

UNIVERSIDADE FEDERAL DO PARANÁ

RODRIGO LANGOWSKI

**RedIBio - Uma ferramenta para visualização de Redes de Interações
Biológicas**

CURITIBA

2017

RODRIGO LANGOWSKI

RedIBio – Uma ferramenta para visualização de Redes de Interações Biológicas

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

Orientador: Prof. Dr. Alessandro Brawerman

Co-orientadora: Prof^a. Dr^a Maria Berenice R. Steffens

CURITIBA

2017

L284 Langowski, Rodrigo
RedIBio – Uma ferramenta para visualização de redes de interações biológicas / Rodrigo Langowski. - Curitiba, 2016.
74 f.: il., tabs, grafs.

Orientador: Prof. Dr. Alessandro Brawerman
Coorientadora: Prof^a. Dra Maria Berenice R. Steffens
Dissertação (Mestrado) – Universidade Federal do Paraná, Setor de Educação Profissional e Tecnológica, Curso de Pós-Graduação em Bioinformática.

1. Rede biológica. 2. Ferramenta biológica. 3. Biologia computacional. 4. Bioinformática. I. Brawerman, Alessandro. II. Steffens, Maria Berenice. III. Título. IV. Universidade Federal do Paraná.

CDD 575.113

TERMO DE APROVAÇÃO

RODRIGO LANGOWSKI

"REDIBIO – UMA FERRAMENTA PARA VISUALIZAÇÃO DE REDES DE INTERAÇÕES BIOLÓGICAS"

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Bioinformática, pelo Programa de Pós-graduação em Bioinformática, Setor de Educação Profissional e Tecnológica, da Universidade Federal do Paraná, pela seguinte banca examinadora:


Dr. Alessandro Brawerman
Universidade Federal do Paraná - UFPR


Dr. André Yoshiaki Kashiwabara
Universidade Tecnológica Federal do Paraná - UTFPR


Dr. Robson Francisco de Souza
Universidade de São Paulo - USP


Dr.ª Maria Berenice Reynaud Steffens
Universidade Federal do Paraná - UFPR

Curitiba, 15 de dezembro de 2016

Dedico esta dissertação aos meus pais Aristeu e Helen, à minha avó Cacilda, ao meu irmão Eduardo e ao meu amor Aniele, que me apoiaram em todos os momentos durante a execução do trabalho.

AGRADECIMENTOS

Ao meu Orientador, Professor Dr. Alessandro Brawerman, por ter me aceitado como orientado e proporcionar esta oportunidade única.

À minha Co-orientadora, Professora Dra. Maria Berenice Steffens, por me incluir no grupo de pesquisa e pelo incentivo de meu amadurecimento profissional e pessoal.

À minha família, por acreditar em minhas escolhas, por estarem de prontidão em meus insucessos e pelo amor que recebi por toda minha vida.

À Aniele Leão, por me apoiar incondicionalmente, caminhar ao meu lado durante os momentos de alegria e nos complicados, compartilhar de seu tempo e paciência para me ajudar em todos os aspectos de minha vida e tornar toda a experiência que este mestrado me proporcionou em um passeio tranquilo. Amo você!

Aos amigos que adquiri durante esses 2 anos, em que pude ter momentos de descontração e expandir meus conhecimentos através de trocas de experiências.

Ao Programa de Pós-Graduação em Bioinformática da UFPR, professores e à Suzana Gobetti por toda a dedicação e atenção.

À CAPES, pelo financiamento.

“Não existe ensino que se compare ao exemplo.”
Robert Baden-Powell

RESUMO

A representação gráfica de dados biológicos é considerada uma problemática crescente no âmbito da pesquisa, pois a quantidade de informações a se interpretar aumenta em um ritmo acelerado. Neste cenário, o desenvolvimento de ferramentas que auxiliem na interpretação de grande quantidade de dados é uma necessidade. A visualização da informação oferece recursos para análise de dados e geração de conhecimentos. Um dos métodos para representação de dados na forma gráfica é a rede, que possibilita a visualização do todo e a detecção de padrões de interação. A entrega do formato visual, no tipo rede, simplificado para a compreensão de informações biológicas complexas e aliado à conexão com uma fonte externa de informação, é o objetivo proposto pelo projeto. A ferramenta desenvolvida e denominada RedIBio - Redes de Interação Biológicas, oferece também o recurso de interação que possibilita a navegabilidade através dos objetos que compõem a estrutura da rede. A rede gerada pela ferramenta é visualizada em três dimensões garantindo que os componentes da rede sejam representados sem sobreposição em uma área maior. Sua disponibilidade e funcionamento ocorre totalmente pela Internet, dispensando, assim, a necessidade de instalação local e preocupações com compatibilidade e atualizações recorrentes. Como resultado final, a RedIBio não se limita apenas em apresentar os dados contidos no arquivo de entrada, pois sua integração com banco de dados biológico permite o acesso a informações detalhadas a respeito do objeto que são úteis ao usuário.

Palavras-chave: Visualização da informação. Rede biológica. Ferramenta biológica.

ABSTRACT

The graphical representation of biological data is considered a growing problem in research, because the amount of information to be interpreted increases at an accelerated pace. In this scenario, the development of tools that aid in the interpretation of large amounts of data is a necessity. Visualization of information provides resources for data analysis and knowledge generation. One of the methods for representing data in graphic form is the network, which enables the visualization of the whole and the detection of patterns of interaction. The delivery of the visual format, in the network type, simplified for the understanding of complex biological information and allied to the connection with an external source of information, is the objective on this project. The tool developed and called RedIBio - Redes de Interação Biológicas, also offers the interaction feature that enables navigability through the objects that make up the network structure. The network generated by the tool is visualized in three dimensions ensuring that the network components are represented without overlapping over a larger area. Its availability and operation occurs entirely over the Internet, thus dispensing with the need for local installation and concerns with compatibility and recurring updates. As a final result, RedIBio is not only limited to presenting the data contained in the input file, since its integration with a biological database allows access to detailed information about the object that is useful to the user.

Key-words: Visualization of information. Biological network. Biological tool.

LISTA DE FIGURAS

FIGURA 1 - PASSOS DA TRANSFORMAÇÃO DE DADOS EM CONHECIMENTO..	18
FIGURA 2 - DUAS MANEIRAS GRÁFICAS DE RELACIONAMENTOS ENTRE ENTIDADES.....	20
FIGURA 3 - MEDIÇÃO DIÁRIA DA TEMPERATURA DE CURITIBA NO MÊS DE MAIO DE 2016.	21
FIGURA 4 - PORCENTAGEM DE CADA DESPESA EM RELAÇÃO AO TOTAL DE DESPESAS.....	22
FIGURA 5 - VARIAÇÕES DAS DESPESAS AO LONGO DO TEMPO	23
FIGURA 6 - DIAGRAMA DE UMA REDE DE COMPUTADORES.....	24
FIGURA 7 - VARIAÇÕES DO ÍCONE DE BATERIA DE UM SMARTPHONE.....	25
FIGURA 8 - REDE DE DOENÇAS HUMANAS	25
FIGURA 9 - REDE DIRETA.....	27
FIGURA 10 - REDE INDIRETA.....	28
FIGURA 11 - ESTRUTURA MVC.....	29
FIGURA 12 - ELEMENTOS DO WEB SERVICE E OPERAÇÕES.....	31
FIGURA 13 - DEMONSTRAÇÃO DE DADOS NA LINGUAGEM XML	32
FIGURA 14 - DEMONSTRAÇÃO DE DADOS NA LINGUAGEM JSON.....	33
FIGURA 15 - FLUXOGRAMA DE FUNCIONAMENTO DA REDIBIO	38
FIGURA 16 - DIAGRAMA DE AÇÕES DA REDIBIO	44
FIGURA 17 - ESTRUTURA DOS ARQUIVOS DO PROJETO REDIBIO.....	45
FIGURA 18 - CAPTURA DE TELA DO MENU NA PÁGINA DA REDIBIO	45
FIGURA 19 - CAPTURA DE TELA DA PÁGINA INICIAL DA REDIBIO	46
FIGURA 20 - CAPTURA DE TELA DA PÁGINA DE INFORMAÇÕES SOBRE A REDIBIO	46
FIGURA 21 - CAPTURA DE TELA DE CONTATO DA PÁGINA DA REDIBIO.....	47
FIGURA 22 - CAPTURA DE TELA DA SESSÃO DE UPLOAD.....	47
FIGURA 23 - INSERÇÃO DOS NÓS NA CENA	49
FIGURA 24 - PARTÍCULAS INSERIDAS PRÓXIMAS AO NÓ DE INTERAÇÃO	50
FIGURA 25 - INSERÇÃO DE LINHAS DE INTERAÇÃO ENTRE O NÓ E O ALVO.	50
FIGURA 26 - ADIÇÃO DOS NOMES DOS ALVOS	51
FIGURA 27 - ADIÇÃO DE ALVOS DE MAIS DE UMA INTERAÇÃO FORMANDO A REDE FINAL.....	52
FIGURA 28 - REPRESENTAÇÃO DOS PONTOS DE UM GRÁFICO 2D PARA 3D	53
FIGURA 29 - TRAJETÓRIA DO RAIOS A PARTIR DA TELA DO MONITOR ATÉ OS OBJETOS	53

FIGURA 30 - ABA EXPANDE AO SELECIONAR UM ALVO.....	54
FIGURA 31 - RESPOSTA DA CAIXA DE TEXTO COM O AUTO-COMPLETAR.....	54
FIGURA 32 - OS RESULTADOS OBTIDOS PELO TOGOWS.....	55
FIGURA 33 - SOLICITAÇÃO DA BUSCA PARA O WEB SERVICE	55
FIGURA 34 - SOLICITAÇÃO DE ENTRADA A PARTIR DO ID PARA O WEB SERVICE	56
FIGURA 35 - EXEMPLO DA ENTRADA DE DADOS DA REDIBIO.....	56
FIGURA 36 - REDE EM 2D DA PREDIÇÃO DE ALVOS PARA 165 RNAs CURTOS	59
FIGURA 38 - CAPTURA DE TELA DE UMA REDE DA REDIBIO DEMONSTRANDO A NAVEGAÇÃO A PARTIR DE UM PONTO ESPECÍFICO	60
FIGURA 37 - CAPTURA DE TELA DE UMA REDE DE INTERAÇÕES GERADA PELA REDIBIO	60
FIGURA 39 – LOCALIZAÇÃO DE UM NÓ COM ALVOS HIPOTÉTICOS.....	61
FIGURA 40 - SELEÇÃO DO HSERO_0164	62
FIGURA 41 - NOME REAL DO NÓ HS_SRNA_GS8 OBTIDO PELA CONSULTA AO BANCO DE DADOS BIOLÓGICO.	62
FIGURA 42 - TEMPO DE EXECUÇÃO PELA VARIAÇÃO DOS NÓS	65
FIGURA 43 - TEMPO DE EXECUÇÃO PELA VARIAÇÃO DOS ALVOS	65

LISTA DE TABELAS

TABELA 1 - SIMULAÇÃO DE DESPESAS	22
TABELA 2 - EXEMPLO DO RESULTADO DA PREDIÇÃO DE UM RNA CURTO...	57
TABELA 3 - EXEMPLO DA ENTRADA DE DADOS DA REDIBIO	58
TABELA 4 - VARIAÇÕES DE FRAMES POR SEGUNDO, TEMPO PARA CARREGAR UM FRAME EM MILISSEGUNDOS E MEMÓRIA ALOCADA EM MEGABYTES	66

LISTA DE QUADROS

QUADRO 1 - CLASSES DE REPRESENTAÇÕES VISUAIS	20
QUADRO 2 - PRINCIPAIS DIFERENÇAS ENTRE DNA E RNA	34
QUADRO 3 – CLASSES DE RNAs E SUAS RESPECTIVAS FUNÇÕES	35
QUADRO 4 - COMPARATIVO ENTRE FERRAMENTAS DE VISUALIZAÇÃO DE REDES EM 3D.....	63

SUMÁRIO

1 INTRODUÇÃO	14
1.1 OBJETIVOS	15
1.1.1 Objetivo geral	16
1.1.2 Objetivos específicos	16
1.2 JUSTIFICATIVA E RELEVÂNCIA DO TRABALHO	16
1.3 ORGANIZAÇÃO DA DISSERTAÇÃO	17
2 FUNDAMENTAÇÃO TEÓRICA	18
2.1 VISUALIZAÇÃO DA INFORMAÇÃO	18
2.2 REDES DE INTERAÇÃO	26
2.3 MVC	28
2.4 WEB SERVICE	29
2.4.1 SOAP	30
2.4.2 REST	31
2.4.3 XML	32
2.4.4 JSON	33
2.5 RNA	34
2.5.1 RNA CURTO	35
2.6 ALGORITMOS DE PREDIÇÕES	37
3 MATERIAL E MÉTODOS	38
3.1 VISÃO GERAL	38
3.2 AMBIENTE SOFTWARE E LINGUAGEM DE PROGRAMAÇÃO	39
3.2.1 HTML	39
3.2.2 PHP	39
3.2.3 <i>JavaScript</i>	40
3.2.3.1 <i>Three.js</i>	40
3.2.3.2 <i>JQuery</i>	41
3.2.4 CodeIgniter	41
3.2.5 TogoWS REST API	41
3.3 DESENVOLVIMENTO	42
3.3.1 Interface e Upload	45
3.3.2 Leitura e Organização	47

3.3.3 Montagem	48
3.3.4 Interação	52
3.3.4.1 Infobox	52
3.3.4.2 NSearch	54
3.3.5 Integração com Banco de Dados Biológico	55
4 RESULTADOS.....	57
4.1 APLICAÇÃO NO ESTUDO DE CASO.....	57
4.2 COMPARATIVO	63
4.3 PERFORMANCE	64
4.3.1 Construção da Rede	64
4.3.2 Renderização	66
5 CONCLUSÃO	68
REFERÊNCIAS.....	70
APÊNDICE 1	74

1 INTRODUÇÃO

A extração de conhecimento, e a sua compreensão, a partir de uma representação gráfica resultante de um montante de informações, requer uma constante preocupação da melhor forma para a apresentação dos dados. A visualização da informação (VI) tem como objetivo auxiliar a compreender uma grande quantidade de informação de uma única vez, bem como extrair conhecimento útil destas informações (FREITAS *et al.*, 2001).

É possível notar a existência da VI no cotidiano como por exemplo, no indicador de bateria dos *smartphones*. Geralmente possuem a forma de uma bateria indicando a carga do aparelho e, conforme o uso do aparelho a bateria é consumida, a figura vai se modificando, e o usuário consegue perceber a informação transmitida por esta figura. Quando a bateria do aparelho está muito baixa, a figura muda de cor apresentando uma bateria vermelha chamando a atenção e informando ao usuário que é necessário recarregar o seu *smartphone*. A representação gráfica da informação, ao se utilizar de recursos de VI, contribui para que os usuários deduzam novas informações, a partir de um objeto gráfico, desenho ou uma animação (DIAS, 2007).

Para interpretar e demonstrar informações biológicas, utilizando da VI, é necessário se preparar para grandes quantidades de dados, como por exemplo os genomas. As informações contidas na estrutura do DNA humano consistem em, aproximadamente, 3 bilhões de pares de base (VOGEL; MOTULSKY, 1997) e cada par de base, ao ser armazenado em modo digital em computadores, representa 2 bits. Desta maneira, se armazenado todo o genoma humano em um disco rígido e compactando as informações, este ocuparia aproximadamente 950 Megabytes. Este número é relativamente baixo, se comparado a outros genomas, podendo chegar até nove vezes o tamanho do genoma humano, como por exemplo, o genoma do pinus *Pinus Lambertiana* que contém 27,6 bilhões de pares de bases (NCBI 2012).

Contribuindo também para o aumento de informações ainda se tem a transcrição do DNA para RNAs funcionais e a tradução dos RNA mensageiros em peptídeos ou proteínas. Todas estas informações nos permitem saber a informação genética armazenada na(s) macromolécula(s) de DNA. Sendo assim, para a análise

e/ou comparação entre genomas, e os desdobramentos que esses genomas produzem, é necessária infraestrutura computacional que garanta armazenamento, processamento e visualização destas informações.

Diversas ferramentas, tais como Arena3D¹, o BioLayout *Express3D*², Cytoscape³ e o NAViGaTOR⁴, que demonstram redes complexas estão disponíveis na Internet, porém, na maioria dos casos, as possibilidades de se colher e filtrar informações e realizar análises a partir do grafo que geram são limitadas.

Algumas destas ferramentas foram desenvolvidas com o uso de equações matemáticas, a fim de garantir melhor visualização da informação. Entretanto, quando se trata de uma massiva entrada de informações, a navegação e a visualização são comprometidas devido à poluição visual gerada na representação da informação, resultando em uma má experiência para o usuário.

Comumente as redes expressam dados de centenas, ou até milhares, interações. Assim, ao se obter uma rede de grande magnitude, sua resolução fica comprometida, fazendo com que a visualização e a identificação precisa dos nós que a compõem seja prejudicada e, conseqüentemente, levando à perda de informações.

Com base no levantamento dos problemas citados e com o intuito de facilitar, expandir e integrar com informações externas contidas nos bancos de dados biológicos, este trabalho tem como objetivo a construção de uma ferramenta Web, que conta com técnicas de visualização da informação, sendo responsiva e 3D, para demonstrar, de maneira interativa, uma rede de predições de alvos de RNAs curtos regulatórios.

1.1 OBJETIVOS

Esta seção apresenta os principais pontos que este trabalho prioriza e objetiva no decorrer do desenvolvimento.

¹ <http://arena3d.org>

² <http://biolayout.org>

³ <http://www.cytoscape.org>

⁴ <http://ophid.utoronto.ca/navigator/index.html>

1.1.1 Objetivo geral

Desenvolver uma ferramenta Web de interface responsiva e 3D, que, usando técnicas de visualização da informação, demonstre de maneira interativa uma rede de RNAs curtos regulatórios e seus alvos.

1.1.2 Objetivos específicos

Os seguintes objetivos específicos foram elencados como forma de se atingir o objetivo geral deste trabalho:

- Verificar e interpretar as informações fornecidas pelo algoritmo de predição de alvos de RNAs curtos, usado como entrada de dados neste projeto;
- Determinar quais dados serão mostrados através do visualizador;
- Definir a melhor forma de visualização dos dados gerados;
- Implementar o processo de análise e geração do grafo de visualização de dados;
- Conectar com banco de dados biológico e disponibilizar informações dos alvos selecionados;
- Implementar técnicas de design responsivo na interface do usuário;
- Permitir interação entre Usuário e Sistema.

1.2 JUSTIFICATIVA E RELEVÂNCIA DO TRABALHO

A visualização de informação é uma estratégia poderosa para organizar e analisar uma grande quantidade de dados (TAO *et al.*, 2004). A partir desta premissa, se faz necessário ferramentas que proporcionam ambientes gráficos diferenciados, que facilitem a compreensão das informações.

No estudo e aplicação da visualização de informação, considera-se a análise realizada pelo usuário, a partir de um grafo apresentado, no intuito de extrair conhecimentos sem a necessidade de outra fonte de pesquisa e este é um dos aspectos abordados no trabalho, que visa a centralização do máximo de informações no grafo gerado por ferramentas especialmente desenvolvidas.

O presente trabalho é uma contribuição, que utiliza os conceitos de visualização da informação, para a saída de um algoritmo de predição de RNAs curtos em procariotos. Pretende-se possibilitar uma visão mais didática da rede de interações gerada e, para isso, a ferramenta foi desenvolvida para visualização em três dimensões e com o intuito de facilitar a compreensão e exposição das informações.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

A estrutura deste trabalho está organizada através de capítulos. No primeiro capítulo foi apresentada a Introdução, na qual se têm os objetivos, geral e específicos, justificativa e explanação da organização. No segundo capítulo é apresentado a revisão bibliográfica referente à visualização da informação, redes de interação, MVC, Web services, RNA e algoritmo de predições. O terceiro capítulo aborda os materiais e métodos utilizados para desenvolvimento do software. O quarto capítulo contém os resultados obtidos e, por fim, no quinto capítulo é apresentada a conclusão do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta um breve estudo da literatura, abordando os principais tópicos relacionados e conceitos básicos, a fim de proporcionar um melhor entendimento do trabalho pelo leitor.

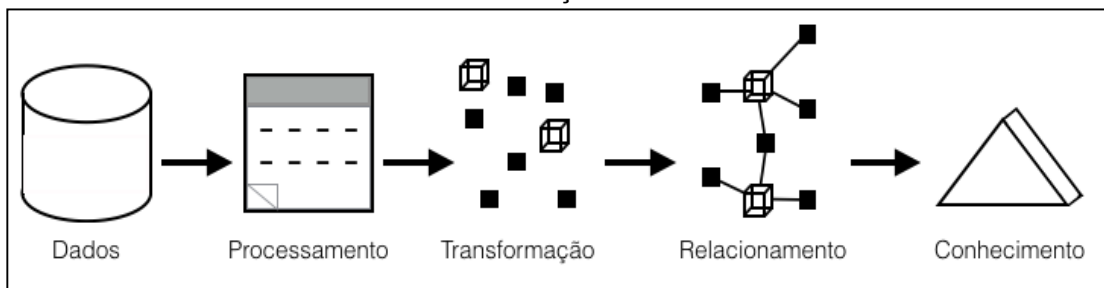
2.1 VISUALIZAÇÃO DA INFORMAÇÃO

O termo visualização da informação é normalmente aplicado para representações visuais de uma coleção, em larga escala, de informações tais como, arquivos, bibliotecas, banco de dados bibliográficos, e redes de relacionamentos de Internet (FRIENDLY, 2009).

