

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

EDUARDO CORRÊA BOTH

**Ferramenta de Avaliação e Comparação da
Produção Acadêmica de Pesquisadores e
Grupos de Pesquisa**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em Ciência
da Computação

Orientador: Prof. Dr. Leandro Krug Wives

Porto Alegre
2017

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitor: Profa. Jane Fraga Tutikian

Pró-Reitor de Graduação: Prof. Vladimir Pinheiro do Nascimento

Diretor do Instituto de Informática: Profa. Carla Maria dal Sasso Freitas

Coordenador do Curso de Ciência de Computação: Prof. Prof. Raul Fernando Weber

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço à minha família, especialmente minha mãe, que sempre se esforçaram para me prover condições de me desenvolver como pessoa. Agradeço também por todo o amor, cuidado e apoio infinitos.

À Ana e Luiza, pela amizade, carinho e companheirismo - durante a graduação e além.

À Letícia, pela paciência, amor e cuidado durante o desenvolvimento deste trabalho.

Ao Professor Doutor Leandro Krug Wives, que me orientou no desenvolvimento deste trabalho.

RESUMO

O objetivo deste trabalho é criar uma ferramenta que avalie o valor da produção científica de pesquisadores e grupos de pesquisa, permitindo uma comparação simples e objetiva entre pares usando esses valores. Os dados-fonte da ferramenta são as publicações acadêmicas, seus respectivos autores e os relacionamentos entre eles coautorias, além de grupos de pesquisa. O valor total da produção científica de um autor ou grupo é calculado em função de uma equação que leva em conta diversos fatores como coautoria e autores serem membros de um mesmo grupo para definir a proporção do valor de cada publicação que é contabilizada no cálculo do resultado.

Palavras-chave: Produção Científica. Avaliação de Pesquisadores. Avaliação de Grupos de Pesquisa. Ranqueamento.

A Tool for Evaluating and Comparing Academic Scientific Production of Researchers and Research Groups

ABSTRACT

The goal of this work is to create a tool that calculates authors and research groups scientific production value, allowing simple and objective comparison between peers using those values. The tools source-data are academic publications, their respective authors and the relationship between them co-authorship, research groups. Value, based on some criteria such as Qualis and type of publication is attributed to every publication. The total value of the scientific production of an author or group is calculated by an equation that takes into account various factors such as co-authorship and authors being members of the same group to define the proportion of the value of each publication that is accounted in the calculation of the result.

Keywords: Scientific Production, Researchers Evaluation, Research Groups Evaluation, Ranking.

LISTA DE FIGURAS

Figura 3.1 Exemplo da <i>board</i> no utilizada no Trello	17
Figura 3.2 Exemplo de dados estruturados como grafo	19
Figura 4.1 Exemplo de grafo formado pelos dados	24
Figura 4.2 Ilustração da requisição feita ao servidor	26
Figura 4.3 Diagrama de atividade do merge de currículo	29
Figura 4.4 Exemplo de corpo da requisição de criação de grupo.	30
Figura 4.5 Exemplo de corpo da requisição de criação de grupo.	30
Figura 5.1 Tela de upload de currículo: dados inseridos	36
Figura 5.2 Tela de upload de currículo: enviado com sucesso.....	37
Figura 5.3 Tela de listagem de autores.....	38
Figura 5.4 Tela de detalhamento de autor	39
Figura 5.5 Tela de listagem de grupos	39
Figura 5.6 Tela de detalhamento de grupo	40

LISTA DE TABELAS

Tabela 4.1	Valores de cada classificação no QUALIS	33
------------	---	----

LISTA DE ABREVIATURAS E SIGLAS

ACID	Atomicity, Consistency, Isolation, Durability
API	Application Programming Interface
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
HATEOAS	Hypermedia As The Engine Of Application State
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
PPGC	Programa de Pós-Graduação em Computação
REST	Representational State Transfer
SGBD	Sistema de Gerenciamento de Banco de Dados
UFRGS	Universidade Federal do Rio Grande do Sul
URL	Uniform Resource Locator
XML	eXtensible Markup Language

SUMÁRIO

1 INTRODUÇÃO	10
2 TRABALHOS RELACIONADOS	12
2.1 Data Warehouse e Ferramenta OLAP para a Análise de Produção Acadêmica do PPGC	12
2.2 Análise de produção científica do Instituto de Informática da UFRGS baseada no currículo Lattes	12
2.3 Sistema de extração de dados acerca da produção científica do PPGC	13
2.4 ScriptLattes	13
2.5 PANDORA	14
2.6 Tabela Comparativa	14
3 METODOLOGIA, CONCEITOS E TECNOLOGIAS	16
3.1 Metodologia de Desenvolvimento	16
3.1.1 Kanban	16
3.2 Conceitos	17
3.2.1 RESTful API	17
3.2.2 Banco de Dados Orientado a Grafos.....	19
3.3 Tecnologias Utilizadas	19
3.3.1 Neo4j	19
3.3.1.1 Cypher Query Language	20
3.3.2 Node.js	21
4 DESENVOLVIMENTO DA SOLUÇÃO	22
4.1 Implementando o Banco de Dados	22
4.1.1 Modelagem dos Dados.....	22
4.1.1.1 Currículo Lattes e Sua Estrutura	22
4.1.1.2 Nós e Relações	23
4.2 Implementando a Web Service API	25
4.2.1 Upload de Currículos.....	25
4.2.2 Gerenciamento de grupos	29
4.2.3 Acesso a Dados	31
4.2.4 Índices	32
4.2.4.1 Calculando o valor de uma publicação	32
4.2.4.2 Calculando os índices de um Autor	33
4.2.4.3 Calculando os índices de um Grupo	34
5 ESTUDO DE CASO	35
5.1 Metodologia	35
5.2 Upload de Currículo	35
5.3 Listagem de Autores	36
5.4 Detalhamento de Autor	36
5.5 Listagem de Grupos	37
5.6 Detalhamento de Grupo	37
5.7 Análise dos Resultados	38
6 CONCLUSÃO	41
6.1 Avaliação dos Resultados	41
6.2 Melhorias	41
6.3 Trabalhos Futuros	42
REFERÊNCIAS	43

1 INTRODUÇÃO

O Programa de Pós-Graduação em Computação (PPGC) da Universidade Federal do Rio Grande do Sul (UFRGS) é avaliado pela CAPES como um dos melhores programas de pós-graduação em Computação do país. O programa conta com cerca de 50 professores, dos quais mais de 30 são bolsistas de Produtividade em Pesquisa do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) (PPGC, 2017). Esses dados evidenciam o grande foco do programa nas atividades de pesquisa e a consequente publicação trabalhos científicos em diversos veículos. Os trabalhos publicados pelos pesquisadores do PPGC são catalogados em seus respectivos Currículos Lattes, na Plataforma Lattes.

O Currículo Lattes tornou-se um padrão nacional no registro da vida pregressa e atual dos estudantes e pesquisadores do país, e é hoje adotado pela maioria das instituições de fomento, universidades e institutos de pesquisa do país. A Plataforma Lattes, provida pelo CNPq, integra bases de dados de Currículos, de Grupos de Pesquisa e de Instituições em um único Sistema de Informações (CNPQ, 2017).

Os veículos, através dos quais os trabalhos são publicados, são classificados pelo sistema QUALIS. O QUALIS é um sistema que disponibiliza uma lista com a classificação dos veículos utilizados pelos programas de pós-graduação para a divulgação da sua produção (CAPES, 2017a).

Porém, mesmo com amplo o registro da produção científica oferecido pela plataforma Lattes e a classificação dos veículos de publicação pelo QUALIS, a avaliação da produção de um pesquisador ou de um grupo ainda é um processo complexo e trabalhoso. Complexo, pois exige observar uma a uma as publicações de pesquisadores e grupos, levando em conta a avaliação do veículo em que foi publicada e, no caso de grupos de pesquisa, quais coautores envolvidos fazem parte do grupo em questão. Trabalhoso, pois devido ao potencial volume de dados a serem analisados e às relações entre os dados que precisam ser estabelecidas e medidas - muitas vezes manualmente. Assim, é de grande interesse desenvolver uma forma de facilitar o processo de avaliação, estabelecendo uma maneira automatizada de gerar indicadores de performance simples e comparáveis, como uma pontuação.

O objetivo deste trabalho é desenvolver uma ferramenta que permita a análise e comparação objetiva e simplificada da produção acadêmica de pesquisadores e grupos de pesquisa do PPGC com a de seus pares tanto do próprio PPGC, quanto externos.

A metodologia proposta é a de calcular indicadores numéricos, facilmente comparáveis, que expressem características como qualidade e volume da produção científica dos indivíduos ou grupos analisados. Os dados de produção científica a partir dos quais serão calculados os indicadores serão os catalogados no Currículo Lattes de cada pesquisador. O cálculo utilizará uma heurística que leva em conta o volume e os tipos de publicação, a classificação no sistema QUALIS atribuída aos veículos nos quais os trabalhos foram publicados e, no caso de grupos de pesquisa, também o grau de cooperação entre autores envolvidos. O escopo deste trabalho e da solução desenvolvida será limitado a publicações do tipo “artigo publicado” e “trabalho em evento”, como definidos no currículo Lattes.

Para tanto, este trabalho está organizado da seguinte forma. O Capítulo 2 analisa trabalhos já realizados na mesma área. O Capítulo 4 aborda o desenvolvimento da solução proposta, detalhando conceitos e tecnologias utilizados e seu funcionamento. O Capítulo 5, apresenta um estudo de caso, onde é apresentado um exemplo prático do uso da solução desenvolvida no capítulo anterior. Finalmente, o Capítulo 6 avalia os resultados obtidos, desafios encontrados no desenvolvimento do trabalho e apresenta sugestões de melhoria e extensão para trabalhos futuros.

2 TRABALHOS RELACIONADOS

Este capítulo irá analisar trabalhos já realizados cuja área é a mesma deste trabalho, observando pontos em comum e destacando diferenças entre os problemas e soluções apresentadas.

