;
2. **Insert:** Identificação e extração dos dados pertinentes à tabela, ou seja, os dados de cada célula da tabela extraída.

Foram extraídas 4.000 páginas divididas igualmente em quatro domínios diferentes, visando aumentar a variedade de tipos de tabelas encontradas, São eles:

- Enciclopédia digital;
- Portal de notícias;
- Loja virtual; e
- Portal de comparação de filmes.

### Configurações

As configurações pertinentes à extração utilizadas pelo *crawler* para cada um dos domínios estão na Tabela 8.

Configuração	Valor
<b>Número mínimo de linhas</b>	3
<b>Número mínimo de colunas</b>	3
<b>Timeout</b>	10000

Tabela 8 - Configuração do *crawler* para experimentos

A ferramenta *WT2SQL* foi executada uma vez para cada domínio, resultando em 2.715 tabelas extraídas. Deste total, foram selecionadas 25 tabelas de cada domínio de forma aleatória e por fim, uma verificação manual da amostragem foi realizada para o cálculo das métricas.

### 5.1 MÉTRICAS E VARIÁVEIS

Com o objetivo de avaliar a ferramenta proposta, foram utilizadas métricas de Recuperação da Informação: Precisão, Revocação e Medida-F (RIJSBERGEN,1979).

Para o cálculo de cada uma delas foram utilizadas as seguintes variáveis.

- *headerSQL*: Conjunto de rótulos de cabeçalho gerados pela ferramenta e adicionados no *script* de criação;
- *headerHTML*: Conjunto de rótulos de cabeçalho existentes na página;
- *insertSQL*: Conjunto de dados gerados para inserção pela ferramenta;
- *insertHTML*: Conjunto de dados existentes na página.

$$headerPrecision = \frac{|headerSQL \cap headerHTML|}{headerSQL}$$

**Equação 1 - Equação da precisão do cabeçalho**

$$headerRevocation = \frac{|headerSQL \cap headerHTML|}{headerHTML}$$

**Equação 2 - Equação da revocação do cabeçalho**

$$headerFMeasure = \frac{2 \times headerPrecision \times headerRevocation}{headerPrecision + headerRevocation}$$

**Equação 3 - Equação da medida F do cabeçalho**

$$insertPrecision = \frac{|insertSQL \cap insertHTML|}{insertSQL}$$

**Equação 4 - Equação da precisão da inserção**

$$insertRevocation = \frac{|insertSQL \cap insertHTML|}{insertHTML}$$

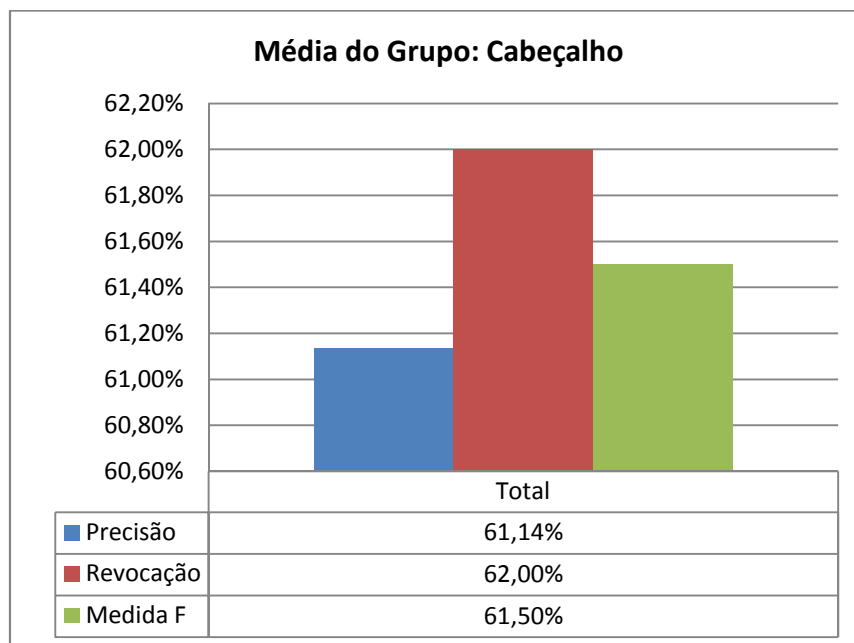
**Equação 5 - Equação da revocação da inserção**

$$insertFMeasure = \frac{2 \times insertPrecision \times insertRevocation}{insertPrecision + insertRevocation}$$

