

UNIVERSIDADE DE CAXIAS DO SUL
CENTRO DE COMPUTAÇÃO E TECNOLOGIA DA INFORMAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ANDRE PRETTO DE PARIS

TRANSFERÊNCIA DA APLICAÇÃO DIS2PPI
PARA A PLATAFORMA WEB

CAXIAS DO SUL, JUNHO DE 2014.

ANDRE PRETTO DE PARIS

**TRANSFERÊNCIA DA APLICAÇÃO DIS2PPI
PARA A PLATAFORMA WEB**

Trabalho de Conclusão do Título de Bacharel em Sistemas de Informação pela Universidade de Caxias do Sul, como requisito parcial para obtenção do título de Bacharel, orientado pelo Prof. Dr. Daniel Luis Notari.

CAXIAS DO SUL, JUNHO DE 2014.

Dedico este trabalho a todos aqueles que contribuíram de forma direta ou indireta para conclusão do mesmo.

RESUMO

O estudo do DNA humano se tornou cada vez mais complexo ao longo dos anos, e a tecnologia ajuda a compreender e a realizar as pesquisas dos profissionais da área de biologia. A bioinformática, uma união entre a biologia e a ciência da computação, utiliza-se de programas de computador para fazer inferências a partir de dados obtidos da biologia molecular, fazendo conexões entre eles e derivando previsões importantes e relevantes. A biologia de sistemas, uma subárea da bioinformática, auxilia no entendimento das interações entre as populações de moléculas, células e outros organismos. Para facilitar as análises realizadas pelos biólogos, os experimentos podem ser organizados usando-se *workflows* ou *pipelines*. Um *workflow* científico é um experimento realizado através de uma sequência determinada de etapas. As etapas caracterizam um fluxo de execução com o propósito de integrar e processar dados para um fim específico. Para tal, foi criada a ferramenta Dis2PPI, um *workflow* científico que possibilitou integrar o banco de dados de doenças monogênicas OMIM (*Online Mendelian Inheritance in Man*) com o programa de metabuscas STRING (*Search Tool for the Retrieval of Interacting Genes/Proteins*) com o objetivo de criar uma rede de interação proteína-proteína associada com doenças de natureza monogênica. A ferramenta foi desenvolvida na plataforma Java como uma aplicação *desktop*, onde os usuários necessitam realizar o *download* da mesma para poder utilizá-la. Seguindo a metodologia de evolução de software de Sommerville e utilizando as tecnologias *Java Server Faces*, *Spring*, *PrimeFaces* e o servidor de aplicação *Glassfish*, a aplicação foi transferida para a plataforma *web* seguindo os padrões de arquitetura *Model-View-Controller* com o objetivo de gerenciar melhor o acesso à ferramenta e facilitar a utilização da mesma por seus usuários, além de melhorar os recursos oferecidos pela ferramenta.

Palavras-chave: Migração de interface gráfica para *web*, Integração de banco de dados, *workflow* científico.

LISTA DE ILUSTRAÇÕES

FIGURA 1. FLUXO DO <i>WORKFLOW</i> CIENTÍFICO DO DIS2PPI.....	15
FIGURA 2. TELA DE PESQUISA DE DOENÇAS NA BASE DE DADOS OMIM..	15
FIGURA 3. TELA DE SELEÇÃO DO RELATÓRIO DE DOENÇA	16
FIGURA 4. TELA DE SELEÇÃO DE PROTEÍNAS, ONDE É POSSÍVEL A SELEÇÃO DAS PROTEÍNAS DO RELATÓRIO DE DOENÇA (C), OU A POSSIBILIDADE DE INCLUSÃO DE NOVAS PROTEÍNAS (D) ALÉM DA ESCOLHA DO ORGANISMO (F) PARA VALIDAÇÃO NA BASE STRING (E).	16
FIGURA 5. TELA DE OCORRÊNCIAS DE PROTEÍNAS VALIDADAS NA BASE STRING (G) E OBTENÇÃO DA REDE DE INTERAÇÃO (H).	17
FIGURA 6. TELA DE VISUALIZAÇÃO DAS REDES DE INTERAÇÃO COM AS OPÇÕES DE XML, IMAGEM, TEXTO (I) OU PARA IMPORTAÇÃO NO CYTOSCAPE (J)	17
FIGURA 7. REDE DE INTERAÇÃO DA PROTEÍNA PUF60, UTILIZADA COMO EXEMPLO DO FUNCIONAMENTO DA FERRAMENTA DIS2PPI.	18
FIGURA 8. TELA DE PROTEÍNAS A DESCONSIDERAR NA EXTRAÇÃO DOS DADOS DA BASE OMIM.....	18
FIGURA 9. TELA DE CONFIGURAÇÃO DA FERRAMENTA DIS2PPI.....	19
FIGURA 10. TELA INICIAL DO DASMIWEB.....	20
FIGURA 11. RESULTADO DA CONSULTA NO DASMIWEB – INFORMAÇÕES DO IDENTIFICADOR	20
FIGURA 12. RESULTADO DA CONSULTA NO DASMIWEB – INTERAÇÕES...	21
FIGURA 13. INTERFACE <i>DESKTOP</i> DA FERRAMENTA JBIOWH COM AS BASES DE DADOS A SEREM CONSULTADAS.....	22
FIGURA 14. RESULTADOS DA CONSULTA REALIZADA NO SNOW – PARTE 1	23
FIGURA 15. RESULTADOS DA CONSULTA REALIZADA NO SNOW – PARTE 2	24
FIGURA 16. PROCESSO DE EVOLUÇÃO DE <i>SOFTWARE</i>	27
FIGURA 17. PROCESSO DE REENGENHARIA DE <i>SOFTWARE</i>	28
FIGURA 18. O PADRÃO MVC.....	29
FIGURA 19. ARQUITETURA DE <i>SOFTWARE</i> DA FERRAMENTA DIS2PPI.....	33
FIGURA 20. ARQUITETURA DA FERRAMENTA DIS2PPI ENTRE <i>SOFTWARES</i> 34	
FIGURA 21. TELA INICIAL DO DIS2PPI	35
FIGURA 22. TELA DE SELEÇÃO DO RELATÓRIO DE DOENÇAS	35
FIGURA 23. TELA DE SELEÇÃO DE PROTEÍNAS	36
FIGURA 24. TELA DE PROTEÍNAS INVÁLIDAS	36
FIGURA 25. TELA DE SELEÇÃO DAS PROTEÍNAS VÁLIDAS PARA GERAÇÃO DA REDE DE INTERAÇÃO	37
FIGURA 26. TELA DE DOWNLOAD DA REDE DE INTERAÇÃO.....	37

FIGURA 27. TELA PARA VISUALIZAÇÃO DA REDE DE INTERAÇÃO UTILIZANDO OS <i>PLUGINS</i> DO CYTOSCAPE WEB.....	38
FIGURA 28. TELA DE <i>LOG</i> DA APLICAÇÃO.....	38
FIGURA 29. TELA DE PROTEÍNAS A DESCONSIDERAR.....	39
FIGURA 30. ESTRUTURA DO PROJETO DIS2PPIWEB.....	43
FIGURA 31. EXEMPLO DE INTEGRAÇÃO ENTRE <i>HTML</i> E CLASSE <i>JAVA</i>	44
FIGURA 32. CLASSE <i>JAVA</i> COM MAPEAMENTO DO CAMPO LISTADO NO <i>HTML</i>	44
FIGURA 33. <i>JSON</i> COM ESTRUTURA UTILIZADA PARA VISUALIZAÇÃO DA REDE DE INTERAÇÃO.....	46
FIGURA 34. NOVA TELA INICIAL DO DIS2PPIWEB PARA BUSCA DE DOENÇAS NA BASE DE DADOS OMIM	48
FIGURA 35. NOVA TELA DE SELEÇÃO DO RELATÓRIO DE DOENÇA.....	49
FIGURA 36. NOVA TELA DE SELEÇÃO DE PROTEÍNAS, COM POSSIBILIDADE DE INCLUSÃO DE NOVAS PROTEINAS, ALÉM DA SELEÇÃO DO ORGANISMO PARA VALIDAÇÃO NA BASE STRING.....	49
FIGURA 37. NOVA TELA DE OCORRÊNCIAS DE PROTEÍNAS VALIDADAS NA BASE STRING PARA OBTENÇÃO DA REDE DE INTERAÇÃO.....	50
FIGURA 38. NOVA TELA DE PROTEÍNAS INVÁLIDAS NA BASE STRING.....	50
FIGURA 39. NOVA TELA DE VISUALIZAÇÃO DE REDES DE INTERAÇÃO COM AS OPÇÕES <i>XML</i> , <i>IMAGEM</i> , <i>TEXTO</i> OU PARA ABRIR O CYTOSCAPE WEB.....	51
FIGURA 40. NOVA TELA DO CYTOSCAPE WEB INTEGRADA AO DIS2PPIWEB.....	52
FIGURA 41. COMPARAÇÃO DA REDE DE INTERAÇÃO RETORNADA PELO DIS2PPI E DIS2PPIWEB	52
FIGURA 42. TELA DE LISTA DE PROTEÍNAS A DESCONSIDERAR.....	53
FIGURA 43. NOVA TELA DE <i>LOG</i> DA APLICAÇÃO.....	53

LISTA DE TABELAS

TABELA 1. TABELA DE COMPARAÇÃO ENTRE AS FERRAMENTAS.....	25
TABELA 2. TABELA DE CATEGORIAS DE IMPACTO.....	40
TABELA 3. RESUMO DA ANÁLISE DE IMPACTO.....	41
TABELA 4. CASO DE TESTE 1 - VISUALIZAR ARQUIVOS TXT, IMAGEM, XML	54
TABELA 5. CASO DE TESTE 2 - VISUALIZAR REDE DE INTERAÇÃO NO CYTOSCAPE WEB.....	54
TABELA 6. CASO DE TESTE 3 - INCLUIR PROTEÍNAS VÁLIDAS MANUALMENTE.....	55
TABELA 7. CASO DE TESTE 4 - INCLUIR PROTEÍNAS INVÁLIDAS MANUALMENTE.....	56
TABELA 8- CASO DE TESTE 5 - VALIDAR TELA DE PROTEÍNAS INVÁLIDAS	57
TABELA 9. CASO DE TESTE 6 - VALIDAR INCLUSÃO MANUAL DE PROTEÍNAS A DESCONSIDERAR.....	58
TABELA 10. CASO DE TESTE 7 - VALIDAR <i>DOWNLOAD</i> DE PROTEÍNAS A DESCONSIDARAR.....	60
TABELA 11. CASO DE TESTE 8 - VALIDAR <i>UPLOAD</i> DE PROTEÍNAS A DESCONSIDARAR.....	61
TABELA 12. CASO DE TESTE 9 - VALIDAR TELA DE <i>LOG</i> DA APLICAÇÃO..	62

LISTA DE SIGLAS

DIS2PPI	<i>Disease To Protein-Protein Interaction</i>
DNA	<i>Deoxyribonucleic acid</i>
GPL	<i>General Public License</i>
HTML	<i>HiperText Markup Language</i>
IDE	<i>Integrated Development Environment</i>
JBioWH	<i>Java BioWareHouse</i>
JDK	<i>Java Development Kit</i>
JSF	<i>Java Server Faces</i>
JSON	<i>JavaScript Object Notation</i>
JSP	<i>Java Server Pages</i>
MVC	<i>Model-View-Controller</i>
NCBI	<i>National Center for Biotechnology Information</i>
OMIM	<i>Online Mendelian Inheritance in Man</i>
SIF	<i>Simple Interaction Format</i>
SNOW	<i>Studying Networks in the Omics World</i>
SQL	<i>Structured Query Language</i>
STRING	<i>Search Tool for the Retrieval of Interacting Genes/Proteins</i>
SSH	<i>Secure Shell</i>
URL	<i>Uniform Resource Locator</i>
XML	<i>eXtensible Markup Language</i>

SUMÁRIO

1 INTRODUÇÃO.....	9
1.1 CONTEXTUALIZAÇÃO	9
1.2 PROBLEMA E QUESTÃO DE PESQUISA.....	10
1.3 OBJETIVOS.....	10
1.4 ESTRUTURA DO TEXTO.....	11
2 REVISÃO BIBLIOGRÁFICA GERAL	12
2.1 CONCEITUAÇÃO.....	12
2.1.1 Proteínas, Proteomas e Interatomos.....	12
2.1.2 Workflow Científico.....	13
2.1.3 OMIM	13
2.1.4 STRING	14
2.1.5 Cytoscape.....	14
2.2 A FERRAMENTA DIS2PPI.....	14
2.3 FERRAMENTAS DE WORKFLOWS CIENTÍFICOS	19
2.3.1 Ferramenta DASMIweb.....	20
2.3.2 Ferramenta JBioWH	21
2.3.3 Ferramenta SNOW.....	23
2.3.4 Comparação das ferramentas.....	25
2.4 EVOLUÇÃO DE <i>SOFTWARE</i>	26
2.5 ARQUITETURA DE <i>SOFTWARE</i>	29
2.6 CONSIDERAÇÕES FINAIS	30
3 PROPOSTA DE TRANSFORMAÇÃO DA FERRAMENTA DIS2PPI PARA WEB ..	32
3.1 SOLICITAÇÕES DE MUDANÇAS	32
3.2 ANÁLISE DE IMPACTO.....	40
3.3 PLANEJAMENTO DA <i>RELEASE</i>	42
4 IMPLEMENTAÇÃO.....	43
4.1 IMPLEMENTAÇÃO DE MUDANÇA	43
4.2 LIBERAÇÃO DO SISTEMA	47
5 ESTUDO DE CASO.....	48
6 CASOS DE TESTE.....	54
7 CONSIDERAÇÕES FINAIS	63
8 REFERÊNCIAS.....	65

1 INTRODUÇÃO

Neste capítulo será realizada uma contextualização do trabalho em questão, a questão e o problema de pesquisa identificado. Além disso, serão destacados os objetivos propostos para a execução deste trabalho e uma amostragem da estrutura do trabalho durante os próximos capítulos.

1.1 CONTEXTUALIZAÇÃO

Segundo Prosdocimi et. al. (2002), devido ao surgimento dos sequenciadores automáticos de DNA, na metade da década de 90, houve um aumento na utilização de recursos computacionais para armazenamento e análise dos dados genéticos, facilitando a interpretação dos mesmos pelos profissionais da biologia. Com isso foi criado um novo campo chamado bioinformática (LESK, 2008), uma união entre a biologia e a ciência da computação, na qual são utilizados programas de computador para fazer inferências a partir de dados obtidos da biologia molecular, fazendo conexões entre eles e derivando predições importantes e relevantes. A biologia de sistemas é uma subárea da bioinformática que ajuda a compreender as interações observadas entre populações de moléculas, células e organismos resultantes do aumento da complexidade biológica (KARSENTI, 2012).

A realização dos experimentos da biologia pode ser realizada através de *workflows* científicos, ou seja, sequência de passos que caracterizam um fluxo de execução (SILVA, 2006). Cada passo representa um processo a ser executado, por exemplo, um programa de *software* a ser executado ou um *webservice* a ser invocado (GIL et. al., 2007).

O *software* Dis2PPI foi desenvolvido como uma ferramenta de *workflow* científico, com o objetivo de gerar uma rede de interação entre doenças monogênicas e proteínas específicas a partir de múltiplos conjuntos de dados (NOTARI et. al., 2012). Uma rede de interação mostra associações físicas e funcionais entre as proteínas (JENSEN et. al., 2009). A ferramenta utiliza diferentes bancos de dados disponíveis na *web*, tais como o OMIM e STRING, para busca e validação dos dados.

1.2 PROBLEMA E QUESTÃO DE PESQUISA

Atualmente a ferramenta Dis2PPI encontra-se apenas na plataforma *desktop* e está disponível para *download*¹. Devido à ferramenta ser *desktop* há alguns problemas na sua utilização:

- Licença GPL: a ferramenta Dis2PPI é um *software* livre e possui a licença GNU *General Public License* que garante que todas as versões modificadas e estendidas do programa também sejam *software* livre. Porém como a aplicação está disponível para *download*, não é possível garantir que a mesma não seja comercializada ou adequada para um determinado fim, diferente do proposto.
- Bibliotecas: a ferramenta utiliza bibliotecas existentes na Internet e, devido a restrições, não é possível disponibilizar o *software* com as mesmas. É necessário que o usuário realize o *download* separadamente, conforme explicado no manual de instalação do Dis2PPI.
- Versão do Java: hoje a ferramenta está homologada na versão 1.6 do Java. Caso seja disponibilizada para *download* uma nova versão homologada na versão atual do Java, os usuários terão de realizar o *download* da ferramenta novamente, pois a ferramenta não possui nenhuma forma de atualização automática.
- Proxy: o *software* possui tratamento para trabalhar com *proxy* de rede, porém há a necessidade de configuração do mesmo e pode passar imperceptível ao usuário na utilização da ferramenta, gerando problemas durante a execução.

Baseado no problema de pesquisa descrito anteriormente foi criado a seguinte questão de pesquisa:

“Como gerenciar o acesso ao *software* Dis2PPI, transferindo-o da plataforma *desktop* para a plataforma *web*?”.

1.3 OBJETIVOS

Baseado nos problemas descritos anteriormente foi necessário identificar uma forma de resolver essas questões. Para tal, facilitar o acesso à ferramenta Dis2PPI que se encontra na

¹ *Download* disponível em <http://www.danlian.com.br/dis2ppi>

plataforma *desktop*, transferindo a aplicação para a plataforma *web*, deixando a aplicação sempre disponível e resolvendo problemas com a instalação da aplicação, licenças, bibliotecas restritas e restrições de *proxy* de rede. Para atingir esse objetivo, será necessário:

- Buscar conhecimento teórico sobre técnicas de transferência de *softwares* da plataforma *desktop* para plataforma *web*.
- Encontrar outras aplicações relacionadas à comparação de código genético, para realização de comparações de *workflows* científicos.
- Migrar a aplicação da versão 1.6 do Java para a versão mais recente.
- Realizar a troca da interface para a plataforma *web*, de acordo com a melhor técnica identificada.
- Realizar tratamento para arquivos salvos utilizados durante o processo de análise da aplicação *desktop*.
- Realizar tratamento diferenciado para as mensagens de *LOG* geradas pela aplicação *desktop* em forma de arquivo, que mostra informações de erros de programa, problemas de rede, erros de proteínas não existentes e registro de *links* acessados.
- Providenciar hospedagem da aplicação *web*.
- Realizar testes comparando a aplicação *desktop* e a aplicação *web*.

1.4 ESTRUTURA DO TEXTO

Prevendo alcançar os objetivos citados acima, este trabalho foi estruturado da seguinte forma: uma conceituação referente aos termos envolvidos na ferramenta Dis2PPI, uma explanação de como a ferramenta foi desenvolvida e seu funcionamento e uma comparação com outras ferramentas de *workflow* científico. Além disso, um levantamento referente à como as principais linguagens de programação atuais tratam a questão de migração da aplicação. Após essa conceituação, inicia-se a proposta de solução a ser realizada na ferramenta Dis2PPI. Com base na proposta descrita, é iniciada a implementação das mudanças na ferramenta para atingir os objetivos propostos. Ao finalizar a implementação, é criado um estudo de caso e aplicados vários casos de testes na aplicação para garantir que os objetivos propostos sejam atendidos. Por fim, uma conclusão mostrando os resultados obtidos.

2 REVISÃO BIBLIOGRÁFICA GERAL

Este capítulo apresenta uma revisão bibliográfica referente os conceitos utilizados na concepção da ferramenta Dis2PPI na área biológica e computacional, esclarecendo os termos e o funcionamento da ferramenta. Além disso, apresentam-se algumas informações para facilitar o entendimento da proposta de mudança da ferramenta para a plataforma *web*.

2.1 CONCEITUAÇÃO

A ferramenta Dis2PPI, criada na área de biologia de sistemas, trabalha com proteínas e suas interações, utilizando a base de dados OMIM e o programa de metabuscas STRING para obtenção de seus resultados, com a possibilidade de integração com a ferramenta Cytoscape. Nesta seção, será realizada uma revisão destes assuntos para facilitar o entendimento da ferramenta.

2.1.1 Proteínas, Proteomas e Interatomos

Segundo KARP (2006), proteínas são macromoléculas relacionadas com todas as atividades celulares. Elas são o instrumento molecular que possibilita a atividade celular. Proteoma representa o conjunto completo de proteínas codificadas por um genoma, mas também é frequentemente usado para referir o complemento total das proteínas de uma célula (KLUG et. al., 2010).

Segundo Lesk (2008), proteínas são “como animais sociais, e a vida depende de suas interações”. Como proteínas individuais possuem funções especializadas, são necessários mecanismos de controle para as suas atividades. A biologia de sistemas descreve as interações metabólicas e reguladoras em termos de redes de interações (LESK, 2008).

Uma rede de interação de proteína, também conhecida como interatoma, mostra associações físicas e funcionais entre as proteínas (JENSEN et. al., 2009). Nas células, as duas redes de interações atuam em paralelo (LESK, 2008): (1) uma rede física de complexos proteína-proteína e proteína-ácido nucleico e (2) uma rede lógica de conexões dinâmicas nas quais a atividade de um processo é afetada por uma alteração nas condições externas ou pela

atividade de outro processo. Exemplos de interações puramente físicas incluem a formação de centros de reação fotossintéticos, complexos de proteína e co-fatores que convertem luz em energia química, e a associação de fibras de colágeno no tecido conjuntivo (LESK, 2008). Exemplos de interações lógicas, não mediadas inteiramente pela interação física direta entre proteínas, incluem os ciclos de retroalimentação, nos quais o aumento na concentração de um produto pode inibir uma enzima, ou a secreção de uma molécula como sinal para outras células (LESK, 2008).

2.1.2 **Workflow Científico**

Workflows científicos são usados para descrever experimentos *in silico* em bioinformática (SILVA, 2006; FREIRE et. al., 2006). Estes *workflows* são construídos através de uma sequência de passos que caracterizam um fluxo de execução onde, cada um destes passos usam programas de terceiros (SILVA, 2006). Cada execução do *workflow* é considerada um experimento e, de acordo com o resultado alcançado, os parâmetros podem ser revistos e o *workflow* é novamente executado ou, eventualmente é preciso realizar algumas execuções parciais novamente (SILVA, 2006; FREIRE et. al., 2006). O resultado de um *workflow* pode vir a ser uma entrada em outro *workflow* a ser executado (FREIRE et. al., 2006).

2.1.3 **OMIM**

A base de dados OMIM², utilizada pela ferramenta Dis2PPI, é uma base de dados que contém um catálogo de doenças genéticas mantido pela NCBI (BARNES e GRAY, 2003; LESK, 2008; AMBERGER et. al., 2009). As pesquisas podem ser realizadas no *site* por palavras-chave ou através do uso das ferramentas de consulta do Entrez (BARNES e GRAY, 2003; LESK, 2008; AMBERGER et. al., 2009).

² <http://www.ncbi.nlm.nih.gov/omim>

2.1.4 STRING

O STRING é uma ferramenta para realização de metabuscas com o intuito de se obter redes de interação de proteínas (JENSEN et. al., 2009). Possui um banco de dados sobre predições e interações de proteínas conhecidas, envolvendo associações físicas e funcionais (JENSEN et. al., 2009). Atualmente³, o STRING possui informações sobre mais de 1,1 mil organismos e mais de cinco milhões de proteínas.

2.1.5 Cytoscape

A ferramenta Cytoscape⁴ permite visualizar as informações referentes a uma rede de interação das proteínas e fazer análises de biologia de sistemas usando os *plugins* desenvolvidos para aplicações em conjunto com este *software* (SHANNON et. al., 2003). A flexibilidade da ferramenta encorajou vários usuários a adotá-la e adaptá-la em suas pesquisas (KILLCOYNE et. al., 2009). Embora a ferramenta tenha sido originalmente desenhada para pesquisa biologia, atualmente o Cytoscape é uma plataforma geral para análise de redes complexas e visualização (KILLCOYNE et. al., 2009; <http://www.cytoscape.org>).