2.1 Data Warehouse e Ferramenta OLAP para a Análise de Produção Acadêmica do PPGC

Em (KUPLICH, 2013) é proposta uma solução em que os dados da produção científica do PPGC são armazenados em um banco de dados relacional, seguindo um modelo dimensional. Os dados armazenados são então analisados por uma ferramenta já existente, extraindo informações sobre volume de produção científica por autor, ano, área, etc. Os dados utilizados teriam como origem o Aplicativo de Coleta de Dados da CAPES - atual módulo de Coleta da Plataforma Sucupira (CAPES, 2017c) - mas não apresenta solução para a importação desses dados para o sistema.

O trabalho aqui proposto se assemelha muito ao desenvolvido em (KUPLICH, 2013) no aspecto de ser uma base de dados de produção científica a ser analisada por uma ferramenta - parte da solução desenvolvida aqui - que extrai informação desses dados. Porém, as informações extraídas pela solução aqui proposta são referentes à qualidade da produção científica - ainda que informações sobre volume também possam ser extraídas. Difere também a origem dos dados - aqui originários da plataforma Lattes (CNPQ, 2017) - e o fato de ser implementada uma solução de importação dos dados obtidos da origem para a base de dados.

2.2 Análise de produção científica do Instituto de Informática da UFRGS baseada no currículo Lattes

Em (LIMA, 2014) é proposta uma solução para análise dos dados dos pesquisadores do Instituto de Informática da UFRGS. A solução é composta de uma base de dados que armazena as informações dos pesquisadores e uma interface *web*. Os dados utilizados pela solução proposta em (LIMA, 2014) são uma combinação dos dados cadastrados manualmente nela e currículos em formato XML obtidos na plataforma Lattes. A esses

dados também são agregados dados de classificação de veículos de publicação no Qualis. A interface *web* permite a realização de diversas consultas sobre esses dados, listando informações sobre grupos e autores, exibindo gráficos de volume da produção - por autores, grupos, etc.

O sistema aqui proposto é semelhante ao desenvolvido em (LIMA, 2014) em relação a ser uma base de dados que armazena currículos Lattes e utiliza a classificação do Qualis para extrair informações a respeito da produção científica. Ainda assim, a solução aqui proposta diverge em pontos relevantes, como a forma como os dados dos currículos são inseridos e armazenados no sistema - o arquivo XML do currículo é processado, gerando entradas no banco de dados em vez de armazenar o arquivo em si - e no fato de prover o cálculo de índices de qualidade da produção, além informações de volume.

2.3 Sistema de extração de dados acerca da produção científica do PPGC

Em (DRABACH, 2015) é proposta a criação de um sistema que automatiza a extração de dados da produção científica do PPGC a partir dos relatórios gerados pela plataforma Sucupira (CAPES, 2017c) e disponibiliza essas informações para que possam ser facilmente consumidas, servindo como fonte de informação para outras aplicações.

Assim como trabalho aqui proposto, o sistema em (DRABACH, 2015) também é desenvolvido na forma de uma *web service* API REST, buscando facilitar o uso das funções do sistema por outros sistemas. Diferente do trabalho aqui proposto, em relação às informações extraídas dos dados, a solução em (DRABACH, 2015) não extrai dados indicadores da qualidade da produção acadêmica.

2.4 ScriptLattes

O ScriptLattes é uma ferramenta - proposta em (MENA-CHALCO; JUNIOR, 2009) - que permite a sumarização de dados provenientes da plataforma Lattes, correlacionando - de maneira automatizada - conjuntos de currículos e produzindo relatórios com conhecimento útil sobre a produção científica a partir dos dados. Podem ser gerados relatórios a respeito do volume de produção de um grupo de pesquisadores, quais tipos de publicação - e sua proporção - são produzidos, como é a colaboração entre os pesquisadores envolvidos e até mesmo sua distribuição geográfica, entre outros.

Porém, o ScriptLattes, diferente do trabalho proposto aqui, não atribui a classificação QUALIS aos trabalhos processados e não calcula índices indicadores da qualidade da pesquisa desenvolvida. Ainda outra diferença encontrada, é a de que o ScriptLattes implementa a busca automática aos dados diretamente na plataforma Lattes - a solução aqui proposta depende de meios externos para que os dados sejam obtidos, implementando somente a funcionalidade de inclusão dos dados do currículo na base de dados.

2.5 PANDORA

PANDORA é uma ferramenta - proposta em (NASCIMENTO, 2013) - que permite a visualização e análise de redes de coautoria na forma de uma rede social de pesquisadores. A ferramenta apresenta os dados de maneira gráfica e interativa, além de extrair métricas que expressam dados sobre a rede e os pesquisadores que fazem parte dela - como grau de colaboração de um pesquisador com seus pares, relevância de um pesquisador na rede, entre outras.

Assim como o trabalho aqui proposto, a ferramenta pandora utiliza um grafo para representar as relações de coautoria, além de utilizar os relacionamentos entre autores na composição das métricas exibidas. Diferente da solução proposta, a ferramenta PANDORA só trata de publicações no que diz respeito a obter dados de coautoria, sem tratar de outros aspectos - como métricas das publicações em si, sejam de volume ou, o foco deste trabalho, de qualidade.

2.6 Tabela Comparativa

O objetivo desta tabela é comparar de maneira clara cada um dos trabalhos em relação a características relevantes no contexto deste trabalho. Foi realçada a característica de análise de qualidade da produção científica para destacar o diferencial que o presente trabalho apresenta.

Características	Kuplich	Lima	Drabach	ScriptLattes	Pandora	Ferramenta Proposta
Base de dados de prod. científica	Sim	Sim	Sim	Não	Sim	Sim
Análise Qualidade	Não	Não	Não	Não	Não	Sim
Análise Volume	Sim	Sim	Sim	Sim	Não	Sim
Tem informação de QUALIS	Não	Sim	Sim	Sim	Não	Sim
Dados do Currículo Lattes	Não	Sim	Não	Sim	Não	Sim
Interface	Gráfica	Web	Web, API	Gráfica, Linha de comando	Gráfica	API
Aquisição de dados	Manual	Manual	Manual	Automática	Manual	Manual
Ingestão de dados	Automática	Manual	Automática	Automática	Automática	Automática

Fonte: Elaborada pelo autor

3 METODOLOGIA, CONCEITOS E TECNOLOGIAS

Neste capítulo será apresentada e justificada a abordagem adotada no desenvolvimento da solução proposta. Em seguida serão introduzidos conceitos utilizados na definição da ferramenta. Por fim, serão apresentadas as tecnologias sobre as quais a solução foi desenvolvida.

3.1 Metodologia de Desenvolvimento

O processo de desenvolvimento deste trabalho se baseou nos *princípios ágeis* (The Agile Alliance, 2001), seguindo o método *Kanban* (ANDERSON, 2010) de desenvolvimento de software. No processo estabelecido, o orientador deste trabalho foi o cliente e usuário, estabelecendo requisitos e validando as funcionalidades implementadas. O contato com o cliente ocorreu em reuniões semanais, onde eram estabelecidos requisitos e validadas as entregas.

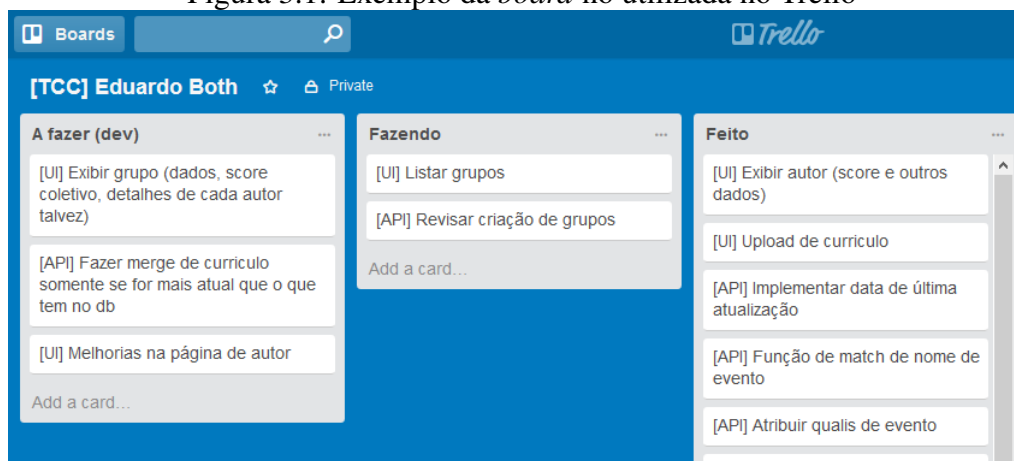
3.1.1 Kanban

O Kanban no contexto de desenvolvimento de software é um método para gerenciar a carga de trabalho em um time de desenvolvimento. É definida uma *board* (ou Kanban), onde são comumente definidas três colunas - o número de colunas pode variar de acordo com as necessidades do time -, onde são colocados os *cards* que representam as tarefas a serem executadas. Cada coluna representa um estado possível para um dado *card* que seja colocado nela:

- **A fazer:** esta coluna é o *backlog* dos *cards* de tarefas que já foram definidas, mas ainda não foram iniciadas.
- **Fazendo:** aqui ficam representadas as tarefas que estão atualmente sendo desenvolvidas; neste trabalho, foi definido um limite de até três tarefas sendo desenvolvidas em paralelo - o que só ocorreu em casos em que as tarefas possuíam relação entre si e a conclusão de uma influenciava na conclusão das demais. Em geral esta coluna contém somente um *card* e raramente se chegou ao limite de tarefas permitidas.
- **Feito:** nesta coluna são colocadas todas as tarefas concluídas.

Neste trabalho a *board* foi construída utilizando o Trello, uma ferramenta web que permite a criação e gerenciamento flexível e compartilhado de *boards*. Na Figura 3.1 é possível visualizar o estado da *board* utilizada no desenvolvimento deste trabalho.

