

UNIVERSIDADE FEDERAL DO PARANÁ

MATTHEUS RODRIGO MARZOLA LEITE

REDER GALLERY: UM PORTAL PARA PUBLICAÇÕES DE
ESTUDOS DE CASO

CURITIBA

2018

MATTHEUS RODRIGO MARZOLA LEITE

REDER GALLERY: UM PORTAL PARA PUBLICAÇÕES DE
ESTUDOS DE CASO

Dissertação apresentada como requisito parcial para obtenção do título de Mestre em Bioinformática, pelo Programa de Pós-Graduação em Bioinformática, Setor de Educação Profissional e Tecnológica, da Universidade Federal do Paraná.

Orientador: Prof. Dr. Mauro Antônio Alves
Castro

CURITIBA

2018

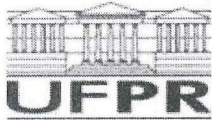
Catálogo na publicação
Sistema de Bibliotecas UFPR
Biblioteca de Educação Profissional e Tecnológica

L533 Leite, Matheus Rodrigo Marzola
RedeR Gallery: um portal para publicações de estudos de caso
[recurso eletrônico] / Matheus Rodrigo Marzola Leite. - Curitiba, 2018.
61 p.: il.

Dissertação (Mestrado) – Universidade Federal do Paraná, Setor de
Educação Profissional e Tecnológica, Curso de Pós-Graduação em
Bioinformática, 2018.
Orientador: Mauro Antônio Alves Castro

1. Sistemas biológicos. 2. Visualização da informação. 3. Ontologia.
4. Redes modulares. 5. Bioinformática. I. Castro, Antônio Alves Castro.
II. Título. III. Universidade Federal do Paraná.

CDD 005.369



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

Pós-Graduação em Bioinformática WWW.BIOINFO.UFPR.BR
E-mail: bioinfo@ufpr.br Tel: 41 33614906

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em BIOINFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **MATTHEUS RODRIGO MARZOLA LEITE** intitulada: "**RedeR Gallery: um portal para publicações de estudos de caso**", após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 08 de maio de 2018.

Dr. Mauro Antonio Alves Castro

Presidente

Programa de Pós-graduação em Bioinformática – UFPR

Dr. Rodrigo Juliani Siqueira Dalmolin

Avaliador Externo ;

Universidade Federal do Rio Grande do Norte - UFRN

Dr. Diéval Guizelini

Avaliador Interno

Programa de Pós-graduação em Bioinformática - UFPR

*Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.*

- Autor desconhecido.

AGRADECIMENTOS

Agradeço à minha família, meus pais e irmãos, por todo apoio durante este período junto ao mestrado, seja com incentivos, ajudas ou momentos de distração, que por vezes ajudam a descansar.

À minha namorada, Victoria Silva Oliveira, por me ser fonte de apoio, incentivo e motivação durante este período, assim como, estar sempre disponível à qualquer momento.

Aos discentes, docentes e técnicos administrativos do Programa de Pós-Graduação em Bioinformática que sempre estiveram à disposição para ajuda, seja ela em questões acadêmicas ou de secretaria. Agradeço às amigas adquiridas na minha turma e no laboratório, em especial aos discentes Kelin, Dionata, Sheyla e Clarice, pela ajuda em dúvidas que eventualmente surgiram, pelas conversas e por sempre serem fonte de inspiração.

Agradeço ao meu orientador, Prof. Dr. Mauro Antônio Alves Castro por todo o apoio, incentivo e todo conhecimento e experiência passados, que serão levados por toda a vida.

E por fim, mas não menos importante, agradeço à PRPPG e aos pró-reitores da gestão anterior e gestão atual, que sempre apoiaram esta empreitada. Aos integrantes da equipe SIGA Jhonny, Marcelo, Nickolas, Alexandre e Felipe, assim como os mais novos integrantes, Jean, Lucas, Mateus e Rafael, por todo o suporte e ajuda nesta caminhada. Agradeço também ao José Henrique Ferreira Pinto por todo incentivo, motivação e muitas vezes pela insistência com que nós, da equipe SIGA, concluíssemos mais este degrau, que foram de suma importância.

RESUMO

Redes complexas se tornaram uma importante ferramenta no que se refere à análise e observação de uma grande quantidade de dados e como eles interagem entre si. Por vezes, essa mesma grande quantidade de dados acaba dificultando a análise e observação dessas redes geradas e, para simplificação da representação, o uso da hierarquia se torna uma importante abordagem. Ferramentas como o RedeR e RedeRWeb tomam proveito desta característica, a hierarquia, seja na representação de redes biológicas ou de redes representando outros tipos de informação. Motivado pelo surgimento recente da ferramenta RedeRWeb e suas características, assim como a ferramenta RedeR, suas características e consolidação na comunidade, este trabalho vem apresentar a RedeR Gallery, uma plataforma de publicação de estudos de caso e cursos, voltados para aprendizagem e colaboração em pesquisas utilizando as ferramentas RedeR e RedeRWeb, visando a ampliação da comunidade de usuários do RedeR, pois diversas áreas podem, e tiram, proveito de estudos e análises de redes complexas. Ao final deste trabalho será apresentado um estudo de caso analisando a rede de interação proteína-proteína do genoma da bactéria *Escherichia coli*, organizando a rede sob a ótica da hierarquia de termos do *Gene Ontology*. Esta organização permite uma ótica global do genoma do organismo, visualizando as interações entre os termos de acordo com os genes anotados no mesmo. E, por fim, todo o fluxo de trabalho do estudo de caso será publicado na RedeR Gallery.

Palavras-chaves: Redes. Biologia de Sistemas. Hierarquia. Ontologia. Redes Biológicas. R. RedeR. RedeRWeb. RedeR Gallery. Bioinformática.

ABSTRACT

Complex networks have become an important tool in analyzing and observing the large amount of data and how they interact with each other. Sometimes this same amount of data makes it difficult to analyze and observe these generated networks and, for simplification of representation, the use of hierarchy becomes an important approach. These tools, such as RedeR and RedeRWeb take advantage of this characteristic, the hierarchy, either in the representation of biological networks or networks representing other types of information. Motivated by the recent emergence of the RedeRWeb tool and its characteristics, as well as the RedeR tool, its characteristics and consolidation in the community, this work presents RedeR Gallery, a platform for publishing case studies and courses, focused on learning and collaboration in research using the RedeR and RedeRWeb tools, with integration with them. At the end of this work, the case study will be presented, analyzing the protein-protein interaction network of the genome of the bacterium *Escherichia coli*, organizing the network from the point of view of the hierarchy of gene Gene Ontology and, finally, the entire workflow case study will be published in RedeR Gallery.

Key-words: Networks. Biology of Systems. Hierarchy. Ontology. Biological Networks. R. RedeR. RedeRWeb. RedeR Gallery. Bioinformatics.

LISTA DE FIGURAS

Figura 1 – As Sete Pontes de Königsberg.	15
Figura 2 – Exemplo de uma rede de mundo pequeno.	17
Figura 3 – Rede de livre escala.	18
Figura 4 – Exemplo de rede modular.	19
Figura 5 – Exemplo de rede hierárquica.	19
Figura 6 – Termos representados hierarquicamente de acordo com o <i>Gene Ontology</i> . .	21
Figura 7 – Exemplo de gráfico setores	22
Figura 8 – Exemplo de gráfico de barras	23
Figura 9 – Interface de comunicação do ambiente R com a interface Java do RedeR . . .	24
Figura 10 – Estrutura hierárquica dos dados no RedeR e RedeRWeb	25
Figura 11 – Exemplo de visualização de uma rede hierárquica no RedeR Web	26
Figura 12 – Fluxograma definindo as etapas de desenvolvimento do projeto	27
Figura 13 – Diagrama de classes	29
Figura 14 – Exemplo de desenvolvimento de JSP	30
Figura 15 – Trecho de uma classe genérica DAO	31
Figura 16 – Exemplo de uso do Bootstrap	32
Figura 17 – Interface de desenvolvimento Eclipse	33
Figura 18 – Modelo físico do banco de dados	34
Figura 19 – Página inicial da RedeRGallery	36
Figura 20 – Rede publicada	37
Figura 21 – Cursos publicados	38
Figura 22 – Informações do curso publicado	38
Figura 23 – Página de cadastro de usuário	39
Figura 24 – Menu de Login E Menu de Usuário	40
Figura 25 – Página de cadastro de uma nova rede	40
Figura 26 – Página de consulta das redes do usuário com sessão ativa	41
Figura 27 – Mensagem de confirmação de exclusão	41
Figura 28 – Página de edição de uma rede	42
Figura 29 – Página de cadastro de um novo curso	42
Figura 30 – Página de edição de um curso existente	43
Figura 31 – Página de cadastro de um novo capítulo	43
Figura 32 – Página de edição de um capítulo	44
Figura 33 – Fluxo do Estudo de Caso	46
Figura 34 – Rede dos termos dos <i>Gene Ontology</i>	48
Figura 35 – Hierarquia dos termos selecionados	49
Figura 36 – Rede PPI dos genes anotados nos termos selecionados	50

Figura 37 – Rede PPI hierárquica dos genes anotados nos termos selecionados	51
Figura 38 – Iniciando o cadastro de um novo curso	52
Figura 39 – Iniciando o cadastro de um novo capítulo	53
Figura 40 – Visualização de um capítulo já cadastrado	54
Figura 41 – Visualização de um curso	55

SUMÁRIO

1	INTRODUÇÃO	12
2	OBJETIVOS	14
2.1	Objetivos Específicos	14
3	REVISÃO BIBLIOGRÁFICA	15
3.1	Redes	15
3.1.1	Teoria dos Grafos	15
3.1.2	Tipos de Redes	16
3.1.3	Redes Biológicas	17
3.1.4	Redes Modulares e Hierárquicas	18
3.2	Gene Ontology	20
3.3	Visualização da Informação	20
3.3.1	Visualização da Informação na Bioinformática	23
3.4	RedeR e RedeRWeb	24
3.4.1	RedeR	24
3.4.2	RedeR Web	25
4	MATERIAIS E MÉTODOS	27
4.1	Desenvolvimento da Ferramenta	27
4.2	MVC	27
4.2.1	Camada de modelo	28
4.2.2	Camada de controle	28
4.2.3	Camada de visão	28
4.3	Java	28
4.4	Java Server Pages (JSP)	29
4.5	Java Persistence API e Hibernate	30
4.6	Jquery	31
4.7	Bootstrap	32
4.8	Eclipse IDE	32
4.9	MariaDB	33
4.10	Git	34
4.11	WildFly 11	35
4.12	R	35
5	APRESENTAÇÃO DA FERRAMENTA	36

5.1	Usuários visitantes	36
5.1.1	Rede publicada	37
5.1.2	Curso publicado	37
5.1.3	Cadastro	39
5.2	Usuários com cadastro	39
5.2.1	Nova Rede	39
5.2.2	Minhas Redes	40
5.2.3	Novo Curso	42
5.2.4	Meus Cursos	44
6	ESTUDO DE CASO	45
6.1	Gene Ontology	45
6.2	Escherichia Coli	45
6.3	Redes Modulares e Hierarquicas	46
6.4	Rede Modular <i>Escherichia Coli</i>	46
6.4.1	Seleção dos genes na rede PPI	47
6.4.2	Rede dos termos do <i>Gene Ontology</i>	47
6.4.3	Selecionando os termos GO na Hierarquia	47
6.4.4	Rede de interações PPI E. Coli	49
6.5	RedeR Gallery	51
6.5.1	Criando um usuário	51
6.5.2	Cadastro de um Novo Curso	51
6.5.3	Cadastro de um Novo Capítulo	52
6.5.4	Resultados	55
7	CONCLUSÃO E PERSPECTIVAS FUTURAS	56
	REFERÊNCIAS BIBLIOGRÁFICAS	58

1 INTRODUÇÃO

Os avanços tecnológicos presenciados nas últimas décadas proporcionaram um grande aumento na capacidade de processamento, armazenamento e análise de grandes quantidade de dados das mais variadas áreas do conhecimento. Esse avanço tecnológico proporcionou uma popularização e evolução dos modelos e técnicas de sequenciamento e, com isso, uma grande quantidade de dados biológicos se acumulou ao longo do tempo, dados como sequenciamentos e anotações de genes, com o Projeto Genoma Humano sendo um grande marco neste período.

Com o grande acúmulo de informações biológicas, novas áreas de pesquisa se tornaram necessárias para analisar e estudar como esses dados estão ligados e como eles interagem, o que é chamado de Era Pós-Genômica. Com isso novas ferramentas para armazenamento desses dados se popularizaram, como por exemplo GenBank, um banco de dados públicos para armazenamento de sequências de nucleotídeos e anotações de genes (BENSON et al., 2013). Assim como o GenBank, diversos outros bancos de dados estão disponíveis, com armazenamento dos mais variados tipos de dados biológicos, como o String DB, um banco de dados de armazenamento de predições e interações de proteínas (SZKLARCZYK et al., 2017). Esta grande quantidade de dados sobre funções e interações de diversos componentes biológicos proporcionou o surgimento da área de Biologia de Sistemas, que visa estudar como esses diversos componentes interagem entre si, entendendo suas funções e influências no organismo, em um nível global (WESTON; HOOD, 2004).

Diversas ferramentas e técnicas para estudar esses elementos surgiram ao longo da era Pós-Genômica, como por exemplo o estudo e análise de redes biológicas, que podem ser desde redes de interações entre proteínas até redes de regulação gênica. No estudo de dados biológicos uma característica importante é o fato de possuírem uma estrutura hierárquica, o que deve ser levado em conta também ao se analisar redes biológicas. Os *softwares* RedeR e RedeRWeb possuem as ferramentas necessárias para analisar e manipular redes biológicas e organizá-las de uma maneira hierárquica e modular.

Neste trabalho será apresentada a ferramenta RedeR Gallery, um portal de publicações de estudos de redes biológicas utilizando as ferramentas RedeR e RedeRWeb, com a proposta de ser um ambiente de colaboração e aprendizado no uso das ferramentas. O RedeR é um pacote R que funciona em conjunto com uma interface Java, onde pesquisadores podem manipular e analisar redes biológicas em uma perspectiva hierárquica, o RedeR é uma ferramenta já consolidada e validada na comunidade e está publicada no repositório do *Bioconductor*, um repositório de pacotes R voltados para análise e manipulação de dados biológicos (BIOCONDUCTOR, 2018). A partir de uma lacuna, no que se refere *softwares web*, o RedeRWeb veio com a proposta de reproduzir a experiência do usuário do RedeR em um ambiente *web*, podendo realizar a

visualização e interação com redes hierárquicas em uma plataforma *online*. Aproveitando-se deste contexto, a RedeR Gallery vem com o objetivo de ser um elo de ligação entre o RedeR e o RedeRWeb, com um ambiente colaborativo e voltado ao compartilhamento de estudos de caso.

No RedeR Gallery o usuário poderá consultar as publicações realizadas por outros pesquisadores, sejam ela estudos de casos simples ou cursos mais elaborados, realizados utilizando o pacote RedeR. Ao realizar seu cadastro, o usuário poderá disponibilizar seu estudo de caso, informando um texto explicativo sobre o estudo, assim como o código na ligação R utilizado naquela análise e a rede gerada como resultado daquela análise. Um usuário com cadastro também poderá disponibilizar no RedeR Gallery um curso ou mini-curso, podendo cadastrar um *workflow* de trabalho completo, podendo cadastrar diversos capítulos e, em cada capítulo, o código em R e a imagem, ou rede, resultada daquela etapa da análise. Com essas características, o RedeR Gallery vem, não só com o intuito de ser um ambiente de colaboração e aprendizado na utilização do pacote RedeR e a plataforma RedeRWeb, mas também uma meio de ampliação da comunidade de usuários, em diferentes áreas de pesquisa, sejam elas biológicas ou não, visto que o estudo de redes complexas esta presente nos mais variados campos da ciência, desde redes sociais, à redes de computadores e redes de colaborações.

Ao final deste trabalho também será apresentado um estudo de caso demonstrando a hierarquia do *Gene Ontology* aplicada à rede de interação entre proteínas da bactéria *Escherichia Coli*. O *Gene Ontology* é uma iniciativa que tem como objeto a padronização de nomenclaturas, denominadas termos, de função e processos moleculares que é dividida em três grandes ontologias: componentes celulares, processos biológicos e funções moleculares. Por possuir uma estrutura hierárquica na sua organização, onde um termo mais específico esta em um nível mais alto e um termo mais genérico esta em um nível mais baixo da hierarquia, o *Gene Ontology* será utilizado como base hierárquica na rede de interações entre proteínas da bactéria *Escherichia Coli*, onde a rede será organizada respeitando a hierarquia imposta pelos termos em que os genes estão presentes. Após a finalização da análise, todo o fluxo de utilizado na ferramenta RedeR será disponibilizada no portal RedeR Gallery no formato de um mini-curso, podendo ser consultado por toda a comunidade.

2 OBJETIVOS

Este trabalho tem como objetivo desenvolver uma ferramenta a ser utilizada como uma plataforma de colaboração e aprendizado, se tornando um ponto de ligação entre a ferramenta RedeR e a plataforma RedeRWeb. Na RedeRGallery o usuário poderá cadastrar seus estudos de redes, com códigos de exemplos na linguagem R e a rede resultado desse código, que será apresentada utilizando a plataforma RedeRWeb. No contexto de colaboração, o usuário da ferramenta apresentada neste trabalho também poderá realizar o cadastro e configuração de cursos e mini-cursos, que poderão ser consultados e utilizados por qualquer usuário, com cadastro ou não na ferramenta. Para alcançar esses objetivos, aqui citados, será necessário atingir alguns objetivos específicos, tais quais serão listados no próximo tópico.

2.1 Objetivos Específicos

Os seguintes objetivos específicos foram estabelecidos neste trabalho:

- Desenvolvimento de uma galeria de publicações de estudos de redes utilizando a linguagem R e a plataforma RedeR. Cada publicação deverá contar com o código fonte R utilizado para gerar a rede do estudo, assim como a própria rede sendo visualizada na plataforma RedeRWeb.
- Desenvolvimento do módulo de cadastro de mini-cursos na galeria de publicações. Cada mini-curso poderá contar com exemplos de código fonte R, assim como visualização de redes.
- Elaboração de um estudo de caso que exemplifique a utilização do módulo de mini-cursos. O estudo de caso deverá demonstrar as características de representação hierárquica de uma rede utilizadas pelas ferramentas RedeR e RedeRWeb.

3 REVISÃO BIBLIOGRÁFICA

3.1 Redes

3.1.1 Teoria dos Grafos

A definição informal de redes, ou grafos, pode ser feita determinando os elementos que as compõem, os nós e as arestas. Onde os nós da rede são elementos do objeto de estudo, como por exemplo cidades, pessoas ou moléculas. Já as arestas, ligações entre os elementos, são as relações que se deseja observar entre os elementos, que podem ser desde um conceito à uma interação física, como por exemplo estradas entre duas cidades, interações entre duas proteínas ou duas pessoas que se conhecem. A análise de redes tem sua origem na Teoria dos Grafos, que se trata de uma disciplina que se dedica a observar características e propriedades inerentes às redes (BONDY; MURTY, 1977).

A Teoria dos Grafos teve início em 1736, com o problema das Sete Pontes de Königsberg, que consistia em definir um trajeto, ou rota, que passasse pelas 7 pontes, que ligavam áreas da cidade de Königsberg, sem com que se repetisse a passagem por nenhuma das pontes (Figura 1).