**Equação 6 - Equação da medida F da inserção**

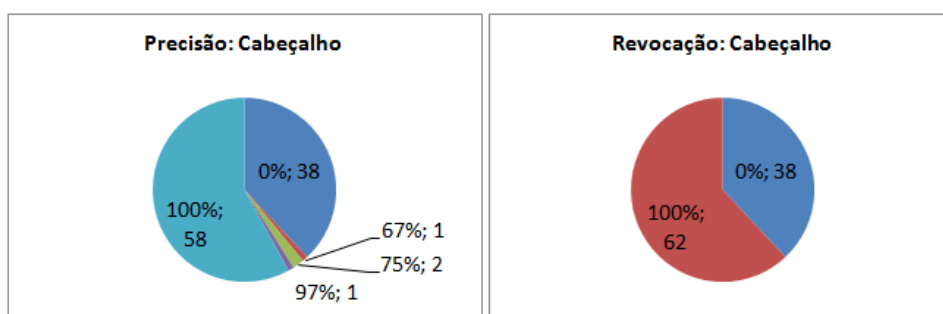
## 5.2 RESULTADOS

Abaixo são apresentados os principais resultados encontrados com os experimentos dos dois grupos analisados – cabeçalho (*header*) e dados (*insert*). A Figura 21 contém as médias das métricas para o grupo de cabeçalhos. A precisão atingiu 61% e a revocação 62%. Isso significa que mais da metade das páginas foi extraída com sucesso.



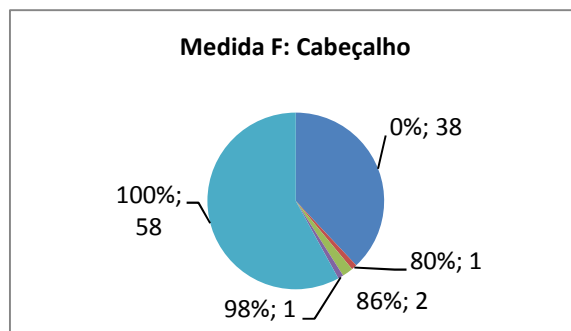
**Figura 21 - Resultado geral para cabeçalhos**

Já as Figura 22 e Figura 23 ilustram os resultados detalhados para cada medida, com isso pode-se concluir que 58 cabeçalhos de tabelas geradas tiveram 100% de precisão, enquanto 38 foram gerados de forma incorreta, representados no gráfico com 0% de precisão. Isto ocorreu porque os cabeçalhos dessas tabelas foram definidos a partir de atributos de estilo HTML (negrito, itálico, cor da fonte) e dessa maneira não foram identificados corretamente. Além disso, é possível observar a revocação e a medida F separadamente.



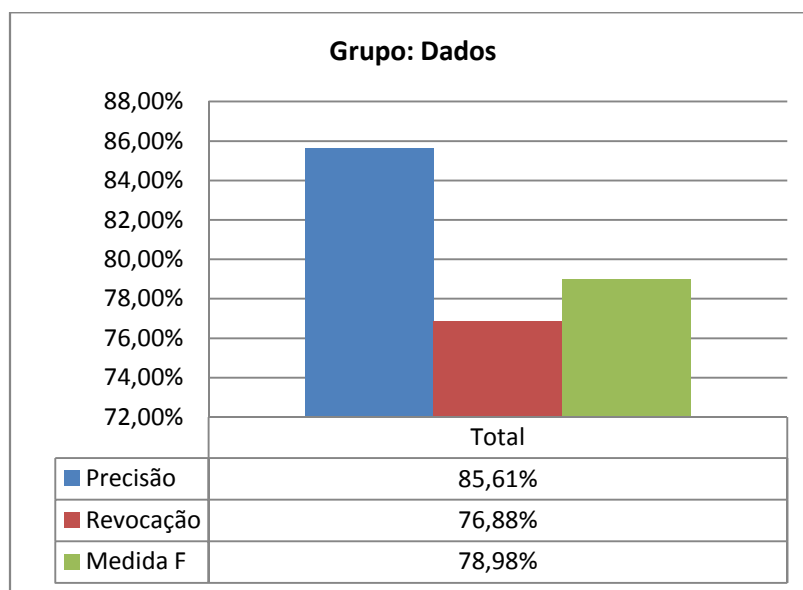
**Figura 22 - Precisão e Revocação para cabeçalhos**