Figura 3.1: Exemplo da *board* no utilizada no Trello



Fonte: Trello

3.2 Conceitos

Com o objetivo de tornar possível e fácil a utilização dos recursos providos por esta solução em outros sistemas, optou-se por defini-la como uma *Web Service Application Program Interface* (API) implementada sobre o protocolo HTTP. Mais especificamente, foi adotado o padrão de arquitetura REST (*Representational State Transfer*) para esta API. Para persistência dos dados do sistema, devido à forte relevância semântica dos relacionamentos entre as entidades armazenadas, foi escolhida uma solução de banco de dados orientada a grafos.

3.2.1 RESTful API

Representational State Transfer (REST) é um estilo de arquitetura para sistemas de hypermedia distribuídos e é definido por seis princípios arquiteturais (FIELDING, 2000):

- **Cliente-servidor:** um serviço REST deve implementar uma arquitetura cliente-servidor.
- **Stateless:** a comunicação entre cliente e servidor deve ser *stateless* por natureza, onde cada requisição do cliente para o servidor deve conter toda informação ne-

cessária pare que seja compreendida, sem depender de qualquer informação de contexto armazenada no servidor. O estado de uma sessão deve ser mantido pelo cliente.

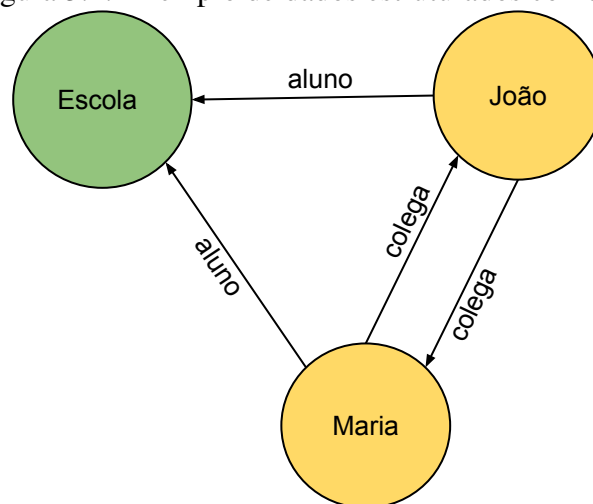
- Cache: os dados em uma resposta do servidor a uma requisição de um cliente devem ser explicitamente identificados como cacheáveis ou não cacheáveis. Se os dados forem marcados como cacheáveis, então um cache cliente poderá reutilizá-los para responder a futuras requisições equivalentes.
- Sistema em camadas: não deve ser possível a um cliente determinar se está interagindo diretamente com um servidor final (como um banco de dados) ou com um intermediário (uma API, cache, *proxy* ou outra camada de abstração).
- Interface uniforme: o serviço deve implementar uma interface uniforme seguindo os seguintes princípios:
 - Identificação de *resources*: *resources* são identificados individualmente nas requisições.
 - Manipulação de *resources* através de representações: uma representação é a forma como o servidor mostra as informações de um *resource* a um cliente. Um cliente que possua uma representação de um *resource* deve ser capaz de manipular (alterar e apagar) este *resource*.
 - Mensagens auto descritivas: cada mensagem entre cliente e servidor deve conter informação suficiente para definir como ela deve ser processada (e.g. informar que os dados da mensagem estão no padrão XML).
 - *Hypermedia as the engine of application state* (HATEOAS): deve ser possível a um cliente interagir com um servidor inteiramente através de *hypermedia* provida dinamicamente pelo servidor.

Uma *web service* API que aplica os princípios REST é denominada *RESTful* API, se baseia no padrão URL para identificar *resources* e usa métodos HTTP (*OPTIONS*, *GET*, *PUT*, *POST* e *DELETE*) para definir a interação com esses *resources* (RICHARDSON; AMUNDSEN; RUBY, 2013).

3.2.2 Banco de Dados Orientado a Grafos

Banco de dados orientado a grafos é um banco de dados baseado em teoria dos grafos, onde os dados são modelados como nós, arestas (ou relações) e propriedades. Nós representam as entidades e contêm suas informações na forma de propriedades; arestas representam as relações entre entidades, podendo também conter propriedades relevantes à relação (ANGLES; GUTIERREZ, 2008). A Figura 3.2 ilustra dados estruturados na forma de um grafo, onde um nó representa uma escola, outros dois nós representam pessoas e as relações entre os nós expressam as relações entre as entidades: João e Maria são colegas e alunos da escola.

Figura 3.2: Exemplo de dados estruturados como grafo



Fonte: Elaborada pelo autor

3.3 Tecnologias Utilizadas

A construção do servidor HTTP sobre o qual a *RESTful* API é definida foi feita sobre o *Node.js*, um interpretador de código *JavaScript*. A definição e construção do banco de dados foi feita utilizando o SGBD *Neo4j*, nativamente orientado a grafos.

3.3.1 *Neo4j*

O *Neo4j*, desenvolvido pela *Neo Technology, Inc.*, é um SGBD nativamente orientado a grafos - ou seja, as informações são gravadas no sistema diretamente como grafos, não como abstração de uma outra estrutura de dados subjacente. Como banco de dados,

o *Neo4j* é classificado como transacional e aderente a propriedades ACID. Os dados são armazenados na forma de nós, relações (arestas) e propriedades - como descrito em 3.2.2. Também é possível adicionar etiquetas a cada um destes elementos, facilitando a construção de *queries*. O *Neo4j* é multiplataforma, com versões para os sistemas operacionais Windows, MacOS e Linux (NEO4J, 2017).

3.3.1.1 Cypher Query Language

A manipulação - criação, alteração e exclusão - de dados no *Neo4j* é feita através da linguagem de *query Cypher*. *Cypher* é uma linguagem declarativa, inspirada na SQL, voltada para uso sobre grafos e desenvolvida para o *Neo4j* (CYPHER, 2017).

A linguagem usa uma sintaxe bastante gráfica para representar *patterns* a serem utilizados pelos comandos da linguagem. Na construção das *queries*, nós são representados por identificadores entre parênteses, e.g. (n) , ou até mesmo parênteses vazios. Relações são representadas por "setas" ligando os nós, e.g. $(a) \rightarrow (b)$, e também podem ter identificadores associados a elas, e.g. $() \text{--} [rel] \text{--} ()$. Relações e nós também podem ter *labels* associadas a eles de forma simples, permitindo delimitar ainda um *match*: $(a:Pessoa) \text{--} [:GOSTA] \text{--} (b:Comida)$. Os principais comandos da linguagem são os seguintes:

- **MATCH:** o comando MATCH exerce função muito semelhante a do comando SELECT da linguagem SQL. Este comando permite buscar na base de dados por dados que se encaixem no *pattern* que é passado como parâmetro. Também é possível usar a clausula WHERE em conjunto com o comando, permitindo filtrar os resultados. O código abaixo busca e retorna todos os nós representando pessoas que gostem de comer feijão e cujo nome não seja Maria:

```
1 MATCH (p:Pessoa) --[:GOSTA]-->(c:Comida)
2 WHERE p.nome <> 'Maria' AND c.nome = 'Pudim'
3 RETURN p
```

- **CREATE:** o comando CREATE é equivalente ao comando INSERT da linguagem SQL. É através dele que são inseridos registros na base de dados. O comando recebe um *pattern* com informações dos nós ou relações a serem criados, podendo conter valores para as propriedades deles. Múltiplos nós e relações podem ser criados em um único comando CREATE. O código abaixo demonstra o uso do comando CREATE, combinado ao comando MATCH em uma *query* que busca uma pessoa chamada Maria, cria uma comida e cria uma relação do tipo de gostar entre a comida e a pessoa chamada Mariad:

```
1 MATCH (p:Pessoa) WHERE p.nome = 'Maria'  
2 CREATE (p)-[:GOSTA]->(c:Comida { nome: 'Arroz' })
```

Como visto, a linguagem *Cypher* permite a combinação de diversos comandos em uma *query*, o que associado aos demais comandos da linguagem não apresentados aqui, possibilita a criação de séries de comandos complexas e poderosas. Informações mais completas sobre a linguagem, tutoriais e documentação podem ser encontradas em (CYPHER, 2017).

3.3.2 Node.js

Node.js é um interpretador de código *JavaScript* mantido pela *Node.js Foundation*, com versões para diversos sistemas operacionais. Tem como característica ser assíncrono e orientado a eventos. O *Node.js* também oferece nativamente funcionalidades que permitem construir um servidor *web* baseado no protocolo HTTP com facilidade (NODEJS, 2017).

4 DESENVOLVIMENTO DA SOLUÇÃO

Neste capítulo serão definidos os componentes e funcionalidades do sistema, detalhando seu funcionamento e processo de desenvolvimento.

4.1 Implementando o Banco de Dados

Esta seção mostra como foi aplicado o conceito de banco de dados orientado a grafos à solução proposta, detalhando modelagem do banco de dados.

4.1.1 Modelagem dos Dados

O *Neo4j* não implementa um *schema* no mesmo sentido em que bancos de dados relacionais costumam implementar. A definição das entidades (nós), relacionamentos e propriedades não é definida de maneira fixa no sistema de gerenciamento do banco de dados - como é o caso de tabelas, chaves estrangeiras, etc. em bancos relacionais. Assim, as definições dos nós, relações e suas propriedades são definidas e mantidas - opcionalmente - pela aplicação utilizando a base de dados, que permanece flexível neste aspecto. A ausência desse tipo de restrição definida diretamente no banco de dados permite a ocorrência de inconsistências entre as estruturas de dados, o que pode ser um problema em sistemas que necessitam de forte consistência, mas é um aspecto desejável na solução implementada neste trabalho, devido a natureza inconsistente das informações do currículo Lattes.