Figura 1 – As Sete Pontes de Königsberg.



Imagem representando as 4 regiões da cidade (letras A, B, C e D) e as sete pontes numeradas de 1 à 7. Fonte: Adaptado de GoogleMaps (2018).

Ao analisar o problema, o físico e matemático suíço Leonhard Euler (1707-1783) descreveu o que é considerado o primeiro teorema dos grafos. Em seu artigo, Euler propôs como era impossível a solução inicial do problema e, para tal definição, utilizou de propriedades dos grafos para representá-la, onde as quatro áreas da cidade eram os nós e as pontes eram as arestas que

ligavam as áreas. O teorema proposto por Euler define propriedades que avaliam se o caminho é possível ou não, podendo ser aplicado em qualquer rede, independente do número de áreas ou pontes (nós ou arestas) (ESTRADA; KNIGHT, 2015, p. 6). Algumas dessas propriedades são: caso exista mais que duas áreas com um número ímpar de pontes o trajeto é impossível, assim como, caso tenha exatamente duas áreas com um número ímpar de pontes, o trajeto é possível apenas se iniciar em uma delas e terminar na outra (EULER, 1736).

3.1.2 Tipos de Redes

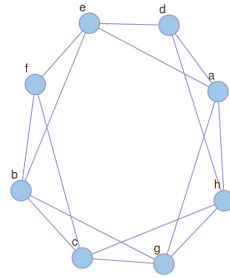
A partir das constatações feitas por Euler, diversos matemáticos deram suas contribuições para este ramo, tais como Arthur Cayley, que fez contribuições na área de estudos de árvores com raízes. Árvores são redes acíclicas, ou seja, não existem arestas formando ciclos, e pode ser definido um único nó como sendo a raiz da árvore. Cayley observou propriedades que definem a quantidade de árvores possíveis dado n nós (ESTRADA; KNIGHT, 2015, p. 8).

Outra importante contribuição parte de Paul Erdős e Alfréd Rényi, que propuseram em seu primeiro artigo um modelo aleatório para a construção de redes (ERDOS; RENYI, 1960). Este modelo, denominado redes aleatórias (ou também redes ER em referência às iniciais dos autores), descreve o crescimento da rede de modo aleatório onde uma rede com N nós recebe K ligações entre eles de forma aleatória (BOCCALETTI et al., 2006). Levando em conta essa característica, de que quanto maior o grau do nó (número de arestas ligadas em dado nó) menor é a chance de receber uma nova ligação, as redes aleatórias são ditas exponenciais (BARABÁSI; BONABEAU, 2003).

As redes aleatórias seguem uma distribuição de Poisson, onde os nós são distribuídos de maneira aleatória e independente. Com isso a rede aleatória tem a característica de que os nós tendem a possuir aproximadamente o mesmo grau e será raramente encontrado um nó que possua um grau muito acima ou abaixo da média (BARABÁSI; BONABEAU, 2003).

Assim como os demais tipos de redes citados acima, existem também as redes *small-world* (Figura 2), ou redes mundo pequeno. As redes de mundo pequeno tem por característica a alta conectividade, onde o menor caminho entre dois nós é sempre composto por poucas ligações (WATTS; STROGATZ, 1998).

Figura 2 – Exemplo de uma rede de mundo pequeno.



Exemplo de uma rede de mundo pequeno (*small-world*), onde a média do caminho entre dois nós é baixa em relação ao número de nós que a rede possui. Adaptado de: Watts e Strogatz (1998).

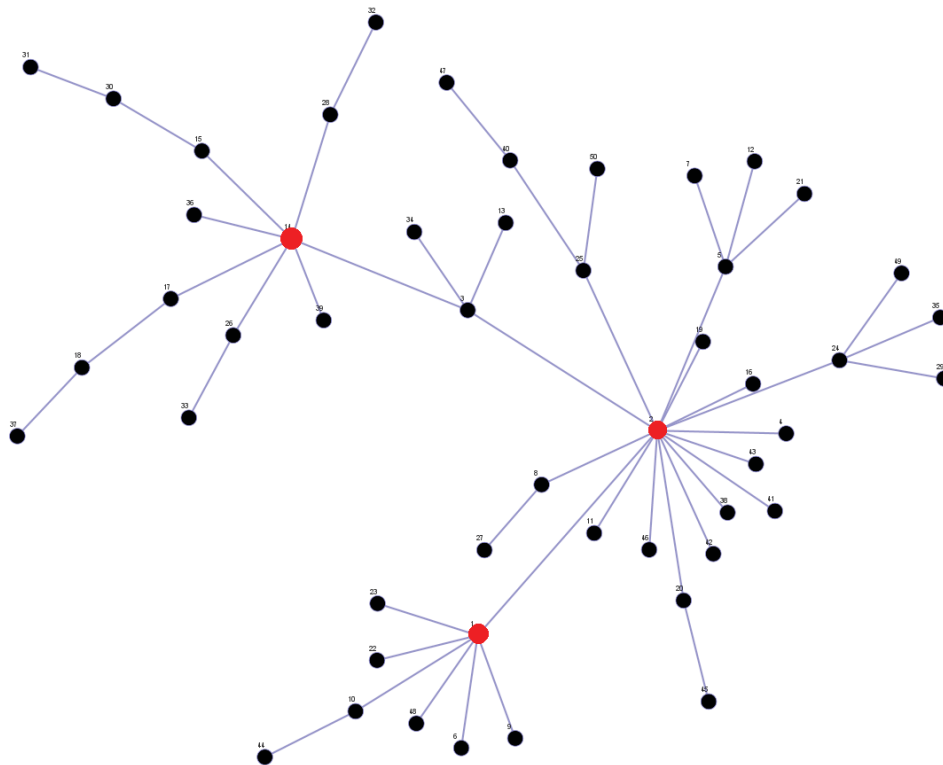
Apesar das diversas características e propriedades estudadas ao longo da história da teoria das redes, constatou-se que essas propriedades não eram suficientes para representar uma característica encontrada em diversas redes presentes em nosso mundo, como redes biológicas e a rede da internet (BARABÁSI; BONABEAU, 2003), e com isso Albert-László Barabási e Eric Bonabeau propuseram um novo modelo que visa explicar como se dá a construção de uma rede: as redes de livres escala (*scale-free*) (Figura 3). Eles observaram, primeiramente ao realizarem um estudo sobre a conectividade entre páginas da internet, onde caso uma página tivesse um *link* com o endereço de outra página então as duas estavam conectadas, a presença de *hubs* (nós com um número de ligações muito acima da média), e com isso constataram que a construção de algumas redes não é feita de modo aleatório, e sim, alguns nós possuem preferência para receber novas ligações (BARABASI; ALBERT, 1999).

Barabási e Bonabeau observaram esta mesma característica também em estudos anteriores à sua publicação. Um estudo sobre os atores de Hollywood, onde se dois atores participaram de um mesmo filme então estavam conectados, também encontrou a presença de *hubs*, no qual alguns atores participavam de mais filmes e, conseqüentemente, iriam obter mais ligações e serem considerados *hubs* (RAVASZ et al., 2002).

3.1.3 Redes Biológicas

Redes tem se mostrado um importante instrumento de análise de redes biológicas, e três tipos de redes biológicas se destacam na utilização desta abordagem: estudo das interações físicas entre proteínas, redes metabólicas e redes de regulação gênica (MASON; VERWOERD, 2007). Redes de interação entre proteínas são utilizadas desde estudos evolucionários e identificação de proteínas associadas à determinada função molecular (FRASER et al., 2002) à observar interação entre proteínas relacionadas ao câncer (KAR; GURSOY; KESKIN, 2009), já redes de regulação, por exemplo, visam observar como proteínas interagem entre si na fase de transcrição gênica (HECKER et al., 2009). Em redes metabólicas, informações de interações entre substratos (nós) através de reações metabólicas (ligações) são utilizadas na representação (JEONG et al., 2000).

Figura 3 – Rede de livre escala.



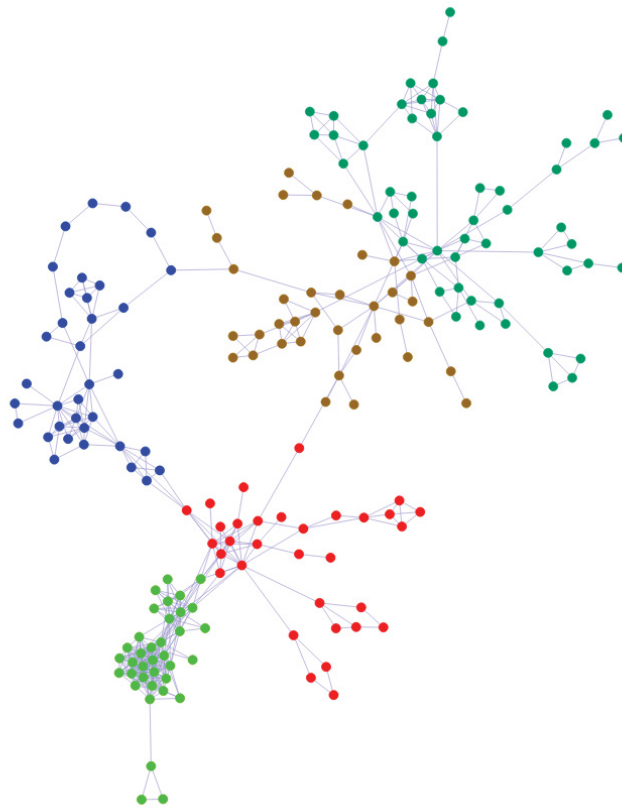
Exemplo de rede de livre escala, onde os nós vermelhos são considerados *hubs*. Adaptado de: Barabasi e Albert (1999).

3.1.4 Redes Modulares e Hierárquicas

Redes biológicas também possuem importantes características em relação à sua topologia, que são o fato de serem modulares e hierárquicas, onde a modularidade pode ser definida utilizando o coeficiente de clusterização da rede e a hierarquia em como esses clusters interagem entre si (RAVASZ; BARABASI, 2002). Essas características são de grande importância para entender como os nós estão organizados em módulos na rede, em como esses módulos interagem, e como isso implica nas funcionalidades da mesma (LUO et al., 2007).

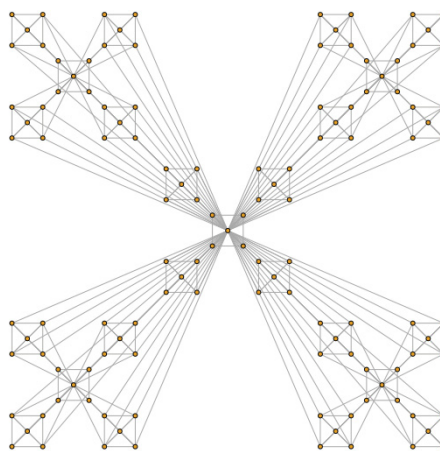
A modularização da rede pode ser medida utilizando o coeficiente de clusterização (RAVASZ; BARABASI, 2002). Este coeficiente leva em consideração informação de um nó i com k_i ligações e pode ser definida pela fórmula $C_i = 2n_i/k_i(k_i - 1)$, onde n_i é o número de ligações entre as ligações k_i e os vizinhos do nó i . (RAVASZ; BARABASI, 2002). A partir do cálculo do coeficiente de clusterização pode-se definir os módulos presentes na rede (Figura 4).

Figura 4 – Exemplo de rede modular.



Exemplo de rede modular, onde é possível observar a presença de módulos sendo diferenciados por cores. Adaptado de: Ravasz et al. (2002).

Figura 5 – Exemplo de rede hierárquica.



Rede representando a estrutura observada em uma rede hierárquica. Adaptado de: Ravasz et al. (2002).

Já uma rede hierárquica é mensurada pela fórmula $C(k) \sim k^{-1}$ (RAVASZ et al., 2002), que mede o índice de clusterização do grau de um nó. Nesta fórmula um nó com poucas ligações, mas pertencente a um modulo, terá um alto C, porém, caso este nó seja um *hub* que conecta este módulo à outros módulos, o C será de valor menor (BARABÁSI; OLTVAI, 2004). Na Figura 5 pode-se observar exemplos de redes hierárquicas.

3.2 Gene Ontology

Com o crescente aumento das informações originadas de anotações gênicas, uma maneira de padronizar a nomenclatura de suas funções, processos em que se envolvem e seus componentes se tornou necessária para uma melhor organização deste conhecimento (ASHBURNER et al., 2000). O consórcio *Gene Ontology* tem por objetivo padronizar as denominações dessas características em 3 grandes ontologias, criando um vocabulário dinâmico que possa ser aplicado à todos eucariotos (ASHBURNER et al., 2000).

O *Gene Ontology* é organizado entre dois grandes componentes, onde o primeiro são as definições das funções gênicas, denominados "termos", que é estruturada de uma maneira hierárquica, onde cada termo pode ser uma definição mais genérica (um nível mais alto da hierarquia) ou uma definição mais específica (um nível mais baixo da hierarquia) (Figura 6). Já o segundo componente é a maneira com que os genes e proteínas são atribuídos à cada termo, ou também chamado de anotação GO, criando um vocabulário padronizado de definições para suas funções, onde as anotações são baseadas em evidências de estudo ligados à publicações da literatura biomédica ou solicitações de atualizações realizadas por diversos pesquisadores, que são revisadas por uma equipe especializada (CARBON et al., 2017).

Esta ontologia de genes faz parte do *Gene Ontology Consortium*, que atualmente consiste, além do *Gene Ontology*, em ferramentas para acessar essa base de conhecimento, consultar e visualizar as anotações e também como os termos se interagem (CARBON et al., 2017). Essa estrutura se torna uma poderosa ferramenta para estudo e análise de dados biológicos, podendo ser utilizada desde, aproveitando sua estrutura, na geração de imagens com alto valor informativo até a inferência de interações proteína-proteína (ZHANG; TANG, 2016).

3.3 Visualização da Informação

Com o crescente aumento da informação acumulada em diversas áreas de pesquisa, desde estudos sociais até pesquisas genômicas, o uso de técnicas de Visualização da Informação (VI) se torna uma importante ferramenta na extração de conhecimento deste grande volume de dados. As técnicas de VI utilizam da capacidade de absorção visual humana para traduzir uma grande quantidade de dados em figuras que facilitem a interpretação da informação, auxiliando assim a absorção do conhecimento (GUIMAR, 2016). O uso das técnicas de VI se baseiam em duas características da visão humana. A primeira é a de que a visão humana é capaz de captar

Figura 6 – Termos representados hierarquicamente de acordo com o *Gene Ontology*

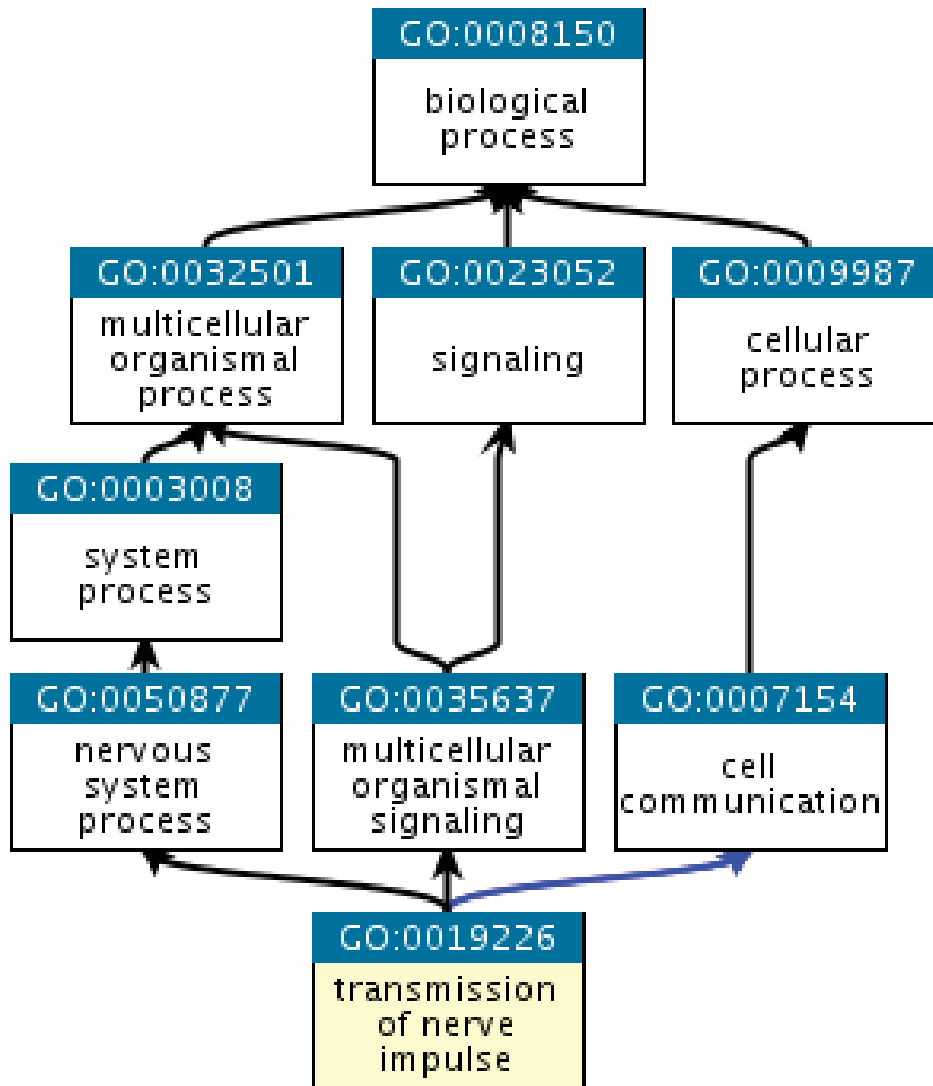


Imagem representando a estrutura hierarquica do *Gene Ontology*, onde o termo *Biological Process* é o termo mais alto na hierarquia e o termo *Transmission of Nerve Impulse* é o termo mais baixo. Fonte: Adaptado de QuickGO (2018).

uma enorme quantidade de informação visual, absorvendo informação de milhões de *pixels* em um único instante, uma quantidade maior até do que a capacidade de assimilação da memória humana. A segunda característica é que a visão humana é altamente eficiente na identificação de padrões, sejam eles como formas, cores, tamanhos e localização (TAO et al., 2004). E assim, a VI junto à técnicas estatísticas, tem o objetivo de comunicar idéias complexas de maneira clara, precisa e eficiente, seguindo alguns preceitos, como por exemplo (TUFTE; GRAVES-MORRIS, 1983):

- Estimular o pensamento do observador para refletir sobre a metodologia e processos utilizados na elaboração da imagem.
- Evitar distorções de interpretações dos dados.
- Encontrar coêrencia em grandes quantidades de informações analisadas.
- Ser intimamente ligada à análises estatísticas e descrições da imagem gerada.

Para tomar proveito dessas características, a VI utiliza de diversas técnicas de organização dos dados para visualização, sejam eles como gráficos de barras, gráficos de setores, gráficos de linhas, *treemap*, dendogramas e grafos (WANG; WANG; ALEXANDER, 2015). Assim como diversas técnicas para a manipulação e interatividades com os dados por parte do usuário, como *zoom*, filtros, e acessos a detalhes (KHAN; KHAN, 2011).