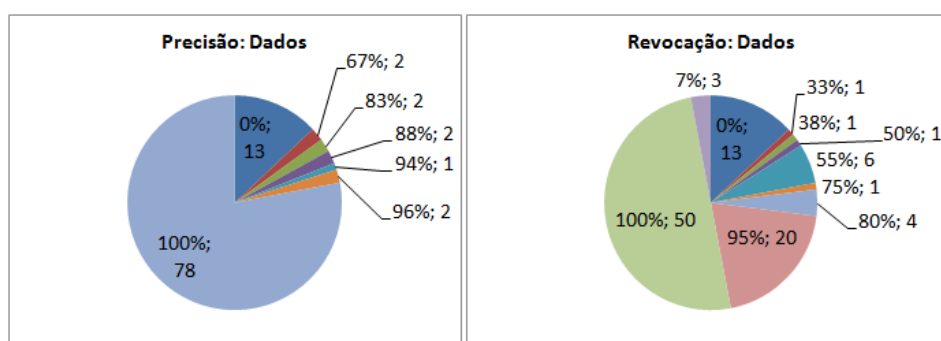


**Figura 23 - Medida F para cabeçalhos**

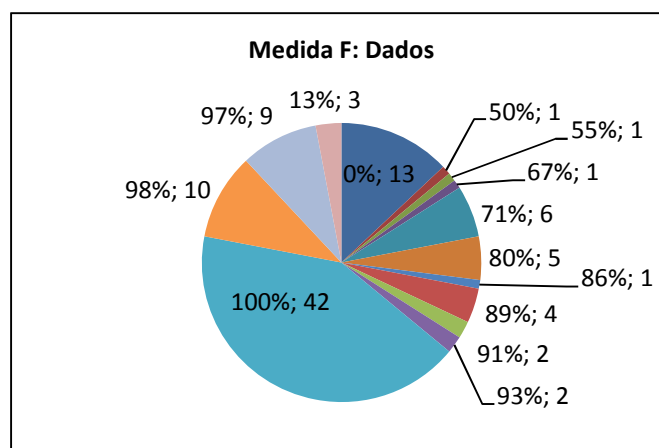
A Figura 24 mostra para o grupo dos dados que os resultados foram melhores, com a precisão atingindo 85% e a revocação 76% na média. As Figuras Figura 25 e Figura 26 detalham a precisão, revocação e medida F para este grupo – de acordo com a precisão, 78 tabelas geradas tiveram os dados extraídos corretamente. Contudo, 13 tabelas não tiveram seus dados extraídos de forma correta devido ao erro na construção do cabeçalho.



**Figura 24 - Resultado geral para dados**



**Figura 25 - Precisão e Revocação para dados**



**Figura 26 - Medida F para dados**

Uma análise complementar sobre cada domínio separadamente mostrou que os melhores resultados de precisão (para cabeçalhos e dados) são da enciclopédia digital e do portal de comparação de filmes, ambos chegando a 98%. Já os piores resultados pertencem ao portal de notícias e a loja virtual, não ultrapassando 60%. Essa diferença ocorre porque os dois primeiros possuem tabelas simples e bem uniformizadas, já os dois últimos apresentam tabelas não padronizadas e com estruturas complexas.

## 6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Os dados estruturados permeiam toda a Internet e seu uso é constante em diversas situações cotidianas. As *WebTables* são apenas um forma desse tipo de dados e possuem representações diferentes na *Web*. Com base nelas é possível recuperar muitos dados e informações úteis para as mais distintas situações, como por exemplo, a construção de bases de conhecimento e o refinamento de pesquisas.

