PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Luis Gustavo Almeida

ALUMNI Tool:

Recuperação de dados pessoais na Web em redes sociais autenticadas

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Informática da PUC-Rio.

Orientador: Prof. Marco Antonio Casanova

Rio de Janeiro

Janeiro de 2018

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Luis Gustavo Almeida

ALUMNI Tool:

Recuperação de dados pessoais na Web em redes sociais autenticadas

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Marco Antonio Casanova
Orientador
Departamento de Informática – PUC-Rio

Prof. Antonio Luz Furtado
Departamento de Informática – PUC-Rio

Profª. Simone Diniz Junqueira Barbosa
Departamento de Informática – PUC-Rio

Prof. Márcio da Silveira Carvalho
Coordenador Setorial do Centro
Técnico Científico – PUC-Rio

Rio de Janeiro, 31 de Janeiro de 2018

Luis Gustavo Almeida

Luis Gustavo Almeida graduou-se em Engenharia Civil pela Unicamp em 2001. Desde então trabalhou como programador freelancer e analista de banco de dados também como autônomo. Possui interesse acadêmico e profissional nas áreas ligadas à programação Web, PHP, HTML, Coleta de Dados, Data Warehouse e Big Data.

Ficha Catalográfica

Almeida, Luis Gustavo

ALUMNI Tool : recuperação de dados pessoais na Web em redes sociais autenticadas / Luis Gustavo Almeida ; orientador: Marco Antonio Casanova. – 2018.

123 f. : il. color. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2018.

Inclui bibliografia

1. Informática – Teses. 2. Web crawling. 3. Selenium. 4. Scraping. 5. Big data. 6. Robôs de busca. I. Casanova, Marco Antonio. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD:004

Agradecimentos

Gostaria de agradecer à PUC-Rio, por ter sido para mim muito mais do que um local de trabalho ao longo dos últimos anos. Agradeço também ao CNPq pela bolsa de estudos durante o segundo ano como aluno de mestrado.

Agradeço a todas as pessoas que contribuíram de alguma forma para a minha formação, em especial a minha família e amigos. Além da minha irmã Tathiana Maria Almeida que é o meu braço direito, meu irmão Thiago Henrique Almeida, meu sobrinho Igor Almeida que me tem como referência e a minha namorada Ruth Ferreira Vieira.

Também gostaria de dedicar este trabalho ao meu pai, José Valdemar Almeida, e à minha mãe Silvia Maria Lolo Almeida, sou metade de cada um deles, obrigado pai e mãezinha por me fazerem.

Outras pessoas a quem também gostaria de agradecer foi a comunidade de estrangeiros que conheci e ficaram meus amigos durantes este período, os funcionários do departamento desde aqueles que preparam nosso café aos da secretaria que organizam nossa papelada nesta bem-conceituada Universidade, que aliás tem em seu campus um dos lugares mais agradáveis da cidade do Rio de Janeiro.

Gostaria de fazer os meus sinceros agradecimentos ao meu orientador Marco Antônio Casanova. Agradeço a ele por toda a paciência que sempre teve comigo. O Casa é um destes seres humanos especiais cujas ações refletem os melhores significados que podemos extrair da palavra “humanidade”. Obrigado por tudo Casa!

Também não posso esquecer dele, daquele todo poderoso que está lá em cima nos céus, olhando por mim, ele que é o cara com quem converso todas as noites e o cara que me fortalece quando estou para baixo. Graças a ele tenho saúde, paz interior e não pensei 2x ao seguir o conselho dele de largar tudo em São Paulo e correr para os braços da cidade maravilhosa, mergulhar de cabeça nestas águas abençoadas. **Obrigado Deus!!!**

Resumo

Almeida, Luis Gustavo; Casanova, Marco Antonio **ALUMNI Tool: Recuperação de dados pessoais na Web em redes sociais autenticadas.** Rio de Janeiro, 2018. 123p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

O uso de robôs de busca para coletar informações para um determinado contexto sempre foi um problema desafiante e tem crescido substancialmente nos últimos anos. Por exemplo, robôs de busca podem ser utilizados para capturar dados de redes sociais profissionais. Em particular, tais redes permitem estudar as trajetórias profissionais dos egressos de uma universidade, e responder diversas perguntas, como por exemplo: Quanto tempo um ex-aluno da PUC-Rio leva para chegar a um cargo de relevância? No entanto, um problema de natureza comum a este cenário é a impossibilidade de coletar informações devido a sistemas de autenticação, impedindo um robô de busca de acessar determinadas páginas e conteúdos. Esta dissertação aborda uma solução para capturar dados, que contorna o problema de autenticação e automatiza o processo de coleta de dados. A solução proposta coleta dados de perfis de usuários de uma rede social profissional para armazenamento em banco de dados e posterior análise. A dissertação contempla ainda a possibilidade de adicionar diversas outras fontes de dados dando ênfase a uma estrutura de armazém de dados.

Palavras-Chave

Web Crawling; Selenium; Scraping; Big Data; Robôs de Busca; Bots; Coleta de dados; Recuperação de informação; Web Spider; Redes Sociais.

Abstract

Almeida, Luis Gustavo; Casanova, Marco Antonio (Advisor). **ALUMNI Tool: Information recovery of personal data on the Web in authenticated social networks.** Rio de Janeiro, 2018. 123p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The use of search bots to collect information for a given context has grown substantially in recent years. For example, search bots may be used to capture data from professional social networks. In particular, such social networks facilitate studying the professional trajectory of the alumni of a given university, and answer several questions such as: How long does a former student of PUC-Rio take to arrive at a management position? However, a common problem in this scenario is the inability to collect information due to authentication systems, preventing a search robot from accessing certain pages and content. This dissertation addresses a solution to capture data, which circumvents the authentication problem and automates the data collection process. The proposed solution collects data from user profiles for later database storage and analysis. The dissertation also contemplates the possibility of adding several other sources of data giving emphasis to a data warehouse structure.

Keywords

Web Crawling; Selenium; Web Scraping; Big Data; Search Engine; Bots; Data Collect; Social Networks; Web Spider; Alumni.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Quadros

Lista de Abreviações

AJAX – Asynchronous Javascript and XML

API – Application Programming Interface

BD – Banco de Dados

BFS – Breadth First Search

BI – Business Intelligence

CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico

CT&I – Ciência Tecnologia e Inovação

DNS – Domain Name Services

DOM – Document Object Model

DW – Data Warehouse

EI – Extração da Informação

FoaF – Friend of a Friend

GUI – Graphical User Interface

HTML – HyperText Markup Language

HTTP – HyperText Transfer Protocol

IA – Inteligência Artificial

ID - IDentifier

IP – Internet Protocol

JSON – JavaScript Object Notation

KDD – Knowledge Discovery in Databases

LAMP – Linux Apache MySQL PHP

MVC – Model View Controller

NLP – Natural Language Processor

OLAP – OnLine Analytical Processor

OSN – Organization Social Network

PCRE – Perl Compatible Regular Expressions

PDO – PHP Data Objects

PHP – PHP Hypertext Preprocessor

PR – PageRank

PSN – Professional Social Networks

RegEx – Regular Expressions

REST – Representational State Transfer

RIA – Rich Internet Application

RI – Recuperação da Informação

SEO – Search Engine Optimization

SGBD – Sistema Gerenciador de Banco de Dados

SPAM – Sending and Posting Advertisement in Mass

TCP/IP – Transfer Control Protocol/Internet Protocol

TIC – Tecnologia da Informação e Comunicação

T-SQL – Transact Structured Query Language

URI – Uniform Resource Identifier

URL – Uniform Resource Locator

URN – Uniform Resource Name

UX – User eXperience

WAMP – Windows Apache MySQL PHP

WSD – WebServices Discovery

1.
Introdução

1.1
Motivação

Capturar dados de valor qualitativo na Web tem sido uma tarefa complexa devido ao fato de não haver, na maioria das vezes, uma fonte de dados confiável, disponível e aberta ao público. Para realizar a tarefa de coleta de dados na Web é recomendada a construção de um rastreador ou *crawler*. Miller e Bharat [46] definem outros sinônimos para rastreador, como robôs, bots e spiders.

Um rastreador é um programa que varre a Web de maneira não assistida e processa as páginas encontradas. Um rastreador é um componente importante de muitos serviços da Web, mas seu projeto não está bem documentado na literatura [47]. O rastreamento de dados da Web consiste em visitar os servidores da Web usando um mecanismo automático para coletar documentos públicos. De acordo com MYLLYMAKI [18], um rastreador deve solicitar e armazenar documentos de servidores Web, extrair links dos documentos coletados e agendar o próximo passo de rastreamento usando os links extraídos.

Alumnus é uma palavra em latim que significa ex-aluno e *alumni* é o correspondente plural. Estudar e minerar alumni tem sido uma questão relevante e muito abordada em diversos meios, inclusive acadêmico e corporativo. No trabalho de DEY [71] verifica-se que nas décadas de 1990 e 2000, o boom das 'pontocom' aumentou a concorrência para os candidatos nos campi universitários, surgiram serviços de carreira que facilitaram a relação entre estudantes e empregadores, assim novas tecnologias de informação emergiram e as redes sociais começaram a redefinir a forma como os alunos fazem sentido da sua experiência e se conectam aos empregadores. Isso contribuiu para um maior interesse em estudar alumni.

De acordo com LOPS [17], a proliferação de redes sociais gerou um enorme volume de dados útil para aprender interesses e gostos dos usuários. Isso abriu espaço onde algumas empresas começaram a concentrar informações

referentes a algum contexto referente ao meio corporativo. Essas empresas procuram vender estas informações sob diversas formas de produtos, tais como acesso pago, publicidade paga, etc. Como consequência, apenas parte destas informações permanecem de domínio público, restando a maior parte de tal conteúdo ser disponibilizado somente a usuários registrados.

A ideia central desta dissertação consiste em recuperar dados da Internet que traduzem a trajetória profissional de ex-alunos da PUC-Rio[1] para os mais variados fins, inclusive respondendo à pergunta de quanto tempo em média leva um ex-aluno de graduação a atingir um cargo de relevância. O caso de uso principal inclui o LinkedIn[2], com a possibilidade de se anexar outras fontes de dados.

Para acessar as páginas do Website do LinkedIn é desejável ser um usuário autenticado. Para isso é imprescindível possuir login e senha de acesso. Este fato impede a ação de um robô de busca tradicional, pois para qualquer URL de pesquisa do LinkedIn o servidor irá redirecionar o navegador para página de login (Figura 4).

Por se tratar de uma empresa em que um dos principais ativos é sua base de dados o LinkedIn pode em seus entendimentos procurar barrar robôs de busca de efetuar coleta de dados em seu Website. Porém, a Suprema Corte dos EUA [60,61,62,63] recentemente deu ganho de causa a uma empresa do ramo de Inteligência Artificial (IA) que se valia da coleta de perfis de usuários do LinkedIn (Figura 1). Conforme argumenta GREENE [60] “Uma vez que os dados foram divulgados publicamente pelos usuários que o publicaram, o LinkedIn não conseguiu demonstrar sua propriedade em um grau que lhe confere o direito de impedir que outros acessem. Qualquer um poderia, teoricamente, clicar em cada perfil e usar uma caneta e papel para copiar todas as informações se tivessem tempo e mão-de-obra suficientes para tal, e é por isso que essas tarefas são feitas usando um algoritmo que reúne e classifica dados”.

Este caso está se configurando como um grande teste de quanto controle um Website de mídia social pode exercer sobre a informação que seus usuários consideraram pública. O LinkedIn argumentou que, mesmo que seus usuários

[1] http://www.puc-rio.br

[2] http://www.linkedin.com

desejem que pelo menos algumas de suas informações estejam disponíveis publicamente, o LinkedIn tem o direito de bloquear o acesso aos dados. FRANKEL [61] argumenta "O LinkedIn se refere a si mesmo como uma ‘comunidade’ e expressa-se como um lugar para se encontrar, trocar idéias, aprender. Quando um proprietário de uma propriedade privada abre sua propriedade ao público, os princípios constitucionais de discurso livre, aberto e o acesso à informação devem ser plenamente respeitados".

A desaceleração econômica de 2008 criou um ambiente ideal para outra mudança de paradigma nos serviços da Universidade, e por essa pressão reinventaram-se os processos de emprego e carreira, pondera DEY [71]. Muitas pesquisas vêm sendo realizadas para estudar egressos de universidades. Porém, sempre se valendo de métodos de coleta de dados pouco automatizados.

Para GONÇALVES [29] é necessária a definição de um método eficiente para obter dados profissionais de alunos de graduação. De acordo com LOUSADA & MARTINS [74], os métodos tradicionais empregados para reunir informações de alunos não têm uma boa taxa de resposta, o que leva a interromper a coleta de tais informações. O problema de gerenciar a informação dos ex-alunos é recorrente em diversas universidades.

Percebe-se no meio acadêmico através de diversos trabalhos consultados um aspecto pouco informático, quando não manual no aspecto de Recuperação da Informação (RI). Sem critérios mais profundos, ou estratégias de crawling acerca da recuperação de informação de ex-alunos.

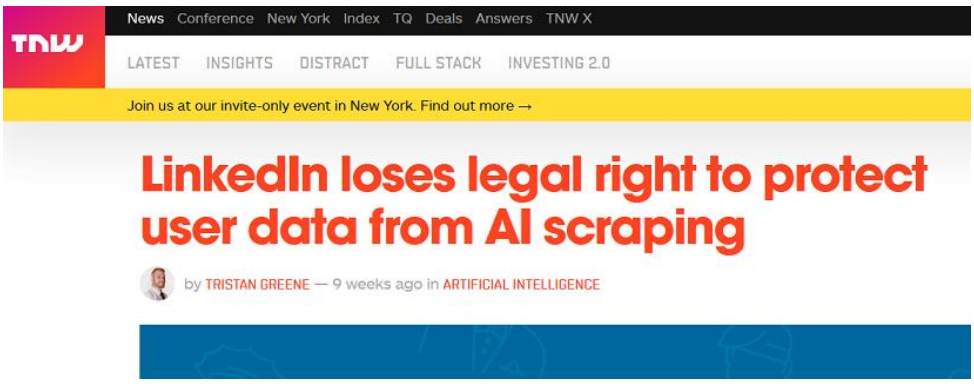


Figura 1: LinkedIn perde ação na justiça americana.

1.2
Domínio de Estudo

Uma componente chave da arquitetura dos motores de busca da Web é o rastreador. São programas usados para percorrer automaticamente a Web, recuperando páginas para criar um índice pesquisável de seu conteúdo [13]. Os rastreadores recebem como entrada um conjunto de páginas semente e obtêm de forma recursiva novas páginas, através de seus links de saída. O autor define a Web escondida (*Hidden Web* ou *Deep Web*) como um contexto onde as informações são exibidas de forma dinâmica, sobretudo através de formulários de pesquisa, diferentemente da Web estática, onde nesta os documentos são recuperados integralmente. A Web escondida também é conhecida pelo aspecto dos formulários implícitos nas páginas que exibem alguns atributos do banco de dados através dos campos do formulário, sobre os quais o usuário especifica filtros e então submete consultas ao Banco de Dados (BD) [11]. A *Deep Web* é a Web cujo esquema e conteúdo não estão completamente visíveis ao usuário [16].

Rastrear a *Deep Web* é um problema muito desafiante por dois motivos. Primeiro é a questão da escala: um estudo recente estima que o tamanho do conteúdo disponível através desses bancos de dados pesquisáveis online é cerca de 400 a 500 vezes maior que o tamanho da Web estática [12]. Segundo, o acesso a esses bancos de dados é fornecido apenas por meio de interfaces de pesquisa restritas, destinadas a serem utilizadas por seres humanos. Portanto, treinar um rastreador para usar essa interface restrita para extrair conteúdo relevante é um problema não trivial.

ALVAREZ [13] também concorda que rastreadores não podem alcançar a maioria das informações contidas na Web, pois uma grande quantidade de informações está escondida por trás das formas de consulta de BD online e/ou é gerada dinamicamente por tecnologias como o JavaScript.

O fundamento principal deste trabalho é automatizar o processo de coleta de dados mantendo a sessão ativa do usuário e adotando uma estratégia de busca que maximize a relação custo-benefício (será abordado no capítulo 4). Nosso foco são as redes sociais profissionais (PSN) e assim capturar dados pertinentes a pesquisa, mais precisamente o perfil dos históricos profissionais e acadêmicos

destes indivíduos. Estes perfis se encaixam dentro da definição de Web escondida pois o resultado destes documentos é exibido como consequência de uma consulta em um formulário de pesquisa.

PENA [31] realizou estudos sobre as dificuldades encontradas na gestão de ex-alunos de instituições educacionais no Brasil. Os autores apresentam o conceito de ex-alunos na esfera brasileira.

Em um trabalho de pesquisa de egressos da USP [114] foram recuperados 185 alumni para diversas análises através da base de dados Lattes.

Também no Departamento de Informática da Universidade Federal de Viçosa foi feita uma pesquisa sobre o perfil profissional dos ex-alunos do programa de graduação [115].

Para abordar este problema devemos escolher a rede social profissional mais significativa da atualidade, no caso o LinkedIn. Neste trabalho será também levado em consideração o curriculum Lattes[3], que disponibiliza arquivos XML contendo as informações sobre alumni e é concedido somente a entidades acadêmicas. Estas duas fontes de dados serão especificadas adiante.

A possibilidade de se analisar e minerar a evolução profissional deste contingente de pessoas abriria uma oportunidade de se responder algumas questões pertinentes a evolução da carreira profissional em conformidade com a carreira acadêmica. Estudar o perfil dos egressos de uma universidade permite responder e analisar muitas dúvidas pertinentes à carreira posterior que estes alumni irão vivenciar, a análise do mercado de trabalho com suas lacunas que serão preenchidas por esta força de trabalho ingressante, etc. E assim esse estudo pode ser útil para as universidades aprimorarem seus cursos.

1.3
Objetivos

Analisar o perfil profissional do LinkedIn (Figura 2) pode ajudar as universidades a investir em novas linhas educacionais ou de pesquisa, como a criação de novas aulas ou grupos de pesquisa para assuntos específicos com base na tendência da carreira dos ex-alunos [15].

[3] http://lattes.cnpq.br

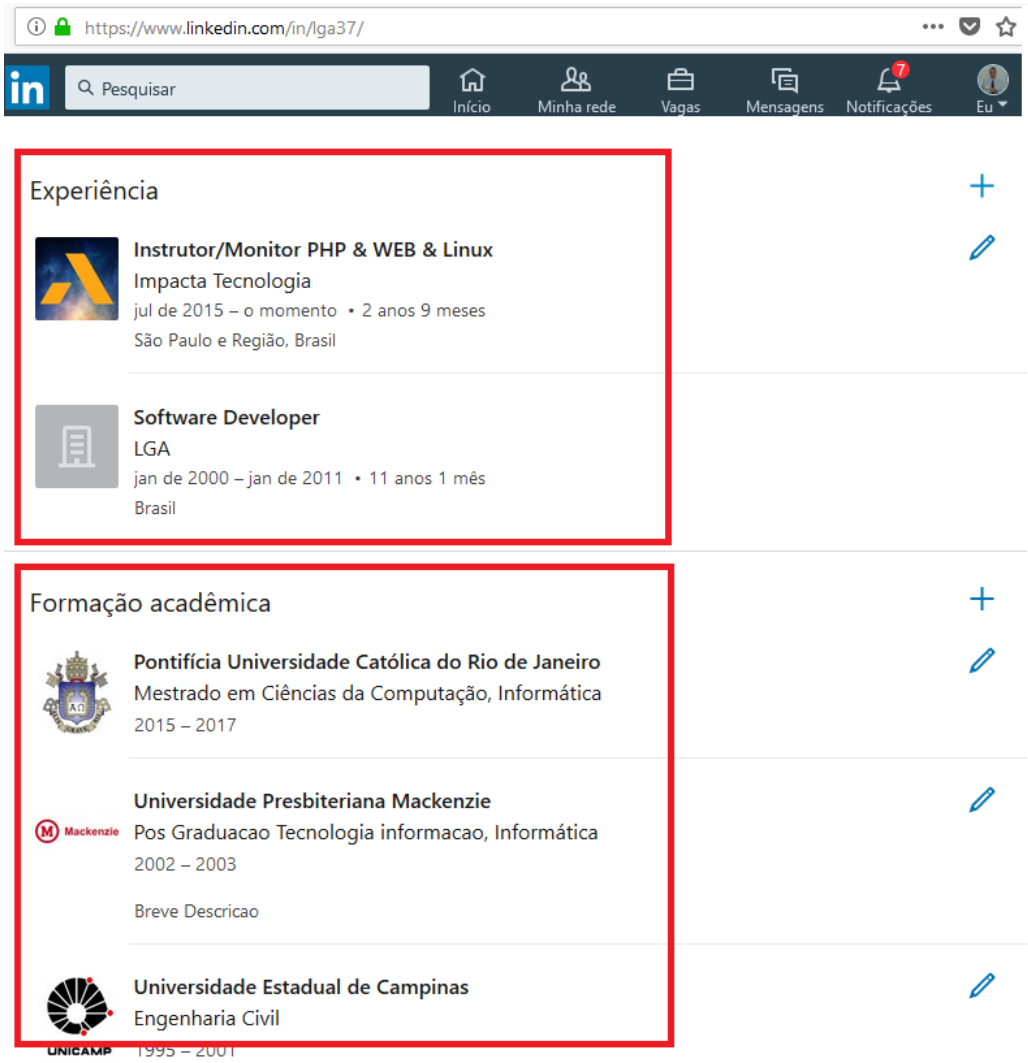


Figura 2: Exemplo de perfil de usuário da rede social LinkedIn.

Esta dissertação pretende recuperar dados, minerar, processar e dentre várias análises responder à seguinte pergunta: Quanto tempo leva em média para que um aluno de graduação da PUC-Rio chegue a um cargo de relevância[4] ? Outras perguntas também podem ser examinadas, tais como:

- Quais são os cursos ou áreas mais procurados? Assim a Universidade pode verificar se sua lista de cursos oferecidos está em conformidade com a demanda do mercado.
- Quais empresas mais contratam? Pode ser estreitado um convênio com

[4] Para este trabalho será considerado as profissões que envolvam liderar pessoas tais como gerente, diretor, CEO ou empresário.

tais empresas para assim criar oportunidades de estágio.

- Quais cursos de pós-graduação são mais procurados? Pode-se analisar a procura por programas stricto sensu versus lato sensu, e como isso contrasta com os cursos de especialização.
- Qual é a média do tempo gasto em graduação ou pós-graduação? Uma dúvida que pode surgir entre os gestores de Universidades é se as pessoas preferem uma segunda graduação ou uma pós-graduação.
- Qual o tempo médio de um emprego por empresa? De acordo com a *snapshot*[5] do LinkedIn quais empresas proporcionam uma maior longevidade no emprego ? Pode-se analisar a rotatividade dos egressos no mercado.
- Qual o tempo médio em um emprego por alumni? De acordo com o LinkedIn, por quanto tempo em média este pessoal se manteve empregado.
- Estatísticas sobre alunos desempregados.

1.4 Contribuições

A observação da trajetória dos ex-alunos serve como fonte de informações gerenciais, permitindo a tomada de decisões sobre o planejamento de cursos, arranjos didático-pedagógicos e modalidades de programas que desenvolvam uma polivalência e identidade profissional capazes de interagir e de atender às mutações do mercado de trabalho [74].

Hoje em dia, em ambientes acadêmicos, uma das principais preocupações das equipes docentes dos programas de graduação é analisar como seus alunos se adaptam à vida profissional após a graduação [29]. O presente trabalho pretende oferecer como contribuições os seguintes itens:

- Um conjunto de ferramentas capaz de extrair dados de Redes Sociais autenticadas de maneira autônoma.

[5] Snapshot é uma expressão em inglês que significa **foto instantânea ou registro instantâneo** e possui diversos significados dentro do mundo contemporâneo da informática, como o **armazenamento da condição de um banco de dados.**

- Uma ferramenta de Recuperação de Informação (RI) e um software para processar os dados.
- Possibilidade de agregar outras fontes de dados a tal ferramenta, desde que haja um denominador comum no que tange aos históricos profissionais e educacionais.
- Análise da trajetória profissional de ex-alunos, ilustrada para o caso da PUC-Rio, mas que pode ser estendida a outras Universidades.

1.5 Estrutura da Dissertação

Esta dissertação divide-se em 6 capítulos. O Capítulo 1 aborda as questões referentes às principais motivações da pesquisa. O Capítulo 2 lista diversos trabalhos relacionados aos principais aspectos científicos especialmente no que tange a crawling e scraping. O Capítulo 3 descreve alguns conceitos básicos e pertinentes ao trabalho, um background com pontos que servem de background antes de adentrar a discriminação técnica da ferramenta que é contemplada integralmente no capítulo quatro.

O Capítulo 4 pode ser considerado o coração do texto da dissertação pois aborda toda a estratégia de crawling usada para RI do trabalho. Outros dois itens importantes do trabalho, que são a inclusão de uma segunda fonte de dados e o processo de scraping também estão neste capítulo.

O Capítulo 5 ilustra alguns cenários de uso de algumas perguntas que podem ser usadas para estudar alumni e por fim responder a principal destas perguntas que é: “Quanto tempo leva um aluno de graduação a atingir um cargo de relevância?”.

O Capítulo 6 inclui as conclusões acerca dos dados recuperados e fala sobre algumas contribuições científicas que o trabalho proporciona a comunidade, trazendo inclusive um benchmark com outros trabalhos, comparando números e resultados.

O Capítulo 7 inclui as referências bibliográficas.

2.
Trabalhos Relacionados

Este trabalho aborda a área de crawlers ou rastreadores, scraping ou Extração da Informação e sobretudo Redes Sociais. Portanto vamos ilustrar alguns trabalhos da literatura em cada uma destas áreas.

2.1
Coleta Manual de Alumni

De acordo com a literatura há uma dificuldade em se automatizar o processo de coleta de dados de alumni. A UNIVILLE realizou sua pesquisa com egressos no ano de 1999 e, segundo o dirigente envolvido, a principal dificuldade foi a localização dos egressos, por conta disso o retorno da pesquisa foi muito pequeno [74].

De acordo com SOUZA [91], o objetivo principal da avaliação das Instituições de Ensino Superior é promover a melhoria do ensino e da aprendizagem. BOTH [103] discorre: A avaliação da Universidade por ex-alunos torna-se um dos componentes de fundamental importância, tendo em vista estar percebendo o aluno que passou pela instituição a real contribuição que seu curso lhe propiciou para o desempenho de suas funções e atividades no dia-a-dia. Entende-se ser o egresso um ponto expressivo de referência para a avaliação do ensino da Universidade, visto estar ele colocando em prática, profissionalmente, o aprendizado que lhe foi proposto na universidade [74].

Na FEBE, está em processo de implantação uma pesquisa baseada em entrevistas com ex-alunos e, segundo um dos dirigentes entrevistados, ainda não foi definida a frequência de aplicação dessa pesquisa. A FURB fez uma pesquisa com egressos no ano de 1998. Existe a intenção de se repetir, mas ainda não se definiu uma data. Segundo um dos dirigentes envolvidos, a grande dificuldade é a de localização dos egressos e de não existir um serviço institucionalizado [74].

2.2
Rastreadores

Os rastreadores da Web são quase tão antigos como a própria Web. O primeiro rastreador, Wanderer, foi desenvolvido por Matthew Gray na primavera de 1993, coincidindo com o primeiro navegador Web, o Mosaico NCSA [47]. Vários artigos sobre rastreamento na Web foram apresentados nas duas primeiras conferências WWW. No entanto, devido à natureza competitiva do negócio envolvendo mecanismos de busca, os projetos desses rastreadores não foram descritos publicamente. Existem duas exceções notáveis: o rastreador do Google e o rastreador Internet Archive.

O Google[6] é um mecanismo de busca em grande escala que faz uso intenso da estrutura presente no hipertexto. O Google foi projetado para rastrear e indexar a Web de forma eficiente e produzir resultados de pesquisa satisfatórios. É um sistema distribuído que usa várias máquinas para rastrear [11]. O rastreador consiste em componentes funcionais com tarefas específicas. As páginas são armazenadas por um processo de indexação, que extrai links de páginas HTML e as salva em um arquivo de disco diferente.

O Internet Archive[7] também usa várias máquinas para rastrear a Web [6]. Cada processo de rastreamento é atribuído para até 64 sites para rastrear e uma vez que uma página é baixada, o rastreador extrai os links contidos nela.

ARPPA [15] (Association Rules for Professional Profile Analysis) é um software de processamento de informações, oriundo de uma dissertação de mestrado, que trata informação de ex-alunos e usa como base informações do LinkedIn, recuperara cerca de 1.500 registros. Seu método de coleta consiste em uma busca pelo nome e faculdade usando a API *search* do LinkedIn, que não existe mais.

HiWe [12] (Hidden Web exposer) é um trabalho sobre crawlers que indexa páginas e depois os links. Etapas como *URL selection, page retrieval e page processing* são executadas primeiramente e depois os links são extraídos como etapa incremental para se analisar os formulários. HiWe então analisa o Modelo

[6] http://www.google.com

[7] http://www.archive.org

de Objeto de Documento (DOM) da página, especialmente as tags do tipo *label* para inferir um significado ao documento.

XU [21] pesquisou o problema de buscar um pareamento de perfis baseado em *keyword matching* e para coleta usa o InMail[8] do LinkedIn, e assim procura resgatar aqueles perfis similares que aparecem correlacionados. O InMail é um produto pago fornecido pelo LinkedIn que permite que os recrutadores atinjam os membros fora de sua rede existente. Empiricamente falando, se comporta como um recrutador de talentos.