A apresentação de informações deve favorecer a percepção do usuário que acessa, todos os dias, grandes volumes de informação, dentre elas, muitas se tornam irrelevantes e desnecessárias, dessa forma ocorre a sobrecarga de informações. Há necessidade não só de uma análise seletiva anterior em relação às informações a serem obtidas, como também da qualidade dos dados trabalhados para a representação de visualização da informação (DIAS, 2007).

A visualização da informação conecta diretamente com o sistema visual e o cérebro humano, o que a torna uma ferramenta essencial para a compreensão e comunicação da informação e capacita o ser humano a realizar a transformação do dado bruto em conhecimento (SCHROEDER; MARTIN; LORENSEN, 2003). Este fluxo de eventos para gerar conhecimentos está apresentado na Figura 1, na qual os dados processados são transformados em símbolos ou objetos, identificando então suas relações e padrões.

FIGURA 1 - PASSOS DA TRANSFORMAÇÃO DE DADOS EM CONHECIMENTO



Os dados brutos passam por um processamento, em que são ordenados e identificados os padrões das informações. No terceiro estágio são criados objetos para representar esses resultados. Os relacionamentos entre os objetos são estabelecidos e, por fim, o usuário pode analisar e extrair conhecimento.

Os estágios da construção da representação consistem na coleta e armazenar os dados, processa-los para transformar os dados em algo mais fácil de manipular. Mapeia-se os dados designados para sua representação visual, na qual é realizado por algoritmos computacionais que produzem uma imagem no monitor.

O processo de visualização pode conter altos níveis de interatividade, por exemplo, em visualizações de dados em 3D onde o usuário pode “voar” a um diferente ponto de vista a fim de melhorar o entendimento da estrutura gerada, alternando com o uso do mouse pode interagir e selecionar o que julga mais interessante. O maior dilema é qual o melhor jeito de transformar uma informação em algo que seja entendível pelas pessoas (WARE, 2013).

Segundo Tufte (2013) uma representação gráfica revela suas informações, então ela deverá:

- Mostrar os dados;
- Induzir o observador a pensar sobre a essência em vez da metodologia, design gráfico, a tecnologia da produção do gráfico ou qualquer outra coisa;
- Evitar distorções do que a informação tem a dizer;
- Apresentar muitos números em um espaço pequeno;
- Fazer uma grande quantidade de dados serem coerentes;
- Encorajar o olho a comparar diferentes pedaços de informação;
- Revelar os dados em diversos níveis de detalhes;
- Servir razoavelmente ao proposito: descrição, exploração, tabulação ou decoração;
- Estar estreitamente integrado com as descrições estatísticas e verbais do conjunto de dados.

De acordo com Freitas e Wagner (1993) há cinco classes principais de representações visuais que surgiram para determinar a melhor maneira de fornecer uma dada informação, sendo eles: gráficos e tabelas, ícones, mapas, diagramas e redes. Cada classe com sua respectiva utilização é demonstrada na Quadro 1.

QUADRO 1 - CLASSES DE REPRESENTAÇÕES VISUAIS

Classe	Utilização
Gráficos/Tabelas	Representação da distribuição no espaço domínio, da dependência/correlação entre atributos.
Ícones	Representação de entidades num contexto, representação de grupos de atributos de diversos tipos.
Mapas	Representação de campos escalares ou categorias, de linhas de contorno de regiões.
Diagramas	Representação de sequência ou referência.
Redes	Representação de relacionamentos diversos.

Fonte: Adaptado de FREITAS, *et al.* (2001)

Muitos dos elementos em uma representação são construídos de maneiras que, por razões perceptuais, tornam uma representação mais fácil de interpretar. As linhas que ligam os vários componentes, por exemplo, são uma notação que é fácil de ler, porque o córtex visual do cérebro contém mecanismos especificamente projetados para procurar contornos contínuos. Outras possíveis notações gráficas para mostrar a conectividade seriam muito menos eficazes. A Figura 2 mostra duas convenções diferentes para demonstrar relações entre entidades. As linhas de ligação à esquerda são muito mais eficazes do que os símbolos à direita (WARE, 2013).

FIGURA 2 - DUAS MANEIRAS GRÁFICAS DE RELACIONAMENTOS ENTRE ENTIDADES



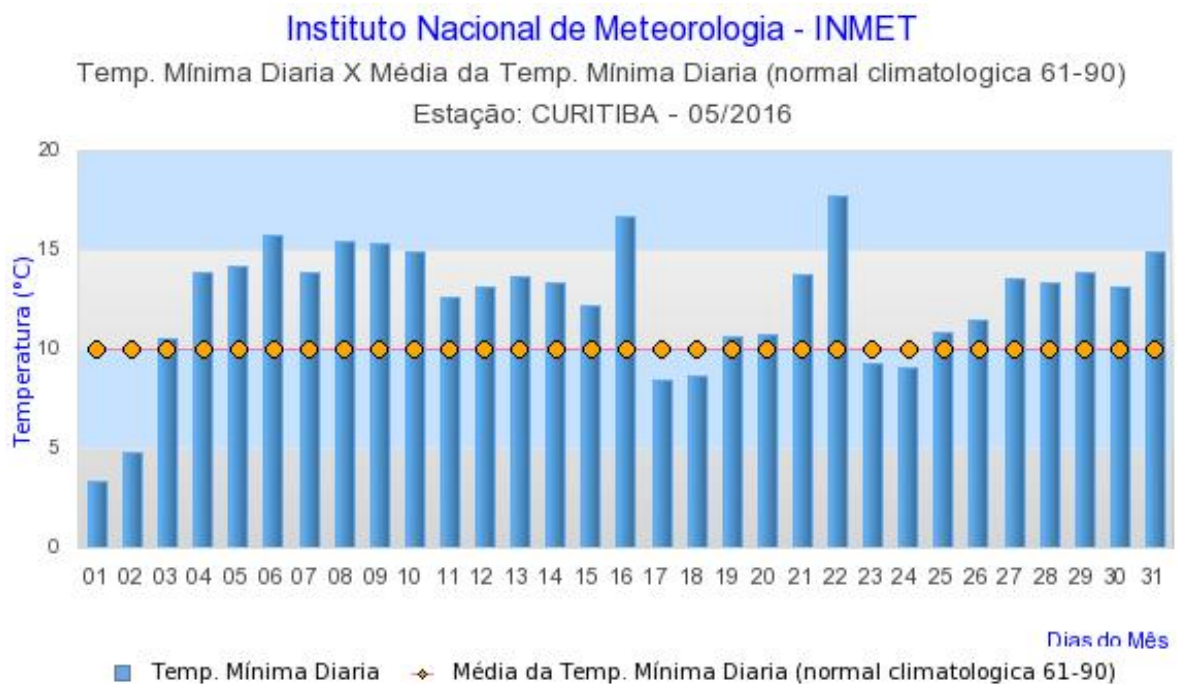
As formas de demonstrar o relacionamento entre elementos, de uma representação gráfica, podem simplificar e agilizar a identificação da informação apresentada. A demonstração da esquerda conecta os elementos, deixando intuitivo o relacionamento. A demonstração a direita utiliza de padrões geométricos e cores para passar a ideia de semelhança, e por sua vez o relacionamento.

Fonte: WARE (2013).

Como podemos ver a seguir, a aplicação da Visualização da Informação pode ocorrer em estatísticas climáticas, gerenciamento doméstico de finanças, manutenção de uma rede local empresarial, demonstração do *status* de um aparelho e até na representação de doenças humanas.

Através de um histórico das medições da temperatura diário, podemos montar um gráfico (Figura 3) com a média do mês de maio de 2016.

FIGURA 3 - MEDIÇÃO DIÁRIA DA TEMPERATURA DE CURITIBA NO MÊS DE MAIO DE 2016.



Temperaturas diárias de Curitiba, durante o mês de maio de 2016, onde se percebe a variação no eixo X em graus Celsius, e no eixo Y os dias. O gráfico também mostra a média das temperaturas no mês.

Fonte: INMET (2016).

A Tabela 1 exemplifica um conjunto de valores que caracterizam as finanças no período de um ano. Com uma simples análise se identifica as despesas fixas e as

variáveis, e com esses dados é possível alcançar algumas conclusões, como em qual mês houve maior gasto.

TABELA 1 - SIMULAÇÃO DE DESPESAS

	Aluguel	Combustível	Colégio	Mercado	Poupança	TOTAL
JAN	R\$600,00	R\$289,00	R\$450,00	R\$500,00	R\$350,00	R\$2189,00
FEV	R\$600,00	R\$290,00	R\$450,00	R\$490,00	R\$250,00	R\$2080,00
MAR	R\$600,00	R\$310,00	R\$450,00	R\$710,00	R\$300,00	R\$2370,00
ABR	R\$600,00	R\$273,00	R\$450,00	R\$520,00	R\$350,00	R\$2193,00
MAI	R\$600,00	R\$400,00	R\$450,00	R\$415,00	R\$200,00	R\$2065,00
JUN	R\$600,00	R\$260,00	R\$450,00	R\$570,00	R\$100,00	R\$1980,00
JUL	R\$600,00	R\$315,00	R\$450,00	R\$490,00	R\$300,00	R\$2155,00
AGO	R\$600,00	R\$340,00	R\$450,00	R\$340,00	R\$200,00	R\$1930,00
SET	R\$600,00	R\$280,00	R\$450,00	R\$620,00	R\$250,00	R\$2200,00
OUT	R\$600,00	R\$270,00	R\$450,00	R\$510,00	R\$280,00	R\$2110,00
NOV	R\$600,00	R\$300,00	R\$450,00	R\$660,00	R\$220,00	R\$2230,00
DEZ	R\$600,00	R\$420,00	R\$450,00	R\$850,00	R\$50,00	R\$2370,00

Fonte - O autor (2016).

As figuras 4 e 5, por sua vez, possibilitam visão do todo, analisar profundamente e extrair informações como ordenar os gastos por suas relevâncias, observar o total, quais têm maior peso e demandam mais atenção.

FIGURA 4 - PORCENTAGEM DE CADA DESPESA EM RELAÇÃO AO TOTAL DE DESPESAS

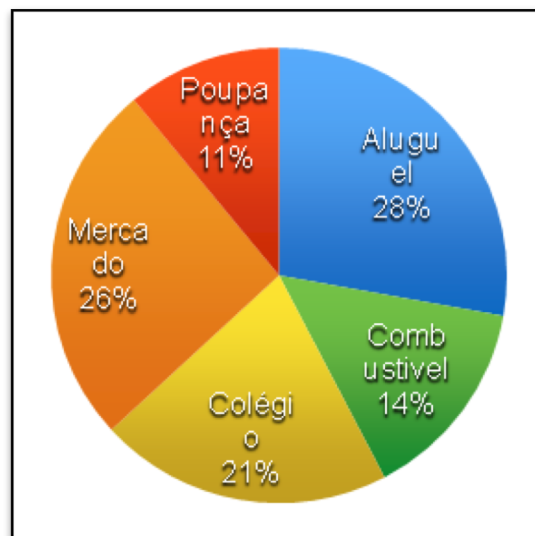
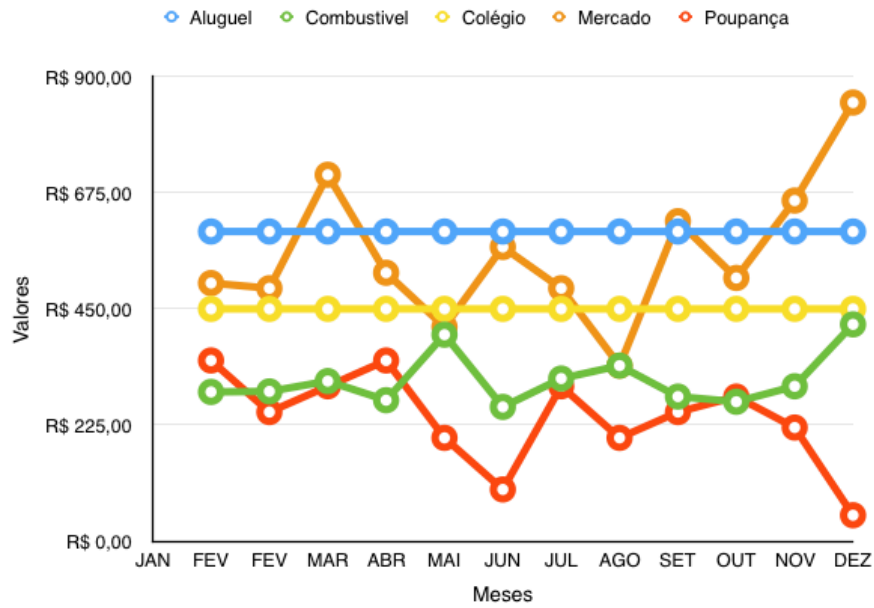


Gráfico tipo pizza que representa cada despesa (Tabela 1), em porcentagem, pelo total.

Fonte: O Autor (2016)

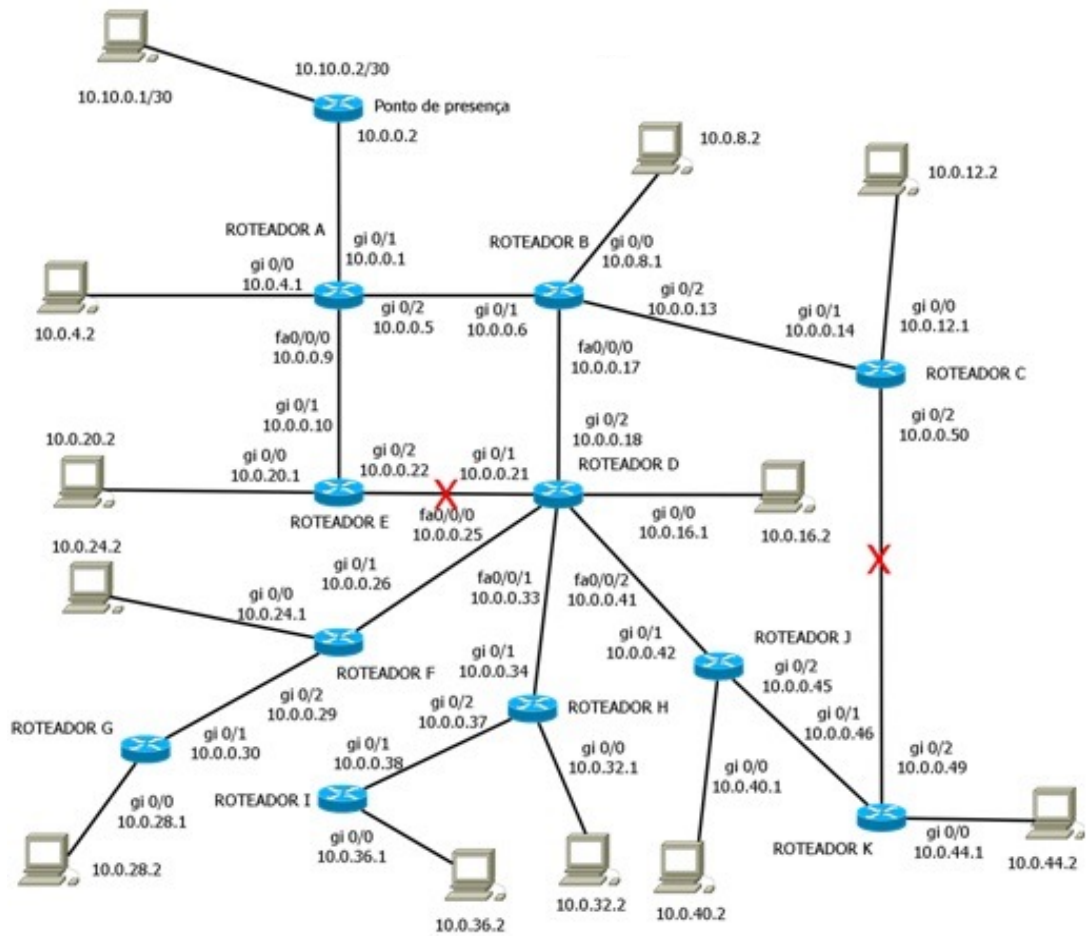
FIGURA 5 - VARIAÇÕES DAS DESPESAS AO LONGO DO TEMPO



Fonte: O autor (2016)

Utilizando a visualização da informação podemos verificar uma estrutura de ligação de uma rede de computadores, onde em caso de defeito ou perda de conexão, a manutenção pode ocorrer de maneira mais fácil. A Figura 6 demonstra um diagrama de uma rede de computadores.

FIGURA 6 - DIAGRAMA DE UMA REDE DE COMPUTADORES

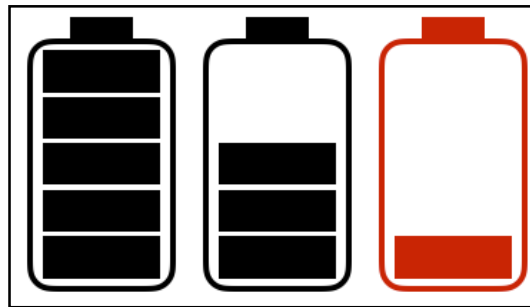


Demonstração de uma rede local de computadores separados por roteadores e onde as conexões são físicas. Cada computador recebe um endereço único para identificação e no diagrama constam duas rupturas no cabeamento (identificadas com o X vermelho) definindo então o local para manutenção.

Fonte: Adaptado de VASCONCELOS (2015)

A classe de representação visual ícone, como o exemplo citado anteriormente no capítulo de Introdução deste trabalho, pode representar uma ideia, status ou até mesmo um aviso ao usuário. A Figura 7 demonstra o estado de bateria cheia, pela metade e pouca carga.

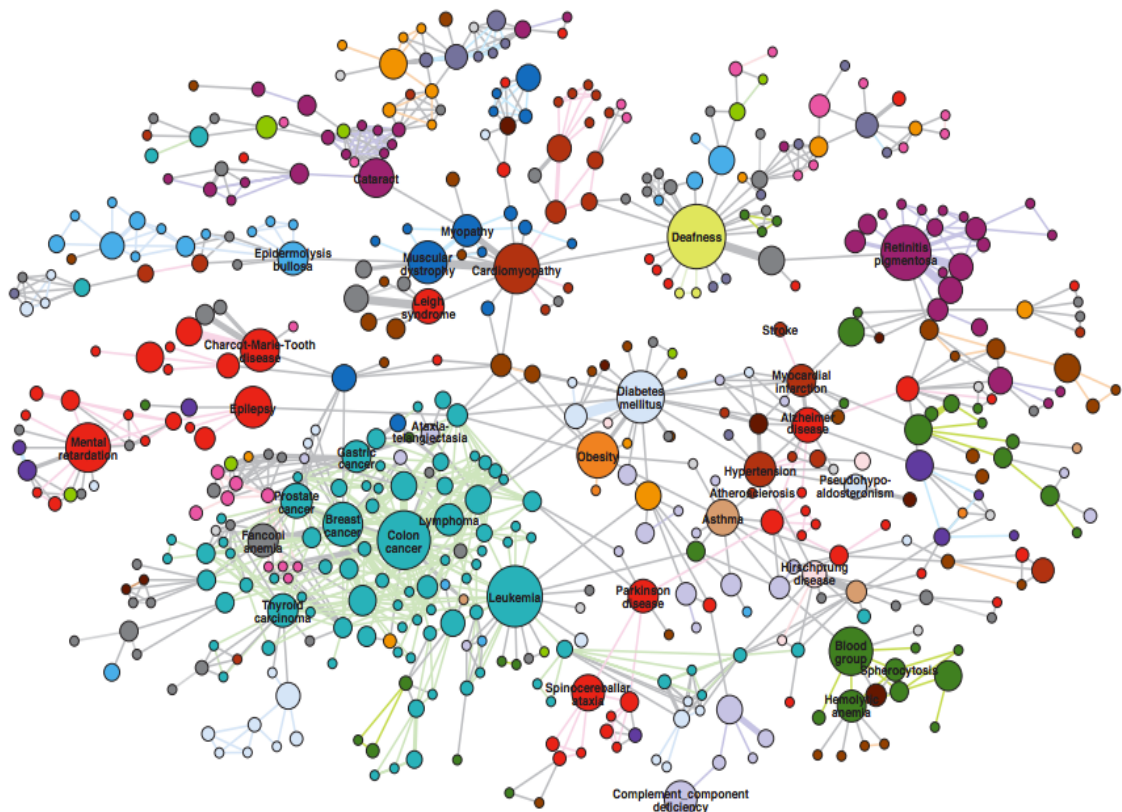
FIGURA 7 - VARIAÇÕES DO ÍCONE DE BATERIA DE UM SMARTPHONE



Fonte: O autor (2016)

A representação visual do tipo rede demonstra ligações entre informações, como ilustrado pela Figura 8. Tem-se uma rede que relaciona doenças a genes humanos, desta maneira o usuário pode encontrar padrões e obter conhecimentos ao analisar a rede.