Figura 7 – Exemplo de gráfico setores

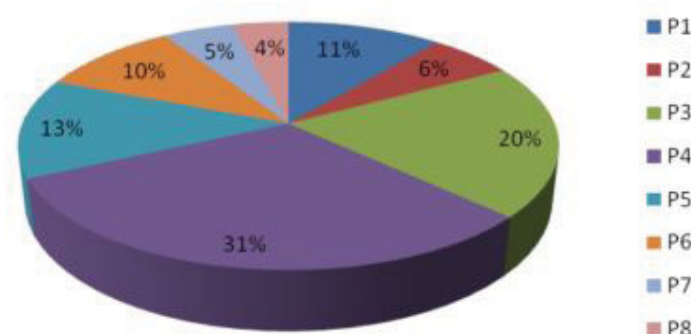


Imagem exemplificando um gráfico *pizza*. Fonte: Khan e Khan (2011).

O gráfico de setores é um gráfico de forma circular, dividido em várias zonas que representam uma parcela dos dados estudados. A totalidade do gráfico representa 100% dos dados, e cada sessão divide os dados em setores correspondentes, representando a porcentagem de um tipo de dados em relação ao conjunto total (Figura 7). A utilização de cores ajuda a evidenciar as classes dos dados e auxilia na utilização de uma legenda (KHAN; KHAN, 2011).

Gráfico em barras é um dos na VI. Sua estrutura é baseada em, no eixo horizontal, estar representado os itens em que se deseja realizar a comparação e, no eixo vertical, a variação dos valores dos objetos sob análise (Figura 8) (KHAN; KHAN, 2011).

Figura 8 – Exemplo de gráfico de barras

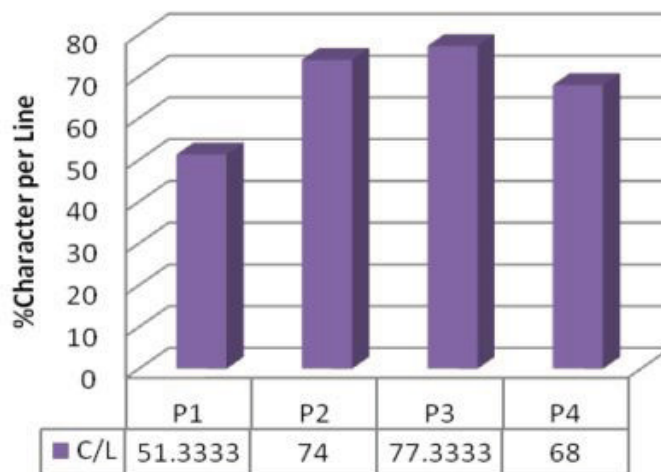


Imagem exemplificando um gráfico barras. Fonte: Khan e Khan (2011).

3.3.1 Visualização da Informação na Bioinformática

Na Bioinformática, como em diversas outras áreas, o uso de VI se torna uma importante ferramenta. Com o grande acúmulo de informações e diferentes tipos de abordagens de análises de dados biológicos, o surgimento de ferramentas que aplicassem técnicas da VI neste contexto se tornou necessária. Dentre essas técnicas utilizadas na Bioinformática, algumas se destacam mais em determinadas áreas, como por exemplo, na área de visualização de genomas. Por ser rico em informações, a visualização e manipulação das informações contidas em genomas se torna de extrema importância na bioinformática, impulsionando o surgimento de ferramentas como o *Integrative Genomics Viewer (IGV)*, ferramenta para visualizar e explorar dados genômicos (THORVALDSDÓTTIR; ROBINSON; MESIROV, 2013).

Um gráfico constantemente usado na Bioinformática é o de dispersão (*scatter plot*), que são gráficos utilizados para avaliar se existe relação entre duas ou mais variáveis (no caso de um gráfico de dispersão de duas ou mais dimensões respectivamente), que pode ser usado para analisar dados de expressão gênica (TAO et al., 2004).

Outra técnica de VI largamente utilizada é a representação em grafos, ou redes. Este tipo de visualização pode ser utilizado para representar as interações entre um grupo de proteínas, assim como, mapas de vias metabólicas. Este tipo de representação facilita a observação e identificação de pontos chaves dessa estrutura, como por exemplo proteínas mais importantes ligadas à determinada função (TAO et al., 2004).

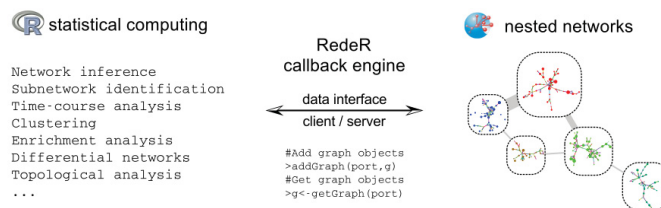
3.4 RedeR e RedeRWeb

Redes biológicas são importantes ferramentas de análise de dados biológicos (MASON; VERWOERD, 2007), com isso, a necessidade de meios para analisar e manipular as interações encontradas em redes biológicas se torna natural e presente. Com essa necessidade, diversos *softwares* e ferramentas foram desenvolvidas ao longo do tempo com o objetivo de suprir esta necessidade, tais como Cytoscape (KOHL; WIESE; WARSCHEID, 2011), CentiBin (JUNKER; KOSCHÜTZKI; SCHREIBER, 2006), Medusa (HOOPER; BORK, 2005), e RedeR (CASTRO et al., 2012).

3.4.1 RedeR

Dados biológicos possuem a característica de apresentarem diferentes interações entre si, que podem ser representadas na forma de redes de interação, como por exemplo redes de interação de proteínas (RAVASZ; BARABASI, 2002). Dados estes que, por sua vez, possuem uma característica importante em estudos biológicos: a hierarquia (BARABÁSI; OLTVAI, 2004).

Figura 9 – Interface de comunicação do ambiente R com a interface Java do RedeR

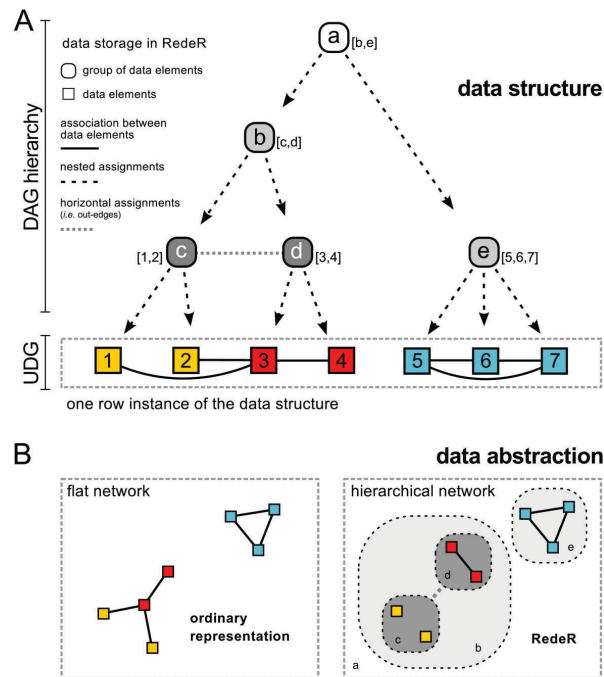


Fonte: Adaptado de Castro et al. (2012).

RedeR é um pacote R integrado com uma aplicação *stand-alone* em Java que é utilizado para manipulação e visualização de redes hierárquicas. Desenvolvido com uma interface simples e uma integração entre o ambiente de desenvolvimento R e a interface Java (Figura 9), o RedeR possui as características para lidar com 3 grandes desafios da análise de redes biológicas: redes biológicas são hierárquicas, métodos estatísticos são necessários para manipulação e visualização de dados e, por último, a necessidade do usuário interagir para ajustar a visualização de dados biológicos relevantes (CASTRO et al., 2012).

Interações que por sua vez podem ser agrupadas em diferentes módulos na rede, como por exemplo proteínas co-reguladoras podem se tornar um módulo na rede e ser representada por um único nó nessa rede. O RedeR leva em consideração esta estrutura hierárquica dos dados para simplificar a visualização de uma rede e focar a representação em informações mais relevantes, o que torna possível escalar esta representação em múltiplos níveis da hierarquia (Figura 10).

Figura 10 – Estrutura hierarquica dos dados no RedeR e RedeRWeb

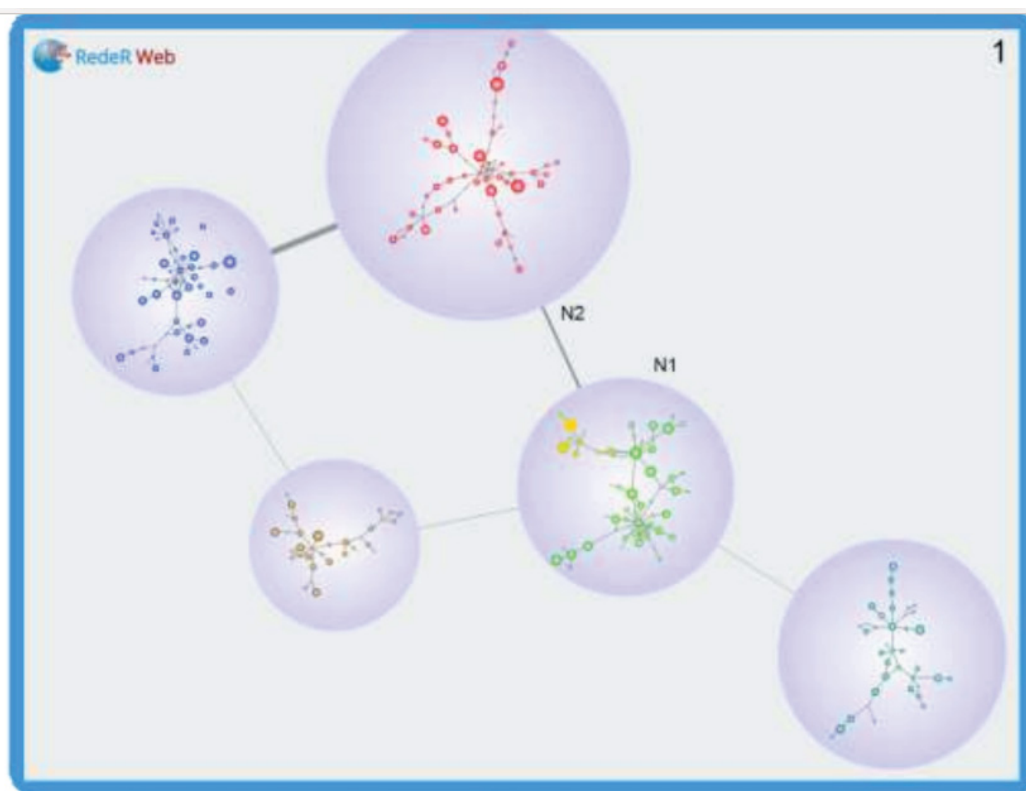


Fonte: Adaptado de Castro et al. (2012).

3.4.2 RedeR Web

Com as características presentes no RedeR e uma lacuna na disponibilidade de ferramentas *web*, observou-se a necessidade do desenvolvimento de uma ferramenta *web* que apresentasse essas características. Com isso o desenvolvimento da ferramenta RedeRWeb para visualização e manipulação de redes hierarquicas na *web* veio para preencher essa lacuna. Assim pesquisadores podem visualizar suas redes sem a necessidade de um ambiente configurado em seu computador, bastando apenas o acesso à internet (Figura 11). Além da facilidade de acesso à visualização e manipulação das redes utilizando uma plataforma *web*, o usuário também pode hospedar sua rede e compartilhar com outros pesquisadores, facilitando o acesso à visualização da rede (SILVA, 2017).

Figura 11 – Exemplo de visualização de uma rede hierarquica no RedeR Web



Fonte: Silva (2017).

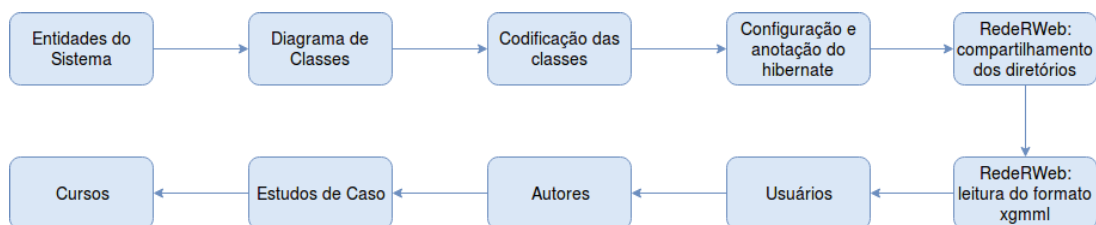
4 MATERIAIS E MÉTODOS

Neste capítulo serão apresentados os materiais e os métodos utilizados durante o desenvolvimento da ferramenta, seja na codificação, persistência dos dados ou hospedagem da ferramenta.

4.1 Desenvolvimento da Ferramenta

O desenvolvimento da ferramenta pode ser dividido em etapas, onde, em cada etapa, uma tecnologia específica é utilizada. Primeiramente foram definidos os atores do sistema, sejam eles os autores, as publicações de estudo de caso e os cursos, que dentro de cada curso pode conter inúmeros capítulos. Após a definição dos atores iniciou o desenvolvimento e codificação das classes do sistema e a criação do banco de dados. Após a criação do banco de dados e da camada de modelo (classes e camada de persistência de dados) foi dado início ao desenvolvimento da camada de visão e dos controladores da aplicação (responsáveis pelas regras de negócio). A partir da definição desses passos, o fluxograma a seguir (Figura 12) foi definido exemplificando o processo de codificação e desenvolvimento do RedeR Gallery.

Figura 12 – Fluxograma definindo as etapas de desenvolvimento do projeto



Fonte: O Autor.

Abaixo serão descritas as tecnologias utilizadas durante os passos definidos no fluxograma, assim como, qual seu papel no projeto e como foi utilizada.

4.2 MVC

O MVC (Model-View-Controller) é um padrão de desenvolvimento de sistemas que visa a modularização do desenvolvimento do *software*, buscando dividi-lo em três camadas: O model, que é a camada onde deverá ser especificadas as lógicas do negócio e a persistência dos dados; a View, a camada de visão onde é desenvolvida a interface que o usuário utilizará para interagir com o sistema; e a Controller, onde deverá ser efetuada a comunicação entre a camada de visão (View) e as regras do negócio (Model) do *software* (ECKSTEIN, 2018). O padrão MVC foi

elaborado pelo cientista da computação Trygve Mikkjel Heyerdahl Reenskaug, que o propôs inicialmente para desenvolvimento de interfaces de usuário, todavia, com a popularização do desenvolvimento de aplicações web, este padrão passou a ser constantemente utilizado para desenvolvimento dessas aplicações (DEV MEDIA, 2018a). Neste trabalho foi utilizado o padrão de desenvolvimento MVC, por ser um padrão que permite a organização e gerenciamento organizado de cada camada do *software*, onde será detalhada cada camada a seguir.

4.2.1 Camada de modelo

Nesta camada foram desenvolvidas as classes que representam os atores do sistema, *User, Network, Course, Author, Chapter*, assim como as regras estabelecidas para cada ator do sistema, como por exemplo: um autor pode publicar nenhum ou vários estudos de caso, porém um estudo de caso deve conter apenas um autor. Por vezes a camada de modelo vem alinhada com as classes responsáveis pela persistência dos dados no banco de dados, assim como o acesso à essa informação posteriormente.

4.2.2 Camada de controle

Esta camada é responsável por receber as requisições realizadas pelo usuário, assim como a validação das informações passadas. Na camada de controle da ferramenta aqui descrita foram desenvolvidas as classes que gerenciaram o cadastro de novos usuários, o acesso a informações dos cursos e publicações, assim como o controle das informações enviadas no momento do cadastro de um novo curso ou de um novo estudo de caso.

4.2.3 Camada de visão

Nesta camada são desenvolvidas as interfaces que o usuário utilizará, sem o meio de interação com o software. Esta camada se comunica com os controladores, descritos no item anterior, enviando requisições e apresentando resultados de requisições realizadas. Nesta camada é onde serão apresentados para o usuários os cursos e estudos de caso de forma visual para que o usuário possa interagir com as informações apresentadas.

Nos próximos itens serão descritos quais tecnologias foram utilizadas em cada camada e também como foram utilizadas.

4.3 Java

Java é uma linguagem de programação orientada a objetos e é difundida mundialmente. Sendo empregue em diferentes áreas, como aplicativos para dispositivos móveis, conteúdo WEB e jogos. Presente em diferentes dispositivos, como computadores, celulares e sistemas embarcados, o Java é uma das linguagens mais utilizadas mundialmente. Uma das grandes vantagens do Java é sua portabilidade. Por ser executada na JVM (Java Virtual Machine) ela não

depende de um ambiente complexo para executar, necessitando apenas da presença da JVM, o que contribuiu para sua popularidade. Atualmente o Java é mantido pela Oracle e está presente em diversos ambientes, desde sistemas corporativos até em dispositivos móveis (JAVA, 2018). O Java foi amplamente utilizado neste projeto, na codificação das classes de controle e modelo, descritas no tópico anterior. As classes de modelo podem ser especificadas utilizando a notação UML (*Unified Modeling Language*), que são modelos padronizados utilizados na documentação de um *software*.

Figura 13 – Diagrama de classes

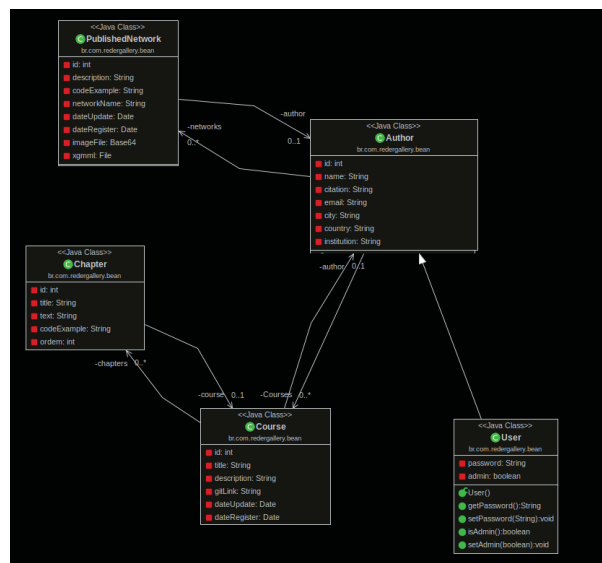


Diagrama de classes representando a estrutura das classes desenvolvidas na camada de modelo. Fonte: o Autor.

4.4 Java Server Pages (JSP)

Lançada inicialmente em 1999 pela Sun Microsystems, o Java Server Pages (JSP) é uma linguagem que veio com o objetivo de adicionar dinamismo na criação de página HTML. Como uma linguagem server-side ela sempre será compilada do lado do web-container, diferentemente de uma página HTML que é interpretada no navegador do usuário, com isso, essa compilação não será visível ao usuário. Por ser uma linguagem de especificações abertas, ela conta com diversos servidores compatíveis, como: GlassFish, WildFly, Tomcat e JBoss. Páginas JSP podem conter tanto conteúdo HTML quanto código escrito na linguagem Java, além de contar com inúmeras bibliotecas de apoio, chamadas de *taglibs*, como por exemplo, taglibs com funções de repetição, como *foreach*, assim como de nós lógicos, como *if/else*. Com isso, páginas JSP tem se mostrado uma alternativa ágil e robusta para a criação de aplicações WEB, por conta de seu dinamismo e a existência de uma vasta biblioteca de apoio (GOMES, 2018). JSPs serão utilizadas na implementação da camada de visão desta ferramenta, camada onde serão desenvolvidas as telas da ferramenta que o usuário utilizará para interagir com a mesma, utilizando em conjunto

tanto HTML, quanto propriedades do CSS (Cascading Style Sheets) e Bootstrap (utilizados na estilização da camada de visão) e bibliotecas próprias do JSF (Figura 14). As telas geradas na camada de visão poderão ser consultadas no capítulo a seguir.