Alguns autores [14, 15, 19, 29-30] usam o recurso de API cedidos inicialmente pela fonte provedora de dados para alimentar suas pesquisas, observamos por exemplo em CETINTAS et al. [19] onde foram recuperados 2.200 perfis. Os autores propõem um trabalho que procura achar perfis parecidos baseados em três categorias: perfil, grafo e *social features* do LinkedIn. Usou um serviço de recuperação de dados da Web terceirizado[9] que já não se encontra mais ativo.

GONÇALVES [29] faz uso da API Google Custom Search Engine[10] (CSE) e assim recupera 357 alumni. CSE (Figura 3) é também detalhado por ALLAUDIN & AZAM [30] onde explica-se suas limitações referentes à API do Google para coleta de dados.

8 https://business.linkedin.com/marketing-solutions/sponsored-inmail

9 https://www.crowdflower.com

10 http://developers.google.com/custom-search

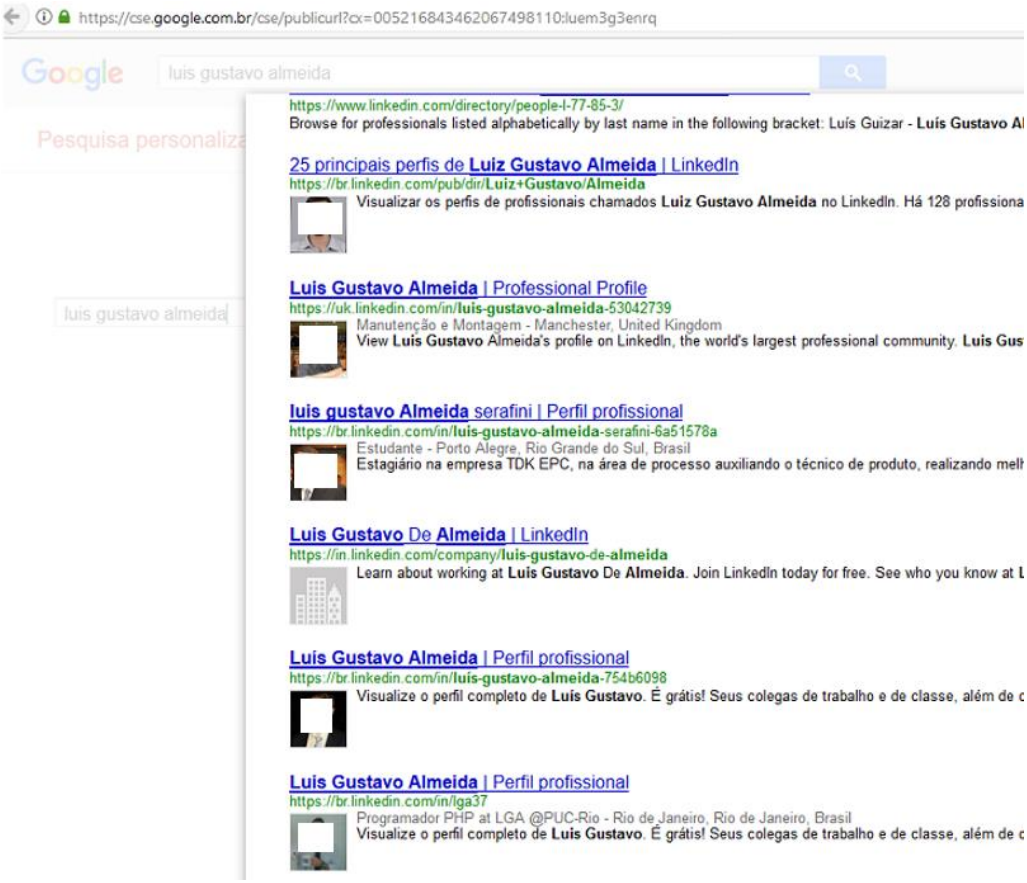


Figura 3: Coleta de dados usando a API CSE do Google.

Mercator [47] é um crawler que varre a Web indexando documentos HTML através de requisições HTTP. É escalável e usa Random Walks[11] para coleta de dados. Sua concepção foi feita em Java[12].

O aplicativo Web skrapp.io[13] oferece a possibilidade de se coletar informações em qualquer contexto do site LinkedIn. Entretanto, a captura de dados é feita manualmente, através da navegação Web. Outros serviços pagos e disponíveis na Internet também oferecem a possibilidade de se fazer coleta de dados, mas devido ao aspecto comercial estão fora do escopo do trabalho.

Pro5 [82] é um programa escrito em PHP[14] para efetuar crawling de sites, fortemente baseado no framework Yii[15]. A entrada de dados referente aos sites

[11] http://en.wikipedia.org/wiki/Random_walk

[12] http://www.java.com

[13] http://www.skrapp.io

[14] http://www.php.net

[15] http://www.yiiframework.com

sementes é informada pelo usuário manualmente. Concebido para basicamente analisar preços de e-commerce, foi um dos únicos crawlers escrito em PHP.

SPHINX [46] (Site-oriented Processors for HTML INformation eXtraction), permite que as regras de rastreamento específicas do site sejam encapsuladas e reutilizadas em analisadores de conteúdo, conhecidos como classificadores. Ele varre a Web segundo uma estrutura de grafos, escrito em C e Java.

Aardvork [44] é um crawler que dentre outras coisas inova ao realizar as análises em sites coletados com um filtro de *mime-types* e DNS feito em Java.

Vizster [43] é um sistema de visualização para usuários finais que aproveita o potencial de exploração de redes sociais; não chega a coletar dados de maneira maciça. Efetua busca, análise visual, identificação e visualização de estruturas comunitárias. Trata-se de uma ferramenta feita em Java que conecta diferentes redes sociais a partir de um nó central.

A divisão no aspecto client/server de um crawler é abordada no trabalho de ALVAREZ et al. [13], onde os autores sugerem inclusive uma classificação para os rastreadores baseados no mecanismo de processamento de dados, que pode ser tanto do lado do *client* (ou navegador) "Crawl do 'lado do cliente'" devido ao fato de muitos Websites usarem linguagens de script e mecanismos de manutenção de sessão para autenticação. "A maioria dos rastreadores convencionais não conseguem lidar com esse tipo de páginas", ponderam. A outra categoria, devido ao fato de muitos sites oferecerem formulários de consulta para acessar o conteúdo de um banco de dados subjacente, diz respeito ao "Crawl do 'lado do servidor'" e assim concluem que os rastreadores convencionais não podem acessar essas páginas porque não sabem como executar consultas nesses formulários (*Deep Web*).

2.3
Scraping

CHANG [25] discorre sobre EI onde a ideia central de seu trabalho é basicamente associar a string HTML com *tokens* em numeração binária e depois gerar uma árvore que assim pode ser recuperada e conferir significância em algum contexto.

NoDoSE [50] (Northwestern Document Structure Extractor) é uma ferramenta interativa para determinar semi-automaticamente a estrutura de documentos e, em seguida, extrair seus dados. Usando uma Graphical User Interface (GUI), o usuário hierarquicamente decompõe o arquivo manualmente, descrevendo suas regiões interessantes e depois descrevendo sua semântica. Uma vez que o formato de um documento foi determinado, seus dados podem ser extraídos. É um extrator que de maneira semiautomática determina a estrutura de documentos e extrai conteúdo, escrito em Java.

Os autores de ANDES [18] (A Nifty Data Extraction System) descrevem uma metodologia baseada em XML para a extração de dados da Web que se estende além da simples raspagem de tela (*scraping*).

Encontramos alguns trabalhos sobre a rede social colaborativa [53,54] Digg[16] para filtragem social. Os autores rasparam o site Digg com a ajuda de wrappers da Web, criados com ferramentas fornecidas pela Fetch Technologies[17].

Vide [23] (Vision-based data extractor) é um extrator que consiste em quatro etapas: construção de árvore de bloco visual, extração de registro de dados, extração de itens e geração de wrapper visual, realizadas com base em recursos visuais, como por exemplo tamanhos de fontes e cores de links e textos.

A respeito de abordagens manuais para construção de wrappers para identificar e extrair campos de dados desejados, algumas das ferramentas mais conhecidas que adotam abordagens manuais são Minerva [107], TSIMMIS [107] e WebOQL [106]. Obviamente, eles têm baixa eficiência e não são escaláveis.

Omini [24,27] é um *wrapper end-to-end* que percorre a árvore DOM de um documento, através de um algoritmo com objetos separadores baseado em tags vizinhas e tags repetidas, combinando uma heurística onde enumera objetos candidatos para a construção do modelo. Foi testado basicamente em um determinado domínio, no caso e-commerce, para EI. Omini efetua o *parsing* das árvores de tags relacionados a um documento HTML da Web.

Algumas abordagens automáticas representativas são Omini, RoadRunner e Iepad. Algumas dessas abordagens executam apenas a extração de registros de

[16] http://www.digg.com

[17] http://www.fetch.com

dados, mas não a extração de itens de dados, como Omini. Já RoadRunner, Iepad e Omini não geram wrappers. Todos eles dependem principalmente da análise do código-fonte das páginas da Web.

Iepad (25) (Information extraction based on patter discovery) inclui árvores, alinhamento de strings e algoritmos de matching para realizar RI. Usa como associação as tags referenciadas com *tokens* binários. Usa padrões repetitivos de tags HTML para identificar e extrair dados.

Em RoadRunner [51] o artigo investiga técnicas de extração de dados de sites HTML através do uso de *wrappers* gerados automaticamente. O artigo desenvolve uma nova técnica para comparar páginas HTML e gerar um *wrapper* baseado em suas semelhanças e diferenças. Através do acme – *align collapse under mismatch and extract* procura validar o conteúdo semântico de uma página HTML e assim inferir páginas semelhantes. São abordadas implementações sobre o uso de expressões regulares sobre páginas HTML para se extrair informações de documentos da Web.

Tanto RoadRunner como o Iepad fazem a suposição simplificadora de que uma tag HTML é sempre parte do modelo da página. Embora, estatisticamente, as tags HTML tenda a ocorrer no modelo, há um número significativo de casos em que ocorrem dentro dos dados.

Em Effective Web Extraction Data [11] usa-se processadores XSLT sobre HTML para abordar questões como *data mapping* e *data integration*. Nesta pesquisa foi adotado o conceito de expressões regulares sobre tags HTML.

2.4
Redes Sociais

Pesquisas sobre redes sociais são realizadas desde meados de 1960. ERDŐS & RÉNYI [111] propuseram uma teoria sobre redes aleatórias,demonstraram que bastava uma conexão entre cada um dos convidados de uma festa para que todos estivessem conectados ao final dela. Concluíram que todos os nós em uma determinada rede teriam aproximadamente a mesma quantidade de conexões, constituindo-se em redes igualitárias. MILGRAM [102] enviou aleatoriamente uma quantidade de cartas a vários indivíduos, solicitando que tentassem redirecioná-las a um alvo específico. Se não conhecessem o alvo, as pessoas

eram solicitadas a enviar as cartas para alguém que acreditassem estar mais próximo a ele. Milgram descobriu que todas as pessoas estavam a poucos graus de separação umas das outras, ou seja, em um ‘mundo pequeno’, também conhecido como ‘princípio dos seis graus de separação’.

Apesar de estabelecer certos padrões, Milgram e posteriormente WATTS [92] tratavam as redes sociais como redes aleatórias, ou seja, os nós eram estabelecidos de modo randômico, exatamente como ERDŐS & RÉNYI [111]. Por sua vez, BARABÁSI [103] demonstrou que as redes não são formadas de modo aleatório, mas que existe uma ordem na dinâmica de estruturação das redes. Este padrão aponta para o fato de que ‘ricos ficam mais ricos’, ou seja, quanto mais conexões um nó possui, maiores as chances de ele ter mais novas conexões. Assim tais redes possuiriam poucos nós altamente conectados (*hubs*). Segundo WATTENBERG [90], a análise de redes sociais envolve três tarefas fundamentais: (1) identificar comunidades: os atores devem ser agrupados em comunidades, de acordo com seus atributos, (2) identificar atores centrais, (3) analisar papel e posições de conexões e atores.

2.5
Conclusões do Capítulo

A grande maioria dos crawlers fora escrito em Java [43, 46, 47, 50, 52, 84].Não foi encontrado nenhum método automatizado de extração de informações de usuários de redes sociais, diretamente da Web. Especialmente redes sociais que exigem autenticação. Também não fora encontrado nenhuma referência a se usar um programa voltado para teste funcional como o Selenium como entrada de dados em algum tipo de aplicação de informática.

3.
Conceitos Básicos

É altamente desencorajante fazer uma coleta de dados na Web quando nos deparamos com um sistema em que é permitido acesso à informação somente a usuario autenticado. Este trabalho visa romper com esta barreira e propiciar ao meio acadêmico uma solução eficaz para o problema.

Neste capítulo vamos descrever as Redes Sociais de uma maneira mais genérica e abordar o LinkedIn e o Lattes, finalizamos o capítulo abordando brevemente extração e interpretação de informação em documentos e arquivos.

3.1
Redes Sociais Profissionais

As tecnologias de informação e comunicação (TICs) estruturaram configurações robustas e acessíveis nas redes computacionais, permitindo um fluxo informacional jamais imaginado [2].

As novas TICs fizeram surgir uma gama de novas possibilidades para a análise de redes sociais e consequentemente de redes de colaboração em Ciência, Tecnologia e Inovação (CT&I) [3]. O advento da Internet é sem dúvida o fato mais significativo, valendo-se da criação de padrões, principalmente baseados em XML, e da disponibilidade de serviços usando a tecnologia de Web Services.

O termo rede social online é geralmente utilizado para descrever um grupo de pessoas que interagem primariamente através de qualquer mídia de comunicação. Consequentemente, redes sociais online existem desde a criação da Internet. WALTER [2] define uma rede social online como um serviço Web que permite a um indivíduo: (1) construir perfis públicos ou semipúblicos dentro de um sistema, (2) articular uma lista de outros usuários com os quais ele compartilha conexões e (3) visualizar e percorrer suas listas de conexões assim como outras listas criadas por outros usuários do sistema.

A variedade, a facilidade de socializar informações e a possibilidade de identificar indivíduos por meio da utilização das TICs faz com que tanto o

comportamento do indivíduo seja diferente, quanto os processos de uma organização no tocante a seleção de pessoas para contratação [7].

Segundo CASTELLS [101], na atual sociedade, as funções e os processos dominantes estariam organizados em redes. “Redes constituem a nova morfologia social de nossas sociedades e a difusão da lógica de redes modifica de forma substancial a operação e os resultados dos processos produtivos e de experiência, de poder e de cultura”.

“Com o desenvolvimento das ferramentas tecnológicas emergem novas formas de comunicação dentre elas as redes sociais” [99]. Percebe-se que as redes sociais passaram a atender as necessidades que os indivíduos têm de se comunicar e de se relacionar. As redes sociais podem ser usadas para diferentes funções entre os indivíduos como, por exemplo, a procura de oportunidades profissionais, contatos, entre outras funções [104].

Segundo CAMARGO [8] as redes sociais são as fontes de contratação que mais crescem no Brasil com procura de 44% em 2013 comparado com 2011 que foi de 16%, além disso dados mais recentes [9,10] informam que temos no Brasil cerca de 25 milhões de usuários em 2016, tendo um crescimento de 25% no último ano, mesmo contrastando com o cenário de crise econômica.

LinkedIn, comparada com as demais redes sociais e sites de emprego, é indicada como uma das preferidas por gestores de recrutamento.

“A própria natureza humana nos liga a outras pessoas e estrutura a sociedade em rede” [105]. Para BASTOS & SANTOS [100] a rede é representada pelas interações que visam a comunicação, a troca e o apoio mútuo, e que aparecem nos compartilhamentos e em momentos vivenciados. As redes são constituídas por uma nova estrutura social de nossa sociedade e a disseminação do seu significado altera os processos produtivos, a cultura, o poder e a experiência [101]. A estrutura em rede facilita o compartilhamento de informações em razão de a rede ser um espaço apreciado para a concepção do conhecimento [105].

AFONSO [99] aponta que algumas redes sociais foram criadas com o foco profissional, sendo conhecidas como redes sociais profissionais, tais como o LinkedIn, no qual “os membros cadastrados interligam-se pelas comunidades de acordo com a natureza do seu trabalho, sua formação, seu conhecimento e até mesmo seus contatos profissionais”.

CETINTAS et al. [19] definem as PSNs como redes sociais orientadas para negócios. Devido à variedade de atividades disponíveis em PSNs, informações sobre seus usuários podem ser obtidas de muitas fontes heterogêneas, como conteúdo de perfil por exemplo.

Descrição e predição podem ser utilizadas para descoberta de conhecimento em redes sociais [78]. Os padrões descritivos são classificados em agrupamento, regras de associação e padrões sequenciais, sendo utilizados, por exemplo, para encontrar padrões que sejam interpretáveis pelo homem e que descrevam os dados [113]. Os padrões preditivos são definidos por regressão e classificação [112] e são utilizados para predizer o valor desconhecido ou futuro de um ou mais atributos com base no valor conhecido dos demais atributos. Métodos de descrição para redes sociais utilizam as técnicas convencionais de mineração de dados em bancos de dados.

Os serviços de redes sociais profissionais estão cada vez mais integrados ao cotidiano de profissionais atuantes no mercado, e seu crescimento tem sido estimulado pelo aumento no número de integrantes. A principal característica destas redes profissionais é a exposição do perfil do profissional, onde nele é possível registrar todo histórico profissional e educacional, incluindo recomendações e servindo assim de vitrine no mundo corporativo.

3.2 LinkedIn

O LinkedIn (Figura 4) é uma rede social profissional, fundada em 2003 por Reid Hoffman, com sede em Mountain View, Califórnia. Tem como missão conectar profissionais do mundo todo, tornando-os mais produtivos e bem-sucedidos [7]. De acordo com GREGO [14], o LinkedIn possui 300 milhões de usuários em mais de 200 países e territórios em todo o mundo. No Brasil, há 25 milhões de usuários cadastrados [9,10]. A rede social profissional LinkedIn pode ser definida como uma empresa de capital aberto com um modelo de negócios variado em que toda receita vem por meio das assinaturas dos usuários, vendas de publicidade e soluções de recrutamento [7].

GREGO [14] aponta que o Brasil ocupa a terceira posição, sobretudo após o lançamento da ferramenta em português e a disponibilidade do aplicativo em

todos os sistemas operacionais móveis. Além disso, o LinkedIn pode favorecer seus usuários em suas carreiras por meio da variedade de funcionalidades disponibilizadas e, deste modo, tem uma base de usuários significativa [111]. Ressalta-se também que a adoção e o uso de tecnologia têm sido motivo de atenção por parte de pesquisadores e profissionais de diferentes áreas afins [6].

A rede social profissional LinkedIn disponibiliza acesso aos usuários independentemente da situação profissional, ou seja, aos profissionais que estão desempregados ou não [8]. Para TELLES [93] "O LinkedIn é uma espécie de currículo profissional, em que consta sua posição atual, os cargos que exerceu, sua escolaridade, seus sites, particularmente sites de empresas, e seu blog."

Em meados de dezembro de 2016 existiam cerca de 66.000 ex-alunos da PUC-Rio registrados no LinkedIn. Estes ex-alunos encontram-se na página oficial da PUC-Rio[18] no LinkedIn. Nesta página estão todos aqueles que tenham, teoricamente em ao menos em um momento de suas vidas, realizado qualquer curso na PUC-Rio e assim registrado tal fato no formulário de usuario do LinkedIn. Ressalta-se que não há nenhum mecanismo de auditoria ou validação desta informação no LinkedIn e este fato abre um precedente para um fenômeno conhecido como 'lavagem de diploma', que é mencionado no capítulo 6.

Em cada perfil de determinado ex-aluno consta, dentre outras informações, seu histórico profissional e seu histórico acadêmico. A presente pesquisa terá como foco inicialmente apenas esses dois campos. No decorrer deste trabalho, aproveitando a nomenclatura do próprio LinkedIn, chamaremos o histórico profissional de **jobs**, e o histórico acadêmico chamaremos de **edus**.

[18] https://www.linkedin.com/school/165595/alumni/

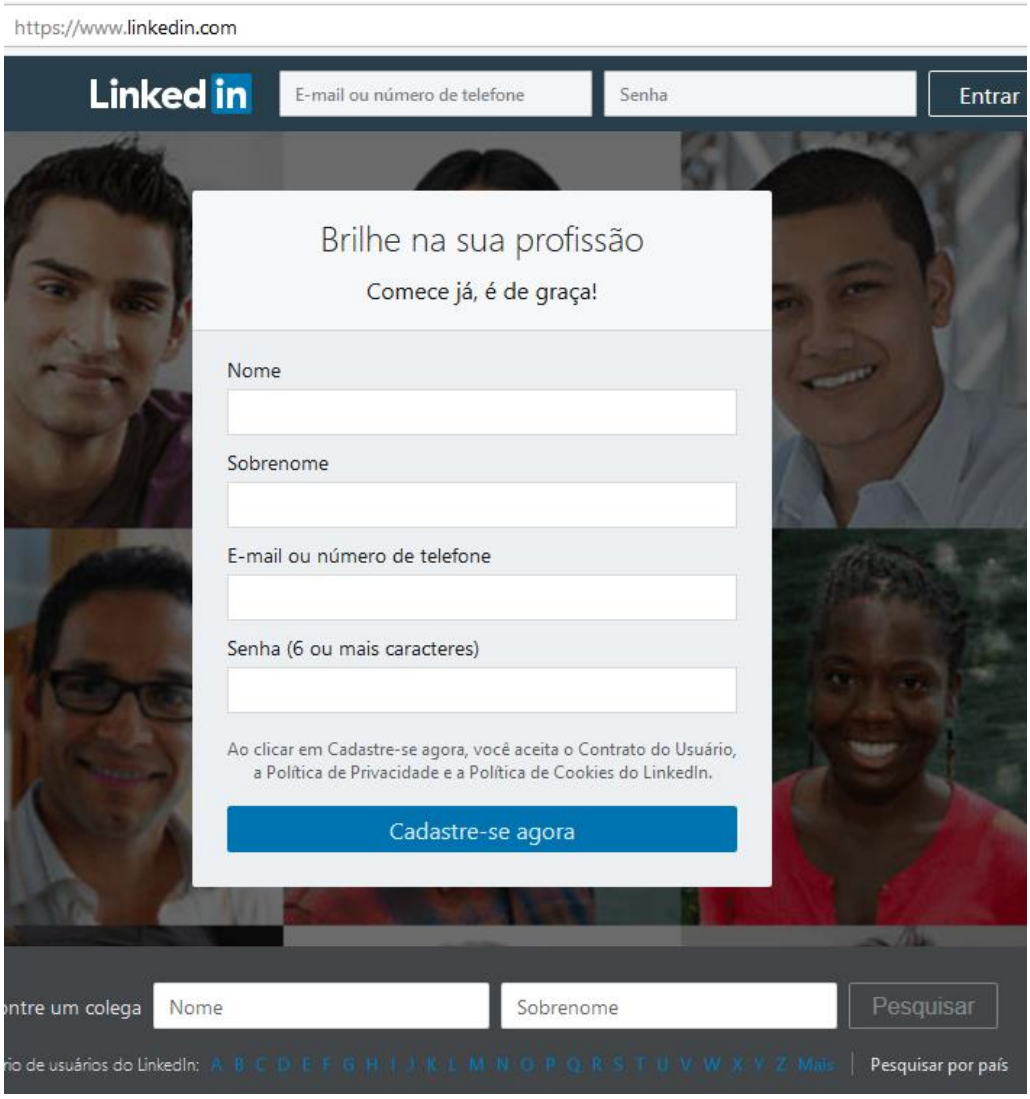


Figura 4: Página de entrada do LinkedIn.

3.3
A Plataforma Lattes

A Plataforma Lattes do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) é uma base de dados que contém, dentre outras informações, os currículos da maior parte dos pesquisadores que atuam no Brasil (Figura 5). Grande parte dos editais de financiamento de projetos feitos por instituições de amparo à pesquisa, como o próprio CNPq, utilizam os currículos Lattes dos pesquisadores como uma das formas de avaliação das propostas. Segundo FARIAS et al. [1] este fato motiva os pesquisadores a manter em seus currículos com informações corretas e atualizadas, tornando a plataforma Lattes uma fonte adequada para análise da produção científica brasileira. Portanto "é

uma base de dados que contém, entre outras informações, os currículos da maior parte dos pesquisadores que atuam no Brasil" [1].

O sistema Lattes Egressos, segundo BOVO [5], foi concebido para ajudar os gestores dos cursos a obter informações sobre os egressos de seu curso que continuam ligados ao sistema nacional de CT&I. O sistema permite a apresentação de análises sobre o perfil de egressos de cursos de uma instituição [3].

O Lattes tornou-se um padrão nacional no registro da vida dos estudantes e pesquisadores do país, e é hoje adotado pela maioria das instituições de fomento, universidades e institutos de pesquisa do Brasil. Por se tratar de uma instituição voltada para fins acadêmicos, sua base é constituída em sua maioria por pesquisadores e professores, consequentemente sua inclusão em qualquer sistema de BI sempre confere uma inclinação ao aspecto mais acadêmico. O acesso a esta informação pode ser feito através de arquivos XML.

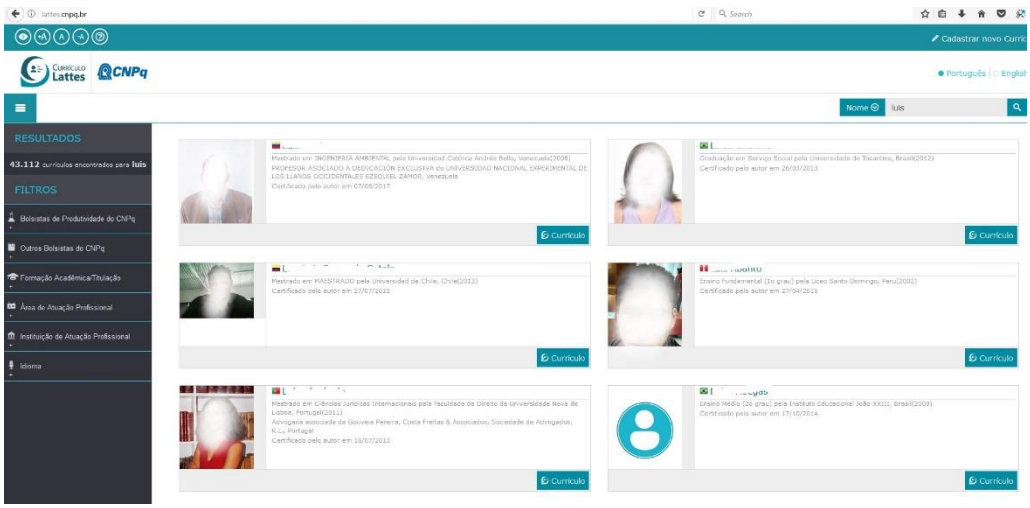


Figura 5: Página oficial do Curriculum Lattes.

3.4 Scraping e Extração de Informação

Extrair dados estruturados de páginas da *Deep Web* é um problema desafiador devido às estruturas intrínsecas subjacentes dessas páginas. Até agora, um grande número de técnicas foi proposto para resolver este problema, mas todas elas têm limitações inerentes [23].

Conforme define MYLLYMAKI [18], um processo de extração de dados ideal é capaz de digerir bancos de dados da Web de destino visíveis apenas como

páginas HTML e criar uma réplica local e idêntica desses bancos de dados como resultado. Um processo abrangente de extração de dados precisa lidar com esses obstáculos, como identificadores de sessão, formulários HTML, JavaScript, problemas de integração de dados e dados faltantes ou conflitantes. A extração adequada de dados também requer um serviço sólido de validação de dados e recuperação de erros para lidar com falhas de extração de dados, que são inevitáveis.

As técnicas de extração de dados da Web emergem como uma ferramenta chave para a análise de dados nos sistemas de negócios e de inteligência competitiva. Permitem reunir uma grande quantidade de dados estruturados gerados e disseminados continuamente pelos usuários das redes sociais e oferece oportunidades sem precedentes para analisar o comportamento humano em grande escala.

Dados extraídos da Web podem servir de trampolim para uma variedade de tarefas, incluindo monitoramento de eventos (notícias e mercado de ações) e comércio eletrônico (comparação de preços). Extrair dados estruturados de sites da Web não é uma tarefa trivial [18]. Dado que o formato dos documentos HTML foi projetado para apresentação, não extração automatizada, e o fato de que a maioria do conteúdo HTML na Web está mal formado (quebrado), extrair dados desses documentos pode ser comparado com a tarefa de extrair estrutura de documentos não estruturados. A agregação de dados de vários sites exige que os dados sejam homogeneizados.

Eventualmente, os dados extraídos podem ser pós-processados, convertidos no formato estruturado mais conveniente e armazenados para uso posterior. A disponibilidade e análise dos dados coletados é um requisito incontestável para entender fenômenos sociais, científicos e econômicos complexos que geram os dados.

Para CHANG et al. [25], a Web é complementar e a pesquisa de extração de informações pode ter impacto crítico em ambos os esforços. Depois de tudo, a marcação XML fornece apenas uma das muitas interpretações semânticas possíveis de um documento.

A explosão no número de documentos aumentou significativamente a dificuldade de recuperar informação em sistemas manuais [55]. O processo de

recuperação de informação compreende basicamente três etapas: indexar, armazenar e recuperar.