FIGURA 8 - REDE DE DOENÇAS HUMANAS



Fonte: Adaptado de GOH et al., 2007.

Grafo do tipo rede que representa doenças humanas, cada cor representa a classe da doença (Ósseas, Câncer, cardiovascular, etc.) e o tamanho dos nós correspondem ao número de genes que estão envolvidos na doença correspondente. Os nós que foram nomeados no grafo possuem mais de 10 genes associados.

O objetivo final do projeto de visualização interativa é otimizar as aplicações para que elas ajudem a realizar o trabalho cognitivo de forma mais eficiente. Otimizar um sistema requer que possamos ter pelo menos alguma concepção de valor. Usamos visualizações porque nos ajudam a resolver problemas mais rapidamente ou melhor, ou nos deixam aprender algo novo (WARE, 2013).

As interfaces de visualização são desenvolvidas para mostrar e divulgar informações a uma grande e potencial comunidade. Portanto, é necessário que a visualização da informação seja efetiva e bem planejada, pois não se sabe qual usuário irá interagir, havendo nível de educação variado, presença ou ausência de experiência, bem como capacidades e necessidades diferentes. Finalmente, a representação visual deve ser acessível, independente do grupo de usuários e se adaptar às necessidades específicas e problemas de cada um (GERSHON; EICK; CARD, 1998).

2.2 REDES DE INTERAÇÃO

Em sua forma mais simples, o grafo é definido como uma porção de vértices ligados em pares por arestas. Os vértices são referenciados como nós e as linhas como arestas. Muitos objetos de estudos sejam físicos, biológicos e sociais podem ser observados através das redes (NEWMAN, 2010).

Esse método gráfico representa o impacto e dependências entre seus componentes, juntando informações e as interligando, possibilitando a análise das informações representadas através de influências, ligações e proximidade.

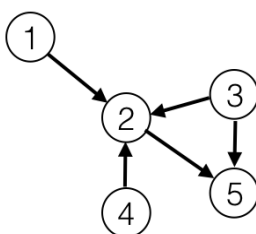
Redes são universais e surgem em numerosas aplicações e em diversos formatos. Dentre as classes, redes físicas são, talvez, o tipo mais comum e de mais fácil identificação. Entre essas redes físicas, as redes de transporte são possivelmente as mais visíveis no cotidiano. Assim, o usuário tende a visualizar obviamente e rapidamente linhas férreas ou rodovias. Porém, a rede pode ter uma visualização não tão óbvia, por exemplo, o planejamento da distribuição complexa de cargas por uma

cidade e suas decisões de logística de entregas (AHUJA, 1993).

O uso das redes também é aplicado para demonstração de informações biológicas, pode-se ter como exemplo as redes genéticas regulatórias, que informam o efeito de proteínas regulatórias, sobre a expressão de genes específicos, ou seja, sobre a expressão de seus alvos. Neste caso, os nós da rede representam as proteínas ou o equivalente do gene e a aresta ligada direta do gene A para o gene B indica que A regula a expressão em B. Uma rede mais sofisticada pode representar se a interação é de promoção ou inibição dando a rede dois tipos distintos de arestas (NEWMAN, 2010).

As redes podem ser representadas de duas maneiras, redes diretas ou indiretas. Nas redes diretas há um sentido de interação, ou seja, um elemento está relacionado diretamente a outro (Figura 9) e apresentam a seguinte expressão matemática no grafo $G = (N, A)$, sendo N o nó e A a aresta. Têm-se para $N = \{1, 2, 3, 4, 5\}$ e $A = \{(1, 2), (2, 5), (3, 2), (3, 5), (4, 2)\}$ e os valores da aresta (A) são representados como (i, j) , onde o sentido da interação é de i para j (AHUJA, 1993).

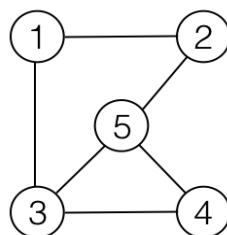
FIGURA 9 - REDE DIRETA



Fonte: O autor (2016)

Para as redes indiretas, tem-se a mesma definição da forma direta, com exceção aos valores de i e j , que podem ser representados na aresta como (i, j) ou (j, i) , pois se entende que a aresta se comporta como uma via de mão dupla (AHUJA, 1993). A sua representação gráfica é demonstrada na Figura 10.

FIGURA 10 - REDE INDIRECTA



Fonte - O autor (2016)

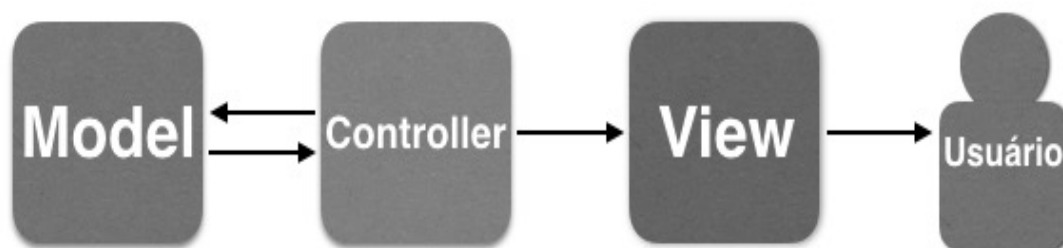
Quanto ao uso do modelo gráfico de redes na Bioinformática, tem-se um fator complicador, muito comum, relacionado à enorme quantidade de nós e arestas no grafo que tende a dificultar o entendimento e a análise do mesmo (NEWMAN, M. E. J., 2010). Não obstante, esse recurso é muito útil na Bioinformática para representar diferentes tipos de redes biológicas como, por exemplo, redes metabólicas, de sinalização celular, transcrição, interação proteína-proteína, interações genes-doenças. Na Figura 6, citada anteriormente, está representada a conexão dos genes às doenças. Quando mutados, estes genes podem causar, ou contribuir, para o desenvolvimento de determinadas doenças e as redes de interação medicamentosas em que as drogas estão ligadas aos seus alvos (MAAYAN, 2011).

A análise de redes de sistemas biológicos vem ganhando espaço, como um método de análise e compreensão dos dados, mesmo que o resultado apresentado ainda seja uma visão geral, devido à grande complexidade que essas redes apresentam.

2.3 MVC

O MVC é um padrão de projeto de desenvolvimento de software muito utilizado. Tem sua sigla formada a partir dos três componentes *Model*, *View* e *Controller* (Figura 11) e é considerado um padrão de arquitetura de desenvolvimento de software (HUSTINAWATI; HIMAWAN; LATIFAH, 2014). Dentre os componentes do MVC, a *View* é responsável por mostrar toda a informação para ao usuário e junto com o *Controller* definir os processos e assimilar as interações da aplicação. Por fim, o *Model* fornece as informações que serão apresentadas através da *View* e que poderão ser alteradas em resposta das interações feitas pelo usuário (LEFF; RAYFIELD, 2001).

FIGURA 11 - ESTRUTURA MVC



Fonte: O autor (2016)

A arquitetura do MVC se baseia em 3 superclasses que centralizam toda a estrutura e padronizam o código, segundo Hustinawati, Himawan e Latifah (2014):

Model: Representa um banco de dados, incluindo a estrutura de relacionamentos entre as tabelas. A principal função do *Model* é manipular os dados dessas tabelas, recuperar as informações do banco de dados e inserir ou alterar dados.

View: Renderiza os dados provenientes do *Model* e transmite as ações feitas pelo usuário demonstrando assim as mudanças que ocorreram. Centraliza toda a parte visual e de apresentação em um lugar, facilitando a manutenção e alterações, sem afetar a lógica e os dados.

Controller: Define e detém os métodos que determinam os comportamentos que ocorrem no sistema, e assim através do seu molde promove o desenrolar das ações do usuário para o *Model*. Atua próximo a *View*, pois a cada interação feita, através desta, uma resposta será fornecida.

As vantagens de se utilizar esta estrutura estão na possibilidade de múltiplas linguagens de programação no mesmo projeto, fácil manutenção da aplicação, e alteração da aplicação sem alterar a estrutura ou sua lógica.

No desenvolvimento deste projeto o padrão MVC foi adotado como uma forma de simplificar e organizar os recursos e linguagens utilizados.

2.4 WEB SERVICE

Web service é uma solução tecnológica, que promove a interação entre computadores e integração entre sistemas por meio da rede (W3C, 2004). Seu

desenvolvimento surgiu da necessidade de integrar operações de diferentes linguagens e estabelecer um padrão nesta comunicação.

Localizados e podendo ser acessados em toda Internet, os *Web services* desempenham funções que variam de simples requisições, até processos complicados, como por exemplo prover cotações de mercado de ações ou uma transação de cartão de crédito. Quando implementado, outras aplicações ou até mesmo outros *Web services* podem utilizar este serviço (TIDWELL, 2000).

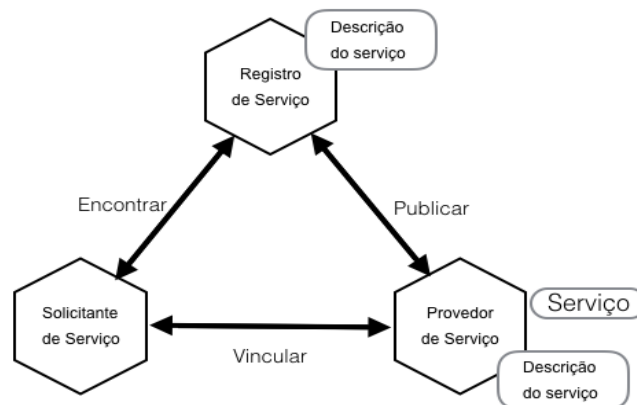
Existem dois protocolos que o *Web service* pode utilizar para sua comunicação, o SOAP (*Service Oriented Architecture Protocol*) e o REST (*Representation State Transfer*).

2.4.1 SOAP

A arquitetura do SOAP é baseada na interação de 3 elementos, o provedor de serviço, registro de serviço e o solicitante de serviço. Estas interações envolvem operações de publicar, encontrar e vincular os serviços transmitidos (KREGGER, 2001).

O provedor de serviço define a descrição do serviço para o *Web service* e o publica para o solicitante de serviço ou para o registro de serviço. O solicitante de serviço utiliza a operação de encontrar para recuperar a descrição do serviço localmente, ou através do registro de serviço, e usa a descrição do serviço para vincular com o provedor de serviço e interagir com a implementação do *Web service*. As funções do provedor de serviço e do solicitante de serviço são de construtores lógicos. Esses elementos e operações com suas interações estão representados na Figura 12 (KREGGER, 2001).

FIGURA 12 - ELEMENTOS DO WEB SERVICE E OPERAÇÕES



Fonte: Adaptado de KREGER 2001

A arquitetura SOAP utiliza como descritor de serviços o WSDL (*Web Services Description Language*), que consiste em um documento, em XML (*EXtensible Markup Language*), que age na forma de um contrato entre o cliente e o servidor. Uma vez estabelecido, o provedor do serviço e o solicitante podem trocar dados padronizados, independente das plataformas ou aplicações em que operam (ALTOVA, 2006).

2.4.2 REST

O REST é um protocolo suportado pelo protocolo HTTP (*HyperText Transfer Protocol*), que possibilita acesso através de uma URL (*Uniform Resource Locator*) (FIELDING, 2000). Diferente do SOAP, não impõe restrições ao formato da mensagem, apesar de existir a possibilidade de usar qualquer formato e comumente são utilizados os formatos JSON (*JavaScript Object Notation*) ou XML para codificar/decodificar (realizar o *parser*) as mensagens trocadas entre sistemas diferentes.

O protocolo REST tem a flexibilidade como maior vantagem. Deixando livre para o desenvolvedor escolher o formato da comunicação do sistema, de acordo com sua necessidade. *Web services* que utilizam REST são mais “leves”, pois sua estrutura contém menos dependências de aplicações mediadoras, como por exemplo servidores de aplicações, e portanto, são mais rápidos (RODRIGUEZ, 2008).

Os *Web services* que possuem estrutura implementada, são comumente chamados de RESTful (RODRIGUEZ, 2008).

2.4.3 XML

O XML é uma linguagem de marcação, sua sintaxe consiste em descrever os dados contidos no documento através de *tags* através de texto, que, podem ser customizáveis fornecendo assim o tipo de informação que carrega, possibilitando a transferência de informações pela Internet e o seu manuseio, independentemente de qual aplicação ou sistema operacional. Desta maneira é facilmente lida e interpretada por aplicações e humanos (ALTOVA, 2006). A Figura 13 demonstra a linguagem e sua gramática.

FIGURA 13 - DEMONSTRAÇÃO DE DADOS NA LINGUAGEM XML

```
<widget>
  <debug>on</debug>
  <window title="Sample Konfabulator Widget">
    <name>main_window</name>
    <width>500</width>
    <height>500</height>
  </window>
  <image src="Images/Sun.png" name="sun1">
    <hOffset>250</hOffset>
    <vOffset>250</vOffset>
    <alignment>center</alignment>
  </image>
  <text data="Click Here" size="36" style="bold">
    <name>text1 </name>
    <hOffset>250</hOffset>
    <vOffset>100</vOffset>
    <alignment>center</alignment>
    <onMouseUp>
      sun1.opacity = (sun1.opacity / 100) * 90;
    </onMouseUp>
  </text>
</widget>
```

Fonte: Adaptado de JSON, 2016.

2.4.4 JSON

O formato JSON é uma formatação de troca de dados computacionais baseados em texto (CROCKFORD, 2006).

Possui a mesma finalidade do formato XML, em funcionalidade, o que difere é sua sintaxe, que permite a possibilidade de uma maior velocidade de escrita e leitura. Sua principal vantagem é integração com o JavaScript (W3SCHOOLS, 2016).

Sua gramática é baseada numa sequência de símbolos. Esses símbolos são compostos por caracteres especiais, que determinam em qual estrutura será montado estes dados, se no formato de *array* (utilizando colchetes []) ou objeto (utilizando chaves { }). Dentro da estrutura há os separadores para nome (símbolo de dois pontos :) e valor dos dados separados por vírgula (CROCKFORD, 2006).

FIGURA 14 - DEMONSTRAÇÃO DE DADOS NA LINGUAGEM JSON

```
{
  "widget": {
    "debug": "on",
    "window": {
      "title": "Sample Konfabulator Widget",
      "name": "main_window",
      "width": 500,
      "height": 500
    },
    "image": {
      "src": "Images/Sun.png",
      "name": "sun1",
      "hOffset": 250,
      "vOffset": 250,
      "alignment": "center"
    },
    "text": {
      "data": "Click Here",
      "size": 36,
      "style": "bold",
      "name": "text1",
      "hOffset": 250,
      "vOffset": 100,
      "alignment": "center",
      "onMouseUp": "sun1.opacity = (sun1.opacity / 100) * 90;"
    }
  }
}
```

Fonte: Adaptado de JSON, 2016.

A rede desenvolvida neste trabalho utiliza o *web service* TogoWS, com o protocolo REST, para ter as informações provenientes dos bancos de dados biológicos. Estes dados retornam para a ferramenta em JSON como parser.

2.5 RNA

A expressão dos genes, de forma constitutiva ou em um dado momento metabólico (expressão induzida), se dá através da síntese de moléculas de RNA funcional e este evento é conhecido como transcrição (NELSON; COX, 2014). As fitas de RNA e DNA podem parecer muito similares à primeira vista, no entanto são diferenciadas em alguns aspectos conforme o Quadro 2 retrata.

QUADRO 2 - PRINCIPAIS DIFERENÇAS ENTRE DNA E RNA

	DNA	RNA
Estrutura	Cadeia dupla helicoidal	Cadeia simples
Pentose	Desoxirribose	Ribose
Bases	Adenina, Guanina, Citosina e Timina	Adenina, Guanina, Citosina e Uracila
Origem	Replicação	Transcrição
Função	Reservatório de informação genética	Codifica produto funcional

Fonte: O autor (2016)

O RNA, de uma forma geral, é a única macromolécula com as funções de armazenamento, transmissão de informações e a catálise, instigando estudos sobre o seu papel como intermediário químico essencial no desenvolvimento da vida neste planeta (NELSON; COX, 2014).

As três classes mais conhecidas de RNA de procariotos e eucariotos são:

- RNA mensageiro (RNAm): polímero de fita simples que contém as informações para síntese de peptídeos ou proteínas;
- RNA transportador (RNAt): tem a função de transferir/acoplar o aminoácido correspondente à informação contida no RNAm durante a tradução;
- RNA ribossomal (RNAr): Componente dos ribossomos que são as estruturas em que ocorre a tradução.

Além desses, há outros tipos, cuja presença em procariotos e/ou eucariotos bem com a função, estão apresentados no Quadro 3. Aqueles RNA que não pertencem à classe do RNAs mensageiro são denominados RNAs não codificadores e dentre esses estão os smallRNAs ou RNAs curtos, que codificam para um grupo

específico de moléculas que interferem de forma positiva ou negativa na transcrição (ZAHA *et al.*; 2014).

QUADRO 3 – CLASSES DE RNAs E SUAS RESPECTIVAS FUNÇÕES

Tipo	Classe	Procarioto	Eucarioto	Função
Codificador	mRNA (mensageiro)	Presença	Presença	Sequências síntese de proteínas
Não Codificador	rRNA (ribossômico)	Presença 70S(50S+30S)	Presença 80S(60S+40S)	Síntese de Proteínas
	tRNA (transportador)	Presença	Presença	Transportador de aminoácido ao local de síntese proteínas
	snoRNA (RNA nucleolar)	Ausência	Presença	Maturação de ribossomos e RNAs transportadores
	snRNA (RNA nuclear)	Presença	Presença	Síntese do mRNA eucariótico
	sRNA (RNA pequeno)	Presença	Presença	Modula a transcrição e a estabilidade de mRNAs
	CRISPR (RNA CRISPR)	Presença	Ausência	Sistema de defesa bacteriana
	miRNA (micro RNA)	Ausência	Presença	Silenciamento gênico pela inibição da tradução e cliva mRNAs
	siRNA (RNA interferência)	Ausência	Presença	Silenciamento gênico
	piRNA (RNA liga-se PIWI)	Ausência	Presença	Silenciamento gênico em células germinativas
	lncRNA (RNA longo)	Ausência	Presença	Regulação biológica e processos de desenvolvimento de doenças

Fonte: Modificado de ZAHA *et al.*; 2014.

2.5.1 RNA CURTO

Os RNA curtos ou RNAs não codificadores procarióticos foram descobertos nos anos 70 e tem função distinta dos RNAs mensageiros e de transferência. Associados a funções celulares diversas, os RNA curtos regulam a estabilidade do RNAm, influenciando a expressão gênica (WASSERMANN, 2001). Isso indica que os procariotos usam estas moléculas para regular sua fisiologia e rapidamente adaptar-se às mudanças do ambiente. Portanto, o entendimento do RNA curto não codificador foi expandido para uma nova área de pesquisas, uma vez que essas pequenas moléculas estão envolvidas em diferentes processos celulares e possibilitam a organização de redes de interações que adicionam novo nível de complexidade à fisiologia celular (MASSÉ *et al.*, 2003).

Até 2011 foram detectados 80 RNAs curtos na bactéria *Escherichia coli* (MODI *et al.*, 2011) e o esperado é que este número seja crescente para *E. coli* e outros

procariotos, uma vez que diversas investigações envolvendo bactérias e arqueias estão em curso. Geralmente, os genes codificadores de RNAs curtos apresentam sequências de promotores tipo sigma 70 e sequência de terminadores Rho-Independente (PICHON; FELDEN, 2008).

Além dos procariotos, RNAs curtos podem ser encontrados em eucariotos (Quadro 4), e nesses organismos são denominados micro RNAs. Da mesma forma, atuam como reguladores de processos celulares em que predomina a interação com mRNA alvos.

O pareamento entre o RNA curto e o RNAm alvo, pode levar a ativação ou inibição da tradução, estabilização ou degradação do RNAm. Frequentemente, essas interações envolvem a participação da proteína Hfq (LIU; CAMILLI, 2010). Outra forma de regulação promovida pelos ncRNAs envolve a alteração da atividade de proteínas, como é o caso dos RNAs 6S e CsrB em *E. coli* (BABITZKE; ROMEO, 2007).

Um alvo de um dado RNA curto não codificante, seja um RNAm ou proteína, é considerado como verdadeiro desde que faça interação física com o RNA e, em consequência, tenha sua função modificada. Portanto, se um RNAm, que deveria ser traduzido, interagir com um RNA curto não codificante, poderá ter seu sítio de interação com o ribossomo bloqueado e não será traduzido. Ao contrário, poderá ter seu sítio de interação com o ribossomo liberado e então será traduzido. Além disso, a interação RNA curto não codificante/mRNA alvo, mediada pela proteína Hfq, direciona o alvo para a degradação (VOGEL; WAGNER, 2007).