Figura 14 – Exemplo de desenvolvimento de JSP

```

11< <section class="content">
12  <jsp:include page="coursenavigation.jsp" />
13< <div class="container">
14<   <section class="content-header">
15<     <h1>
16<       ${course.title} <small>(Author: ${course.author.name})</small>
17<     <small class="pull-right">Updated in: ${course.dateUpdate}. Registered in: ${course.dateRegister} </small>
18<   </h1>
19<   <br>
20< </section>
21<   <div class="row">
22<     <div class="col-md-8 col-sm-8 col-xs-8">
23<       <div class="box box-primary">
24<         <div class="box-body">
25<           ${course.description}
26<
27<           <c:if test="${course.gitLink != null}">
28<             <label>Link to Gif repository:</label><br/>
29<             <span>${course.gitLink}</span>
30<           </c:if>
31<         </div>
32<         <!-- /.box-body -->
33<         <div class="box-footer">
34<           <small>Author: </small><a href="gallery?o=author&a=${course.author.id}">${course.author.name}</a>
35<         </div>
36<         <!-- box-footer -->
37<       </div>
38<     </div>
39<   </div>
40<
41<   <c:forEach items="${course.chapters}" var="chapter">
42<     <%@ include file="chapter.jsp"%>
43<   </c:forEach>
44<
45< </div>
46< </section>
--

```

Exemplo de um trecho de código utilizado para desenvolver a tela Course, onde são visualizadas as informações de um curso. Fonte: o Autor.

4.5 Java Persistence API e Hibernate

A popularização do Java e sua utilização vem sendo cada vez mais comum em grandes corporações, logo notou-se que grande parte do tempo de desenvolvimento dos programadores era dedicado à implementação das classes de persistência de dados. Além da característica de ser custoso para o programador, alterações de grande porte na camada de persistência de dados é bastante arriscada, o que dificulta grandes migrações que, ocasionalmente, possam a vir ser necessárias em uma corporação (CAELUM, 2018). Neste contexto, a necessidade de criação de uma API (*Application Programming Interface*) para gerenciar a persistência dos objetos Java se tornou evidente e, com isso, se deu o surgimento do Java Persistence API (JPA), que atualmente se encontra na versão 2.0. O JPA é uma API Java que especifica como *frameworks* ORM (Mapeamento Objeto-Relacional, ou do inglês *Object-Relational Mapping*) devem ser implementados e tem como base o JDBC (*Java Database Connectivity*) (DEVVMEDIA, 2018b). O JPA possui diversas implementações, como por exemplo o Hibernate, EclipseLink e OpenJPA. Neste trabalho será empregado o Hibernate, a implementação mais popular do JPA.

O Hibernate é um *framework* ORM que tem como objetivo simplificar para o programador o processo de persistência de dados de um objeto Java em um banco de dados relacional. Atualmente se encontra na versão 5.x e é mantido pela companhia RedHat. Para utilização do Hibernate, o programador pode realizar a configuração tanto utilizando um arquivo XML ou

aplicando anotações nos objetos Java, seguindo um padrão definido na API JPA (REDHAT, 2018). Por conta de suas características e consolidação na comunidade, o Hibernate será utilizado para persistência dos dados neste trabalho, fazendo a ligação entre a camada de modelo e o banco de dados. Para esta ligação foram desenvolvidas 6 classes, reponsáveis pela persistência e recuperação de dados de cada entidade do sistema, que são: AuthorDao, ChapterDao, CourseDao, PublishedNetworkDao e UserDao. Onde todas as classes possuem uma relação de herança com a classe GenericDao (Figura 15), herdando os métodos de listagem, recuperação, edição ou cadastro de sua respectiva entidade.

Figura 15 – Trecho de uma classe genérica DAO

```
3 public abstract class GenericDao<T, I extends Serializable> {
4
5     @Inject
6     protected EntityManager entityManager;
7
8     private Class<T> persistedClass;
9
10    protected GenericDao() {
11    }
12
13    protected GenericDao(Class<T> persistedClass) {
14        this();
15        this.persistedClass = persistedClass;
16    }
17
18    public T save(@Valid T entity) {
19        EntityTransaction t = entityManager.getTransaction();
20        t.begin();
21        entityManager.persist(entity);
22        entityManager.flush();
23        entityManager.clear();
24        t.commit();
25        return entity;
26    }
27
28    public T update(@Valid T entity) {
29        EntityTransaction t = entityManager.getTransaction();
30        t.begin();
31        entityManager.merge(entity);
32        entityManager.flush();
33        entityManager.clear();
34        t.commit();
35        return entity;
36    }
37
38    public void delete(int id) {
39        T entity = get(id);
```

Exemplo de um trecho de código utilizado para desenvolver a tela Course, onde são visualizadas as informações de um curso. Fonte: o Autor.

4.6 Jquery

O Jquery é uma biblioteca em JavaScript que tem como objetivo simplificar a manipulação de elementos HTML em páginas WEB. Atualmente é mantida pela *Jquery Foundation* e se encontra na versão 3.3.1. Originalmente criada por John Resig, teve sua primeira versão lançada em 2006 e hoje se encontra sendo utilizada em aproximadamente 70 milhões de páginas web, incluindo grandes sites como Google, Facebook e Twitter (SIMILARTECH, 2018). Possui uma vasta gama de plugins desenvolvidos utilizando sua tecnologia como base e tem como principais características a redução de código, trabalho com Ajax e possui uma alta compatibilidade entre navegadores web (JQUERY, 2018). O Jquery será utilizado para controlar eventos e realizar validações na camada de visão da ferramenta apresentada neste trabalho.

4.7 Bootstrap

Lançado oficialmente em 2011, o Bootstrap é um framework para desenvolvimento de interfaces baseado em Javascript e CSS. Criado pela empresa Twitter e, mais especificamente, pelos desenvolvedores Mark Otto e Jacob Thornton, atualmente se encontra na sua terceira versão. Em sua atual versão, o Bootstrap possui, por padrão, uma abordagem responsiva, o que o torna uma valiosa ferramenta para desenvolvimento de interfaces na era dos dispositivos móveis (BOOTSTRAP, 2018). Por possuir um dinamismo e ser amplamente empregada no desenvolvimento de interfaces responsivas e intuitivas, esta será a ferramenta utilizada no desenvolvimento do *front-end* da ferramenta. A utilização do Bootstrap se dá durante o desenvolvimento da tela, utilizando JSP, por meio de marcações específicas no atributo *class* de elementos HTML (Figura 16).

Figura 16 – Exemplo de uso do Bootstrap

```
<div class="row">
  <div class="col-md-8 col-sm-8 col-xs-8">
    <div class="box box-solid">
      <div class="box-header with-border">
        <h3 id="chapter${chapter.id}" class="box-title">${chapter.title}</h3>
      </div>
      <div class="box-body">
        ${chapter.text}

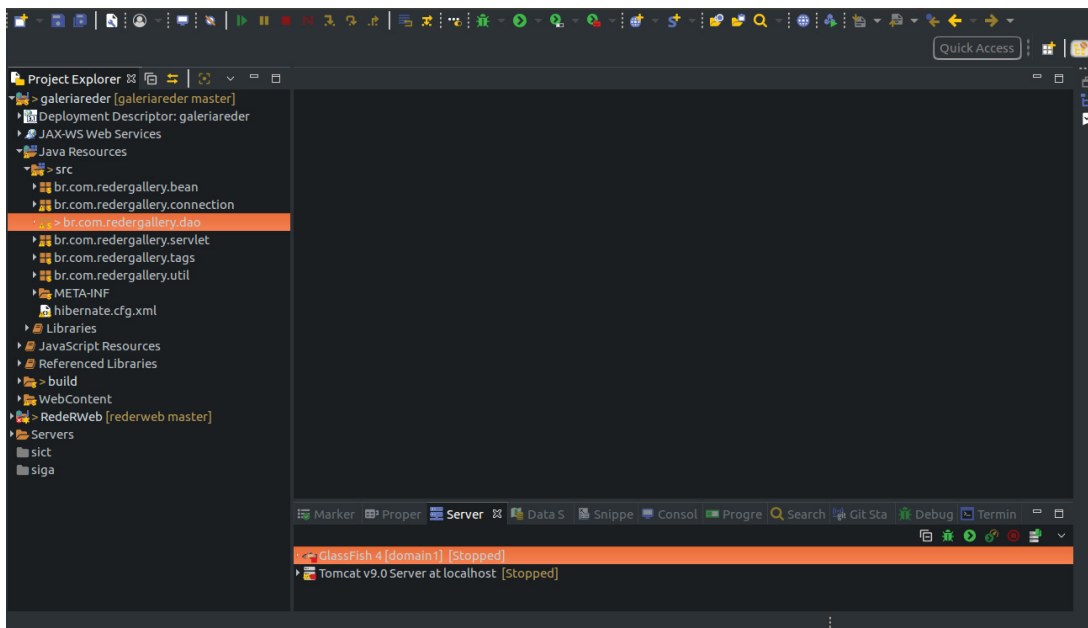
        <hr>
        <div class="col-sm-12 col-sm-12 col-xs-12">
          <pre><code class="r idea hljs">${chapter.codeExample}</code></pre>
        </div>
        <div class="col-sm-12 col-sm-12 col-xs-12">
          <t:ChapterFile idCourse="${course.id}" idChapter="${chapter.id}"/>
        </div>
      </div>
      <!-- box-footer -->
    </div>
  </div>
```

Trecho da codificação de uma JSP com utilização do Bootstrap por meio do atributo *class*. Em destaque os parâmetros "col-md-8 col-sm-8 col-xs-8" são exemplos de utilização do Bootstrap, onde especifica o tamanho do elemento de acordo com o tamanho da do usuário. Fonte: o Autor.

4.8 Eclipse IDE

Eclipse é um Ambiente de Desenvolvimento Integrado (do inglês IDE) feita em java que pode ser utilizada para desenvolvimento de aplicações JEE, JSE e com a utilização de extensões para PHP, Python, entre outras. Projeto iniciado pela IBM em 2001, com o nome de Eclipse Project, seguido da criação da Eclipse Foundation, em 2004, como uma organização independente sem fins lucrativos para atuar como elo com a comunidade (FOUNDATION, 2018a). Pela alta disponibilidade e facilidade de instalação de complementos e por ser uma ferramenta consagrada, esta IDE será amplamente aplicada no desenvolvimento e codificação do software, podendo incluir desde o desenvolvimento da ferramenta, até a integração com o repositório e publicação no servidor de aplicações (Figura 17).

Figura 17 – Interface de desenvolvimento Eclipse



Fonte: o Autor.

4.9 MariaDB

MariaDB é um servidor de banco de dados de código aberto e está entre os mais utilizados atualmente, possuindo grandes empresas em seu repertório, tais como Wikipedia e Google (MARIADB, 2018). O MariaDB é um banco de dados relacional que faz uso de uma interface SQL (do inglês *Structured Query Language*) para acesso aos dados. É mantido pela *MariaDB Foundation* e foi desenvolvido pelos criadores originais do MySQL, atualmente possui grandes nomes em sua lista de colaboradores e doadores, tais como Microsoft e IBM (MARIADB, 2018). Por ser um servidor de banco de dados renomado, *software-livre* e amplamente difundido, o MariaDB será utilizado neste trabalho para armazenamento dos dados e, para isto foi construído um banco de dados seguindo a necessidade implementada nas classes da camada de Modelo (Figura 18).

Figura 18 – Modelo físico do banco de dados

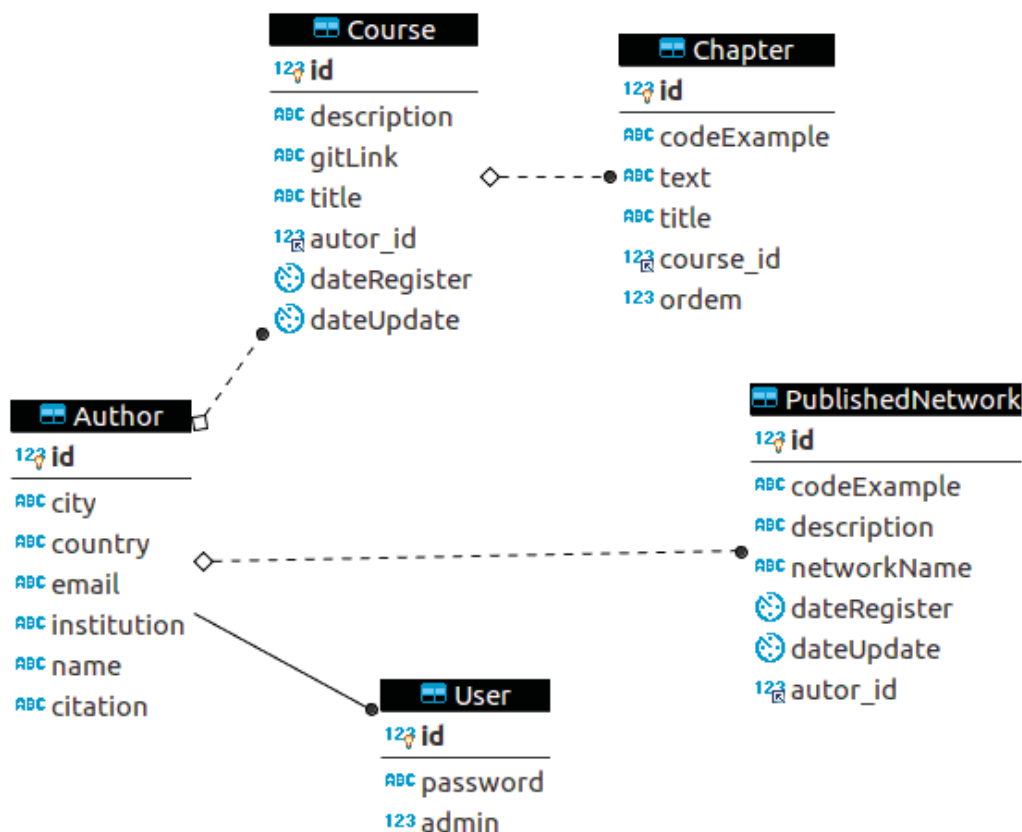


Diagrama especificando a estrutura do banco de dados. Fonte: o Autor.

4.10 Git

O Git é um sistema distribuído de controle de versão gratuito e de código aberto, que é utilizado para gerenciar alterações em um arquivo ou em um conjunto de arquivos ao longo do tempo. Desenvolvido por Linus Torvalds em 2005, criador do Linux, tem como principais objetivos a velocidade, escalabilidade (capacidade de lidar com grandes projetos), ser distribuído, gerenciamento não linear e ser de simples manejo. Uma característica importante de sistemas distribuídos de controle de versão é que cada colaborador do projeto, ao realizar uma cópia dos arquivos, realiza também uma cópia de todo o repositório e seu histórico, sendo um *backup* do próprio repositório (CHACON; STRAUB, 2014). Sistemas de gerenciamento de versão são de suma importância no andamento de um projeto, seja para poder retornar um arquivo a uma versão anterior ou gerenciar alguma alteração que possam vir prejudicar o projeto. O Git tem se tornado referência no controle de versionamento de grandes projetos e é utilizado por grandes empresas, com Microsoft, Twitter, Netflix, entre outras (GIT, 2018). Por ser uma ferramenta consagrada, o Git será amplamente utilizado no gerenciamento do desenvolvimento e versionamento da ferramenta apresentada neste trabalho, sendo utilizado no controle de versão

da codificação do software e do estudo de caso.

4.11 WildFly 11

O WildFly é um servidor de aplicações que segue as especificações do Java EE (do inglês *Java Enterprise Edition*) que é empregue para hospedar aplicações web desenvolvidas na linguagem Java. Inicialmente sob o nome de *JBoss Application Server*, passou a se chamar WildFly em 2013, na versão 8.0 e atualmente se encontra na versão 12.0 (WILDFLY, 2018). Mantido pelo grupo Red Hat, tem como características principais a segurança e a otimização da inicialização do servidor, deixando mais rápido esse processo, o gerenciamento de memória mais eficiente e uma interface de configuração amigável (GROUP, 2018). Por possuir as características citadas anteriormente, o Wildfly será o servidor utilizado na hospedagem do RedeRGallery. Este servidor estará hospedado em um computador com as seguintes configurações:

- Intel(R) Core(TM) i7-7700T CPU @ 2.90GHz
- 2x8gb de memória, 2400 MHz
- HD SATA 1 Terabyte

4.12 R

R é uma linguagem voltada para análises estatísticas e criação de imagens representativas dessas análises. O ambiente R se assemelha ao ambiente da linguagem S, desenvolvida pelo *Bell Laboratories*, e possui uma licença GNU (*General Public License*), estimulando a participação da comunidade de programadores em seu aprimoramento. Inicialmente desenvolvida pelos estatísticos Ross Ihaka e Robert Gentleman na década de 90, hoje o R é mantido pela *The R Foundation* (IBPAD, 2018). No ambiente R, o pesquisador pode utilizar diversas técnicas estatísticas, como regressão linear e não linear, classificações, e também técnicas de representações gráficas, assim como, seu uso pode ser estendido com a criação de pacotes, que são conjuntos de funções destinadas a ampliar as possibilidades de manuseio da ferramenta (FOUNDATION, 2018b). Por ser uma poderosa ferramenta para análises estatísticas e manipulação de dados, tal qual, a origem do pacote RedeR, o ambiente R será amplamente utilizado no estudo de caso deste trabalho, que poderá ser consultado na ferramenta RedeR Gallery.

5 APRESENTAÇÃO DA FERRAMENTA

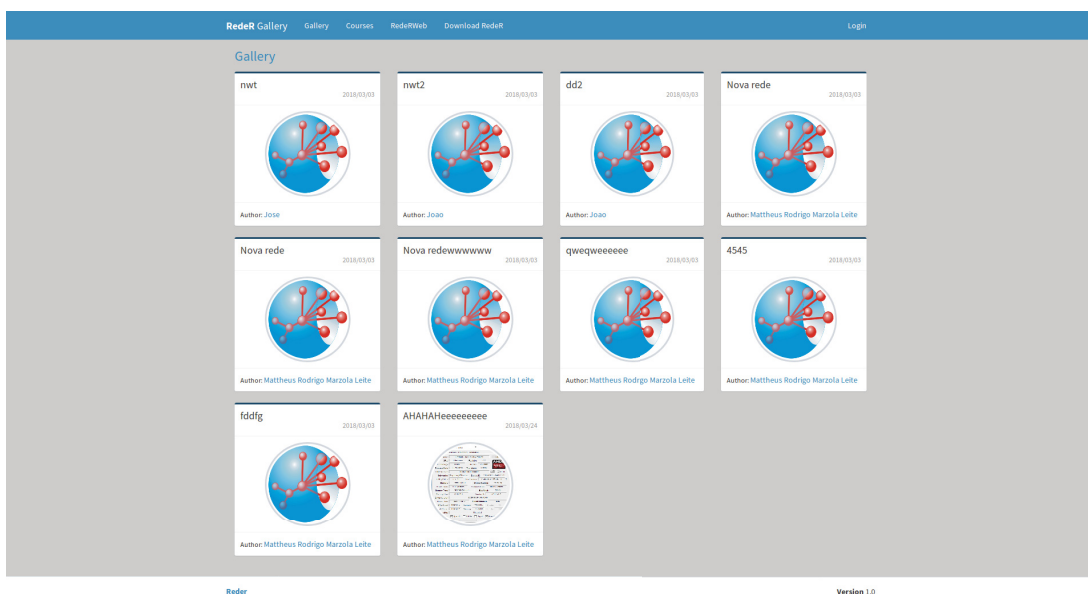
A RedeRGallery vem com o intuito de ser uma plataforma de colaboração em pesquisas relacionadas ao estudo de redes por meio das ferramentas RedeR e RedeRWeb, onde os usuários poderão cadastrar seus estudos de caso, realizar comentários, da mesma maneira que, cadastrar cursos e mini-cursos, sendo o elo de ligação entre as ferramentas RedeR e RedeRWeb. Neste capítulo será realizada a apresentação da ferramenta, exemplificando detalhadamente a utilização de suas funcionalidades.