Para FREITAS [57], a construção de extratores de informação da Web oferece vantagens: o usuario é mais bem atendido, livrando-se de processar manualmente as páginas atrás de dados, e, por isso, a rede fica com menor tráfego, já que muitos ponteiros inúteis não serão listados nem carregados. Bancos de dados, diferentemente da Web, podem ser facilmente consultados, provendo ao usuario consultas semanticamente claras e precisas sobre entidades e relacionamentos entre elas, inclusive combinando e totalizando dados, tarefas que os mecanismos de busca, mesmo os especializados, não conseguem realizar.

3.5
Análise de Dados

A mineração de dados é um conjunto de técnicas para descobrir conhecimentos em uma grande quantidade de dados, permitindo a extração de padrões relevantes, que não podem ser facilmente detectados apenas pelo navegador e pesquisa. Normalmente, os algoritmos de mineração de dados reconhecem padrões relevantes em conjuntos de dados organizados por modelos de dados adequados para processamento e recuperação eficazes de dados [15].

O armazém de dados (DW) abrange arquiteturas, algoritmos e ferramentas para trazer dados selecionados de vários bancos de dados ou outras fontes de informação em um único repositório, adequado para consultas ou análises diretas. Nos últimos anos, o armazenamento de dados tornou-se uma palavra-chave proeminente no setor de banco de dados [67]. Segundo LOPES [58] a evolução histórica dos sistemas de RI apresenta duas linhas principais de desenvolvimento. A primeira tem suas origens nos grandes sistemas de bases de dados desenvolvidas pelas instituições americanas: National Library of Medicine (NLM), Department of Defense (DOD) e pela National Aeronautics and Space Administration (NASA), que indexavam suas bases de dados referenciais utilizando os dicionários específicos de suas áreas temáticas. A segunda linha teve o seu desenvolvimento no campo de direito e envolvia a geração da base de dados com o texto completo das leis. BI emergiu como uma importante área de estudo para ambos os praticantes e pesquisadores, refletindo

a magnitude e impacto de problemas relacionados a dados a serem resolvidos em organizações empresariais [65].

Em sistemas gerenciadores de bancos de dados (SGBD), os símbolos são armazenados em uma estrutura matricial em campos determinados, com metadados que lhes conferem certo sentido ontológico. Para recuperar dados específicos, basta especificar as restrições necessárias aos campos de pesquisa e codificá-las numa query (argumento de entrada no sistema) para que se tenha a resposta exata, fruto de busca completa e exaustiva [56].

3.6
Conclusões do Capítulo

Analisar perfis profissionais para rastreamento e mineração do LinkedIn é um problema desafiador [15]. Particularmente para o rastreamento, existem dados duplicados, SPAM, restrições de acesso e problemas de ambiguidade que devem ser superados. Além disso, a escalada maciça do LinkedIn impõe limitações para extração, transformação e armazenamento de dados.

4.
A Ferramenta ALUMNI

Este capítulo apresenta a ferramenta ALUMNI, que possui um aspecto inédito ao combinar um programa voltado para teste funcional com um robô de busca escrito em PHP para assim alimentar de maneira programática um banco de dados com informações de perfis de usuario recuperados do LinkedIn.

4.1
Arquitetura

Para realizar toda a coleta de dados com a subsequente extração de dados de páginas Web, os componentes da ferramenta foram divididos em dois módulos:

robô: Trata-se de um conjunto de programas em PHP que, através de trabalho em conjunto com Selenium (Figura 10), acessa a URL do LinkedIn tanto para recuperar **hashes** (identificadores únicos de alumni ou ID) quanto para recuperar perfis. Estes programas devem ser usados na linha de comando do sistema operacional.

web: Aplicação final para análise dos dados, onde são disponibilizados os gráficos e análises que respondem os objetivos da pesquisa através de um Website. Todos os arquivos são scripts PHP responsáveis por processar os dados e exibir as informações de maneira visual, no formato de páginas HTML, gráficos e tabelas.

A manutenção da sessão ativa após realizado o acesso autorizado e o esquema de busca e pesquisa em URLs e formulários são os principais problemas enfrentados pela ferramenta.

Foi adotada a programação PHP estruturada e o padrão de projeto MVC[19] sobre uma estrutura LAMP[20], que também é compatível com WAMP[21], onde cada uma das camadas apresenta a seguinte abordagem:

Model: Corresponde basicamente ao banco de dados e às consultas, que serão realizadas com PDO[22] usando para tal a linguagem T-SQL correspondente ao banco de dados MySQL[23].

Controller: Utiliza parâmetros fornecidos na URL ($_GET[24]) e formulários Web ($_POST[25]) para consultar as informações do banco de dados e passar para a View.

View: Através do componente TWIG[26] que é voltado para confeccionar sistema de templates, onde ele recebe as informações geralmente no formato JSON do controller e as renderiza no *browser*.

Será levada em consideração também a criação de usuários para acessar a aplicação com login e senha, caso decida-se, em trabalhos futuros, hospedar as páginas em provedor externo.

O método baseado em requisição HTTP consiste em iniciar o servidor Selenium no Windows[27] ou no Linux[28], feito isso o servidor Selenium estará disponível à aplicação na porta 4444 (Figura 6) atendendo prontamente a qualquer requisição por parte do programa PHP. Referente ao script PHP, o objeto *$webDriver* (instância do Selenium WebDriver) é que se encarregará de comandar o controle da navegação autônoma das páginas (quadro 1) e de quais URLs deverão ser acessadas e em qual ordem.

---

19 http://pt.wikipedia.org/wiki/MVC

20 LAMP é uma combinação de softwares livres como Linux, Apache, MySQL e PHP que viabilizam o desenvolvimento de aplicações Web.

21 WAMP tal qual LAMP é a combinação de Windows, Apache, MySQL e PHP.

22 http://php.net/manual/book.pdo.php

23 http://www.mysql.com

24 https://secure.php.net/manual/pt_BR/reserved.variables.get.php

25 https://secure.php.net/manual/pt_BR/reserved.variables.post.php

26 http://twig.symphony.com

27 C:\java selenium-server-[versao].jar

28 #java selenium-server-[versao].jar

```
$capabilities = DesiredCapabilities::firefox();
$webDriver = RemoteWebDriver::create('http://localhost:4444/wd/hub', $capabilities);
$webDriver->get('https://www.linkedin.com/uas/login');
```

Quadro 1: Inicialização do objeto $webDriver, responsável por navegar na Web.

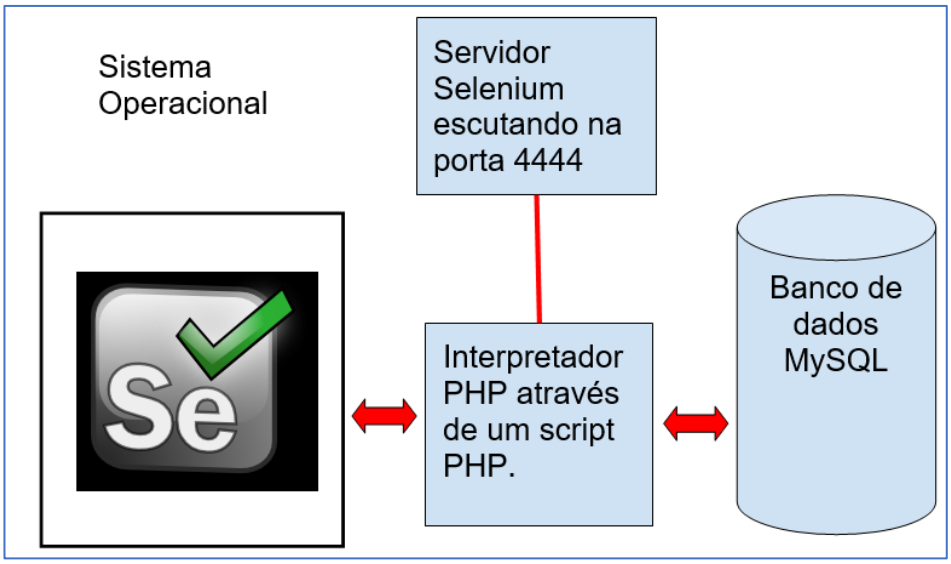


Figura 6: Esquema da arquitetura básica da solução.

4.2
Coleta de Dados

Para vencer a autenticação do LinkedIn, é necessário programar o robô para acessar a página de login do LinkedIn e, com um login e uma senha válidos de qualquer usuario cadastrado, obter acesso autorizado ao Website e navegar de maneira autônoma para as mais diversas páginas, mantendo sua sessão ativa. Para isso será usado o login e senha do autor[29]. A inovação deste trabalho é suprimir os pontos falhos do rastreamento de URLs como, por exemplo:

- Manutenção da sessão ativa no navegador.
- Conteúdo HTML que pode ser gerado por AJAX.
- Sistema de busca do LinkedIn que é voltado para um usuário convencional (ser humano).

[29] https://www.linkedin.com/in/lga37

- Eventuais barreiras que o LinkedIn pode adotar contra robôs de busca.

4.2.1
Coleta Manual de Dados

A coleta manual de dados de um Website como o LinkedIn somente seria viável se dispuséssemos de uma grande equipe e, ainda assim, se a base de dados em questão fosse pequena. Afinal todo este processo estaria vinculado a um ser humano, que operando algum programa específico, ou até mesmo simples planilhas de dados, iria se incumbir de processar os históricos acadêmicos e profissionais.

Na literatura podemos observar o trabalho de PENA [31] onde coletou-se manualmente dados relacionados a 357 alumni. Já IMBRIZI & FILFO [37] coletaram 94 registros, e no trabalho de CAMARGO [8] foram usado inclusive questionários como entrada de dados. LOUSADA & MARTINS [74] realizaram entrevistas para coleta de dados a fim de estudar o comportamento de egressos.

Acessar cerca de 66.000 perfis sem auxílio de uma ferramenta de automatização e recuperar integralmente os dados de maneira organizada para o BD, é uma tarefa que para uma única pessoa seria completamente inviável na prática.

4.2.2
Coleta de Dados usando uma API

Uma API[30] refere-se a um conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por aplicativos que não pretendem envolver-se em detalhes da implementação do software, mas apenas usar seus dados e serviços.

Na Web de hoje, os dados geralmente estão disponíveis via APIs, a área de Web Service Discovery (WSD) tem uma importância fundamental para a recuperação de informação na Web [30]. No entanto, a maior quantidade de dados está disponível principalmente em formatos semi-estruturados, como HTML. Para recuperar dados da Web é essencial fornecer meios para

[30] http://pt.wikipedia.org/wiki/api

transformar aplicativos e sites da Web automaticamente em *WebServices*, permitindo um acesso estruturado e unificado a fontes heterogêneas.

A API do LinkedIn deve ser autenticada com um perfil de usuário registrado na rede; depois de autenticar, ela só permite ao usuário visitar e recuperar automaticamente dados de outros usuários que estão separados de tal usuário por poucos graus de separação [29]. Além disso, outra limitação da API é o número de atributos disponíveis para consulta: a rede social é mais cuidadosa ao expor seus dados através da API que permitem a extração; portanto, o LinkedIn fornece mais atributos nas páginas públicas da rede disponíveis na Web, embora estes atributos não incluam **jobs** e **edus**.

Apesar da dificuldade inerente ao rastreamento de dados públicos, o rastreamento de dados das redes sociais é ainda mais difícil, pois os servidores de redes sociais geralmente não estão disponíveis gratuitamente para o rastreamento [13]. A rede social do LinkedIn referente a oportunidades de trabalho e negócios apresenta sua própria *API People Search* para pesquisar pessoas, porém este recurso foi descontinuado em 2015 [15].

A API do LinkedIn usa a arquitetura do tipo REST[31] baseada em endpoints. Uma das principais vantagens desta arquitetura é a organização do fluxo de informação cujas informações podem ser traduzidas em códigos de resposta HTTP (*status codes*[32]). Em uma API REST, da mesma maneira que acontece com os verbos HTTP, os códigos de resposta formam um padrão facilmente reconhecido por quem for consumir o WebService. Os códigos de sucesso têm o padrão 20x, os de redirecionamento 30x, os de erro de *client* 40x e os de erro de servidor 50x.

Poderíamos usar a API do LinkedIn, mas este recurso não é totalmente aberto e disponível. Na verdade, a API disponível apenas oferece informações pouco relevantes, como o nome do usuário por exemplo. Sendo ‘informação’ o *core business* do LinkedIn, é provável e assim supomos que a empresa queira obter lucros com este domínio de informações, ou simplesmente, vender em última análise tais informações. De qualquer forma, ao se tentar requisitar os

[31] REpresentational State Transfer é um estilo arquitetural que consiste em princípios e regras que permitem a criação de um projeto com interfaces bem definidas.

[32] https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.HTML

recursos da API um código de retorno HTTP 403 (proibido) é exibido (Figura 7). Para o usuário acessar dados através da API, ele deve entrar em contato com o LinkedIn e obter uma autorização específica de acesso aos parâmetros de pesquisa, o que envolve uma relação comercial com o LinkedIn, o que está além do escopo deste trabalho.

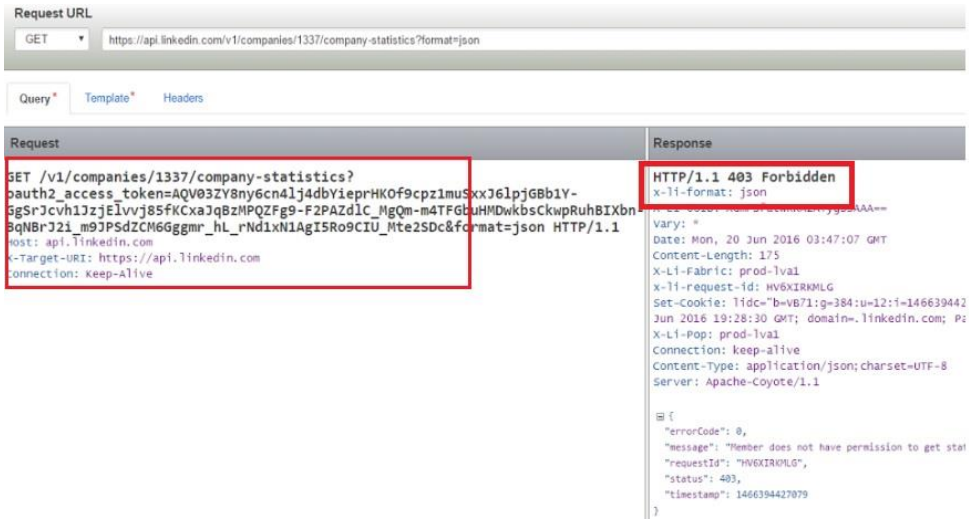


Figura 7: API do LinkedIn bloqueada para acesso público.

4.2.3
Coleta de Dados usando o Protocolo OAuth2

A especificação OAuth2[33] define um protocolo de delegação (Figura 8) que é útil para transmitir decisões de autorização em uma rede de aplicativos e API habilitados para Web. OAuth2 é usado em uma grande variedade de aplicativos, incluindo o fornecimento de mecanismos para autenticação de usuários.

Se existem recursos protegidos na API, ou seja, que devem ter o acesso controlado e só podem ser acessados por um usuário específico, então é preciso considerar como levar em conta essas diversas aplicações cliente, porque, afinal, o usuário não acessará o recurso diretamente, mas irá, na verdade, delegar essa tarefa a uma aplicação cliente.

[33] http://oauth2.net/2

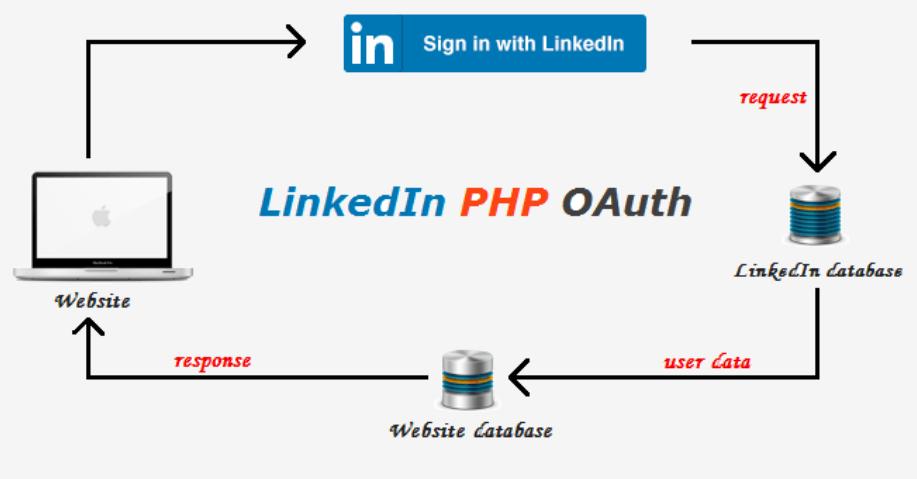


Figura 8: Arquitetura do protocolo de delegação OAuth2.

O OAuth2 estabelece que, quando uma aplicação cliente precisa acessar um recurso protegido no servidor, ela deve obter um token de acesso. Esse token de acesso contém as informações que caracterizam o acesso e isso possibilita recuperação de informação.

Em termos práticos para nosso trabalho, a autenticação OAuth2 resume-se a coletar dados do usuário, mas somente se o usuário efetuar login em algum aplicativo usando por exemplo o recurso **Login com LinkedIn**. Ao se logar desta maneira, o aplicativo, através das trocas de informações baseadas em tokens, tem acesso às informações do usuário, inclusive aos seus históricos profissionais (Figura 9). A inviabilidade deste método é que teríamos que convencer todos os ex-alunos a acessarem alguma aplicação com qualquer propósito, o que envolveria entrar em contato via InMail por exemplo e solicitar acesso, porém isso poderia ser caracterizado como SPAM.

No trabalho de YANG et al. [64], os autores discutem o potencial deste protocolo de delegação para obter acesso a fontes de dados, sobretudo redes sociais. Especula-se o fato de existirem um bilhão de dispositivos móveis, e a hipótese de se usar o OAuth2 para recuperar informações, já que o protocolo também pode ser usado em dispositivos móveis.

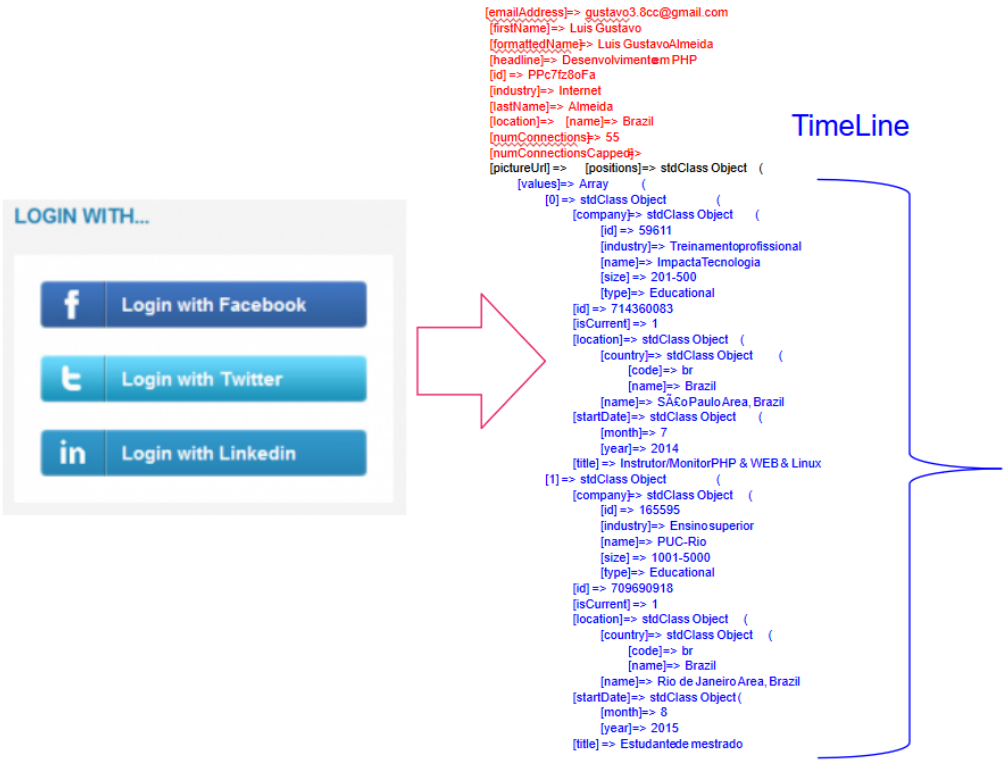


Figura 9: Exemplo de saída ao adotar autenticação OAuth2 no LinkedIn.

4.2.4
Coleta de Dados usando Requisições HTTP

A maioria dos rastreadores realiza uma pesquisa em amplitude (*Breadth-first search* - BFS) no grafo da rede social. A BFS termina quando o grafo inteiro é visitado ou, alternativamente, quando um dado critério de parada é cumprido. Estaremos assim percorrendo nosso grafo de interesse baseado em um esquema BFS primeiro recuperamos os links de acordo com algum critério de busca e depois acessamos os perfis.

Segundo HEYDON & NAKORK [47], o algoritmo básico executado por qualquer rastreador da Web recebe como entrada uma lista de URLs e executa repetidamente as seguintes etapas:

1) Remova um URL da lista de URL.

2) Determine o endereço IP do seu nome de host.

3) Faça o download do correspondente documento.

4) Extraia os links nele contidos.

5) Para cada um dos links extraídos, assegure-se de que é uma URL absoluta (desativando-o, se necessário) e adicione-o à lista de URLs a serem baixados, desde que não tenha sido encontrado antes.

Porém, conforme conclui MANGARAVITE [32], executar um rastreador é uma tarefa desafiante. De fato, é a tarefa mais frágil em motores de busca e não envolve apenas interação com centenas de milhares de servidores da Web, mas também com vários servidores de nomes que estão todos além do controle do sistema. Ele deve decidir cuidadosamente quais URLs devem visitar e em qual ordem. Enquanto os rastreadores tradicionais rastreiam de forma exaustiva as páginas na Web, outros incorporam foco para gerar tópicos, coleções específicas. Um 'rastreador focado' analisa sua fronteira para encontrar os links que provavelmente serão mais relevantes para o rastreamento, evitando regiões irrelevantes.

Rastreadores focados ou rastreadores tópicos são rastreadores de propósito especial que servem para obter da Web coleções menores e mais restritas de páginas [35]. Eles têm como objetivo principal rastrear de forma eficiente páginas que são relevantes para um tópico específico. Esse processo geralmente é realizado por meio de heurísticas adequadas.

Uma 'armadilha de rastreador' é definida por HEYDON & NAJORK [47] como sendo uma URL ou conjunto de URLs que fazem com que um rastreador entre em loop infinito. Algumas armadilhas não são intencionais. Outras são introduzidas intencionalmente. Não conhecemos nenhuma técnica automática para evitar armadilhas de rastreador. No entanto, os Websites que contêm armadilhas de rastreamento são facilmente percebidos devido ao grande número de documentos lá descobertos. Um operador humano pode verificar a existência de uma armadilha e excluir manualmente tal URL.

A solução proposta pode ser classificada como um rastreador focado, ao restringir seu leque de ação única e exclusivamente a um único Website, e somente à parte que contempla o mecanismo de busca de alumni referentes à PUC-Rio. Está excluída a possibilidade de haver 'armadilha de rastreador' pois o conjunto de sementes é previamente conhecido.

"Como efetuar a manutenção da sessão ativa do robô ?"

Os requisitos para ferramentas na área de testes de aplicativos da Web são para lidar com o AJAX e HTML dinâmico, uma ferramenta de código aberto

amplamente utilizada para testes de aplicativos da Web é o Selenium. As aplicações da Web tendem a evoluir continuamente e, portanto, precisam de testes completos. O Selenium (Figura 10) é um software de código aberto portátil disponível para Windows, Linux e Macintosh [83].

Temos referência ao uso da suíte Selenium[34] [84, 85, 86] enfatizando o caráter exclusivo de testes de software, porém não é abordada a questão de se usar o programa como mecanismo de entrada de dados em um sistema de Business Intelligence (BI). SINGH & SHARMA [84] discorrem sobre a suíte Selenium e devido ao advento das aplicações do tipo Rich Internet Application (RIA) faz do Selenium uma grande promessa, pois aplicações RIA possuem sua estrutura fortemente baseada em AJAX. Com o advento das RIAs a natureza dinâmica das aplicações AJAX é excelente para os usuários, mas pode se tornar um pesadelo para os testadores, pois o acompanhamento dos elementos aparecendo, desaparecendo, oculto e duplicado pode ficar complicado [83].

Selenium é uma ferramenta para testar aplicações Web pelo browser de forma automatizada. Selenium se refere ao *acceptance testing* (teste funcional) que envolve rodar testes num sistema finalizado. Os testes rodam diretamente no browser, exatamente como o usuário utilizaria tal software. O pacote Selenium tem quatro ferramentas: Selenium IDE, Selenium RC, Selenium Webdriver, Selenium Grid [84]. Adotaremos a ferramenta Selenium WebDriver.

O Selenium WebDriver é a ferramenta que carregará uma janela do navegador, e de forma autônoma se encarrega de fornecer login e senha de um usuário LinkedIn válido e assim navegar de forma autônoma e programática para recuperar as informações contidas no código HTML de cada página e salvar estes trechos diretamente no BD, para posterior processamento.

Para a linguagem PHP a aplicação do tipo *client* compatível para o Selenium WebDriver pode ser encontrada no seguinte endereço: https://github.com/facebook/php-webdriver

[34] http://www.seleniumhq.org

Figura 10: Página oficial da suíte de automação Selenium.

Assim que o programa de maneira autônoma carrega uma janela no navegador é possível armazenar todo, ou em partes, o conteúdo HTML. Outra característica do Selenium é que ele mantém a sessão ativa e isso é vital para que o programa se mantenha autenticado no LinkedIn e assim não perca a conexão autenticada.

A solução referente a este processo será composta de três etapas (Quadro 2):

1. Crawling LinkedIn IDs (ou hashes).
2. Crawling LinkedIn *profiles* (ou perfis).
3. Scraping HTML.

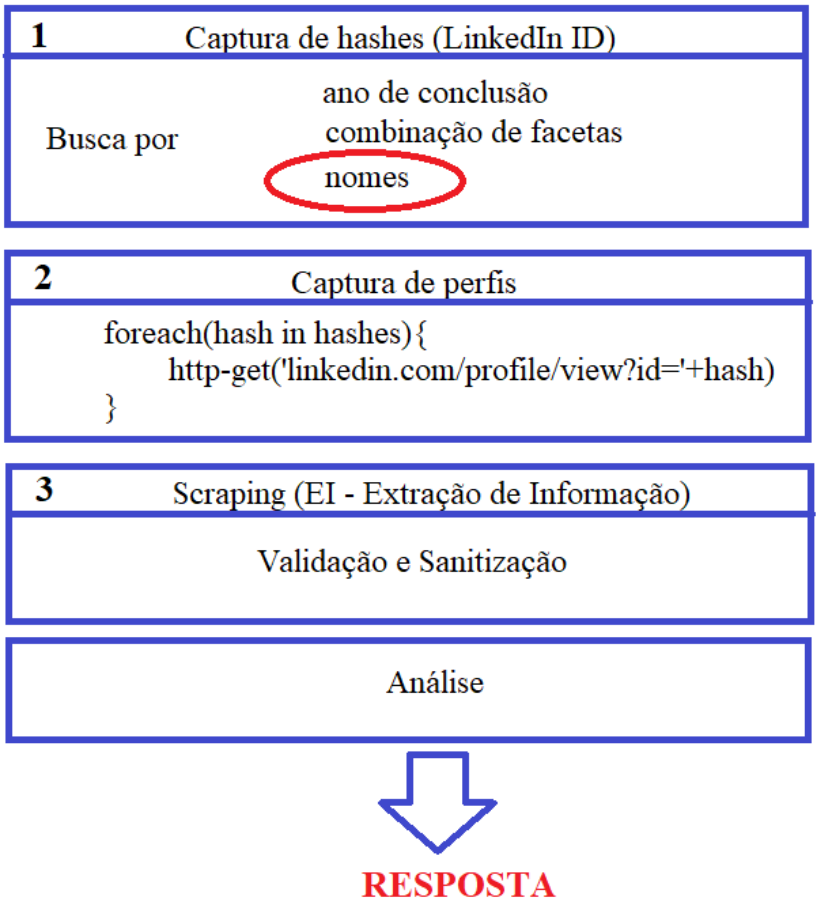
Durante o tempo de realização deste trabalho o LinkedIn mudou o visual de seu site, portanto em algumas imagens é possível que haja diferenças de layout.

Através da tabela denominada **alumni** vamos armazenar os ex-alunos levando em conta diversos campos. Cada LinkedIn ID que convencionamos chamar de **hash**, no site do LinkedIn terá seu respectivo valor armazenado no campo de mesmo nome **hash** na tabela **alumni** do banco de dados. Essa informação será usada para em etapa posterior, permitir ao robô navegar em páginas de perfis.

4.3
O Processo de Coleta de Dados do LinkedIn

A extração de dados de HTML geralmente é realizada por módulos de software chamados *wrappers* (ou *parsers*). As primeiras abordagens para o envolvimento de sites da Web foram baseadas em técnicas manuais [2, 9, 17, 4, 1].

Um **hash** será armazenado no campo de igual nome hash (com restrição do tipo *unique*) na tabela **alumni,** ao lado de um campo **id** cuja principal característica é ser auto-incrementável, este valor corresponde ao campo **alumni_id** nas tabelas **jobs** e **edus** (Figura 11). Essa estratégia evitará assim duplicidade de registros no banco de dados.



Quadro 2: Fluxograma da ferramenta dividido em três etapas.