2.2 A FERRAMENTA DIS2PPI

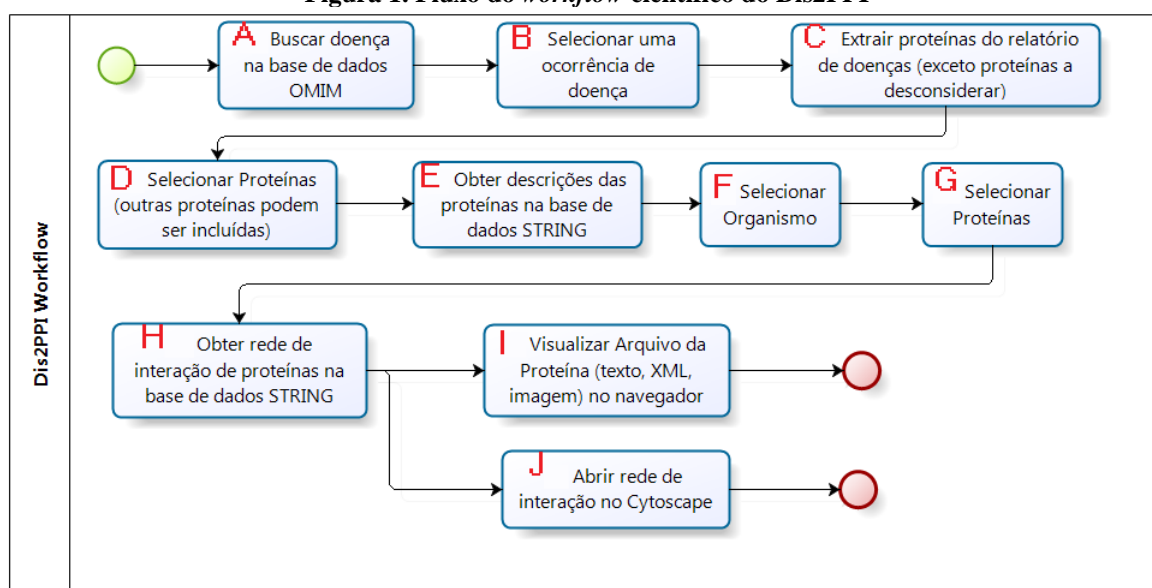
O *software* Dis2PPI foi desenvolvido como uma ferramenta de *workflow* científico, com o objetivo de gerar uma rede de interação entre doenças monogênicas e proteínas específicas a partir de múltiplos conjuntos de dados (NOTARI et. al., 2012). A aplicação extrai nomes de proteínas de relatórios de doenças e se utiliza dessa informação para criar a rede de interação.

A ferramenta foi desenvolvida na linguagem Java, no JDK 6 (*Java Development Kit Version 6*). Utiliza o banco de dados de doenças monogênicas OMIM e o banco de dados de interações de proteína STRING, para busca e validação dos dados. O acesso é realizado via *webservice* ou URL (*Uniform Resource Locator*).

³ Informação disponível em 20/10/2013 no site <http://www.string-db.org/>

⁴ <http://www.cytoscape.org>

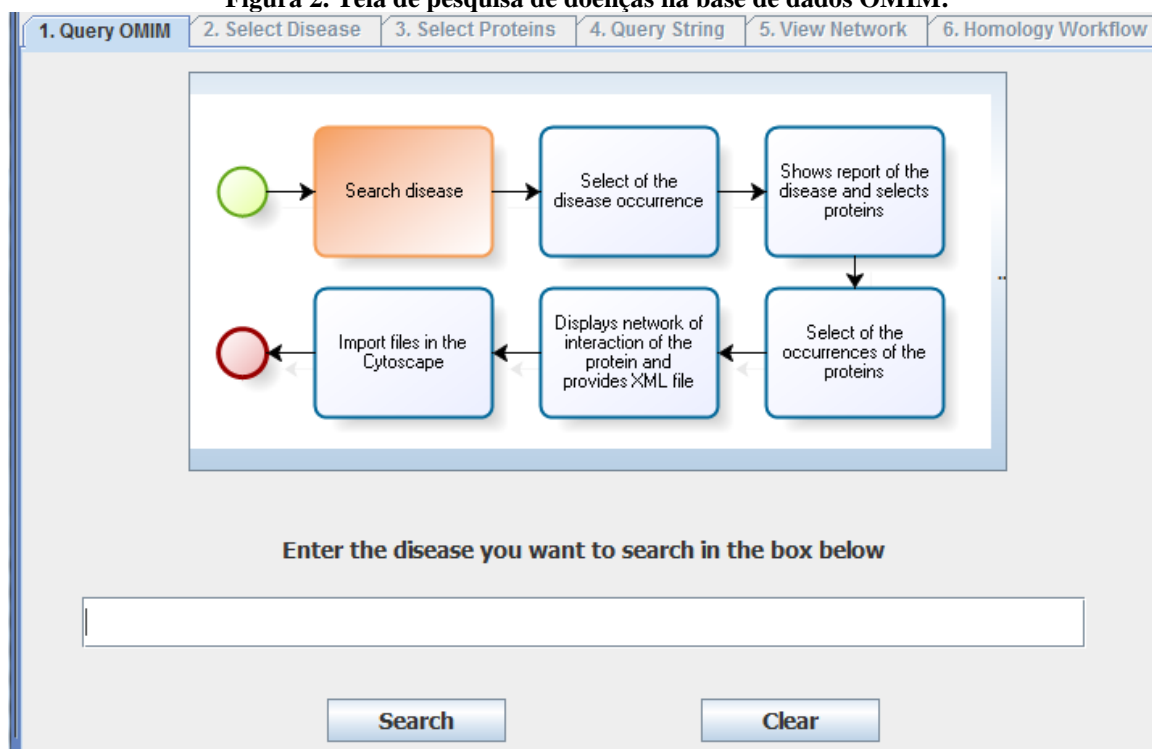
Figura 1. Fluxo do *workflow* científico do Dis2PPI



Fonte: Imagem adaptada de NOTARI et. al., 2012

A Figura 1 apresenta o *workflow* científico da ferramenta DIS2PPI. Esta ferramenta realiza uma busca (Figura 1A) através de palavras-chave na base de dados da OMIM, visualizado na Figura 2, comunicando-se através de uma chamada de um *webservice* utilizando a biblioteca Apache Axis2 for Java (AMBERGER et. al., 2009).

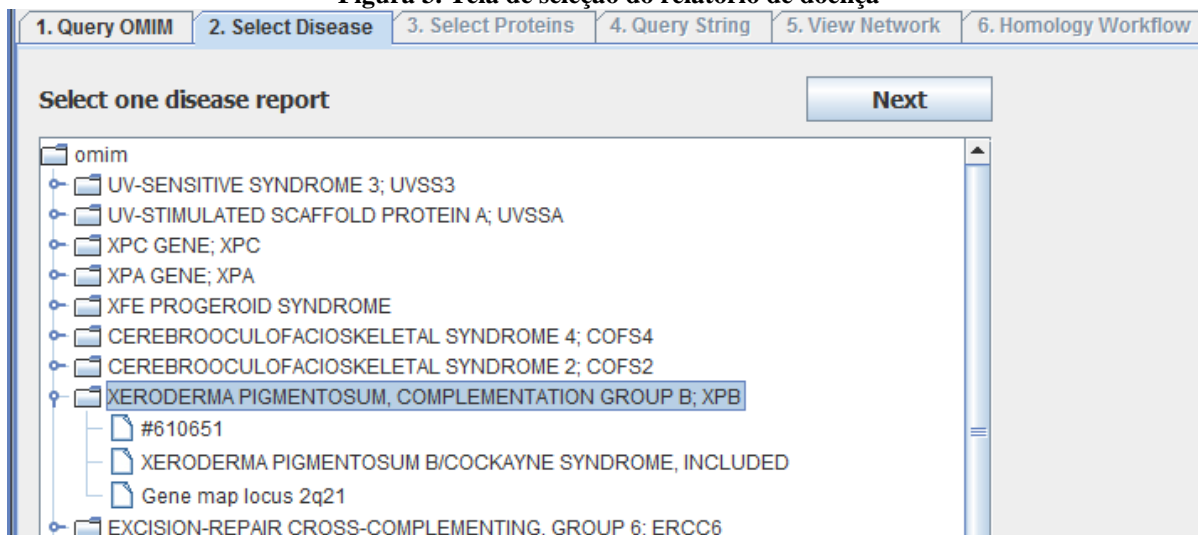
Figura 2. Tela de pesquisa de doenças na base de dados OMIM.



Fonte: Autor

Como retorno recebe um documento HTML (*HyperText Markup Language*) em forma de um relatório com todas as doenças monogênicas relacionadas à palavra-chave informada, para avaliação do usuário. O *software* apresenta essas informações em forma de árvore, conforme Figura 3, possibilitando selecionar o dado desejado (Figura 1B).

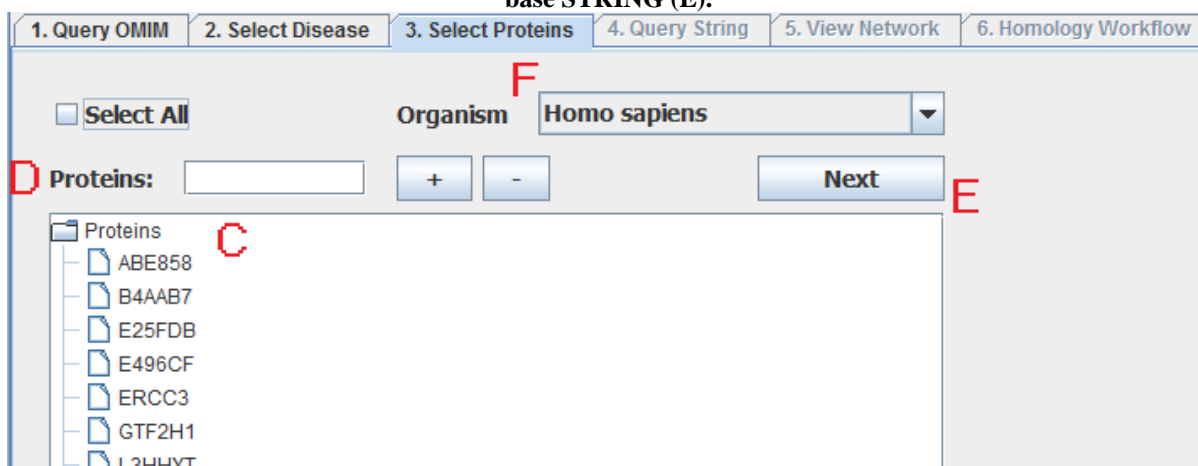
Figura 3. Tela de seleção do relatório de doença



Fonte: Autor

Após a seleção do dado, o *software* realiza uma nova pesquisa, extraindo do relatório o nome das proteínas e mostrando as mesmas ao usuário em uma nova árvore (Figura 1 e Figura 4C). As proteínas mostradas na árvore também são passíveis de seleção, com a possibilidade de inclusão de outras proteínas manualmente (Figura 1 e Figura 4D). Ao finalizar a seleção é gerada uma string URL para realização de uma validação das proteínas selecionadas na base de dados STRING (Figura 1 e Figura 4E). Essas funcionalidades são mostradas na Figura 4.

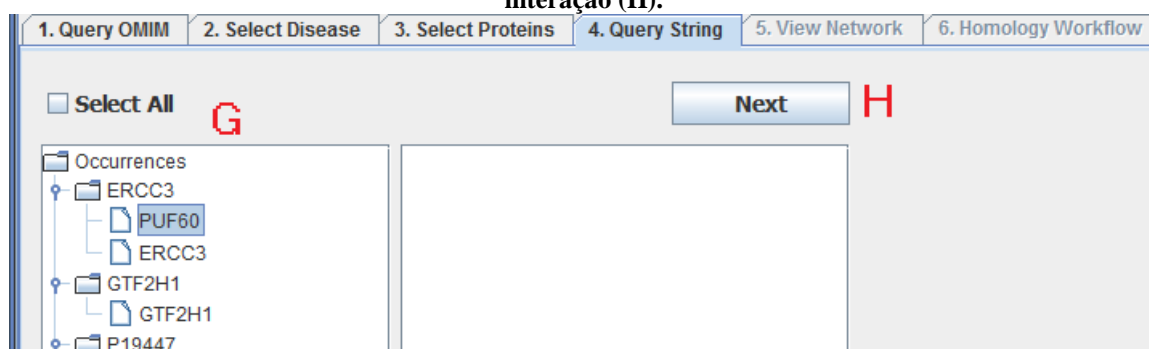
Figura 4. Tela de seleção de proteínas, onde é possível a seleção das proteínas do relatório de doença (C), ou a possibilidade de inclusão de novas proteínas (D) além da escolha do organismo (F) para validação na base STRING (E).



Fonte: Autor

Além da seleção das proteínas, deve-se selecionar o organismo no qual as proteínas serão verificadas (Figura 1 e Figura 4F). Proteínas inexistentes são descartadas neste momento pela aplicação e mostradas ao usuário no relatório de erros. Com base nas proteínas válidas restantes, mostradas na Figura 5, é possível selecionar proteínas (Figura 1 e Figura 5G) para realização de uma nova busca na base de dados STRING para obtenção da rede de interação (Figura 1 e Figura 5H).

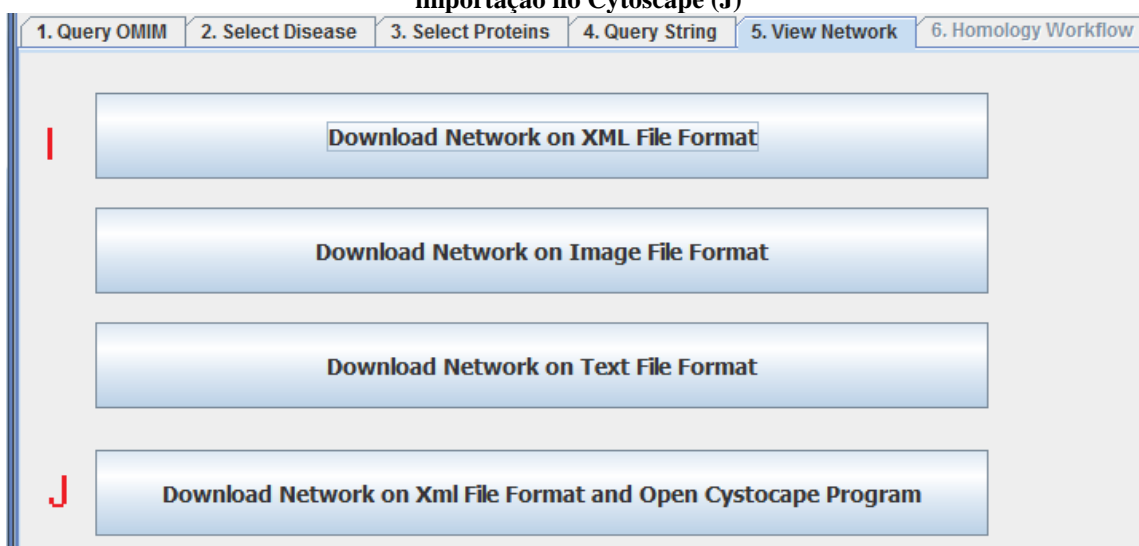
Figura 5. Tela de Ocorrências de proteínas validadas na base STRING (G) e obtenção da rede de interação (H).



Fonte: Autor

O resultado pode ser mostrado pelo navegador da *web* em forma de texto, XML (*eXtensible Markup Language*) ou imagem (Figura 1 e Figura 6I), ou ainda pode ser importado pela ferramenta Cytoscape (SHANNON et. al., 2003; <http://www.cytoscape.org>) para análise (Figura 1 e Figura 6J), conforme Figura 6. A aplicação *desktop* também possibilita uma avaliação do resultado pelo *Homology Workflow*, porém este *workflow* não está no contexto deste trabalho.

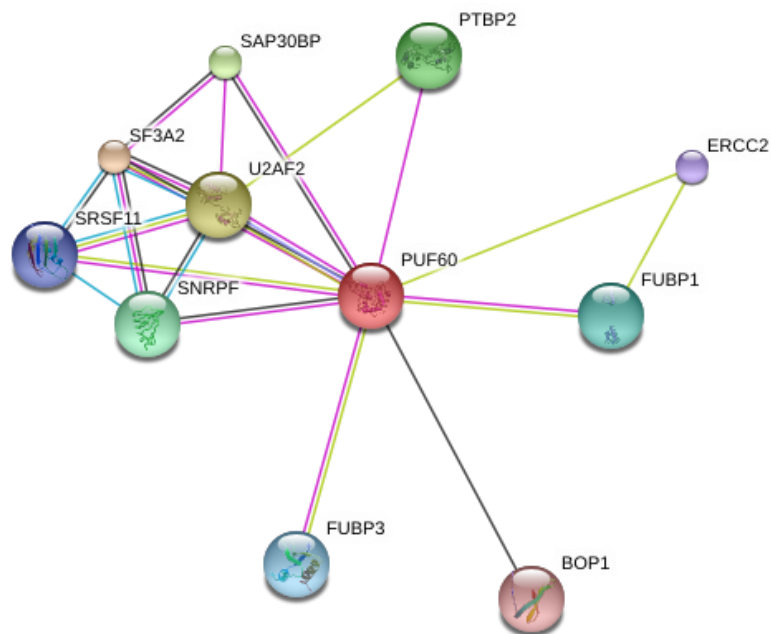
Figura 6. Tela de visualização das redes de interação com as opções de XML, Imagem, Texto (I) ou para importação no Cytoscape (J)



Fonte: Autor

O exemplo utilizado para mostrar o funcionamento da ferramenta pesquisando pela doença “XERODERMA” na base de dados OMIM e selecionando a ocorrência “PUF60” após a validação na base STRING tem como resultado a rede de interação mostrada na Figura 7.

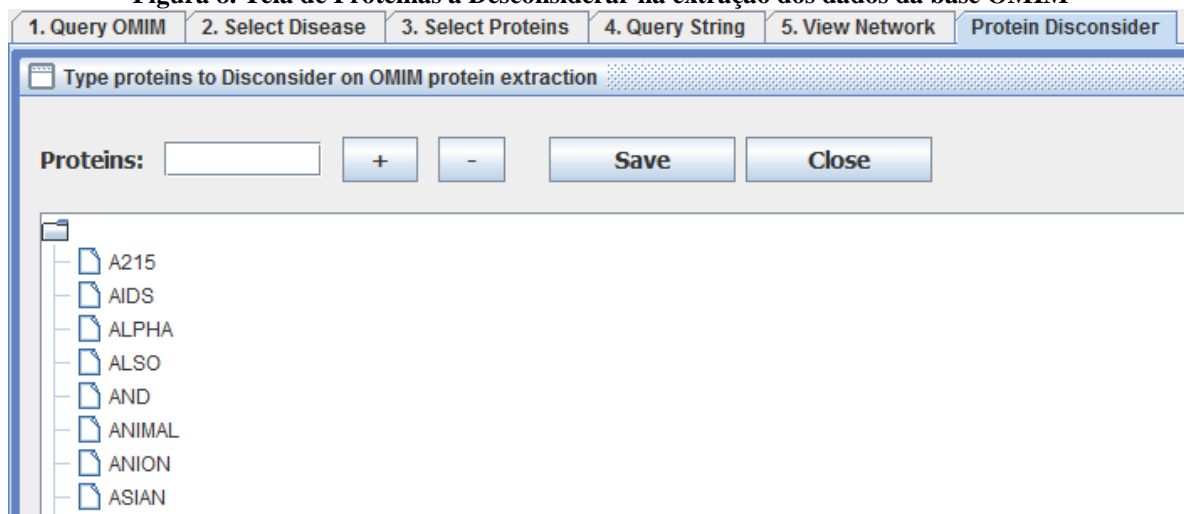
Figura 7. Rede de Interação da proteína PUF60, utilizada como exemplo do funcionamento da ferramenta Dis2PPI.



Fonte: Autor

Na aplicação, também existe a possibilidade de realizar o cadastro de proteínas a desconsiderar, conforme Figura 8. Já existem proteínas pré-cadastradas nesta lista, e é possível ao usuário incluir novas. As proteínas cadastradas nesta tela são desconsideradas ao realizar a validação na base OMIM (Figura 3).

Figura 8. Tela de Proteínas a Desconsiderar na extração dos dados da base OMIM



Fonte: Autor

Para o funcionamento da ferramenta Dis2PPI na plataforma *desktop* são necessárias algumas parametrizações a serem realizadas (Figura 9), são elas: *i*) definição do navegador utilizado para execução de determinados arquivos (A); *ii*) definição do caminho onde a ferramenta Cytoscape está instalada (B); *iii*) definições de *Proxy* de rede (C), caso existam; e *iv*) definição de diretórios temporários para extração de dados (D), que não serão tratados neste trabalho.

Figura 9. Tela de Configuração da ferramenta Dis2PPI

The screenshot shows the 'Applications Settings' window for Dis2PPI. It is organized into several sections:

- Tools Settings:** Contains two rows. The first row, labeled 'A', is for the 'Browser' and has a 'Choose Application' button and a text field containing 'C:\Program Files (x86)\Mozilla Firefox\firefox.exe'. The second row, labeled 'B', is for 'Cytoscape' and has a 'Choose Application' button and a text field containing 'C:\Cytoscape_v2.7.0\Cytoscape.exe'.
- Temp Folder:** Labeled 'D', it contains a 'Folder' field with a 'Choose Temp Folder' button. Below it are five text fields for different data types: 'String's Network' (C:/Temp/string), 'Blast Results' (C:/Temp/blast), 'Fasta' (C:/Temp/fasta), 'PDB' (C:/Temp/pdb), and 'NetworkBlast' (C:/Temp/nct).
- Proxy:** Labeled 'C', it contains two text fields for 'Host' and 'Port', both of which are currently empty.
- Buttons:** At the bottom of the window are two buttons: 'Save' and 'Close'.

Fonte: Autor

2.3 FERRAMENTAS DE WORKFLOWS CIENTÍFICOS

Nesta seção serão analisadas algumas ferramentas de *workflows* científicos existentes, que trabalham de forma semelhante a Dis2PPI. Ao fim da seção é apresentado um resumo comparando estas ferramentas, observando detalhes como tipo de dado, tipo de retorno, quantidade de bases de dados, funcionamento, forma de exportação dos dados, linguagens utilizadas e plataforma na qual a aplicação é executada. As ferramentas analisadas foram DASMIweb, JBioWH e SNOW.

2.3.1 Ferramenta DASMIweb

A ferramenta DASMIweb foi desenvolvida como um portal de dados de interatoma de vários repositórios de dados disponíveis na *web*, usando as linguagens de programação Java e Perl para suporte a diferentes níveis de interações moleculares (BLANKENBURG et. al., 2009). A aplicação fornece acesso a trinta e cinco bancos de dados⁵ diferentes, contendo interações de proteínas e domínios proteicos experimentais.

Esta ferramenta⁶ funciona da seguinte forma: *i*) inicialmente deve-se informar um identificador de gene existente em uma das bases de dados UniProtKB, RefSeq, Entrez Gene, Ensembl, GeneInfo, e Pfam. Atualmente a plataforma não realiza a pesquisa pelo nome do gene, por exemplo, "BRCA2"; *ii*) após informar o identificador, deve-se clicar no botão "Query" (A), conforme mostrado na Figura 10.

Figura 10. Tela inicial do DASMIweb

Fonte: Autor

iii) A ferramenta retorna o resultado, em (B), mostrado na Figura 11, aparecem informações da consulta, tais como, *links* de acesso às bases de dados, descrição, espécie, tipo, entre outros.

Figura 11. Resultado da consulta no DASMIweb – Informações do Identificador

Interactor Information	
B	<p>Identifier: 961 (Entrez Gene) (Leukocyte surface antigen CD47 precursor (Integrin-associated protein)(IAP) (Antigenic surface determinant protein OA3) (Protein MER6).) (CD47 molecule)</p> <p>UniGene Hs.446414,</p> <p>UniProtKB CD47_HUMAN, A8K198_HUMAN, Q71A41_HUMAN,</p> <p>Entrez Gene 961,</p> <p>UniParc UPI0000049C6C, UPI00001BDF2Z,</p>

Fonte: Autor

⁵ Banco de dados disponíveis no artigo referenciado.