4.1.1.1 Currículo Lattes e Sua Estrutura

O currículo Lattes de um pesquisador é obtido a partir da Plataforma Lattes, que oferece a funcionalidade de exportar um currículo como um arquivo XML. Este arquivo contém informações pessoais, profissionais e histórico de produção científica do pesquisador - as mesmas informações exibidas na página do pesquisador na Plataforma Lattes.

As informações contidas no currículo são, em grande parte, inseridas manualmente pelo titular e pouco têm sua consistência verificada pelo sistema, o que resulta na natureza inconsistente das informações citada acima - e.g. um pesquisador cadastra em seu currículo um artigo publicado inserindo os nomes completos dos autores; enquanto

isso, outro pesquisador, coautor do artigo, cadastra o mesmo artigo, mas inserindo os nomes dos autores de forma abreviada; a Plataforma Lattes não faz automaticamente a correlação entre os nomes dos autores escritos de forma diferente, introduzindo assim uma inconsistência.

É fundamental, então, que a solução proposta seja capaz de acomodar inconsistências - tanto na modelagem dos dados, quanto no processamento deles.

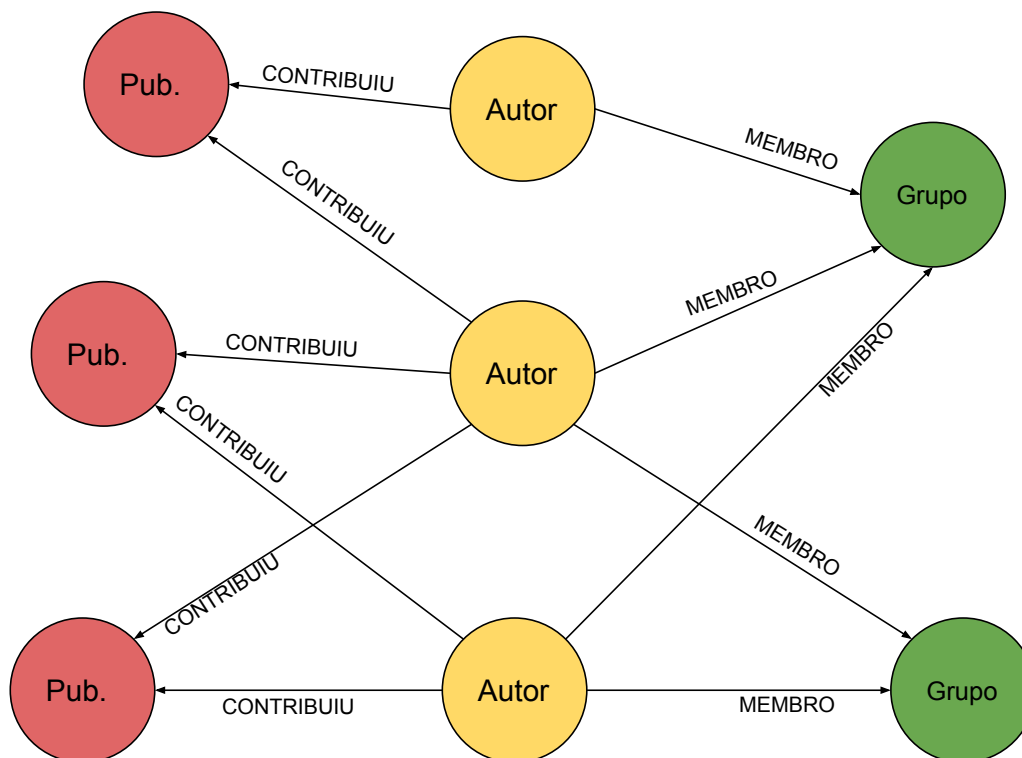
4.1.1.2 Nós e Relações

A solução implementada utiliza três tipos de nós:

- **Autor:** representa um autor, que pode ser tanto um pesquisador cujo currículo foi completamente inserido no sistema, quanto um autor que contribuiu para alguma publicação presente no sistema. A diferenciação entre os dois tipos é feita por uma *flag*, que sinaliza se o nó representa um autor cujo currículo já foi completamente incluído na base de dados ou se é apenas um autor sem mais dados. Este nó armazena em suas propriedades informações do autor relevantes ao sistema: nome completo, data da última atualização do currículo e um número identificador único que é atribuído ao currículo pela plataforma Lattes. O único dado obrigatório para este nó é o nome completo do autor, pois é o único sempre presente no currículo, tanto no caso de um pesquisador, quanto no caso de um coautor listado em uma publicação. É importante notar que autores listados como coautores de uma publicação listada no currículo de um pesquisador podem nem sequer ter cadastro na plataforma Lattes. Um nó deste tipo tem uma *tag* “Autor” atribuída a ele.
- **Publicação:** representa uma publicação de um ou mais autores. No currículo Lattes de um autor as publicações podem ser agrupadas em diversos tipos - artigos publicados, livros, conferências, etc. Esta característica é refletida neste nó pelo campo tipo. Cada tipo de publicação tem seu conjunto de propriedades, que tende a diferir entre cada tipo, com algumas propriedades comuns a todos os tipos - como QUALIS atribuído à publicação, ano no qual aquele QUALIS foi atribuído e o próprio tipo da publicação. Informações dos autores da publicação não são mantidas nas propriedades de seu nó. Toda publicação também possui como propriedade um identificador único atribuído no momento de sua inserção no sistema. Um nó deste tipo tem uma *tag* “Publicacao” atribuída a ele.
- **Grupo:** representa uma associação de autores, podendo representar desde um grupo

de pesquisa até um grupo arbitrário qualquer - conjunto dos pesquisadores membros do PPGC, por exemplo. Um nó do tipo Grupo possui apenas o nome do grupo e um identificador único como propriedades. Um nó deste tipo tem uma *tag* “Grupo” atribuída a ele.

Figura 4.1: Exemplo de grafo formado pelos dados



Fonte: Elaborada pelo autor

Os nós são ligados entre si por dois tipos de relação:

- **Contribuição:** é a relação que indica que um autor contribuiu com uma publicação - ou seja, está na lista de autores dela. Um autor pode ter relações de contribuição com diversas publicações e uma publicação pode ter relações de contribuição com diversos autores, caracterizando uma relação "n:n". As relações deste tipo são identificadas pela *tag* “CONTRIBUIU”.
- **Membro:** relaciona grupos e seus membros, indicando que um autor é membro de um grupo. É uma relação “n:n” entre grupos e autores. As relações deste tipo são identificadas pela *tag* “MEMBRO”.

4.2 Implementando a *Web Service API*

Nesta seção será detalhada a implementação da *Web Service API*, a interface que fornece métodos para manipulação dos dados armazenados no banco de dados, bem como realiza o cálculo dos índices indicadores de performance da produção científica de pesquisadores e grupos de pesquisa. Inicialmente, serão apresentadas as funcionalidades oferecidas pela API e seu funcionamento. Em seguida, será detalhada a forma de cálculo dos índices e como eles foram definidos.

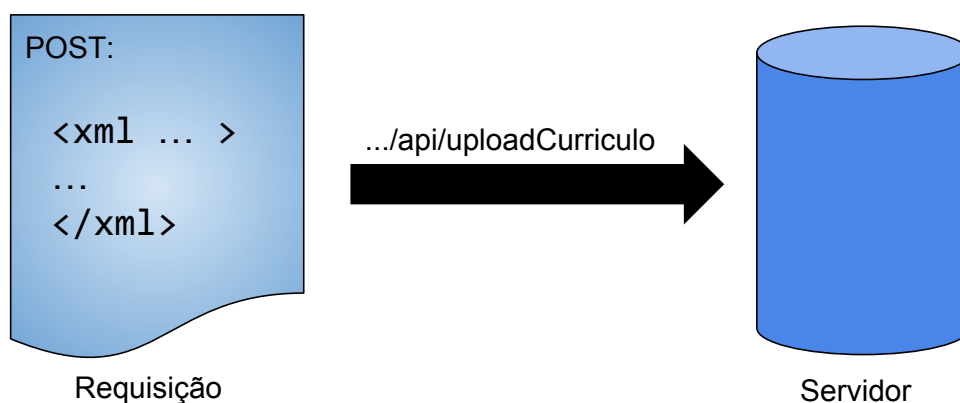
4.2.1 *Upload de Currículos*

Uma das funcionalidades mais relevantes oferecidas pela solução proposta e fundamental para o funcionamento da solução proposta é o *upload* de currículos Lattes - a ação de transferir os dados de um currículo Lattes para a base de dados da aplicação. Este processo consiste em fazer uma requisição para a API enviando o conteúdo do arquivo XML do currículo de um pesquisador exportado da plataforma Lattes. Estes dados são processados, interpretados e salvos no banco de dados do sistema. É importante notar que a obtenção destes arquivos não faz parte do escopo deste trabalho.

O *upload* de um currículo passa pelas seguintes etapas:

- **Chamada da API pelo cliente:** através da rede, o cliente faz uma requisição HTTP do tipo POST à URL *http://<host>/api/uploadCurriculo* - onde *<host>* é o endereço *web* do servidor. O cliente envia no corpo desta requisição o conteúdo do arquivo XML do currículo que deseja inserir na base de dados. Este processo é ilustrado na figura 4.2.
- **Parsing dos dados:** ao receber a requisição, o servidor faz o *parsing* do conteúdo da requisição, extraindo as informações do currículo relevantes ao sistema:
 - **Pesquisador:** do pesquisador ao qual o currículo pertence, são extraídos o número de identificação dele na plataforma Lattes, a data da última atualização do currículo e o nome completo.
 - **Publicações:** de todas as publicações constantes no currículo, são extraídas somente as de natureza “completa” e que sejam dos tipos “artigo publicado” ou “trabalho em evento”. Esta limitação existe devido ao fato de serem os

Figura 4.2: Ilustração da requisição feita ao servidor



Fonte: Elaborada pelo autor

únicos tipos de publicação cujos veículos têm classificação no QUALIS. As publicações também contêm os dados de seus autores.