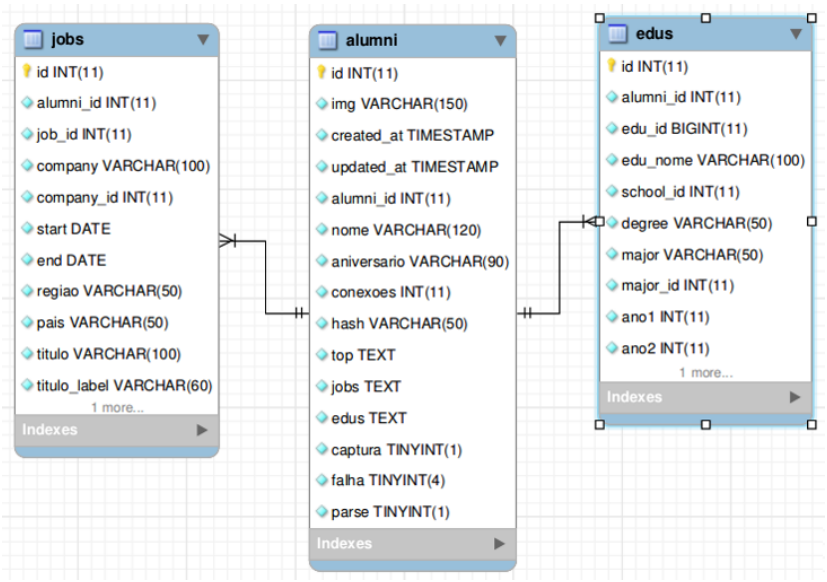


Figura 11: Principais tabelas do banco de dados.

4.3.1
Crawling LinkedIn Hashes

Referente à captura de hashes inicialmente teríamos a possibilidade de fazer três tipos de buscas:

- Busca por Ano de Conclusão.
- Busca por Combinações de Facetas.
- Busca por Nome.

Todas elas enfrentam a mesma dificuldade: limitação na paginação de resultados. Em média para um usuário comum e gratuito do LinkedIn são exibidas cerca de 10 páginas de resultados, onde cada página contém em média 10 hashes; a navegação entre as páginas Web é via *scrollDown* (rolagem da barra lateral). Os resultados a partir das páginas 11 em diante não serão exibidos para um usuário comum (Figura 12).

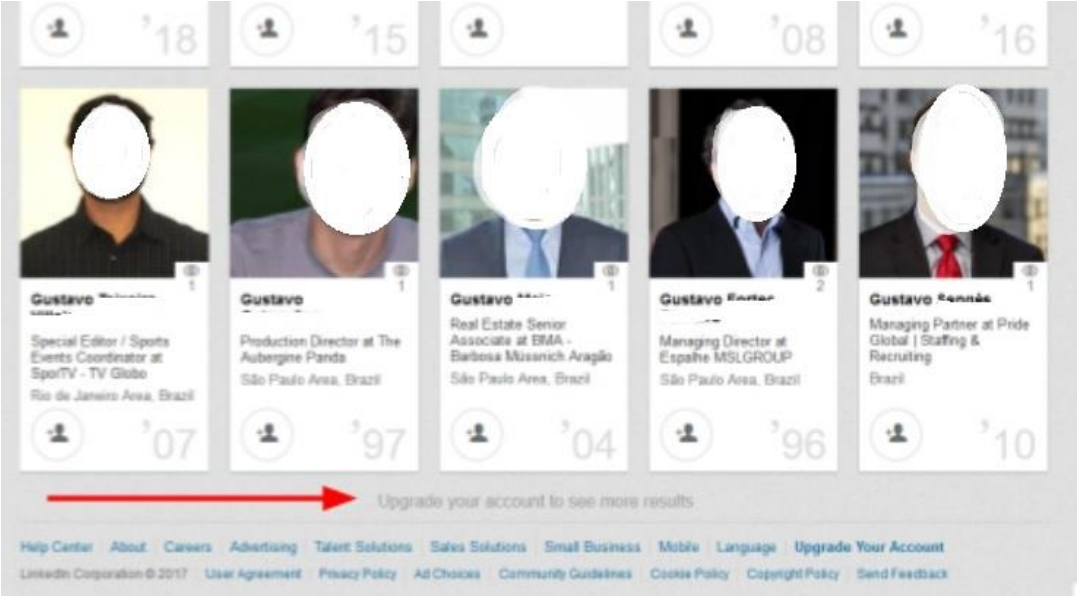


Figura 12: Um dos obstáculos do processo é a limitação da paginação.

Em meados de 2016 de acordo com as regras de negócio do LinkedIn referente a busca por alumni a cada requisição ao site obtemos em média cerca de 100 hashes, pois temos 10 resultados por página e 10 páginas de acordo com o perfil básico. A estratégia de busca escolhida será aquela em que a coleta de hashes oferecer a melhor relação custo/benefício (Tabela 2).

O critério de escolha das URLs do LinkedIn se configura como o coração da ferramenta ALUMNI onde a formação do conjunto de sementes, seus critérios de parada e a ordem de visitar estas URLs é que irão determinar a acurácia e precisão dos dados recuperados bem como o tempo de processamento para se atingir empiricamente o estado da arte, que neste caso seria recuperar 66.000 hashes com o mínimo de esforço do robô PHP.

Em qualquer uma das estratégias de coleta de hashes, o trecho referente a EI do único valor que nos interessa, o **hash**, é extraído com a navegação na árvore DOM da página HTML processada, combinado com uma expressão regular (Figura 13).

Figura 13: Trecho do programa onde usamos RegEx para extrair o hash.

4.3.1.1
Busca por Ano de Conclusão

A busca de acordo com o ano de conclusão seria a menos eficiente, pois se considerarmos o período que compreende do ano 1977 ao 2017, que é o intervalo de datas válido no LinkedIn, teríamos como resultado apenas 4.000 registros afinal 2017-1977=40 e, considerando que temos até 100 resultados por critério então teríamos como resultado 40x100=4.000 hashes (Quadro 3).

```
for(ano=1977; ano<=2017; ano++){
        HTML = http-get(url_LinkedIN + ano)
        parseHTMLandSaveData(HTML)
        paginas = extrai-total-paginas(HTML)
        for(i=2; i<=paginas; i++){
                HTML = http-get(url_LinkedIN + ano + i)
                parseHTMLandSaveData(HTML)
        }
}
```

Quadro 3: Pseudocódigo da etapa de busca por ano de conclusão.

4.3.1.2
Busca por Combinação de Facetas

Faceta é uma nomenclatura adotada pelo LinkedIn para elaborar o infográfico referente aos alumni de determinada instituição. Sob as facetas, pode-se comparar as carreiras dos graduados sob diferentes aspectos. Um exemplo seria "o que estudaram" ou ainda “onde vivem”.

Para registrar todas as possíveis facetas e seus valores é feito um único *scraping* manual na página oficial da PUC-Rio, para assim extrair os 20 valores de cada uma das 5 colunas do infográfico presentes na página da PUC-Rio. Após extrair os 100 valores referentes as facetas, estes valores são inseridos no BD na tabela **facetas** (Figura 15) e através de um script PHP geramos as combinações de facetas através de arranjos e permutações (Quadro 4) que chamamos de **rotas** (Figura 16). Cada uma destas rotas pertence ao conjunto das 15.504 requisições HTTP do tipo GET que podem ser usadas para coletar teoricamente até 100% dos registros. O resultado da combinação de facetas irá gerar 15.504 rotas por análise combinatória (Quadro 5).

A busca de acordo com a combinação das cinco facetas (combinadas ou separadas) que o LinkedIn aborda: ‘vivem’, ‘trabalham’, ‘fazem’, ‘estudaram’, ‘skills’ é uma estratégia que tem a vantagem de recuperar teoricamente até 100% dos hashes com o inconveniente de que deveríamos efetuar todas as combinações possíveis, resultando assim em um número imenso de requisições.

Cada valor presente no infográfico é um link cujo valor é uma string e cujo atributo *href* é composto por uma combinação de dois a quatro caracteres alfa, um ponto e um valor numérico, por exemplo CC.7293 é um valor presente na coluna ‘onde trabalham’ e na linha que corresponde à Rede Globo, portanto para o LinkedIn o código CC corresponde ao parâmetro facetCurrentCompany e 7293 corresponde à Rede Globo. Isso pode ser usado para se montar uma URL parametrizada e obter conjuntos de registros que obedeçam tais critérios.

Para se organizar a montagem destas combinações (**rotas**) vamos levar em conta somente os dois caracteres alfabéticos à esquerda do atributo *href*. Para se correlacionar o que está armazenado no BD com aquilo que vamos ter que usar

na URL criamos uma convenção em que se define para cada faceta uma nomenclatura, um código e o correspondente parâmetro da URL (Tabela 1).

Através de combinatória vamos processar estes 100 valores presentes na tabela **facetas** e assim cada resultado desta análise corresponde à uma rota e este valor é armazenado na tabela **rotas** (Figura 16). Por exemplo, uma **rota** cujo valor seja a string CN.24,CC.7293,G..6046 após passar pela função PHP **formataRota** (Figura 14) é gerado uma saída do tipo string como por exemplo *facetCurrentCompany=7293&facetCurrentFunction=24&facetGeoRegion=6046* (podemos usar até 5 parâmetros) e assim irá ser combinada na URL padrão gerando a seguinte URL: https://www.linkedin.com/school/165595/alumni/?facetCurrentCompany=7293&facetCurrentFunction=24&facetGeoRegion=6046 e, neste caso, acessando essa URL o robô irá obter 81 resultados (Figura 17).

Faceta URL	Nomenclatura	Código Faceta
facetCurrentCompany	trabalham	CC
facetCurrentFunction	fazem	CN
facetGeoRegion	vivem	G.
facetFieldOfStudy	estudaram	FS
facetSkillExplicit	skills	KE

Tabela 1: Correspondência entre nomenclatura das facetas do LinkedIn.

```php
#formataRota('CN.23,KE.154,CC.420');

function formataRota(string $campo){

    $labels = [
                'G.'=>"facetGeoRegion",
                'CC'=>"facetCurrentCompany",
                'CN'=>"facetCurrentFunction",
                'FS'=>"facetFieldOfStudy",
                'KE'=>"facetSkillExplicit",
               ];

    $partes = explode(',',$campo);
    $rota="?";
    foreach ($partes as $parte) {
        $segmentos = explode('.',$parte);
        $letra = $segmentos[0];
        $num = $segmentos[1];
        $rota.="{$labels[$letra]}=$num&";
    }

    $rota = trim($rota,'&');
    return $rota;

}
```

Figura 14: Função que converte uma rota em parâmetros de facetas para a URL.


```
1 • select * from facetas
2
```

Result Grid | Filter Rows: | Edit: | Export/Impo

#	id	faceta	name	faceta_id
47	50	trabalho	The CocaCola Company	CC.1694
48	51	trabalho	Technip	CC.5124
49	52	trabalho	Shell	CC.1271
50	53	trabalho	Estacio de Sa	CC.325339
51	54	trabalho	Globo com	CC.8198
52	55	fazem	Desenvolvimento comercial	CN.4
53	56	fazem	Educacao	CN.7

Figura 15: Exemplo de registros da tabela facetas.

```
1 • select * from rotas
```

Result Grid | Filter Rows:

#	id	rota
1...	16221	CN.23,KE.2185
1...	16222	CN.23,KE.176
1...	16227	CN.23,KE.1836
1...	16228	CN.23,KE.107
1...	16229	CN.23,KE.727
1...	16230	FS.101475,KE.366
1...	16231	FS.101475,KE.163
1...	16232	FS.101475,KE.274

Figura 16: Exemplos de registros da tabela rotas.

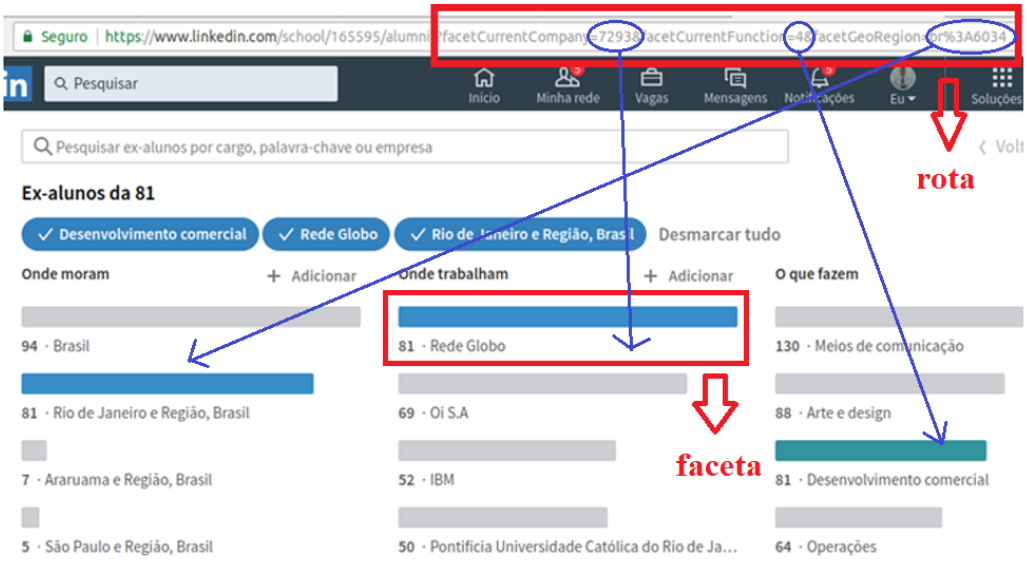


Figura 17: Resultados obtidos ao se combinar valores de facetas em uma rota.

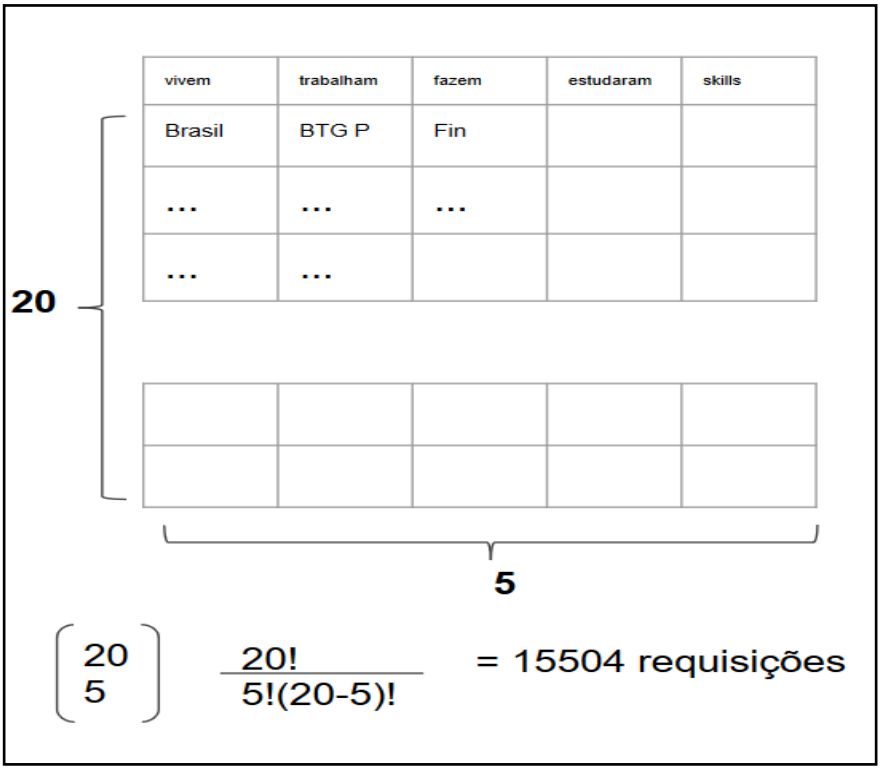
Além do fato já mencionado do alto número de requisições, na prática esta abordagem se mostrou insuficiente, pois a grande maioria das combinações gerava uma amostragem muito baixa (próximo de 10 hashes) ou excessivamente alta (algo em torno de 2.000 hashes).

```
facetas = 'SELECT faceta_id FROM facetas';
foreach(facetas as faceta){
        rotas[ ]=analiseCombinatoria(facetas, faceta)
}

foreach(rotas as rota){
        HTML = http-get(url_LinkedIN + rota)
        parseHTMLandSaveData(HTML)
        paginas = extrai-total-paginas(HTML)
        for(i=2; i<=paginas; i++){
                HTML = http-get(url_LinkedIN + rota + i)
                parseHTMLandSaveData(HTML)
        }
}
```

Quadro 4: Pseudocódigo da etapa de busca por facetas.

Esta é uma busca ineficaz se formos levar em conta o número de requisições, porém é a única busca que em tese irá garantir um êxito de até 100% na recuperação de dados, pois explora todas as possibilidades combinatórias.



Quadro 5: Análise combinatória do total de possibilidades usando facetas.

4.3.1.3
Busca por Nome

A busca escolhida após exaustivos testes práticos foi a **Busca por Nome** (Figura 18). Com uma lista de cerca de 1.800 nomes simples (Figura 21), conseguimos o melhor desempenho (cerca de 90% do total de ex-alunos) e ainda com aproximadamente 1.800 requisições. Este desempenho pode ser melhorado caso seja usada uma lista de nomes mais bem elaborada, por exemplo com nomes compostos, ou ainda uma lista de nomes fornecida pela própria instituição, o que aumentaria a acurácia e precisão do processo.

Um ponto fraco da estratégia escolhida inclui situações como o caso de um alumnus que tenha um nome fora da lista, este não iria ser recuperado, assim como nomes bem comuns poderiam resultar em cenários ruins, por exemplo se tivermos 101 registros homônimos 'João da Silva' pelo menos um registro não seria recuperado, afinal em qualquer busca por 'João' ou 'Silva' consideramos até 100 registros por requisição. O estado da arte da ferramenta estaria atrelado a qualidade da lista, a distribuição dos nomes de alumni no LinkedIn e como o LinkedIn processa e devolve esta informação, afinal nada impede que exista uma regra de negócio no LinkedIn que processe de maneira igual os resultados para Luis, Luís, Luiz, Luiza e Luíza. Esse conhecimento está fora da nossa alçada.

Busca por	Núm. Requisições HTTP	Núm. Máximo alumni	%	Alumni/Req
Ano	40	4.000	6	100
Facetas	15.504	66.000	100	4,25
Nome	1.800	57.901	88	32,16

Tabela 2: Melhor relação alumni recuperados por requisição.

Aqui cabe falar um pouco sobre a qualidade da lista, pois nome comuns como José, Luis e Maria irão sempre gerar grandes resultados pois são nomes muito comuns no Brasil, porém somente 100 resultados iriam ser recuperados frente às limitações da paginação. No caso de nomes compostos registros que não foram recuperados com um nome, teriam outra possibilidade de serem

recuperados com outro nome da lista, por exemplo ‘Luis Gustavo Almeida’ terá três chances de ser recuperado pelo crawler.

Essa lista poderia ser criada junto ao departamento administrativo da faculdade com uma lista dos nomes de ex-alunos mais comuns pertencentes à Universidade, ranqueados em ordem decrescente e ordenados alfabeticamente para eventualmente excluir algum par manualmente. Pode-se também fazer uma lista de nomes compostos mais comuns, ou ainda fazer uma intersecção dos nomes mais comuns no Brasil com os nomes mais comuns da Universidade.

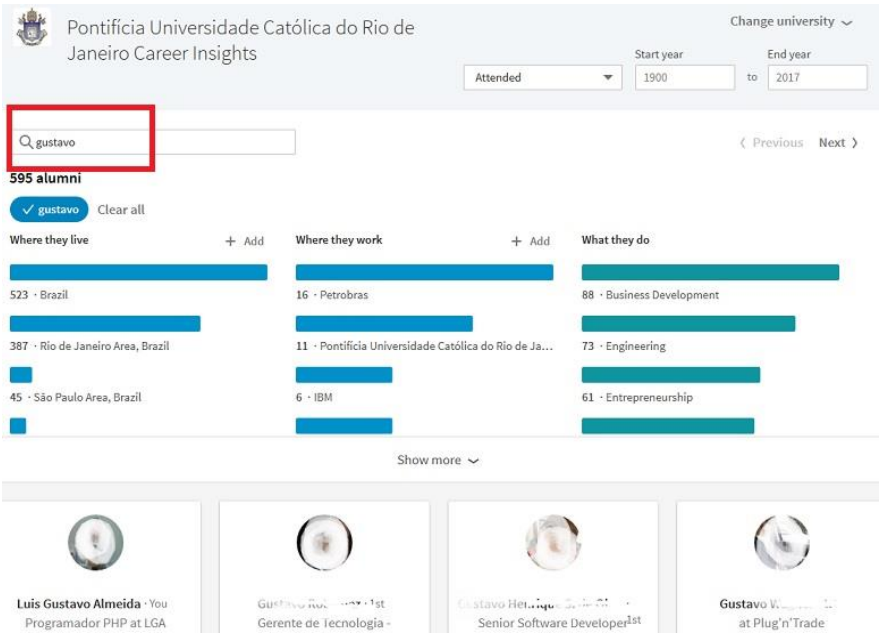


Figura 18: Resultado de uma busca por nome.

Como podemos perceber na Tabela 3 a lista de nomes ideal seria a lista em que nomes como Alan, que resultam em 74 registros, seriam todos perfeitamente recuperados frente aos limites da paginação. Em uma conta bem simples e supondo que não houvesse homônimos, precisaríamos de hipoteticamente 66.000/100 = 660 nomes diferentes para preencher 100% do objetivo, ainda que desconsiderando nomes compostos. A lista adotada possui um caráter aleatório, com 1.813 nomes simples e recuperamos cerca de 90%. Isso garantiu a melhor relação registros/requisições conforme se verifica na

Tabela 2.

Jose	869
Maria	1315
Gustavo	595
Almeida	1400
Alberto	239
Marcos Paulo	6
Alan	74

Tabela 3: Alguns exemplos de registros encontrados para alguns nomes.

Um outro fato relativo ao esquema de paginação e listagem do LinkedIn diz respeito à distribuição destes alumni na paginação com uma certa inclinação em dar prioridade a usuários com grau de conexão mais próximo do usuário ativo do robô (ou seja, eu, o autor). A regra de negócio responsável pela exibição dos resultados da busca não é conhecida, porém é claramente notado a influência do grau de proximidade, para supostamente encontrar antigos amigos de faculdade por exemplo.

Neste caso os pressupostos de design do LinkedIn acabam sendo um inimigo da acurácia da ferramenta. Atualmente essa dificuldade de recuperar hashes diferentes é notada à medida em que os resultados são exibidos dando preferência ao menor grau de conexão (Figura 20).

Por exemplo se "Joaquim Barbosa da Silva" é meu amigo (grau de conexão 1) e "Maria Barbosa Antonieta" é amiga de "Joaquim Barbosa da Silva" (grau de conexão 2 comigo ou, conceitualmente, *foaf*[35]), e supondo que um dos nomes da lista seja "Joaquim" em um primeiro momento este será recuperado normalmente, porém se o próximo nome da lista for "Barbosa" a busca terá uma inclinação em dar prioridade novamente a "Joaquim Barbosa da Silva" e este **hash** que já fora recuperado irá aparecer novamente na busca, tomando a preferência de um outro registro que ainda não foi recuperado, a partir deste

[35] Foaf : friend of a friend. Ontologia para descrever relacionamento entre indivíduos de uma mesma rede social.

momento o **hash** referente a “Joaquim Barbosa da Silva” será enviado pela ferramenta ao BD e assim teremos duas inserções repetidas no BD que obviamente devido a restrição do tipo *unique* não é incluído no BD. No caso de termos mais de 100 registros na paginação, a situação somente tende a se complicar mais.

É óbvio que a minha rede de grau 2 é muito maior que a minha rede de grau 1, que no momento de execução deste trabalho gira em torno de 350, o excesso de conexões do usuário robô também pode se transformar em um problema nesta etapa à medida em que se dá prioridade a exibir registros repetidos de grau 1. Afinal para recuperar hashes basta que este alumnus alvo esteja na minha rede *foaf* e a minha rede *foaf* cresce à medida que cresce a minha popularidade.

O processo de leitura dos nomes da lista consiste em processar linha a linha (um laço no arquivo), sendo que cada linha do arquivo corresponde a um nome e assim vamos concatenar este nome com a URL padrão para assim efetuar uma requisição HTTP do tipo GET correspondente (Quadro 6), por exemplo: se um dos nomes da lista for *gustavo* então a URL final será www.linkedin.com/school/165595/alumni/?keywords=gustavo

```
nomes = extractFile('nomes.txt');
foreach(nomes as nome){
        HTML = http-get(url_LinkedIN + nome)
        parseHTMLandSaveData(HTML);
        paginas = extrai-total-paginas(HTML)
        for(i=2; i<=paginas; i++){
                HTML = http-get((url_LinkedIN + nome + i)
                parseHTMLandSaveData(HTML);
        }
}
```

Quadro 6: Pseudocódigo da etapa de busca por nomes.

Observou-se que acentos costumam gerar resultados inconsistentes. Portanto, antes de se montar a URL correspondente é necessário passar por uma

função que remove todo e qualquer tipo de acento; esta função é uma combinação das funções *remove_accents* e *seems_utf8* (Figura 19).

```php
function remove_accents($string) {
    if ( !preg_match('/[\x80-\xff]/', $string) )
        return $string;

    if (seems_utf8($string)) {
        $chars = array(
        // Decompositions for Latin-1 Supplement
        chr(195).chr(128) => 'A', chr(195).chr(129) => 'A',
        chr(195).chr(130) => 'A', chr(195).chr(131) => 'A',
        chr(195).chr(132) => 'A', chr(195).chr(133) => 'A',
        chr(195).chr(135) => 'C', chr(195).chr(136) => 'E',
        chr(195).chr(137) => 'E', chr(195).chr(138) => 'E',
        chr(195).chr(139) => 'E', chr(195).chr(140) => 'I',
        $chars['out'] = "EfSZszYcYuAAAAAACEEEEIIIINOOOOOOUUUUYaaa

        $string = strtr($string, $chars['in'], $chars['out']);
        $double_chars['in'] = array(chr(140), chr(156), chr(198),
        $double_chars['out'] = array('OE', 'oe', 'AE', 'DH', 'TH'
        $string = str_replace($double_chars['in'], $double_chars[
    }

    return $string;
}
```

```php
function seems_utf8($str){
    $length = strlen($str);
    for ($i=0; $i < $length; $i++) {
        $c = ord($str[$i]);
        if ($c < 0x80) $n = 0; # 0bbbbbbb
        elseif (($c & 0xE0) == 0xC0) $n=1; #
        elseif (($c & 0xF0) == 0xE0) $n=2; #
        elseif (($c & 0xF8) == 0xF0) $n=3; #
        elseif (($c & 0xFC) == 0xF8) $n=4; #
        elseif (($c & 0xFE) == 0xFC) $n=5; #
        else return false; # Does not match
        for ($j=0; $j<$n; $j++) { # n bytes
            if ((++$i == $length) || ((ord($
                return false;
        }
    }
    return true;
}
```

Figura 19: Uso de funções combinadas para remover acentuação.

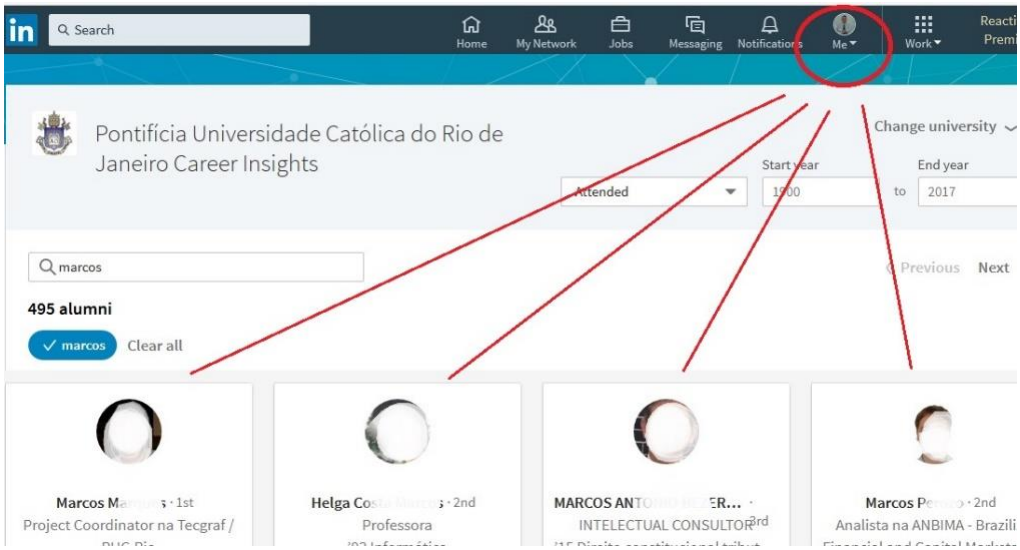


Figura 20: Priorização de exibir usuários com grau de proximidade.

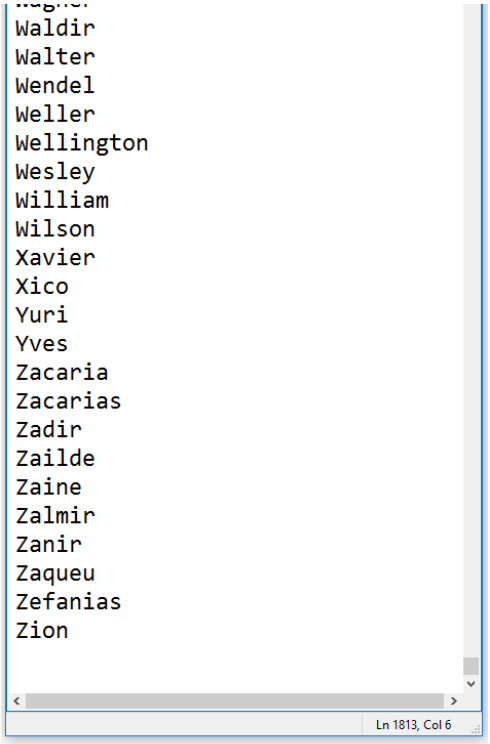


Figura 21: Lista usada no programa, contendo apenas 1.813 nomes simples.