⁶ Disponível em <http://dasmi.de/documents.php>

iv) Em (C), mostrado na Figura 12, todos os detalhes das interações são mostrados, seguidos dos detalhes de todos os interatores participantes; v) ao clicar nos *links* (D) da base de dados destacados, aparecem detalhes sobre o interatoma e *links* (E) para acesso a essas informações; e, vi) a ferramenta também disponibiliza os dados para *download* (F) nos formatos SIF (*Simple Interaction Format*) utilizado para visualização no Cytoscape e no formato MITAB 2.5 tabular especificado pela HUPO-PSI.

Figura 12. Resultado da consulta no DASMIweb – Interações

The screenshot shows the DASMIweb interface with a table of interactions and a detailed view of an interactor. The table has columns for Name, ID, Description, and four source categories: BIOVERSE, INTACT, HPIDB, and HOMOLINT. The detailed view shows information for interactor 7059, including its link, confidence score, query string, species, type, description, synonyms, chromosome, location, and a detailed description.

27 interactions (showing 1 to 27)				5	X	X	1
				BIOVERSE	INTACT	HPIDB	HOMOLINT
Name	ID	Description					
null	5058	Serine/threonine-protein kinase PAK 1 (EC 2.7.11.1) (p21-activatedkinase 1) (PAK-1) (P65-PAK) (Alpha-PAK).					
COMP	1311	cartilage oligomeric matrix protein					
THBS1	7057	thrombospondin 1					
THBS2	7058	thrombospondin 2					
THBS3	7059	thrombospondin 3					

E Details from source Bioverse

link	http://bioverse.compbio.washington.edu/
originalConfidenceScore	0.277000010013580322265625
DAS query string	http://dasmi.bioinf.mpi-inf.mpg.de/das/bioverse/interaction?interactor=961

Additional details for interactor 7059:

species	9606
type	protein-coding
description	thrombospondin 3
synonyms	MGC119564 MGC119565 TSP3
chromosome	1
location	1q21
description	Thrombospondin-3 precursor.

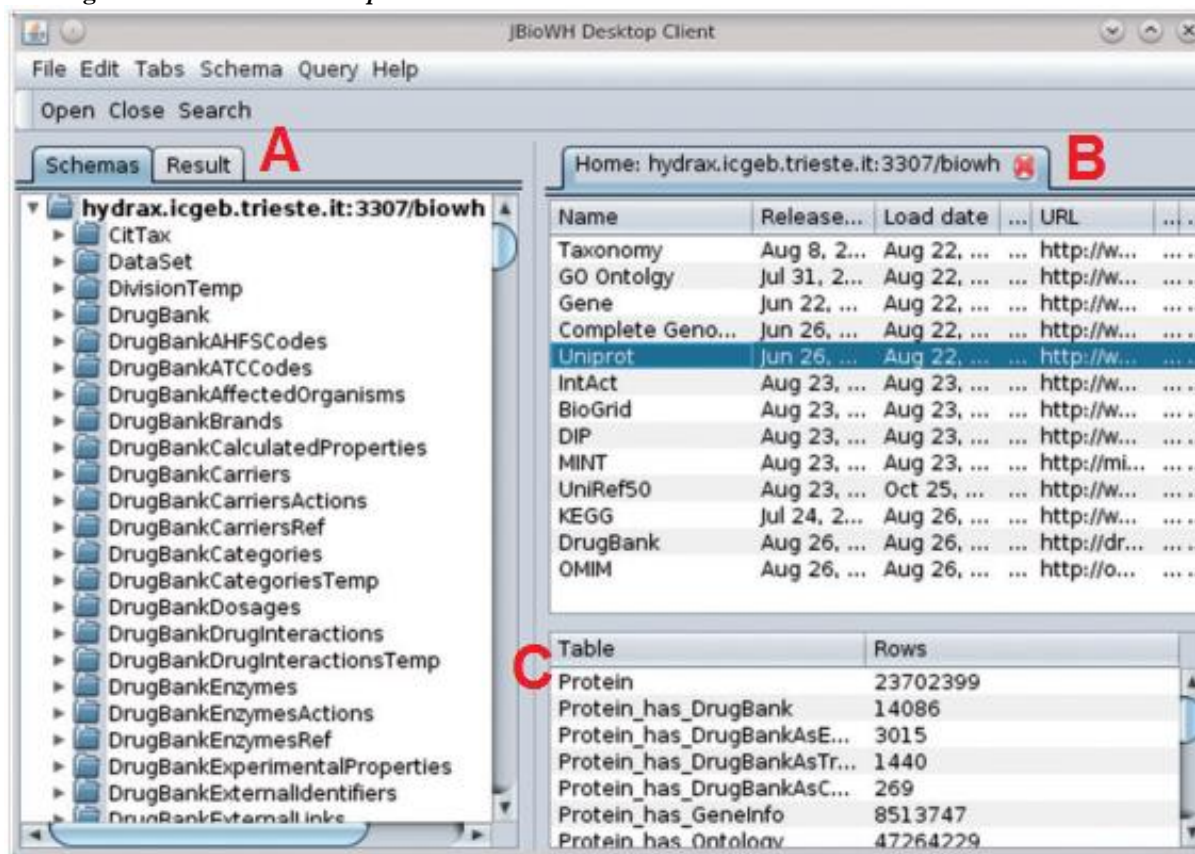
Fonte: Autor

2.3.2 Ferramenta JBioWH

A ferramenta JBioWH (Java BioWareHouse) é um *framework* com o objetivo de realizar pesquisas em múltiplas bases de dados de forma flexível (VERA et. al., 2013). A aplicação tem código aberto e foi dividida da seguinte forma: *i*) a bases de dados contém o acesso a vinte bases de dados públicas e está em contínuo desenvolvimento, incluindo novas bases e funcionalidades à ferramenta, *ii*) um esquema relacional estruturado em uma base de dados relacional que contém trezentas e sessenta e três tabelas para armazenamento dos

dados, referências cruzadas e aumento da velocidade nas pesquisas, *iii*) uma API que inclui um pacote núcleo para manutenção da ferramenta, *iv*) uma aplicação *Desktop* para uso dos usuários e, *v*) um pacote auxiliar que contém linhas de comando para executar rápidas consultas.

Figura 13. Interface *Desktop* da ferramenta JBioWH com as bases de dados a serem consultadas



Fonte: Imagem adaptada de VERA et. al., 2013

A aplicação *Desktop* foi criada para usuários não familiarizados com scripts SQL (*Structured Query Language*) e programação Java. Fornece uma interface gráfica para acessar, manipular e executar consultas complexas apenas clicando nas bases de dados integradas, conforme mostrado na Figura 13. O painel esquerdo (A) mostra as bases de dados abertas, no lado superior direito (B) são visualizadas as bases de dados incluídas no esquema relacional, enquanto no lado inferior direito (C) é possível visualizar as tabelas da base de dados selecionada.

Ao selecionar a base de dados, inicia-se a pesquisa principal pelo tipo de dados selecionado que pode ser por taxonomia, proteína, gene, entre outros. Também é possível realizar a pesquisa por outros parâmetros para melhorar o desempenho da consulta. Após selecionar os parâmetros da pesquisa, é criada uma lista com os resultados. Os resultados

podem ser salvos e, baseado neles pode ser iniciada uma nova pesquisa, refinando ainda mais os resultados a serem obtidos. Mais detalhes sobre o funcionamento do aplicativo estão disponíveis na internet⁷.

2.3.3 Ferramenta SNOW

Essencialmente, SNOW (*Studying Networks in the Omics World*) recebe uma lista de proteínas (ou genes) e os mapeia em um interatoma (MINGUEZ et. al., 2009). Desenvolvida na plataforma *web* utilizando Perl e JavaScript e utilizando Perl e Python na montagem das consultas para os *webservices*.

Inicialmente, o SNOW recebe como entrada uma coleção de identificadores de proteínas (ou genes) em formato texto ou no formato SIF usado pelo Cytoscape. Com base na coleção inserida, a ferramenta testa quatro parâmetros topológicos: primeiro, o grau de conexão dos nodos, que é computado pelo número de arestas (eventos de interação) de um nodo, segundo, a intermediação, medida pelo menor número de vias de passagem através de um nodo, terceiro, o coeficiente de *cluster*, que mede a ligação da vizinhança e quarto, o número de componentes da rede. Além dos parâmetros, a ferramenta busca o número de bicomponentes e pontos de articulação e os descreve detalhadamente.

Como resultado, a ferramenta contém um resumo (Figura 14A) dos dados de entrada e dos parâmetros de rede (Figura 15B) estimados juntamente com a significância estatística para o coeficiente de conectividade, intermediação e *cluster*.

Figura 14. Resultados da consulta realizada no SNOW – Parte 1

The screenshot shows the SNOW web application interface. At the top, there are logos for 'Bioinformatics' and 'PRINCIPE TELIFE CENTRO DE INVESTIGACIONES'. The main title is 'snow Studying Networks in the Omic World : results'. Below the title, there are links for 'Help', 'References', and 'Send comments'. The 'Input Parameters' section is highlighted with a red 'A' and lists the following parameters and options:

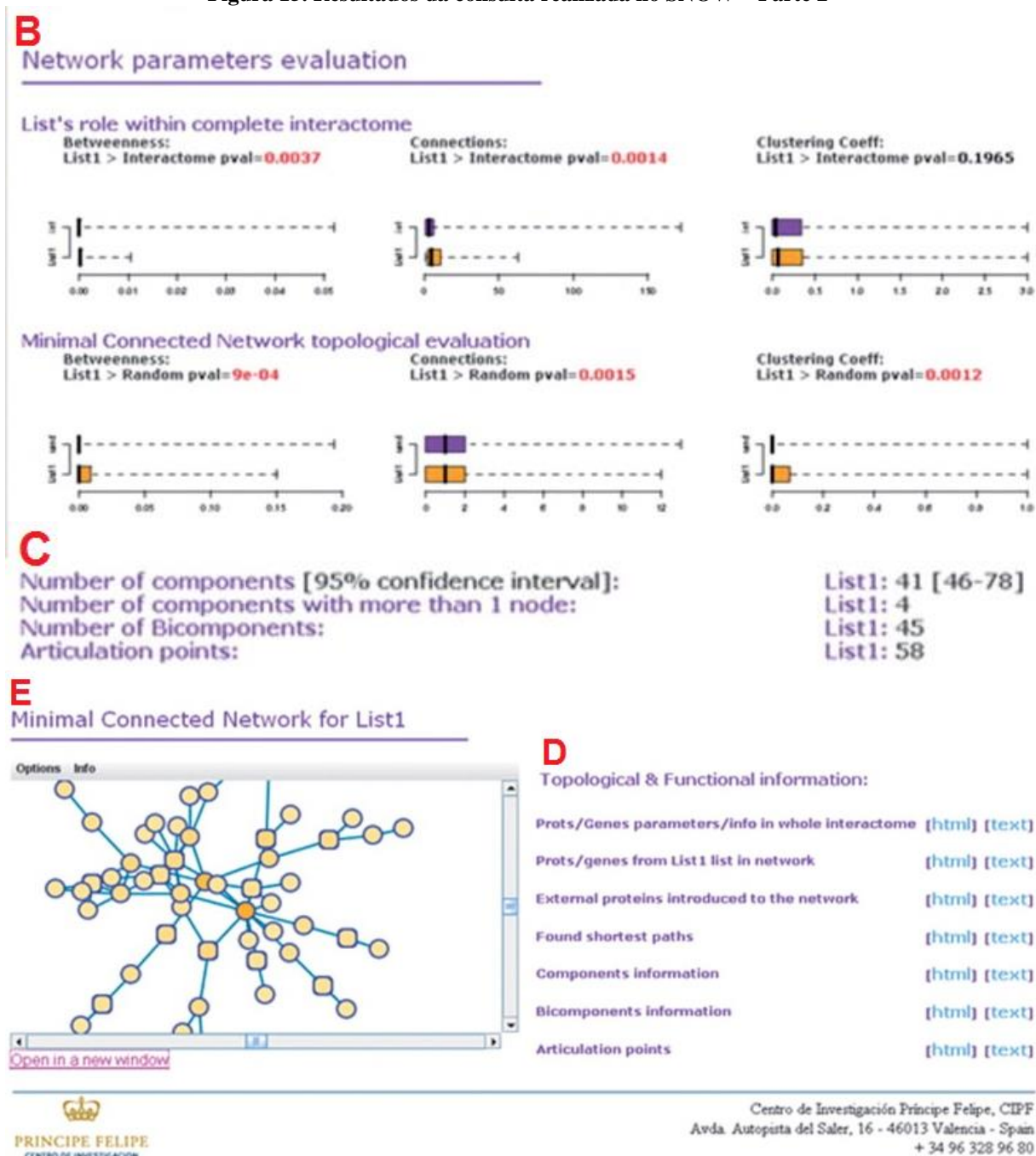
- List1 list: [html] [text]
- unknown prots/genes in List1 list: [html] [text]
- Interactome: Interactions in at least two of the methods [text]
- Max number of external proteins introduced: 1
- Project name: serum_fibroblast_cellcycle
- Download all results: zip file

Fonte: Imagem adaptada de MINGUEZ et. al., 2009

⁷ Disponível em <http://code.google.com/p/jbiowh/wiki/Ex1SearchTab>

Componentes, bicomponentes e pontos de articulação também são fornecidos (Figura 15C). Além disso, a informação detalhada dos genes na lista, os genes incluídos na análise, os caminhos mais curtos também são acessíveis a partir de *links* (Figura 15D). Por fim, um visualizador de rede interativa (Figura 15E) pode ser iniciado a partir da página.

Figura 15. Resultados da consulta realizada no SNOW – Parte 2



Fonte: Imagem adaptada de MINGUEZ et. al., 2009

2.3.4 Comparação das ferramentas

Conforme Tabela 1, as ferramentas trabalham de forma semelhante, algumas inclusive consultam as mesmas bases de dados, porém mostram o resultado de forma diferente. Os critérios de comparação utilizados foram: *i)* linguagem, para identificar as linguagens utilizadas pelas ferramentas criadas; *ii)* plataforma, identificando se a ferramenta foi criada para *desktop* ou *web*; *iii)* quantidade de bases de dados, onde identifica-se quantas bases de dados a ferramenta tem acesso; *iv)* tipos de dados para pesquisa, onde identifica-se quais as entradas necessárias para obtenção dos resultados; *v)* tipo de retorno, onde é identificado o tipo de resultado da ferramenta; *vi)* exportação de dados, onde identifica-se as formas de exportação de dados da ferramenta para análise em outras ferramentas; e o principal, *vii)* funcionamento, onde identifica-se as facilidades na utilização da ferramenta.

Tabela 1. Tabela de comparação entre as Ferramentas

Ferramenta	Dis2PPI	DASMIweb	JBioWH	SNOW
Linguagem	Java	Java e Perl	Java	Perl, JavaScript e Python
Plataforma	<i>Desktop</i>	<i>Web</i>	<i>Desktop</i>	<i>Web</i>
Bases de Dados	2	35	25	5
Tipo de dado para pesquisa	Proteína	Proteína, gene	Compostos químicos, reações bioquímicas, caminhos metabólicos, proteínas, organismos, genes, sequências de proteína e ácido nucleico, taxonomias.	Proteína, gene
Tipo de Retorno	Rede de interação de proteína	<i>Links</i> de interação entre os tipos de dados selecionados.	Consultas dos tipos de dados específicos e a relações entre elas.	Interatoma, resultados de cálculos relevantes para busca do interatoma, dados referente componentes, bicomponentes e articulações.

Ferramenta	Dis2PPI	DASMIweb	JBioWH	SNOW
Funcionamento	Foi desenvolvida para mostrar os passos a serem seguidos para realização das consultas de forma simples.	Trabalha de forma simples através da consulta, porém a mesma é feita por identificadores, o que dificulta as pesquisas para alguém sem conhecimento técnico das bases de dados.	A aplicação <i>Desktop</i> tem vários recursos, porém alguns deles podem ser complicados de serem usados sem alguns conhecimentos técnicos.	Simple de utilizar, com informações sobre as redes de interação de proteínas, porém com certas limitações.
Exportação de dados	Texto, Imagem, XML	SIF, MITAB2.5	Indisponível	HTML e Texto

Fonte: Autor

Analisando as informações da tabela, podem-se observar as seguintes informações: *i)* a linguagem normalmente utilizada nas ferramentas da bioinformática é a linguagem Java, algumas aplicações novas estão utilizando *python* por sua facilidade na manipulação de *strings*; *ii)* há uma variedade de aplicações tanto na plataforma *desktop* quanto na plataforma *web*, ferramentas mais complexas tendem a ser criadas inicialmente na plataforma *desktop* devido ao processamento; *iii)* as ferramentas trabalham com as mais diversas bases de dados, dependendo da necessidade a qual a mesma atende; *iv)* os tipos de dados dependem dos objetivos da ferramentas; e seu *v)* resultado é uma consequência deste objetivo; *vi)* o funcionamento, um dos critérios mais relevantes nesta análise, identifica que as aplicações tendem a ter um funcionamento simples para não dificultar o uso pelos profissionais da biologia; por fim *vii)* a maioria das aplicações trabalham nos mesmos padrões de exportação de dados, para que seja possível utilizar os resultados em outras ferramentas.

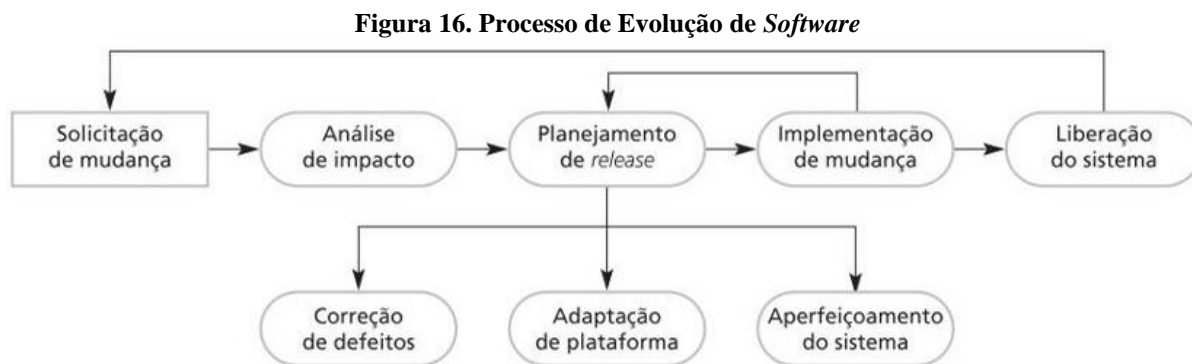
2.4 EVOLUÇÃO DE SOFTWARE

Segundo Sommerville (2011), o processo de desenvolvimento de *software* não é interrompido quando o *software* é entregue, mas continua por toda a vida útil do mesmo.

Qualquer mudança ou defeito pode desencadear uma evolução no *software*, até mesmo uma alteração de *hardware* onde o *software* é utilizado pode acarretar em mudanças para

atender a necessidade (PRESSMAN, 2011; SOMMERVILLE, 2011). Sommerville (2011) complementa que os processos de mudança e evolução de *software* podem variar de acordo com o tipo de *software* que está sendo mantido.

Conforme mostrado na Figura 16, o processo de *software* inicia com uma solicitação de mudança (SOMMERVILLE, 2011). A partir desta, é realizada a análise de impacto da mudança na aplicação. Depois da análise, é realizado o planejamento da *release*, considerando a mudança uma correção de defeitos, adaptação de plataforma ou aperfeiçoamento do sistema. A partir do planejamento, é realizada a implementação da mudança, onde talvez seja necessário um replanejamento da *release*, dependendo do andamento da implementação. Ao finalizar a implementação, é realizada a liberação do sistema e, com base na nova aplicação podem ser geradas novas solicitações de mudanças.



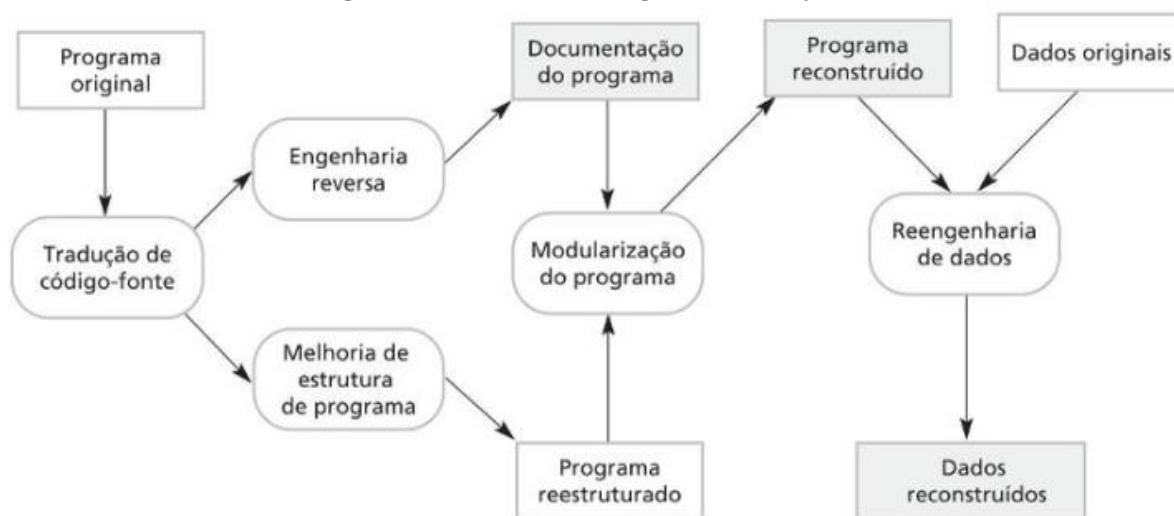
Fonte: Sommerville, 2011

Para a realização das mudanças pode ser necessário executar a etapa de reengenharia de *software* (SOMMERVILLE, 2011), pois algumas mudanças podem implicar em uma reformulação do projeto inicial, adaptando-o a uma nova tecnologia, ou apenas aprimorando as funcionalidades do *software* utilizando novas funcionalidades existentes na linguagem utilizada. Pfleeger (2004) aborda o termo reengenharia como rejuvenescimento de *software*, que trata o desafio da manutenção no sentido de aumentar a qualidade de um sistema existente, analisando seus produtos de trabalho com o objetivo de tentar conseguir mais informações ou reformatá-los de uma forma mais compreensível. Pressman (2011) e Sommerville (2011) também abordam a reengenharia de *software*, onde citam que toda e qualquer mudança realizada em um *software* pode danificar sua estrutura, pois muitas vezes não são aplicadas boas práticas de engenharia durante as alterações, e que se deve aplicar a reengenharia para melhorar a estrutura e inteligibilidade do mesmo, visando mantê-lo de forma mais fácil. A reengenharia pode envolver redocumentação da aplicação, refatoração da arquitetura do sistema, mudança da linguagem de programação para uma linguagem mais

moderna, além de modificações e atualizações de estruturas de dados do sistema (PRESSMAN, 2011; SOMMERVILLE, 2011).

Aplicar a reengenharia proporciona dois benefícios importantes (SOMMERVILLE, 2011): *i*) risco reduzido, pois desenvolver novamente um *software* crítico de negócios tem um risco muito alto, e, *ii*) custo reduzido, pois o custo da reengenharia é significativamente menor do que o desenvolvimento de uma nova aplicação.