- **Atribuição da classificação QUALIS às publicações:** cada publicação extraída do currículo recebe como novo atributo sua classificação no QUALIS - A1, A2, B1, B2, B3, B4, B5 ou C - de acordo com o ano de publicação e o veículo em que foi publicada. A heurística para atribuição de uma classificação é a seguinte:

```

1   Busca a classificação do veículo no ano de publicação.
2   Se encontrada, atribui a classificação a publicação.
3   Se não encontrar, busca nos anos anteriores disponíveis e retorna o mais
    recente encontrado.
4   Se encontrada, atribui a classificação a publicação.
5   Se não encontrar, busca nos anos posteriores disponíveis e retorna o primeiro
    encontrado.
6   Se encontrada, atribui a classificação a publicação.
7   Se não encontrar, QUALIS fica com o valor "NA" (não atribuído).

```

Essa heurística é motivada por diversos fatores. A classificação dos veículos no QUALIS não foi sempre feita anualmente - houve um período em que era feita a cada três anos, motivando a necessidade de procurar a classificação de um veículo não só no ano exato da publicação. Outro fator é o de que existem produções que foram publicadas em anos em que não há classificação do QUALIS disponível - seja em anos anteriores a existência do QUALIS, seja no ano atual, em que ainda não foi publicada a classificação dos veículos. -, motivando a busca em outros anos. O objetivo é de que a maioria das publicações tenham alguma classificação - mesmo que seja só uma aproximação.

- **Merge do Autor:** após a extração dos dados desejados do currículo, é iniciado o

processo de *merge* dos dados no sistema. O primeiro *merge* a ser feito é o do autor, seguindo o seguinte algoritmo:

```

1   Verifica se já existe um nó do tipo Autor contendo o mesmo número
    identificador do autor a ser adicionado.
2   Se não existe , é criado um novo nó com as informações do autor e a flag
    indicando que o currículo deste autor já foi adicionado ao sistema (flag "
    curriculoCompleto") marcada como verdadeira.
3   Se existe , é verificada a flag "curriculoCompleto":
4       Caso seja falsa , o nó é atualizado com as informações do currículo e a
    flag "curriculoCompleto" é marcada como verdadeira.
5       Caso seja verdadeira , é verificada se a data da última atualização
    presente no nó é anterior ou posterior a do currículo recebido:
6           Caso seja anterior , os dados são atualizados e o processo prossegue.
7           Caso seja igual ou posterior , todo o processo de upload do currículo é
    finalizado sem alterações nos dados do banco , pois ele já está atualizado.

```

O algoritmo é implementado na forma de uma *query* da linguagem *Cypher*, submetida ao banco de dados com as informações do autor:

```

1   MERGE (a:Autor { numeroIdCNPQ: {dadosAutor}.numeroIdCNPQ })
2   ON CREATE SET a = {dadosAutor}, a.curriculoCompleto = true
3   ON MATCH SET a.curriculoCompleto = true , a.ultimaAtualizacao = {dadosAutor}.
    ultimaAtualizacao

```

Onde *{dadosAutor}* é um objeto representando os dados do autor, passado como parâmetro para o comando MERGE.

- **Merge das Publicações:** caso o processo não tenha sido finalizado após o *merge* do pesquisador, é feito o *merge* das publicações. Para cada publicação, é seguido o seguinte algoritmo:

```

1   Busca um nó do tipo Publicação com mesmo tipo , título , ano de publicação e
    QUALIS.
2   Se não encontrado , é criado um nó para representar a publicação.
3   É criada uma relação do tipo CONTRIBUIU entre o nó encontrado/criado e o
    pesquisador , caso não exista.
4   Para cada autor da publicação:
5       Se o autor possuir número identificador do Lattes como propriedade :
6           Busca nó do tipo Autor com o mesmo número identificador .
7           Se não encontrado , é criado um nó do tipo Autor contendo os dados
    desse autor , com a flag "curriculoCompleto" marcada como falsa .
8           É criada uma relação do tipo CONTRIBUIU entre o nó encontrado/criado e
    o da publicação , caso não exista.
9       Se não possuir:
10          Busca nó do tipo Autor com o mesmo nome completo .
11          Se não encontrado , é criado um nó do tipo Autor contendo os dados
    desse autor , com a flag "curriculoCompleto" marcada como falsa .
12          É criada uma relação do tipo CONTRIBUIU entre o nó encontrado/criado e
    o da publicação , caso não exista.

```

A *query* utilizada para a implementar o algoritmo do *merge* das publicações consiste de um conjunto de comandos da linguagem *Cypher* executados em sequência. Como cada tipo de publicação tem propriedades diferentes, é necessária uma *query* diferente para cada tipo de publicação. No entanto, as diferenças são mínimas entre as *queries* de cada tipo e a lógica é idêntica para todos os tipos de publicação. Assim, o exemplo a seguir, correspondente ao *merge* de publicações do tipo "artigo publicado", basta para demonstrar a implementação:

```

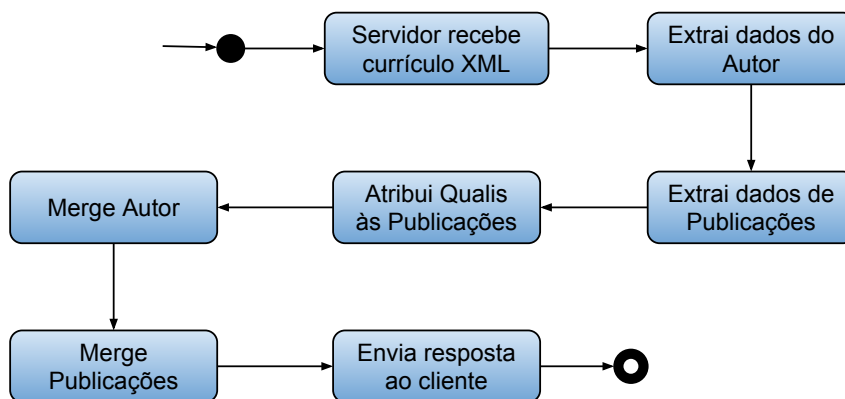
1  UNWIND {artigosPublicados} as artigo
2  MERGE (p:Publicacao
3      {
4          tipo: artigo.dados.tipo,
5          natureza: artigo.dados.natureza,
6          tituloNormalizadoDoArtigo: artigo.dados.tituloNormalizadoDoArtigo,
7          anoDoArtigo: artigo.dados.anoDoArtigo,
8          qualis: artigo.dados.qualis
9      })
10 ON CREATE SET p = artigo.dados
11 WITH p, artigo
12 UNWIND artigo.autores AS autor
13 OPTIONAL MATCH (a:Autor)
14   WHERE a.numeroIdCNPQ = autor.dados.numeroIdCNPQ AND a.numeroIdCNPQ <> ''
15 WITH count(a) as encontrados, autor, p
16 FOREACH(ignoreMe IN CASE WHEN encontrados = 0 THEN [1] ELSE [] END |
17   MERGE (aa:Autor { nomeCompletoNormalizado: autor.dados.
18     nomeCompletoNormalizado })
19   ON CREATE SET aa = autor.dados, aa.curriculoCompleto = false
20   MERGE (aa) -[:CONTRIBUIU]->(p)
21 )
22 FOREACH(ignoreMe IN CASE WHEN encontrados > 0 THEN [1] ELSE [] END |
23   MERGE (aa:Autor { numeroIdCNPQ: autor.dados.numeroIdCNPQ })
24   ON CREATE SET aa = autor.dados, aa.curriculoCompleto = false
25   MERGE (aa) -[:CONTRIBUIU]->(p)
26 )

```

Aqui *{artigosPublicados}* é a lista de publicações desse tipo a ser inserida no banco de dados. O comando UNWIND na primeira linha faz com que todos os demais comandos sejam aplicados uma vez a cada elemento da lista de artigos. Da linha 2 à linha 10 é feito o *merge* da publicação em si. Da linha 12 à linha 25 é feito o *merge* dos autores da publicação, onde comando UNWIND da linha 12 faz com que os próximos comandos sejam aplicados a cada um dos autores da publicação - implementando da linha 4 em diante do algoritmo. Nas linhas 16 e 21 o comando FOREACH é utilizado como *workaround* para a ausência de um comando de execução condicional na linguagem, fazendo com que para cada autor do artigo somente os comandos das linhas 17 à 19 ou 22 à 24 sejam executados.

É importante destacar que publicações não têm um número identificador no sistema Lattes da mesma forma que pesquisadores têm, por isso é necessário fazer a busca por uma publicação da forma como é feita na primeira linha do algoritmo. É também na linha um do algoritmo um dos diversos pontos onde podem ser introduzidas inconsistências: se dois pesquisadores forem coautores de uma mesma publicação e cada um deles adicioná-la com dados ligeiramente diferentes - como a grafia do título, por exemplo - podem ser criados dois nós distintos representando o que seria essencialmente a mesma publicação. Este problema pode ser mitigado utilizando formas mais sofisticadas de busca, mas sua eliminação é impossível devido à natureza dos dados disponíveis.

Figura 4.3: Diagrama de atividade do merge de currículo



Fonte: Elaborada pelo autor

4.2.2 Gerenciamento de grupos

A API oferece métodos para o gerenciamento - criação, atualização, exclusão - de grupos. Aqui descrevemos essas funcionalidades:

- **Criação:** para criar um grupo, o cliente faz uma requisição HTTP do tipo POST à URL `http://<host>/api/grupos` - onde `<host>` é o endereço *web* do servidor. O corpo dessa requisição deve ser um objeto no formato JSON contendo o nome do grupo e uma lista de números identificadores do sistema Lattes representando a lista de autores que fazem parte do grupo (Figura 4.4). O servidor recebe essa requisição e cria: um nó do tipo Grupo e relações do tipo MEMBRO entre o grupo e os nós do tipo Autor cujos números identificadores constem na lista. Também é atribuído um identificador único ao nó do grupo.