4.3.2
Crawling de Perfis

As páginas Web que descrevem um indivíduo em uma rede online social são tipicamente bem estruturadas, pois geralmente são geradas de forma automática. Isto implica um código HTML bem definido, ao contrário das demais páginas da Web, que podem ser publicadas por qualquer pessoa [41]. Portanto, podemos

ter certeza sobre quais dados podem ser obtidos depois de rastrear as páginas da Web de uma determinada PSN.

Uma vez que todos os hashes estão recuperados no BD, basta programar o robô para fazer um laço e assim visitar a URL correspondente no LinkedIn. Por exemplo, se o **hash** armazenado é AAMAABqD5EcBqBmtDD, então a URL final será www.linkedin.com/profile/view?id=AAMAABqD5EcBqBmtDD (Quadro 7). No caso de não acessarmos o perfil então o HTML será vazio e uma flag correspondente a este fato é registrada junto à este hash para futuras atualizações.

Uma vez que o robô tenha acesso a esta página de determinado perfil, adotamos intercalar intervalos de cerca de dois segundos ao longo do script (Figura 22). A prática de bombardear requisições a Websites é conhecida como *RapidFire* e sendo assim queremos evitar este tipo de ocorrência. Esse intervalo (que poderia ser randômico) é feito no sentido de obter o correto armazenamento do HTML correspondente a tal página.


```
$hashs = $results->fetchAll(PDO::FETCH_ASSOC);
foreach($hashs as $row){
    $hash = $row['hash'];
    $u = "https://www.linkedin.com/profile/view?id=".$hash;
    $webDriver->get($u);
    sleep(2);
    $top=$jobs=$edus=$aniversario=false;
    ################ trecho referente a top
    try {
        $top=$webDriver->findElement(WebDriverBy::id('top-card'))->getAttribute('outerHTML');
        saveHTMLPerfil_top($top,$hash);

    } catch (NoSuchElementException $e) {
    }
    ################ trecho referente a jobs
    try {
        $jobs=$webDriver->findElement(WebDriverBy::id('background-experience'))->getAttribute('
            outerHTML');
        saveHTMLPerfil_jobs($jobs,$hash);
    } catch (NoSuchElementException $e) {
    }
    ################ trecho referente a edus
    try {
        $edus=$webDriver->findElement(WebDriverBy::id('background-education'))->getAttribute('
            outerHTML');
        saveHTMLPerfil_edus($edus,$hash);
    } catch (NoSuchElementException $e) {
    }
```

Figura 22: Recebe o perfil em HTML e salva em top, edus e jobs no BD.

O LinkedIn adota um critério chamado grau de conexão que vai de 1 a 3+ (Figura 24) onde o grau 1 corresponde a conexão direta, o grau 2 seria uma conexão do tipo *foaf* e o grau 3+ seria uma distância maior que estas duas iniciais. O acesso a página do perfil de um alumnus é vinculado a condição de existir uma

relação de grau menor que 3 entre o usuário do robô e o usuário do perfil em questão, ou seja, perfis LinkedIn com grau 3+ não são acessíveis (Figura 26). Este problema pode ser minimizado se aumentarmos o número de conexões do usuário LinkedIn do robô, assim a probabilidade do grau de conexão com qualquer perfil aleatório ser menor que três diminui à medida que aumentamos a popularidade do usuário robô.

"Eu dependo da popularidade de meus amigos."

Conforme esclarecimentos anteriores, um alto número de conexões de grau 1 poderia resultar em um número elevado de hashes repetidos na etapa anterior. O ideal seria ter poucos amigos para que a paginação de hashes na etapa 1 tenha um caráter mais aleatório. Porém populares, para que o alcance da minha rede *foaf* aumente e assim o acesso aos perfis da etapa 2 estaria garantido, pois isso aumenta a probabilidade do próximo alumnus estar dentro da minha rede *foaf*.

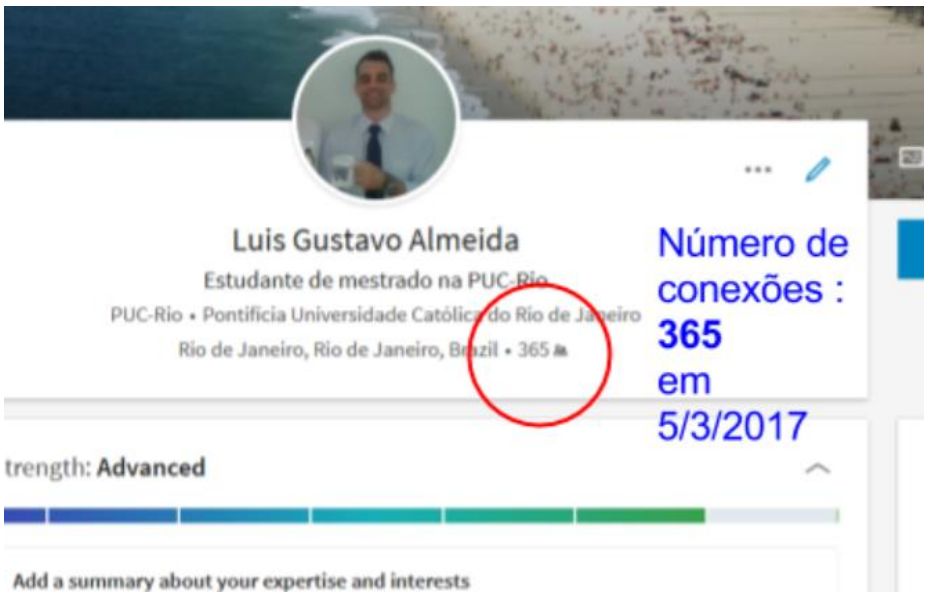


Figura 23: O número de conexões do usuário corrente é fundamental.

```
hashs = query('SELECT hash FROM alumni WHERE parse=0')
foreach(hashs as hash){
        HTML = http-get(url_LinkedIN + hash)
        if(parseHTMLandSaveData(HTML)){
                'UPDATE alumni SET parse=1 WHERE hash=hash'
        }else{
                'UPDATE alumni SET falha=2 WHERE hash=hash'
        }
}
```

Quadro 7: Pseudocódigo da etapa de recuperação de perfis (etapa 2).

Aumentar o número de conexões da rede de amigos do LinkedIn pode ser conseguido através de convidar mais usuários (e ser aceito) ou receber convites de amizade (e aceitar). A primeira poderia esbarrar em uma questão de SPAM e assim receber severas punições, restando a segunda opção.

Porém não é uma tarefa simples, muito menos rápida, pois para isso, seria necessário produzir conteúdo relevante, angariar seguidores enfim tornar-se popular, interessante e uma fonte de ajuda ou inspiração para outros que, assim, se sintam compelidos a pedir conexão.

Ter poucos amigos, ou ainda ter amigos com também poucos amigos na rede, impacta sensivelmente a acurácia da ferramenta pois diminui o potencial de alcance da minha rede *foaf*, e assim diminui o número de alumni recuperados, afinal podemos requisitar todas as páginas normalmente, mas o que vai definir se vamos recuperar a informação será o grau de conexão vigente ser menor que 3.

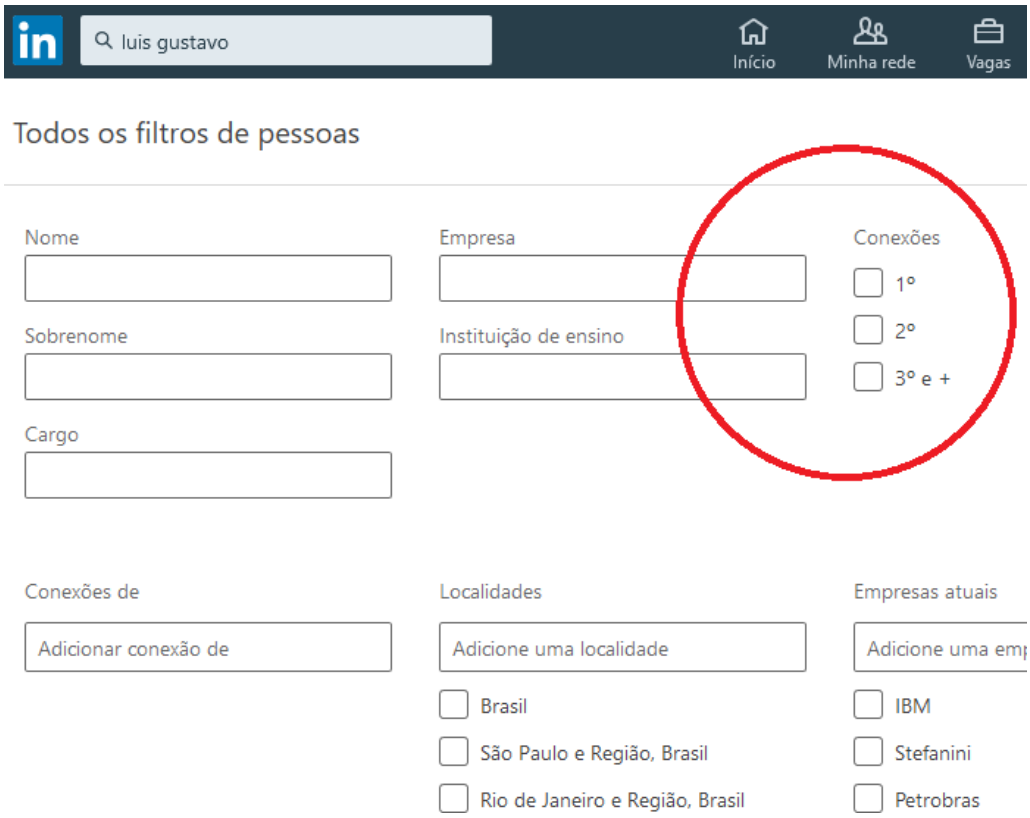


Figura 24: Existem até 3 graus de separação entre usuários LinkedIn.

Vale ressaltar que as configurações de perfil privado valem somente para acesso público (acesso ao perfil sem login), os históricos referente a **jobs** e **edus** são recuperados integralmente independente do perfil ser privado (Figura 25).

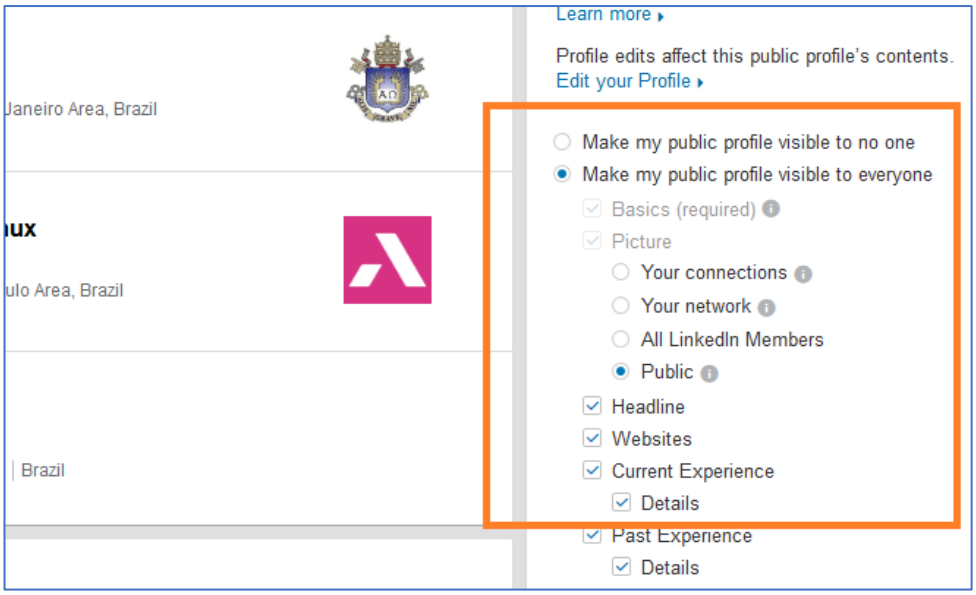


Figura 25: O usuário pode optar por indisponibilizar seu perfil publicamente.

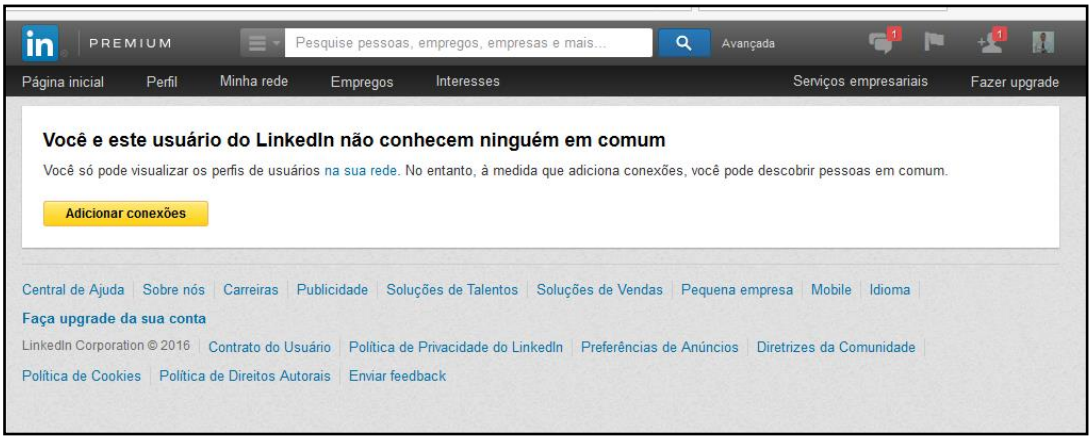


Figura 26: Principal problema encontrado na etapa crawling de perfis.

4.3.3
Scraping HTML

Em uma página HTML, os nós de cada documento são organizados em uma árvore DOM (Figura 27). Os objetos na árvore DOM podem ser endereçados e manipulados pelo uso de métodos sobre os objetos. A classe nativa da linguagem PHP que manipula o DOM é a DOMDocument[36]. A classe DOMDocument representa um documento inteiro HTML ou XML; sua instância serve como a raiz da árvore de documentos, assim percorre-se a árvore DOM até o nó desejado, extraindo dados com o uso de Expressões Regulares (RegEx).

PCRE[37] é o padrão que a linguagem PHP implementa RegEx; as funções nativas do PHP que estamos utilizando aqui são as funções *preg_match*, *preg_replace* e *preg_match_all*. Na maioria das situações faremos uso de uma combinação de se navegar na árvore DOM e, uma vez definido o nó HTML, usaremos uma RegEx para extrair seu conteúdo.

O resultado da extração de dados do perfil de cada alumnus será armazenado na tabela **alumni**, de acordo com os seguintes trechos do perfil HTML e que possuem campos de mesmo nome no BD:

- **top**: trecho de HTML que inclui a foto, nome e o número de conexões.
- **jobs**: trecho que corresponde ao histórico profissional.

[36] http://php.net/manual/pt_BR/class.domdocument.php

[37] http://secure.php.net/manual/pt_BR/book.pcre.php

- **edus**: trecho correspondente ao histórico acadêmico.

As tabelas **jobs** e **edus** receberão os dados provenientes do HTML destes campos correspondentes; os valores são extraídos com uma combinação de navegação no DOM combinado com o uso de uma expressão regular adequada para o dado em questão. Para isso basta executar o script PHP responsável pelo *parsing* (Quadro 8). Este script então varre a tabela **alumni** inicialmente selecionando os registros marcados com a flag parse=1 e para cada registro analisa os campos **top**, **edus** e **jobs** que contém os trechos do HTML correspondente e efetua uma triagem para as tabelas **jobs** e **edus** (Figura 28), através de funções auxiliares (Figura 30). Assim fazemos o populamento das tabelas com os registros e no caso de haver falhas nestes campos a função ajusta as flags necessárias para que isso seja corrigido depois.

```
lines = query('SELECT hash,top,edus,jobs FROM alumni WHERE parse=1')
extract(lines)
foreach(hashes as hash){
        saveHTMLPerfil_top(top,hash)
        saveHTMLPerfil_edus(edus,hash)
        saveHTMLPerfil_jobs(jobs,hash)
}
```

Quadro 8: Pseudocódigo da etapa de parsing do HTML.

O LinkedIn fornece um número de identificação para cada job e edu que será aproveitado para garantir a unicidade dos dados nas tabelas correspondentes (Figura 29), estes valores serão armazenados nos campos **job_id** e o **edu_id**.

- Campos armazenados na tabela **edus**:

- id = identificador único do registro, campo com auto-incremento.
- alumni_id = chave estrangeira referente à tabela alumni (id).
- edu_id = valor aproveitado da estrutura HTML, valor único.
- degree = grau ou tipo de curso, ex: graduação, pós, mestrado, etc.
- major = curso informado pelo usuário e que deverá ser mapeado.
- major_id = código numérico referente a major.
- school = nome da instituição de ensino.
- school_id = código numérico referente a school.
- ano1 = ano correspondente à data de início do registro.

- ano2 = ano correspondente à data de final do registro.

- Campos armazenados na tabela **jobs**:

- id = identificador único do registro, campo com auto-incremento.
- alumni_id = chave estrangeira referente a tabela alumni (id).
- job_id = valor aproveitado da estrutura HTML, valor único.
- regiao = região geográfica.
- pais = país referente ao trabalho atual.
- título = profissão.
- titulo_label = SLUG para título.
- company = nome da empresa.
- company_id = código numérico referente a company.
- start = ano correspondente à data de início do registro.
- end = ano correspondente à data de início do registro.

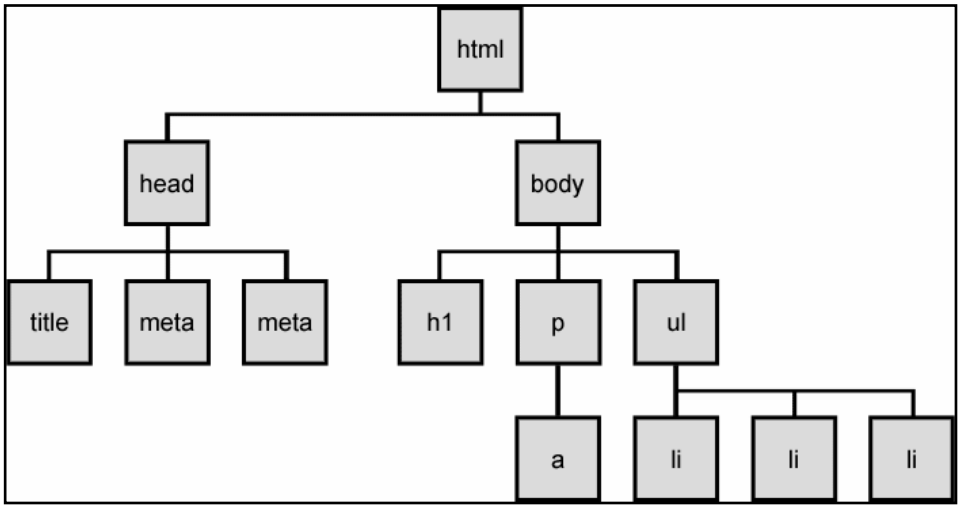


Figura 27: Exemplo de uma árvore DOM.

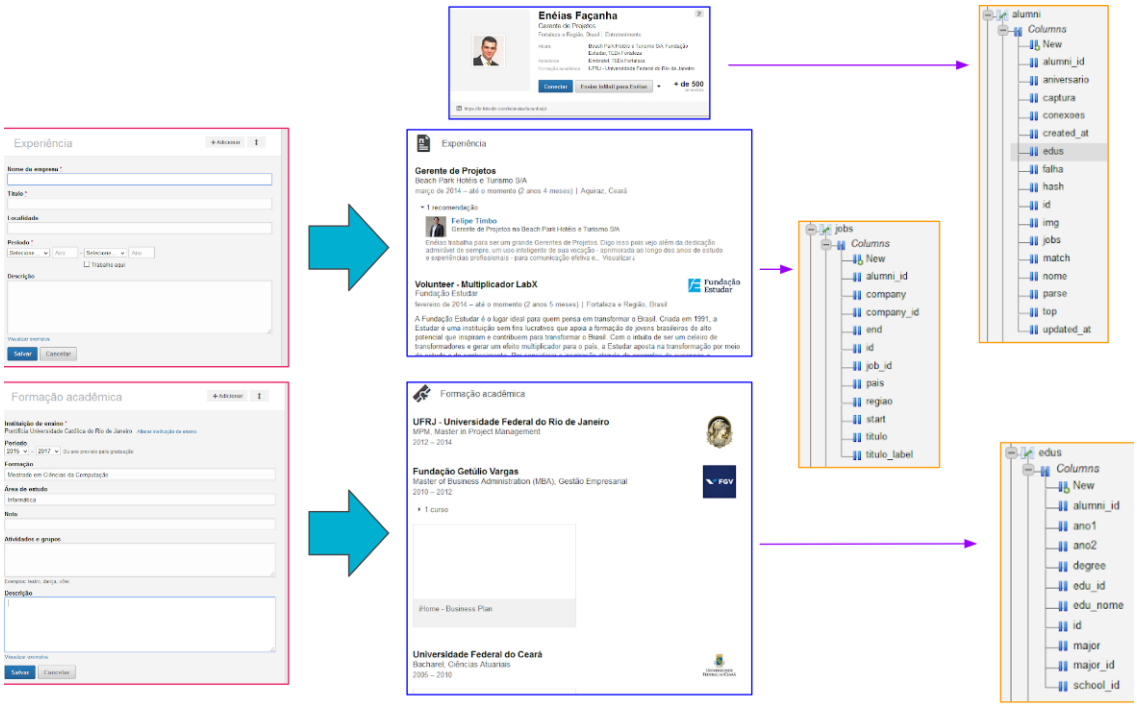


Figura 28: Informações cadastradas no LinkedIn e suas correlações no BD.

Figura 29: Exemplo de um trecho de HTML da página de perfil.

```php
function saveHTMLPerfil_edus($edus,$hash){
    global $pdo;
    $tabela = "alumni";
    $pk="hash";

    if($edus) {
        $perfil['edus']=$edus;
        $perfil['parse']=2;
        $perfil['hash']=$hash;
    } else {
        $perfil['parse']=3;
        $perfil['falha']=3;
    }

    $set="";
    foreach ($perfil as $campo => $valor) {
        $set .= $campo .'=:'.$campo.",";
    }
    $set = trim($set,",");

    $q = sprintf("UPDATE %s SET %s WHERE %s='%s';",$tabela,$set,$pk,$hash);
    echo "\nEDUS:".$q;
    $stm = $pdo->prepare($q);
    foreach($perfil as $campo => $valor){
        $tipo = getTipo($valor);
        $stm->bindValue(':'.$campo, $valor, $tipo);
    }

    $stm->execute();

}
```

Figura 30: Função para edus que no caso de erros usa a flag adequada no BD.

4.4
Sanitização e Validação

Uma vez recuperados os dados das páginas do LinkedIn, o próximo passo consiste na validação, limpeza e higienização dos dados. Esta etapa será feita através de regras T-SQL, combinadas com scripts em PHP. Por exemplo, podemos eliminar hashes duplicados, ou ainda **jobs** que não tenham correlação com a tabela **alumni** (registros órfãos).

Um dos problemas encontrados nesta etapa refere-se ao *encoding* de dados textuais, onde nomes como *Alberto ChÃƒÂ¡vez LÃƒÂ³pez, Marcos ValladÃ£o e FlÃƒÂ¡vio Silva Jardim*, devem ser convertidos para sua forma correta. Isso foi solucionado com uma função onde mapeamos os caracteres errados.

A função escrita em PHP **limpaStr2BD** (Figura 31) converte cada caractere para seu respectivo código ASCII e, caso este código numérico esteja fora de determinados intervalos, o caractere é simplesmente descartado na reconstrução da nova string.

```php
function limpaStr2BD($str){
    ######## intervalos permitidos : da white list
    #32 - 38 = espaco - &
    #40 - 90 = ( - Z
    #97 - 122 = a - z

    $permitidos1 = range(32,38);
    $permitidos2 = range(40,90);
    $permitidos3 = range(97,122);
    $permit = array_merge($permitidos1,$permitidos2,$permi

    $str_nova = "";

    for($i=0;$i<strlen($str);$i++){
        if(in_array(ord($str[$i]),$permit)){
            $str_nova .= $str[$i];
        }
    }

    return $str_nova;
}
```

Figura 31: Função em PHP responsável por limpar caracteres irregulares.

Outras impurezas apresentadas e que foram corrigidas:

1) Espaçamento único entre todos os nomes, evitando assim caracteres como [.,-] entre nomes compostos como, por exemplo, *Sefaz- RJ*, *Secretaria da Fazenda - Rio de Janeiro* foi convertido para *Sefaz RJ Secretaria da Fazenda Rio de Janeiro*.
2) Sensibilidade a maiúsculas e minúsculas.
3) Retirada de acentos.

Para contornar o problema da análise de possíveis registros duplicados usaremos como critério o conceito acerca de consultas SQL baseado em funções de agregação para se chegar a respostas sobre perfis duplicados e assim eliminar alguma redundância, mesmo tendo sido usado com critério campos com restrição do tipo *unique*.

Uma função de agregação processa um conjunto de valores contidos em uma única coluna de uma tabela e retorna um único valor como resultado. Vamos usar consultas SQL baseadas em funções de agregação para identificar registros repetidos e assim excluí-los.

Outros problemas que podem ocorrer é que alguns perfis não apresentam **top** e/ou **edus** e/ou **jobs**, afinal o usuário pode simplesmente não preenchê-los nos formulários do LinkedIn. Ou ainda estes campos podem se perder devido aos gargalos das conexões HTTP, onde eventualmente pacotes TCP/IP se perdem no meio da requisição, ou ainda qualquer outro erro de natureza técnica.

Assim como uma simples instrução T-SQL irá mostrar quais dados são duplicados, obtendo o total de repetidos, podemos usar outra instrução para relatar um conjunto de registros incompletos. No primeiro caso simplesmente deletamos do BD no segundo caso adotaremos *flags*, caso o registro apresente algum problema.

Um erro que também pode ocorrer é o trecho de HTML que pode estar corrompido. Neste caso ao se processar o referido campo caso haja algum erro é atribuído uma flag para o erro (parse=2) e outra para o tipo de erro (falha=2/3/4). A relação de valores possíveis para os campos **parse** e **falha** estão nas Tabelas 4 e 5. A Figura 31 ilustra um exemplo de código onde se faz uso das flags para marcação de campos do BD com erros e que deverão ser tratados depois.

valor parse	Significado
0	Hash registrado na 1ª etapa
1	Perfil capturado na 2ª etapa
2	Efetuado o *parsing* do HTML do registro com sucesso
3	Ocorreu um erro no *parsing* do HTML

Tabela 4: Possíveis valores para o campo *parse*.

valor falha	Significado
0	registro incluído (valor default)
1	registro vazio
2	campo top com problemas/vazio
3	campo edus com problemas/vazio
4	campo jobs com problemas/vazio

Tabela 5: Possíveis valores para o campo *falha*.

Uma simples consulta T-SQL na tabela **alumni** onde o campo **parse=3** trará todos os registros com erros. Se quisermos ser mais específicos basta

acrescentarmos o campo falha na consulta, exemplo: *select * from alumni where parse=3 and falha in ('2','3','4')* onde respectivamente 2,3,4 correspondem a **top**, **edus** e **jobs**. De posse destas informações temos um controle mais preciso sobre os registros problemáticos e então devemos rodar novamente o crawling focado aos registros com problemas e corrigir sobrescrevendo os campos HTML novamente (Quadro 9). A função **parseHTMLandSaveData** trata o HTML novamente e caso o resultado desta vez esteja correto ajusta a flag parse=2 para este registro, indicando para o sistema que neste caso a correção foi feita com sucesso.

```
hashs=query('SELECT hash FROM alumni WHERE parse= 3 and falha in (2,3,4)')
foreach(hashs as hash){
        HTML = http-get(url_LinkedIN + hash)
        if(parseHTMLandSaveData(HTML)){
                'UPDATE alumni SET parse= 2 WHERE hash=hash'
        }
}
```

Quadro 9: Pseudocódigo para se re-processar perfis com HTML falho.

4.5
Incluindo Dados da Plataforma Lattes

A base de dados Lattes será analisada parcialmente, deixando uma lacuna para a possibilidade de se incluir outras bases de dados de valores semanticamente compatíveis com a estrutura do LinkedIn, ou seja, com histórico profissional e educacional contendo datas de início e conclusão e nomes de entidades.

A base de dados Lattes permite exportar os currículos de um grupo de pessoas em um arquivo XML, que geralmente não é de domínio público, e é concedido exclusivamente aos departamentos solicitantes de determinadas universidades.

Para uma melhor homogeneização é desejável que a parte do XML que trata do histórico profissional deve ser o mais próximo possível da estrutura **jobs**. A relação de equivalência entre **jobs** do LinkedIn e **jobs** do Lattes/XML, bem como **edus** do LinkedIn e **edus** do Lattes/XML é que irá determinar a inclusão ou não na ferramenta de mais uma fonte de dados.