Figura 17. Processo de Reengenharia de *Software*



Fonte: Sommerville, 2011

O processo de reengenharia possui as seguintes atividades (SOMMERVILLE, 2011) (Figura 17):

- Tradução de código-fonte: utilização de ferramentas de tradução, onde o programa é convertido a partir de uma linguagem de programação antiga para uma versão mais moderna da mesma linguagem ou outra linguagem diferente;
- Engenharia reversa: o *software* é analisado e a partir dele são extraídas as informações. Isso auxilia a documentar sua organização e funcionalidade;
- Melhoria de estrutura de programa: a estrutura de controle é analisada e modificada para facilitar sua leitura e entendimento;
- Modularização de programa: partes relacionadas de um programa são agrupadas, e se possível, eliminam-se redundâncias. Em alguns casos, esse estágio pode envolver refatoração de arquitetura;
- Reengenharia de dados: os dados processados podem ser alterados para se adequar às mudanças realizadas na aplicação. Isso pode significar alterações no

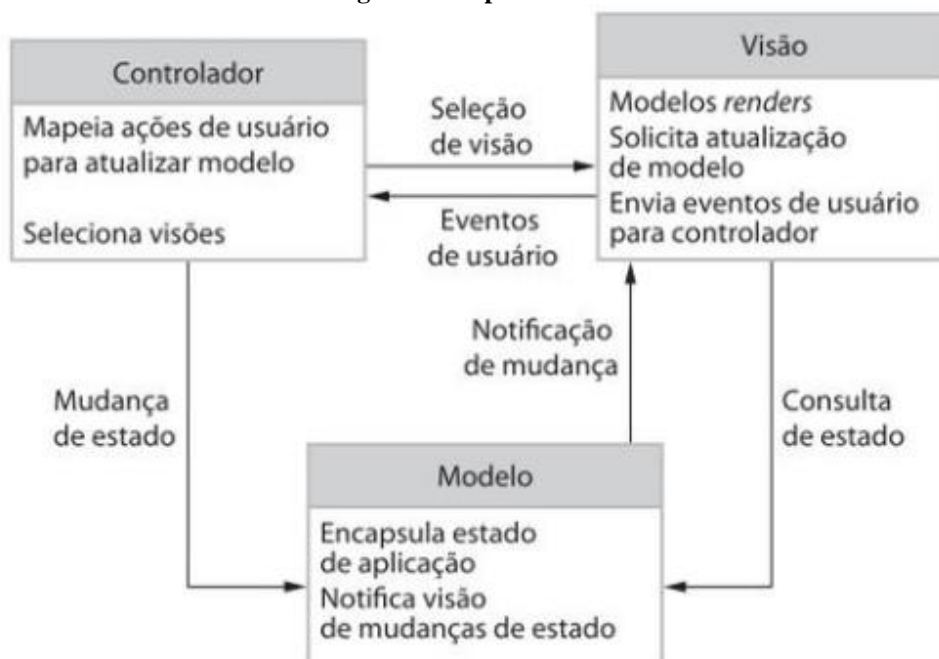
esquema de banco de dados e conversões de dados existentes em novas estruturas;

Pressman (2011) sugere que se o tempo e os recursos estiverem escassos para realização das atividades, é possível aplicar o princípio de Pareto, aplicando a reengenharia aos 20% do *software*, que geram 80% dos problemas. Não é necessário utilizar todas as atividades da reengenharia em um *software*, deve-se adequar a necessidade para manter a aplicação. Algumas dessas atividades dependem de como foi realizada a arquitetura do *software*.

2.5 ARQUITETURA DE SOFTWARE

A arquitetura de *software* representa a estrutura e a organização dos componentes de *software*, suas propriedades e as conexões entre eles (PRESSMAN, 2011). Os componentes incluem módulos de programas e as várias representações de dados manipuladas pelo programa. Sommerville (2011) apresenta alguns padrões de arquitetura comumente usados em diferentes tipos de sistemas, tais como o MVC (*Model-Vision-Controller*), arquitetura em camadas, arquitetura de repositório e arquitetura cliente-servidor.

Figura 18. O padrão MVC



Fonte: Sommerville, 2011

O padrão MVC (Figura 18) é a base do gerenciamento de interação em muitos sistemas baseados na plataforma *web* (SOMMERVILLE, 2011). É estruturado em três

componentes lógicos que interagem entre si (SOMMERVILLE, 2011): *i*) o componente modelo gerencia os sistemas de dados e as operações associadas a esses dados; *ii*) o componente visão define e gerencia como os dados serão apresentados ao usuário; e *iii*) o componente controlador que gerencia as interações do usuário e as transmite para o modelo e a visão.

O padrão de arquitetura em camadas é outra maneira de conseguir a separação e independência, permitindo alterações localizadas (SOMMERVILLE, 2011). O padrão MVC é organizado em camadas (SOMMERVILLE, 2011). As funcionalidades do sistema são organizadas em camadas diferentes (PRESSMAN, 2011; SOMMERVILLE, 2011). Cada camada é independente e fornece serviços para uma camada externa e atua como um cliente para a camada mais interna (PFLEEGER, 2004; SOMMERVILLE, 2011). A arquitetura em camadas é mutável e portátil, ou seja, se houver necessidade de alterar algo em uma camada, apenas a camada adjacente sofrerá alterações (SOMMERVILLE, 2011).

A arquitetura cliente-servidor é muito usada para sistemas distribuídos (por consequência na plataforma *web*) e é composta por um conjunto de servidores que oferecem serviços a outros componentes, um conjunto de clientes que podem chamar os serviços oferecidos pelos servidores e uma rede que permite aos clientes acessar esses serviços (SOMMERVILLE, 2011). Pfleeger (2004) complementa que na maioria dos casos o servidor não tem conhecimento de quantos clientes irão acessá-lo enquanto o sistema estiver em operação, entretanto o cliente conhece a identidade do servidor e envia mensagens para o servidor utilizando uma chamada de procedimento.

2.6 CONSIDERAÇÕES FINAIS

Observando as informações deste capítulo foi identificado que a ferramenta Dis2PPI foi construída de forma a possibilitar a expansão da mesma, tanto na inclusão de novas funcionalidades, quanto à transferência da mesma para a plataforma *web*. A arquitetura da aplicação, por ter sido criada em camadas seguindo o modelo MVC, possibilita a transferência da mesma para a nova plataforma, sem a necessidade de alterações complexas no núcleo da aplicação, por exemplo, não será necessário alterar a busca das proteínas na base STRING,

pois o método atual retorna uma coleção que agora deverá ser visualizada em um componente *web* e não mais em um componente *desktop*.

A ferramenta Dis2PPI foi criada de forma a facilitar a interação com o usuário e devido a isso, a mesma terá o mesmo conceito na proposta para a plataforma *web*. O objetivo principal é deixá-la com o mesmo funcionamento da aplicação *desktop*, para não dificultar a adaptação dos usuários. Alguns detalhes necessitam de maior atenção, tais como o *LOG* da aplicação, onde será necessário descer as diversas camadas para realizar a implementação na plataforma *web*. Também será necessária uma análise referente à importação dos dados na ferramenta Cytoscape, pois não será possível abrir a aplicação diretamente do navegador. Uma possibilidade é a utilização de *plugins* existentes no Cytoscape para utilização na *web*.

Baseado em todas as informações anteriores, o próximo capítulo traz a proposta de transformação da ferramenta para *web*.

3 PROPOSTA DE TRANSFORMAÇÃO DA FERRAMENTA DIS2PPI PARA WEB

A proposta de transformação da ferramenta Dis2PPI para *web* seguirá a metodologia de evolução de *software* de Sommerville (Figura 16). Para tal, serão detalhadas as solicitações de mudanças descritas como objetivos específicos do projeto, realização da análise de impacto dessas mudanças e o planejamento da *release*.

3.1 SOLICITAÇÕES DE MUDANÇAS

As solicitações de mudança correspondem aos objetivos específicos já definidos anteriormente:

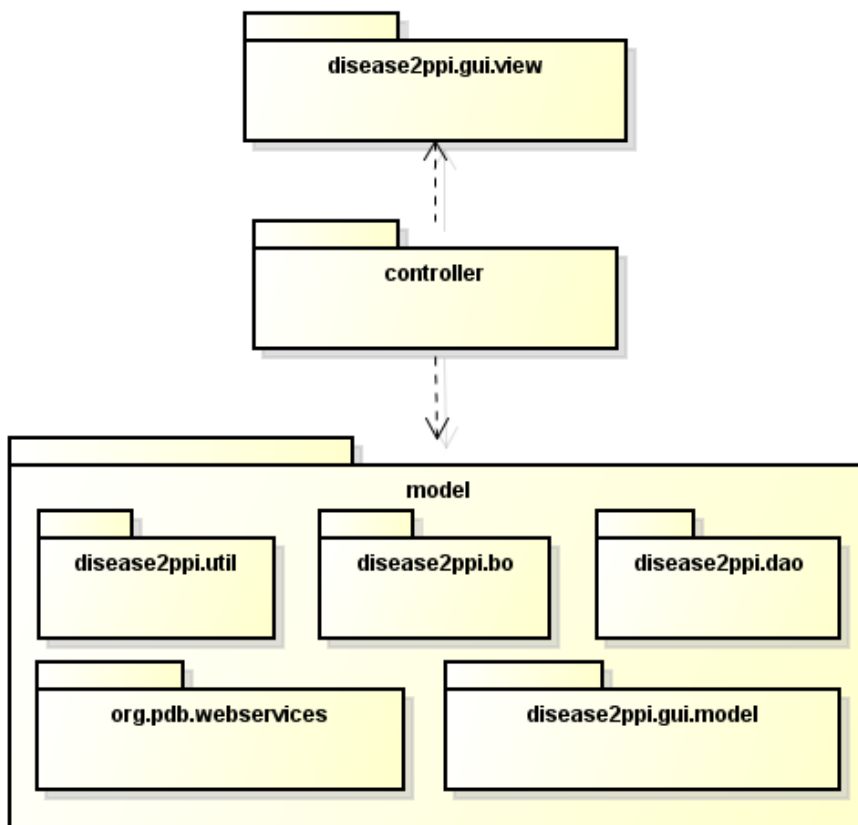
- Migrar a aplicação da versão 1.6 do Java para a versão mais recente.
- Realizar a troca da interface para a plataforma *web*, de acordo com a melhor técnica identificada.
- Realizar tratamento para arquivos salvos utilizados durante o processo de análise da aplicação *desktop*.
- Remover a tela de configurações da ferramenta, pois não será mais necessária.
- Realizar tratamento diferenciado para as mensagens de *LOG* geradas pela aplicação *desktop* em forma de arquivo, que mostra informações de erros de programa, problemas de rede, erros de proteínas não existentes e registro de *links* acessados.
- Providenciar hospedagem da aplicação *web*.

Para a realização da migração da aplicação da versão 1.6 do Java para a versão mais recente, será utilizada a técnica de tradução de código-fonte utilizada pela reengenharia de *software* citada no Capítulo 2. A IDE (*Integrated Development Environment*) NetBeans⁸, utilizada para concepção do projeto do Dis2PPI *desktop* permite essa migração, resolvendo possíveis conflitos gerados pela mudança. Após a migração da versão do Java e a resolução de conflitos, será iniciada a criação do projeto a ser utilizado para construção da aplicação na plataforma *web*.

⁸ <https://netbeans.org/>

Para realização da troca da plataforma *desktop* para a plataforma *web* será necessário uma mudança na camada de visão do *software*. A aplicação foi desenvolvida seguindo o padrão de arquitetura de *software* MVC e em camadas (Figura 19), por isso as alterações necessárias para a aplicação ser transferida para a plataforma *web* serão centralizadas na camada visão.

Figura 19. Arquitetura de *Software* da Ferramenta Dis2PPI



Fonte: Autor

Será utilizado o servidor de aplicação *GlassFish*⁹ versão 4.0, um servidor de aplicação desenvolvido pela *Sun Microsystems* para a plataforma *Java Enterprise Edition*. Ele possui uma versão em código aberto, licenciado também pela *GPL*, que oferece desempenho, confiabilidade, produtividade e facilidade de uso superior a um custo bem menor em relação aos servidores de aplicações proprietários.

Além do servidor de aplicação, serão utilizados os *frameworks* *Spring Web MVC*¹⁰ e *JavaServer Faces* (JSF) que utilizará componentes do *PrimeFaces*¹¹. O *Spring Web MVC*

⁹ <https://glassfish.java.net/>

¹⁰ <http://projects.spring.io/spring-framework/>

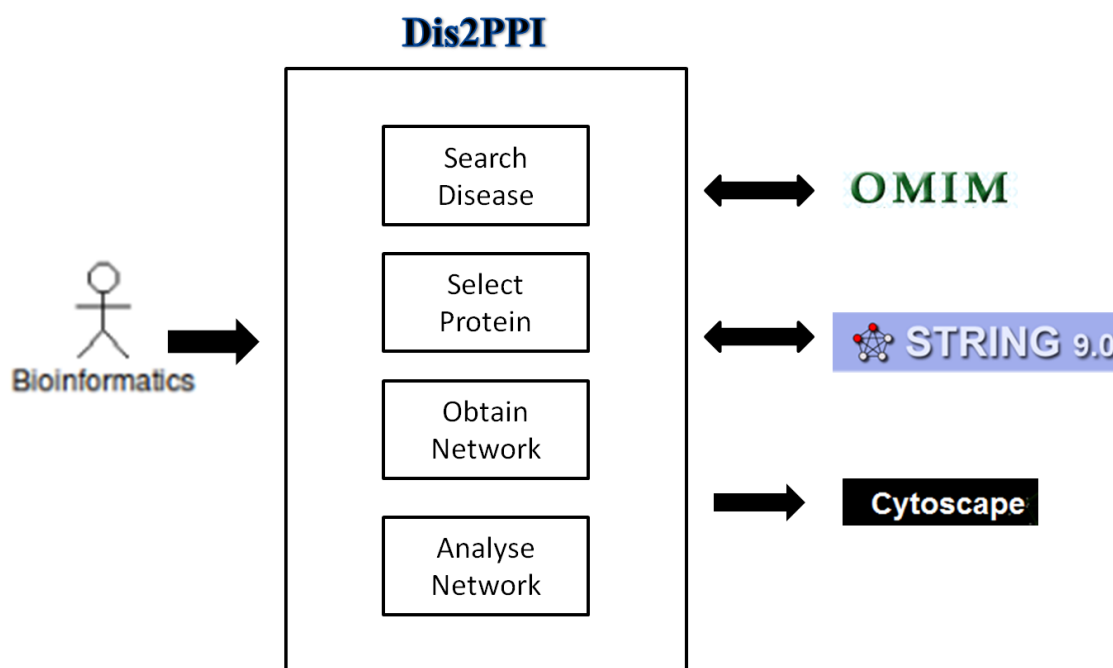
¹¹ <http://primefaces.org/>

Framework é um *framework* robusto, flexível e bem projetado para o desenvolvimento rápido de aplicações *web* usando o padrão de projeto MVC.

O JSF é um *framework* desenvolvido na tecnologia Java que utiliza os padrões de projetos atuais, possui uma poderosa arquitetura para o desenvolvimento de aplicações de grande porte com qualidade e fornece recursos e componentes *web* tornando assim possível a criação de produtos interativos. O JSF oferece algumas vantagens no desenvolvimento de aplicações *web*, pois permite a criação de interfaces de usuário através de um conjunto de componentes pré-definidos; fornece um conjunto de *tags* para acessar os componentes; permite o reuso de componentes da página; associa os eventos do lado cliente com os manipuladores dos eventos do lado servidor e fornece separação de funções que envolvem a construção de aplicações *web*.

A partir da criação do projeto, serão importados os pacotes de controle e modelo existentes na aplicação Dis2PPI. A arquitetura de interação entre os *softwares* permanecerá a mesma, seguindo o modelo mostrado na Figura 20.

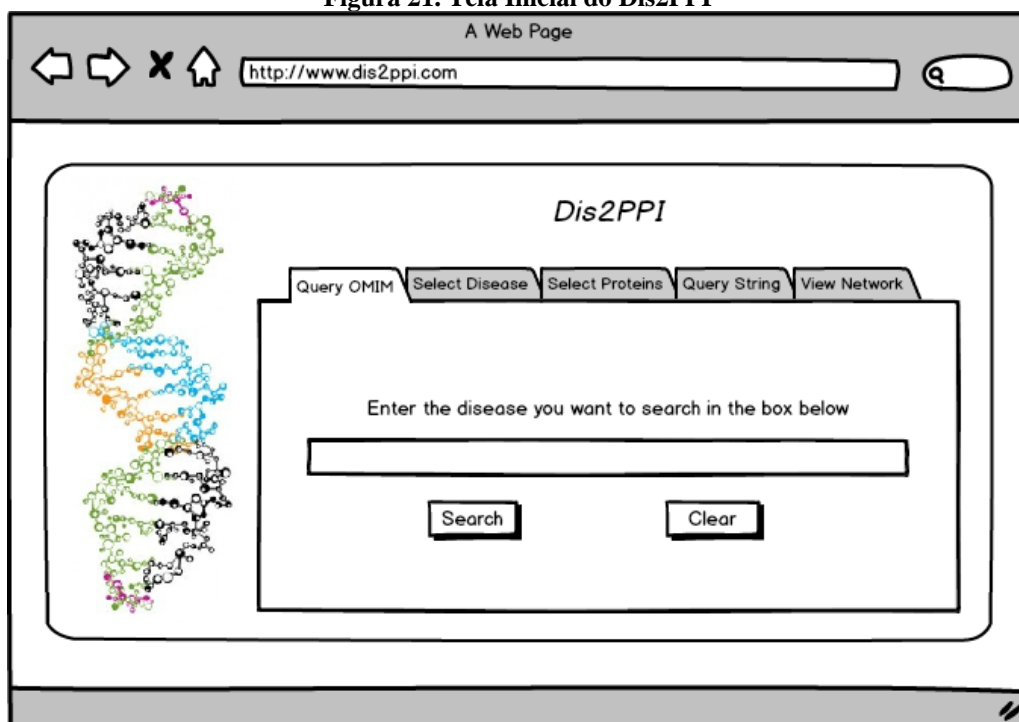
Figura 20. Arquitetura da Ferramenta Dis2PPI entre *Softwares*



Fonte: Disponível em: <http://www.danlian.com.br/dis2ppi/images/dis2ppi_architecture.png>

A partir disso será iniciada a criação do novo pacote de visão, baseado na prototipagem realizada entre a Figura 21 e a Figura 28.

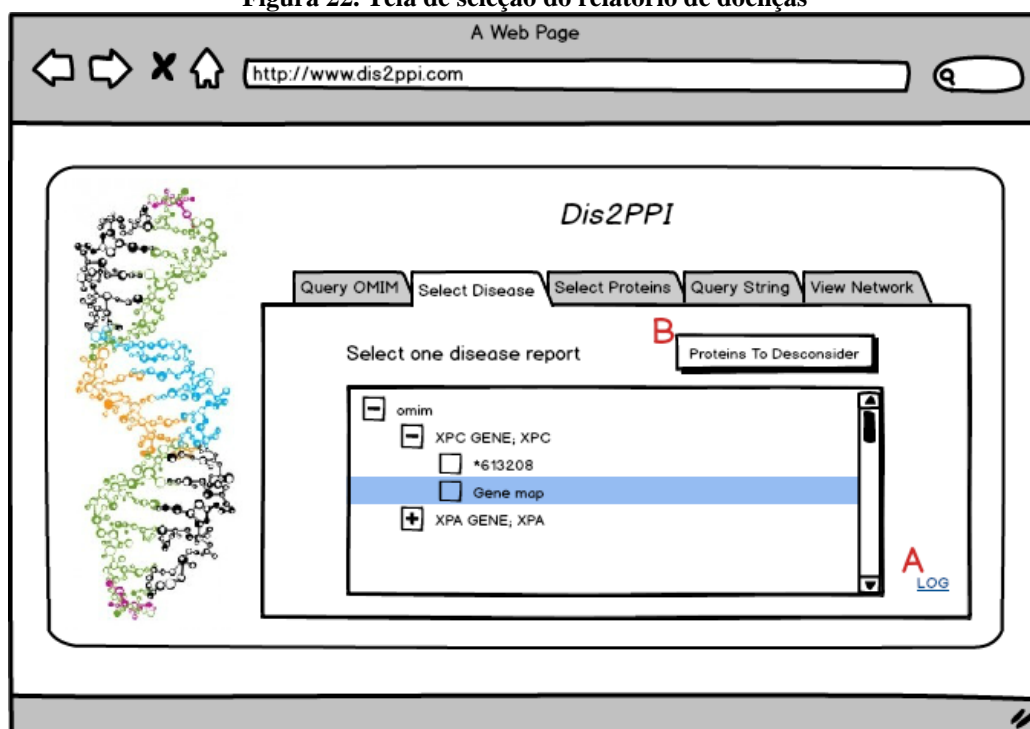
Figura 21. Tela Inicial do Dis2PPI



Fonte: Autor

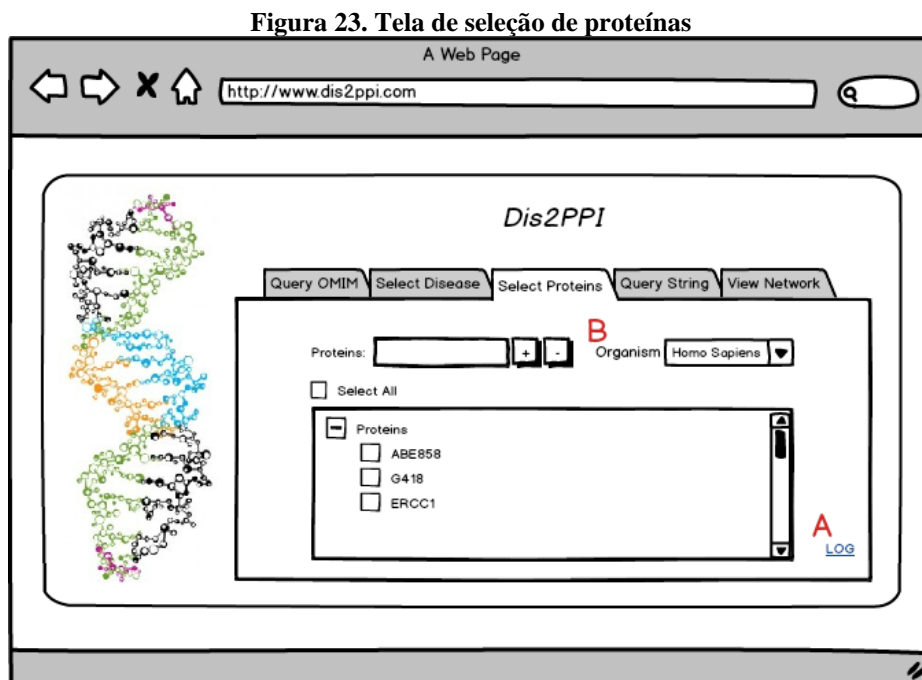
A Figura 21 mostra a tela inicial do Dis2PPI, esta segue o mesmo padrão da tela na plataforma *desktop*, para facilitar a familiarização dos usuários. A Figura 22 mostra a tela de seleção de doenças, que também segue o modelo da plataforma *web*.

Figura 22. Tela de seleção do relatório de doenças



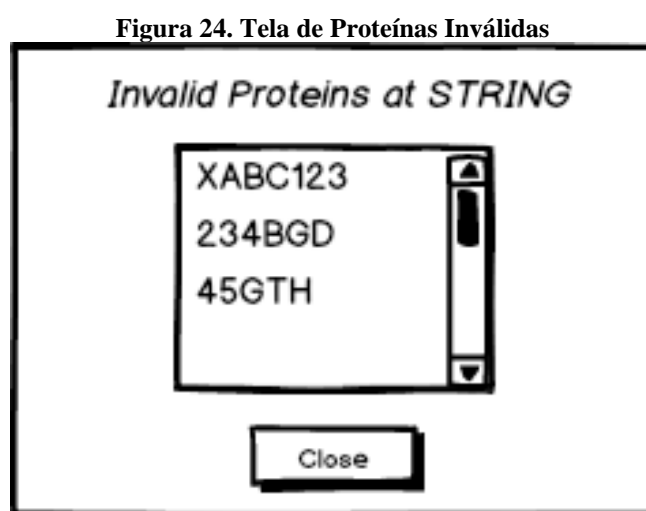
Fonte: Autor

A tela de seleção das proteínas também seguirá o modelo da versão *desktop*, conforme visualizado na Figura 23. Na Figura 23B será carregado uma lista de organismos previamente definidos, conforme é realizado atualmente pela versão na plataforma *desktop*.