Figura 4.4: Exemplo de corpo da requisição de criação de grupo.

```

1  {
2    "dados": {
3      "nome": "Exemplo de Grupo"
4    },
5    "numeroIdCNPQAutores": [
6      "5558354805733623",
7      "5996993884846655"
8    ]
9  }

```

A *query* executada na criação do grupo no banco de dados é a seguinte:

```

1  WITH {grupo}.dados AS dadosGrupo
2  MERGE (g:Grupo { nome: dadosGrupo.nome }) SET g = dadosGrupo
3  WITH g
4  UNWIND {grupo}.numeroIdCNPQAutores AS idsAutores
5  MATCH (a:Autor) WHERE a.numeroIdCNPQ IN idsAutores
6  MERGE (a)-[:MEMBRO]->(g)

```

Onde *{grupo}* é o objeto JSON enviado na requisição.

- **Atualização:** para atualizar um grupo, o cliente faz uma requisição HTTP do tipo PUT à URL `http://<host>/api/grupos/<id>` - onde *<host>* é o endereço *web* do servidor e *<id>* é o identificador único do nó do grupo. O corpo dessa requisição deve ser um objeto no formato JSON contendo os dados do grupo atualizados (Figura 4.5).

Figura 4.5: Exemplo de corpo da requisição de criação de grupo.

```

1  {
2    "dados": {
3      "nome": "Exemplo de Grupo"
4    },
5    "numeroIdCNPQAutores": [
6      "5558354805733623",
7      "5996993884846655",
8      "5646846884846852",
9      "9874563215821684"
10   ]
11  }

```

- **Exclusão:** para excluir um grupo, o cliente faz uma requisição HTTP do tipo DELETE à URL `http://<host>/api/grupos/<id>` - onde *<host>* é o endereço *web* do servidor e *<id>* é o identificador único do nó do grupo. Esta requisição não tem corpo. O servidor exclui do banco de dados o nó que representa o grupo e todas as relações do tipo MEMBRO que existam entre ele e nós do tipo Autor.

4.2.3 Acesso a Dados

A API oferece diversos métodos para acesso aos dados, tanto listagem, quanto busca individual de registros e outras funcionalidades. Os métodos de busca de dados oferecidos pela API são os seguintes:

- **Lista Autores:** o cliente faz uma requisição HTTP do tipo GET à URL *http://<host>/api/autores* - onde *<host>* é o endereço *web* do servidor. O servidor responde a requisição com uma lista no formato JSON contendo os dados de todos os autores cuja *flag* "curriculoCompleto" está marcada como verdadeira.
- **Detalha Autor:** o cliente faz uma requisição HTTP do tipo GET à URL *http://<host>/api/autores/<id>* - onde *<host>* é o endereço *web* do servidor e *<id>* é o número identificador Lattes do autor buscado. O servidor responde a requisição com um objeto no formato JSON contendo os dados do autor, incluindo as publicações para as quais contribuiu e seus índices calculados - o detalhamento dos índices disponíveis e como são calculados é feito na seção 4.2.4.
- **Lista Publicações:** o cliente faz uma requisição HTTP do tipo GET à URL *http://<host>/api/publicacoes* - onde *<host>* é o endereço *web* do servidor. O servidor responde a requisição com uma lista no formato JSON contendo os dados de todas as publicações.
- **Detalha Publicação:** o cliente faz uma requisição HTTP do tipo GET à URL *http://<host>/api/publicacoes/<id>* - onde *<host>* é o endereço *web* do servidor e *<id>* é o identificador único atribuído à publicação. O servidor responde a requisição com um objeto no formato JSON contendo os dados da publicação, incluindo seus autores.
- **Lista Grupos:** o cliente faz uma requisição HTTP do tipo GET à URL *http://<host>/api/grupos* - onde *<host>* é o endereço *web* do servidor. O servidor responde a requisição com uma lista no formato JSON contendo os dados de todos os grupos.
- **Detalha Grupo:** o cliente faz uma requisição HTTP do tipo GET à URL *http://<host>/api/grupos/<id>* - onde *<host>* é o endereço *web* do servidor e *<id>* é o identificador único atribuído ao grupo. O servidor responde a requisição com

um objeto no formato JSON contendo os dados do grupo, incluindo seus autores membros e os índices do grupo calculados - o detalhamento dos índices disponíveis e como são calculados é feito na seção 4.2.4.

- **Calcula índices de grupo anônimo:** o cliente faz uma requisição HTTP do tipo POST à URL *http://<host>/api/indices* - onde *<host>* é o endereço *web* do servidor. No corpo desta requisição deve ser enviado um objeto do tipo JSON contendo uma lista de um ou mais números identificadores do sistema Lattes. O servidor responde a requisição com um objeto no formato JSON contendo os índices calculados para o grupo (ou indivíduo) listado - o detalhamento dos índices disponíveis e como são calculados é feito na seção 4.2.4. O objetivo deste método é prover uma forma genérica de cálculo de índices.

4.2.4 Índices

A solução proposta baseia o cálculo dos índices indicadores de desempenho sobre as publicações, que recebem pontuações de acordo com seu tipo e classificação no QUALIS. Outros modificadores podem ser aplicados sobre esses valores dependendo do tipo de índice sendo calculado. Os índices e sua forma de cálculo são baseados nos "Índices para Avaliação da Produção Bibliográfica Qualificada" definidos e utilizados pela CAPES para avaliação de produção científica - definidos nos documentos de área (CAPES, 2017b). Para o cálculo dos índices de apenas um autor, cada publicação do autor tem seu valor calculado e os valores de todas as publicações são somados - esta soma é sujeita a restrições dependendo do índice sendo calculado. O cálculo dos índices de um grupo é feito somando os índices dos autores membros do grupo - porém cada publicação tem seu valor dividido pelo número de autores dela presentes no grupo cada vez que os índices de um de seus autores é calculado, com objetivo que cada publicação seja contada somente uma vez (somando as parcelas) no cálculo final. Esses processos serão detalhados a seguir.

4.2.4.1 Calculando o valor de uma publicação

O valor de uma publicação é calculado a partir de seu tipo e de sua classificação no QUALIS seguindo a fórmula:

$$v = \text{tipo} * \text{qualis} / \text{nColaboradores}$$

Onde v é o valor final que a publicação contribui para o cálculo do índice, *tipo* é

um número entre 0 e 1 definido para o tipo da publicação - ambos os tipos de publicação no escopo deste trabalho, artigo publicado e trabalho em evento, receberam o valor 1 - e *qualis* é o valor atribuído a classificação do veículo de publicação no QUALIS (Tabela 4.1). A variável *nColaboradores* é relevante somente quando estiverem sendo calculados os índices de um grupo - nesse caso, o valor de uma publicação é dividido pelo o número de autores que dela que estão presentes no grupo sendo analisado.

Tabela 4.1: Valores de cada classificação no QUALIS

<i>QUALIS</i>	<i>Valor</i>
A1	1
A2	0,85
B1	0,7
B2	0,5
B3	0,2
B4	0,1
B5	0,05
C	0
NA	0

Fonte: Elaborada pelo autor

4.2.4.2 Calculando os índices de um Autor

Os índices de um autor são calculados somando os valores de todas as suas publicações, seguindo as seguintes restrições:

- **Proporção 1:3 de artigos para trabalho em evento:** somente três publicações do tipo *trabalho em evento* são levadas em conta para cada publicação do tipo *artigo publicado* - caso seja necessário excluir alguma publicação do cálculo devido a esta restrição, as de menor valor são excluídas primeiro.
- **Quadriênio mais recente:** somente publicações dos últimos quatro anos são levadas em conta no cálculo.

Observadas as restrições, os índices calculados para um Autor são os seguintes:

- **Geral:** é a soma dos valores de todas as publicações do autor.
- **Restrito:** é a soma dos valores de todas as publicações do autor que tenham classificação A1, A2 ou B1 no QUALIS.
- **Geral Para Artigos Publicados:** *idem* Geral, porém leva em conta somente publicações do tipo *artigo publicado*.

- **Restrito Para Artigos Publicados:** *idem* Restrito, porém leva em conta somente publicações do tipo *artigo publicado*.
- **Geral Para Trabalho em Evento:** *idem* Geral, porém leva em conta somente publicações do tipo *trabalho em evento*.
- **Restrito Para Trabalho em Evento:** *idem* Restrito, porém leva em conta somente publicações do tipo *Trabalho em Evento*.

4.2.4.3 Calculando os índices de um Grupo

Um grupo possui os mesmo índices de um autor. Os valores deles são calculados somando os valores dos índices de todos os autores membros do grupo - que neste caso sofrem uma alteração em seu valor influenciada pela variável *nColaboradores* apresentada em 4.2.4.1.

5 ESTUDO DE CASO

O capítulo anterior a apresentou a solução proposta, suas funcionalidades e os conceitos envolvidos em seu desenvolvimento. Este capítulo apresentará um estudo de caso de uso da *web service* API desenvolvida no capítulo anterior. O caso de uso consiste em uma interface de usuário na forma de uma aplicação *web* que age como cliente da API, consumindo seus métodos para prover funcionalidades ao usuário. A seguir, são apresentadas as telas dessa aplicação, suas funcionalidades e a forma como consomem a API.

O objetivo deste estudo é validar a efetividade da API em prover métodos de manipulação dos dados - leitura, escrita - e cálculo de índices. Como objetivo secundário, busca-se demonstrar a viabilidade da construção de uma interface de usuário - uma interface *web*, neste caso - que acessa e exibe as informações de produção científica.

Este estudo de caso não é exaustivo no uso das funcionalidades oferecidas pela API, focando apenas nas funcionalidades mais relevantes.