Este trabalho abordou questões relacionadas à extração de tabelas na Internet, desde o reconhecimento de sua estrutura, cabeçalho e dados para a resolução do problema proposto: A extração de *WebTables*. Para isso foi desenvolvido uma ferramenta (WT2SQL) capaz de navegar na Internet e executar os procedimentos necessários para reconhecer a estrutura de uma tabela e extrair os dados contidos nela para um modelo relacional.

Para a realização do descrito acima, foi necessário o cumprimento dos seguintes pontos:

- Desenvolvimento do *Crawler* WT2SQL para a navegação automática nas páginas HTML utilizando a linguagem Java;
- Definição e implementação de heurística para reconhecimento de tabelas de dados, excluindo tabelas de formatação, levando em consideração a quantidade de linhas e colunas assim como tabelas aninhadas;
- Definição e implementação de heurística para reconhecimento de estruturas de tabelas (horizontais, verticais, multivaloradas, etc.) para aumentar a precisão da extração de dados;
- Criação de arquivos SQL para inserção em um SGBD para futuro tratamento dos dados;
- Realização de experimentos para verificar a precisão da ferramenta em 100 casos analisados manualmente.

Comparando aos trabalhos relacionados apresentado no Capítulo 3 com a ferramenta desenvolvida, é possível verificar que a ferramenta desenvolvida não precisa de conhecimento anterior a extração para ser considerada eficiente e automática ao mesmo tempo. Assim como as ferramentas Octopus e Probase, o resultado do WT2SQL são tabelas formatadas. Essa comparação é apresentada na Tabela 9.

Ferramenta	Automação	Resultado da Extração	Retro alimentação	Integração de dados	Técnica
Octopus	Sim	Tabelas formatadas	Não	Sim	Aprendizado de máquina
Probase	Sim	Tabelas formatadas	Sim	Não	Ontologia
Web Table Extractor	Não	Área de transferência	Não	Não	Cópia de dados
TableTools2	Não	Área de transferência	Não	Não	Cópia de dados
Web Page Table Extractor	Não	Arquivo HTML	Não	Não	Cópia de dados
WT2SQL	Sim	Tabela Formatada	Não	Não	Heurística

**Tabela 9 - Tabela comparativa com WT2SQL**

Adicionalmente, o trabalho contribuiu para o artigo *Web Table Taxonomy and Formalization* aceito em 2013 no *SIGMOD Record* e aguarda a publicação. Este artigo foi um trabalho conjunto de Larissa L. Lautert e Carina F. Dorneles (LAUTERT; SCHEIDT; DORNELES, 2013).

### Trabalhos Futuros

A seguir são elencados alguns pontos que não foram abordados neste trabalho, mas devem ser considerados para trabalhos futuros. Sendo eles:

- Identificar cabeçalhos de tabelas levando em consideração atributos de estilo na página HTML (negrito, itálico, cor da fonte, cor do fundo, etc.) como, por exemplo, a Tabela 10 abaixo que apresenta os rótulos das colunas em negrito.

Nome	Sujeito A	Sujeito B	Sujeito C	Sujeito D	Sujeito E
<b>Idade</b>	28	59	45	39	41
<b>Profissão</b>	Professor	Médico	Policial	Executivo	Advogado

**Tabela 10 - Tabela com cabeçalho em negrito**

- Identificar cabeçalhos levando em consideração estilos definidos em arquivos CSS. Diferencia-se do ponto anterior porque a interpretação de arquivos de estilo é diferente da interpretação da árvore DOM de um arquivo HTML.
- Extrair dados levando em consideração sufixos repetidos em diversas colunas, caracterizando uma nova coluna a ser criada no arquivo SQL, como o observado na Tabela 11, onde o sufixo PT e GE aparecem várias vezes. A extração deve criar uma nova coluna para abrigar e normalizar esses dados, resultando na

Tabela 12.