Fonte: Autor

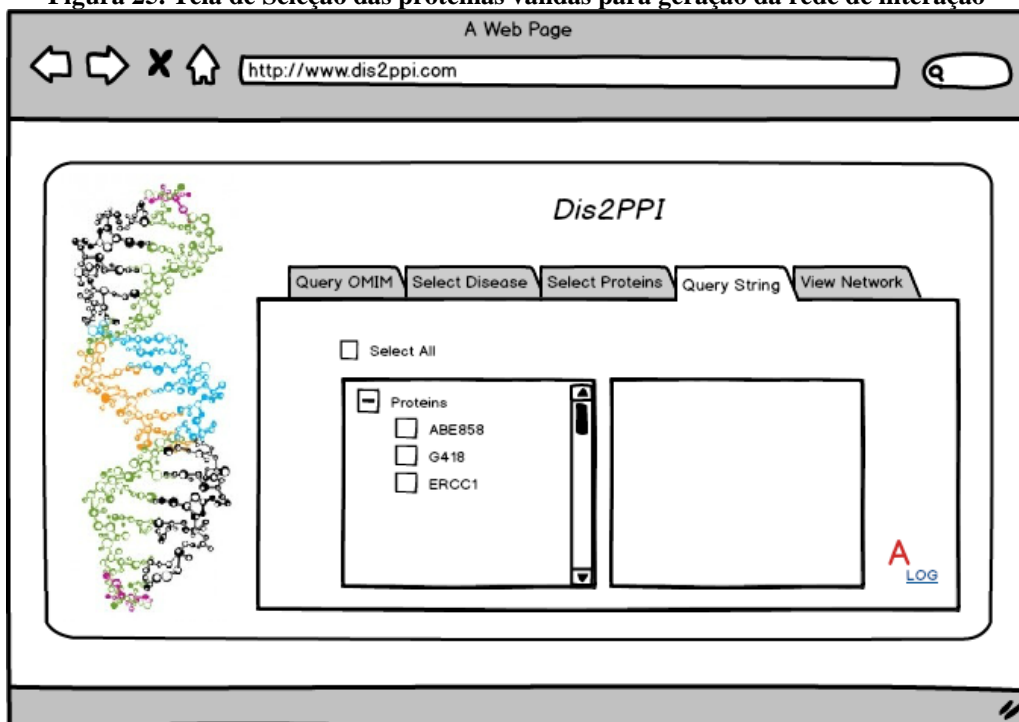
Ao selecionar as proteínas e clicar na aba “*Query String*”, serão realizadas as validações das proteínas na base STRING, e caso algumas não existam na base, as mesmas serão mostradas em uma lista, conforme Figura 24.



Fonte: Autor

Após a visualização das proteínas inválidas, será aberta a tela de seleção das proteínas para geração da rede de interação, conforme Figura 25.

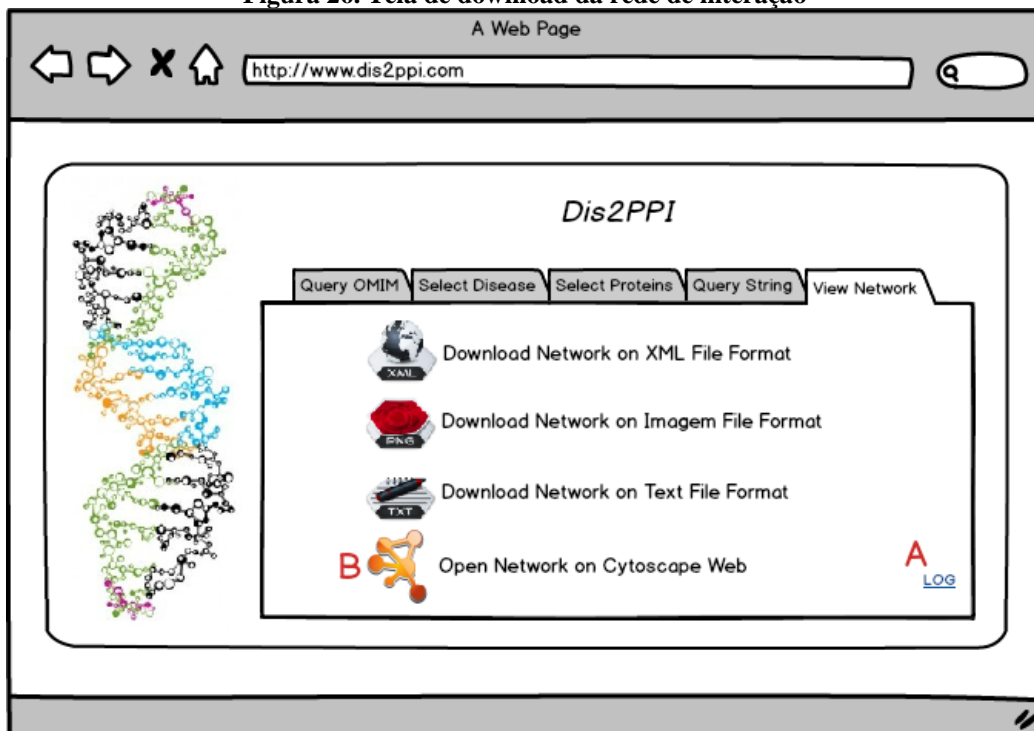
Figura 25. Tela de Seleção das proteínas válidas para geração da rede de interação



Fonte: Autor

26. Ao selecionar as proteínas, será aberta a tela de visualização da rede, conforme Figura

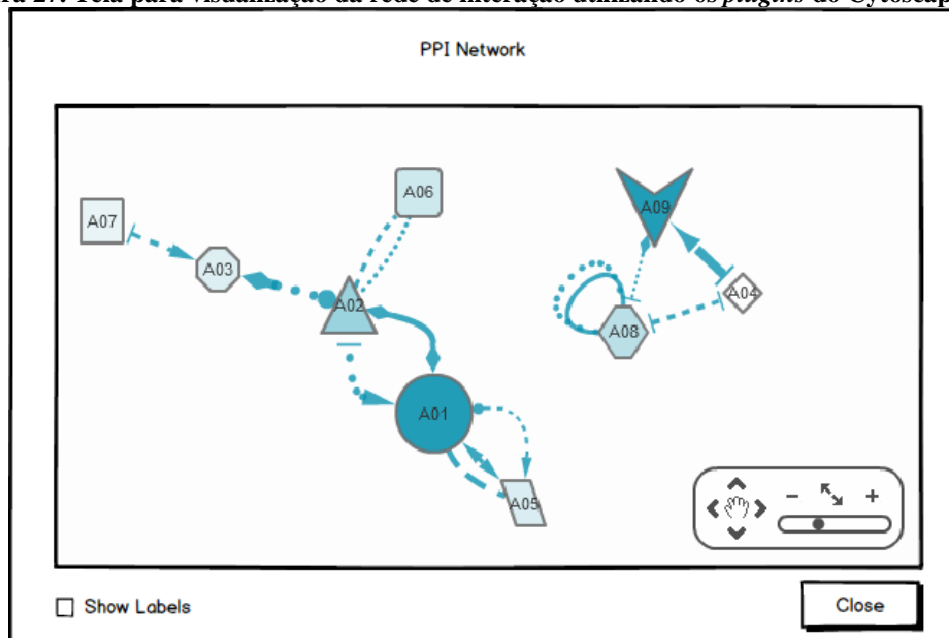
Figura 26. Tela de download da rede de interação



Fonte: Autor

Ao clicar no ícone de abertura do Cytoscape Web (B), será aberta uma nova tela de *pop-up*, com a implementação do *plugin* da ferramenta, conforme Figura 27. Nesta tela será possível visualizar a rede importada, retirando ou não os *labels* de cada proteína.

Figura 27. Tela para visualização da rede de interação utilizando os *plugins* do Cytoscape Web



Fonte: Autor

Nas telas de seleção de doenças (Figura 22A), seleção de proteínas (Figura 23A), validação das proteínas na base STRING (Figura 25A) e *download* da rede de interação (Figura 26A) serão criados *links* para visualização do *LOG* da aplicação. Ao clicar no *link* será aberta uma nova tela *pop-up* conforme Figura 28, mostrando informações de data e hora e informações do *LOG*. Alguns *LOGs* terão detalhes que poderão ser visualizados em uma área de texto abaixo do *LOG*.

Figura 28. Tela de *LOG* da aplicação

Date Time	Log
01/11/2013 15:00:00	Search for XERODERMA
01/11/2013 15:01:00	Connect OMIM
01/11/2013 15:02:00	Select Disease Report
01/11/2013 15:03:00	Add Protein XSYAD

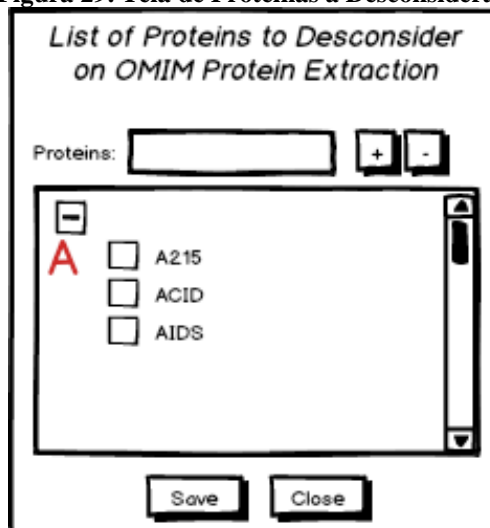
Details

Close

Fonte: Autor

O botão “*Proteins to Disconsider*” (Figura 22B) permite ao usuário a possibilidade de inclusão/exclusão de proteínas a serem desconsideradas durante a extração da base OMIM. Para tal, será aberta uma tela conforme Figura 29.

Figura 29. Tela de Proteínas a Desconsiderar



Fonte: Autor

Na Figura 29A será carregada uma lista prévia de proteínas a desconsiderar utilizada pela aplicação, essa lista poderá ser alterada pelo usuário e salva para utilização em todas as análises realizadas.

Após a implementação das alterações na ferramenta, será realizada a hospedagem da aplicação na web. Para tal, será utilizado o domínio “www.danlian.com.br/dis2ppiWeb/”, utilizando o mesmo domínio onde a ferramenta *desktop* está disponível.

As alterações a serem realizadas para possibilitar o *download* dos arquivos salvos pela aplicação serão de impacto médio baixo, pois será necessária a adaptação do projeto para a plataforma *web*, os formatos de arquivo continuarão sendo os mesmos. Porém neste caso, será necessária uma alteração quanto ao arquivo XML a ser utilizado para importação pelo sistema Cytoscape, pois a versão 3.0 não reconhece mais a versão gerada pela aplicação atual. Para facilitar essa alteração, serão utilizados *plugins* para acesso ao Cytoscape Web¹².

A remoção da tela de configurações (Figura 9) é possível, pois a mesma proporciona informações sobre caminho do navegador, caminho do Cytoscape e informações de *proxy*. Como a ferramenta será transferida para a plataforma *web*, o navegador utilizado será o mesmo onde a ferramenta está aberta, as configurações de *proxy* já serão realizadas no

¹² Exemplos disponíveis em <http://cytoscapeweb.cytoscape.org/demos>

navegador caso exista necessidade e o Cytoscape será alterado para os *plugins* do Cytoscape Web.

Para amostragem dos resultados será criado um caso de testes a ser aplicado na ferramenta Dis2PPI tanto na plataforma *desktop* quanto na plataforma *web*. Esse caso de teste terá o objetivo de mostrar que as funcionalidades da ferramenta permaneceram intactas, e que apenas a camada visual da aplicação foi alterada. Além do caso de teste a ser aplicado, serão mostrados os resultados obtidos pela facilidade na utilização da ferramenta na plataforma *web*, baseados nos objetivos específicos deste trabalho.

Baseado nas solicitações de mudança citadas para transferência de plataforma e melhorias nas funcionalidades, será realizada a análise de impacto e o planejamento da *release*. A implementação das mudanças e liberação do sistema será realizada apenas na segunda parte deste projeto.

3.2 ANÁLISE DE IMPACTO

A análise de impacto será realizada conforme os critérios especificados na Tabela 2 que explicará cada uma das classificações aplicadas as solicitações de mudanças. As categorias iniciam com impacto baixo, representando a alteração mais simples no código-fonte e vai até impacto alto que representa a alteração mais complexa na aplicação.

Tabela 2. Tabela de categorias de impacto

Categoria	Definição
Baixo	Alterações de código simples, com impacto baixo no código-fonte da aplicação.
Médio-Baixo	Alterações de código simples, com impacto baixo em vários pontos no código fonte com o mesmo propósito.
Médio	Alterações de códigos de complexidade média, que tem impacto médio em um ponto específico ou em vários pontos do código-fonte.
Médio-Alto	Alterações de códigos complexas, que tem impacto alto em um ponto específico do código-fonte.
Alto	Alterações de códigos complexas, que tem impacto alto em vários pontos do código-fonte.

Fonte: Criada pelo autor baseado nas definições de Sommerville (2011) e Pfleeger (2004)

A análise de impacto será realizada conforme as solicitações de mudanças já citadas anteriormente. As mesmas serão classificadas dentre as categorias descritas. Devido à migração da versão do Java ser realizada através da IDE do projeto, a alteração terá um impacto baixo no projeto.

A alteração da interface para a plataforma *web* tem impacto médio baixo, devido ao volume de alterações a serem feitas. A remoção da tela de configuração tem impacto baixo, pois a transferência para a plataforma *web* dispensa a necessidade da mesma. As alterações a serem realizadas para possibilitar o *download* dos arquivos salvos pela aplicação serão de impacto médio baixo.

As alterações para tratamento de *LOG* terão impacto médio, pois serão necessárias alterações nas outras camadas da aplicação para atender essa necessidade. Os *LOGs* serão separados e mostrados de acordo com a necessidade. A hospedagem terá impacto baixo, pois hoje a ferramenta já está disponível para *download* na *web*, devido a isso a estrutura para possibilitar a liberação da aplicação na *web* já existe. A Tabela 3 mostra o resumo das atividades com seus respectivos impactos para alteração da aplicação.

Tabela 3. Resumo da análise de impacto

Atividades	Impacto
Migração da versão do Java	Baixo
Alteração da interface para plataforma <i>web</i>	Médio-Baixo
Remoção da tela de configuração	Baixo
<i>Download</i> de arquivos	Médio-Baixo
Alterações para tratamento de <i>LOG</i>	Médio
Hospedagem	Baixo

Fonte: Autor

Baseado nessas informações será iniciado o planejamento da *release* durante próxima seção, detalhando as alterações necessárias e prevendo alcançar os objetivos anteriormente citados.

3.3 PLANEJAMENTO DA *RELEASE*

Para realizar o planejamento da *release*, será realizada a análise dos objetivos para identificar mais detalhadamente as mudanças a serem realizadas. Conforme levantado na análise de impacto, a migração da versão do Java será realizada através da IDE NetBeans, resolvendo os conflitos pela própria ferramenta.

Baseado na análise foi realizado o planejamento da *release* com as seguintes atividades:

- Migração da aplicação para a versão 1.8 do Java utilizando a IDE NetBeans;
- Implementação da mudança da interface para a plataforma *web* seguindo a prototipagem mostrada anteriormente;
- Implementação da possibilidade de *download* dos arquivos gerados pela aplicação;
- Implementação da nova versão do *LOG*, seguindo a prototipagem mostrada anteriormente;
- Implementação dos *plugins* do Cytoscape Web, devido à identificação das mudanças para leitura dos arquivos XML gerados pela ferramenta;
- Realizar testes de funcionalidade, comparando a facilidade de utilização entre a aplicação *desktop* e a aplicação *web*.

A partir do planejamento finalizado, iniciou-se o processo de implementação de mudança para criação do Dis2PPIWeb.

4 IMPLEMENTAÇÃO

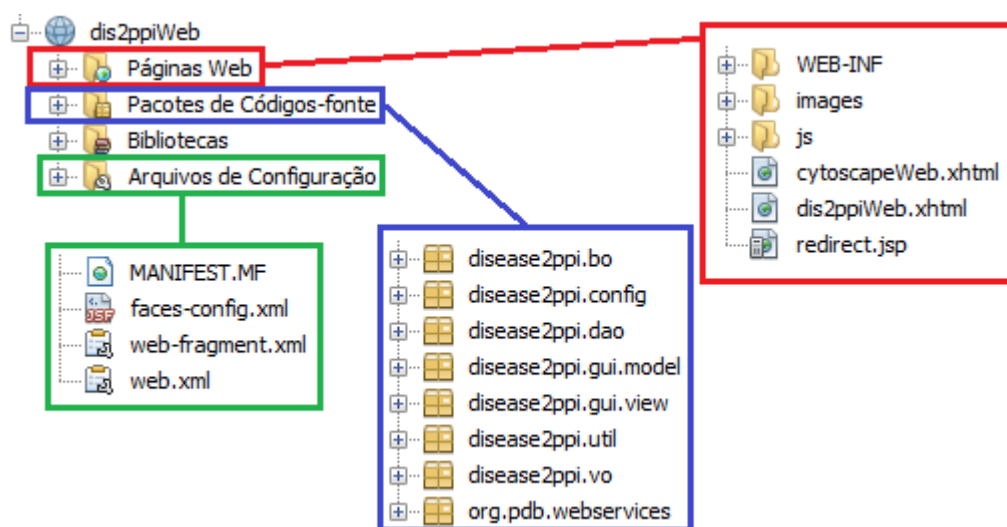
Prosseguindo com a metodologia de evolução de software de Sommerville (2011) detalhada na sessão 2.4 e proposta para realização deste trabalho, iniciou-se o processo de implementação da mudança e liberação do sistema.

4.1 IMPLEMENTAÇÃO DE MUDANÇA

A implementação de mudança foi iniciada primeiramente com a adaptação da plataforma, primeiramente migrando a versão do Java para 1.8 e em seguida alterando as classes Java necessárias para que a aplicação pudesse ser utilizada na plataforma *web*. Após a adaptação, iniciou-se o aperfeiçoamento do sistema, criando as novas funcionalidades, e realizando as alterações necessárias para que a ferramenta não tenha mais as restrições existentes na versão *desktop*. Após o aperfeiçoamento, iniciaram-se os testes e com base nos mesmos, foram realizadas as correções necessárias.

A adaptação de plataforma iniciou-se com a criação de um projeto *web* na ferramenta NetBeans. Neste novo projeto, foram importadas as *packages* já existentes da ferramenta Dis2PPI, exceto a *package view*. Foi criada uma nova *package view* para a inclusão das classes Java que fazem a ligação entre o *html* e as outras camadas do projeto.

Figura 30. Estrutura do projeto Dis2PPIWeb



Fonte: Autor

A Figura 30 mostra a nova estrutura do projeto Dis2PPI, onde temos as Páginas *Web*, os Pacotes de Códigos Fonte, Bibliotecas e os Arquivos de Configuração. As páginas *web* contêm os arquivos *html* juntamente com os arquivos JSP criados pelo NetBeans. A ligação entre a classe *Java* e o HTML foi realizada diretamente, conforme pode ser visualizado na Figura 31.

Figura 31. Exemplo de Integração entre HTML e Classe Java

```

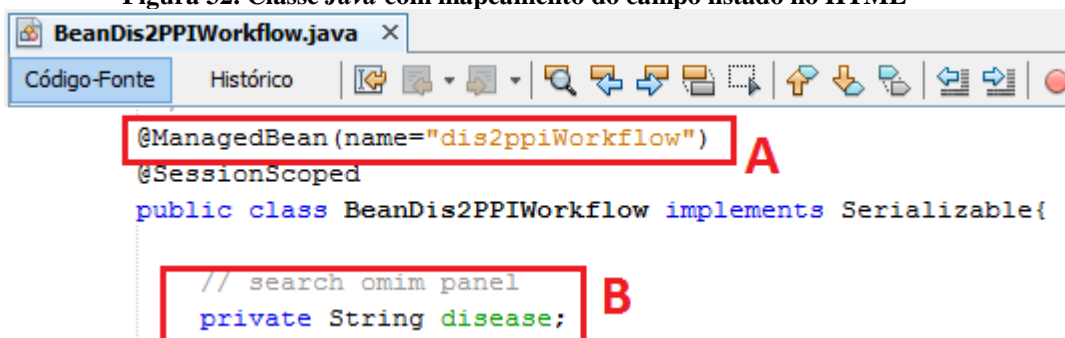
<p:tab title="Query OMIM">
  <p:panel id="QueryOMIM" styleClass="panelDefault" style="text-align: center;">
    <h:form style="margin-top:18px">
      <p:panel style="border: none">
        <h:outputLabel value="Enter the disease you want to search in the box below"/>
      </p:panel>
      <p:panel style="border: none">
        <p:inputText id="inputDisease" value="#{dis2ppiWorkflow.disease}" required="true"
        </p:panel>
        <p:commandButton value="Search" actionListener="#{dis2ppiWorkflow.searchDiseases()}"
        <p:commandButton value="Clear" actionListener="#{dis2ppiWorkflow.clearSearch()}" upda
      </h:form>
    </p:panel>
  </p:tab>

```

Fonte: Autor

O valor destacado no HTML “#{dis2ppiWorkflow.disease}”, devido ao mapeamento realizado na Figura 32 (A), representa diretamente o atributo (B) na classe *Java*. A transferência de valores entre o HTML e a classe *Java* ocorrem através dos métodos *GETTER* e *SETTER* da classe *Java* interpretados pelo servidor de aplicação.

Figura 32. Classe Java com mapeamento do campo listado no HTML



```

BeanDis2PPIWorkflow.java
Código-Fonte  Histórico
@ManagedBean(name="dis2ppiWorkflow")
@SessionScoped
public class BeanDis2PPIWorkflow implements Serializable{

    // search omim panel
    private String disease;

```

Fonte: Autor

Após a criação da nova interface, iniciou-se a integração entre as camadas do projeto. Para tal, foram necessárias mudanças nos outros pacotes de códigos-fonte para integrar as camadas. O pacote *disease2ppi.gui.model*, por trabalhar com componentes do *Swing* (pacote para criação de interface na versão *desktop*), foi adaptado para utilização dos novos componentes da biblioteca Prime Faces.

Ao finalizar a criação da nova interface, iniciaram-se as alterações para aperfeiçoamento do sistema. Inicialmente, foi realizada a alteração para desconsiderar a configuração de *proxy* de rede utilizada na aplicação *desktop*. Como o consumo dos *webservices* para busca dos dados nas bases OMIM e STRING é realizado pelo servidor, as configurações utilizadas serão as do servidor. Com isso, não haverá necessidade do usuário realizar nenhuma configuração para o funcionamento da aplicação.

Após a finalização das alterações referente *proxy* de rede, iniciou-se a alteração referente as proteínas a desconsiderar. Em um projeto *web* não é possível a utilização de arquivos locais temporários para carregamento das proteínas a desconsiderar. Devido a isso, foi realizada uma alteração para possibilitar o *upload* e *download* da lista de proteínas a desconsiderar. Isso facilitará ao usuário o carregamento da lista de proteínas conforme a sua necessidade. Esses dados não foram salvos em banco de dados, pois como a ferramenta não possui um controle de *login*, não seria possível identificar qual usuário está efetivamente conectado a aplicação para identificar quais informações foram salvas.

Também foi criado a visualização das interações pelo Cytoscape Web. Para tal, foi necessário a utilização de uma biblioteca específica criada para receber um arquivo XML e/ou JSON¹³ (*JavaScript Object Notation*) com a estrutura a ser visualizada. Como o arquivo tem uma estrutura mais simples do que o XML criado pela base de dados STRING, foi criado um arquivo JSON baseado nas informações do XML. Por ora, apenas as informações referente as proteínas e suas interações foram interpretadas para montagem da rede de interação. É possível vincular outras informações, tanto para o nodo de proteína, quanto para a ligação entre eles. Como o objetivo era visualizar os dados existentes, essas alterações não foram implementadas.

A Figura 33 mostra um exemplo de JSON criado pelo Dis2PPIWeb e utilizado no Cytoscape Web. Existem campos obrigatórios para criação dos nodos (*id*) e arestas (*id*, *source* e *target*) a serem utilizados na estrutura do JSON para que o mesmo funcione corretamente na aplicação, outros podem ser criados com informações adicionais, montando um *dataSchema* junto ao corpo do JSON, para serem utilizadas durante a aplicação de forma programável.