5.1 Usuários visitantes

Nesta sessão serão apresentadas funcionalidades disponíveis para usuários que não possuem um cadastro na ferramenta. Funcionalidades que são: consultar e acessar as informações de publicações de redes e cursos, assim como, realizar seu cadastro para cadastrar novos conteúdos na ferramenta.

Em seu primeiro acesso à RedeRGallery, o usuário poderá navegar pelo portal ou realizar um cadastro (Figura 24). Para acessar o conteúdo do portal não é necessário que o usuário possua um cadastro, porém, para cadastrar conteúdos ou realizar comentários é necessária a realização do cadastro na ferramenta.

Figura 19 – Página inicial da RedeRGallery

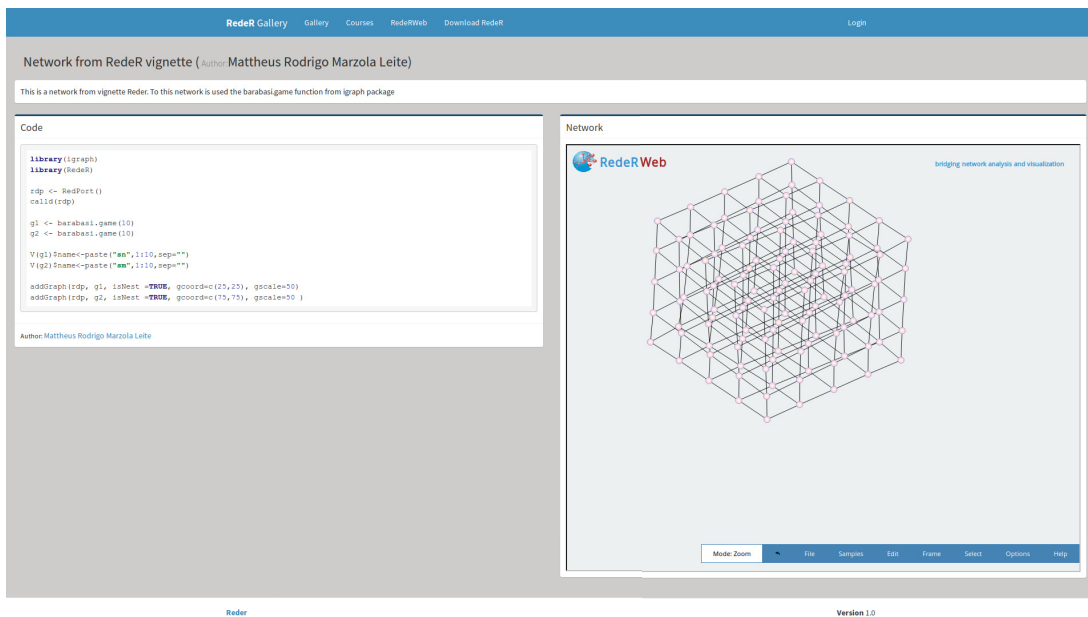


Fonte: O autor.

5.1.1 Rede publicada

Ao selecionar uma publicação de uma rede, clicando sobre o título da publicação ou sobre a imagem de visualização, o visitante será redirecionado para uma página onde serão apresentadas as informações desta rede. Nesta página o usuário poderá interagir com a visualização da rede, utilizando a plataforma RedeRWeb, assim como consultar o código em R utilizado para a geração desta rede e um texto explicativo sobre o que se trata aquele estudo de caso (Figura 20).

Figura 20 – Rede publicada



The screenshot displays the RedeRWeb interface for a published network. The top navigation bar includes 'RedeR Gallery', 'Gallery', 'Courses', 'RedeRWeb', 'Download RedeR', and 'Login'. The main content area is titled 'Network from RedeR vignette (Author: Mattheus Rodrigo Marzola Leite)'. Below the title, there is a brief description: 'This is a network from vignette RedeR. To this network is used the barabasi.game function from igraph package'. The interface is split into two main sections: 'Code' and 'Network'. The 'Code' section contains R code for generating the network, including library imports, function calls, and network generation parameters. The 'Network' section shows a 3D visualization of the network structure, with nodes and edges. The interface also includes a 'Mode Zoom' button and a 'File' menu.

Página mostrando as informações de uma publicação de uma rede. Fonte: O autor.

5.1.2 Curso publicado

Ao selecionar, no menu superior, a opção *Courses*, o usuário será redirecionado para a listagem dos cursos cadastrados (Figura 22), onde poderá selecionar um curso para visualizar as informações.

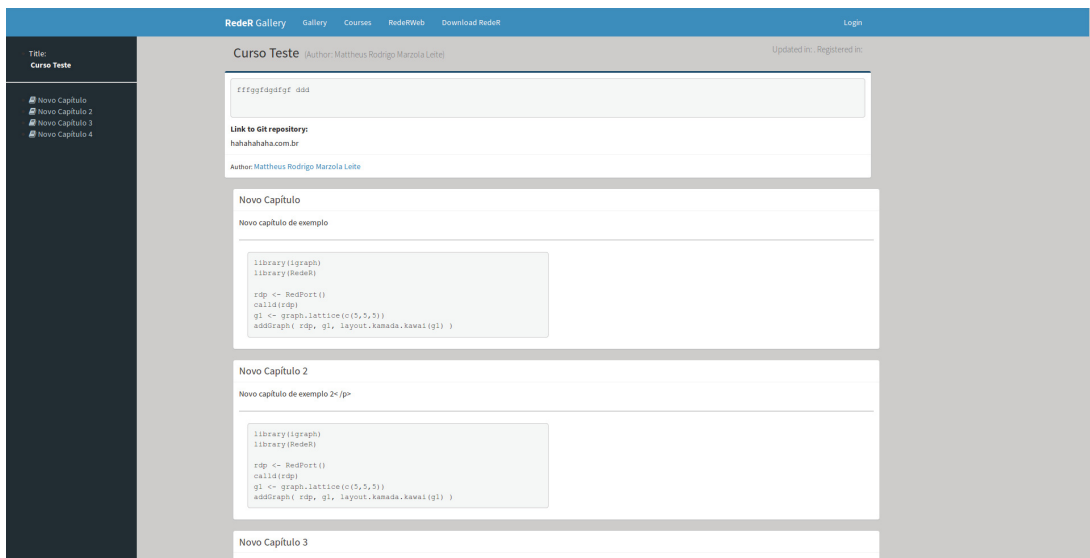
Figura 21 – Cursos publicados



Fonte: O autor.

Clicando na imagem de apresentação do curso, ou em seu título, o visitante será redirecionado para a página de visualização do curso, onde poderá ter acesso à todo seu conteúdo. Nesta página o usuário poderá visualizar todos os capítulos, com o exemplo de código em R e a visualização de seu resultado e, para a navegação no curso, o visitante poderá utilizar o menu lateral esquerdo, onde serão listados todos os capítulos presentes naquele curso.

Figura 22 – Informações do curso publicado

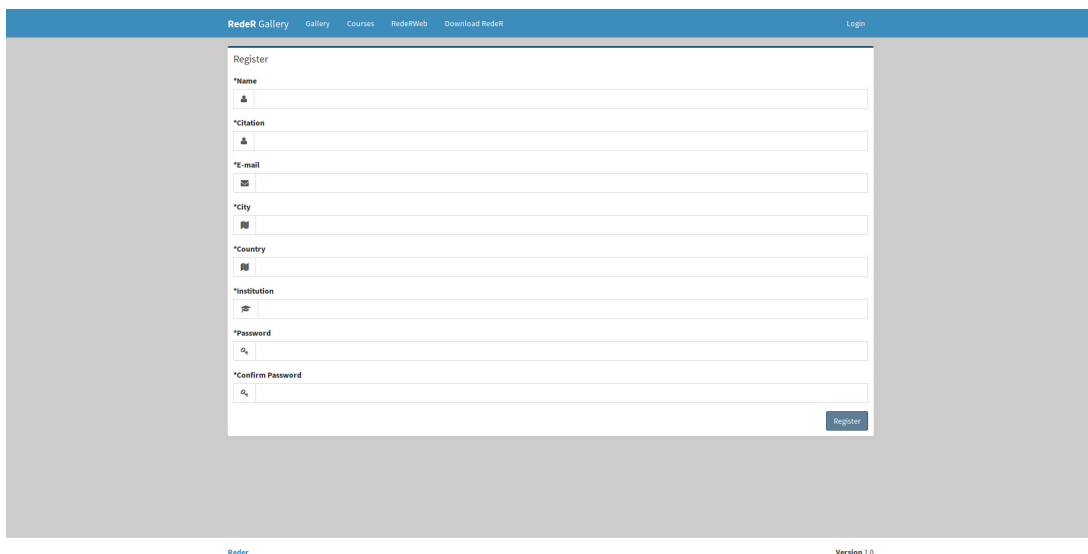


Fonte: O autor.

5.1.3 Cadastro

Para realizar seu cadastro ou iniciar sua sessão, o usuário deve clicar no *menu* superior esquerdo, onde se encontra a palavra *Login*. Ao clicar neste *menu*, será apresentada duas opções: *Login* e *Register* (Figura 24, à esquerda). Selecionando a opção *Register* o sistema irá apresentar a página de cadastro. Na página de cadastro será solicitada algumas informações básicas de cadastro, bem como o país de residência e a instituição ao qual o pesquisador faz parte.

Figura 23 – Página de cadastro de usuário



Fonte: O autor.

Após realizar seu cadastro e iniciar a sessão informando seu *login* e senha, o *menu* superior esquerdo irá sofrer alterações e mostrar as opções disponíveis para o usuário, que são elas: *My Networks*, *New Network*, *My Courses*, *New Course* e *My Perfil* (Figura 24, à direita).

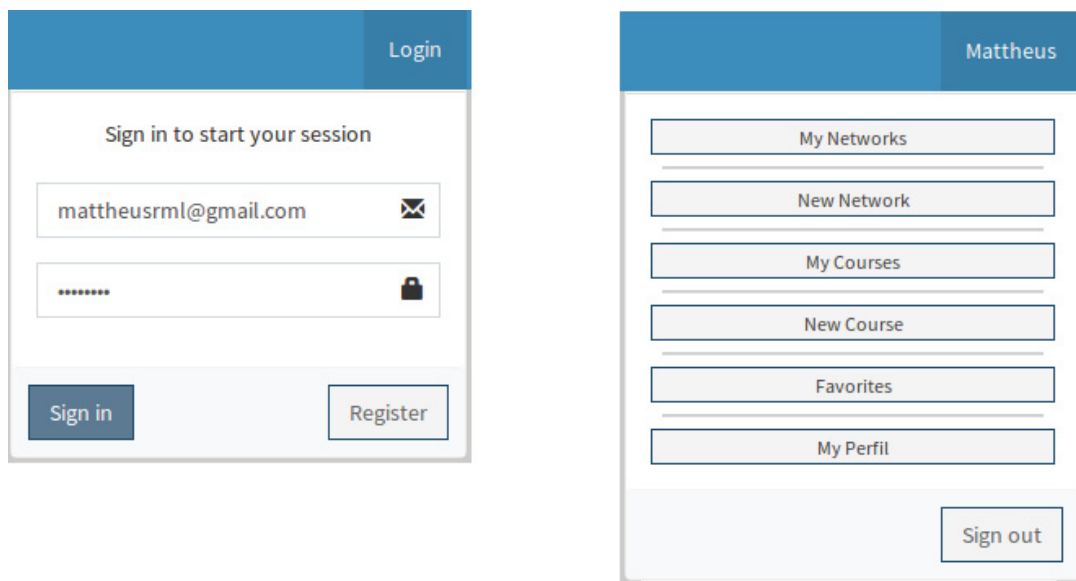
5.2 Usuários com cadastro

A seguir serão apresentadas as funcionalidades disponíveis para usuários que possuem um cadastro ativo na ferramenta, funcionalidade que são: cadastrar, editar ou excluir uma rede sua autoria.

5.2.1 Nova Rede

Ao selecionar, no menu de usuário, a opção *New Network* o usuário será redirecionado para a página de cadastro de uma nova rede, onde informando os dados da rede, como o nome, a descrição do que se trata aquele estudo, o código em na linguagem R, a imagem de apresentação do estudo (caso o usuário não informe esta imagem, será utilizada a padrão do RedeRGallery)

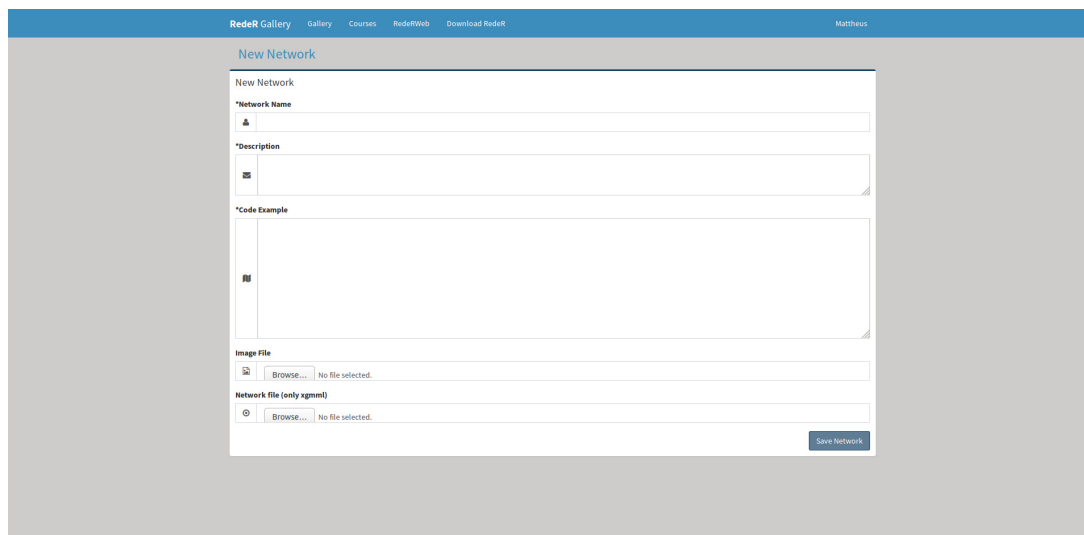
Figura 24 – Menu de Login E Menu de Usuário



À esquerda o menu de login, onde o usuário poderá iniciar a sessão no sistema ou selecionar a opção para realizar seu cadastro. À direita o menu de usuário, onde poderá acessar todas as opções de seu perfil. Fonte: O autor.

(Figura 25). No campo *Network File* deve ser inserida a rede no formato XGMML, logo a mesma poderá ser visualizada utilizando a plataforma RedeRWeb.

Figura 25 – Página de cadastro de uma nova rede



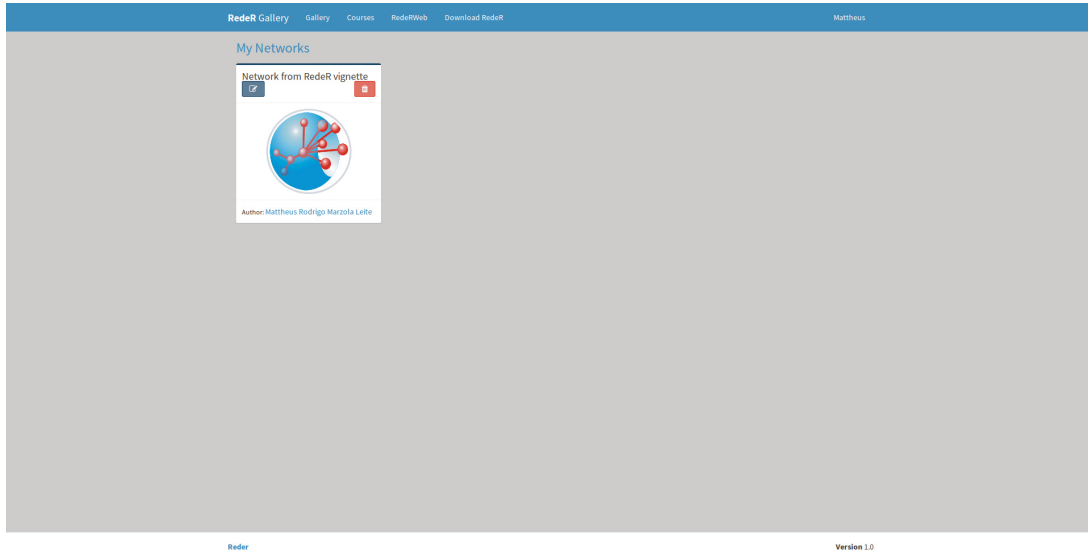
Fonte: O autor.

5.2.2 Minhas Redes

Selecionando, no menu de usuário, a opção *My Networks* o usuário será redirecionado para uma página onde terá listadas as redes que são de sua autoria (Figura 26). Nesta página será

apresentada duas opções para cada rede: deletar a rede, indicada pelo ícone com o desenho de uma lixeira na cor vermelha, e editar a rede, indicada por um ícone na cor azul.

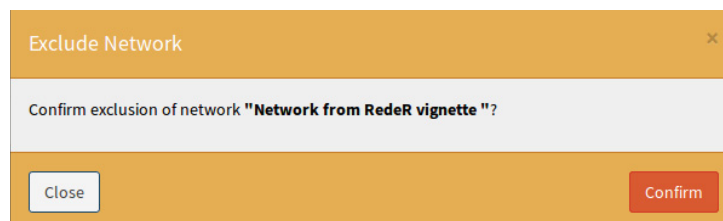
Figura 26 – Página de consulta das redes do usuário com sessão ativa



Fonte: O autor.

Ao clicar na opção de excluir a rede, uma confirmação será solicitada (Figura 27), e apenas após à confirmação a rede será excluída. Clicando na opção de editar, o usuário será redirecionado para a página de edição da rede (Figura 28).

Figura 27 – Mensagem de confirmação de exclusão



Fonte: O autor.

Figura 28 – Página de edição de uma rede

Fonte: O autor.

5.2.3 Novo Curso

Selecionando a opção de cadastrar um novo curso, o usuário será redirecionado para a página de cadastro do curso (Figura 29). Nesta página ele efetuará, inicialmente, o cadastro básico do curso, onde poderá informar o título, a descrição, um *link* para um repositório Git, para caso houver materiais a serem disponibilizados, e uma imagem de apresentação do curso.