A tarefa de fazer a correspondência entre a estrutura do LinkedIn e a estrutura do XML Lattes é efetuar uma leitura do arquivo XML extraindo os dados através das funções nativas do PHP referentes a XML[38] e assim salvar nas tabelas **jobs_lattes** e **edus_lattes**. Neste caso em específico podemos usar a mesma estrutura de dados das tabelas **edus** e **jobs**, ou seja, estabelecer uma total equivalência entre as duas diferentes fontes de dados.

A Tabela 6 faz um comparativo entre os campos do trecho XML correspondente e os campos das tabelas **edus** e **jobs**, observamos que conseguimos associar corretamente todos os campos das duas fontes de dados, inclusive no que tange às datas (Figura 32).

Tabela	Campo	Nó XML
alumni	nome	NOME-COMPLETO
alumni	hash	NUMERO-IDENTIFICADOR
jobs	start	MES-INICIO + ANO-INICIO
jobs	end	MES-FIM + ANO-FIM
jobs	company	NOME-INSTITUICAO
jobs	company_id	CODIGO-INSTITUICAO
jobs	titulo	ENQUADRAMENTO-FUNCIONAL
edus	ano1	ANO-DE-INICIO
edus	ano2	ANO-DE-CONCLUSAO
edus	degree	<TAG RAIZ>
edus	major	NOME-CURSO
edus	major_id	CODIGO-CURSO
edus	edu_nome	NOME-INSTITUICAO
edus	school_id	CODIGO-INSTITUICAO

Tabela 6: Equivalência entre campos do XML Lattes e BD.

[38] http://php.net/manual/pt_BR/book.simplexml.php

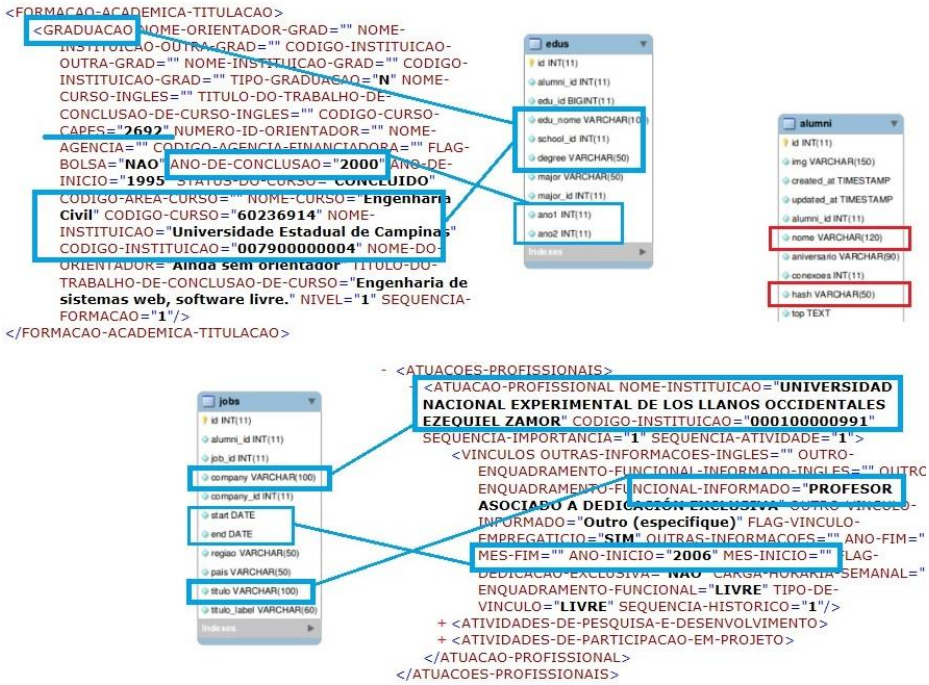


Figura 32: Mapeamento entre o XML e os campos do BD.

Devido ao fato de neste caso a estrutura de dados das tabelas ter por coincidência uma total equivalência, os dados do Lattes podem ser incluídos diretamente nas tabelas edus e jobs, bastando apenas adicionar uma flag booleana na tabela para se diferenciar a origem dos dados e assim poderíamos seguir com a criação de cenários para interpretação dos dados. É óbvio que esta inclusão impactará os resultados por exemplo, se levarmos em consideração que professor é um cargo de relevância e sendo o Lattes uma base composta em sua maioria por professores o resultado penderá para o lado da balança em que o Lattes se encontra.

4.6
Conclusões do Capítulo

Neste capítulo enfocamos as etapas responsáveis pela obtenção e tratamento dos dados. Abordamos conceitos e estratégias para obtenção dos dados e, além disso, destacamos a parte técnica.

O resultado obtido foi positivo, e atesta que o conceito por trás da ideia de se usar um programa voltado para teste funcional como mecanismo de manutenção da sessão ativa se mostrou eficaz. Assim, contribui para que outros

trabalhos futuros, mesmo que fora do contexto de pesquisa de alumni, usem esta sistemática para assim aprimorar outros rastreadores.

A adoção de uma segunda fonte de dados e seu respectivo enquadramento na estrutura de dados se mostrou conveniente no sentido de ampliar a dimensão do trabalho, ideal para futuros trabalhos que explorem o caráter de DW da ferramenta.

5.
Cenários de Uso

Diversos cenários ou aplicações podem ser criados para se interpretar os dados e responder perguntas pertinentes a formação acadêmica e empregabilidade. Vamos estudar alguns aspectos sobre as áreas mais destacadas e extrair indicadores que possam dar forma a estes cenários e por fim abordaremos a principal pergunta deste trabalho.

5.1
Introdução

A aplicação final referente ao módulo **web** será desenvolvida tendo como base consultas complexas em T-SQL no banco de dados. Tais resultados serão exibidos no formato JSON[39]. Assim o script PHP fará o carregamento destes dados para posterior visualização na página Web.

A primeira demanda da aplicação correspondente ao módulo **web** trata-se de uma página que replica e abstrai toda a essência do trabalho em uma linha do tempo interna com um link disponível para o LinkedIn do alumnus em questão (Figura 33). A ideia aqui é exatamente poder comparar em tempo real o resultado recuperado no BD com o LinkedIn equivalente.

[39] http://www.json.org

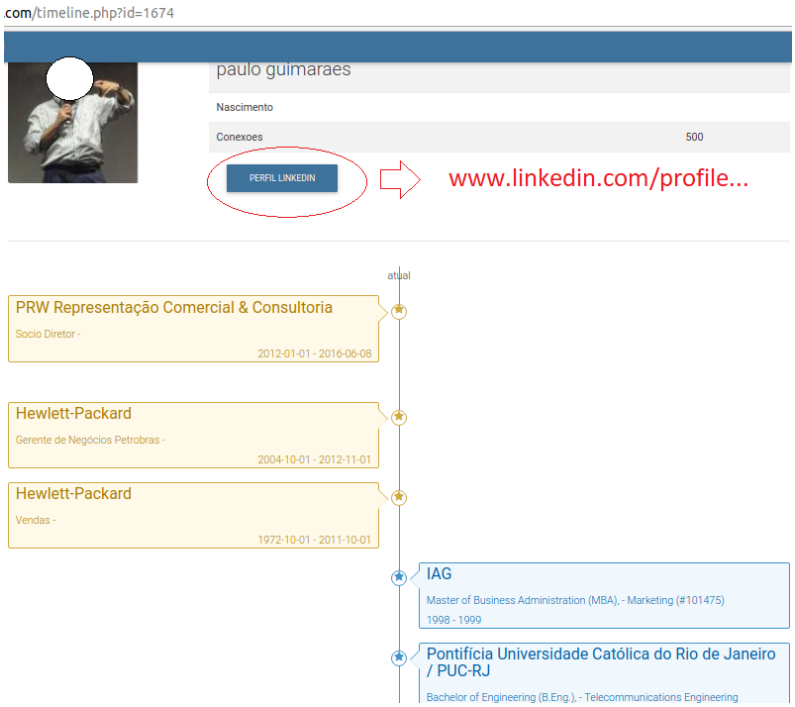


Figura 33: Timeline do módulo web com verificação on-the-fly no LinkedIn.

5.2
Dados Coletados

A Tabela 7 resume estatísticas sobre os dados coletados. A próxima etapa de análise responderá às perguntas iniciais, sendo o resultado final armazenado em um banco de dados MySQL com 9 tabelas. Nosso objetivo é chegar o mais próximo possível de 66.000. Entretanto devido a diversos fatores já discutidos anteriormente a Tabela 8 apresenta os totais mais importantes no que diz respeito à RI.

Tabela	Registros
alumni	57.901
edus	118.911
empresas	11.344
escolas	587
facetas	126
jobs	199.705
majors	292
usuarios	2
Total de registros	**388.868**

Tabela 7: Totais de registros recuperados do LinkedIn.

Total aprox. de alumni no LinkedIn em 2016 = 66.000		
Descrição	**Coletado**	**% do objetivo**
Total de hashes	57.901	88%
Total de alumni	43.113	66%

Tabela 8: Percentual de alumni recuperados.

Temos cerca de 118.000 registros de **edus** versus 200.000 **jobs** e cerca de 11.000 empresas, 580 escolas e 300 **majors** (cursos), que representam bem mais que os cerca de 30 cursos de graduação que são oferecidos na PUC-Rio. Aqui ressaltamos que no caso de **empresas**, **escolas** e **majors** estes valores são sugeridos na hora do preenchimento do formulário e possuem os valores correspondentes numéricos (Tabela 10).

Contabilizamos diversos resultados únicos que essencialmente derivam dos valores obtidos em **edus** e **jobs**. Estas grandezas serão estudadas mais a fundo nas próximas seções. Foram criadas tabelas separadas para **majors**, **empresas** e **escolas** onde, tal qual a estratégia abordada em **edus** e **jobs,** aproveitamos o próprio código numérico fornecido pelo LinkedIn. Infelizmente o LinkedIn não segue a mesma lógica para os campos **degree** e **título**. Esta mesma lógica não existe nem no campo **título**, que corresponde à profissão do

alumnus, nem no campo **degree**, que corresponde ao tipo de curso. Portanto, deveremos seguir alguma heurística para processar estas informações.

A tabela **majors**, cuja tradução seria cursos, é uma tabela que emerge como um resultado de todos os valores únicos encontrados no campo **major** na tabela **edus**. Assim ocorre também com as tabelas **escolas** e **empresas**, as quais serão populadas com valores únicos encontrados em **edus (school** e **school_id)** e **jobs (company** e **company_id)**, respectivamente. A título de exemplo para a *PUC-Rio*, o código que a identifica em **escolas** é o 10582 (campo school_id), assim como o código da *IBM* na tabela **empresas** é 1009 (campo company_id).

Referente a **major** (cursos), um exemplo de registro seria *Programmer* que corresponde ao código 100179 (campo major_id). Não existe um curso de graduação chamado *Programmer* nem tampouco *Programador*, devemos estabelecer uma relação de equivalência, ou seja, todo **major** futuramente deverá ser mapeado para um *curso*. Como exemplo neste caso *Programmer* será mapeado para *Informática*.

Estes campos (major_id, school_id e company_id) impediriam que um usuário do LinkedIn cadastrasse empresas, escolas e cursos com duplicidade ou não legíveis. Infelizmente **título** e **degree** não fazem parte deste esquema e sua representação é baseada em string, o que sem dúvida gera um enorme conjunto de valores e possibilidades.

A Tabela 10 informa o número total de valores distintos correspondentes aos campos degree, título e titulo_label. Esse é o universo de strings que precisam ser categorizadas e que não possuem um valor numérico de referência O campo **major** também será mapeado, porém este possui apenas 292 ocorrências por justamente possuir vínculo numérico no momento do preenchimento do formulário.

Devido ao fato do LinkedIn usar sempre duas datas para qualquer **job** ou **edu** podemos subentender que trata-se dos anos iniciais e final de cada registro e isso nos leva a crer por exemplo no caso de **jobs** que estamos nos referindo ao ano da contratação e desligamento da empresa; assim como no caso de **edus** estamos nos referindo ao ano em que se inicia um curso e o ano em que se conclui um curso qualquer. Assim, cursos de graduação podemos fazer uma analogia destas duas datas com o vestibular e formatura, assim como fazer uma associação entre a admissão e desligamento no caso de empregos (Tabela 9).

Equivalência	jobs	edus
ano1	Admissão	Vestibular
ano2	Desligamento	Formatura

Tabela 9: Nomenclatura inferida com as datas de início e fim.

Campo	Tabela	Total
school_id	edus	587
degree	edus	25.594
titulo	jobs	82.267
titulo_label (SLUG)	jobs	44.487
major_id	edus	292
company_id	jobs	11.344

Tabela 10: Número máximo de ocorrências distintas.

5.3
Análise Preliminar

Nesta seção vamos nos ater a estudar quais estatísticas podemos obter e assim, poder responder algumas questões que foram abordadas no início (Capítulo 1).

(1) Quais são os cursos e/ou áreas mais procurados?

As respostas estão concentradas nos rankings da tabela **majors** (Figura 34) e também nos rankings da tabela **edus** referente ao **título** (Figura 35) e seu correspondente SLUG[40] **titulo_label** (Figura 36). A leitura destes dois rankings pode dizer quais profissões e áreas de estudo são mais proeminentes no mercado.

[40] SLUG – nomenclatura usada em URL legível para humanos e máquinas. Exemplo: analista-de-producao pode ser usado em uma URL e pode ser entendido por um ser humano.

```
select distinct
    format((count(e.id)/25226)*100,2) as va
    ,substring(m.major,1,32) as label
    from edus as e
    left join majors as m using(major_id)
    where e.major_id > 0
    group by e.major_id
order by value desc limit 12
```

Result Grid | Filter Rows: | Export: | Wrap Cell Co

#	value	label
1	7.26	Economics
2	6.11	Business Strategy Operations
3	4.00	Psychology
4	3.44	Computer Science
5	3.41	Electrical Engineering Decision
6	2.71	Information Systems
7	2.70	Corporate Finance Capital Mark
8	2.48	Wearable Computers Beauty Techn
9	2.14	Design
10	16.70	marketing
11	14.69	Business Administration and Mana

Figura 34: Ranking de majors onde Economics desponta em 7.26% em edus.

```
select count(1) as total,titulo from jobs group by titul
having titulo <> ''
order by total desc
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

#	total	titulo
1	3137	Estagiario
2	2805	ESTAGIARIA
3	2238	Intern
4	1862	Analista de Sistemas
5	1669	PROFESSOR
6	1444	Trainee
7	1295	Project Manager
8	1178	advogada
9	1161	Consultant
10	1038	Designer
11	1000	Owner
12	944	Advogado
13	941	Partner
14	796	Gerente de Projetos

Figura 35: Ranking do campo título com valores absolutos.

```
1 • select count(1) as total,titulo_label from jobs group by titulo_lab
2   having titulo_label <> ''
3   order by total desc
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

#	total	titulo_label
1	2235	intern
2	1861	analista-de-sistemas
3	1669	professor
4	1443	trainee
5	1295	project-manager
6	1168	advogada
7	1160	consultant
8	1037	designer
9	1000	owner
10	942	advogado
11	941	partner
12	795	gerente-de-projetos
13	753	consultor
14	714	diretor

Figura 36: Ranking do campo titulo_label (SLUG) com valores absolutos.

(2) Quais empresas mais contratam?

De acordo com a Figura 38 podemos observar um ranking das principais empresas presentes nestes dados que compõem a *timeline* de um alumnus, vale lembrar que cada empresa também possui uma página específica no LinkedIn onde é possível obter algumas informações destas empresas.

Na tabela empresas temos os campos company e company_id, podemos usar o valor de company_id para acessar a página da empresa no LinkedIn. Por exemplo para a IBM cujo company_id é 1009 ao se acessar a URL www.linkedin.com/company/1009 temos acesso a página da IBM onde a informação mais relevante e que poderia ser usada para mensurar o tamanho da empresa é o número de funcionários presentes no LinkedIn (Figura 37).

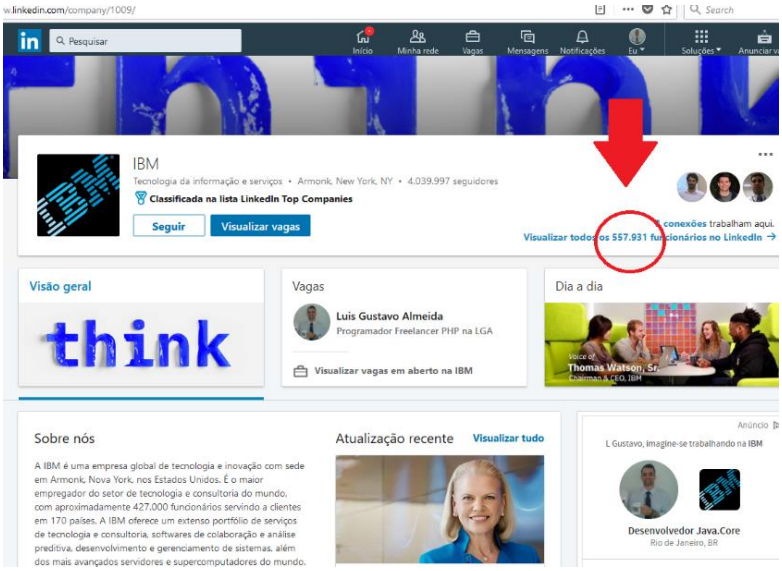


Figura 37: Página de uma empresa no LinkedIn.

```
select distinct
    format((count(j.id)/76551)*100,2) as
    ,substring(e.company,1,15) as label
    from jobs as j
    left join empresas as e using(compan
    where e.company_id > 0
    group by e.company_id
order by value desc limit 10
```

Result Grid | Filter Rows: | Export: | Wrap Cel

#	value	label
1	7.30	PUC Rio
2	5.09	Infotec Petrob
3	5.02	Medicos Sem Fro
4	3.72	IBM
5	3.56	Rede Globo
6	2.91	Telemar
7	2.02	Accenture
8	2.01	Vale
9	1.77	Embratel
10	1.60	TIM Brasil

Figura 38: Ranking de empresas referente à tabela jobs.

(3) Quais cursos de pós-graduação são mais procurados?

Conforme a Figura 39 estamos elencando as principais nomenclaturas de degree encontradas em **edus**, é importante frisar que para uma correta interpretação dos dados deve-se levar em conta as strings com o mesmo significado e grafias diferentes por exemplo *Bacharel* e *Bachelor* que se referem à bacharelado.

```
1 • select count(1) as total,degree from edus group by degree
2   having degree <> ''
3   order by total desc
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

#	total	degree
1	4594	MBA
2	4553	Bachelors Degree
3	3324	Master of Business Adminis
4	3170	Bacharel
5	2734	Graduacao
6	2041	Bachelor
7	2034	Masters degree
8	1818	Pos Graduacao
9	1711	Pos-Graduacao
10	1077	Master
11	1038	Mestrado
12	1013	Master of Business Adminis
13	990	Graduation

Figura 39: Ranking do campo degree presente na tabela edus.

(4) Qual é a média do tempo gasto em graduação e/ou pós-graduação?

De acordo com o gráfico da Figura 40 observamos que 22% dos registros de **edus** exceto a PUC-Rio (Universidades concorrentes) tem 1 ano de duração (isso sugere curso de extensão) enquanto que quando restringimos a **edus** pertencentes a PUC-Rio este número vai para 19% com duração de 4 anos (isso sugere uma graduação). No gráfico da Figura 41 observamos quais são as outras instituições fora a PUC-Rio (Universidades concorrentes) que possuem mais registros de cursos em **edus**. Ou seja podemos analisar qual(is) seria(m) a(s) principal(is) concorrente(s) da PUC-Rio.

Figura 40: Exemplo de gráficos no módulo web.

Figura 41: Exemplo de gráficos no módulo web.

(5) Qual o tempo médio de um emprego por empresa?

(6) Qual o tempo médio empregado por alumni?

(7) Estatísticas sobre alunos desempregados.

Referente à informação sobre empregabilidade ou cargo atual, esta é obtida única e exclusivamente através da presença da string ‘o momento’ conforme a Figura 42 onde na etapa 3 o *parsing* PHP é configurado para através de uma expressão regular adequada informar à ferramenta que este **job** trata-se de um emprego atual. Somente o usuário pode alterar tal registro, no caso de falecimento de usuário o LinkedIn disponibiliza um recurso chamado *deceased member* onde outros usuários informam tal falecimento e a partir daí abre-se um processo interno para remoção do perfil (Figura 43).

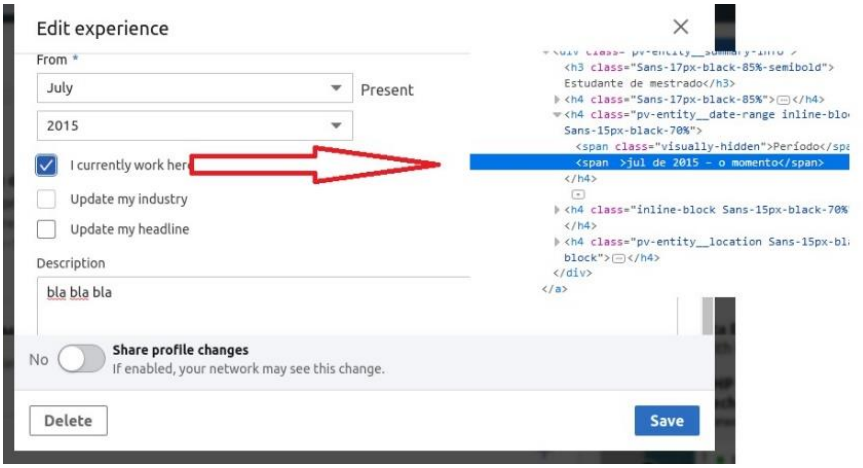


Figura 42: O conceito de emprego atual é baseado na marcação do checkbox.

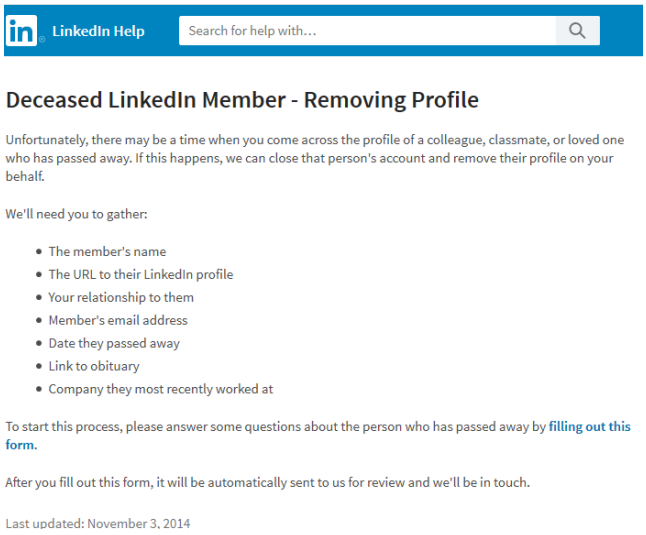


Figura 43: Política do LinkedIn para falecimentos de usuários.

No caso de mudança de cargo (título) em uma mesma empresa o usuário tem a opção de cadastrar mais um **job** com a mesma empresa porém com um título diferente, ou pode alterar o título do **job** atual e assim esta informação seria processada somente após um outro futuro crawling da ferramenta. **A interpretação dos dados está vinculada a uma *snapshot* do crawling.**

A figura 46 representa a interface com alguns indicadores que são obtidos com base nessas premissas, e através de consultas SQL, por exemplo averiguamos que em média estes alumni possuem 5 empregos, 3 cursos, possuem em média 21 anos empregados e 7 anos de tempo gasto em cursos incluindo graduação e pós. A média de tempo destes **jobs** são 4,82 anos bem como a média de duração destes **edus** são de 2,76 anos. Cerca de 1400 alumni não possuem registro de emprego no perfil. Esses números estão vinculados à snapshot do crawling atual.

(8) Total de Alumni Recuperados

Número de hashes recuperados através de métodos de busca, referentes a etapa 1. O objetivo seria algo em torno de 66.000 registros (Figura 44), obtivemos 57901.

(9) Total de Alumni Processados

Número de perfis processados e cuja triagem para as tabelas **edus** e **jobs** ocorreu normalmente, ou seja, registros em **alumni** que tenham filhos em **edus** e/ou **jobs**. Como o campo **nome** é obrigatório no perfil analisamos quais

registros tiveram o nome extraído corretamente (Figura 44). Ou seja se o alumnus tem o campo nome vazio é porque não foi possível coletar o HTML, por estar fora da rede *foaf*. Obtivemos 43.113 sendo que a diferença de 14.788 podem ser recuperados em tese caso o número de conexões do usuário robô aumente e assim efetuarmos um novo crawling destes perfis.

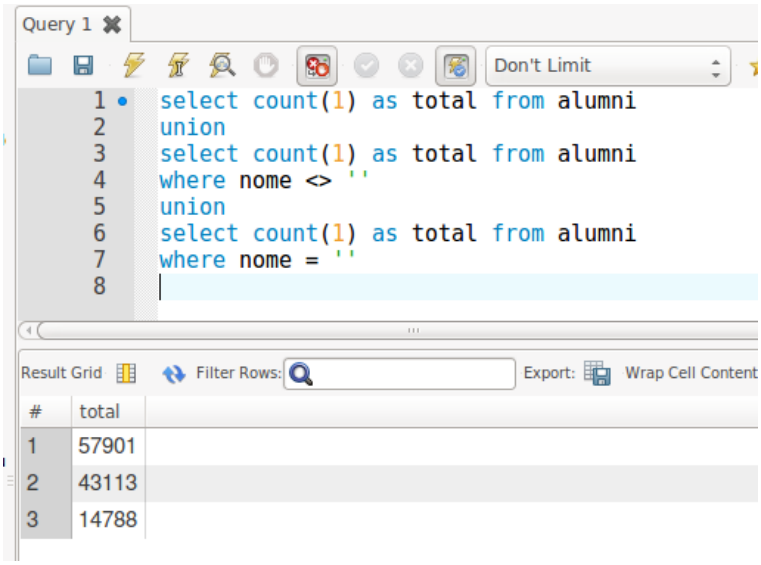


Figura 44: Número de hashes e perfis obtidos ao final do crawling.

(10) Número médio de conexões para cada alumnus

Lembrando que caso se tenha mais de 500 conexões o LinkedIn exibe apenas como **500+** (Figura 45), esta análise fica invalidada, afinal por exemplo esse número não corresponde à realidade. Esta média seria como se o máximo de conexões fosse 500, pois é esse dado que alimenta o sistema.

Figura 45: A partir de 500 conexões esse número é mostrado como 500+.

Figura 46: Interface com diversos indicadores obtidos através de consultas SQL.

Vale ressaltar que toda e qualquer análise estatística é estritamente baseada em valores informados pelo usuário. Em alguns casos no momento da digitação ele pode receber uma lista de sugestões de acordo com os primeiros caracteres digitados e assim optar por uma delas. Assim é natural que um ou mais valores se referem a um mesmo dado, por exemplo as regiões 'Rio de Janeiro – Area', 'Rio de Janeiro, Rio de Janeiro, Brazil' e 'Rio de Janeiro' (Figura 48) fazem referência a três valores distintos informados na caixa de sugestão que podem até significar 'a grande Rio de Janeiro' e a 'capital do Rio de Janeiro' respectivamente. No momento da análise isso pode levar à interpretações duvidosas (Figura 47). Diferentemente por exemplo de 'Espírito Santo, Vitória' onde neste caso temos uma string diferente para a capital e outra para o estado, tanto 'Rio de Janeiro' quanto 'São Paulo' por exemplo darão margem a interpretações equivocadas, portanto devemos levar este fator em consideração.

```
select distinct format((count(a.id)/
    (
        select sum(num_regioes_jobs) as total from (
            select distinct count(a.id) as num_regioes_jobs,
            substring(j.regiao,1,20) from alumni as a
            left join jobs as j using(alumni_id)
            where length(j.regiao)>2
            group by j.regiao order by num_regioes_jobs desc
        ) as q
    )
)*100,2) as value, substring(j.regiao,1,25) as label from alumni as a
left join jobs as j using(alumni_id)
where length(j.regiao)>2 group by j.regiao order by value desc limit 12
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	value	label
1	30.40	Rio de Janeiro Area
2	3.63	Sao Paulo Area
3	19.40	Rio de Janeiro e Regiao
4	11.33	Rio de Janeiro
5	1.77	Sao Paulo
6	1.61	Sao Paulo e Regiao
7	1.16	Brazil
8	0.65	Brasil
9	0.50	Niteroi
10	0.39	London

Figura 47: Ranking de regiões obtida através da tabela jobs.

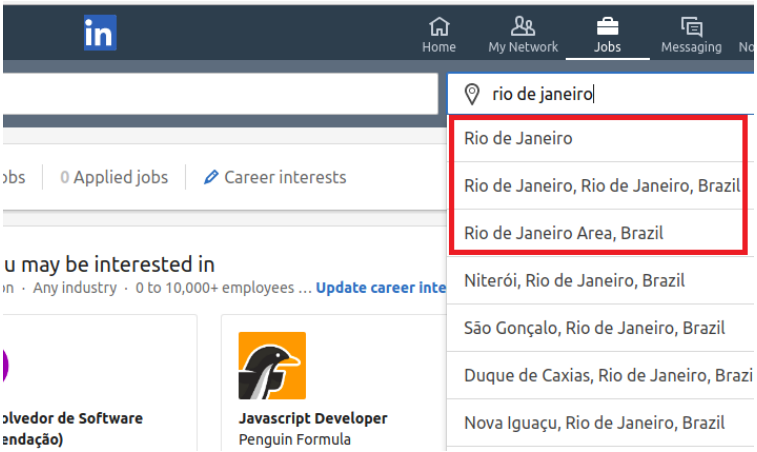


Figura 48: Sugestões do LinkedIn que podem gerar interpretações confusas.