Com o aumento da importância dos RNAs curtos não codificantes, algoritmos foram desenvolvidos para a predição de sequências candidatas que se baseiam na localização de sequências de DNA altamente conservadas e presentes em regiões intergênicas de espécies relacionadas. Alguns algoritmos também se baseiam na identificação de sinais de início e de término de transcrição distantes 50-600 pares de nucleotídeos, características observadas em ncRNAs previamente conhecidos por métodos de detecção direta (4.5S RNA, tmRNA, 6S RNA, RNaseP RNA, and Spot42 RNA) ou durante estudos da interação de ncRNAs com proteínas (OxyS RNA, Crp Tic RNA, CsrB RNA, e GcvB) (ALTUVIA, 2007). Atualmente, há também grande interesse na predição das possíveis interações entre o RNA curto não codificador e seu(s) alvo(s) e novos algoritmos estão disponíveis para fim como o TargetRNA2, o RNApredator e o Inta RNA (WANG et al., 2015).

2.6 ALGORITMOS DE PREDIÇÕES

Os primeiros estudos em predição de RNA curto foram realizados em *E. coli*. O procedimento padrão deste processo de predição consiste em 4 etapas.

Primeiro, das regiões intergênicas. Identifica-se as sequências conservadas, então as mesmas são agrupadas e comparadas por pareamento ou alinhamento múltiplo. Finalmente, esses alinhamentos são pontuados a partir de um algoritmo de predições escolhido (BACKOFEN; HESS, 2010).

Os algoritmos podem ser utilizados em abordagens mais específicas para predição de alvos, como o TargetRNA (TJADEN, 2008), que atribui um valor ao pareamento de bases, calculando também valores de energia livre mínima, resultando em uma lista ordenada de melhores candidatos a alvos.

Outros programas de predição como TargetRNA2, RNApredator, sRNATarget, sTarPicker e CopraRNA, seguem a mesma premissa. A estratégia para descoberta do alvo primeiramente ocorre na predição pela bioinformática, para depois ter a comprovação por meio de experimentos (WANG *et al.*, 2015).

O resultado do algoritmo de predição pode ser inserido em um *script* ou *software*, gerando gráficos complexos que demonstrem as interações dos alvos no formato de rede, como os *scripts* de montagem da rede ARACNE, CLR, MRNET e RN (Relevance Network).

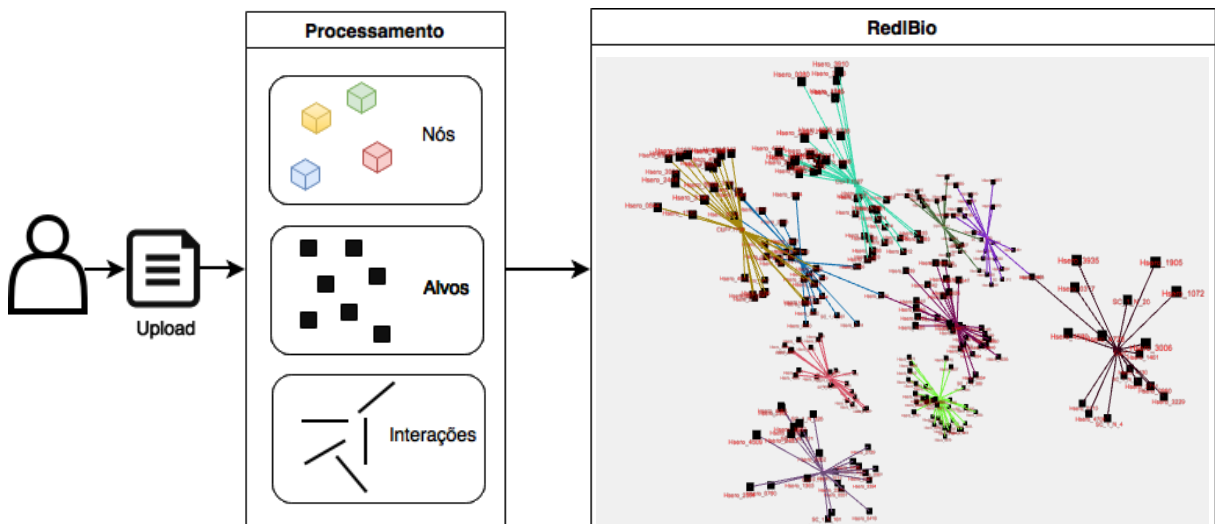
3 MATERIAL E MÉTODOS

Este capítulo apresenta os instrumentos utilizados para o planejamento e construção da ferramenta, com a denominação RedIBio (Redes de Interação Biológicas). Também contém uma explicação das estratégias adotadas para a entrada de dados e de desenvolvimento.

3.1 VISÃO GERAL

A RedIBio entrega ao usuário uma rede de interações interativa, responsiva e 3D capturando as informações contidas no arquivo enviado pelo usuário e separando em três grupos os nós, os alvos e as interações (Figura 15). Esses grupos são organizados e são criados objetos para representa-los na forma gráfica. Para cada nó, e suas devidas interações, é escolhida uma cor para sua representação. Por fim, a RedIBio une os objetos, os nomeia e disponibiliza uma rede de interações para o usuário, que pode navegar e obter mais informações dos alvos.

FIGURA 15 - FLUXOGRAMA DE FUNCIONAMENTO DA REDIBIO



O usuário efetua um *upload* do seu arquivo (xls ou xlsx) que é processado e identifica-se os nós, alvos e suas interações. Esses elementos são transformados em objetos, cubos para os nós, quadrados para os alvos e linhas para as interações. A combinação dos objetos forma a rede de interações apresentada pela RedIBio.

Fonte: O Autor (2016).

3.2 AMBIENTE SOFTWARE E LINGUAGEM DE PROGRAMAÇÃO

Esta seção apresenta uma breve descrição das ferramentas que compõem a RedIBio e seus usos.

3.2.1 HTML

HTML é a linguagem de marcação para a criação de páginas na Internet. Possibilita a publicação de documentos *on-line* com cabeçalhos, textos, tabelas, listas, imagens, etc. Permite também a interligação entre documentos pelos *hypertext links* (W3C, 2016). O uso do HTML ocorre através dos navegadores que interpretam a linguagem e a reproduzem visualmente.

Para a RedIBio foi de suma importância o uso da quinta versão do HTML, o HTML 5, que possui elementos gráficos avançados, permitindo o desenvolvimento em três dimensões, e possibilitando a construção de uma ferramenta na qual o usuário não precise instalar nenhum software para visualizar o gráfico.

3.2.2 PHP

O PHP (acrônimo recursivo para PHP: Hypertext Preprocessor)⁵ é uma linguagem de código aberto, que se adequa para o desenvolvimento Web.

Sua principal vantagem é poder mesclar com o HTML para inserir os códigos. Esta ação ocorre pelas instruções de processamento (tags) de início e fim `<?php e ?>`.

Comparado com o Javascript, que também é usado neste trabalho, o PHP faz todo o trabalho no servidor, fazendo com que o navegador do usuário apenas mostre o resultado junto com o HTML, não permitindo qualquer tipo de acesso ao código pelo usuário.

⁵ <https://secure.php.net/>

3.2.3 JavaScript

O JavaScript⁶ é uma linguagem voltada, assim como o PHP, para interpretação dos navegadores *Web*. Porém, se diferencia por ser executado no lado do usuário, controlando os elementos contidos na página e da interação com o usuário.

A linguagem manipula a estrutura de HTML e CSS (Cascading Style Sheets), conferindo dinamicidade aos elementos contidos na página. Como exemplo a interação de um menu, revelando seu conteúdo, ao passar sobre o mesmo com a seta do *mouse*.

Através da linguagem, com o auxílio da biblioteca *Three.js*, a rede de interações biológicas é construída.

3.2.3.1 *Three.js*

Esta biblioteca⁷ tem como finalidade o auxílio na criação de ambientes em 3D, fornecendo suporte com os elementos básicos:

- Cena: Espaço onde os objetos estarão embutidos;
- Câmera: Literalmente os "olhos" do usuário, reproduz o campo de visão, com direito a poder configurar ângulos e iluminações;
- Renderizador: Classe do código que materializa o conteúdo adicionado na Cena e põe a Câmera em seu devido lugar;

A partir do básico, é possível a criação de gráficos. Na página de exemplos da biblioteca se pode ter um vislumbre das possibilidades, desde mapas onde é possível explorar, réplicas de carros, *tour* em 360° de um museu, até representações gráficas da linha do tempo do ano de 2012 (GOOGLE, 2012).

Foi utilizada esta biblioteca para gerar objetos de acordo com a entrada do usuário, alocar estes objetos no espaço destinado através de coordenadas (eixos X,

⁶ <https://www.javascript.com/>

⁷ <https://threejs.org/>

Y e Z), interpretar as interações e desenhar as linhas entre os nós e seus alvos e por fim, a movimentação que o usuário pode fazer dentro do gráfico.

3.2.3.2 JQuery

O *JQuery*⁸ é uma biblioteca em JavaScript que simplifica e agiliza a manipulação de elementos HTML, disponibilizando para o desenvolvedor a possibilidade de animar e lidar com eventos da página.

Na RedIBio esta biblioteca foi utilizada para animar as abas de informações para o usuário interagir.

3.2.4 CodeIgniter

O CodeIgniter⁹ é um *Framework* para desenvolvimento de Web sites funcionando como uma caixa de ferramentas. Tem o objetivo de tornar um projeto mais rápido e ágil na construção. Conta com bibliotecas para tarefas rotineiras, assim como uma estrutura lógica e uma interface simples, reduzindo a quantidade de código necessárias para estas tarefas.

3.2.5 TogoWS REST API

Esta API tem o intuito de localizar, retornar, analisar e converter as entradas dos banco de dados biológicos. Com integração nativa a estes bancos, o TogoWS os acessa de maneira unificada, através de notação URI (KATAYAMA; NAKAO; TAKAGI, 2010).

Desta maneira, o sistema que se utiliza desta API pode executar as requisições incluindo na URI. Para fazer uma busca tem-se a seguinte configuração:

[http://togows.dbcls.jp/search/DATABASE/query+string\[/offset,limit\]\[.format\]](http://togows.dbcls.jp/search/DATABASE/query+string[/offset,limit][.format])

⁸ <https://jquery.com/>

⁹ <https://www.codeigniter.com/>

Em DATABASE é substituído pelo nome do banco em que se deseja fazer a consulta. Sendo também como requisito mínimo uma *query* de busca. Esta *query* deve estar de acordo com o banco em que se deseja consultar, por exemplo se selecionado o banco Uniprot, pode-se usar como busca “*lung cancer*”. Os demais termos da URI não são obrigatórios sendo então OFFSET - o número inicial desta pesquisa (valor padrão, caso não configurado, é 1), LIMIT - quantos resultados vai retornar (valor padrão é 50) e o FORMAT - onde se escolhe o formato desta resposta, se será em HTML ou JSON e caso não selecionado o padrão adotado é HTML.

Para acessar o registro de uma entrada do banco de dados a seguinte configuração é necessária:

http://togows.org/entry/database/entry_id[,entry_id2,...][/field][.format]

A diferença, em relação a interação anterior, se encontra no primeiro campo informado, *entry*, que determina qual será o tipo de ação a ser realizado pelo TogoWS. O campo DATABASE se comporta da mesma maneira que a busca. Para ENTRY_ID deve-se usar o código de identificação condizente utilizado pelo banco selecionado, pode-se também fazer múltiplas consultas separando os códigos por vírgulas. Na opção FIELD, não obrigatória, pode-se determinar qual campo do registro à retornar, caso nenhum seja selecionado, são retornados todos os campos. Por fim, o FORMAT se comporta de maneira idêntica ao sistema de busca.

O uso do TogoWS, pela RedlBio, ocorre na busca e captura das informações dos alvos, do grafo, com quais o usuário interage.

3.3 DESENVOLVIMENTO

Nesta seção são tratadas as questões do desenvolvimento da ferramenta, sendo divididas em cinco etapas, de acordo com a Figura 16.

A primeira etapa, representada no diagrama pela cor azul, corresponde a interface da ferramenta, onde o usuário pode fazer o *upload* de seu arquivo a fim de gerar seu grafo.

A segunda etapa, cor verde, consiste na captura das informações contidas no arquivo de *upload*, e sua ordenação. Todas as ações tomadas, para esta etapa, foram desenvolvidas na linguagem PHP.

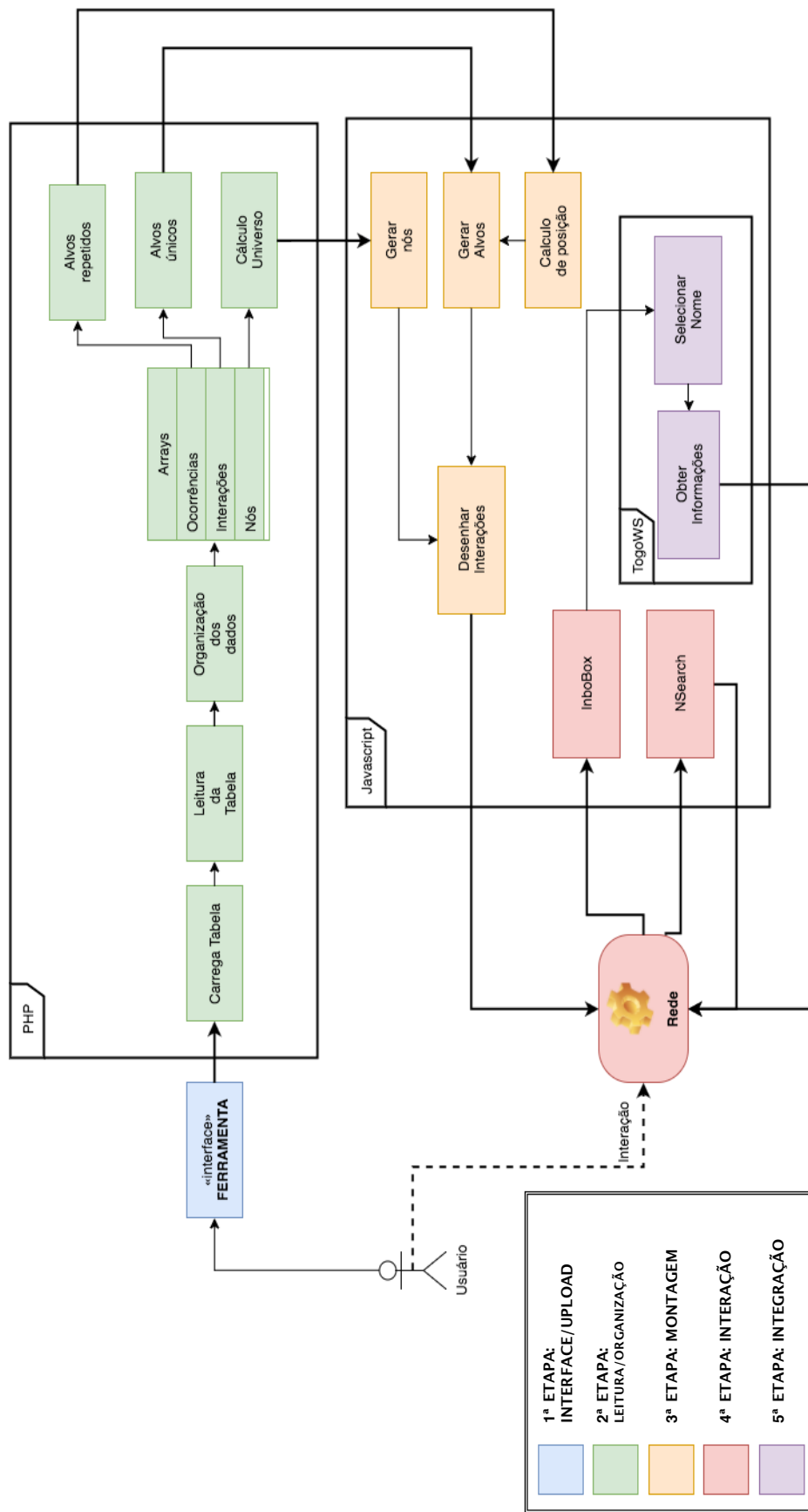
Na sequência, a terceira etapa (laranja) contém as ações tomadas pela RedIBio para a montagem da rede, a linguagem utilizada para esta etapa é a JavaScript.

Com o grafo criado o usuário poderá interagir com o mesmo. As ações da quarta etapa estão demonstradas com a cor vermelha. Além da interação de navegação, duas opções estão disponíveis para o usuário o Infobox e o NSearch.

O Infobox conecta a RedIBio ao TogoWS e executa as ações em roxo do diagrama, que estão descritas na quinta, e última etapa, a de integração.

Cada etapa é abordada pelas subseções com maiores informações.

FIGURA 16 - DIAGRAMA DE AÇÕES DA REDIBIO

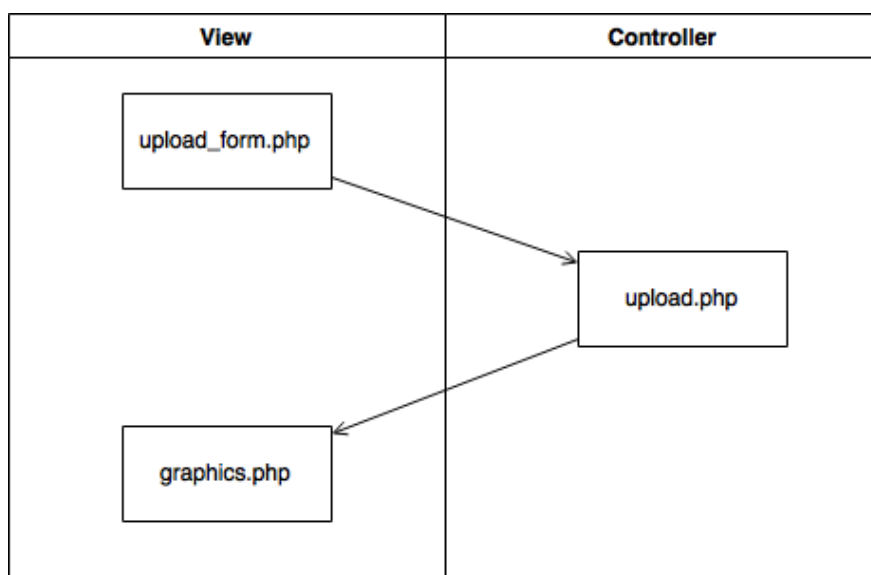


Fonte - O autor (2016)

3.3.1 Interface e Upload

O projeto é organizado em três arquivos, situados entre superclasses *View* e *Controller*. Sua estrutura é apresentada na Figura 17.

FIGURA 17 - ESTRUTURA DOS ARQUIVOS DO PROJETO REDIBIO



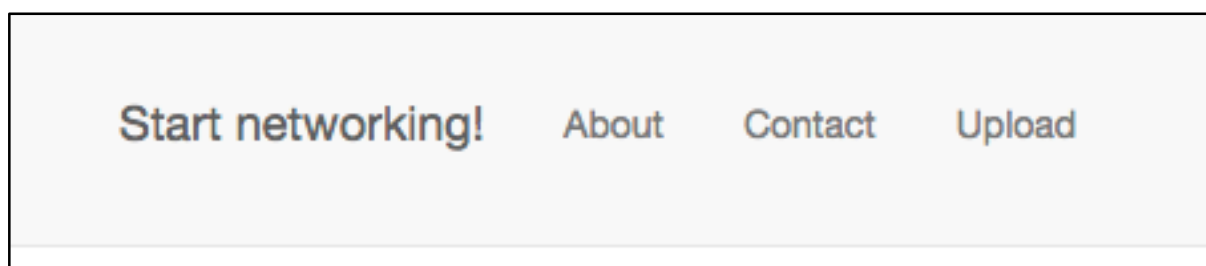
Esquema de seqüências dos arquivos da ferramenta RedIBio e a alternância entre as superclasses, da arquitetura MVC, *View* e *Controller*.

Fonte - O autor (2016)

A interface em que o usuário visualiza ao acessar a RedIBio, se encontra na parte de *View* do projeto organizado pelo CodeIgniter.

O arquivo `upload_form.php` contém o HTML da página principal e o formulário para o *upload*. A página é separada em quatro sessões, disponibilizadas verticalmente, onde o usuário pode deslocar, para cima ou para baixo, ao longo da página, para passar de sessão ou clicar no menu superior, conforme a Figura 18 demonstra.