Figura 29 – Página de cadastro de um novo curso

Fonte: O autor.

Após realizar o cadastro básico do curso, o usuário será redirecionado para a página de edição do curso, onde poderá editar as informações do curso, assim como adicionar novos

capítulos ou editar capítulos já existentes (Figura 29).

Figura 30 – Página de edição de um curso existente

The screenshot shows the 'New Course' page in the RedeR application. The page title is 'Curso Teste'. The form includes the following fields and content:

- Course Title:** Curso Teste
- Description:** Apenas um curso teste de apresentação
- GitHub:** hahahahaha.com.br
- Cover Image:** A globe icon with red nodes.

A 'Save' button is located at the bottom right of the form. The left sidebar shows a 'New Chapter' button. The footer indicates 'RedeR Version 1.0'.

Fonte: O autor.

Na página de editar curso (Figura 30), o usuário terá acesso à opção de cadastrar novos capítulos para este curso e, para isto, basta selecionar a opção *New Chapter*, no menu lateral esquerdo.

Figura 31 – Página de cadastro de um novo capítulo

The screenshot shows the 'New Chapter' page in the RedeR application. The page title is 'New Chapter'. The form includes the following fields and content:

- Title:** (empty)
- Text:** (empty)
- Code Example:** (empty)
- Network file (jpg or png):** (empty)

A 'Save' button is located at the bottom right of the form. The left sidebar shows a 'New Chapter' button. The footer indicates 'RedeR Version 1.0'.

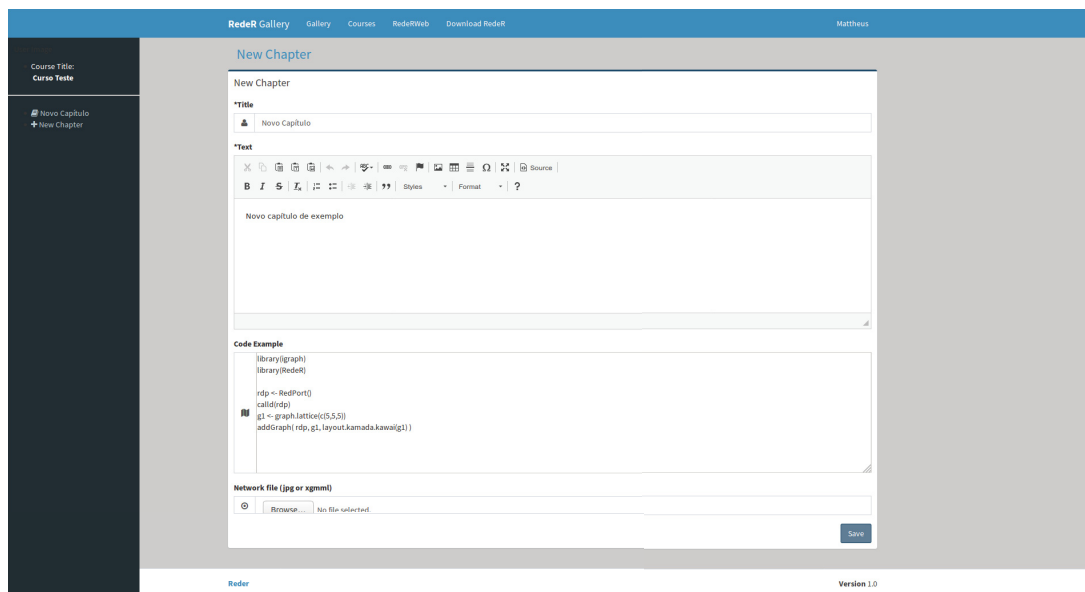
Fonte: O autor.

Ao selecionar esta opção, será apresentada a página de cadastro de um novo capítulo (Figura 31), onde poderá ser informado o título do capítulo, o texto, a código de exemplo

da linguagem R e um arquivo de uma imagem ou uma rede, no formato XGMML, que será apresentado como o resultado do código em R informado.

Após cadastrar um novo capítulo, o mesmo será listado no menu lateral esquerdo. Selecionando um capítulo, o usuário poderá realizar alterações e atualizações nas informações contidas no mesmo (Figura 32).

Figura 32 – Página de edição de um capítulo



Fonte: O autor.

5.2.4 Meus Cursos

Ao selecionar a opção *My Courses* o usuário será direcionado para uma página onde será listado todos os cursos que ele é autor. Nesta página, assim como na página *My Networks*, ele terá duas opções: excluir um curso existente ou editar um curso existente. Ao selecionar a opção excluir, será solicitada uma confirmação de exclusão e, ao selecionar a opção de edição de um curso, o usuário será redirecionado para a página de edição de curso, podendo cadastrar novos capítulos ou editar os existentes (Figura 30).

6 ESTUDO DE CASO

Redes biológicas possuem a característica de serem hierárquicas e explorar esta característica é de suma importância na sua análise e interpretação dos dados (LUO et al., 2007). Para exemplificar o uso desta estrutura de hierarquia, neste estudo de caso será demonstrada uma representação hierárquica da rede de interação entre proteínas (redes PPI) presentes no genoma da bactéria *Escherichia coli strain K12* e para isso será utilizado o pacote de anotação *org.EcK12.eg.db* da ferramenta R (CARLSON, 2017b). Para representação da hierarquia desta rede será empregada a estrutura do Gene Ontology (ASHBURNER et al., 2000), se aproveitando de sua estrutura e confiabilidade dos dados e, para isso, será empregue o pacote *GO.db* (CARLSON, 2017a), que possui todas as informações necessárias para manipular os dados do Gene Ontology. Após o processo de tratamento dos dados e visualização da informação, todo o fluxo deste estudo de caso será cadastrado como um curso, por meio da ferramenta RedeR Gallery.

6.1 Gene Ontology

Como descrito na seção 3.2, o Gene Ontology (GO) é um projeto com o objetivo de padronização de nomenclaturas e anotações funcionais de genes, tal qual, uma representação computacional deste conhecimento. Atualmente estão representadas mais de 40 mil conceito biológicos, que são constantemente atualizados e revisados (CONSORTIUM, 2018). Neste estudo de caso, o GO será utilizado na representação hierárquica da rede de interação da bactéria *Escherichia Coli*, atribuindo à cada nível da hierarquia (termos) seus respectivos genes anotados.

6.2 Escherichia Coli

A *Escheriachia Coli* (conhecida também como *E Coli*) é uma bactéria comumente encontrada na flora intestinal da maioria dos animais, incluindo os humanos. A *Escheriachia Coli* também possui estirpes responsáveis por doenças intestinais, tais como infecções do trato urinário e diarreia. Neste estudo de caso será utilizada as proteínas anotadas da *Escheriachia Coli* da estirpe k-12, onde em sua primeira anotação continha 4.639.221 pares de base, dos quais codificam 4288 proteínas (BLATTNER et al., 1997). Para o manuseio dos dados de anotação da bactéria *Escheriachia Coli* será utilizado o pacote (CARLSON, 2017b), onde é possível acessar as informações referentes aos genes desta estirpe, assim como as informações de anotação destes genes à seus respectivos termos.

6.3 Redes Modulares e Hierárquicas

Neste capítulo iremos apresentar a modelagem e criação de uma rede modular e hierárquica utilizando o ambiente de desenvolvimento R e o pacote RedeR (BLATTNER et al., 1997), onde será representada a rede de interação proteína-proteína da bactéria *Escheriachia Coli* e, aplicando as informações de anotações do GO, a rede será estruturada de maneira modular e hierárquica. Para criação e manipulação dos objetos e dados referentes à rede no ambiente R será utilizado o pacote *igraph* (CSARDI; NEPUSZ, 2006). Ao final deste processo, será disponibilizada, na ferramenta RedeRGallery, todo o fluxo de desenvolvimento e análise, no formato de um curso.

6.4 Rede Modular *Escherichia Coli*

O objetivo deste estudo de caso é demonstrar as vantagens na visualização e análise de redes ao se utilizar de uma estrutura hierárquica para representação dos dados a serem estudados. O estudo aqui presente se deu de acordo com a seguinte abordagem: será demonstrada a hierarquia dos termos que estão presentes nas anotações das proteínas da *Escheriachia Coli*. Uma vez visualizada essa hierarquia é possível selecionar os termos mais relevantes e organizar a rede de proteínas da *Escheriachia Coli* conforme a estrutura dos termos selecionados. O fluxo utilizado neste estudo de caso esta exemplificado na imagem abaixo (Figura 33).

Figura 33 – Fluxo do Estudo de Caso

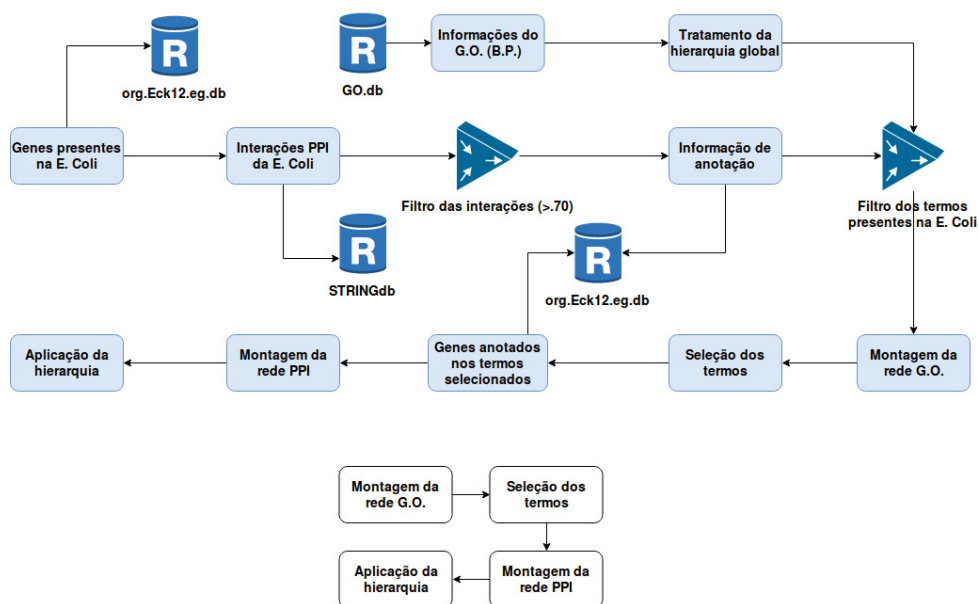


Imagem definindo o fluxo do estudo de caso e todas as etapas presentes no mesmo. Onde as quatro etapa separadas abaixo definem os passos principais. Fonte: o Autor.

Ao longo dos próximos tópicos será detalhado como foi realizado o estudo, desde o

tratamento das informações, até a aplicação da hierarquia na rede de interação proteína-proteína, seguindo os quatro passos destacados na imagem acima (Figura 33).

6.4.1 Seleção dos genes na rede PPI

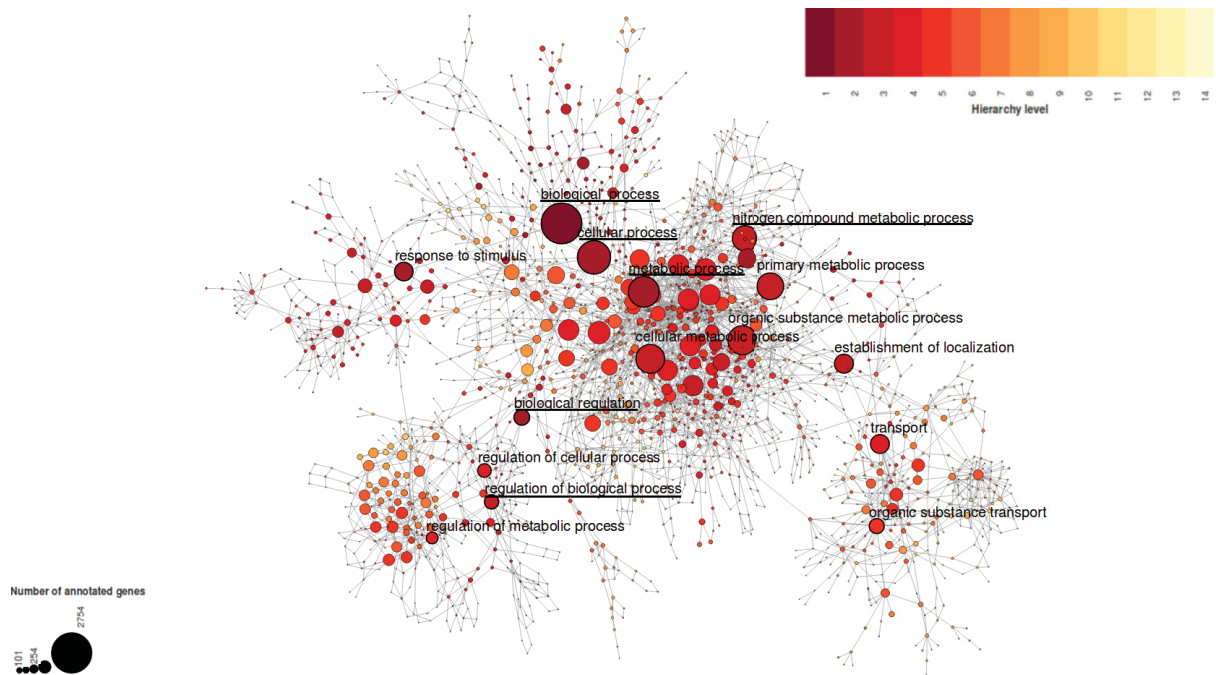
Nesta etapa do estudo de caso, recuperamos a lista de genes encontrados no genoma da *Escherichia Coli* utilizando o objeto `org.EcK12.egSYMBOL`, do pacote `org.EcK12.eg.db`. Após recuperado os genes, foi empregue o pacote `STRINGdb`, para acessar as informações referentes às interações entre as proteínas. Uma vez carregada as informações referentes às interações, foi aplicado um filtro sobre o valor de *Combined Score*, que é considerado relevante para valores > 700 . O *Combined Score* é um valor estatístico originado da análise de sete evidências utilizadas para definir cada interação, que são elas: evidência experimental, evidências de banco de dados curados, mineração textual, evidência de coexpressão gênica, proximidade no genoma, coocorrência e proteínas ortólogas. Apesar de > 700 ser considerado um valor padrão de corte, neste estudo de caso foi utilizado o valor > 900 como corte, com isso, são descartadas interações que possuem um *Combined Score* abaixo deste valor e, conseqüentemente, alguns genes serão excluídos da rede de interações.

6.4.2 Rede dos termos do *Gene Ontology*

Uma vez estipulados os genes relevantes na rede de interações PPI, foi dado início à montagem da rede referente aos termos do *Gene Ontology* onde, se um termo é pai de outro termo, então é atribuída uma ligação direcional entre os dois termos. À partir dos genes restantes, descritos no tópico anterior, foram selecionados os termos aos quais estes genes pertencem, com as informações contidas no objeto `org.EcK12.egGO2ALLEGS`, do pacote `org.EcK12.eg.db`, e, no final deste processo, os termos que restaram sem nenhum gene atribuído e os genes que não estavam anotados em nenhum termo foram descartados. Selecionados os termos, foi contabilizada a quantidade de genes presentes em cada termo, refletindo assim no tamanho e na cor do nó que representa este termo. Com isto, a rede dos termos do *Gene Ontology* pode ser montada (Figura 34).

6.4.3 Selecionando os termos GO na Hierarquia

Uma vez representada a hierarquia dos termos GO utilizando a rede, é possível observar os termos mais relevantes. Uma característica presente nesta rede é que os nós mais centrais da rede possuem um tamanho maior em relação com o restante da rede, representando a grande quantidade de genes concentrados nestes termos. Isto pode ser explicado por, os termos que se encontram na região central da rede, serem termos localizados em um nível mais alto da hierarquia do GO e, por ser um termo mais genérico, possua mais genes anotados. Outra característica observada é a existência de nós localizados nas periferias da rede e que possuem um tamanho e colocação de destaque, podendo indicar um termo relevante nesta rede, pois,

Figura 34 – Rede dos termos dos *Gene Ontology*

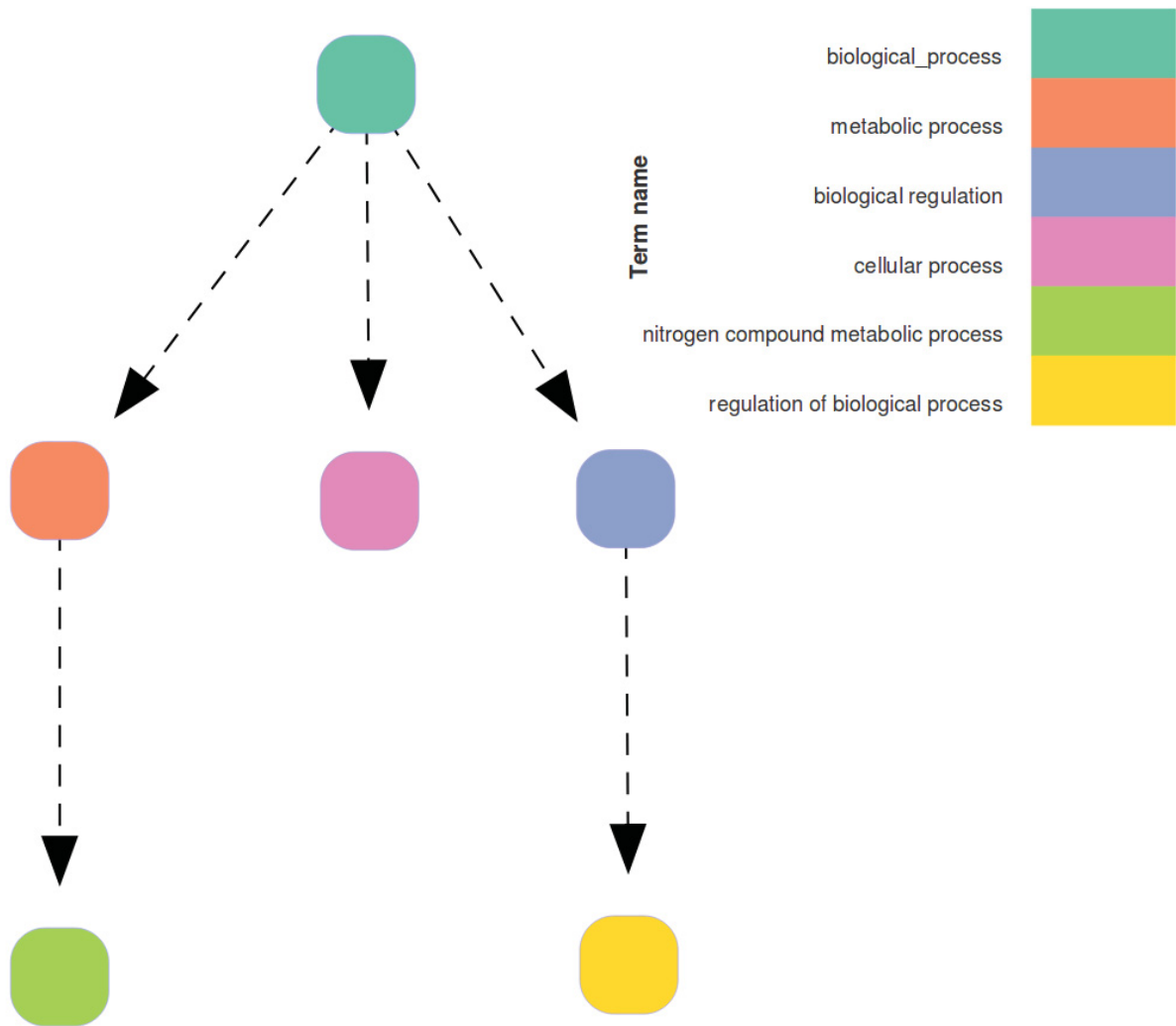
Rede representando a hierarquia do *Gene Ontology*, onde o nó representa cada termo e a ligação a hierarquia presente naquele termo. Fonte: o Autor.

mesmo sendo um termo mais específico, ainda assim, possui uma quantidade considerável de genes anotados. Neste estudo de caso foi utilizada esta abordagem para visualizar a rede de anotação do genoma completo da *Escheriachia Coli*, porém a mesma técnica pode ser utilizada para visualizar a rede de uma família de genes específica ou até mesmo genes selecionados que estão envolvidos diretamente à algum processo celular.