5.4
Análise Aprofundada

Nesta seção, trataremos algumas questões sobre os alumni que possuem uma maior complexidade.

(1) Quanto tempo leva um alumnus de graduação PUC-Rio a assumir um cargo de relevância?

Respondidas as perguntas preliminares, podemos responder então quanto tempo em média um ex-aluno de graduação leva para chegar a um cargo de relevância.

A partir desta pergunta automaticamente surgem outras duas perguntas:

- O que vem a ser um alumnus de graduação PUC-Rio?

- O que vem a ser um cargo de relevância?

A primeira questão é definir o que seria um *cargo de relevância,* e como isto poderia ser mapeado frente às diversas combinações que o LinkedIn permite que seus usuários preencham estas informações nos formulários.

A segunda tarefa, igualmente complexa, é definir o que seria uma *graduação.* Por exemplo o usuário pode registrar seu **degree** como *graduação* ou *bacharelado*, ou ainda pode ter registrado em inglês ou ainda em outro idioma. É fato que a *PUC-Rio* é representada pelo código *10582* e agora a filtragem deve se concentrar em encontrar registros em **edus** que correspondam a graduação cujo school_id=10582 e registros em **jobs** cujo campo título atendam à condição de relevância (Tabela 11). Devemos encontrar alumni que atendam às duas condições e depois, generalizando, achar a média desta diferença de tempo (entre a formatura e admissão no cargo), que estará neste seleto grupo de registros.

graduação PUC-Rio	degree=’graduação’ e school_id=10582
Cargo de relevância	Possui (ou possuiu) em sua timeline a palavra gerente, diretor, presidente ou CEO

Tabela 11: Significado dos dois conceitos necessários para a resposta.

Nesta etapa nosso objetivo é amarrar as possibilidades de significado entre o que se digita no LinkedIn e seu correspondente valor de referência que nos interessa, para que a ferramenta possa percorrer estes dados e assim informar um número que será a resposta para a principal pergunta desta dissertação.

Algumas informações já se encontram disponíveis nas tabelas **edus** e **jobs**, como por exemplo as datas de início e fim respectivamente de cada **job** ou **edu**.O campo **título** e seu respectivo *SLUG* **titulo_label** já fornecem uma informação que diz respeito ao cargo exercido, bastando apenas mapear com as profissões que se entendam serem de relevância.

Convencionou-se que cargos de relevância devem corresponder a profissões que envolvam liderar pessoas, portanto devem ser correlacionadas com as palavras: *gerente*, *diretor*, *CEO* e *empresário* e suas respectivas variantes

em outros idiomas ou com grafias erradas/equivalentes, nosso objetivo é filtrar para os alvos previamente determinados (Figura 49).

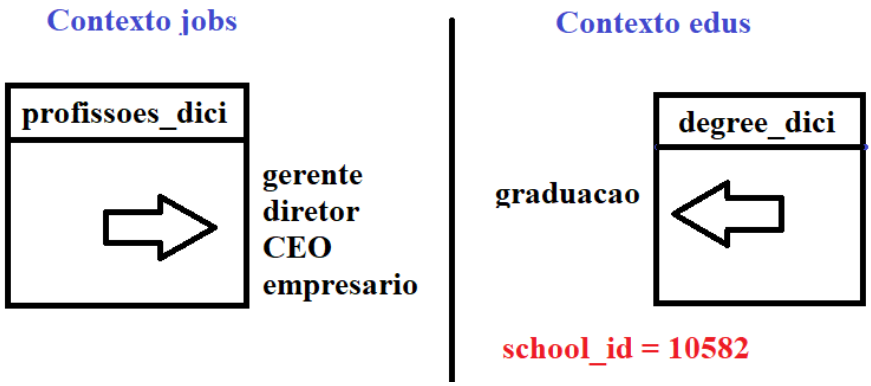


Figura 49: Divisão em dois contextos para se chegar na resposta.

Por exemplo, existem vários sinônimos e grafias diferentes para professor, como *Professor*, *Instrutor*, *Teacher*, *Instructor*, etc. Temos ainda as formas abreviadas, como *PROF*., *Instr*., *prof*. e ainda as formas escritas em diversas lìnguas estrangeiras, além de grafias digitadas erradas como por exemplo *Profsor*. Note que *Prof*. também pode ser interpretado como *Profissional*, o que abre espaço para uma análise mais apurada, sendo necessário até mesmo analisar caso a caso e assim inferir seu correto significado.

“O alumnus pode confundir degree com major”.

No que diz respeito a informação acadêmica, **degree** e **título** não possuem sua correspondência biunívoca no BD, primeiro porque dependemos da interpretação do usuário no momento em que ele cadastra tais informações textuais no LinkedIn, segundo porque muitas vezes o usuário pode se confundir e entender que **major** diz respeito ao tipo de curso, assim como **degree** ser a área de formação. A solução é amarrar estes significados e unificar estas duas grandezas com tabelas de referência através de um **relacionamento N:M**. Para tal, serão criadas seis tabelas onde três delas corresponderão aos valores que vamos mapear e três delas para os valores de associação :

- Profissões (**título**)
- Cursos (**major**)
- Degrees (**degree**)

Cada um destes esquemas N:M irá receber sua tabela intermediária com a terminação **_dici** enquanto que a tabela de valores (escolhidos por um ser humano) terá a terminação **_tab** (Figura 51).

As tabelas com a terminação **_tab** serão populadas de acordo com nossas definições, ou seja aquilo que queremos mapear, nosso alvo. Essas tabelas terão poucos registros, em torno de 10 (Figura 50). Já as tabelas do tipo **_dici** serão as tabelas intermediárias que farão a ligação entre valores do LinkedIn e valores mapeados, essas tabelas serão populadas através do módulo **web**, geralmente representados por um componente HTML do tipo *select* (Figura 52), os valores deste *select* correspondem à coluna correspondente em **_dici** e os labels do select representam as grafias.

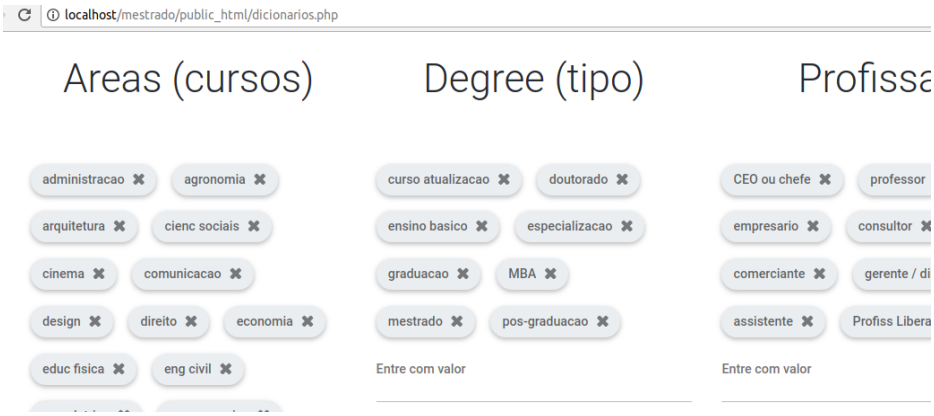


Figura 50: Página onde se define os valores das tabelas _tab a serem mapeados.

Alguns exemplos de mapeamentos adotados: quanto a **profissões**, palavras como *Estagiário*, *estágio*, *intern*, *estagio-em-economia*, serão mapeados para *assistente***,** por exemplo. No caso de degrees palavras como *MSc*, *mestrado*, *Master* estarão na tabela **degree_dici** e terão sua representação por exemplo como *mestrado*.

Ao observarmos a tabela de **jobs** as palavras que desejamos mapear são fruto daquilo que o usuário digita no seu cadastro no LinkedIn e nos apresentam das mais variadas formas, sendo que **titulo_label** é apenas um *SLUG*. O LinkedIn transforma a string digitada pelo usuário, fazendo uso da troca de caracteres como acentuação, espaço e minúsculas para assim transformar por exemplo *Analista de Produção* para *analista-de-producao* assim este segundo termo pode ser usado em uma URL sendo entendido tanto por máquinas quanto

por seres humanos ao contrário da primeira. Vale ressaltar que esta conversão pode gerar resultados controversos pois desconhecemos a regra de negócio usada pelo LinkedIn.

Assim, sucessivamente, faremos com as outras tabelas de tal maneira que ao final do processo teremos um controle sobre o mapeamento realizado entre o universo de palavras que temos no BD em contrapartida ao número reduzido de termos mapeados. Esse raciocínio é válido para as três entidades que recebem as respectivas tabelas associadas (profissões, cursos e degrees).

No caso específico da tabela de profissões, teremos em especial um campo booleano chamado **lider**, que determinará se esta profissão corresponde a um cargo de liderança (ou relevância), também gerenciável pela aplicação. Este flag deverá ser marcado em um tag HTML do tipo *checkbox* manualmente (Figura 53) de acordo com os entendimentos e consenso do que seria realmente um cargo de liderança.

Para mapear a nomenclatura de cursos, degrees e profissões, usamos as páginas do módulo **web**. Temos ainda a possibilidade de definir conforme nossos entendimentos quais seriam os cargos de relevância que para este caso será: *CEO ou chefe*, *empresário* e *gerente/diretor,* ou seja estes registros receberão o valor **1** no campo **lider** da tabela **profissoes_tab** enquanto que outros valores de título que não pertencem ao grupo de relevância receberão o valor **0**.

As associações devem ser feitas conforme entendimentos que se fazem com bom senso, por exemplo *recursos humanos* e *logística* não são cursos comuns em graduação. Neste caso, o mapeamento adotado foi *humanas* e *administração,* ainda *odontologia*, *enfermagem* e *medicina* se encaixam em *medicina*, tal qual *análise de sistemas, banco de dados, Web design e ciência da computação* mapeiam para *informática*.

Referente ao mapeamento, no contexto de **degree** a intenção é segmentar por exemplo em: *graduação, pós-graduação, mestrado, doutorado,* etc. Neste caso, incluiremos *MBA* também por ser muito frequente. Definimos que palavras como *graduação, grad, bacharel, bsc*, etc. farão sua correspondência com *graduação*. No caso também criamos a tag *especialização* para mapear diversos outros cursos que entendemos se tratar de menor significância e que estejam fora do nosso escopo, como por exemplo *curso de data mining, curso de inglês, curso*

de programação Java, etc., e por fim *pós-graduação* que engloba cursos de pós graduação lato sensu.

O que se observou no momento de fazer o mapeamento manual nas três tabelas respectivamente **curso**, **degree** e **profissao** foi uma inclinação do usuário LinkedIn de maquiar seu título profissional. Este título corresponde a essência de sua *timeline*, e a palavra que ele registrar correspondente a sua profissão irá ser a imagem projetada no meio corporativo. O mesmo efeito ocorre também com curso e degree embora não seja tão notável quanto em **título**.

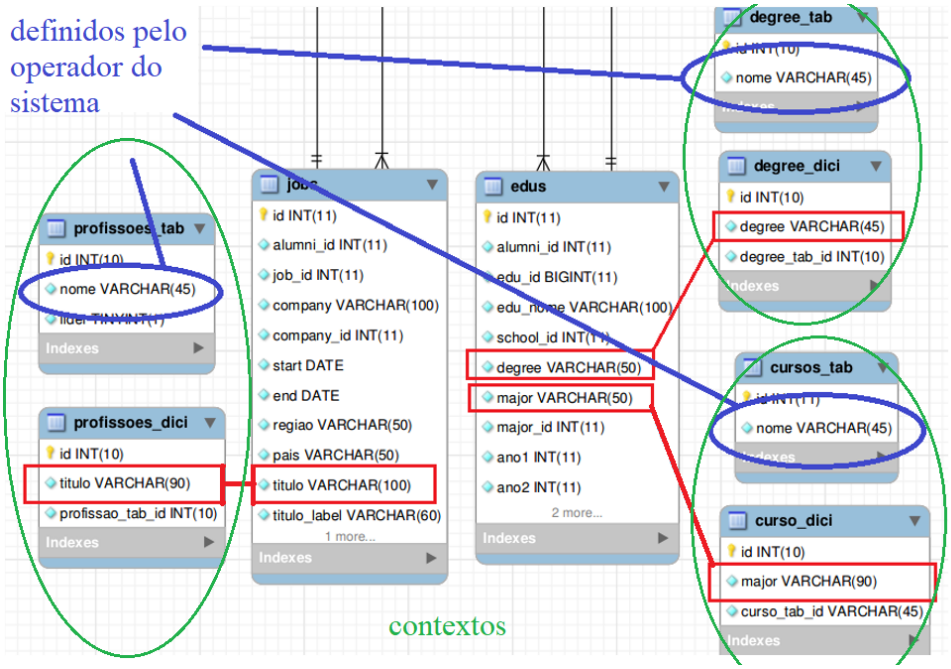


Figura 51: Esquema dos 3 relacionamentos N:M.

Vamos adotar um mapeamento manual, palavra a palavra, feito por um ser humano com discernimento de carreiras e profissões para saber que um *gerente de contas* não é na prática um gerente e está mais para um profissional de gestão, com boa probabilidade de ter cursado administração de empresas. De forma semelhante um *administrador de banco de dados* também não confere um status de gerência; na verdade este título pertence a um analista de banco de dados com formação em informática.

As palavras são então agrupadas e disponibilizadas para que um analista da ferramenta selecione manualmente qual palavra que corresponde aquela grafia (conforme a Figura 52). Essa relação de equivalência fica armazenada no BD e numa atualização dos dados com um novo crawling para todas as palavras

que já possuem seu correspondente significado a análise está garantida, restando para o analista da ferramenta marcar as novas que porventura possam aparecer, a tendência é que a cada novo crawling esse número de novas palavras iria diminuir contribuindo para uma estabilidade das palavras em novas futuras atualizações.

Palavras como *analista* e *consultor* são largamente empregadas e são levadas em consideração, muitos querem majorar seu título. Quando se trata de sócios ou donos de empresas é mais fácil a associação com *CEO* ou simplesmente *empresário*.

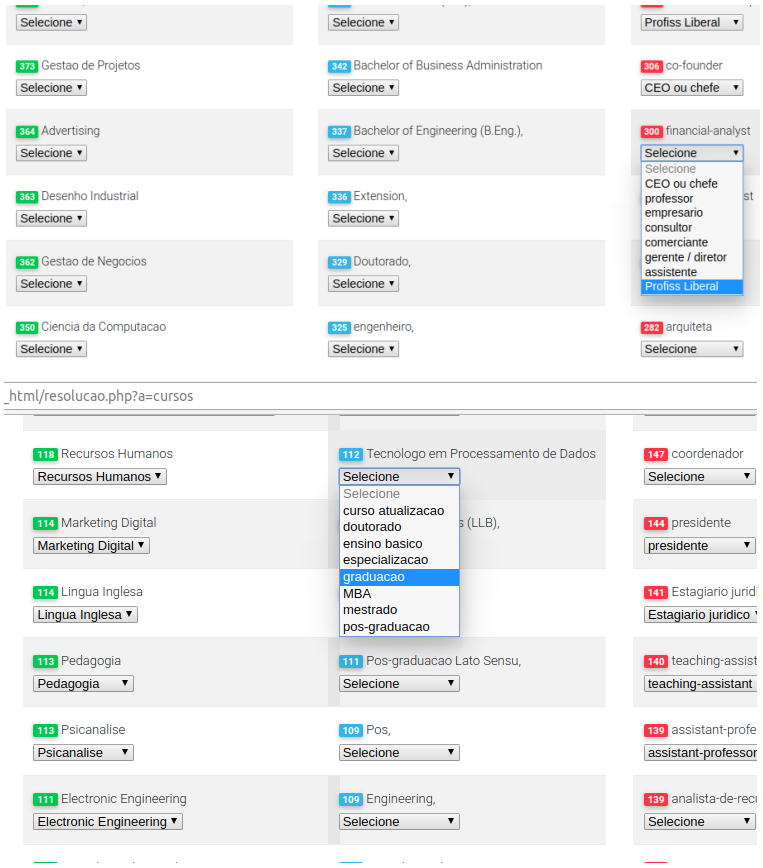


Figura 52: Interface gráfica onde um ser humano define os mapeamentos.

O critério do BD adotado para cargo de liderança foi atribuir a flag **lider** o valor **1** para as profissões *CEO/empresário* e *gerente/diretor*. Foi adotado um critério de analisar se a grafia digitada possui as palavras *gerente* ou *diretor* (ou suas respectivas variações) em seu conteúdo, por exemplo *gerente de marketing*, *diretor de logística*, *financial manager*, etc.

É importante frisar que o título *professor* não fora incluído no hall de cargos de relevância. Na hipótese de se efetivamente adicionar a segunda fonte de dados do Lattes cujo caráter é essencialmente acadêmico o impacto no resultado final iria distorcer o resultado, afinal no Lattes temos uma maioria de *professores*.

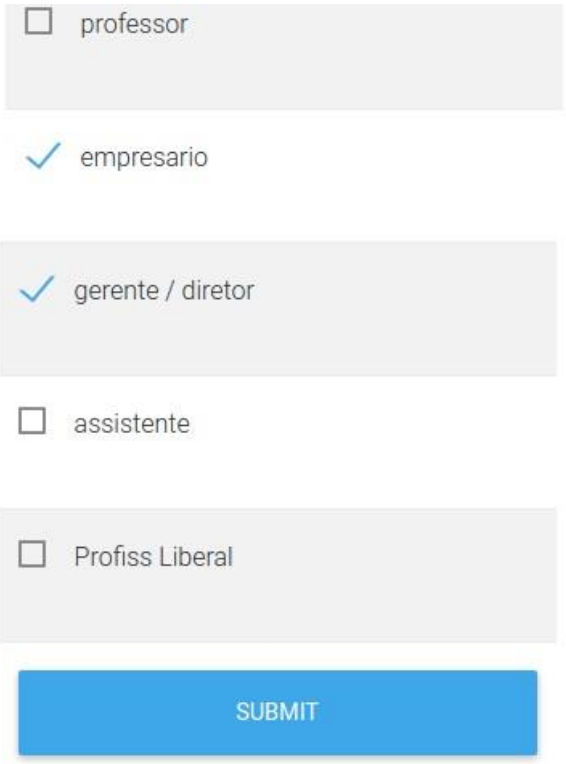


Figura 53: Interface gráfica onde se define o conceito de cargo de relevância.

Como alguns exemplos dos critérios adotados, títulos como *analista* ou *consultor* são mapeados para *consultor*, assim como *engenheiro, médico e advogado* são mapeados para *profissional liberal*, temos ainda *tutor, professor, teacher, intérprete*, etc. para corresponder com *professor* e por fim temos *estagiário, assistente, internship, trainee* que corresponde a *assistente*.

Como se calcular o tempo que se leva para chegar a um cargo de relevância? Por exemplo, analisando um caso isolado (Figura 54), um caso típico de um alumnus com registros em **edus** e **jobs**, como todo registro no BD ele terá um código **alumni_id** comum a ambos.

Inicialmente devemos encontrar pelo menos uma linha para cada indivíduo das tabelas **edus** e **jobs** que atendam os critérios desejados e observando que estes registros devem pertencer ao mesmo **alumni_id**. No caso de haver mais de

um registro que atenda à condição, deve haver um critério de exclusão para se escolher apenas um, neste caso vamos nos ater à primeira admissão e à primeira formatura.

Para este caso em específico (Figura 54) o resultado foi 1994-1987=7 anos. Podemos nos guiar pelo seguinte raciocínio:

1) anoA:Selecione o menor *ano2* de **edus** onde **school_id**=*10582* e **degree**=*'graduacao'*.

2) anoB:Selecione o menor *ano1 (campo start)* de **jobs** onde **título** equivale a *'gerente/diretor'* (campo referente à profissão cuja flag **lider** seja **1**).

3) Compute a diferença do **anoB – anoA**.

4) Caso este valor seja negativo adote **0** como resultado.

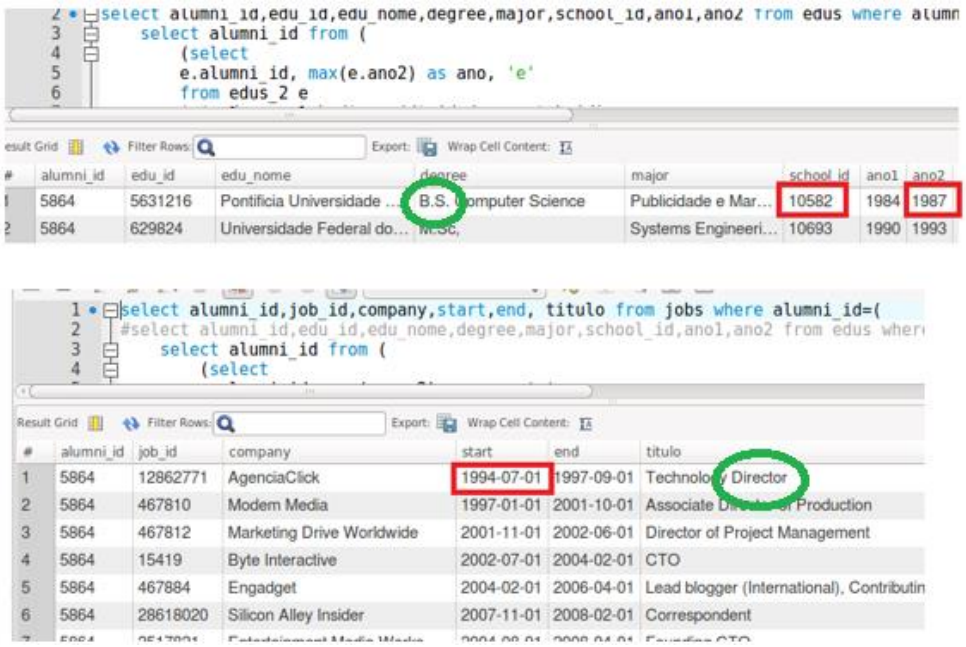


Figura 54: Exemplo de jobs e edus que atendam os critérios procurados.

O ponto crucial desta etapa seria encontrar a média deste número, considerando todos os registros, que tais como o do exemplo acima tenha feito **degree**=*'graduacao' na PUC-Rio* (school_id=10582) cuja flag **lider** seja 1.

Entretanto um indivíduo poderia ter uma outra graduação antes da PUC-Rio, poderia estar em um cargo de gerência antes da graduação da PUC-Rio, poderia assumir um posto de gerência durante a faculdade, este resultado poderia ser inclusive negativo e isso impactaria no cálculo da média. Aqueles que

atingirem o cargo de relevância em que este ano seja inferior ao ano de formatura, considerar esta diferença como **zero**.

Em uma primeira consulta SQL fazemos as devidas filtragens de acordo com nosso objetivo. Vamos usar agora os dados que foram normalizados para as tabelas **jobs_2** e **edus_2** (Figura 55), tabelas estas que possuem somente campos numéricos, aumentando sensivelmente a performance da consulta.

Os campos **degree_tab_id**, **curso_tab_id** em **edus_2** e **profissao_tab_id** em **jobs_2** abstraem todo o universo de possibilidades e se correlacionam com os valores de nosso interesse. Concluímos o mapeamento de strings restando agora analisar qual instrução T-SQL responderá nossa pergunta.

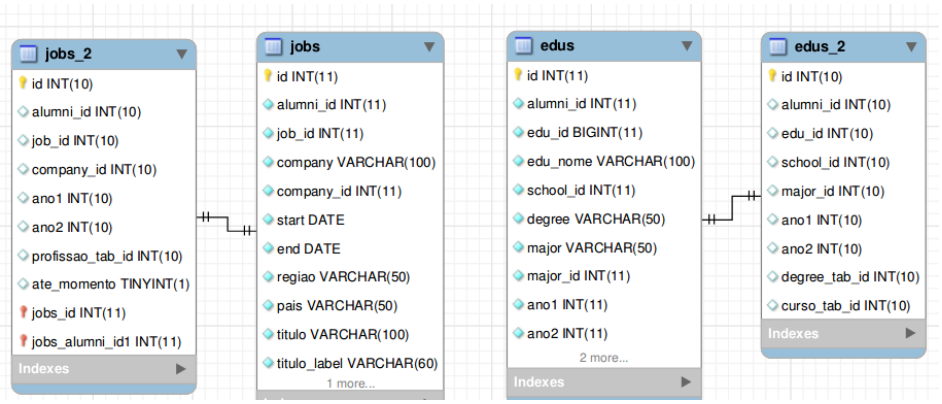


Figura 55: Tabelas edus_2 e jobs_2.

Para graduação, o valor correspondente em **edus_2** que responderá parte de nossa pergunta será **degree_tab_id**=*2*; para a PUC-Rio temos o nosso número já conhecido *10582*. Filtrando tudo para anos superiores a 1971 (Figura 56) evitamos de incluir em nossa consulta erros de usuários como por exemplo quem erra a data na hora de digitar (no MySQL esses erros de data são representados pela data 01/01/1900).

Já para **jobs_2** devemos encontrar os registros cuja flag **lider** seja **1**, indicando tratar-se de um cargo de relevância e tal qual em **edus_2** e filtramos todos os **ano1** acima de 1971.

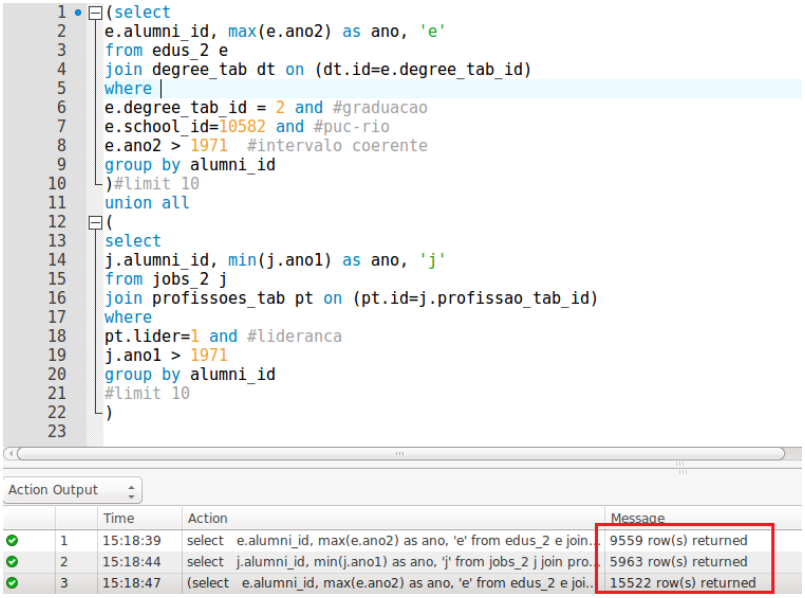


Figura 56: 9559 registros graduação PUC-Rio e 5963 registros liderança.

Os critérios que foram adotados até agora para definir os termos que designam cargo de relevância podem ser alterados a qualquer momento. A inteligência do sistema deve estar atrelada à definição do que vem a ser um cargo de relevância. Ou seja, podemos mudar essa regra de negócio a qualquer momento. Se a partir de agora definirmos que um cargo de relevância também engloba *profissional liberal* por exemplo basta marcar o *checkbox* correspondente (Figura 53) e acessar a SQL novamente no BD.

Resposta: 9,3 anos (Figura 57).

Ressaltamos aqui que todos estes valores são considerados exclusivamente sobre a base de dados informada no LinkedIn e referente à *snapshot* em questão, não havendo nenhuma validação concreta destes dados. O próximo capítulo tece algumas conclusões e menciona algumas sugestões para se aferir a qualidade dos dados.

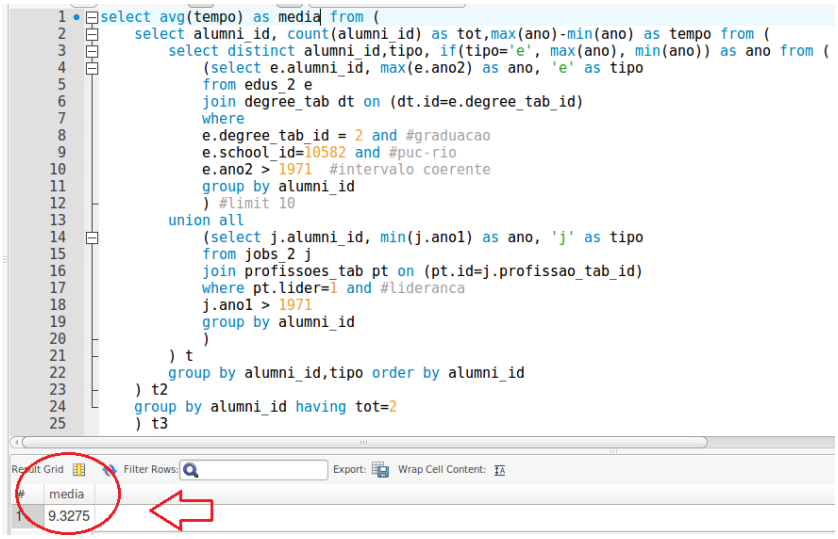


Figura 57: Resultado obtido após análise aprofundada.

6.
Conclusão

6.1
Contribuições

Neste trabalho discutimos diversas possibilidades de RI voltadas para a análise de alumni. Para analisar e minerar o comportamento e evolução destes alumni exclusivos da PUC-Rio exploramos diversas possibilidades sobre a única fonte de dados adotada, porém deixando em aberto a possibilidade de se incrementar o DW adicionando-se outras fontes de dados e analisando o impacto desta fonte de dados no resultado final. A inovação da ferramenta desenvolvida está na combinação de um programa de teste funcional com um robô para efetuar a captura de dados. O desempenho da ferramenta se mostrou muito eficaz atingindo a marca de 88% de hashes e 66% de alumni. Com um tempo de processamento de cerca de 3 meses para recuperação da informação, considerando:

- 30 dias para a etapa de hashes.
- 60 dias para a etapa de perfis.
- 5 dias para a etapa de scraping.
- 1 dia para as definições e sincronização dos mapeamentos.