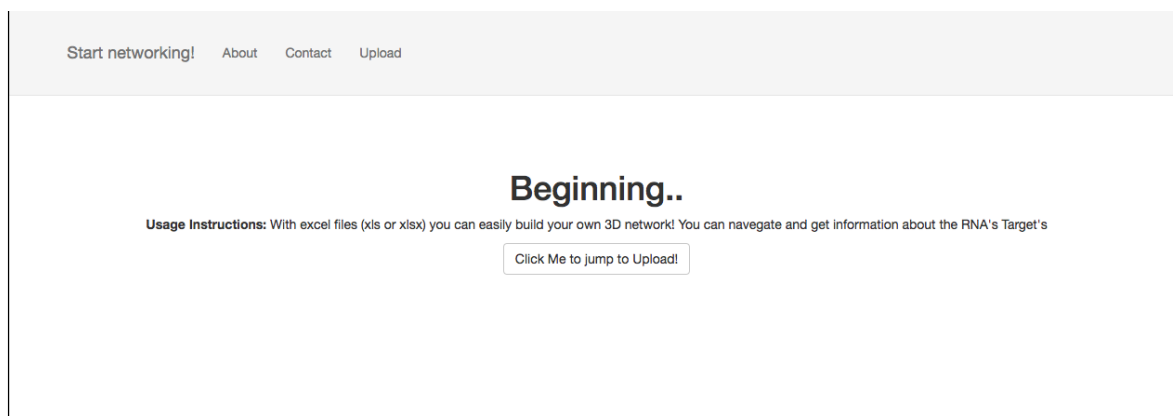
FIGURA 18 - CAPTURA DE TELA DO MENU NA PÁGINA DA REDIBIO



Fonte - O autor (2016)

O primeiro item, *Start networking!* (Figura 19), contém a mensagem inicial juntamente com a instrução de quais extensões de arquivo a RedIBio suporta (.xls e .xlsx).

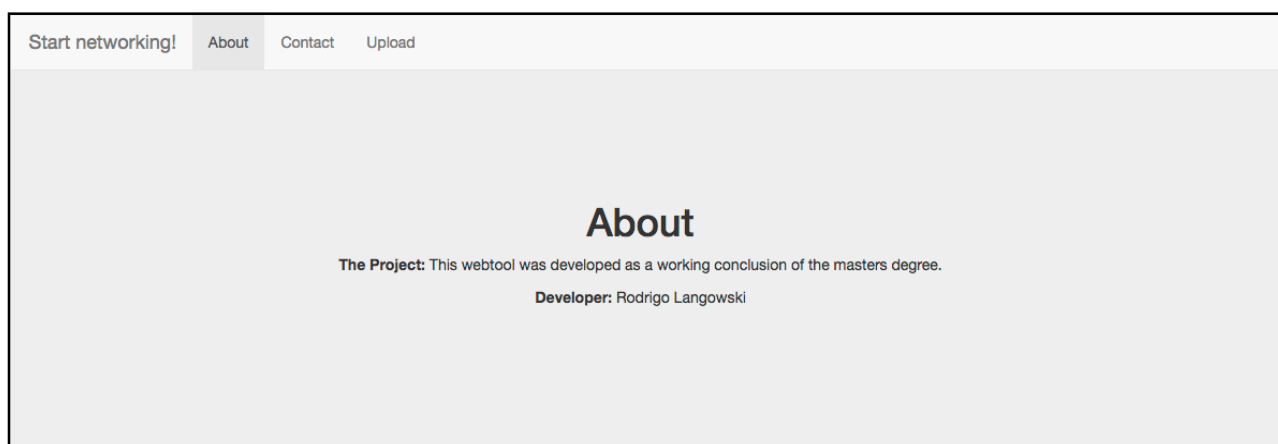
FIGURA 19 - CAPTURA DE TELA DA PÁGINA INICIAL DA REDIBIO



Fonte - O autor (2016)

Já o segundo item, *About* (Figura 20), apresenta as informações de conceituação da ferramenta.

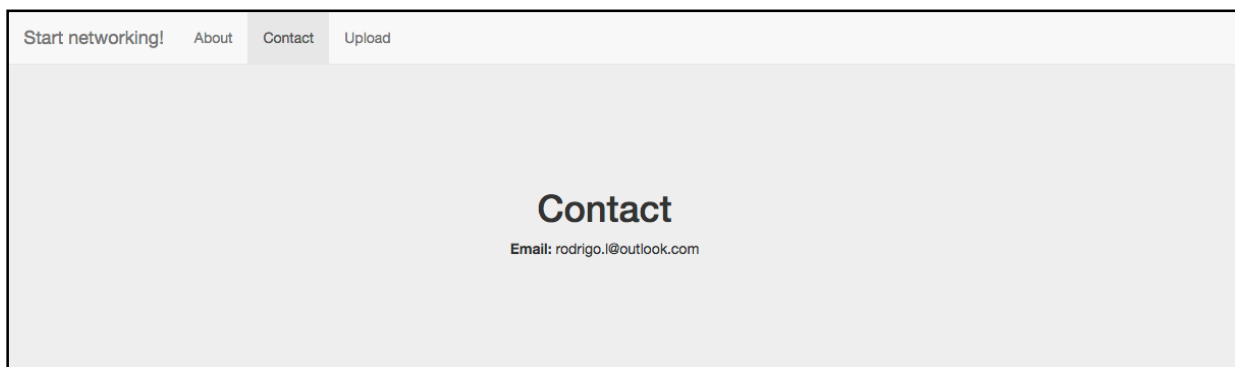
FIGURA 20 - CAPTURA DE TELA DA PÁGINA DE INFORMAÇÕES SOBRE A REDIBIO



Fonte - O autor (2016)

No terceiro item, *Contact* (Figura 21), os dados do desenvolvedor são apresentados.

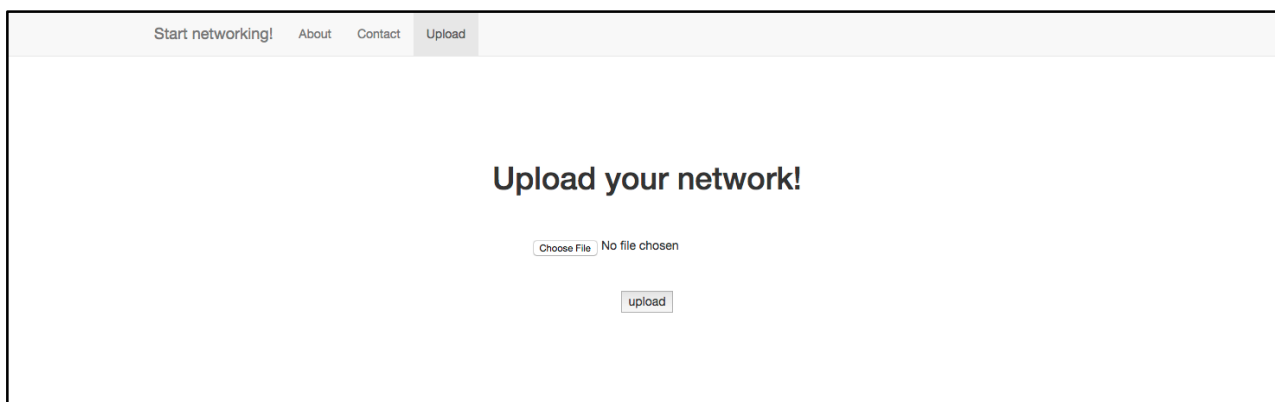
FIGURA 21 - CAPTURA DE TELA DE CONTATO DA PÁGINA DA REDIBIO



Fonte - O Autor.

Finalmente no quarto item, Upload, está o formulário de *upload*. Neste, o usuário pode escolher o arquivo e assim gerar sua rede. Conforme a Figura 22 demonstra.

FIGURA 22 - CAPTURA DE TELA DA SESSÃO DE UPLOAD



Fonte - O autor (2016).

3.3.2 Leitura e Organização

A leitura do arquivo, realizado na etapa de *upload*, ocorre no *Controller* do projeto e com o seu desenvolvimento na linguagem PHP. Após o *upload*, o arquivo é enviado para a função `do_upload()`, onde será validado com as seguintes configurações.

```
$config['upload_path'] = './uploads/';
```

```
$config['allowed_types'] = 'xls|xlsx';
```

O arquivo será temporariamente salvo em um diretório, /uploads/, até completar toda a captura de informações, e logo após isso é excluído.

A ferramenta verifica se o arquivo está de acordo com os tipos permitidos, xls ou xlsx. Caso não for, retorna uma mensagem de erro para a *View* que contem o formulário, informando o erro. Se o arquivo estiver de acordo, é repassado para a função `ReadSheet()`.

O próximo passo é a leitura da tabela, inicialmente capturando os nomes dos nós e dos alvos e montando uma *array* das interações.

```
$interact[] = array("row" => $rowData[0][0], "col" => $tags[($k-1)], "value" =>
number_format($v,3));
```

No passo em que é feita a varredura e a construção do *array*, é realizada uma contagem das ocorrências dos alvos, pois havendo repetições, o alvo não pode ser duplicado na montagem da rede.

Em outro momento é contado a quantidade de nós da rede. Este número influenciará no espaço e na disposição espacial dos objetos, ou seja, quanto mais nós a rede tiver, maior será o espaço ocupado, conseqüentemente a disposição dos objetos da rede deverá ser expandida.

Por fim, todas as informações são repassadas para a *view* `Graphics.php`.

3.3.3 Montagem

A montagem do grafo ocorre no lado *View* do projeto, sendo utilizado a linguagem JavaScript para a construção da rede.

Utilizando a biblioteca `Three.js`, é estabelecido a cena, câmera e o renderizador que irão atuar, bem como suas configurações.

A recepção dos dados provenientes do PHP para o JavaScript ocorre pelo JSON.

```
var jsonNode = <?php echo json_encode($node); ?>;
var jsonInteract = <?php echo json_encode($interactions); ?>;
var jsonOccur = <?php echo json_encode($occurrences); ?>;
```

Com os arrays de nós e interações uma rotina é criada. Para cada nó um cubo é criado e inserido na cena.

```
cube.position.x = (Math.random() * (dimension*10) - ((dimension*10)/2)) * odd ;
```

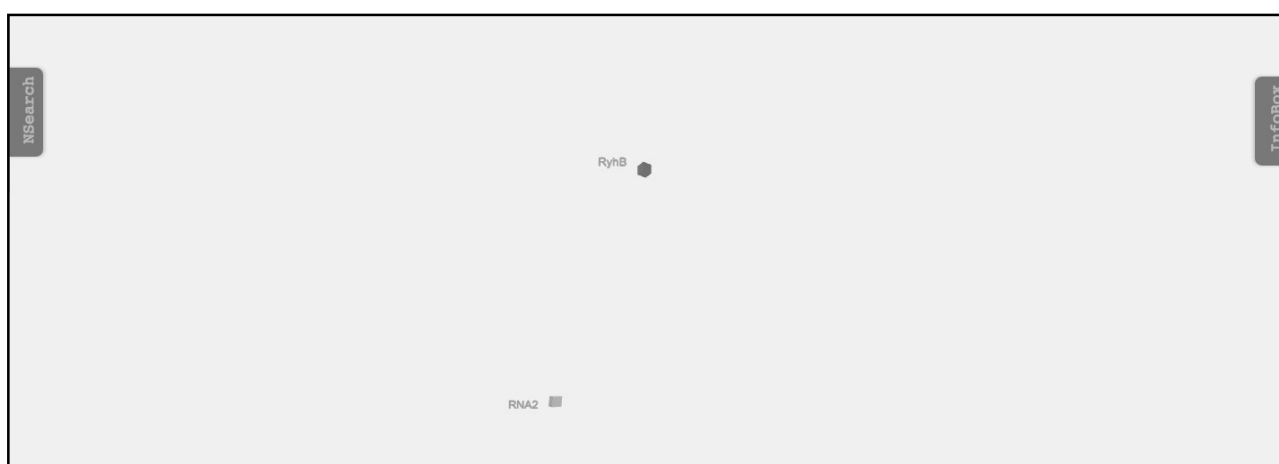
```
cube.position.y = (Math.random() * (dimension * 4)) * odd ;
```

```
cube.position.z = (Math.random() * (dimension * 8) - ((dimension * 8)/2)) * odd ;
```

Seu posicionamento considera um número aleatório, a dimensão prevista pelo total de nós e um valor (*odd*) que varia entre positivo e negativo. A medida em que o nó é criado, também se vincula um *label* contendo o nome do nó (Figura 23).

```
addLabel(name, cube.position);
```

FIGURA 23 - INSERÇÃO DOS NÓS NA CENA



Fonte: O autor (2016)

Com o nó tendo sua posição definida, busca-se no array de interações, os alvos relacionados a este nó. Diferente dos nós, os alvos são representados na rede através de uma imagem em duas dimensões, chamadas de partículas.

Uma nova rotina é feita para gerar as partículas que representarão os alvos que não possuem interação com mais de um nó. Para o posicionamento é utilizado como base inicial a posição do nó.

```
pX = cube.position.x + ((Math.random() * 250) * odd),
```

```
pY = cube.position.y + ((Math.random() * 250) * odd),
```

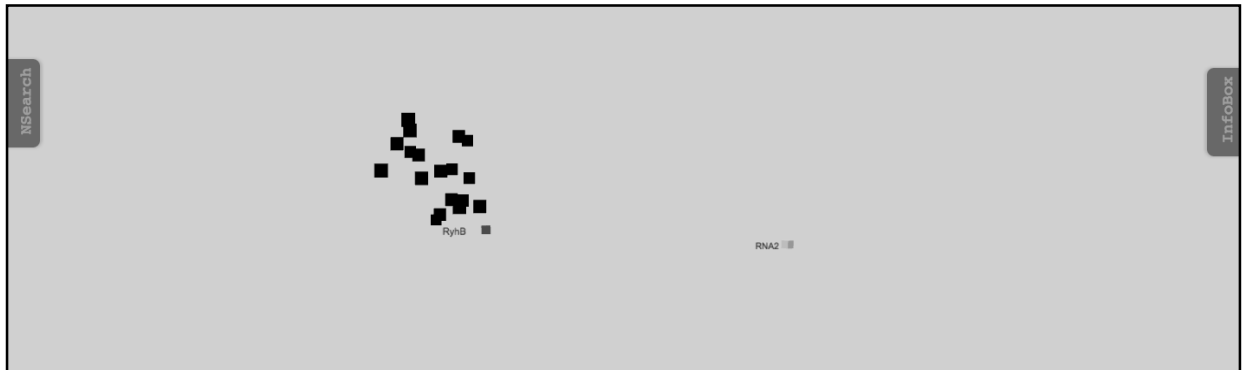
```
pZ = cube.position.z + ((Math.random() * 250) * odd)
```

Além da posição, as partículas também recebem os atributos de nome do alvo.

Cada partícula gerada é adicionada em um sistema de partículas, para que somente no fim da rotina, sejam renderizadas na cena (Figura 24).

```
particleSystem.userData.particles.push( particle );
```

FIGURA 24 - PARTÍCULAS INSERIDAS PRÓXIMAS AO NÓ DE INTERAÇÃO

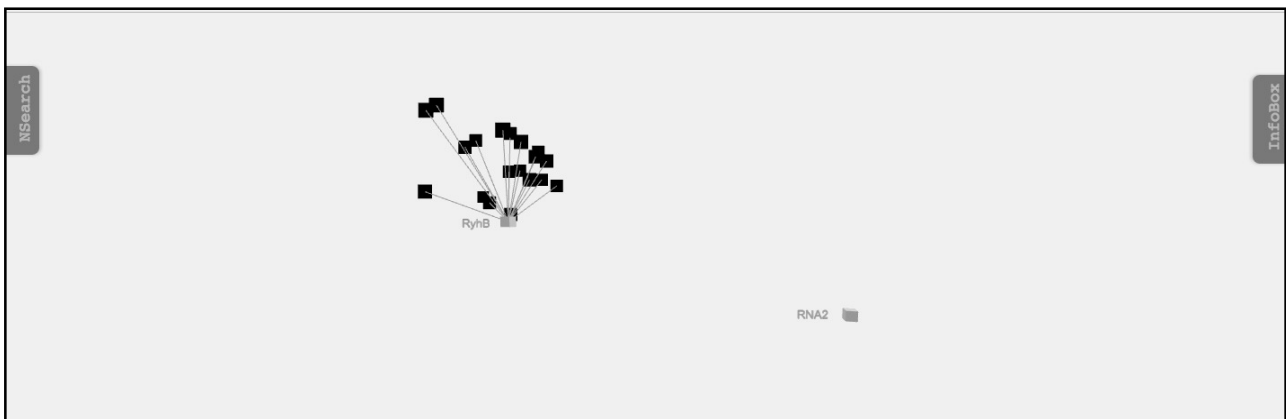


Fonte - O Autor.

O próximo passo está na criação da linha entre o nó e o alvo (Figura 25). Para isso é acionada uma função específica, os dados de entrada da função consistem no nó e alvo em questão.

```
addLine(cube, particle);
```

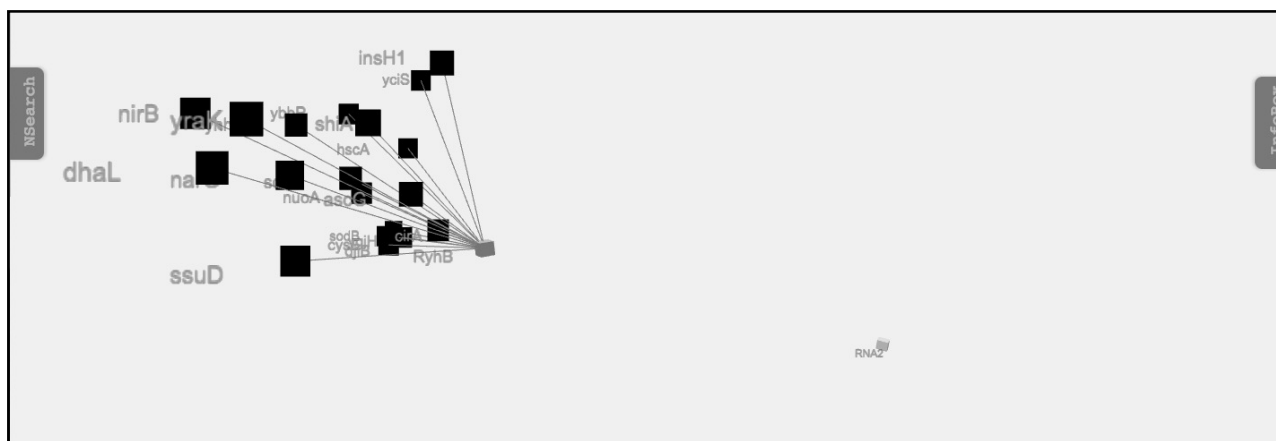
FIGURA 25 - INSERÇÃO DE LINHAS DE INTERAÇÃO ENTRE O NÓ E O ALVO



Fonte - O autor (2016)

E da mesma maneira em que é adicionado um *label* no nó, a partícula recebe a sua (Figura 26).

FIGURA 26 - ADIÇÃO DOS NOMES DOS ALVOS



Fonte - O autor (2016)

O último passo de criação da rede é adicionar, na cena, os alvos que possuem mais de uma interação, para isto uma rotina é criada para, a cada registro no array de ocorrências, enviar o nome do alvo com repetição para a função `addRepTarget()`.

Esta função busca todos os nós relacionados ao alvo, soma as posições e divide pela quantidade, tendo assim a média espacial para adicionar a partícula.

```
allX = allX + splineHelperObjects[nodeObject].position.x;
```

```
allY = allY + splineHelperObjects[nodeObject].position.y;
```

```
allZ = allZ + splineHelperObjects[nodeObject].position.z;
```

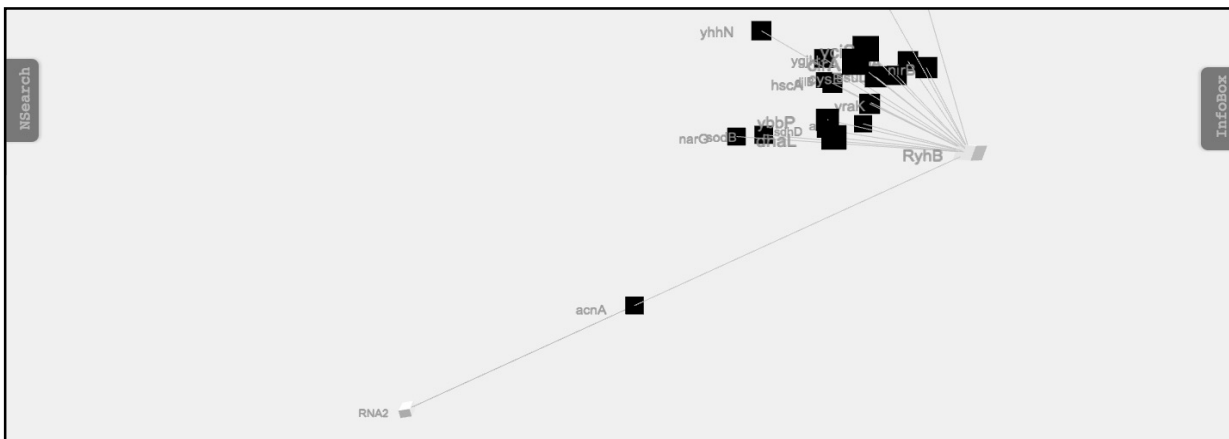
```
particleX = allX/size,
```

```
particleY = allY/size,
```

```
particleZ = allZ/size,
```

Por fim, a estas novas partículas, são desenhadas as linhas de interação e seus *labels* (Figura 27). A função `render()` então renderiza a cena, contendo todos os objetos citados, e a câmera.