O próximo passo foi a seleção de termos aos quais deseja utilizar para aplicar a hierarquia na rede PPI da *Escheriachia Coli*. Observando a rede, foram selecionados alguns termos centrais, assim como alguns termos periféricos considerados relevantes. Para a representação da hierarquia foram selecionados os seguintes termos e, a partir destes termos, montada a hierarquia que será utilizada (Figura 35):

1. Biological Process
2. Cellular Process
3. Metabolic Process
4. Nitrogen Compound Metabolic Process
5. Biological Regulation
6. Regulation of Biological Process

Figura 35 – Hierarquia dos termos selecionados

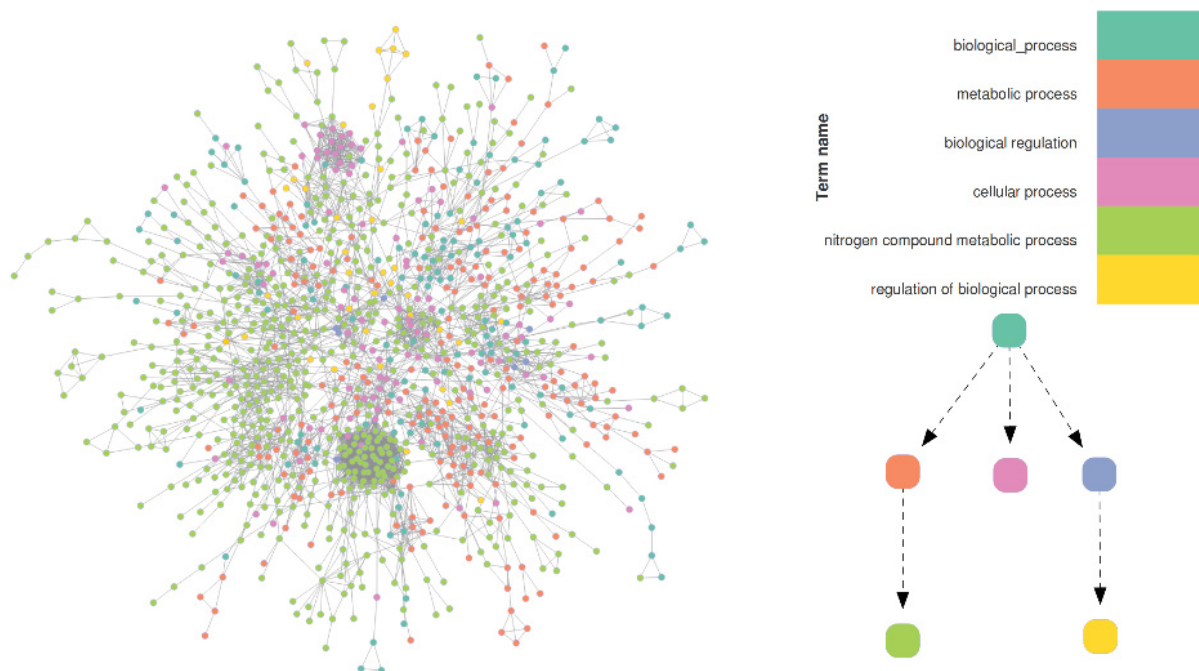


Rede representando a hierarquia selecionada na rede do *Gene Ontology*. Fonte: o Autor.

6.4.4 Rede de interações PPI E. Coli

A partir da seleção dos termos que serão utilizados na hierarquica, foi efetuada a filtra-gems dos genes que se encontram nesses termos, ou seja, a rede de interações PPI da *Escheriachia Coli* refletirá os termos selecionados, onde irá conter apenas os genes que se encontram anotados nestes termos. Nesta rede foi atribuída à cor de cada gene ao termo ao qual está anotado . A partir da formação da rede, foi selecionado apenas o módulo central, excluindo as redes adjacentes formadas (que não possuam ligação com o módulo central) (Figura 36).

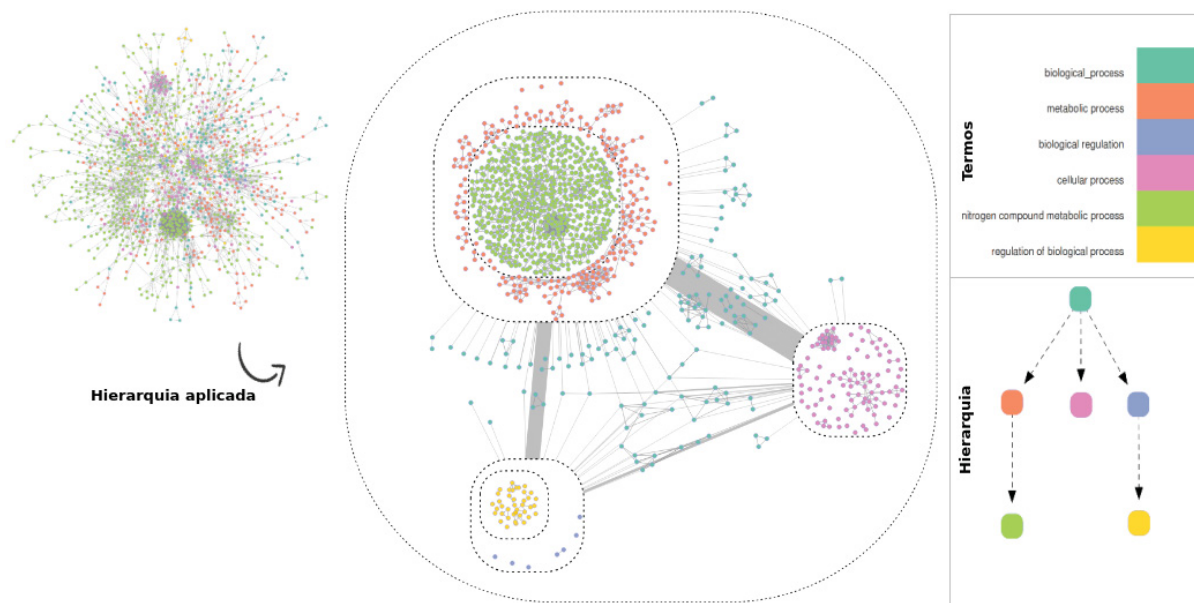
Figura 36 – Rede PPI dos genes anotados nos termos selecionados



Rede de interação PPI dos genes anotados nos termos selecionados, onde a cor do gene representa o termo em que está anotado. Fonte: o Autor.

Com a rede de interações PPI e os termos representando a hierarquia, utilizou-se o pacote RedeR para aplicar a estrutura hierárquica dos termos na rede, obtendo-se assim, a rede a seguir (Figura 37).

Figura 37 – Rede PPI hierárquica dos genes anotados nos termos selecionados



Rede PPI hierárquica dos genes anotados nos termos selecionados. Cada termo é representado como um módulo da rede, que está disposta de maneira hierárquica. Fonte: o Autor.

6.5 RedeR Gallery

Uma vez finalizado todo o processo de análise e construção da rede hierárquica de interações proteína-proteína da *Escherichia Coli*, o fluxo desenvolvido será disponibilizado na plataforma RedeR Gallery, onde poderá ser consultado pela comunidade.

6.5.1 Criando um usuário

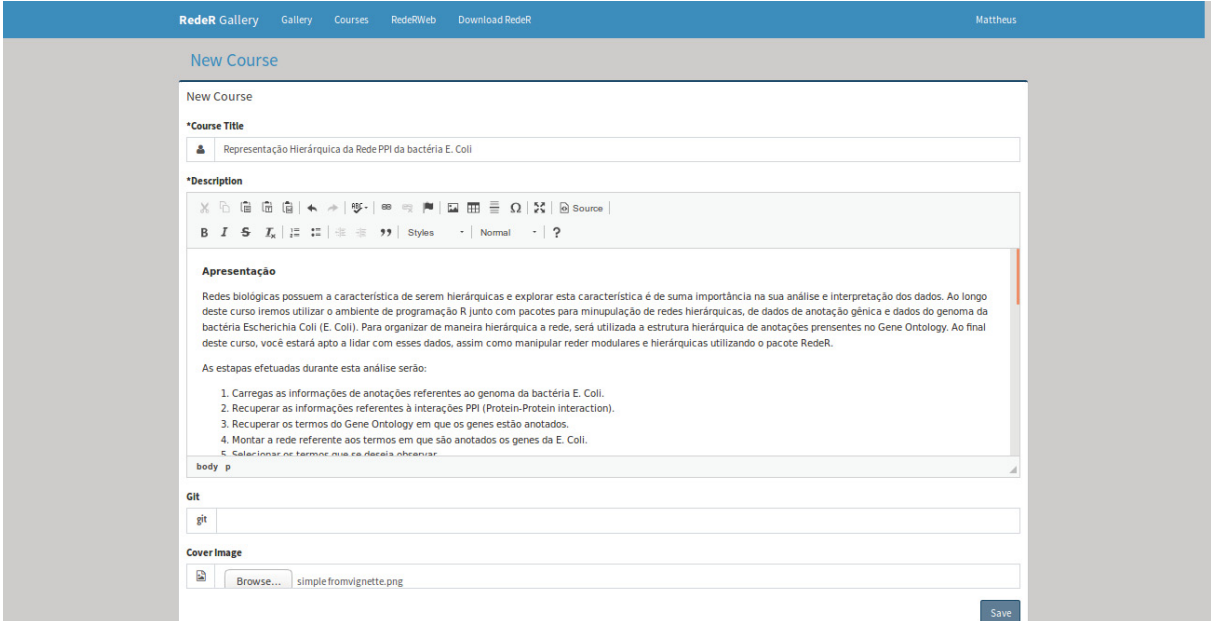
O primeiro passo para cadastrar e disponibilizar um novo curso para a comunidade é efetuar o cadastro do usuário e, para isso, basta acessar a página inicial, selecionar a opção *Login* que se encontra no canto superior esquerdo e, posteriormente, selecionar a opção *Register*. Uma vez selecionada esta opção, o sistema apresentará a página de cadastro, descrita na subseção 5.1.3.

6.5.2 Cadastro de um Novo Curso

Uma vez efetuado o cadastro, o usuário poderá iniciar sua sessão e assim cadastrar o curso desejado. Para isto basta selecionar novamente a opção *Login* e informar suas credenciais de acesso definidas no cadastro e, logo após iniciar a sessão, poderá selecionar a opção *New Course*, descrita na subseção 5.2.3.

Uma vez na página de cadastro de um novo curso, basta preencher os dados solicitados para realizar o cadastro deste curso (Figura 38).

Figura 38 – Iniciando o cadastro de um novo curso



The screenshot shows the 'New Course' form in the RedeR Gallery. The form is titled 'New Course' and has a blue header with navigation links: 'RedeR Gallery', 'Gallery', 'Courses', 'RedeRWeb', and 'Download RedeR'. The user's name 'Matheus' is visible in the top right corner. The form fields are:

- *Course Title:** Representação Hierárquica da Rede PPI da bactéria E. Coli
- *Description:** A rich text editor containing the following text:

Apresentação

Redes biológicas possuem a característica de serem hierárquicas e explorar esta característica é de suma importância na sua análise e interpretação dos dados. Ao longo deste curso iremos utilizar o ambiente de programação R junto com pacotes para minipulação de redes hierárquicas, de dados de anotação gênica e dados do genoma da bactéria Escherichia Coli (E. Coli). Para organizar de maneira hierárquica a rede, será utilizada a estrutura hierárquica de anotações presentes no Gene Ontology. Ao final deste curso, você estará apto a lidar com esses dados, assim como manipular rede modulares e hierárquicas utilizando o pacote RedeR.

As etapas efetuadas durante esta análise serão:

 1. Carregas as informações de anotações referentes ao genoma da bactéria E. Coli.
 2. Recuperar as informações referentes à interações PPI (Protein-Protein Interaction).
 3. Recuperar os termos do Gene Ontology em que os genes estão anotados.
 4. Montar a rede referente aos termos em que são anotados os genes da E. Coli.
 5. Salvar os termos na rede e observar.
- Git:** git
- Cover Image:** simple from vignette.png

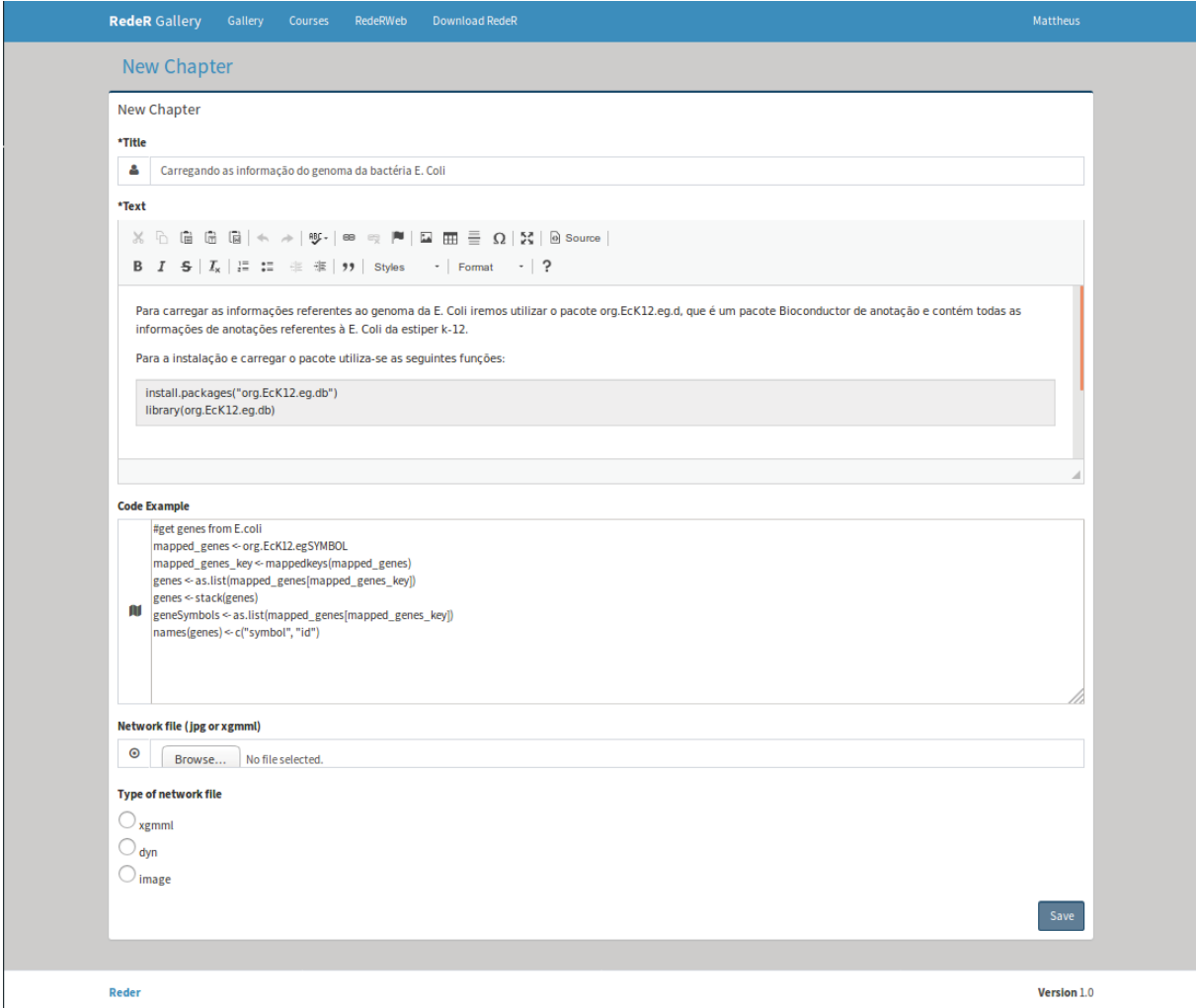
A 'Save' button is located at the bottom right of the form.

Imagem demonstrando o cadastro de um novo curso com as informações deste estudo de caso. Fonte: o Autor.

6.5.3 Cadastro de um Novo Capítulo

Após realizado o cadastro do curso, o sistema irá exibir a página de edição e atualização do curso, onde será possível atualizar suas informações, assim como, cadastrar um novo capítulo ou editar um capítulo já existente. Na figura a seguir é demonstrado a página de cadastrar um capítulo (Figura 39).

Figura 39 – Iniciando o cadastro de um novo capítulo



The screenshot shows the 'New Chapter' form in the RedeR Gallery. The form is titled 'New Chapter' and has a blue header with navigation links: 'RedeR Gallery', 'Gallery', 'Courses', 'RedeRWeb', and 'Download RedeR'. The user's name 'Mattheus' is visible in the top right corner.

The form contains the following sections:

- Title:** A text input field containing 'Carregando as informações do genoma da bactéria E. Coli'.
- Text:** A rich text editor with a toolbar. The text area contains the following text:

Para carregar as informações referentes ao genoma da E. Coli iremos utilizar o pacote `org.EcK12.eg.d`, que é um pacote Bioconductor de anotação e contém todas as informações de anotações referentes à E. Coli da estirpe k-12.

Para a instalação e carregar o pacote utiliza-se as seguintes funções:

```
install.packages("org.EcK12.eg.db")
library(org.EcK12.eg.db)
```
- Code Example:** A code editor with the following R code:

```
#get genes from E.coli
mapped_genes <- org.EcK12.egSYMBOL
mapped_genes_key <- mappedkeys(mapped_genes)
genes <- as.list(mapped_genes[mapped_genes_key])
genes <- stack(genes)
geneSymbols <- as.list(mapped_genes[mapped_genes_key])
names(genes) <- c("symbol", "id")
```
- Network file (jpg or xgmml):** A file upload field with a 'Browse...' button and the text 'No file selected.' Below it are radio buttons for the type of network file:
 - xgmml
 - dyn
 - image

A 'Save' button is located at the bottom right of the form. The footer of the page shows 'Reder' on the left and 'Version 1.0' on the right.

Imagem demonstrando o cadastro de um novo capítulo com as informações deste estudo de caso. Fonte: o Autor.

No cadastro de um novo capítulo existe a opção de enviar um arquivo, que será utilizado para demonstrar o resultado da análise efetuada naquele capítulo. O usuário poderá selecionar qual o formato do arquivo que está enviando e, caso seja um arquivo formato *dyn* ou *xgmml*, a rede em questão será disponibilizada na plataforma RedeRWeb, caso seja uma imagem, a mesma será mostrada ao final do capítulo (Figura 40).