5.1 Metodologia

A bordagem utilizada foi, primeiro, definir uma interface *web* seguindo o *design pattern master-detail* (ORACLE, 2001), utilizando Grupos e Autores como entidades *master*, que poderiam ser detalhadas para se obter listagens de seus elementos, que por sua vez também poderiam ser detalhados para exibir suas diversas propriedades, principalmente seus índices. Em seguida, procurou-se implementar a interface que foi definida para, por fim, avaliar se foi possível atingir o resultado desejado utilizando apenas os métodos oferecidos pela API.

5.2 Upload de Currículo

A funcionalidade de *upload* de currículo foi implementada nessa interface utilizando um formulário onde o usuário insere o conteúdo do arquivo XML de um currículo exportado da plataforma (Figura 5.1) Lattes.

Quando o usuário pressiona o botão “enviar”, a aplicação faz a requisição de *upload* de currículo ao servidor, enviando os dados inseridos no formulário. Se o currículo

Figura 5.1: Tela de upload de currículo: dados inseridos

Início Autores Grupos Upload

Upload de Currículo XML

Insira o conteúdo do arquivo .xml

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?><CURRICULO-VITAE SISTEMA-ORIGEM-XML="LATTES_OFFLINE" NUMERO-IDENTIFICADOR="097883796467017" DATA-ATUALIZACAO="07072017" HORA-ATUALIZACAO="091748"><DADOS-GERAIS NOME-COMPLETO="Ana Lucia Ceterlich Bazzan" NOME-EM-CITACOES-BIBLIOGRAFICAS="BAZZAN, A. L. C.;BAZZAN, A;BAZZAN, ANA;BAZZAN, ANA L.C.;BAZZAN, ANA L. C.;Bazzan, Ana L.C.;CETERLICH BAZZAN, ANA LUCIA;Ana Bazzan" NACIONALIDADE="B" PAIS-DE-NASCIMENTO="Brasil" UF-NASCIMENTO="SP" CIDADE-NASCIMENTO="São Paulo" PERMISSAO-DE-DIVULGACAO="NAO" DATA-FALECIMENTO="" SIGLA-PAIS-NACIONALIDADE="BRA" PAIS-DE-NACIONALIDADE="Brasil"><RESUMO-CV TEXTO-RESUMO-CV-RH="Graduou-se em Engenharia pela Universidade de São Paulo (EPUSP); possui ainda os graus de mestre em Ciências da Computação pela Universidade Federal do Rio Grande do Sul (1992) e doutora pela Universidade de Karlsruhe (1997). Entre 1997 e 1998 atuou como "Research Assistant" na Universidade de Massachusetts (Amherst) no grupo do prof. Victor Lesser. Desde 1999 ocupa o cargo de professora na Universidade Federal do Rio Grande do Sul (desde 2016 como professora titular). Realizou pós-doutorado (2006-2007) na Universidade Würzburg com fellowship da Fundação Alexander von Humboldt. É bolsista de Produtividade em Pesquisa do CNPq - Nível 1C. Atividades Profissionais: Co-general chair da conferência 13th AAMAS (Paris, 2014), Area chair da conferência IJCAI 2017, general chair da conferência MATES 2014; invited plenary speaker na conferência AAMAS 2017. Assoc. Editor dos periódicos Journal of Autonomous Agents and Multiagent Systems (2010-2017) e Advances in Complex Systems (2007-2010); membro do corpo editorial do periódico Multiagent and Grid Systems (até 2017). Membro do board do IFAAMAS (2004-2008 e 2014-2020). Revisora de periódicos (seleção): Artificial Intelligence: Transportation research, Part C (Emerging technologies); IEEE Transactions on Knowledge and Data Engineering; IEEE Transactions on Intelligent Transportation Systems; Journal of Int. Transportation Systems; Int. J. of Autonomous Agents and Multiagent Systems; Int. J. of Modern Physics C; Games and Economic Behavior; Int. J. of Computational Intelligence and Applications; IEEE Transactions on Systems Man and Cybernetics; International journal of cooperative information systems; Eng. Applications of AI; Revista de Informatica Teórica e Aplicada; Journal of the Brazilian Computer Society. Tem experiência na área de Ciência da Computação, com ênfase em Sistemas Multiagentes, atuando principalmente nos seguintes temas: aprendizado em sistemas multiagente, alocação de tarefas em sistemas multiagente, coordenação em sistemas multiagente, simulação de tráfego veicular, controle semafórico, RoboCup Rescue, aplicações de teoria de jogos." TEXTO-RESUMO-CV-RH-EN="Ana Bazzan received her PhD in 1997 from the University of Karlsruhe, Germany. Her previous degrees are in Engineering from the Politechnic School of the University of São Paulo, Brazil, and an MSc. in Computer Science from the Institute of Informatics at the University of Rio Grande do Sul (UFRGS) in Porto Alegre, Brazil. From 1997 to 1998, she had a postdoc research associate position in the Multi-
```

⚡ Enviar!

Fonte: Elaborada pelo autor

for enviado com sucesso, o texto do formulário é substituído pela mensagem “Enviado” (Figura 5.2).

5.3 Listagem de Autores

Ao acessar a opção “Autores” no menu superior, o usuário é levado à tela de listagem de autores. A lista de autores mostrada é obtida a partir de uma chamada ao método da API que lista todos os autores cadastrados. (Figura 5.3)

5.4 Detalhamento de Autor

Ao clicar na coluna “#CNPQ” de uma das entradas da lista de autores (Figura 5.3), o usuário é levado à tela que detalha diversas informações do autor selecionado - nome, publicações, índices, grupos dos quais faz parte, etc. (Figura 5.4)

Figura 5.2: Tela de upload de currículo: enviado com sucesso

A imagem mostra a interface de usuário para o upload de currículo XML. No topo, há uma barra de navegação com os itens 'Início', 'Autores', 'Grupos' e 'Upload'. Abaixo, um cabeçalho centralizado indica 'Upload de Currículo XML'. O formulário principal contém o texto 'Insira o conteúdo do arquivo .xml' e uma área de texto grande que exibe o status 'Enviado'. Na base do formulário, há um botão verde com o ícone de um relâmpago e o texto 'Enviar!'.

Fonte: Elaborada pelo autor

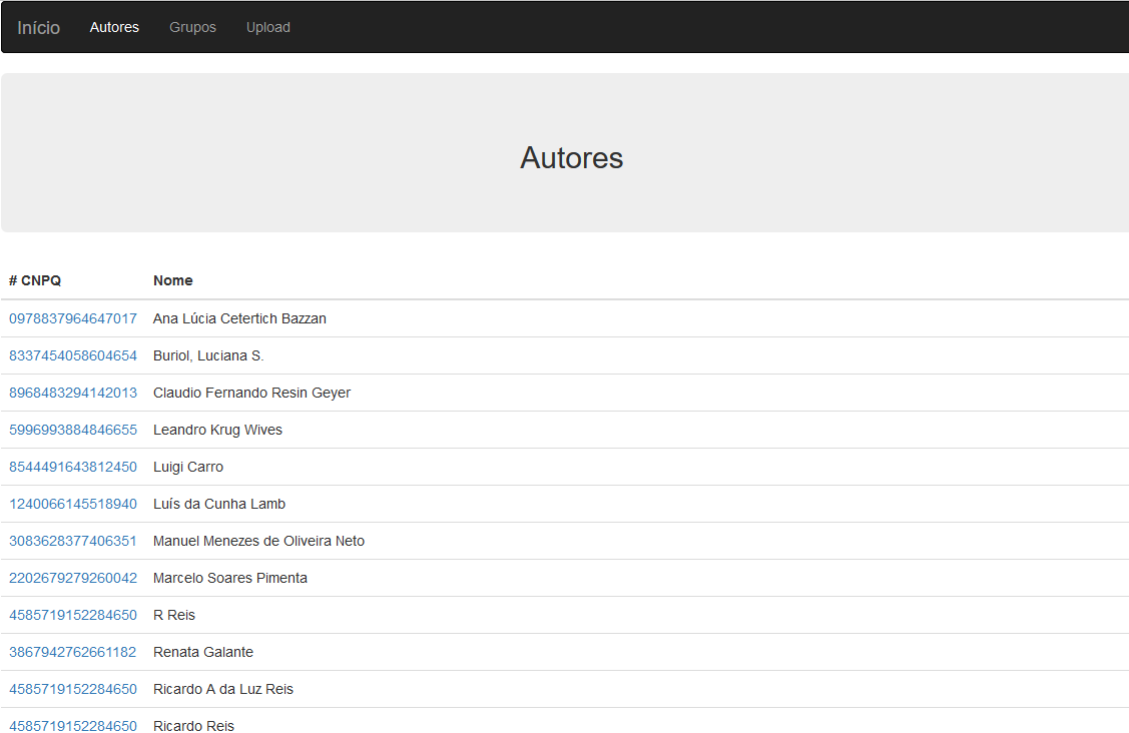
5.5 Listagem de Grupos

Ao acessar a opção “Grupos” no menu superior, o usuário é levado à tela de listagem de grupos. A lista de grupos mostrada é obtida a partir de uma chamada ao método da API que lista todos os grupos criados. (Figura 5.5)

5.6 Detalhamento de Grupo

Ao clicar na coluna “ID” de uma das entradas da lista de grupos (Figura 5.5), o usuário é levado à tela que detalha diversas informações do grupo selecionado - nome, índices e lista de membros (Figura 5.6). Ao clicar na coluna “#CNPQ” de uma das entradas da lista de membros, o usuário é levado à página de detalhamento do autor selecionado (Figura 5.4). As informações do grupo são obtidas a partir de uma chamada ao método da API de detalhamento do grupo. Os índices, porém, são obtidos separadamente, fazendo uma chamada a outro método da API, o de cálculo de índices de grupo.

Figura 5.3: Tela de listagem de autores



# CNPQ	Nome
0978837964647017	Ana Lúcia Cetertich Bazzan
8337454058604654	Buriol, Luciana S.
8968483294142013	Claudio Fernando Resin Geyer
5996993884846655	Leandro Krug Wives
8544491643812450	Luigi Carro
1240066145518940	Luis da Cunha Lamb
3083628377406351	Manuel Menezes de Oliveira Neto
2202679279260042	Marcelo Soares Pimenta
4585719152284650	R Reis
3867942762661182	Renata Galante
4585719152284650	Ricardo A da Luz Reis
4585719152284650	Ricardo Reis

Fonte: Elaborada pelo autor

5.7 Análise dos Resultados

Avalia-se que foi possível atingir o objetivo do estudo, demonstrando de maneira clara a viabilidade de se construir interfaces de usuário utilizando a API, exemplificando o uso de seus diversos métodos. Foi possível fazer a inserção de currículos na base de dados de maneira simples e fácil; os métodos da API proveem dados suficientes das entidades, facilmente exibidos sem necessidade de manipulações elaboradas. Foi identificado apenas um ponto de possível melhoria, no detalhamento de um grupo - os índices poderiam ser retornados diretamente na chamada ao método de detalhamento de grupo, em vez de exigir uma chamada de método extra para obtê-los.

Figura 5.4: Tela de detalhamento de autor

Leandro Krug Wives

WIVES, L. K.
Última Atualização: 11/07/2017 22:12:29

ÍND. GERAL: **6.85**
ÍND. REST.: **5.85**

Grupos

PPGC INF UFRGS

Colaborações

ALESSANDRO DA SILVEIRA DIAS
ANTÔNIO RODRIGO DELEPIANE DE VIT