¹³ <http://www.json.org/>

Figura 33. JSON com estrutura utilizada para visualização da rede de interação

```
var network = {  
  dataSchema: {  
    nodes: [  
      {  
        name: "id",    type: "string"  
      },  
      {  
        name: "label", type: "string"  
      }  
    ]  
  },  
  data: {  
    nodes: [  
      {  
        id: "PUF60", label: "PUF60"  
      },  
      {  
        id: "FUBP3", label: "FUBP3"  
      }  
    ],  
    edges: [  
      {  
        id: "PUF60toFUBP3", source:"PUF60", target:"FUBP3"  
      },  
      {  
        id: "FUBP3toPUF60", source:"FUBP3", target:"PUF60"  
      }  
    ]  
  }  
};
```

Fonte: Autor

Por último, foi realizada a mudança para visualização do *LOG*. Primeiro, foram separados e visualizados de forma diferente as exceções referente a não existência de uma proteína na base STRING. Foram criados *LOGs* para acompanhamento do processo pelo usuário, para que o mesmo possa identificar os passos realizados durante a execução da aplicação. Nesse mesmo *LOG* é possível encontrar exceções específicas da aplicação, caso ocorram.

Não haviam correções a serem aplicadas na versão *desktop* da aplicação, com isso as únicas correções a serem aplicadas foram referentes as novas alterações realizadas. Foram encontrados algumas falhas na nova versão encontradas durante os testes que já foram corrigidas antes da liberação do sistema. Mais detalhes sobre os testes estão disponíveis nas considerações finais deste trabalho.

4.2 LIBERAÇÃO DO SISTEMA

Para liberação do sistema foi necessário a instalação do servidor de aplicação *Glassfish*. Para instalação do mesmo, seguiu-se as especificações descritas no manual¹⁴ de instalação.

Para realização da liberação da aplicação, primeiramente é necessário realizar a inicialização do domínio. Este pode ser o criado juntamente com a instalação do *Glassfish* ou um criado com as especificações necessárias pelo usuário. Neste caso, foi utilizado o domínio padrão criado juntamente com a instalação do servidor para disponibilização da aplicação. A realização deste processo foi feita através de uma conexão SSH (*Secure Shell*), seguindo o manual¹⁵ de instalação disponível no site do servidor da aplicação. Os comandos utilizados foram: (1) “`asadmin start-domain domain1`”, comando que inicia o domínio; e (2) “`asadmin deploy [path]/dis2ppiWeb.war`”, comando que realiza o *deploy* da aplicação.

Ao finalizar o *deploy* da aplicação, foi montado um caso de uso a ser aplicado para validação dos resultados esperados, além de vários casos de testes para validação das alterações aplicadas que foram descritos juntamente com as considerações finais deste trabalho. O *deploy* foi realizado localmente para realização dos testes e aplicação do caso de uso, pois devido a restrições no servidor de hospedagem onde a aplicação *desktop* estava disponível para *download* não foi possível instalar o servidor de aplicação.

¹⁴ Manual disponível em <https://glassfish.java.net/docs/4.0/installation-guide.pdf>, acessado em 06/2014.

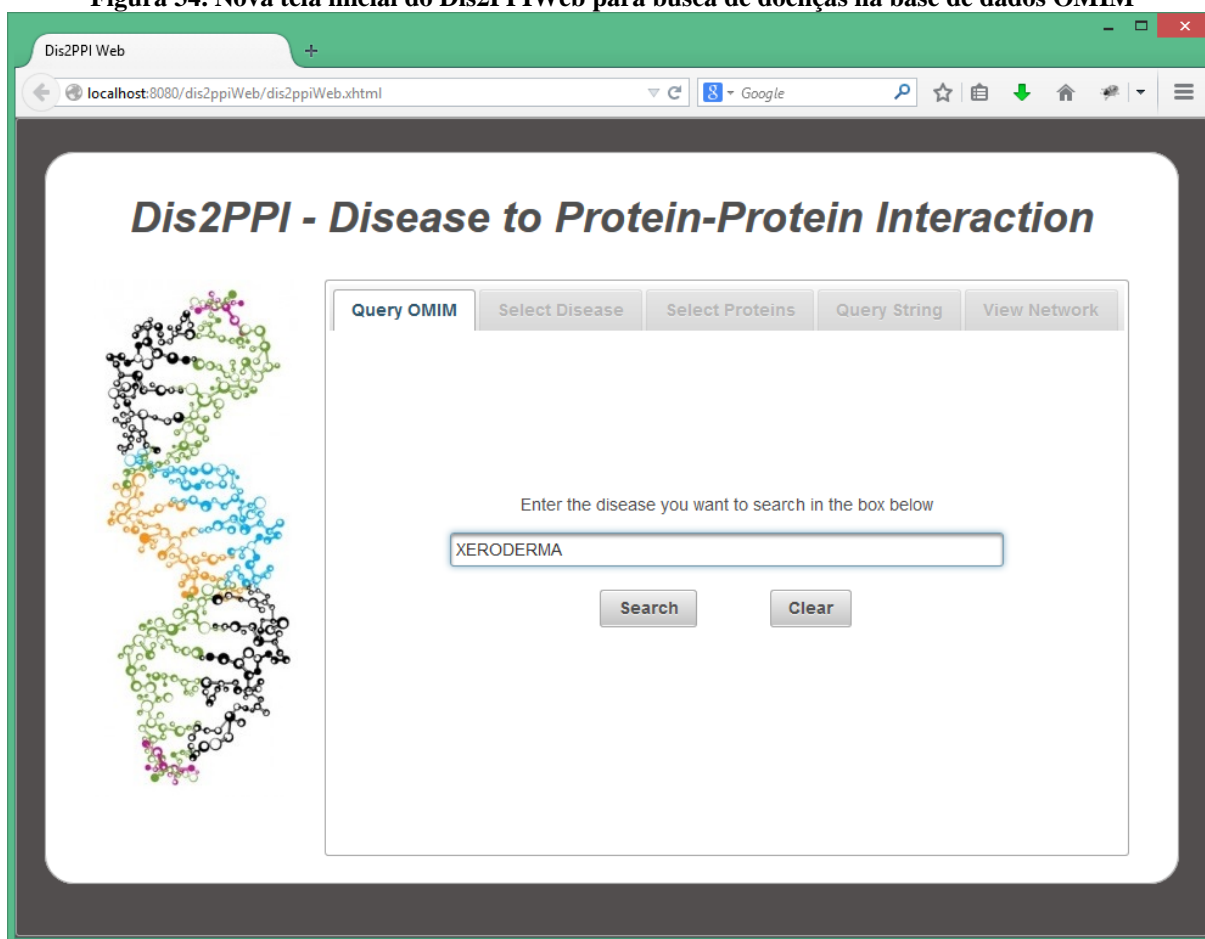
¹⁵ Manual disponível em <https://glassfish.java.net/docs/4.0/application-deployment-guide.pdf>, acessado em 06/2014.

5 ESTUDO DE CASO

Após a finalização das alterações, foi iniciado um estudo de caso seguindo as mesmas informações para demonstração do funcionamento da ferramenta na versão *desktop*, realizada na sessão 2.2 deste documento.

A tela inicial da nova versão Dis2PPIWeb¹⁶ (Figura 34). Nela foi realizada a pesquisa por “XERODERMA”.

Figura 34. Nova tela inicial do Dis2PPIWeb para busca de doenças na base de dados OMIM

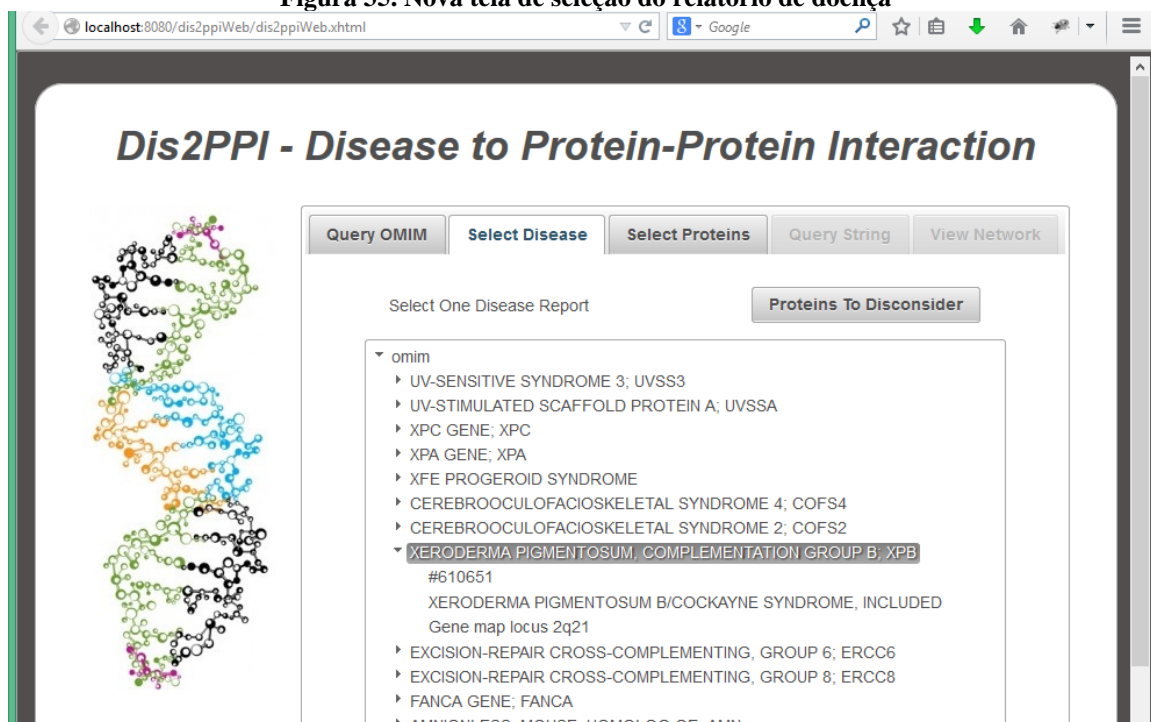


Fonte: Autor

Ao clicar no botão “*Search*” a pesquisa retornou os mesmos relatórios de doenças carregados pela aplicação *desktop*, conforme pode ser visualizado na Figura 35.

¹⁶ <http://www.danlian.com.br/dis2ppiWeb/>

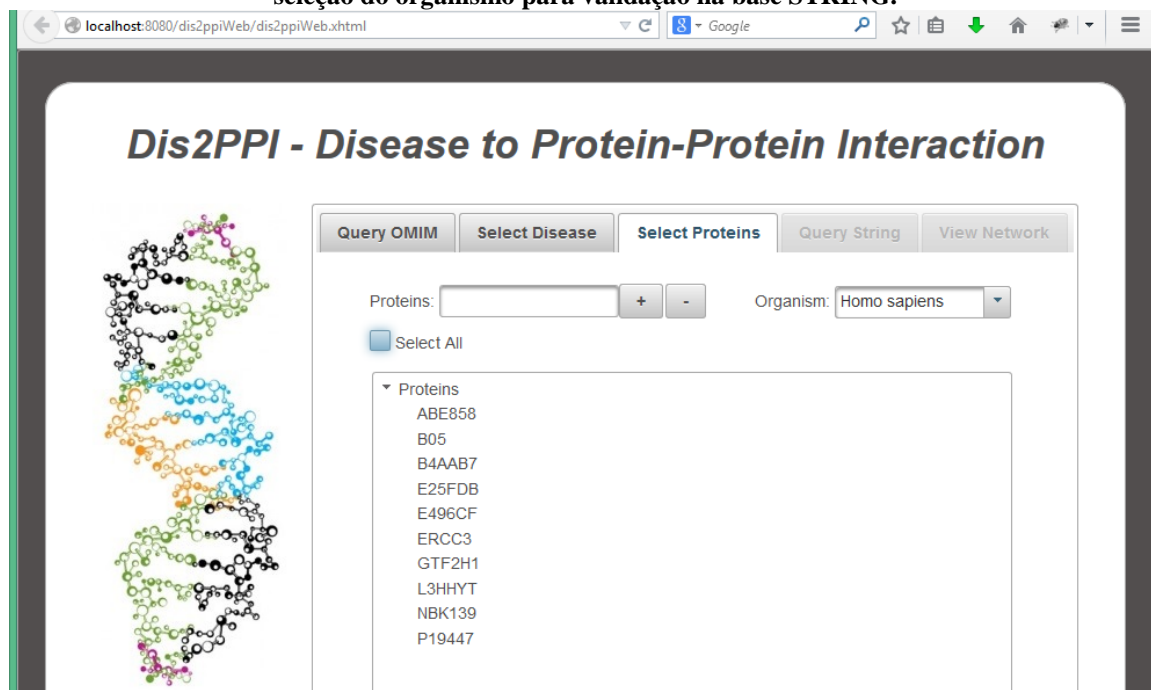
Figura 35. Nova tela de seleção do relatório de doença



Fonte: Autor

Ao selecionar o relatório de doenças, são carregadas as proteínas extraídas do relatório de doenças conforme mostrado na Figura 36.

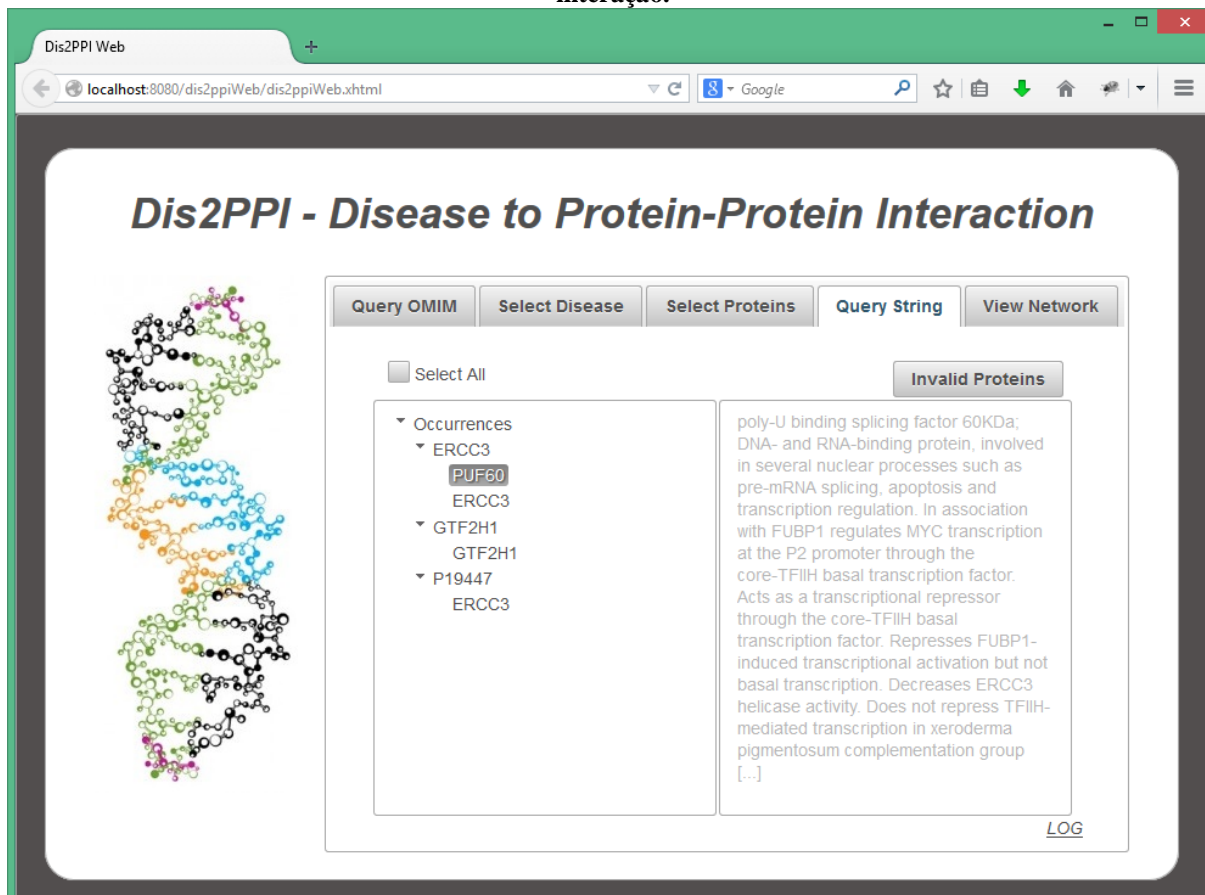
Figura 36. Nova tela de seleção de proteínas, com possibilidade de inclusão de novas proteínas, além da seleção do organismo para validação na base STRING.



Fonte: Autor

Após a seleção de proteínas e validação na base de dados STRING, é possível selecionar as proteínas para busca das redes de interações, conforme mostrado na Figura 37.

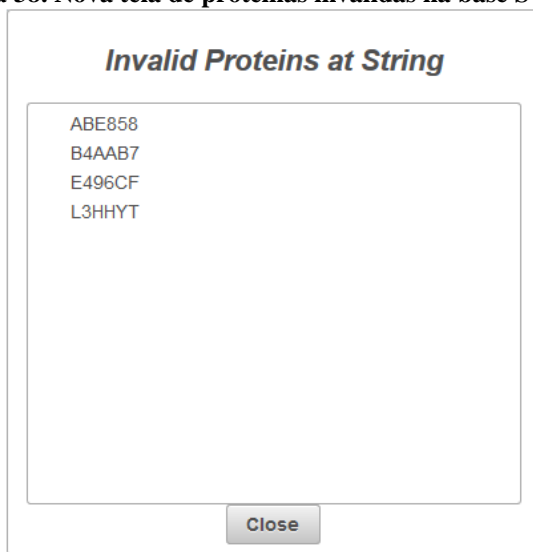
Figura 37. Nova tela de ocorrências de proteínas validadas na base STRING para obtenção da rede de interação.



Fonte: Autor

A tela de proteínas inválidas será aberta ao clicar no botão “Invalid Proteins” existente nesta tela (Figura 38).

Figura 38. Nova tela de proteínas inválidas na base STRING

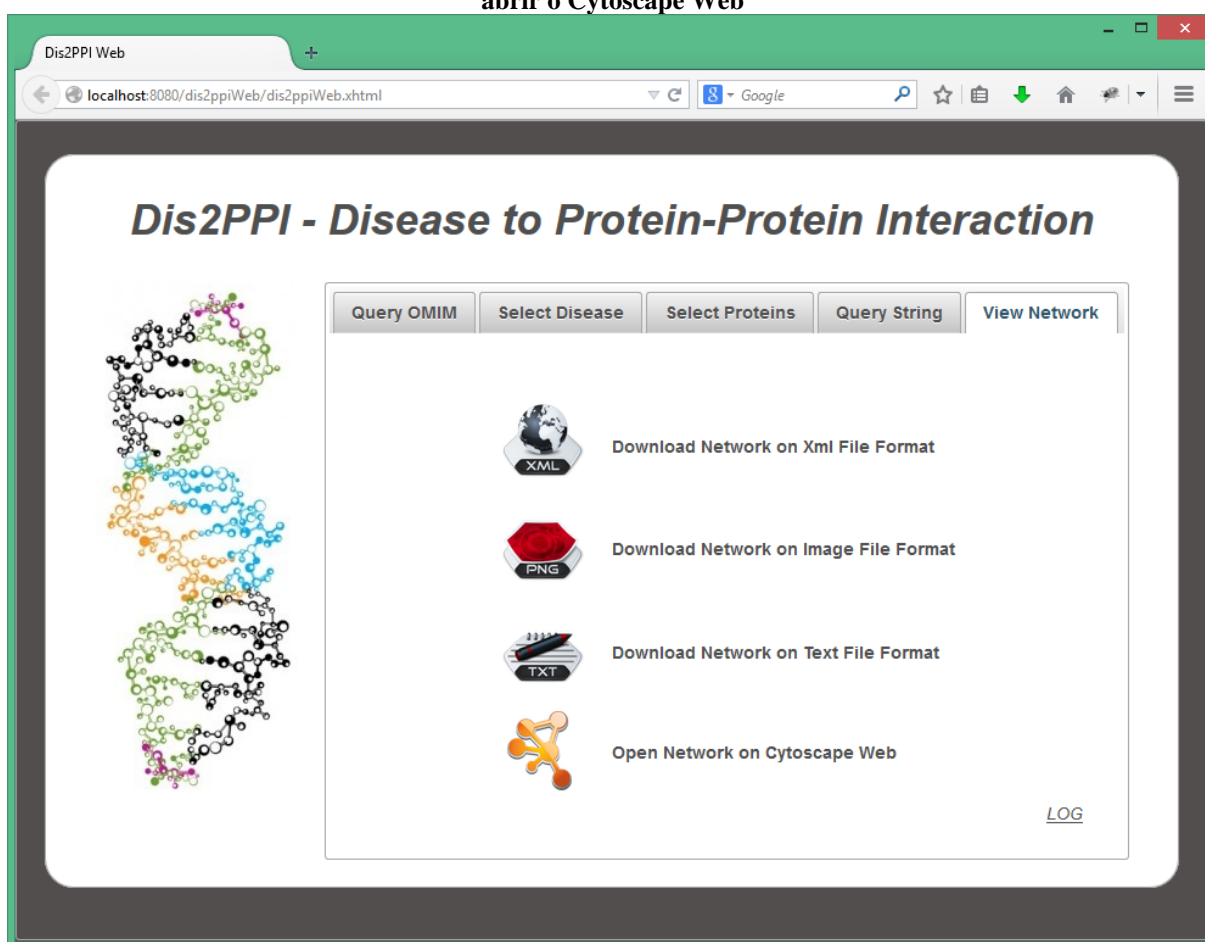


Fonte: Autor

A tela não foi aberta automaticamente como previsto inicialmente, pois não é obrigatória a visualização dessa informação pelo usuário. Dessa forma, fica a critério do usuário acessar ou não a tela de proteínas inválidas.

Ao selecionar uma ou mais ocorrências validadas pela base STRING, é possível visualizar a rede de interação pela aba “View Network”, conforme mostrado na Figura 39. Esta aba continua possibilitando ao usuário a visualização dos dados da base STRING através de XML, texto e imagem.

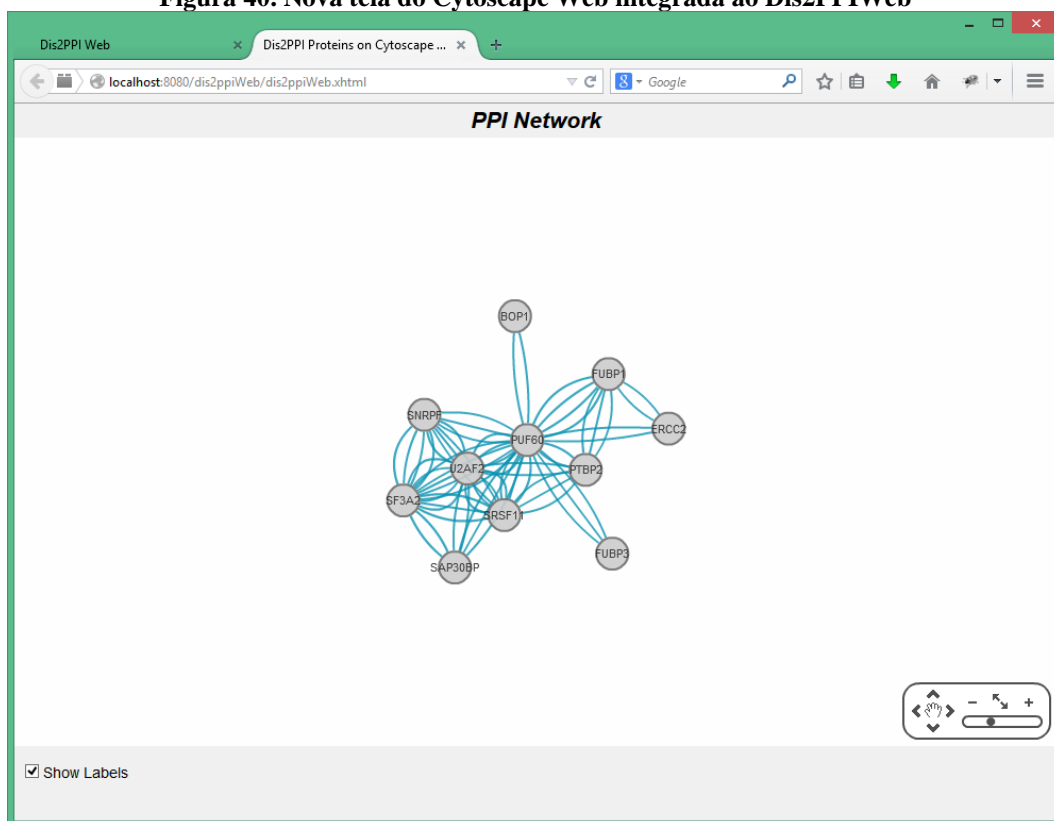
Figura 39. Nova tela de visualização de redes de interação com as opções XML, imagem, texto ou para abrir o Cytoscape Web



Fonte: Autor

Ao clicar no link para abertura do Cytoscape Web disponível na Figura 39, é aberta uma nova janela com o Cytoscape Web carregando as informações das redes de interação da proteína “PUF60”, mostrado na Figura 40.