Referente a coleta de dados específicas de alumni a grande maioria dos trabalhos presentes na literatura aborda um aspecto manual e sobretudo com um número muito baixo de registros, conforme sintetizado na Tabela 12.

Referência	Método Usado	Nº de registros
ARPPA [15]	API Search	1500
XU [21]	InMail	Não informa
CETINTAS et al [19]	API	2200
PENA [31]	CSE	357
Univille [74]	Não informa	Retorno pequeno
FEBE [74]	Entrevistas	Não informa
IMBRIZI [37]	Manual	94
CAMARGO [8]	Questionários (link)	800
LOUSADA&MARTINS [74]	Entrevistas	Não informa
Usp [114]	Lattes	185

Tabela 12: Comparativo com alguns trabalhos no aspecto de coleta de alumni.

Somados os diversos fatores acima temos como contribuição da dissertação a possibilidade de se efetuar um crawling em qualquer Website que faça autenticação do usuário, baseado no mecanismo de manutenção da sessão ativa do navegador, usando uma solução baseada em Selenium.

6.2
Trabalhos Futuros

Um aspecto relevante para futuros trabalhos que venham a usar os recursos aqui explicados seria analisar a qualidade dos dados oriundos da fonte LinkedIn. Para se chegar a este valor de 9,3 anos, analisamos apenas uma base de dados, ou seja, a resposta para a pergunta está estritamente vinculada à qualidade dos dados. Eis então que surge outra pergunta: O quão são confiáveis os dados oriundos do LinkedIn?

Um raciocínio válido para se averiguar a qualidade destes dados seria, por exemplo, se X informa que trabalhou por 10 anos na IBM, de 2005 a 2015, e foi *programador*, apenas a própria IBM poderá realmente garantir esta informação, assim como se ele informa que estudou cinema na PUC-Rio de 2003 a 2007, apenas a PUC-Rio pode validar este dado. O que se informa no LinkedIn pode ser inclusive inventado.

Ao analisar as informações complementares ao perfil do usuário no LinkedIn, percebemos que temos diversas outras informações tais como **skills** e

recomendações, que são sempre endossadas por outras pessoas. Qualquer usuário pode registrar até de maneira arbitrária que trabalhou em A, B, C e estudou em X, Y, Z, pode inclusive registrar um **skill** em PHP e HTML, por exemplo, mas até então ela estará não endossada, vazia, esperando o endosso de outras pessoas. Se o perfil não for *fake*, o endosso teoricamente não pode ser forjado, deve ser feito por outras pessoas, que supostamente tenham conhecido o trabalho feito por tal pessoa, o mesmo para recomendações. Essa medida pode ser usada como uma estimativa do quanto este perfil é fidedigno à realidade.

Perfis *fake* no LinkedIn seguem a mesma tratativa que perfis de pessoas falecidas, ou seja é esperado que a própria comunidade aponte estes perfis e assim abre-se um processo interno que culmina na exclusão (Figura 58).

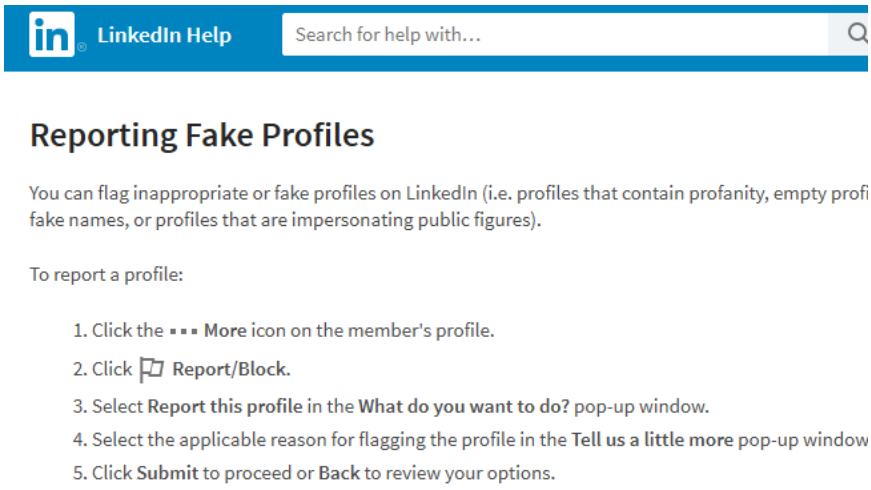


Figura 58: Política do LinkedIn para combater perfis fake.

Outro aspecto é que a própria universidade mantém alguns registros de ex-alunos em suas mais variadas formas que vão desde planilhas *Excel*, formulários em papel e fichas armazenadas em arquivos físicos. Estas informações poderiam ser compiladas para dar algum suporte à veracidade destas conclusões. Também poderiam ser usadas na elaboração de uma lista de nomes, conforme discutido no capítulo 4.

Um último ponto seria entender melhor o processo de "lavagem de diploma", que consiste em fazer uma graduação em uma Universidade de pouco ou nenhum renome nacional e, após concluir tal graduação, fazer a posteriori um curso de especialização lato sensu em uma Universidade de ponta, e assim

destacar no currículo tal especialização, dando a entender que sua formação acadêmica se consagrou na Universidade de ponta.

7.
Referências bibliográficas

[1] DE FARIAS, Lucas R. et al. Um sistema para análise de redes de pesquisa baseado na Plataforma Lattes. **Escola Regional de Banco de Dados**, 2012.

[2] JUNIOR, Walter Teixeira Lima. Mídia social conectada: produção colaborativa de informação de relevância social em ambiente tecnológico digital. **LÍBERO. ISSN impresso: 1517-3283/ISSN online: 2525-3166**, n. 24, p. 95-106, 2016.

[3] BALANCIERI, Renato et al. A análise de redes de colaboração científica sob as novas tecnologias de informação e comunicação: um estudo na Plataforma Lattes. **Ciência da informação**, v. 34, n. 1, p. 64-77, 2005.

[4] ALA-MUTKA, Kirsti. Social computing: Study on the use and impacts of collaborative content. In: **JRC Scientific and Technical Reports EUR 23572 EN). Seville: European Commission-Joint Research Centre-Institute for Prospective Technological Studies. Available from EURdoc/JRC47511. pdf**. 2008.

[5] BOVO, Alessandro Botelho et al. Um método de tradução de fontes de informação em um formato padrão que viabilize a extração de conhecimento por meio de link analysis e teoria dos grafos. **UFSC**, 2004.

[6] BUENO, U. ZWICKER, OLIVEIRA, M. A. de. Um estudo comparativo do modelo de aceitação de tecnologia aplicado em sistemas de informações e comércio eletrônico. In: **Congresso internacional de gestão de tecnologia e sistemas de informação**, v. 1., 2004. São Paulo, SP. Anais. São Paulo: [s.n], 2004.

[7] CHAIM, Daniel Faria; MARTINELLI, Camila Rocha; DE AZEVEDO, Marília Macorin. Redes Sociais on-line e seleção de pessoas: LinkedIn e SERVQUAL/Social Networking online and selecting people: LinkedIn and SERVQUAL. **Revista de Tecnologia Aplicada**, v. 1, n. 3, p. 30-42, 2012.

[8] CAMARGO, Lilian Cristina Carvalho et al. ESTUDO DA ADOÇÃO INDIVIDUAL DA REDE SOCIAL PROFISSIONAL LINKEDIN. **UNIMEP** , 2015.

[9] LinkedIn cresce 25% em base de usuários no Brasil. <https://exame.abril.com.br/tecnologia/linkedin-cresce-25-em-base-de-usuarios-no-brasil/> **Revista Exame**, 2016, acesso em: 21/02/2018.

[10] LinkedIn tem crescimento de 25% em base de usuários no Brasil. <http://g1.globo.com/tecnologia/noticia/2016/05/linkedin-tem-crescimento-de-25-em-base-de-usuarios-no-brasil-20160505175013345050.html> **Rede Globo**, acesso em: 21/02/2018.

[11] DOS SANTOS, Leonardo Bres; DORNELES, Carina F.; DOS SANTOS MELLO, Ronaldo. Uma Abordagem para Detecção e Extração de Rótulos em Formulários Web. In: **SBBD (Short Papers)**. 2012. p. 233-239.

[12] RAGHAVAN, Sriram; GARCIA-MOLINA, Hector. Crawling the hidden web. **Paper, Stanford**, <http://ilpubs.stanford.edu:8090/456/1/2000-36.pdf> Acesso em: 25/02/2018. 2000.

[13] ÁLVAREZ, Manuel et al. Crawling the content hidden behind web forms. In: **International Conference on Computational Science and Its Applications**. Springer, Berlin, Heidelberg, 2007. p. 322-333.

[14] Grego, M. LinkedIN atinge 15 milhões de usuários no Brasil. **Revista EXAME.** Disponível em: <http://exame.abril.com.br/tecnologia/noticias/LinkedIN-atinge-15-milhoes-de-usuarios-no-brasil-2>, 2013. Acesso em: 25/02/2018.

[15] SILVA, Paula RC; BRANDÃO, Wladmir C. ARPPA: Mining Professional Profiles from LinkedIn Using Association Rules. **eKNOW 2015: The Seventh International Conference on Information, Process, and Knowledge Management.** 2015.

[16] MADHAVAN, Jayant et al. Harnessing the deep web: Present and future. **ArXiv preprint arXiv:0909.1785**, 2009.

[17] LOPS, Pasquale et al. Leveraging the linkedin social network data for extracting content-based user profiles. In: **Proceedings of the fifth ACM conference on Recommender systems**. ACM, 2011. p. 293-296.

[18] MYLLYMAKI, Jussi. Effective web data extraction with standard XML technologies. **Computer Networks**, v. 39, n. 5, p. 635-644, 2002.

[19] CETINTAS, Suleyman et al. Identifying similar people in professional social networks with discriminative probabilistic models. In: **Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval**. ACM, 2011. p. 1209-1210.

[20] DE PAIVA MENEZES, Thiago et al. Análise do perfil de aluno e egresso de cursos técnicos por meio de data mining: estudo de caso no Instituto Federal Fluminense. **#Tear: Revista de Educação, Ciência e Tecnologia**, v. 3, n. 1, 2014.

[21] XU, Ye et al. Modeling professional similarity by mining professional career trajectories. In: **Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining**. ACM, 2014. p. 1945-1954.

[22] DELAFONTAINE, Julien; PRADERVAND, Sylvain. A RESTful API to serve BAM file with OAuth2 compatible authorization. **BioRxiv**, p. 151787, 2017.

[23] LIU, Wei; MENG, Xiaofeng; MENG, Weiyi. Vide: A vision-based approach for deep web data extraction. **IEEE Transactions on Knowledge and Data Engineering**, v. 22, n. 3, p. 447-460, 2010.

[24] BUTTLER, David; LIU, Ling; PU, Calton. A fully automated object extraction system for the World Wide Web. In: **Distributed Computing Systems, 2001. 21st International Conference on.** IEEE, 2001. p. 361-370.

[25] CHANG, Chia-Hui; HSU, Chun-Nan; LUI, Shao-Cheng. Automatic information extraction from semi-structured web pages by pattern discovery. **Decision Support Systems**, v. 35, n. 1, p. 129-147, 2003.

[26] CRESCENZI, Valter et al. Roadrunner: Towards automatic data extraction from large web sites. In: **VLDB**. 2001. p. 109-118.

[27] NOUREDDINE, M.; BASHROUSH, R. A provisioning model towards Oauth 2.0 performance optimization. In: **Cybernetic Intelligent Systems (CIS), 2011 IEEE 10th International Conference on**. IEEE, 2011. p. 76-80.

[28] CHANG, Chia-Hui; HSU, Chun-Nan; LUI, Shao-Cheng. Automatic information extraction from semi-structured web pages by pattern discovery. **Decision Support Systems**, v. 35, n. 1, p. 129-147, 2003.

[29] GONÇALVES, Gabriel Resende et al. Gathering alumni information from a web social network. In: **Web Congress (LA-WEB), 2014 9th Latin American**. IEEE, 2014. p. 100-108.

[30] ALLAUDDIN, Maria; AZAM, Farooque. Service crawling using google custom search API. **International Journal of Computer Applications**, v. 34, n. 7, p. 2011, 2011.

[31] PENA, M. D. Alumni monitoring: Conceptual analysis and its application in Brazilian educational context**. Technological Education of Belo Horizonte**, vol. 5, no. 2, pp. 25-30, 2000.

PENA, M. D. Alumni monitoring: Conceptual analysis and its application in Brazilian educational context (in Portuguese). **Technological Education of Belo Horizonte**, vol. 5, no. 2, pp. 25-30, 2000.

[32] MANGARAVITE, Vitor; ASSIS, Guilherme Tavares_de_; FERREIRA, Anderson A. Improving the efficiency of a genre-aware approach to focused crawling based on link context. In: **Web Congress (LA-WEB), 2012 Eighth Latin American**. IEEE, 2012. p. 17-23.

[33] FERRARA, Emilio et al. Web data extraction, applications and techniques: A survey. **Knowledge-based systems**, v. 70, p. 301-323, 2014.

[34] GJOKA, Minas et al. Walking in facebook: A case study of unbiased sampling of osns. In: **Infocom, 2010 Proceedings IEEE**. IEEE, 2010. p. 1-9.

[35] DE ASSIS, Guilherme T. et al. A genre-aware approach to focused crawling. **World Wide Web**, v. 12, n. 3, p. 285-319, 2009.

[36] PANT, Gautam; SRINIVASAN, Padmini. Learning to crawl: Comparing classification schemes. **ACM Transactions on Information Systems (TOIS)**, v. 23, n. 4, p. 430-462, 2005.

[37] IMBRIZI, J. L.; FILFO, F. G. Pesquisa aos egressos**.** <http://www.dpi.ufv.br/arquivos/diversos/pesqegressos.pdf> **Universidade Federal de Viçosa**, 2003. Acesso em: 20/02/2018.

[38] BENEVENUTO, Fabrício. Redes sociais online: Técnicas de coleta, abordagens de medição e desafios futuros. **Tópicos em Sistemas**

Colaborativos, Interativos, Multimidia, Web e Banco de Dados, p. 41-70, 2010.

[39] SEMERARO, Giovanni et al. Combining Learning and Word Sense Disambiguation for Intelligent User Profiling. In: **IJCAI**. 2007. p. 2856-2861.

[40] GORI, Marco; PUCCI, Augusto. Research paper recommender systems: A random-walk based approach. In: **Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on**. IEEE, 2006. p. 778-781.

[41] CHAU, Duen Horng et al. Parallel crawling for online social networks. In: **Proceedings of the 16th international conference on World Wide Web**. ACM, 2007. p. 1283-1284.

[42] CHO, Junghoo; GARCIA-MOLINA, Hector. Parallel crawlers. In: **Proceedings of the 11th international conference on World Wide Web**. ACM, 2002. p. 124-135.

[43] HEER, Jeffrey; BOYD, Danah. Vizster: Visualizing online social networks. In: **Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on**. IEEE, 2005. p. 32-39.

[44] HOROWITZ, Damon; KAMVAR, Sepandar D. The anatomy of a large-scale social search engine. In: **Proceedings of the 19th international conference on World Wide Web**. ACM, 2010. p. 431-440.

[45] BRIN, Sergey; PAGE, Lawrence. The anatomy of a large-scale hypertextual web search engine. **Computer networks and ISDN systems**, v. 30, n. 1-7, p. 107-117, 1998.

[46] MILLER, Robert C.; BHARAT, Krishna. SPHINX: a framework for creating personal, site-specific Web crawlers. **Computer Networks and ISDN systems**, v. 30, n. 1-7, p. 119-130, 1998.

[47] HEYDON, Allan; NAJORK, Marc. Mercator: A scalable, extensible web crawler. **World Wide Web**, v. 2, n. 4, p. 219-229, 1999.

[48] DILIGENTI, Michelangelo et al. Focused Crawling Using Context Graphs. In: **VLDB**. 2000. p. 527-534.

[49] CHO, Junghoo; GARCIA-MOLINA, Hector; PAGE, Lawrence. Efficient crawling through URL ordering. **Computer Networks and ISDN Systems**, v. 30, n. 1-7, p. 161-172, 1998.

[50] ADELBERG, Brad. NoDoSE - a tool for semi-automatically extracting structured and semistructured data from text documents. In: **ACM Sigmod Record**. ACM, 1998. p. 283-294.

[51] ARASU, Arvind; GARCIA-MOLINA, Hector. Extracting structured data from web pages. In: **Proceedings of the 2003 ACM SIGMOD international conference on Management of data**. ACM, 2003. p. 337-348.

[52] LAENDER, Alberto HF et al. A brief survey of web data extraction tools. **ACM Sigmod Record**, v. 31, n. 2, p. 84-93, 2002.

[53] SHARDANAND, Upendra; MAES, Pattie. Social information filtering: algorithms for automating “word of mouth”. In: **Proceedings of the SIGCHI conference on Human factors in computing systems**. ACM Press/Addison-Wesley Publishing Co., 1995. p. 210-217.

[54] LERMAN, Kristina. Social networks and social information filtering on digg. **ArXiv preprint cs/0612046**, 2006.

[55] TEIXEIRA, Cenidalva Miranda de Sousa; SCHIEL, Ulrich. A Internet e seu impacto nos processos de recuperação da informação. **Ciência da Informação**, v. 26, n. 1, 1997.

[56] SOUZA, Renato Rocha et al. Sistemas de recuperação de informações e mecanismos de busca na web: panorama atual e tendências. **Perspectivas em ciência da informação**, v. 11, n. 2, p. 161-173, 2006.

[57] DE FREITAS, Frederico LG et al. Sistemas Multiagentes Cognitivos para recuperação, classificação e extração integradas de informação da WEB. Tese UFSC, 2002.

[58] LOPES, Ilza Leite. Estratégia de busca na recuperação da informação: revisão da literatura. **Ci. Inf.**, Brasília, v. 31, n. 2, p. 60-71, maio/ago. 2002.

[59] ATAY, Mustafa et al. Efficient schema-based XML-to-Relational data mapping. **Information Systems**, v. 32, n. 3, p. 458-476, 2007.

[60] GREENE, T. LinkedIN loses legal right to protect user data from AI scraping. <https://thenextWeb.com/artificial-intelligence/2017/08/15/LinkedIN-loses-legal-right-to-protect-user-data-from-ai-scraping/>, 2017. Acesso em: 20/02/2018.

[61] FRANKEL, A. Data scraper's case v. LinkedIN pits free speech against CFAA, DMCA. <https://www.reuters.com/article/us-otc-LinkedIN/data-scrapers-case-v-LinkedIN-pits-free-speech-against-cfaa-dmca-idUSKBN19B2WE>. Acesso em: 20/02/2018.

[62] GOLDFEIN, S.; KEYTE, J. Big Data, Web Scraping and Competition Law: The Debate Continues. <http://www.newyorklawjournal.com/id=1202797578445/Big-Data-Web-Scraping-and-Competition-Law-The-Debate-Continues?slreturn=20170911194911>, **New York Law Journal, 2017**. Acesso em: 20/02/2018.

[63] LinkedIN cant block start-up from public profile data. <https://www.itnews.com.au/news/LinkedIN-cant-block-start-up-from-public-profile-data-470886>, Acesso em: 20/02/2018.

[64] YANG, Ronghai; LAU, Wing Cheong; LIU, Tianyu. Signing into one billion mobile app accounts effortlessly with oauth2. 0. **blackhat Europe**, 2016.

[65] CHEN, Hsinchun; CHIANG, Roger HL; STOREY, Veda C. Business intelligence and analytics: from big data to big impact. **MIS quarterly**, p. 1165-1188, 2012.

[66] FREITAS, C. M. D. S. et al. Extração de conhecimento e análise visual de redes sociais. **SEMISH-Seminário Integrado de Software e Hardware, Belém do Pará, Brasil, SBC**, p. 106-120, 2008.

[67] WIDOM, Jennifer. Research problems in data warehousing. In: **Proceedings of the fourth international conference on Information and knowledge management**. ACM, 1995. p. 25-30.

[68] CORLIS, Rebecca. LinkedIn 277% More Effective for Lead Generation than Facebook & Twitter [New Data]. **HubSpot Blog, January**, v. 30, 2012.

[69] OGURI P. et. al. **Aprendizado de Máquina para o Problema de Sentiment Classification**. 2006. Tese de Doutorado. PUC-Rio.

[70] AGRAWAL, Rakesh et al. Fast algorithms for mining association rules. In: **Proc. 20th int. conf. very large data bases, VLDB**. 1994. p. 487-499.

[71] DEY, Farouk; CRUZVERGARA, Christine Y. Evolution of career services in higher education. **New Directions for Student Services**, v. 2014, n. 148, p. 5-18, 2014.

[72] MELCHIORI, Gerlinda S. Alumni research: An introduction. **New Directions for Institutional Research**, v. 1988, n. 60, p. 5-11, 1988.

[73] WOLFE, Kristen Elaine. **Understanding the careers of the alumni of the MIT Mechanical Engineering Department**. 2004. Tese de Doutorado. Massachusetts Institute of Technology.

[74] LOUSADA, Ana Cristina Zenha; MARTINS, Gilberto de Andadre. Egressos como fonte de informação à gestão dos cursos de Ciências Contábeis. **Revista Contabilidade & Finanças**, v. 16, n. 37, p. 73-84, 2005.

[75] ALMEIDA, Letícia Laurino; MERLO, Álvaro Roberto Crespo. Manda quem pode, obedece quem tem juízo: prazer e sofrimento psíquico em cargos de gerência. **Cadernos de Psicologia Social do Trabalho**, v. 11, n. 2, p. 139-157, 2008.

[76] THUSOO, Ashish et al. Data warehousing and analytics infrastructure at facebook. In: **Proceedings of the 2010 ACM SIGMOD International Conference on Management of data**. ACM, 2010. p. 1013-1020.

[77] NEMATI, Hamid R. et al. Knowledge warehouse: an architectural integration of knowledge management, decision support, artificial intelligence and data warehousing. **Decision Support Systems**, v. 33, n. 2, p. 143-161, 2002.

[78] HAN, Jiawei; PEI, Jian; KAMBER, Micheline. **Data mining: concepts and techniques**. Elsevier, 2011.

[79] HUANG, Zhexue. A fast clustering algorithm to cluster very large categorical data sets in data mining. **DMKD**, v. 3, n. 8, p. 34-39, 1997.

[80] ROMERO, Cristobal; VENTURA, Sebastian. Educational data mining: A survey from 1995 to 2005. **Expert systems with applications**, v. 33, n. 1, p. 135-146, 2007.

[81] CHEN, Ming-Syan; HAN, Jiawei; YU, Philip S. Data mining: an overview from a database perspective. **IEEE Transactions on Knowledge and data Engineering**, v. 8, n. 6, p. 866-883, 1996.

[82] HARALSON, Dmitriy. Automating website crawling using web scraping techniques provided by PHP. Disponível em: < https://www.theseus.fi/bitstream/handle/10024/111624/ThesisDmitriyHaralson.pdf >, Acesso em 25/02/2018, Tese, 2016.

[83] BRUNS, Andreas; KORNSTADT, Andreas; WICHMANN, Dennis. Web application tests with selenium. **IEEE software**, v. 26, n. 5, 2009.

[84] SINGH, Jagdish; SHARMA, Monika. Performance Evaluation and Comparison of Sahi Pro and Selenium Webdriver. **International Journal of Computer Applications**, v. 129, n. 8, p. 23-26, 2015.

[85] BINDAL, Purnima; GUPTA, Sonika. Test Automation Selenium WebDriver using TestNG. **Journal of Engineering Computers & Applied Sciences (JECAS) ISSN No**, v. 2319, p. 5606, 2012.

[86] LIMA, Gustavo Simão; NETO, Antonio Carvalho; TANURE, Betania. Executivos jovens e seniores no topo da carreira: conflitos e complementaridades. **Revista Eletrônica de Administração**, v. 18, n. 1, p. 63-96, 2012.

[87] BRODER, Andrei et al. Graph structure in the web. **Computer networks**, v. 33, n. 1-6, p. 309-320, 2000.

[88] LUZ, S.; GIANINI, T. O caminho para o topo na carreira. <https://exame.abril.com.br/revista-exame/o-caminho-para-o-topo-m0131320/>. **Revista Exame**, Acesso em: 20/02/2018.

[89] Bretas, V. 7 executivos que chegaram ao topo antes dos 40.

<https://exame.abril.com.br/revista-exame/7-executivos-que-chegaram-ao-topo-antes-dos-40/>. **Revista Exame**, Acesso em: 20/02/2018.

[90] WATTENBERG, Martin. Visual exploration of multivariate graphs. In: **Proceedings of the SIGCHI conference on Human Factors in computing systems**. ACM, 2006. p. 811-819.

[91] SOUSA, Eda CB. Machado de. Avaliação Institucional: uma abordagem prática. **IN. SOUSA, Eda CB Machado de (org.). Avaliação Institucional**, v. 6, p. 2, 1996.

[92] BOWER, David F. Six Degrees: The Science of a Connected Age. **Complicity: An International Journal of Complexity and Education**, v. 2, n. 1, 2005.

[93] TELLES, A. A revolução das mídias sociais: cases, conceitos, dicas e ferramentas. São Paulo: M.Books do Brasil Editora Ltda, 2010.

[94] HAUGSET, Borge; HANSSEN, Geir Kjetil. Automated acceptance testing: A literature review and an industrial case study. In: **Agile, 2008. AGILE'08. Conference**. IEEE, 2008. p. 27-38.

[95] JARGAS, Aurelio Marinho. Shell Script Profissional. **Novatec Editora**, 2008.

[96] SCHRENK, Michael. Webbots, spiders, and screen scrapers: A guide to developing Internet agents with PHP/CURL. **No Starch Press**, 2012.

[97] ANDERSON, Shay Michael. Instant Simple Botting with PHP. **Packt Publishing Ltd**, 2013.

[98] RUSSELL, Matthew A. Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More. **OReilly Media, Inc.**, 2013.

[99] AFONSO, A. S. **Uma análise da utilização das redes sociais em ambientes corporativos. 2009.170 f**. 2009. Tese de Doutorado. Dissertação (Mestrado em Tecnologia da Inteligência e Design). Pontifica Universidade Católica de São Paulo. São Paulo.

[100] BASTOS, Virgílio B. et al. Redes sociais informais e compartilhamento de significados sobre mudança organizacional. **Revista de Administração de Empresas**, v. 47, n. 3, p. 1-13, 2007.

[101] CASTELLS, Manuel; MAJER, Roneide Venâncio; GERHARDT, Klauss Brandini. **A sociedade em rede**. Fundação Calouste Gulbenkian, 2002.

[102] TRAVERS, Jeffrey; MILGRAM, Stanley. The small world problem. **Phychology Today**, v. 1, n. 1, p. 61-67, 1967.

[103] BOTH, Ivo José. Avaliar a universidade é preciso: agente de modernização administrativa e da educação. **SOUZA, Eda CB Machado (org). Avaliação Institucional**, v. 2, 1999.

[104] SHIMAZAKI, Vinicius Kenji; PINTO, Maria Márcia Matos. A influência das redes sociais na rotina dos seres humanos. **FaSCi-Tech**, v. 1, n. 5, 2016.

[105] TOMAÉL, Maria Inês; ALCARÁ, Adriana Rosecler; DI CHIARA, Ivone Guerreiro. Das redes sociais à inovação. **Ciência da informação**, v. 34, n. 2, p. 93-104, 2005.

[106] AROCENA, Gustavo O.; MENDELZON, Alberto O. WebOQL: Restructuring documents, databases and Webs. In: **Data Engineering, 1998. Proceedings, 14th International Conference on**. IEEE, 1998. p. 24-33.

[107] HAMMER, Joachim; MCHUGH, Jason; GARCIA-MOLINA, Hector. Semistructured Data: **The TSIMMIS Experience**. 1997.

[108] BERNERS-LEE, Tim et al. World-wide web: The information universe. **Internet Research**, v. 20, n. 4, p. 461-471, 2010.

[109] SHADBOLT, Nigel; BERNERS-LEE, Tim; HALL, Wendy. The semantic web revisited. **IEEE intelligent systems**, v. 21, n. 3, p. 96-101, 2006.

[110] ELLISON, Nicole B. et al. Social network sites: Definition, history, and scholarship. **Journal of computer-mediated Communication**, v. 13, n. 1, p. 210-230, 2007.

[111] ERDŐS, Paul; RÉNYI, Alfréd. On the evolution of random graphs. **Publ. Math. Inst. Hung. Acad. Sci**, v. 5, n. 1, p. 17-60, 1960.

[112] FAYYD, Usama M.; SHAPIRO, Gregory P.; SMYTH, Padhraic. From data mining to knowledge discovery: An overview. Menlo Park: AAAI Press. 611p. p.11-34, 1996.

[113] JOHN, George H. **Enhancements to the data mining process**. 1997. Tese de Doutorado. Stanford University.

[114] BEUREN, Ilse Maria et al. Redes de pesquisa entre os egressos do Doutorado em Ciências Contábeis da FEA/USP. **Contabilidade, Gestão e Governança**, v. 12, n. 3, 2010.

[115] SILVEIRA, S.; 67% dos usuários brasileiros do LinkedIn têm diploma universitário. Folha de São Paulo. 2014. Disponível em: <http://www1.folha.uol.com.br/tec/2014/04/1442291-67-dos-usuarios-brasileiros-do-linkedin-tem-diploma-universitario.shtml> Acesso em 25/02/2018.