FIGURA 27 - ADIÇÃO DE ALVOS DE MAIS DE UMA INTERAÇÃO FORMANDO A REDE FINAL



Fonte - O autor (2016).

3.3.4 Interação

O usuário, nesta etapa, já visualiza a rede e pode navegar por ela através do mouse, Figura 27. As ações, que o usuário usa, são executadas pela função `OrbitControls()`, onde:

- *scroll up/down*: aproximar/afastar a câmera dos objetos.
- botão direito: pressionado permite a movimentação pelos eixos x e y com a câmera.
- botão esquerdo: conta com duas funcionalidades, pressionado permite a rotação da câmera e ao clicar em um alvo aciona a aba InfoBox, onde o usuário obtém as informações do alvo.

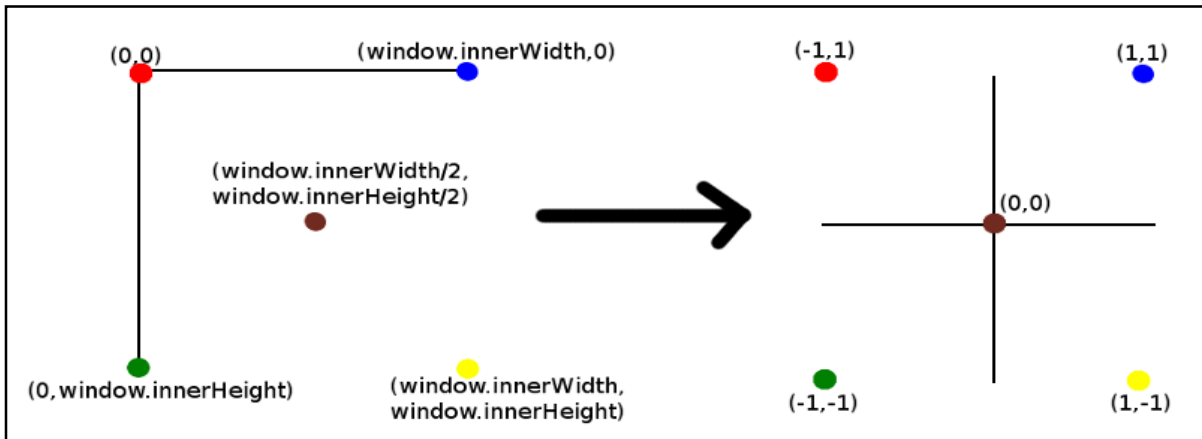
Existem duas abas dispostas na tela para auxiliar o usuário, como descrito anteriormente, a InfoBox e a NSearch.

3.3.4.1 InfoBox

Ao visualizar a rede completa o usuário pode, além de navegar pelo grafo, selecionar os alvos em que deseja a fim de obter informações do mesmo. Com este objetivo a InfoBox disponibiliza estas informações, que são provenientes dos bancos de dados biológicos.

Para localizar o objeto em que o mouse se sobrepõe são necessários alguns passos. Primeiro se calcula a posição em que a seta representa no ambiente 3D, conforme ilustra a Figura 28.

FIGURA 28 - REPRESENTAÇÃO DOS PONTOS DE UM GRÁFICO 2D PARA 3D

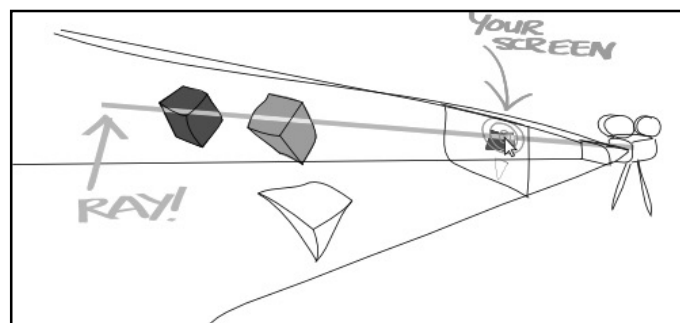


Ao lado esquerdo, as localizações e quais valores adotados para cada ponto, respectivo a sua cor, para um plano cartesiano, utilizado para gráficos 3D. Tem-se o valor de `window.innerHeight` o tamanho da tela usada pelo usuário e `window.innerWidth` a largura.

Fonte: Byte (2016)

Tendo a posição equivalente, um raio (raycaster) é projetado a partir da posição geométrica da tela, Figura 29, e no primeiro contato deste raio com algum objeto, a ferramenta captura o nome deste objeto.

FIGURA 29 - TRAJETÓRIA DO RAIOS A PARTIR DA TELA DO MONITOR ATÉ OS OBJETOS

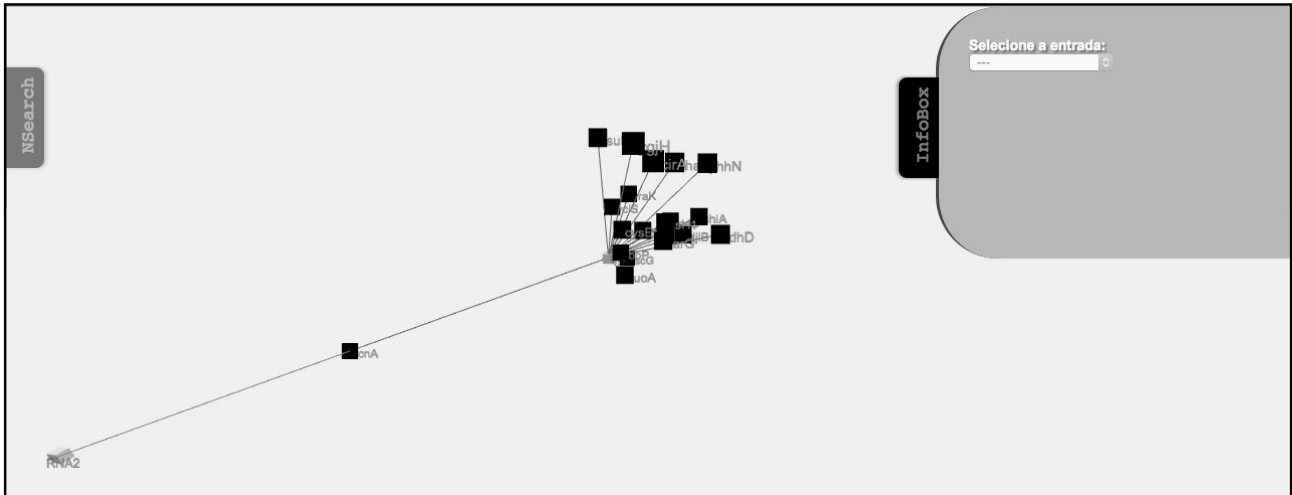


Demonstração do percurso do raio (ray) através dos objetos contidos em um ambiente de 3 dimensões.

Fonte: Penadés (2016)

Tendo o nome do objeto a ser interagido, é acionada a função `callTargetInfo()` para a consulta no banco de dados biológico, conforme a Figura 30. Os detalhes da conexão serão abordados na próxima etapa.

FIGURA 30 - ABA EXPANDE AO SELECIONAR UM ALVO



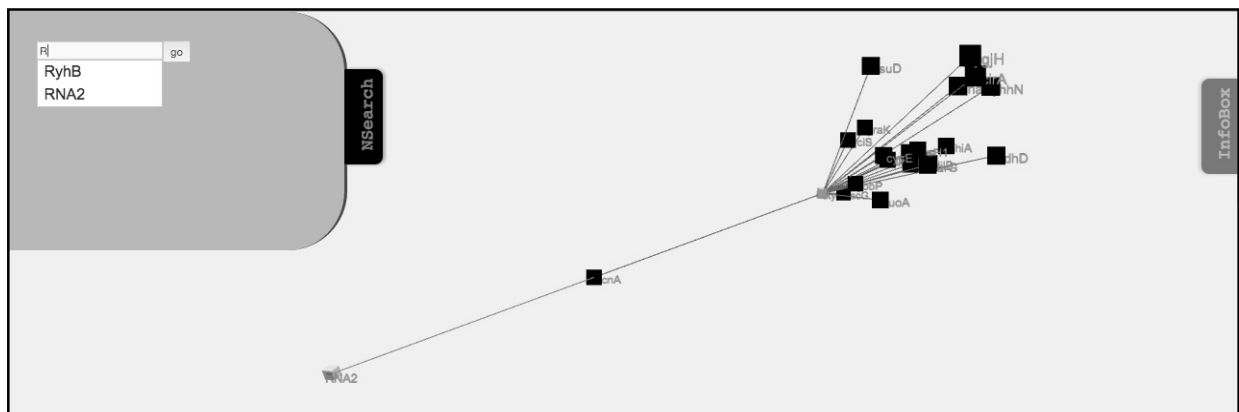
Fonte: O autor (2016)

3.3.4.2 NSearch

Esta função visa a localização de um dos nós da rede, no estudo de caso um RNA curto, centralizando-o na tela. Facilitando assim uma observação das interações deste elemento.

Em sua aba consta uma caixa de texto, onde possui os nomes dos nós contidos na rede, e através de JQUERY a caixa de texto sugere os nós a medida em que o usuário digita, a Figura 31 apresenta está função.

FIGURA 31 - RESPOSTA DA CAIXA DE TEXTO COM O AUTO-COMPLETAR



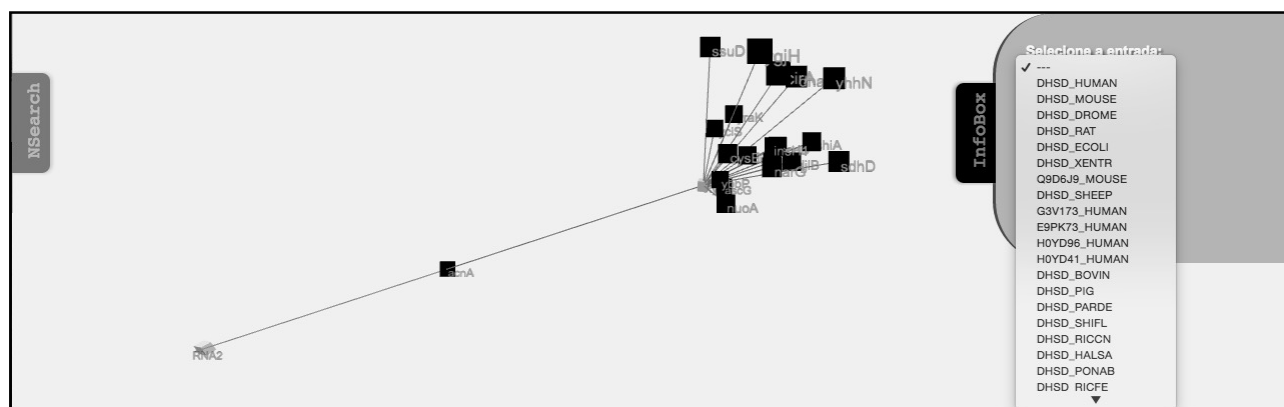
Fonte: O autor (2016)

Assim que o usuário escolhe o nó e aciona o botão GO, a função `focusNode()` é chamada. Esta função por sua vez busca a posição do nó e direciona a câmera para este ponto, deixando em evidência o nó desejado.

3.3.5 Integração com Banco de Dados Biológico

Ao clicar com mouse no alvo, esta função é acionada. Inicialmente a RedIBio se conecta através do TogoWS, com o banco de dados biológico, que retorna os 50 primeiros resultados, Figura 32, da pesquisa utilizando o nome do alvo.

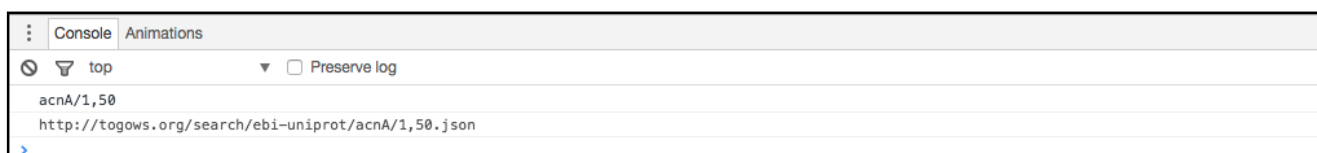
FIGURA 32 - OS RESULTADOS OBTIDOS PELO TOGOWS



Fonte: O autor (2016)

A conexão é realizada na função `callTargetInfo()`, onde é feita uma requisição passando o nome do alvo (Figura 33). O TogoWS retorna em JSON uma lista de entradas, é montado então um *select* em HTML para disponibilizar para o usuário estas opções.

FIGURA 33 - SOLICITAÇÃO DA BUSCA PARA O WEB SERVICE



Captura de tela do console do navegador. Em primeiro momento a RedIBio envia para o *web service* o nome do alvo selecionado (`acnA`) e os valores de configuração para a busca (1,50), onde determina os primeiros cinquenta resultados.

Fonte: O autor (2016)

O usuário deverá então selecionar uma das entradas, uma segunda conexão é realizada, utilizando desta vez o ID da seleção, conforme mostra a Figura 34.

FIGURA 34 - SOLICITAÇÃO DE ENTRADA A PARTIR DO ID PARA O WEB SERVICE

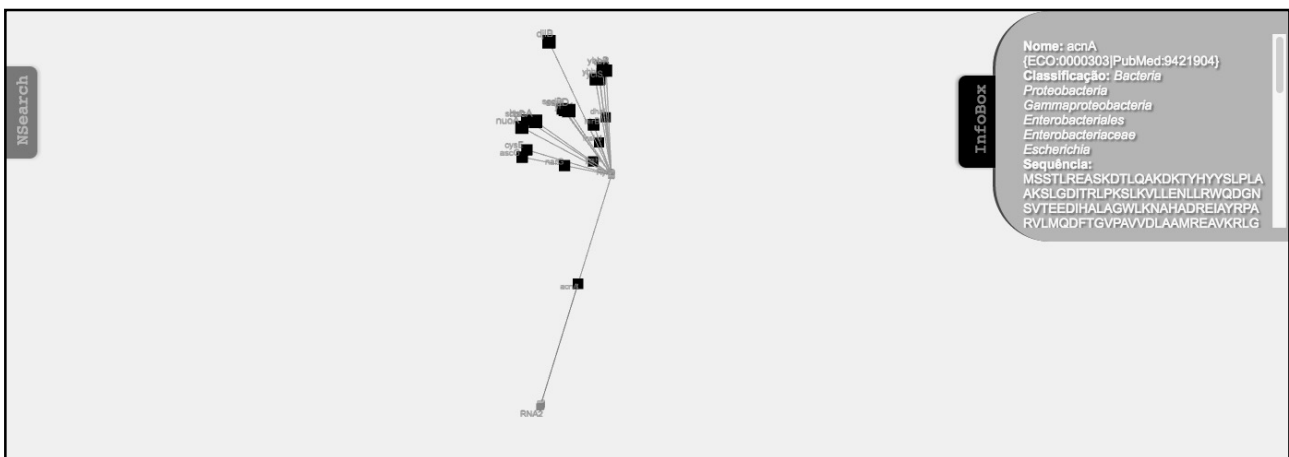


Fonte: O autor (2016)

Novamente o TogoWS retorna em JSON, porém é chamada uma função para atualizar as informações na aba, UpdateInfoBox(), que injeta no HTML os resultados do *parse*.

A RedIBio por fim apresenta os resultados contendo o nome, classificação e sua sequência.

FIGURA 35 - EXEMPLO DA ENTRADA DE DADOS DA REDIBIO



Captura de tela do console do navegador. Após o usuário escolher qual organismo deseja ter as informações, a RedIBio envia para o *web service* uma segunda solicitação com o ID selecionado (ACNA_ECOLI).

Fonte: O autor (2016)

4 RESULTADOS

A RedIBio teve seu desenvolvimento orientado através de técnicas de visualização da informação com o intuito de ser uma ferramenta Web 3D e interativa, que seja capaz de formar um grafo a partir de um arquivo, xls ou xlsx. A validação de seu uso se deu através da aplicação no estudo de caso que envolve RNAs curtos regulatórios e seus alvos, é possível observar a rede de interações biológicas construída pela RedIBio e as facilidades disponíveis para o usuário.

4.1 APLICAÇÃO NO ESTUDO DE CASO

Os preditores de redes de interação de RNA curtos/alvos retornam uma tabela simples (Tabela 2), contendo os alvos (Gene e Sinônimo) encontrados e ordenados por um valor (P-value), que determina a probabilidade de interação entre o alvo e o RNA curto ao acaso.

TABELA 2 - EXEMPLO DO RESULTADO DA PREDIÇÃO DE UM RNA CURTO

Hs sRNA Gs12			
Posição	Gene	Sinônimo	P-value
1	-	Hsero_1802	-15.83
2	-	Hsero_4198	-15.41
3	vsrA	Hsero_1201	-11.46
4	mltB	Hsero_0987	-11.15
5	-	Hsero_3754	-11.09
6	-	Hsero_0223	-10.38
7	-	Hsero_1759	-10.05
8	recX	Hsero_0656	-9.85
9	-	Hsero_1696	-9.6
10	-	Hsero_0972	-9.31
11	ftsY	Hsero_4146	-9.15
12	-	Hsero_4515	-8.89
13	-	Hsero_4451	-8.47
14	livG	Hsero_1787	-8.37
15	-	Hsero_2224	-8.3

Dados provenientes (*output*) da ferramenta de predição TargetRNA2 para o RNA curto Hs_sRNA_Gs12, salvos em uma tabela de *Excel*. Os alvos (genes) são ordenados pelo P-value, valor que determina a força da interação ao acaso, do mais provável (Hsero_1802) até o menos provável (Hsero_2224). Os alvos são apresentados de acordo com a anotação do genoma (Coluna Sinônimo) e alguns já tem identidade conhecida (coluna Gene).

Fonte: O autor (2016)

Ao utilizar esses resultados, normalmente em arquivos *Excel* (.xls), os montadores de redes de interação requerem algumas alterações na estrutura dos dados de entrada, como, por exemplo, o *script* de montagem de rede CLR do pacote MINET¹⁰, que somente aceita uma matriz quadrada, na qual o relacionamento deve ser descrito de todos contra todos. Os objetos da rede estariam dispostos na primeira linha e espelhados na primeira coluna. O Apêndice 1 demonstra esta configuração.

Porém com essa adaptação, que é recorrente para o uso de outras ferramentas de montagem de rede, surge o problema da redundância de informação que resulta em aumento do tamanho do arquivo, seja no armazenamento ou na visualização. Atrelada a isso, a manutenção se torna mais trabalhosa, e pode ser necessária se o computador do usuário tiver capacidade suficiente para abrir um arquivo com milhares de linhas, com o mesmo número de colunas.

Visando simplificar, tanto na montagem dos dados, quanto diminuir o processamento da RedIBio ao ler toda a matriz, foi adaptado o código para receber os RNAs curtos na primeira linha e os alvos relacionados na primeira coluna, a Tabela 3 representa este formato. Para o RNA curto Hs_sRNA_G12 estão preditos 14 alvos com p- Value variando de -15,83 a -8,30

TABELA 3 - EXEMPLO DA ENTRADA DE DADOS DA REDIBIO

	Hs_sRNA_Gs12
Hsero_1802	-15.83
Hsero_4198	-15.41
vsrA	-11.46
mltB	-11.15
Hsero_3754	-11.09
Hsero_0223	-10.38
Hsero_1759	-10.05
recX	-9.85
Hsero_1696	-9.6
Hsero_0972	-9.31
ftsY	-9.15
Hsero_4515	-8.89
Hsero_4451	-8.47
livG	-8.37
Hsero_2224	-8.3

As tabelas para entrada na RedIBio contem a configuração onde o RNA curto é disposto por coluna e seus alvos organizados por linhas.

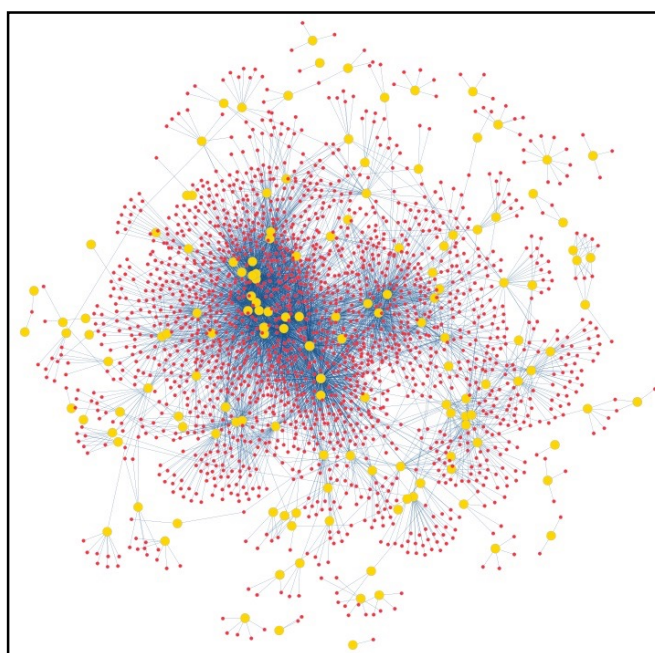
Fonte: O autor (2016)

¹⁰ <http://bioconductor.org/packages/release/bioc/html/minet.html>

Então se optou por um formato mais próximo com a formatação utilizada no processo de captação das informações e sem redundância.