Agostinho Bombassaro Junior
Alejandro Fernandez
Alencar Machado
Alexandre Hauber da Silva
Alexandre de Oliveira Zamberlam
Alicia Díaz
Amel Bouzeghoub

Artigos Publicados

Artigo	Índ. Geral	Índ. Rest.
Reactive, Proactive, and Extensible Situation-Awareness in Ambient Assisted Living (2017)	A1	3.15
The influence of readability aspects on the user's perception of helpfulness of online reviews (2016)	B5	2.85
Pylinguistics: an open source library for readability assessment of texts written in Portuguese (2016)	B5	
Improving business process retrieval using categorization and multimodal search (2016)	A2	
Grouping of business processes models based on an incremental clustering algorithm using fuzzy similarity and multimodal search (2016)	A1	

Fonte: Elaborada pelo autor

Figura 5.5: Tela de listagem de grupos

Grupos

ID	Nome
rkGadlrb	Banca da Defesa
ByKD5rtSW	PPGC INF UFRGS

Fonte: Elaborada pelo autor

Figura 5.6: Tela de detalhamento de grupo



The screenshot shows a web interface with a dark navigation bar at the top containing the links 'Início', 'Autores', 'Grupos', and 'Upload'. Below this, a white box displays the title 'Banca da Defesa' on the left and two numerical indicators on the right: 'ÍND. GERAL' with a value of **13.66** and 'ÍND. REST.' with a value of **12.56**. Below the white box is a table with two columns: '# CNPQ' and 'Nome'. The table lists three members: Leandro Krug Wives (CNPQ: 5996993884846655), Renata Galante (CNPQ: 3867942762661182), and Marcelo Soares Pimenta (CNPQ: 2202679279260042).

# CNPQ	Nome
5996993884846655	Leandro Krug Wives
3867942762661182	Renata Galante
2202679279260042	Marcelo Soares Pimenta

Fonte: Elaborada pelo autor

6 CONCLUSÃO

O presente trabalho teve como objeto desenvolver uma ferramenta que permita a análise e comparação objetiva e simplificada da produção acadêmica de pesquisadores e grupos de pesquisa do PPGC com a de seus pares tanto do próprio PPGC, quanto externos. Para tanto, foi apresentado o desenvolvimento de uma *web service* API que possibilita a obtenção de índices indicadores de performance da produção científica de pesquisadores e grupos de pesquisa, os quais podem ser facilmente compreendidos e comparados.

6.1 Avaliação dos Resultados

A solução desenvolvida alcançou o objetivo proposto, gerando, de maneira simples e automática, índices numéricos indicadores da qualidade da produção científica de pesquisadores e grupos que são calculados a partir de uma fórmula bem definida e são facilmente comparáveis entre si de maneira objetiva. Entretanto, constatou-se que os dados da plataforma Lattes, sobre os quais a ferramenta calcula seus índices, possuem uma natureza inconsistente. Como resultado, os valores obtidos pela ferramenta, ainda que tendam a se aproximar da realidade, devem ser compreendidos como aproximações, não valores absolutos.

Em relação ao requisito de simplicidade, que faz parte do objetivo do trabalho, a solução apresentada facilita a coleta das informações necessárias à análise através de um método de processamento automático de um currículo Lattes em seu formato XML. Ainda assim, não é oferecida a funcionalidade de coleta automatizada desses currículos a partir da plataforma Lattes, funcionalidade que reforçaria o aspecto da simplicidade na obtenção de indicadores.

6.2 Melhorias

Um ponto de melhoria que aprimoraria a qualidade dos resultados obtidos seria o aprimoramento do processo de correlação entre dados de autores e suas publicações. Implementar heurísticas mais sofisticadas nesse processo permitiria mitigar o efeito da natureza inconsistente das informações sobre os resultados.

Outro aprimoramento seria ampliar a análise da produção científica para mais tipos de publicação, o que exigiria propor novos métodos de classificação para os demais tipos, dado que não são avaliados pelo QUALIS. Também há espaço para a criação de mais índices, levando em conta diversas outras métricas. Por fim, poderia ser implementado também um sistema de autenticação de usuários, para que haja controle sobre quem pode criar, alterar ou excluir dados.

6.3 Trabalhos Futuros

Fica a cargo de trabalhos futuros explorar as melhorias sugeridas na seção anterior; criar interfaces de usuário que utilizem os métodos oferecidos pela API - apresentando a informação obtida por ela de maneira clara e prática - e utilizá-la em outros sistemas. Outros trabalhos também poderiam explorar o uso do *Digital Object Identifier* (DOI) como identificador único para as publicações. Sugere-se também o desenvolvimento de uma solução para coleta automatizada dos dados da plataforma Lattes.

REFERÊNCIAS

- ANDERSON, D. J. **Kanban: Successful Evolutionary Change for Your Technology Business**. [S.l.]: Blue Hole Press, 2010. ISBN 978-0984521401.
- ANGLES, R.; GUTIERREZ, C. Survey of graph database models. **ACM Computing Surveys**, v. 40, n. 1, Fevereiro 2008.
- CAPES. **Classificação da Produção Intelectual**. 2017. Disponível em: <<http://www.capes.gov.br/avaliacao/instrumentos-de-apoio/classificacao-da-producao-intelectual>>. Acessado em: junho de 2017.
- CAPES. **Classificação da Produção Intelectual - Documentos de Área**. 2017. Disponível em: <<http://www.capes.gov.br/avaliacao/instrumentos-de-apoio/documentos-de-area>>. Acessado em: junho de 2017.
- CAPES. **Módulo Coleta de Dados**. 2017. Disponível em: <<http://www.capes.gov.br/avaliacao/plataforma-sucupira/modulo-coleta-de-dados>>. Acessado em: junho de 2017.
- CNPQ. **Portal da Plataforma Lattes**. 2017. Disponível em: <<http://memoria.cnpq.br/web/portal-lattes/sobre-a-plataforma>>. Acessado em: junho de 2017.
- CYPHER. **Neo4j's Graph Query Language: An Introduction to Cypher**. 2017. Disponível em: <<https://neo4j.com/developer/cypher-query-language/>>. Acessado em: junho de 2017.
- DRABACH, R. M. **Sistema de extração e análise de dados acerca da produção científica do PPGC**. 2015. TCC (Graduação) - Curso de Ciência da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2015.
- FIELDING, R. T. **Architectural Styles and the Design of Network-based Software Architectures**. Thesis (Doctoral dissertation) — University of California, Irvine, 2000.
- KUPLICH, C. R. **Desenvolvimento de Data Warehouse e Ferramenta OLAP para a Análise da Produção Acadêmica de Pesquisadores: Estudo de Caso no PPGC**. 2013. TCC (Graduação) - Curso de Ciência da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.
- LIMA, J. L. de O. **Ferramenta para análise da produção científica: estudo de caso no Instituto de Informática da UFRGS**. 2014. TCC (Graduação) - Curso de Ciência da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2014.
- MENA-CHALCO, J. P.; JUNIOR, R. M. C. Scriptlattes: an open-source knowledge extraction system from the lattes platform. **J. Braz. Comp. Soc.**, v. 15, n. 4, p. 31–39, Dezembro 2009.
- NASCIMENTO, C. S. do. **PANDORA - Uma Ferramenta Para Visualização Incremental e Análise de Redes Sociais Acadêmicas**. 2013. Dissertação (Mestrado) - PPGC, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.
- NEO4J. **Neo4j Graph Database**. 2017. Disponível em: <<https://neo4j.com/>>. Acessado em: junho de 2017.

NODEJS. **Node.js**. 2017. Disponível em: <<https://nodejs.org/en/about/>>. Acessado em: junho de 2017.

ORACLE. **Oracle Alta UI Patterns: Master-Detail**. 2001. Disponível em: <<http://www.oracle.com/webfolder/ux/middleware/alta/patterns/MasterDetail.html>>. Acessado em: junho de 2017.

PPGC. **Página do Programa de Pós-Graduação em Computação do Instituto de Informática da UFRGS**. 2017. Disponível em: <<http://www.inf.ufrgs.br/ppgc/institucional/apresentacao/>>. Acessado em: junho de 2017.

RICHARDSON, L.; AMUNDSEN, M.; RUBY, S. **RESTful Web APIs**. [S.l.]: O'Reilly Media, 2013. ISBN 9781449358068.

The Agile Alliance. **Agile Manifesto**. 2001. Disponível em: <<http://agilemanifesto.org>>. Acessado em: junho de 2017.