Word	Meaning
Table	PT: Tabela, GE: Tabelle
Extraction	PT: Extração, GE: Gewinnung

**Tabela 11 - Tabela com sufixos repetidos**

Word	Meaning	Título de Rótulo Gerado
Table	Tabela	PT
Table	Tabelle	GE
Extraction	Extração	PT
Extraction	Gewinnung	GE

**Tabela 12 - Tabela de sufixos normalizada**

- Inferir tipos de dados para as tabelas extraídas e a possível eleição de chaves primárias. No estado atual, todas as colunas são criadas com o tipo TEXT.
- Realizar o processo de navegação e extração de forma paralela para aumentar o desempenho da ferramenta. No estado atual, o processo de navegação é paralisado para realizar a extração de dados quando for possível.

## REFERÊNCIAS

APACHE. **Apache License v2.0**. Disponível em <<http://www.apache.org/licenses/LICENSE-2.0.html>>. Acesso em 11 de 25 de outubro de 2012.

CALDAS, P. de O. **Geração de Regras de Extração de Dados em Páginas HTML**. 2003. 63 f. Dissertação (Mestrado) – Curso de Ciência da Computação, Instituto de Informática, UFRS, Porto Alegre, 2003.

CAFARELLA, M. J.; HALEVY, A. Y.; WANG, Z. D.; WU, E; ZHANG, Y. **WebTables: Exploring the Power of Tables on the Web**. VLDB, p538-549, 2008.

CAFARELLA, M. J.; HALEVY, A.; KHOUSSAINOVA, N. **Data Integration for the Relational Web**. VLDB, p1090-1101, 2009.

CHO, J. **Crawling the Web: Discovery and Maintenance of Large-Scale Web Data**. 2001. 172 f. Tese (Doutor) - Curso de Ciência da Computação, Departamento de Ciência da Computação, Stanford University, Stanford, 2002.

CRESTAN, E.; PANTEL, P. **Web-Scale Table Census and Classification**. WSDM Hong Kong – China, 2011.

ECLIPSE. **The Eclipse Foundation**. Disponível em <<http://www.eclipse.org>>. Acesso em 18 de setembro de 2012.

GOOGLE. **Googlebot**. Disponível em <<http://support.google.com/webmasters/bin/answer.py?hl=en&answer=182072>>. Acesso em 10 de dezembro de 2012.

HTMLUNIT. **The HtmlUnit Project**. Disponível em <<http://htmlunit.sourceforge.net/>>. Acesso em 10 de setembro de 2012.

LAENDER, A. H. F.; RIBEIRO-NETO, B. A.; da SILVA, A. S.; TEIXEIRA, J. S. **A Brief Survey of Web Data Extraction Tools**. SIGMOD Record, New York, v. 31, n. 2, Junho, 2002.

LAUTERT, L. R.; SCHEIDT, M. M.; DORNELES, C. F. **Web Table Taxonomy and Formalization**. SIGMOD Records, 2013 (aceito).

LIMAYE, G.; SARAWAGI, S.; CHAKRABARTI, S. **Annotating and Searching Web Tables Using Entities, Types and Relationships**. VLDB Endowment, Vol. 3, No. 1, 2010.

LIN, C. X.; ZHAO, B.; WENINGER, T.; HAN, J.; LIU, B. **Entity Relation Discovery from Web Tables and Links**. University of Illinois, 2010.

MIAO, G.; TATEMURA, J.; HSIUNG, W.; SAWIRES, A.; MOSER, L. E. **Extrating Data Records from Web Using Tag Path Clustering**. World Wide Web Conference. Madri – Espanha, 2009.

OOLUTION TECHNOLOGIES. **Web Page Table Extractor**. Disponível em: <<http://wpte.oolutiontech.com>>. Acesso em 21 de março de 2013.

ORACLE. **Java Technology**. Disponível em <<http://www.oracle.com/technetwork/java/index.html>>. Acesso em 23 de setembro de 2012.

PORTAL BRASIL. **Taxa de juros da poupança**. Disponível em <[http://www.portalbrasil.net/poupanca\\_mensal.htm](http://www.portalbrasil.net/poupanca_mensal.htm)>. Acesso em 10 de maio de 2013.

RIJSBERGEN, C. J. Van. **Information Retrieval**. 2a Edição – Butterworths, Londres, 1979.