Foi definido que o formato de entrada deveria ser uma tabela contendo os RNAs curtos do organismo *Herbaspirillum seropedicae* SmR1 e seus alvos. Essa tabela possui 165 RNAs curtos com 255 alvos e isso significa que em uma rede serão 420 objetos interligados. A Figura 36 contém a rede formada a partir destes dados utilizando o pacote Rede R¹¹ do Bioconductor¹².

FIGURA 36 - REDE EM 2D DA PREDIÇÃO DE ALVOS PARA 165 RNAS CURTOS



Rede construída através do Rede R, demonstrando o resultado de um arquivo contendo 165 RNAs curtos (amarelo) de *Herbaspirillum seropedicae* SmR1 e seus alvos preditos (vermelho). O ambiente criado para rede utiliza duas dimensões.

Fonte: Dobrzanski (2016).

Nessa rede há somente duas dimensões e como consequência todos os elementos do grafo ficam próximos devido a limitação de espaço. A identificação dos RNA's curtos e seus alvos é comprometida, assim como suas ligações.

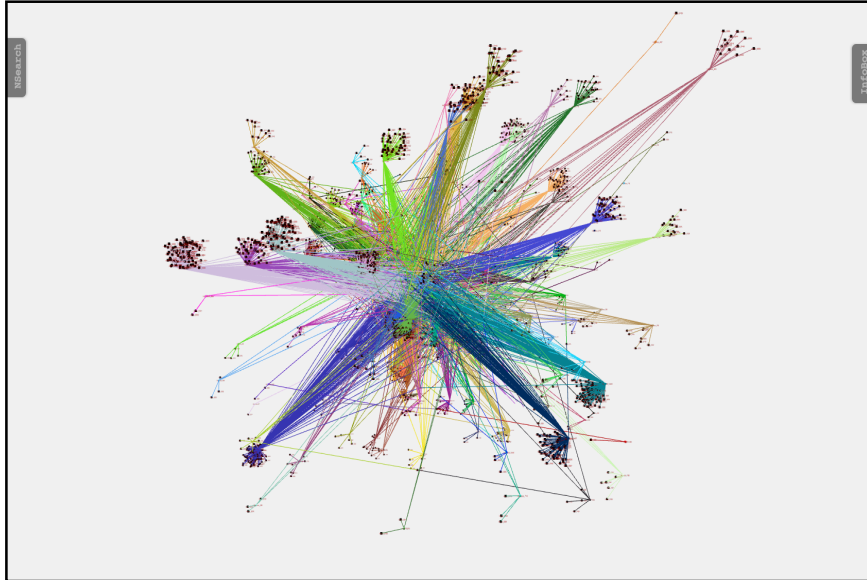
Gráficos em duas dimensões também limitam as ações tomadas pelo usuário para a navegação, basicamente tendo os movimentos de aproximar e afastar. Com o recurso gráfico em três dimensões, a navegação pela rede fica mais completa para explorar e

¹¹ <https://bioconductor.org/packages/release/bioc/html/RedeR.html>

¹² <https://bioconductor.org>

visualizar os itens que são disponibilizados. Aplicando os mesmos dados na RedIBio (Figuras 37 e 38) observa-se a vantagem da utilização da terceira dimensão.

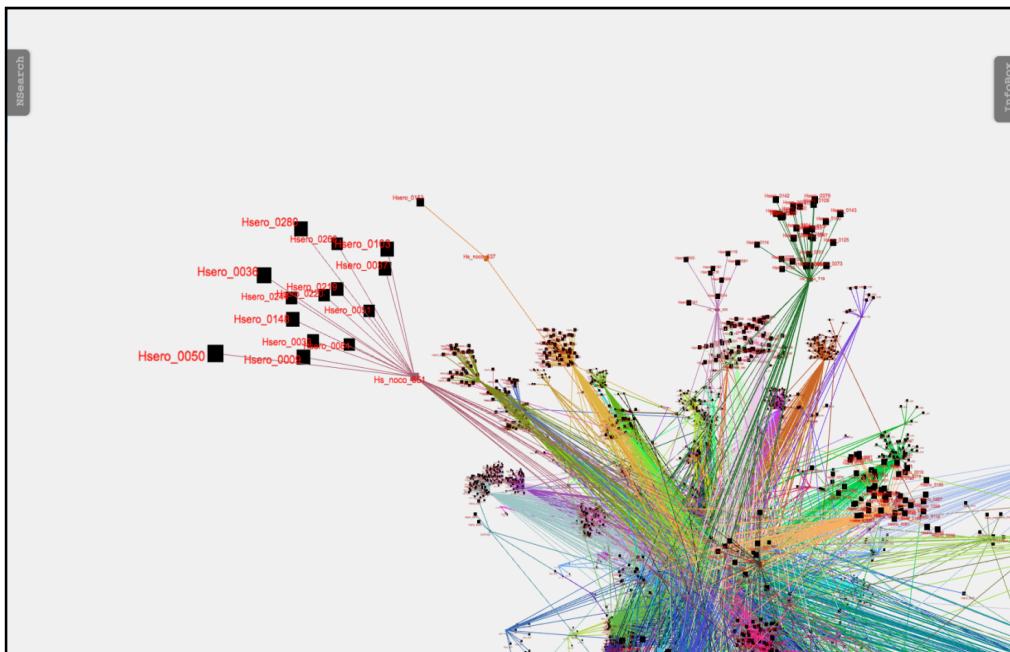
FIGURA 38 - CAPTURA DE TELA DE UMA REDE DE INTERAÇÕES GERADA PELA REDIBIO



Captura de tela contendo a rede de interações construída pela RedIBio dos 165 RNAs curtos da bactéria *Herbaspirillum Seropedicae* SMR1, onde cada RNA curto (nó) e suas linhas de interação possuem uma cor única.

Fonte: O autor (2016)

FIGURA 37 - CAPTURA DE TELA DE UMA REDE DA REDIBIO DEMONSTRANDO A NAVEGAÇÃO A PARTIR DE UM PONTO ESPECÍFICO



Captura de tela contendo a imagem aproximada de parte da rede de interação (Figura 35). O nó de nome HS_noco651 e seus alvos estão em evidência.

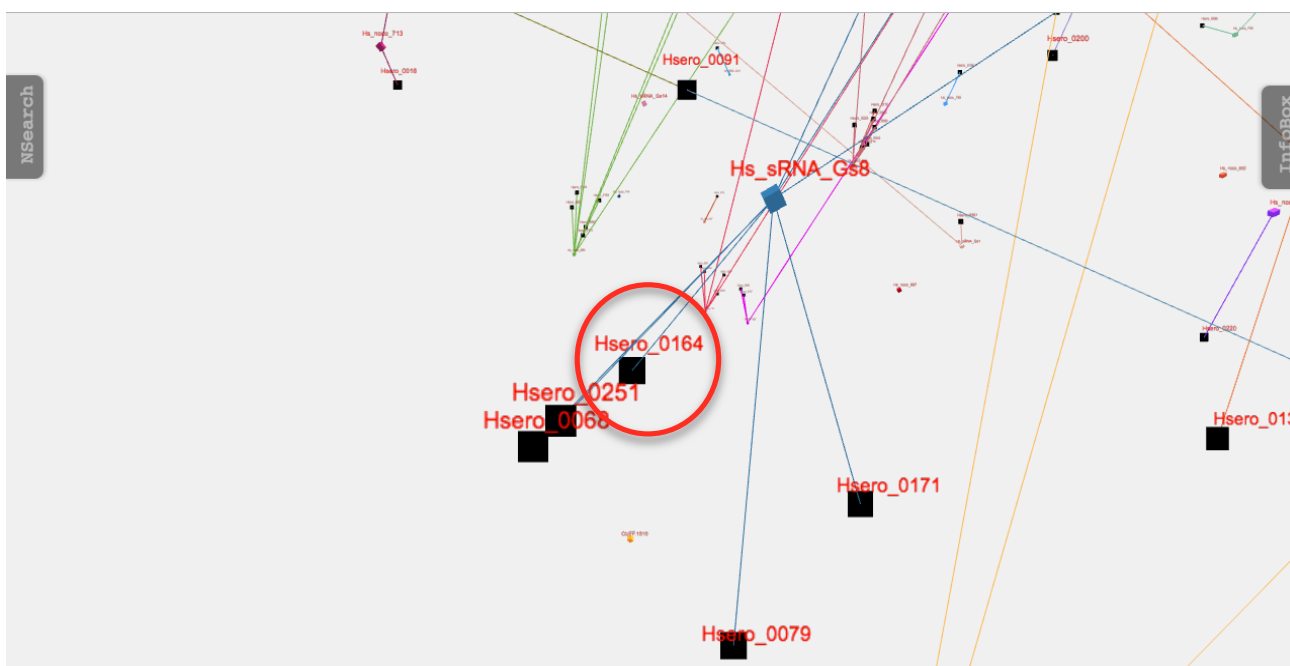
Fonte: O autor (2016)

Após se verificar que na extração de dados biológicos, obtém-se uma enorme quantidade informações e as representações gráficas são ineficientes em transmitir o conhecimento inerente, a estratégia de utilizar três dimensões para o gráfico foi considerada como solução para a visualização de grande quantidade de itens a serem visualizados na rede.

Além de disponibilizar ao usuário a possibilidade de navegação pela rede de interações gerada, a RedIBio identifica nominalmente os objetos (nós e alvos), além de contar com a integração com um banco de dados. Isto ocorre quando o usuário seleciona, com o *mouse*, um alvo (Figura 39).

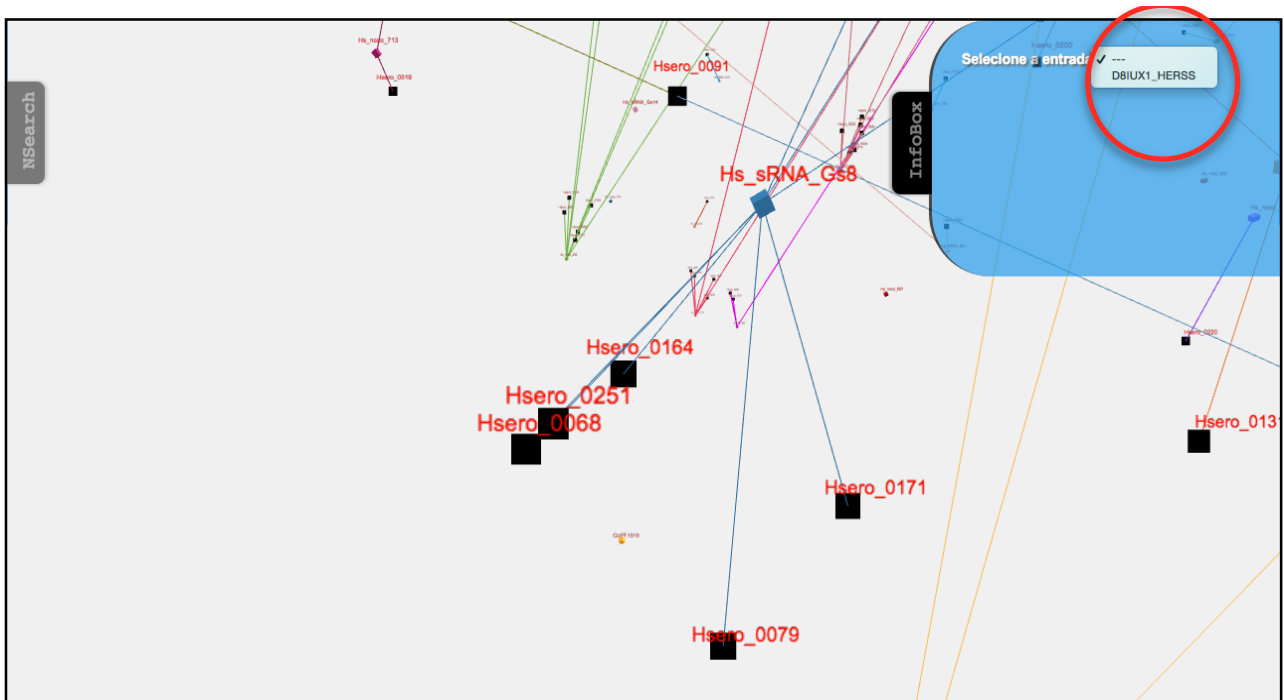
Por mais que as informações da ferramenta de predição retornem alvos sem nome ou proteínas hipotéticas, a conexão com o banco de dados feita pela RedIBio pode fornecer informações relevantes que indiquem a correta identidade do alvo (Figuras 39, 40 e 41). No exemplo, foi localizado um nó com alvos hipotéticos (Figura 39) e foi selecionado o alvo Hsero_0164 (Figura 40). Esse alvo foi processado no banco de dados e como resultado obtém-se o nome da proteína: YibP (Figura 41). Além disso, o usuário obtém a sequência primária da proteína e informações a respeito do organismo.

FIGURA 39 – LOCALIZAÇÃO DE UM NÓ COM ALVOS HIPOTÉTICOS



Captura de tela onde está destacando o nó Hs_sRNA_Gs8 que possui como alvos apenas genes hipotéticos o gene Hsero_0164(círculo vermelho).

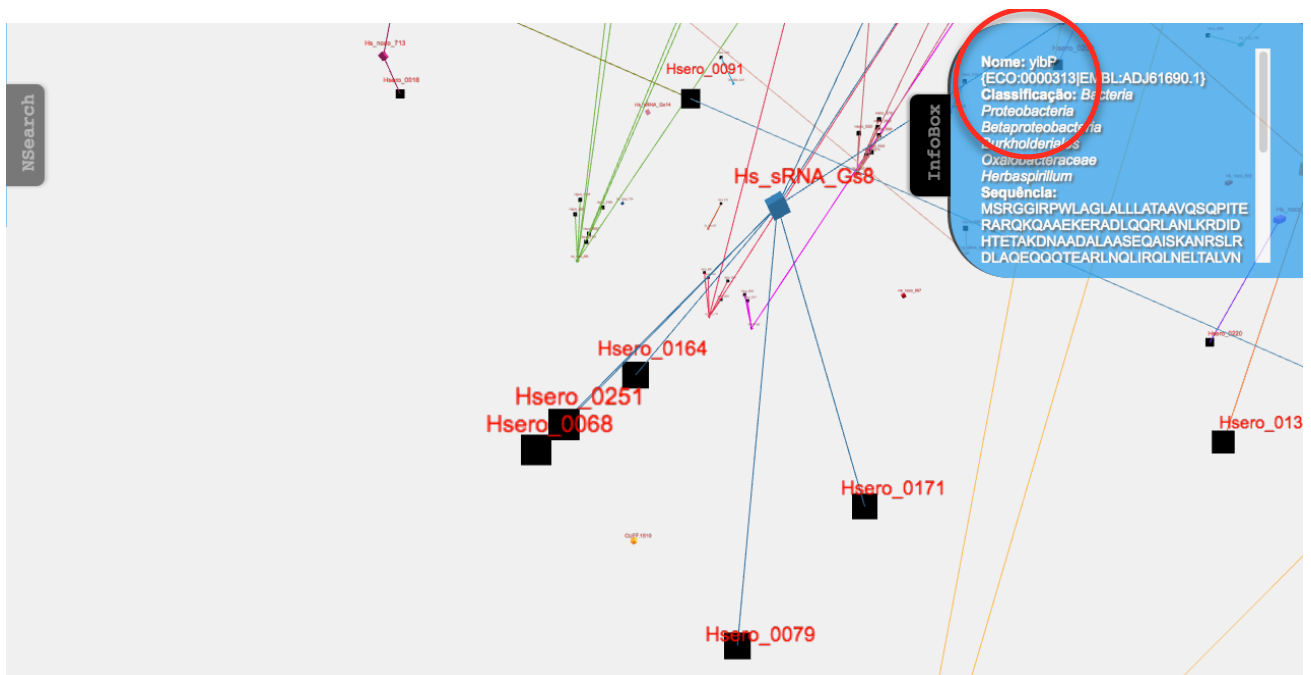
FIGURA 40 - SELEÇÃO DO HSERO_0164



Captura de tela onde está destacando o nó Hs_sRNA_Gs8 que possui como alvos apenas genes hipotéticos o gene Hsero_0164(círculo vermelho).

Fonte: O autor (2016)

FIGURA 41 - NOME REAL DO NÓ HS_SRNA_GS8 OBTIDO PELA CONSULTA AO BANCO DE DADOS BIOLÓGICO.



Captura de tela contendo a imagem aproximada evidenciando o nó Hs_sRNA_Gs8 identificado como YibP através de consulta no banco de dados biológico (círculo vermelho).

Fonte: O autor (2016)

4.2 COMPARATIVO

A RedIBio foi comparada com outras três ferramentas de visualização, que apresentam a possibilidade de formar uma rede em 3D, são eles o Arena3D, o BioLayout *Express3D* e o NAViGaTOR (Quadro 4).

QUADRO 4 - COMPARATIVO ENTRE FERRAMENTAS DE VISUALIZAÇÃO DE REDES EM 3D

	Sistemas	Download	Formato de entrada	Dependências	Integração (Banco de dados biológico)
Arena3D	Windows Linux Mac	Sim	txt	Sim	Não
BL Express3D	Windows Mac	Sim	layout/tgf/txt/sif/owl/expression/ matrix/graphml/mepn/xml	Não	Não
NAViGaTOR	Windows Linux Mac	Sim	xml	Não	Não
RedIBio	Windows Linux Mac	Não	xls/xlsx	Não	Sim

Fonte: O autor (2016)

Os softwares comparados com a RedIBio possuem um ponto em comum, a informação no grafo limitada ao arquivo de entrada, esta estratégia faz com que os arquivos fiquem complexos e ocupem mais espaço de armazenamento, como exemplo do arquivo em formato de texto adotado pelo Arena3D que necessita informar, a URL de cada nó, as nomenclaturas dos nomes de nós e as ligações entre os itens do grafo. Para contornar isso a RedIBio se integra a um banco de dados biológico, aumentando assim sua capacidade de informar o usuário.

A RedIBio também se diferencia na questão de não necessitar de *download* para funcionar, esse fator implica que não é necessário o desenvolvimento da ferramenta para cada sistema operacional, assim como suas manutenções, e o usuário não necessita buscar por novas versões.

Por ser uma ferramenta *Web*, seu desenvolvimento ocorreu através do uso do HTML 5, eliminando então qualquer necessidade do usuário adquirir *plugins* para o navegador executar gráficos 3D.

4.3 PERFORMANCE

Para demonstrar as capacidades que a RedIBio tem para a leitura e organização de dados de um arquivo, e a construção gráfica de uma rede de interações foi realizados testes de performance.

O objetivo destes testes é identificar possíveis pontos críticos na execução da ferramenta.

Os testes descritos a seguir foram realizados utilizando o seguinte computador:

Modelo: MacBook Pro;

Processador: Intel Core i7, 2,9GHz;

Memória: 4GB;

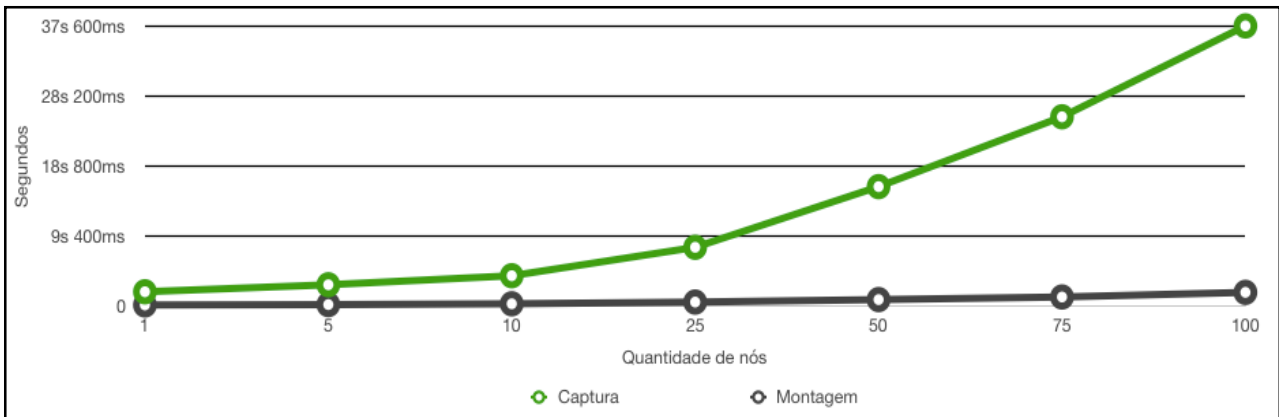
Gráficos: Intel HD Graphics 4000, 1 GB;

Sistema: OS X Yosemite;

4.3.1 Construção da Rede

A RedIBio foi testada de duas maneiras para observar seu processamento na construção da rede. O primeiro teste, apresentado no Figura 42, consta com uma entrada de 100 nós e 4500 interações, onde a cada repetição se diminuía a quantidade de nós e contabilizava-se o tempo de execução em dois momentos, na captação dos dados e montagem gráfica da rede.