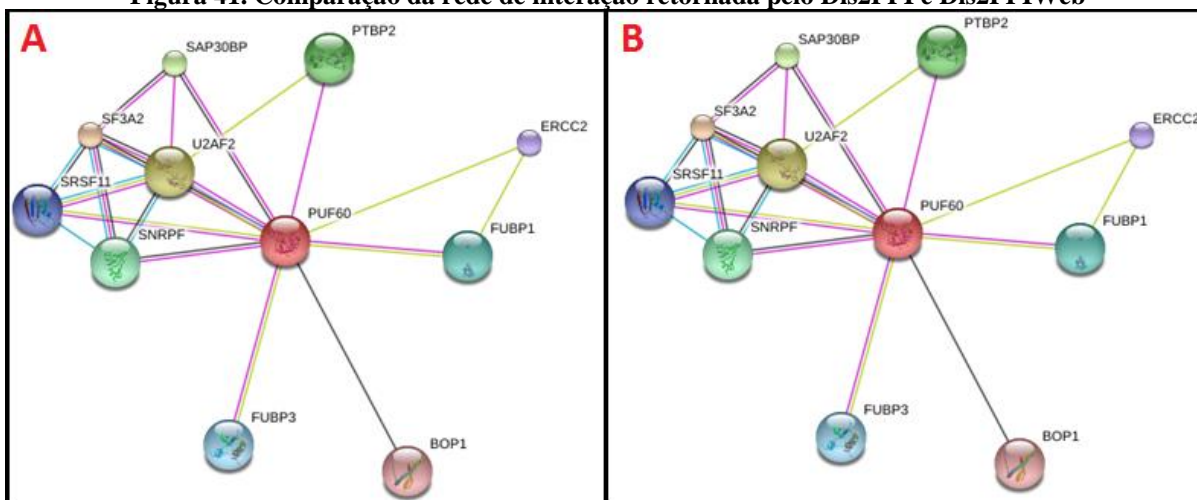
Figura 40. Nova tela do Cytoscape Web integrada ao Dis2PPIWeb



Fonte: Autor

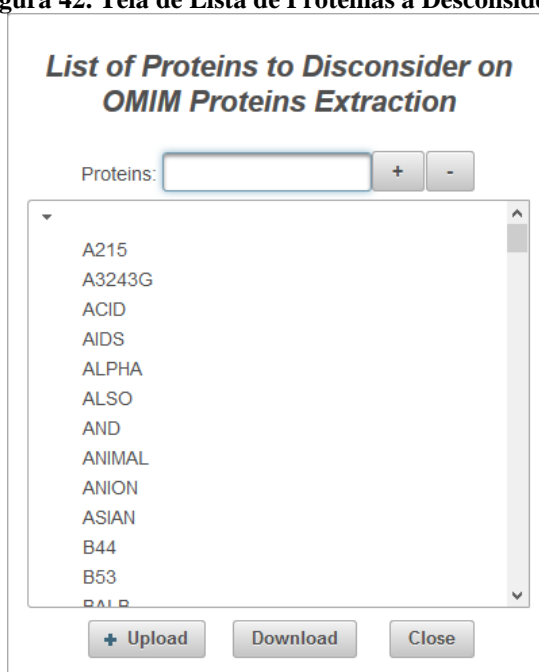
Assim como a aplicação *desktop* Figura 41 (A), a aplicação *web* retornou a mesma rede de interação para a proteína “PUF60”, como mostra a Figura 41 (B).

Figura 41. Comparação da rede de interação retornada pelo Dis2PPI e Dis2PPIWeb



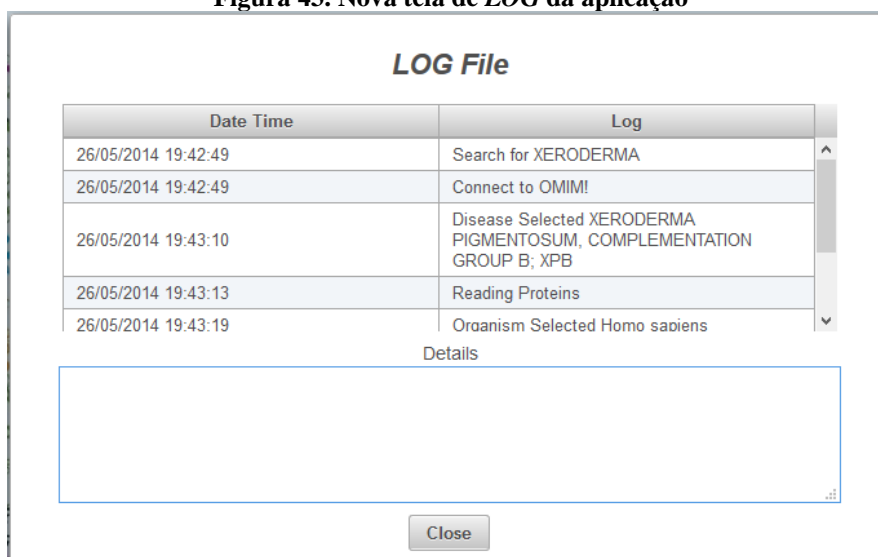
Fonte: Autor

O botão “*Proteins to Disconsider*”, disponível na tela na Figura 35, abre a tela de proteínas a desconsiderar, conforme a Figura 42.

Figura 42. Tela de Lista de Proteínas a Desconsiderar

Fonte: Autor

Nesta tela, foi realizada uma mudança para permitir o *upload* e *download* da lista de proteínas, para facilitar o uso do usuário. Essa alteração é uma das melhorias realizadas para aperfeiçoamento do sistema em sua versão *web*, visto que não é possível arquivos temporários em aplicações *web*. A nova tela de *LOG* foi implementada conforme a Figura 43.

Figura 43. Nova tela de LOG da aplicação

Fonte: Autor

6 CASOS DE TESTE

Foram aplicados diversos casos de teste a fim de validar as alterações realizadas na ferramenta. Os casos de teste 1 (Tabela 4) e 2 (Tabela 5) descritos abaixo são referentes ao estudo de caso demonstrado no capítulo 5.

Tabela 4. Caso de Teste 1 - Visualizar Arquivos TXT, Imagem, XML

Caso de Teste	01 – Visualizar Arquivos TXT, Imagem, XML
Pré-Condição	Existirem dados válidos na base de dados OMIM e STRING
Passos	<ol style="list-style-type: none"> 1. Informar uma doença a ser pesquisada na base OMIM 2. Clicar em “<i>Search</i>” 3. Selecionar um relatório de doença 4. Clicar na aba “<i>Select Proteins</i>” 5. Selecionar as proteínas extraídas, junto com o organismo HOMO SAPIENS 6. Clicar na aba “<i>Query String</i>” 7. Selecionar as ocorrências válidas 8. Clicar na aba “<i>View Network</i>” 9. Visualizar os arquivos gerados
Resultados Esperados	Arquivos gerados pela base STRING visualizados com sucesso
Comentários	Conforme mostrado no estudo de caso do capítulo 4, os arquivos foram visualizados com sucesso, com os mesmos resultados da aplicação <i>desktop</i> .

Fonte: Autor

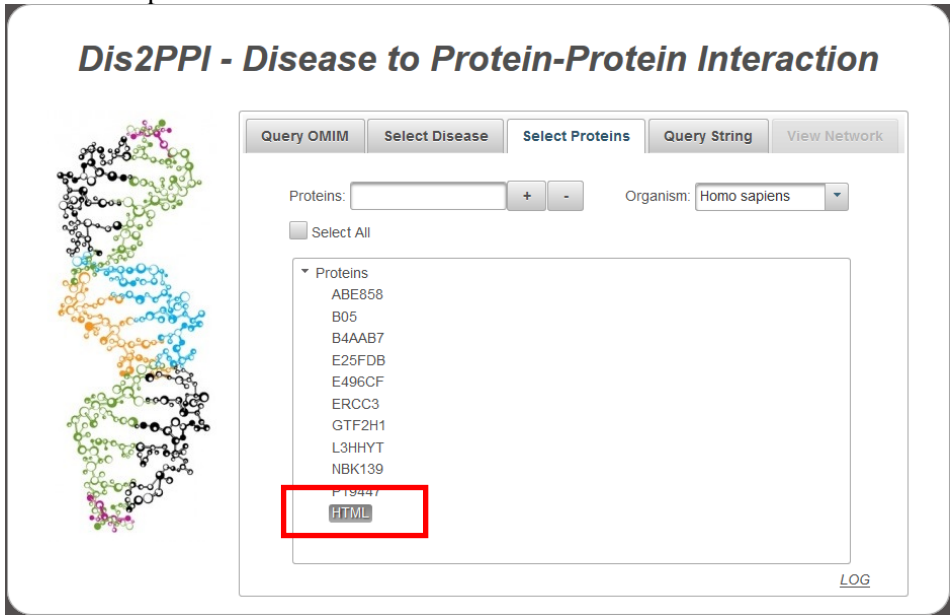
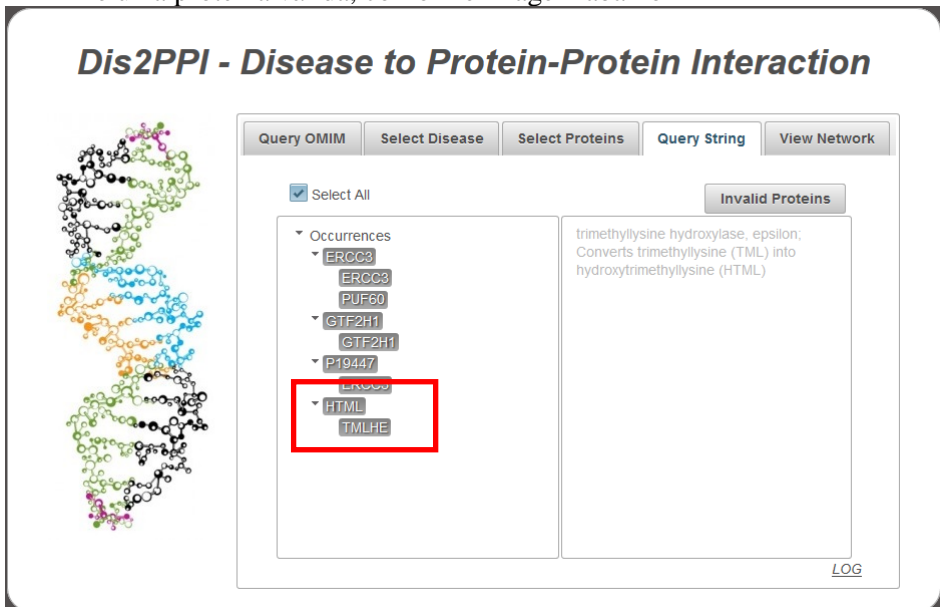
Tabela 5. Caso de Teste 2 - Visualizar Rede de Interação no Cytoscape Web

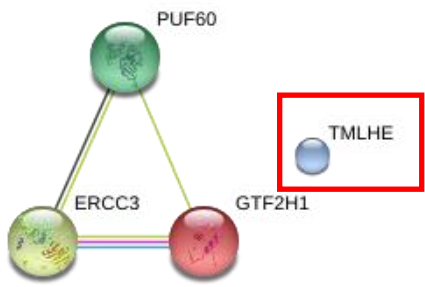
Caso de Teste	02 – Visualizar Rede de Interação no Cytoscape Web
Pré-Condição	Existirem dados válidos na base de dados OMIM e STRING
Passos	<ol style="list-style-type: none"> 1. Informar uma doença a ser pesquisada na base OMIM 2. Clicar em “<i>Search</i>” 3. Selecionar um relatório de doença 4. Clicar na aba “<i>Select Proteins</i>” 5. Selecionar as proteínas extraídas, junto com o organismo HOMO SAPIENS 6. Clicar na aba “<i>Query String</i>” 7. Selecionar as ocorrências válidas 8. Clicar na aba “<i>View Network</i>” 9. Clicar no link do Cytoscape Web
Resultados Esperados	Visualizar a rede de interação pelo <i>plugin</i> do Cytoscape Web, validando os dados das interações comparados aos dados da base STRING
Comentários	Conforme mostrado no estudo de caso o Cytoscape Web mostrou as mesmas interações entre proteínas que a imagem gerada pela base de dados STRING

Fonte: Autor

Ao finalizar o fluxo padrão do *workflow*, iniciou-se a validação das demais funcionalidades da ferramenta. O caso de teste 3 (Tabela 6) demonstra a validação de inclusão de proteínas válidas manualmente na aplicação para buscar as ligações dessa proteínas com outras encontradas no relatório de doenças.

Tabela 6. Caso de Teste 3 - Incluir proteínas válidas manualmente

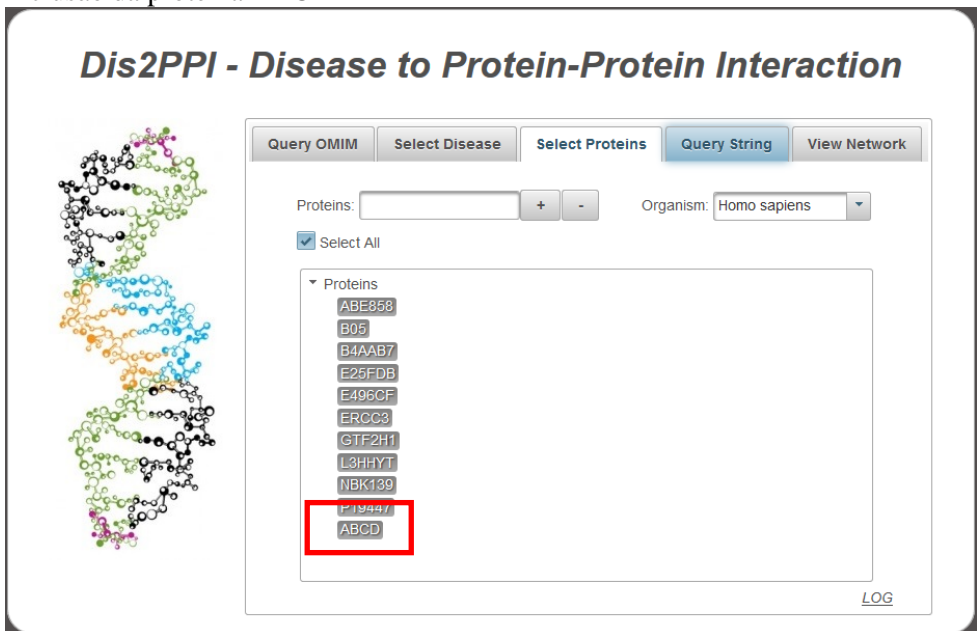
Caso de Teste	03 – Incluir proteínas válidas manualmente
Pré-Condição	Existirem dados válidos na base de dados OMIM e STRING
Passos	<ol style="list-style-type: none"> 1. Informar uma doença a ser pesquisada na base OMIM 2. Clicar em “<i>Search</i>” 3. Selecionar um relatório de doença 4. Clicar na aba “<i>Select Proteins</i>” 5. Incluir uma ou mais proteínas válidas manualmente. EX: HTML 6. Selecionar as proteínas extraídas incluindo as informadas manualmente, junto com o organismo HOMO SAPIENS 7. Clicar na aba “<i>Query String</i>” 8. Selecionar as ocorrências válidas 9. Clicar na aba “<i>View Network</i>” 10. Visualizar arquivos gerados pela base STRING ou clicar no <i>link</i> do Cytoscape Web
Resultados Esperados	As proteínas informadas manualmente deverão aparecer nos arquivos disponibilizados pela base STRING e também no Cytoscape Web
Comentários	<p>Inclusão da proteína HTML</p>  <p>HTML é uma proteína válida, conforme imagem abaixo</p> 

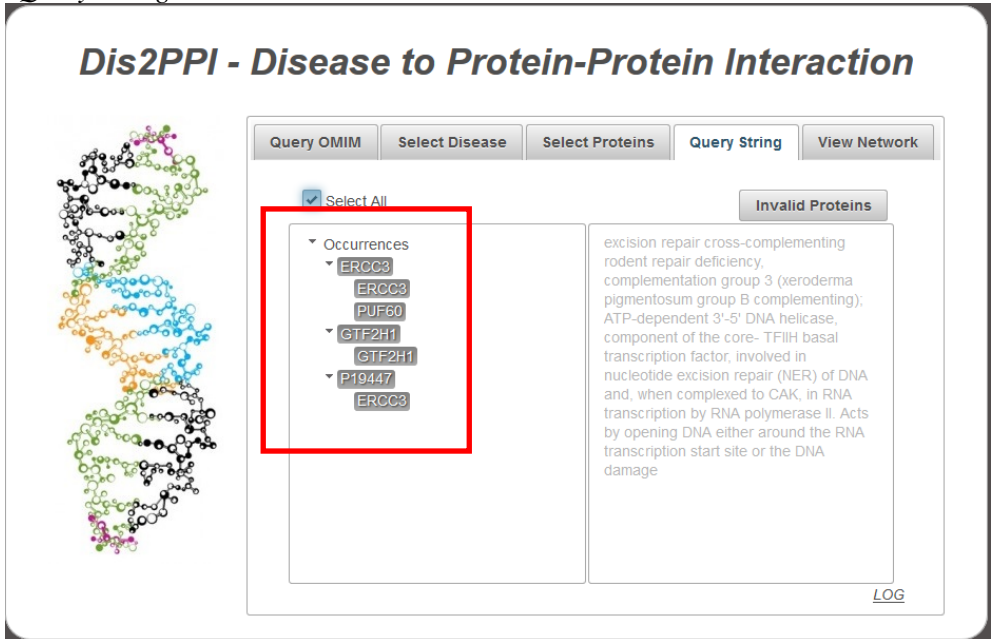
Caso de Teste	03 – Incluir proteínas válidas manualmente
	A proteína aparece na imagem gerada pela base STRING
	

Fonte: Autor

Além de validar a inclusão de proteínas válidas, foi realizada a validação para verificar se o *workflow* reconhece proteínas inválidas incluídas manualmente, conforme mostra o caso de teste 4 (Tabela 7).

Tabela 7. Caso de Teste 4 - Incluir proteínas inválidas manualmente

Caso de Teste	04 – Incluir proteínas inválidas manualmente
Pré-Condição	Existirem dados válidos na base de dados OMIM e STRING Não existir na base OMIM a(s) proteína(s) informadas manualmente.
Passos	<ol style="list-style-type: none"> 1. Informar uma doença a ser pesquisada na base OMIM 2. Clicar em “Search” 3. Selecionar um relatório de doença 4. Clicar na aba “Select Proteins” 5. Incluir uma ou mais proteínas válidas manualmente. EX: ABCD 6. Selecionar as proteínas extraídas incluindo as informadas manualmente, junto com o organismo HOMO SAPIENS 7. Clicar na aba “Query String” 8. Verificar existência ou não da proteína incluída manualmente.
Resultados Esperados	As proteínas informadas manualmente não deverão aparecer na aba “Query String”
Comentários	<p>Inclusão da proteína ABCD</p> 

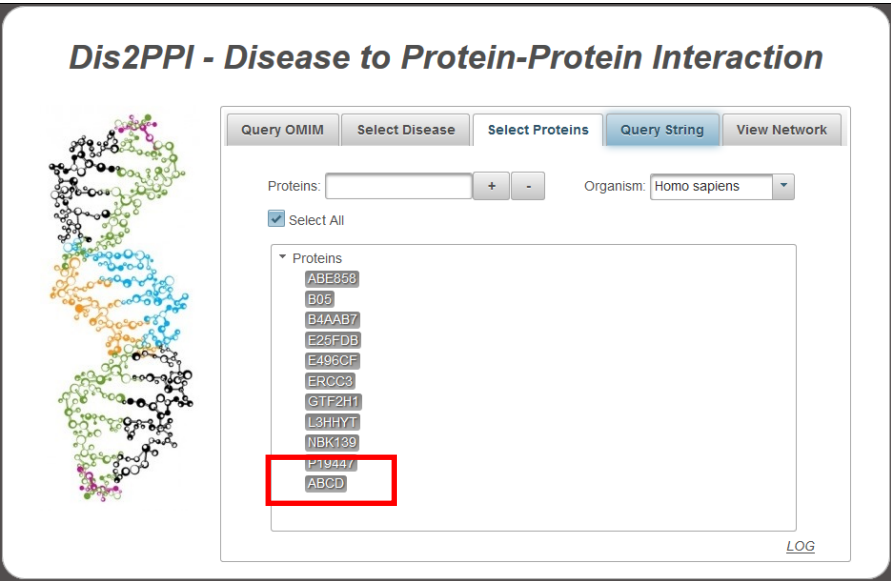
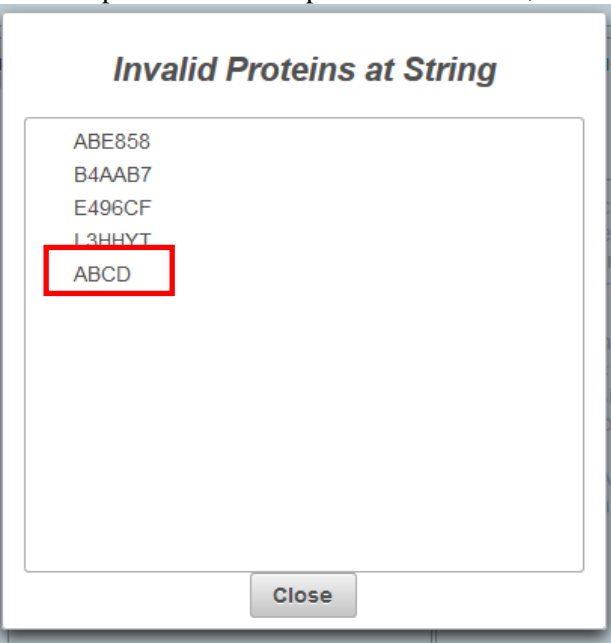
Caso de Teste	04 – Incluir proteínas inválidas manualmente
	<p>ABCD não é uma proteína válida, pois não aparece na árvore “Occurrences” da aba “Query String”</p> 

Fonte: Autor

Com a identificação das proteínas inválidas no caso de teste 4, foi possível realizar a validação da tela de proteínas inválidas retornadas pela base STRING através do caso de teste 5 (Tabela 8).