Figura 40 – Visualização de um capítulo já cadastrado

Selecionando a Hierarquia Desejada

Com a visualização da hierarquia no pacote RedeR, conseguimos selecionar manualmente os termos que desejamos e recupera-los para o ambiente R.

Após selecionados estes termos podemos construir um novo objeto para a hierarquia selecionada, realizando o filtro da hierarquia conforme descrito no capítulo Filtrando a Hierarquia do GO, assim como aplicar todos os tratamentos realizados nos atributos dos nós.

Ao final do processo, a hierarquia selecionada pode ser visualizada novamente no pacote RedeR

```

goGraphSelected <- getGraph(rdp, status= "selected")

##### Pegando os genes selecionados e montando a rede a partir deles, é necessário
montar a hierarquia novamente
#####

#get hierarchy from selected terms
hierarquiaSelecionada <- moverHierarquia(termosParaManter = V(goGraphSelected)$name, hier
arquia = hierarquia)

goTree <- goHierarchyToDataTree(hierarquiaSelecionada)
goGraphS <- directGraphGoHierarchy(hierarquiaSelecionada)

#apply nodeSize, normalizes and style color
V(goGraphS)$nodeSize <- as.numeric(dfMapped_GO[which(dfMapped_GO$terms %in% V(goGraphS)$
name),2])
V(goGraphS)$nodeSize <- sqrt(V(goGraphS)$nodeSize)
V(goGraphS)$nodeSize<- V(goGraphS)$nodeSize*3
goGraphS <- att.setv(g=goGraphS, from="nodeSize", to="nodeColor", breaks=unique(V(goGra
phS)$nodeSize), pal=2)
V(goGraphS)$nodeAlias <- Term(GOTERM[V(goGraphS)$name])
V(goGraphS)$nodeFontSize <- 36

resetd(rdp)
addGraph(rdp, goGraphS)

```

Imagem demonstrando como é visualizado um capítulo já cadastrado e com um arquivo no formato *dyn*, para visualização da rede. Fonte: o Autor.

Uma vez finalizado o cadastro de todos os capítulos, o curso poderá ser acessado selecionando a opção *Courses*, no menu superior, e, logo após, selecionar o curso desejado. Com isso, o sistema irá carregar as informações desse curso, como os capítulos e imagens, e o usuário poderá navegar por todas as suas informações (Figura 41).

Figura 41 – Visualização de um curso

The screenshot displays the 'RedeR Gallery' interface. The main content area is titled 'Representação Hierárquica da Rede PPI da bactéria E. Coli' by author Matheus Rodrigo Marzola Leite. The course description states that biological networks are hierarchical and that the course aims to explore this characteristic using R programming. It details the steps: 1. Carregar as informações de anotações referentes ao genoma da bactéria E. Coli. 2. Recuperar as informações referentes à interações PPI (Protein-Protein Interaction). 3. Recuperar os termos do Gene Ontology em que os genes estão anotados. 4. Montar a rede referente aos termos em que são anotados os genes da E. Coli. 5. Selecionar os termos que se deseja observar. 6. Filtrar os genes que estão anotados nos termos selecionados. 7. Gerar a rede PPI referente ao gene filtrado no passo anterior. 8. Aplicar a hierarquia do termo do GO na rede PPI. The 'Pacotes que serão utilizados' section lists: org.Hs.eq.db, org.EcK12.eq.db, Cytoscape, Igraph, Rcolor, GO.db, data.tree, and RColorBrewer. The 'Referências' section includes: LIO, F. et al. Modular organization of protein interaction networks. *Bioinformatics*, v. 23, n. 2, p. 207-214, 2007. ISSN 13674803. SILVA, M. M. da. RedeRWeb: uma plataforma web para organização e análise de redes modulares. *Dissertação (Mestrado) - Universidade Federal do Paraná*, 2017. CASTRO, M. A. et al. RedeR: Bioconductor package for representing modular structures, nested networks and multiple levels of hierarchical associations. *Genome Biology*, 2012. CARLSON, M. org.eck12.eq.db: Genome wide annotation for e coli strain k12. r package version 3.5.0. Bioconductor, 2017. CARLSON, M. Go.db: A set of annotation maps describing the entire gene ontology. r package version 3.5.0. Bioconductor, 2017. CUARDI, G. NEPHELI. The graph software package for complex network research. *InterJournal*, 2006. Disponível em: <http://www.interjournal.org/doi/10.21963/interjournal.2006.060101>

Imagem do acesso ao curso cadastrado com as informações deste estudo de caso. Fonte: o Autor.

6.5.4 Resultados

Análise de redes complexas não é uma tarefa trivial e, muitas vezes, uma forma de simplificar esta representação é uma fator chave para observar determinadas características. Neste estudo de caso foi demonstrada uma maneira de simplificação dessa representação, utilizando a estrutura hierárquica do *Gene Ontology* na rede PPI da *Escherichia Coli*. Com isso, foi possível observar, além das interações entre as proteínas, como estas interações estão organizadas em um nível hierárquico. Esta é uma característica importante das ferramentas RedeR e RedeRWeb, que dispõem de meios de realizarem representações hierárquicas, gerando imagens com alto valor informacional, como demonstrado neste estudo de caso. Utilizando a plataforma RedeR Gallery, os pesquisadores poderão compartilhar suas análises e técnicas utilizadas junto às ferramentas citadas, contribuindo assim para o aprimoramento das mesmas, análises que poderão ser consultadas por toda a comunidade e, conseqüentemente, colaborando com a experiência na utilização da ferramenta, que pode ser utilizada para representação, não apenas de redes biológicas hierárquicas, mas sim de qualquer rede hierárquica.

7 CONCLUSÃO E PERSPECTIVAS FUTURAS

Nas últimas décadas, com o avanço tecnológico, houve um grande aumento na capacidade de coleta, armazenamento e processamento de dados e novas técnicas para observar e gerar conhecimento sobre esses dados se tornaram necessárias. A análise de redes se torna uma importante ferramenta nesse contexto, tendo como base a Teoria dos Grafos, vem evoluindo e encontrando novas propriedades ao longo da história, como por exemplo as redes de livre escala e redes hierárquicas. Sendo possível observar como os dados interagem entre si em um nível global, análise de redes pode ser utilizada nos mais diversos tipos de dados, seja para analisar a interação entre proteínas de um organismo ou em como pessoas interagem em uma rede social.

Por vezes, essa grande quantidade de dados, pode dificultar a análise de redes, resultando em uma rede de difícil observação e interpretação. Para contornar este problema pode ser utilizados meios de simplificação dessa representação, seja na seleção dos dados ou na representação da rede final. Uma forma de simplificar a representação é utilizando a característica da hierarquia presente nos mais diversos tipos de dados, como por exemplo redes biológicas. Redes biológicas possuem a característica de serem hierárquicas, e tirar proveito dessa hierarquia não só simplifica a representação da rede, como também permite a observação das interações em um nível sistêmico, podendo observar como os diversos níveis da hierarquia interagem entre si.

Neste trabalho apresentamos a RedeR Gallery, uma plataforma para publicação de estudos de caso e cursos e mini-cursos utilizando as ferramentas RedeR e RedeRWeb. As ferramentas RedeR e RedeRWeb são destinadas à análises e manipulações de redes hierárquicas, o que não é uma tarefa trivial. A RedeR Gallery vem com o objetivo de auxiliar novos usuários das ferramentas RedeR e RedeRWeb e também aprimorar a experiência de usuários já habituados em sua utilização. RedeR Gallery será a ponte de ligação entre as ferramentas RedeR e RedeRWeb, sendo utilizada como um local de colaboração e aprendizado, assim como um ponto de entrada para pesquisadores de outras áreas de pesquisa que utilizam de técnicas de análise de redes em outros objetos de estudo. Futuramente iremos ampliar esta integração da RedeR Gallery e as ferramentas RedeR e RedeRWeb, disponibilizando mais opções, como por exemplo, a possibilidade de acessar a galeria utilizando a ferramenta RedeR e acessar as redes de modo integrado. Outro objetivo futuro seria disponibilizar novos conteúdos para a galeria, seja em treinamentos para um primeiro usuário ou até mesmo para um pesquisador interessado no aprendizado do ambiente de desenvolvimento R.

Neste trabalho também é apresentado um estudo de caso onde estruturamos a informação contida no *Gene Ontology* e realcionamos essa informação com os genes contidos no genoma da bactéria *Scherichia Ecoli*, onde consideramos esta estrutura tanto na visualização de anotação global do genoma da bactéria, quando na organização hierárquica deste genoma, levando em

consideração a hierarquia presente no *Gene Ontology*. Esta abordagem se mostrou bastante relevante, pois pode ser aplicada em diferentes perspectivas, seja de ontologia ou de conjuntos de genes utilizados na visualização, como por exemplo genes envolvidos em um tipo específico de câncer, que podem ser visualizados sob a ótica, tanto da ontologia de Processos Biológicos, como da ontologia de Funções Moleculares. Com os resultados deste estudo de caso, pretende-se realizar o desenvolvimento e elaboração de um pacote R voltado à manipulação e organização da estrutura hierárquica do *Gene Ontology* de acordo com um determinado conjunto de genes.

REFERÊNCIAS BIBLIOGRÁFICAS

- ASHBURNER, M. et al. Gene ontology: Tool for the unification of biology. *Nature Genetics*, v. 25, p. 25–29, 2000. Citado 2 vezes nas páginas 20 e 45.
- BARABASI, A.-L.; ALBERT, R. Emergence of scaling in random networks. *science*. v. 286, p. 509–12, 11 1999. Citado 2 vezes nas páginas 17 e 18.
- BARABÁSI, A.-L.; OLTVAI, Z. N. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, v. 5, n. 2, p. 101–113, 2004. Citado 2 vezes nas páginas 20 e 24.
- BARABÁSI, B. A.-L.; BONABEAU, E. Scale-free networks. *Scientific American*, v. 288, n. 5, p. 50–59, 2003. Citado 2 vezes nas páginas 16 e 17.
- BENSON, D. A. et al. Genbank. *Nucleic Acids Research*, 2013. Citado na página 12.
- BIOCONDUCTOR. *About Bioconductor*. 2018. Disponível em: <<https://www.bioconductor.org/about/>>. Citado na página 12.
- BLATTNER, F. R. et al. The complete genome sequence of escherichia coli k-12. *Science*, American Association for the Advancement of Science, 1997. Citado 2 vezes nas páginas 45 e 46.
- BOCCALETTI, S. et al. Complex networks: Structure and dynamics. v. 424, n. 4-5, p. 175–308, 2006. Citado na página 16.
- BONDY, J. A.; MURTY, U. S. R. Graph Theory with Applications. v. 28, n. 1, p. 237, 1977. ISSN 00303623. Citado na página 15.
- BOOTSTRAP. *About*. 2018. Disponível em: <<http://getbootstrap.com.br/docs/4.0/about/overview/>>. Citado na página 32.
- CAELUM. *Uma introdução prática ao JPA com Hibernate*. 2018. Disponível em: <<https://www.caelum.com.br/apostila-java-web/uma-introducao-pratica-ao-jpa-com-hibernate/>>. Citado na página 30.
- CARBON, S. et al. Expansion of the gene ontology knowledgebase and resources: The gene ontology consortium. *Nucleic Acids Research*, v. 45, p. D331–D338, 2017. Citado na página 20.
- CARLSON, M. Go.db: A set of annotation maps describing the entire gene ontology. r package version 3.5.0. *Bioconductor*, 2017. Citado na página 45.
- CARLSON, M. org.eck12.eg.db: Genome wide annotation for e coli strain k12. r package version 3.5.0. *Bioconductor*, 2017. Citado na página 45.
- CASTRO, M. A. et al. Reder: R/bioconductor package for representing modular structures, nested networks and multiple levels of hierarchical associations. *Genome Biology*, 2012. Citado 2 vezes nas páginas 24 e 25.
- CHACON, S.; STRAUB, B. *Pro Git*. [S.l.: s.n.], 2014. 13 p. ISBN 978-1484200773. Citado na página 34.

- CONSORTIUM, G. O. *The Gene Ontology Project*. 2018. Disponível em: <<http://www.geneontology.org/page/about>>. Citado na página 45.
- CSARDI, G.; NEPUSZ, T. The igraph software package for complex network research. *InterJournal*, 2006. Disponível em: <<http://igraph.org>>. Citado na página 46.
- DEVMEDIA. *Introdução ao Padrão MVC*. 2018. Disponível em: <<https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>>. Citado na página 28.
- DEVMEDIA. *Introdução à JPA - Java Persistence API*. 2018. Disponível em: <<https://www.devmedia.com.br/introducao-a-jpa-java-persistence-api/28173>>. Citado na página 30.
- ECKSTEIN, R. *Java SE Application Design With MVC*. 2018. Disponível em: <<http://www.oracle.com/technetwork/articles/javase/index-142890.html#1>>. Citado na página 27.
- ERDOS, P.; RENYI, A. On the evolution of random graphs. *Publ. Math. Inst. Hungary. Acad. Sci.*, v. 5, 1960. Citado na página 16.
- ESTRADA, E.; KNIGHT, P. *A First Course in Network Theory*. [S.l.: s.n.], 2015. 272 p. ISBN 0198726457. Citado na página 16.
- EULER, L. Solutio problematis ad geometriam situs pertinentis. v. 8, p. 128–140, 1736. Citado na página 16.
- FOUNDATION, E. *About the Eclipse Foundation*. 2018. Disponível em: <<http://www.eclipse.org/org>>. Citado na página 32.
- FOUNDATION, T. R. *What is R?* 2018. Disponível em: <<https://www.r-project.org/about.html>>. Citado na página 35.
- FRASER, H. B. et al. Evolutionary rate in the protein interaction network. *Science*, v. 296, p. 750–752, 2002. Citado na página 17.
- GIT. *Git*. 2018. Disponível em: <<https://git-scm.com/>>. Citado na página 34.
- GOMES, F. *Introdução ao Java Server Pages (JSP)*. 2018. Disponível em: <<http://www.devmedia.com.br/introducao-ao-java-server-pages-jsp/25602>>. Citado na página 29.
- GOOGLEMAPS. *Google Maps*. 2018. Disponível em: <www.google.com/maps>. Citado na página 15.
- GROUP, J. U. *Wildfly - Desvendando-o do básico ao avançado*. 2018. Disponível em: <<https://www.gitbook.com/book/jboss-books/wildfly/details>>. Citado na página 35.
- GUIMAR, C. Considerações sobre o uso de Visualização de Informação no auxílio à gestão de informação. 2016. Citado na página 20.
- HECKER, M. et al. Gene regulatory network inference: Data integration in dynamic models-A review. *BioSystems*, v. 96, p. 86–103, 2009. Citado na página 17.
- HOOPEER, S. D.; BORK, P. Medusa: a simple tool for interaction graph analysis. *Bioinformatics*, 2005. Citado na página 24.

IBPAD, I. B. d. P. e. A. d. D. *O que é Programação ou Linguagem em R?* 2018. Disponível em: <<https://www.ibpad.com.br/blog/o-que-e-programacao-ou-linguagem-em-r/>>. Citado na página 35.

JAVA. *Java*. 2018. Disponível em: <www.java.com>. Citado na página 29.

JEONG, H. et al. The large-scale organization of metabolic networks. *Nature*, v. 407, p. 651–654, 2000. Citado na página 17.

JQUERY. *Jquery*. 2018. Disponível em: <<https://jquery.org/>>. Citado na página 31.

JUNKER, B. H.; KOSCHÜTZKI, D.; SCHREIBER, F. Exploration of biological network centralities with centibin. *BMC Bioinformatics*, p. 219, 2006. Citado na página 24.

KAR, G.; GURSOY, A.; KESKIN, O. Human cancer protein-protein interaction network: A structural perspective. *PLoS Computational Biology*, v. 5, n. 12, 2009. Citado na página 17.

KHAN, M.; KHAN, S. S. Data and Information Visualization Methods , and Interactive Mechanisms : A Survey. v. 34, p. 1–14, 2011. Citado 2 vezes nas páginas 22 e 23.

KOHL, M.; WIESE, S.; WARSCHEID, B. Cytoscape: software for visualization and analysis of biological networks. In: *Data Mining in Proteomics*. [S.l.]: Springer, 2011. p. 291–303. Citado na página 24.

LUO, F. et al. Modular organization of protein interaction networks. *Bioinformatics*, v. 23, n. 2, p. 207–214, 2007. ISSN 13674803. Citado 2 vezes nas páginas 18 e 45.

MARIADB. *About MariaDB*. 2018. Disponível em: <<https://mariadb.org/about/>>. Citado na página 33.

MASON, O.; VERWOERD, M. Graph theory and networks in biology. *IET Systems Biology*, v. 1, n. 2, March 2007. Citado 2 vezes nas páginas 17 e 24.

QUICKGO. *QuickGO: Gene Ontology and GO Annotations*. 2018. Disponível em: <<https://www.ebi.ac.uk/QuickGO/>>. Citado na página 21.

RAVASZ, E.; BARABASI, A.-L. Hierarchical Organization in Complex Networks. 2002. Citado 2 vezes nas páginas 18 e 24.

RAVASZ, E. et al. Networks Hierarchical Organization of Modularity in Metabolic Networks. *Science*, v. 297, n. August, p. 1551–1555, 2002. ISSN 1095-9203. Citado 3 vezes nas páginas 17, 19 e 20.

REDHAT. *Hibernate ORM*. 2018. Disponível em: <<http://hibernate.org/orm/>>. Citado na página 31.

SILVA, N. M. da. *RedeR Web: uma plataforma web para organização e análise de redes modulares*. Dissertação (Mestrado) — Universidade Federal do Paraná, 2017. Citado 2 vezes nas páginas 25 e 26.

SIMILARTECH. *SimilarTech*. 2018. Disponível em: <<https://www.similartech.com/technologies/jquery>>. Citado na página 31.

- SZKLARCZYK, D. et al. The string database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic Acids Research*, 2017. Citado na página 12.
- TAO, Y. et al. Information visualization techniques in bioinformatics during the postgenomic era. *Drug Discovery Today: BIOSILICO*, v. 2, 2004. Citado 2 vezes nas páginas 22 e 23.
- THORVALDSDÓTTIR, H.; ROBINSON, J. T.; MESIROV, J. P. Integrative Genomics Viewer (IGV): High-performance genomics data visualization and exploration. *Briefings in Bioinformatics*, v. 14, 2013. Citado na página 23.
- TUFTE, E.; GRAVES-MORRIS, P. *The visual display of quantitative information*. 1983. Citado na página 22.
- WANG, L.; WANG, G.; ALEXANDER, C. A. Big Data and Visualization: Methods, Challenges and Technology Progress. *Digital Technologies*, v. 1, p. 33–38, 2015. Citado na página 22.
- WATTS, D. J.; STROGATZ, S. H. Collective dynamics of 'small-world' networks. *Nature*, v. 393, n. 6684, p. 440–442, 1998. Citado 2 vezes nas páginas 16 e 17.
- WESTON, A. D.; HOOD, L. Systems biology, proteomics, and the future of health care: toward predictive, preventative, and personalized medicine. *Journal of proteome research*, ACS Publications, 2004. Citado na página 12.
- WILDFLY. *What is WildFly?* 2018. Disponível em: <<http://wildfly.org/about/>>. Citado na página 35.
- ZHANG, S. B.; TANG, Q. R. Protein-protein interaction inference based on semantic similarity of Gene Ontology terms. *Journal of Theoretical Biology*, v. 401, p. 30–37, 2016. Citado na página 20.