FIGURA 42 - TEMPO DE EXECUÇÃO PELA VARIAÇÃO DOS NÓS



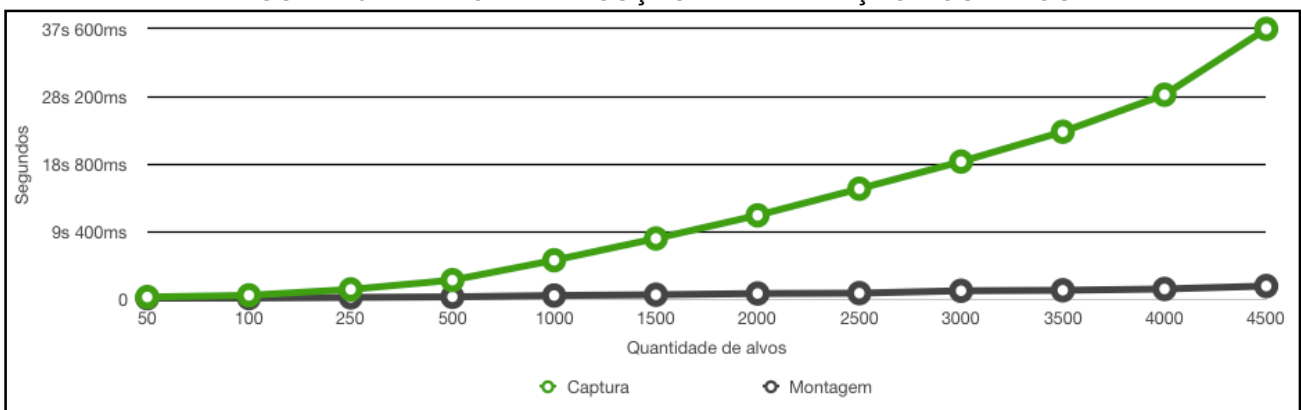
Tempo de execução, em segundos, para a criação da rede de interação (eixo y) *versus* o aumento da quantidade de nós (eixo x). Foram comparados dois momentos - captura dos dados do arquivo de entrada e sua ordenação (linha verde) e montagem da rede de interação (linha preta).

Fonte: O autor (2016)

O teste demonstra que entre 25 e 50 nós, a velocidade da captura das informações do arquivo de *upload* aumenta, em aproximadamente 10 segundos por cada 25 nós adicionados. Enquanto na montagem da rede a variação de tempo é imperceptível.

O segundo teste priorizou a variação de alvos, sem variar o número de nós da entrada. O resultado está apresentado na Figura 43.

FIGURA 43 - TEMPO DE EXECUÇÃO PELA VARIAÇÃO DOS ALVOS



Tempo de execução, em segundos, para a criação da rede de interação (eixo y) *versus* o aumento da quantidade de alvos. Foram comparados dois momentos - captura dos dados do arquivo de entrada e sua ordenação (linha verde) e montagem da rede de interação (linha preta).

Fonte: O autor (2016)

A captação de alvos, no arquivo de *upload*, ultrapassa 10 segundos entre 1500 e 2000 alvos. Comparando a Figura 42, percebe-se que os acréscimos de tempo na captura

têm o mesmo comportamento, porém para os nós a variação é em dezenas, e para os alvos ocorre nos milhares.

Os testes demonstram que a captura dos dados é o ponto crítico da ferramenta. Situados na parte do *Controller* e na linguagem em PHP, a leitura do arquivo e a organização das informações consomem quase todo o tempo de processamento, enquanto a montagem da rede se altera minimamente ao longo do grafo.

4.3.2 Renderização

Os seguintes testes determinam o comportamento do grafo após a construção, contando com quantos *frames* a ferramenta consegue renderizar por segundo (FPS), quantos milissegundos são necessários para renderizar um *frame* e quantos megabytes são alocados na memória pelo navegador. A Tabela 5 demonstra o comportamento da ferramenta através da variação de nós, iniciando com 1 até 100.

TABELA 4 - VARIAÇÕES DE FRAMES POR SEGUNDO, TEMPO PARA CARREGAR UM FRAME EM MILISSEGUNDOS E MEMÓRIA ALOCADA EM MEGABYTES

NÓ	FPS	MS	MB
1	60	16ms	36,00
5	60	16ms	36,00
10	60	16ms	36,00
25	60	16ms	36,00
50	43	38ms	36,00
75	25	50ms	36,00
100	14	75ms	36,00

Variações nos valores FPS (Frames Por Segundo), MS (MicroSegundos) e MB (MegaBytes) de acordo com o acréscimo de nós durante a visualização de uma rede de interações.

Fonte: O autor (2016)

A queda de renderização ocorre a partir de 25 nós, onde diminui 28,33% o FPS e ocorre o dobro de tempo para gerar um *frame* (MS). Ao chegar em 100 nós, a renderização perde desempenho, tendo queda 76,6% no FPS e o tempo para gerar o *frame* aumentou

em 368%. Já a alocação na memória do navegador permaneceu inalterado ao longo da variação do número de nós.

Este teste refere-se ao processamento gráfico da RedIBio de acordo com as configurações do computador utilizado e citadas anteriormente, porém o seu desempenho pode variar de acordo com as configurações da máquina em que a ferramenta está acessando.

5 CONCLUSÃO

Este projeto apresentou o desenvolvimento de uma ferramenta que, com a entrada de uma planilha Excel, gera um grafo 3D em forma de rede de interações e integrado a banco de dados biológicos. A RedIBio atua como uma ferramenta Web de interface responsiva utilizando técnicas de visualização da informação.

A aplicação da visualização de informação, neste trabalho, possibilita o usuário extrair conhecimentos da rede de maneira visual, rápida e simples, apenas selecionando o alvo sobre o qual deseja ter mais informações.

A RedIBio contribuiu como visualizador de um algoritmo de predição de redes de interações de RNAs curtos em procariotos, sendo uma melhor opção de criação e visualização de redes. Destaca-se como principal vantagem a integração com um banco de dados biológicos, não observada em trabalhos similares.

Outras vantagens oferecidas:

- Visualização através de um navegador: o usuário não precisa efetuar download de um programa e/ou uma extensão que possibilite gerar o grafo, ou mesmo preocupar-se com a compatibilidade do seu sistema operacional.
- Arquivo de entrada simplificado: Diferente dos softwares disponíveis, a RedIBio não necessita das informações dos componentes da rede no arquivo.
- Manutenção imperceptível ao usuário: Caso alguma correção tenha que ser feita, como melhoramento de desempenho ou que melhore a experiência do usuário seja feita, não haverá nenhum custo para usuário.
- Possibilidade de expansão: A possibilidade de continuação e adição de novas funcionalidades à RedIBio, que assim como a manutenção, não impactará o usuário em questões de atualização de versão em seu computador.

Durante os testes foi comprovado a existência de um ponto crítico nas ações de leitura e organização das informações, sendo assim como trabalhos futuros sugere-se uma otimização da captura de dados, ou a possibilidade de outros tipos de arquivos para o upload.

Para trabalhos futuros, também se recomenda a possibilidade de escolha de outros bancos de dados para a integração, criar outras formas de visualizar (*View*) a rede. Assim

como a manipulação dos objetos da rede, deixando livre para o usuário realocar os objetos pelo espaço 3D.

REFERÊNCIAS

AHUJA, Ravindra K; MAGNANTI, Thomas .; ORLIN, James B. **Network Flows: Theory, Algorithms, and Applications**. New Jersey: Prentice Hall, 1993. 846 p.

ARENA3D. **ARENA3D**. Disponível em: <<http://arena3d.org>>. Acesso em: 20 set. 2016.

ALTAY, G.; EMMERT-STREIB, F. Revealing differences in gene network interference algorithms on the network level by ensemble methods. **Bioinformatics**, v 26, n 14, p 1738-1744, 2010.

ALTOVA WhitePaper Web services: Benefits, challenges, and a unique, visual development solution, Altova GmbH, White Paper, 2006.

ALTSCHUL, S. F. et al. Basic local alignment search tool. **J Mol Biol**, v. 215, n. 3, p. 403-10, 1990.

ALTUVIA, S. Identification of bacterial small non-coding RNAs: experimental approaches. **Curr Opin Microbiol**, v. 10, n. 3, p. 257-261, Jun 2007.

BABITZKE, P.; ROMEO, T. CsrB sRNA family: sequestration of RNA-binding regulatory proteins. **Curr Opin Microbiol**, v. 10, n. 2, p. 156-63, Apr 2007.

BACKOFEN, R; HESS, W.R. Computational prediction of sRNAs and their targets in bacteria. **RNA Biology**, v. 7, p. 33-42, 2010.

BYTE, Bark Of The. **Three.js projecting mouse clicks to a 3D scene - how to do it and how it works**. Disponível em: <<http://barkofthebyte.azurewebsites.net/post/2014/05/05/three-js-projecting-mouse-clicks-to-a-3d-scene-how-to-do-it-and-how-it-works>>. Acesso em: 28 set. 2016.

DIAS, M. P. A contribuição da Visualização da Informação para a Ciência da Informação. 2007 Dissertação (Mestrado)-Pontifícia Universidade Católica de Campinas,Centro de Ciências Sociais Aplicadas, Pós-Graduação em Ciência da Informação, Campinas, 2007. Disponível em:< http://www.bibliotecadigital.puc-campinas.edu.br/tde_arquivos/2/TDE-2007-08-02T080844Z-1353/Publico/Mateus%20Pereira%20Dias.pdf> Acesso em: 12 set. 2016.

DOBRZANSKI, Tatiane. **Rede**. [mensagem pessoal] Mensagem recebida por: <rodrigo.l@outlook.com>. em: 16 out. 2016.

CodeIgniter. **CodeIgniter**. Disponível em: <<http://www.codeigniter.com>>. Acesso em: 15 set. 2016.

CROCKFORD, Douglas. **The application/json Media Type for JavaScript Object Notation (JSON)**. 2006. Disponível em: <<http://www.ietf.org/rfc/rfc4627.txt?number=4627>>. Acesso em: 13 set. 2016.

EXPRESS3D, Biolayout. **BioLayout Express3D**. Disponível em: <<http://www.biolayout.org>>. Acesso em: 22 set. 2016.

FIELDING, Roy Thomas. **Architectural Styles and the Design of Network-based Software Architectures**. 2000. 162 f. Tese (Doutorado) - Curso de Information And Computer Science, University Of California, Irvine, 2000.

FREITAS, C. M. D. S.; CHUBACHI, O. M.; LUZZARDI, P. R. G.; CAVA R. A. Introdução à Visualização de Informações. RITA – Revista de Informática Teórica e Aplicada, Instituto de Informática, v. 8, p. 143-158, 2001.

FREITAS, Carla Maria dal Sasso; WAGNER, Flavio Rech. A Methodology for Selecting Visual Representation in Scientific and Simulation Applications. In: SIBGRAPI, 6., 1993, Porto Alegre. **Anais...** . Porto Alegre: Sibgrapi, 1993. p. 89 - 97. Disponível em: <<http://sibgrapi.sid.inpe.br/rep/sid.inpe.br/sibgrapi/2012/12.11.15.18>>. Acesso em: 5 ago. 2016.

FRIENDLY, Michael. **Milestones in the history of thematic cartography, statistical graphics, and data visualization**. 2008. Disponível em: <http://www.math.usu.edu/~symanzik/teaching/2009_stat6560/Downloads/Friendly_milestones.pdf>. Acesso em: 10 ago. 2016.

GERSHON, Nahum; EICK, Stephen G.; CARD, Stuart. Design: Information Visualization. **Interactions**, New York, v. 2, n. 5, p.9-15, abr. 1998.

GOH, K.-i. et al. The human disease network. **Proceedings Of The National Academy Of Sciences**, [s.l.], v. 104, n. 21, p.8685-8690, 14 maio 2007. Proceedings of the National Academy of Sciences. <http://dx.doi.org/10.1073/pnas.0701361104>.

GOOGLE. **Zeitgeist 2012**. Disponível em: <<https://www.google.com/zeitgeist/2012/#the-world>>. Acesso em: 21 set. 2016.

HUSTINAWATI; HIMAWAN, Albert Kurnia; LATIFAH. Performance Analysis Framework Codeigniter and CakePHP in Website Creation. **International Journal Of Computer Applications**. New York, p. 6-11. maio 2014.

JAVASCRIPT. **JavaScript**. Disponível em: <<http://www.javascript.com>>. Acesso em: 13 set. 2016.

JQUERY. **JQuery**. Disponível em: <<http://www.jquery.com>>. Acesso em: 13 set. 2016.

JSON. **JSON**. Disponível em: <<http://www.json.org>>. Acesso em: 13 set. 2016.

KATAYAMA, T.; NAKAO, M.; TAKAGI, T.. TogoWS: integrated SOAP and REST APIs for interoperable bioinformatics Web services. **Nucleic Acids Research**, [s.l.], v. 38, n. , p.706-711, 14 maio 2010. Oxford University Press (OUP). <http://dx.doi.org/10.1093/nar/gkq386>.

KREGGER, Heather. Web Services Conceptual Architecture 1.0. Somers: Ibm Software Group, 2001. 40 p.

LALAOUNA, D; SIMONEAU-ROY, M.; LAFONTAINE, D.; MASSÉ, E. Regulatory RNAs and target mRNA decay in prokaryotes. **Elsevier**, 2013.

LEFF, A.; RAYFIELD, J.t.. Web-application development using the Model/View/Controller design pattern. **Proceedings Fifth IEEE International Enterprise Distributed Object**

Computing Conference, [s.l.], p.118-127, jan. 2001. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/edoc.2001.950428>.

LIU, J. M.; CAMILLI, A. A broadening world of bacterial small RNAs. *Curr Opin Microbiol*, v. 13, n. 1, p. 18-23, Feb 2010.

MASSÉ, E.; MAJDALANI, N.; GOTTESMAN, S. Regulatory roles for small RNAs in bacteria. *Curr Opin Microbiol*, 6:120-124, 2003.

MA'AYAN, A.. Introduction to Network Analysis in Systems Biology. **Science Signaling**, [s.l.], v. 4, n. 190, p.1-11, 6 set. 2011. American Association for the Advancement of Science (AAAS). <http://dx.doi.org/10.1126/scisignal.2001965>.

METEOROLOGIA, Instituto Nacional de. **Gráficos**. Disponível em: <<http://www.inmet.gov.br/portal/index.php?r=tempo/graficos>>. Acesso em: 11 jun. 2016.

MODI, S. R. et al. Functional characterization of bacterial sRNAs using a network biology approach. **Proceedings Of The National Academy Of Sciences**, [s.l.], v. 108, n. 37, p.15522-15527, 29 ago. 2011. Proceedings of the National Academy of Sciences. <http://dx.doi.org/10.1073/pnas.1104318108>.

NAVIGATOR. **NAVIGATOR**. Disponível em: <<http://ophid.utoronto.ca/navigator/index.html>>. Acesso em: 23 set. 2016.

NELSON, David L; COX, Michael M. Princípios de Bioquímica de Lehninger. 6. ed.: Artmed, 2014.

NEWMAN, M. E. J.; **Networks: an introduction**. 2010.

PENADÉS, Soledad. **Object picking**. Disponível em: <<https://soledadpenades.com/articles/three-js-tutorials/object-picking/>>. Acesso em: 29 set. 2016.

PHP. **PHP**. Disponível em: < <https://secure.php.net/> >. Acesso em: 02 set. 2016.

PICHON, C.; FELDEN, B. Small RNA gene identification and mRNA target predictions in bacteria. *Bioinformatics*, v. 24, n. 24, p. 2807-13, Dec 15 2008.

RODRIGUEZ, Alex. **RESTful Web services: The basics**. 2008. Disponível em: <<https://www.ibm.com/developerworks/webservices/library/ws-restful>>. Acesso em: 09 set. 2016.

SCHROEDER, William J.; MARTIN, Kenneth M.; LORENSEN, William E.. **The Visualization Toolkit: An Object Oriented Approach to 3D Graphics**. 3. ed. New York: Kitware, Inc., 2003.

TAO, Ying et al. Information visualization techniques in bioinformatics during the postgenomic era. **Drug Discovery Today: BIOSILICO**, [s.l.], v. 2, n. 6, p.237-245, nov. 2004. Elsevier BV. [http://dx.doi.org/10.1016/s1741-8364\(04\)02423-0](http://dx.doi.org/10.1016/s1741-8364(04)02423-0).

THREE JS. **Three JS**. Disponível em: < <https://threejs.org/> >. Acesso em: 14 set. 2016.

TIDWELL, Doug. **Web Services: The Web's next Revolution**. 2000. Disponível em: <<http://www-106.ibm.com/developerworks/webservices/>>. Acesso em: 20 ago. 2016.

TJADEN, B. TargetRNA: a tool for predicting targets of small RNA action in bacteria. **Nucleic Acids Res**, v. 36, n. Web Server issue, p. W109-13, Jul 1 2008.

TUFTE, Edward Rolf. **The Visual Display of Quantitative Information**. 2. ed. Connecticut: Graphics Press LLC, 2013.

VASCONCELOS, Laércio. **Como fazer diagramas de redes**. 2015. Disponível em: <<http://www.laercio.com.br/2015-como-fazer-diagramas-de-redes/>>. Acesso em: 27 out. 2016.

VOGEL, Friedrich; MOTULSKY, Arno G.. **Vogel and Motulsky's Human Genetics: Problems and Approaches**. 3. ed. Heidelberg: Pro Edit, 1997.

VOGEL, Jörg; WAGNER, e Gerhart H. Target identification of small noncoding RNAs in bacteria. **Current Opinion In Microbiology**, [s.l.], v. 10, n. 3, p.262-270, jun. 2007. Elsevier BV. <http://dx.doi.org/10.1016/j.mib.2007.06.001>.

ZAHA, Arnaldo; FERREIRA, Henrique Bunselmeyer; PASSAGLIA, Luciane M. P.. **Biologia Molecular Básica**. 5. ed. Porto Alegre: Artmed, 2014.

W3C. **W3C**. Disponível em: <<https://www.w3.org>>. Acesso em: 14 set. 2016.

W3SCHOOLS. **W3schools**. Disponível em: <<http://www.w3schools.com>>. Acesso em: 02 jul. 2016.

WASSARMAN, K. M. et al. Identification of novel small RNAs using comparative genomics and microarrays. **Genes & Development**, v. 15, n. 13, p. 1637-1651, Jul 1 2001.

WANG, Jiang et al. SRNATarBase 3.0: an updated database for sRNA-target interactions in bacteria. **Nucleic Acids Res**, [s.l.], v. 44, n. 1, p.248-253, 25 out. 2015. Oxford University Press (OUP). <http://dx.doi.org/10.1093/nar/gkv1127>.

WARE, Colin. **Information Visualization: Perception for Design**. 3. ed. Waltham: Elsevier, 2013.

WU, T. H.; CHANG, I. Y.; CHU, L. J.; HUANG, H.; NG, W. V. Modularity of *Escherichia coli* sRNA regulation revealed by sRNA-target and protein network analysis. **BMC Bioinfo**, 11, 1-7, 2011.

APÊNDICE 1

EXEMPLO DE UMA MATRIZ PARA A CRIAÇÃO DE REDE

	Hs_sRNA_Gs12	Hs_sRNA_1802	Hs_sRNA_4198	vsrA	mltB	Hsero_3754	Hsero_0223	Hsero_1759	recX	Hsero_1696	Hsero_0972	ftsY	Hsero_4515	Hsero_4451	livG	Hsero_2224
Hs_sRNA_Gs12	0	-15.83	-15.41	-11.46	-11.15	-11.09	-10.38	-10.05	-9.85	-9.6	-9.31	-9.15	-8.89	-8.47	-8.37	-8.3
Hs_sRNA_1802	-15.83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hs_sRNA_4198	-15.41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
vsrA	-11.46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mltB	-11.15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hsero_3754	-11.09	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hsero_0223	-10.38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hsero_1759	-10.05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
recX	-9.85	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hsero_1696	-9.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hsero_0972	-9.31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ftsY	-9.15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hsero_4515	-8.89	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hsero_4451	-8.47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
livG	-8.37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hsero_2224	-8.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabela derivada das informações contidas na Tabela 2, construída no formato de matriz quadrada. As primeiras linha e coluna contêm as mesmas informações, espelhadas, e nas interseções (células) estão os valores dos relacionamentos. O número zero indica a falta de interação entre o RNA curto/alvos.

Fonte: O autor (2016).