ROBOTSTXT. **The Robots Exclusion Protocol**. Disponível em: <<http://www.robotstxt.org>>. Acesso em 16 de setembro de 2012.

TABLETOOLS2. **TableTools2**. Disponível em: <<https://addons.mozilla.org/en-us/firefox/addon/tabletools2/>>. Acesso em 22 de março de 2013.

WANG, J., WANG, H., WANG, Z., ZHU, K. Q. **Understanding Tables on the Web**. ER, p141-155, 2012.

WTE. **Web Table Extractor**. Disponível em: <<http://mozilla2.software.informer.com/download-mozilla-firefox-html-table-extractor/>>. Acesso em 21 de março de 2013.

## APÊNDICE – TABELA EXEMPLO E ARQUIVO SQL

Tabela Exemplo

Naipes	Nome	Observação
<b>Reis</b>		
Ouros	Júlio César	
Espadas	Davi	Rei israelita
Copas	Carlos Magno	
Paus	Alexandre, o Grande	
<b>Damas</b>		
Ouros	Raquel	Esposa de Jacó
Espadas	Atena	Deusa grega
Copas	Judite	Personagem bíblica católica
Paus	Elisabeth I	
<b>Valetes</b>		
Ouros	Heitor	Príncipe de Tróia
Espadas	Hogier	Primo de Carlos Magno
Copas	La Hire (Étienne de Vignolles)	
Paus	Sir Lancelot	

### SQL Correspondente à Tabela Exemplo

```
CREATE TABLE table_17062013153022000 (
  column_naipes TEXT,
  column_nome TEXT,
  column_observacao TEXT,
  column_genereted TEXT
);
```

```
INSERT INTO table_17062013153022000(column_naipes, column_nome,
column_observacao, column_genereted)
VALUES ('Ouros', 'Júlio César', NULL, 'Reis');
```

```
INSERT INTO table_17062013153022000(column_naipes, column_nome,
column_observacao, column_genereted)
VALUES ('Espada', 'Davi', 'Rei israelita', 'Reis');
```

```
INSERT INTO table_17062013153022000(column_naipes, column_nome,
column_observacao, column_genereted)
VALUES ('Copas', 'Carlos Magno', NULL, 'Reis');
```

```
INSERT INTO table_17062013153022000(column_naipes, column_nome,
column_observacao, column_genereted)
VALUES ('Paus', 'Alexandre, o Grande', NULL, 'Reis');
```

```
INSERT INTO table_17062013153022000(column_naipes, column_nome,
column_observacao, column_genereted)
```



```
VALUES ('Ouros', 'Raquel', 'Esposa de Jacó', 'Damas');
```

```
INSERT INTO table_17062013153022000(column_naipe, column_nome,  
column_observacao, column_genereted)  
VALUES ('Espada', 'Atena', 'Deusa grega', 'Damas');
```

```
INSERT INTO table_17062013153022000(column_naipe, column_nome,  
column_observacao, column_genereted)  
VALUES ('Copas', 'Judite', 'Personagem bíblica católica', 'Damas');
```

```
INSERT INTO table_17062013153022000(column_naipe, column_nome,  
column_observacao, column_genereted)  
VALUES ('Paus', 'Elisabeth I', NULL, 'Damas');
```

```
INSERT INTO table_17062013153022000(column_naipe, column_nome,  
column_observacao, column_genereted)  
VALUES ('Ouros', 'Heitor', 'Príncipe de Tróia', 'Valetes');
```

```
INSERT INTO table_17062013153022000(column_naipe, column_nome,  
column_observacao, column_genereted)  
VALUES ('Espada', 'Hogier', 'Primo de Carlos Magno', 'Valetes');
```

```
INSERT INTO table_17062013153022000(column_naipe, column_nome,  
column_observacao, column_genereted)  
VALUES ('Copas', 'La Hire (Étienne de Vignolles)', NULL, 'Valetes');
```

```
INSERT INTO table_17062013153022000(column_naipe, column_nome,  
column_observacao, column_genereted)  
VALUES ('Paus', 'Sir Lancelot', NULL, 'Valetes');
```