Tabela 8. Caso de Teste 5 - Validar tela de Proteínas Inválidas

Caso de Teste	05 – Validar tela de Proteínas Inválidas
Pré-Condição	Existirem dados válidos na base de dados OMIM e STRING Não existir na base STRING a(s) proteína(s) informadas manualmente.
Passos	<ol style="list-style-type: none"> 1. Consultar a base STRING através do link http://www.string-db.org/ 2. Digitar a proteína e clicar em “Go” para verificar se a proteína não existe 3. Repetir os passos 1 e 2 até achar uma ou mais proteínas inválidas 4. Informar uma doença a ser pesquisada na base OMIM 5. Clicar em “Search” 6. Selecionar um relatório de doença 7. Clicar na aba “Select Proteins” 8. Incluir uma ou mais proteínas inválidas manualmente. EX: ABCD 9. Selecionar as proteínas extraídas incluindo as informadas manualmente, junto com o organismo HOMO SAPIENS 10. Clicar no botão “Invalid Proteins”
Resultados Esperados	As proteínas selecionadas na aba “Select Proteins” que não aparecem na árvore “Occurrences” da aba “Query String” devem estar na Lista de Proteínas Inválidas
Comentários	Inclusão da proteína ABCD

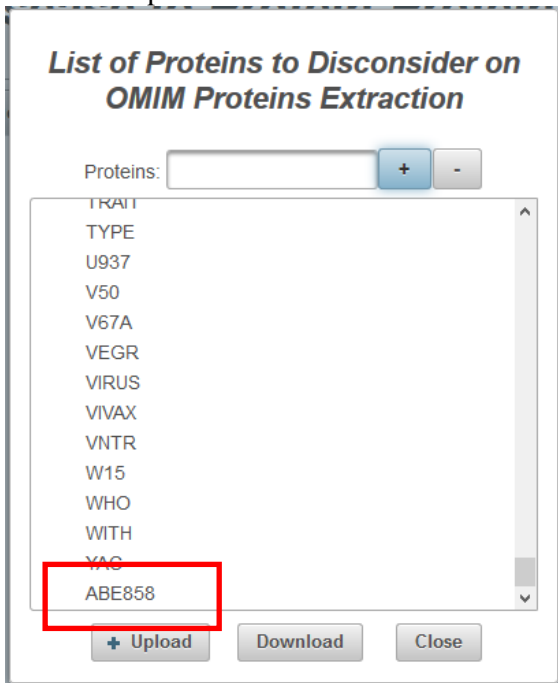
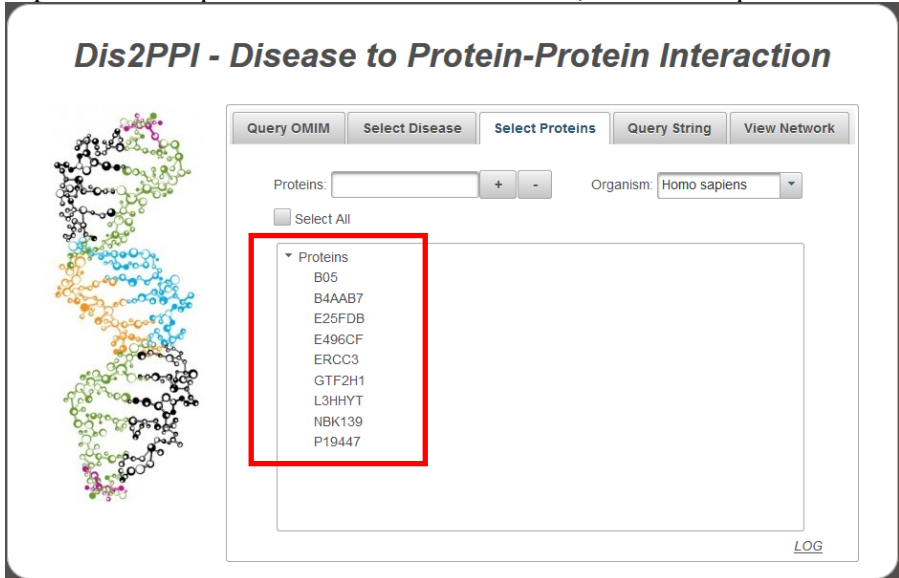
Caso de Teste	05 – Validar tela de Proteínas Inválidas
	 <p>ABCD aparece na tela de proteínas inválidas, conforme esperado.</p> 

Fonte: Autor

O caso de teste 6 (Tabela 9) iniciou-se as validações na tela de Proteínas a Desconsiderar da aba “*Select Disease*”. Primeiro é realizada a validação da inclusão manual de proteínas a desconsiderar.

Tabela 9. Caso de Teste 6 - Validar Inclusão manual de Proteínas a Desconsiderar

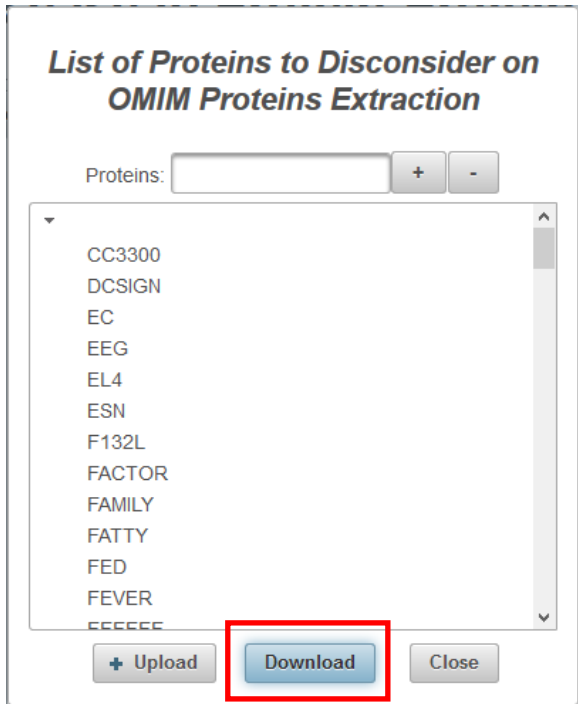
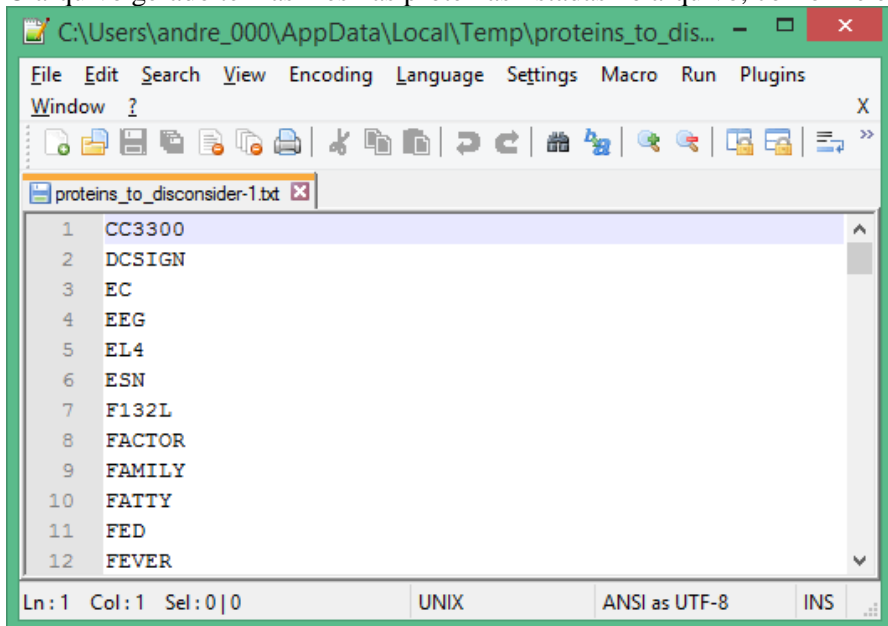
Caso de Teste	06 – Validar Inclusão manual de Proteínas a Desconsiderar
Pré-Condição	Existirem dados válidos na base de dados OMIM e STRING
Passos	<ol style="list-style-type: none"> 1. Informar uma doença a ser pesquisada na base OMIM 2. Clicar em “<i>Search</i>” 3. Clicar no botão “<i>Proteins To Disconsider</i>” 4. Informar uma proteína. EX: ABE858 5. Clicar no botão “+” 6. Clicar no botão “<i>Close</i>”

Caso de Teste	06 – Validar Inclusão manual de Proteínas a Desconsiderar
	7. Selecionar doença 8. Clicar na aba “ <i>Select Proteins</i> ”
Resultados Esperados	A aba “ <i>Select Proteins</i> ” não deve mostrar as proteínas incluídas.
Comentários	<p>Inclusão da proteína ABE858</p>  <p>A proteína não aparece na aba “<i>Select Proteins</i>”, conforme esperado.</p> 

Fonte: Autor

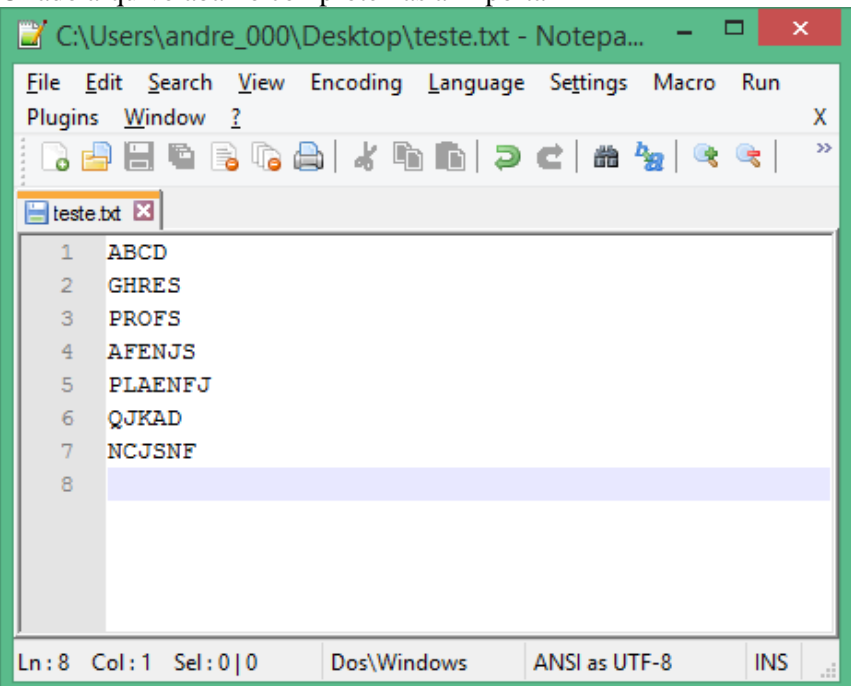
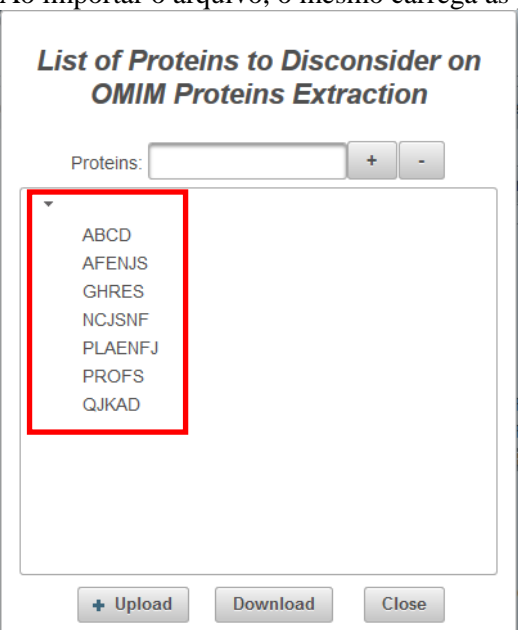
Após a validação da inclusão manual de proteínas, foi realizada a validação do facilitador criado para utilização das proteínas a desconsiderar. O caso de teste 7 (Tabela 10) mostra a validação para *download* das proteínas enquanto o caso de teste 8 (Tabela 11) mostra a validação do *upload* de um arquivo com a lista de proteínas a desconsiderar.

Tabela 10. Caso de Teste 7 - Validar *Download* de Proteínas a Desconsiderar

Caso de Teste	07 – Validar <i>Download</i> de Proteínas a Desconsiderar
Pré-Condição	Existirem dados válidos na base de dados OMIM e STRING
Passos	<ol style="list-style-type: none"> 1. Informar uma doença a ser pesquisada na base OMIM 2. Clicar em “<i>Search</i>” 3. Clicar no botão “<i>Proteins To Disconsider</i>” 4. Clicar no botão “<i>Download</i>” 5. Salvar arquivo
Resultados Esperados	O arquivo deve conter as mesmas proteínas listadas na tela de proteínas a desconsiderar.
Comentários	<p>Ao clicar no botão “<i>Download</i>”</p>  <p>O arquivo gerado tem as mesmas proteínas listadas no arquivo, conforme esperado.</p> 

Fonte: Autor

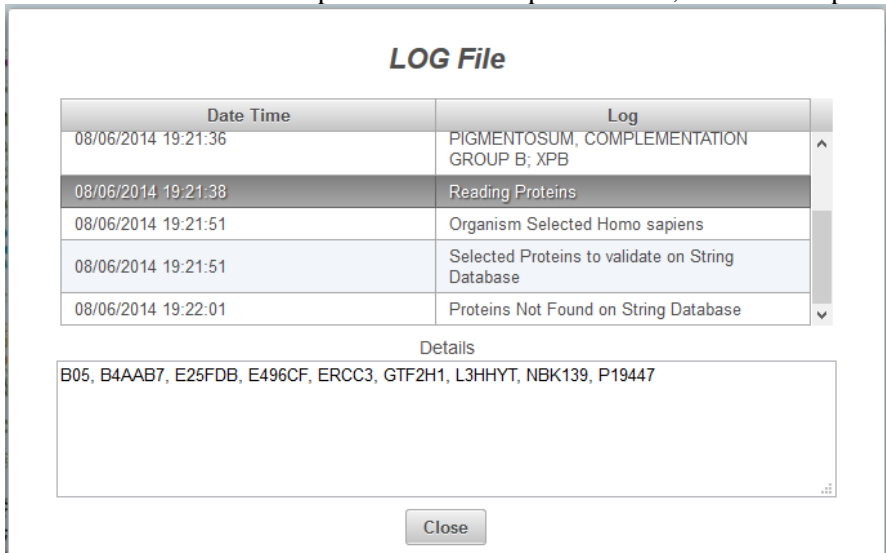
Tabela 11. Caso de Teste 8 - Validar *Upload* de Proteínas a Desconsiderar

Caso de Teste	08 – Validar <i>Upload</i> de Proteínas a Desconsiderar
Pré-Condição	Existirem dados válidos na base de dados OMIM e STRING
Passos	<ol style="list-style-type: none"> 1. Informar uma doença a ser pesquisada na base OMIM 2. Clicar em “<i>Search</i>” 3. Clicar no botão “<i>Proteins To Disconsider</i>” 4. Clicar no botão “<i>Upload</i>” 5. Selecionar um arquivo .txt 6. Importar arquivo
Resultados Esperados	A tela de proteínas a desconsiderar deve conter as mesmas proteínas listadas no arquivo.
Comentários	<p>Criado arquivo abaixo com proteínas a importar</p>  <p>Ao importar o arquivo, o mesmo carrega as proteínas listadas, conforme esperado.</p> 

Fonte: Autor

Por fim, o caso de teste 9 (Tabela 12) demonstra a validação do *LOG* da aplicação.

Tabela 12. Caso de Teste 9 - Validar tela de *LOG* da aplicação

Caso de Teste	09 – Validar tela de <i>LOG</i> da aplicação												
Pré-Condição	Existirem dados válidos na base de dados OMIM e STRING												
Passos	<ol style="list-style-type: none"> 1. Informar uma doença a ser pesquisada na base OMIM 2. Clicar em “<i>Search</i>” 3. Visualizar <i>LOG</i> 4. Selecionar um relatório de doença 5. Clicar na aba “<i>Select Proteins</i>” 6. Visualizar <i>LOG</i> 7. Incluir uma ou mais proteínas válidas manualmente. EX: TESTE 8. Visualizar <i>LOG</i> 9. Remover uma ou mais proteínas. 10. Visualizar <i>LOG</i> 11. Selecionar as proteínas extraídas incluindo as informadas manualmente, junto com o organismo HOMO SAPIENS 12. Clicar na aba “<i>Query String</i>” 13. Visualizar <i>LOG</i> 14. Selecionar as ocorrências válidas 15. Clicar na aba “<i>View Network</i>” 16. Visualizar <i>LOG</i> 17. Visualizar arquivos gerados pela base STRING ou clicar no <i>link</i> do Cytoscape Web 18. Visualizar <i>LOG</i> 												
Resultados Esperados	A tela de <i>LOG</i> deve mostrar os passos realizados pelo usuário.												
Comentários	<p>O <i>LOG</i> está mostrando os passos realizados pelo usuário, conforme esperado.</p>  <p>The screenshot shows a window titled "LOG File" with a table of log entries. The table has two columns: "Date Time" and "Log". The entries are as follows:</p> <table border="1"> <thead> <tr> <th>Date Time</th> <th>Log</th> </tr> </thead> <tbody> <tr> <td>08/06/2014 19:21:36</td> <td>PIGMENTOSUM, COMPLEMENTATION GROUP B; XPB</td> </tr> <tr> <td>08/06/2014 19:21:38</td> <td>Reading Proteins</td> </tr> <tr> <td>08/06/2014 19:21:51</td> <td>Organism Selected Homo sapiens</td> </tr> <tr> <td>08/06/2014 19:21:51</td> <td>Selected Proteins to validate on String Database</td> </tr> <tr> <td>08/06/2014 19:22:01</td> <td>Proteins Not Found on String Database</td> </tr> </tbody> </table> <p>Below the table is a "Details" section with a text area containing the following text: B05, B4AAB7, E25FDB, E496CF, ERCC3, GTF2H1, L3HHYT, NBK139, P19447. A "Close" button is located at the bottom of the window.</p>	Date Time	Log	08/06/2014 19:21:36	PIGMENTOSUM, COMPLEMENTATION GROUP B; XPB	08/06/2014 19:21:38	Reading Proteins	08/06/2014 19:21:51	Organism Selected Homo sapiens	08/06/2014 19:21:51	Selected Proteins to validate on String Database	08/06/2014 19:22:01	Proteins Not Found on String Database
Date Time	Log												
08/06/2014 19:21:36	PIGMENTOSUM, COMPLEMENTATION GROUP B; XPB												
08/06/2014 19:21:38	Reading Proteins												
08/06/2014 19:21:51	Organism Selected Homo sapiens												
08/06/2014 19:21:51	Selected Proteins to validate on String Database												
08/06/2014 19:22:01	Proteins Not Found on String Database												

Fonte: Autor

7 CONSIDERAÇÕES FINAIS

Após a aplicação dos casos de testes concluiu-se que: *i)* o objetivo principal deste projeto foi atingido, a transferência da aplicação para a *web* está funcionando trazendo os mesmos resultados da aplicação *desktop*; *ii)* não existe mais a necessidade de instalação da aplicação e nem configuração de *proxy* de rede, deixando o uso para qualquer profissional da biologia, sem necessidade de conhecimentos técnicos de informática para instalação e utilização da aplicação; *iii)* a utilização do *plugin* do Cytoscape Web foi implementada com sucesso para visualização das redes de interação; *iv)* os *LOGs* da aplicação ficaram mais amigáveis ao usuário, assim como a visualização das proteínas inválidas retornadas pela base *STRING*; *v)* a alteração para possibilitar o *upload* e *download* da lista de proteínas a desconsiderar está funcionando corretamente, possibilitando ao usuário um facilitador para carregar essas informações durante a utilização da aplicação; *vi)* a migração da versão do Java para a versão atual ocorreu com sucesso para a versão 1.8 do Java, liberada durante a execução deste trabalho, deixando assim a ferramenta na versão mais recente disponível.

A arquitetura em camadas facilitou as alterações realizadas, onde foram necessárias apenas alterações nos níveis superiores (*view* e *model*) da aplicação. Nas outras camadas, fez-se necessário alterações pontuais, ou criação de classes novas para utilização nas camadas superiores, como por exemplo as classes de controle do novo *LOG* da aplicação.

Além disso, seguem algumas considerações a respeito da nova aplicação: *i)* foi realizada a validação nos navegadores Firefox e Internet Explorer para garantir a disponibilidade da aplicação funcionando nos diversos navegadores existentes; *ii)* a aplicação não foi validada em dispositivos móveis, pois não é o objetivo da aplicação funcionar neste tipo de dispositivo devido ao processamento que a mesma realiza para busca dos dados; e *iii)* não será criado um controle de usuário, pois não existe necessidade para tal controle.

Dificuldades foram encontradas para hospedagem da aplicação no servidor. Por se tratar de um terceiro que fornece esse serviço, existiram algumas restrições para instalação dos aplicativos necessários, como por exemplo a versão 1.8 do *Java* e o servidor de aplicação *Glassfish*, para a execução da ferramenta e utilização do serviço de hospedagem, o que dificultou a hospedagem da ferramenta. Mesmo atendendo algumas diretrizes do servidor de hospedagem, como por exemplo a troca do servidor de aplicação *Glassfish* para o *Tomcat*, não foi possível disponibilizar a ferramenta na *web* até o momento da entrega deste trabalho.

Foram identificadas algumas melhorias que poderão ser realizadas neste projeto, tais como, *i*) a criação de um controle por *login* de usuário, com registros das consultas realizadas; *ii*) criação da lista de proteínas a desconsiderar vinculada ao usuário, sem ter a necessidade de realizar *upload* e *download* da lista ao acessar a aplicação; e a *iii*) implementação de outras funcionalidades no *plugin* do Cytoscape Web, tais como a inclusão de novas proteínas, inclusão de interações entre as proteínas, visualização das proteínas e interações com mais informações herdadas da base de dados STRING.

8 REFERÊNCIAS

- AMBERGER, Joanna et. al. McKusick's online Mendelian inheritance in man (OMIM®). *Nucleic acids research*, v. 37, n. suppl 1, p. D793-D796, 2009.
- BARNES, Michael R.; GRAY, Ian C. (Ed.). *Bioinformatics for geneticists*. John Wiley & Sons, 2003.
- BLANKENBURG, Hagen et. al. DASMIweb: online integration, analysis and assessment of distributed protein interaction data. *Nucleic acids research*, v. 37, n. suppl 2, p. W122-W128, 2009.
- FREIRE, Juliana et al. Managing rapidly-evolving scientific workflows. In: *Provenance and Annotation of Data*. Springer Berlin Heidelberg, 2006. p. 10-18.
- GIL, Yolanda et. al. Examining the challenges of scientific workflows. *Computer*, v. 40, n. 12, p. 24-32, 2007.
- JENSEN, Lars J. et. al. STRING 8—a global view on proteins and their functional interactions in 630 organisms. *Nucleic acids research*, v. 37, n. suppl 1, p. D412-D416, 2009.
- KARP, Gerald. *Biologia celular e molecular*. Editora Manole Ltda, 2005.
- KARSENTI, Eric. Towards an ‘oceans systems biology’. *Molecular Systems Biology*, v. 8, n. 1, 2012.
- KILLCOYNE, Sarah et al. Cytoscape: a community-based framework for network modeling. In: *Protein Networks and Pathway Analysis*. Humana Press, 2009. p. 219-239.
- KLUG, William. S. et. al. *Conceitos de Genética*. 9ª Edição. Tradução: M. R. Borges-Osório & R. Fischer. ArtMed Editora. Porto Alegre, 2010.
- MINGUEZ, Pablo et. al. SNOW, a web-based tool for the statistical analysis of protein–protein interaction networks. *Nucleic acids research*, v. 37, n. suppl 2, p. W109-W114, 2009.
- NOTARI, Daniel Luis et. al. Dis2PPI: A Workflow Designed to Integrate Proteomic and Genetic Disease Data. *International Journal of Knowledge Discovery in Bioinformatics (IJKDB)*, v. 3, n. 3, p. 67-85, 2012.
- PRESSMAN, Roger S. *Engenharia de software: uma abordagem profissional*. 7ª Edição. Ed: McGraw Hill, 2011.
- PFLEEGER, Shari Lawrence. *Engenharia de software: teoria e prática*. 2ª Edição, Prentice Hall, 2004.
- PROSDOCIMI, Francisco et. al. *Bioinformática: manual do usuário*. *Biotecnologia Ciência & Desenvolvimento*, v. 29, p. 12-25, 2002.

SHANNON, Paul et. al. Cytoscape: a *software* environment for integrated models of biomolecular interaction networks. *Genome research*, v. 13, n. 11, p. 2498-2504, 2003.

SILVA, Fabricio Nogueira; CAVALCANTI, Maria Claudia; DAVILA, Alberto MR. In *Services: Data Management for In Silico Workflows*. In: *Database and Expert Systems Applications*, 2006. DEXA'06. 17th International Workshop on. IEEE, 2006. p. 206-210.

SOMMERVILLE, Ian. *Engenharia de Software*. 9º ed.: Addison Wesley, 2011. 532 pp.

VERA, Roberto et. al. JBioWH: an open-source Java framework for bioinformatics data integration. *Database: the journal of biological databases and curation*, v. 2013, 2013.