

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL  
FACULDADE DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**wCReF: UMA INTERFACE WEB PARA  
O MÉTODO CREF DE PREDIÇÃO DA  
ESTRUTURA 3D APROXIMADA DE PROTEÍNAS**

VANESSA STANGHERLIN MACHADO

Dissertação apresentada como requisito para  
obtenção do grau de Mestre no curso de Pós-  
Graduação em Ciência da Computação da pela  
Pontificia Universidade Católica do Rio Grande  
do Sul.

Orientador: Prof. Dr. Osmar Norberto de Souza

Porto Alegre  
2016

### **Dados Internacionais de Catalogação na Publicação (CIP)**

M149w Machado, Vanessa Stangherlin  
wCReF : uma interface web para o método CReF de  
predição da estrutura 3D aproximada de proteínas / Vanessa  
Stangherlin Machado. –2016.  
199 p.

Diss. (Mestrado) – Faculdade de Informática, PUCRS.  
Orientador: Prof. Dr. Osmar Norberto de Souza

1. Biologia Computacional. 2. Interface com o Usuário.  
3. Informática. I. Souza, Osmar Norberto de. II. Título.


CDD 23 ed. 005.74

**Loiva Duarte Novak CRB 10/2079**  
**Setor de Tratamento da Informação da BC-PUCRS**



## TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "wCReF: Uma Interface Web para o Método CREF de Predição da Estrutura 3D Aproximada de Proteínas." apresentada por Vanessa Stangherlin Machado como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, aprovada em 26 de agosto de 2015 pela Comissão Examinadora:

  
Prof. Dr. Osmar Norberto de Souza -  
Orientador

PPGCC/PUCRS

  
Profa. Dra. Márcia de Borba Campos -

PPGCC/PUCRS

  
Prof. Dr. Rafael Andrade Caceres -

UFCSPA

Homologada em...../...../....., conforme Ata No. .... pela Comissão Coordenadora.

Prof. Dr. Luiz Gustavo Leão Fernandes  
Coordenador.

Dedico esta dissertação aos meus pais Vilma Stangherlin Machado e a Claito Soares Machado (*In Memoriam*). A vocês dois, que não me deram a vida, mas que me ensinaram a vivê-la. Que iluminaram o meu caminho com afeto e dedicação. Que souberam me educar, através do seu exemplo e caráter. Que se doaram por inteiro e renunciaram aos seus sonhos, para que, muitas vezes, eu pudesse realizar os meus. A vocês, pais por opção e amor, não bastaria dizer, que não tenho palavras para agradecer tudo que fizeram por mim. Jamais seria traduzida em palavras todo amor que tenho por vocês. Amo vocês.

## AGRADECIMENTOS

Gostaria de agradecer primeiramente, a Deus, pela presença real e constante em todos os momentos. Por me fazer suportar as privações, por me encorajar e me apoiar todos os dias. Pela proteção diária. Por todas as vezes que pensei negativamente, que não conseguiria, entreguei nas mãos Dele, e fui conduzida ao sucesso. Por toda a chance que me proporciona para meu crescimento. Pela chance de me aproximar de pessoas maravilhosas. Por me dar a vida e a saúde necessária para trabalhar e a oportunidade de realizar este trabalho.

A minha família, pelo amor, incentivo e apoio, principalmente nos momentos de dificuldade. A minha MÃE, que não descansou na torcida, e que certamente orou dias e noites para que Deus estivesse sempre comigo.

Ao Professor Dr. Osmar Norberto de Souza pela orientação dedicada na realização deste trabalho, pela amizade e companheirismo na convivência diária, por seus dotes culinários, pela atenção demonstrada a equipe do LABIO, por ser um excelente educador e principalmente pela confiança em mim depositada. Agradeço por ter acreditado em mim e ter me dado a chance de me tornar mestre com sua orientação.

A Prof.<sup>a</sup> Dr.<sup>a</sup> Milene Selbach Silveira pelas sugestões para complementação deste trabalho, pela amizade e por sua disposição em sempre ajudar, a ouvir e discutir minhas ideias, contribuindo com seu vasto conhecimento.

Agradeço aos especialistas e aos usuários que concordaram em participar desse estudo de forma voluntária, disponibilizando tempo e conhecimento. Sua contribuição foi primordial para a realização deste estudo.

Aos membros da banca de defesa da dissertação Profs. Márcia de Borba Campos e Rafael Andrade Caceres por aceitarem a colaborar para o enriquecimento do trabalho e meu crescimento pessoal, fortalecendo o caráter científico desenvolvido.

Obrigada pelo estímulo e apoio financeiro recebido pelo CNPQ sem os quais não teria sido possível fazer este curso. Obrigada pelo apoio da PUCRS e do Programa de Pós-graduação em Ciência da Computação da PUCRS, em forma de bolsa de isenção de taxas, sem a qual também não teria sido possível sua realização.

Agradeço aos meus colegas do LABIO pela convivência diária, pelas palavras amigas nas horas difíceis, pelos momentos de alegria, pelas reuniões de estudo, de confraternização, pelos lanches compartilhados, pelo sorriso e carinho em que sou recebida diariamente, pela união de todo grupo. Enfim por estarem comigo nesta caminhada tornando-a mais fácil, agradável e feliz. Em especial, a Michele Silva dos Santos e a Walter Ritzel Paixão-Cortes por

compartilharem seu conhecimento, pela disposição em ajudar e auxiliar sempre que solicitado, pela amizade e paciência.

Agradeço aos meus colegas do mestrado pela convivência, pelo auxílio nos trabalhos e dificuldades, por compartilhar suas alegrias, conquistas e preocupações, e principalmente pelos laços de amizade que foram criados. Em especial, a Kassiano José Matteussi pelos dias disponibilizados para ensinar o conteúdo da disciplina de Lógica e a organizar grupos de estudo.

Aos meus queridos amigos, que mesmo estando distante fisicamente, nunca deixaram de me demonstrar carinho, atenção e amizade, se fazendo presente nestes dois anos em que estou morando em Porto Alegre. A minha “irmã de coração” Carmine Ferrari, minha melhor amiga, que sempre me apoiou, me aconselhou, me demonstrou tanto carinho e amizade incondicional, me fazendo rir mesmo nos momentos mais difíceis. As Maravilhosas Aline Tavares, Juh Kochem, Kassiebe Medeiros, a “Barbie” Josiane Novroth, não existe um grupo igual a vocês. Obrigada por me manter sempre presente. Saiba que as palavras diárias que recebo de cada uma de vocês faz meu dia ser mais feliz. Aos meus amigos Marcelo Guerini, Giordani Tavares, Maiquel Copetti, Fernando Bachege, Daiane Bruno, Raquel Reisdorfer, Carla Zysko, Cristiane Cardoso, Dani Lemos, Andréia Mieth, Jokassia Dornelles, Neusa Cavalheiro, Viviane Velasques, Rosane Luz, Aline Zanin, Sibebe Rodrigues e Fernanda Moreira quero dizer que vocês são incríveis. São os melhores amigos do mundo! A todos vocês citados aqui, quero dizer que jamais esquecerei nossa história. Saiba que sempre levarei vocês no coração, e que sua amizade é um bem precioso que tenho na vida, um presente de Deus.

Por fim agradeço ao meu namorado Walter Ritzel Paixão-Cortes, ouvinte atento de algumas dúvidas, inquietações, desânimos e sucessos, pelo apoio, pela confiança e pela valorização sempre tão entusiasta do meu trabalho, dando-me, desta forma, coragem para ultrapassar as dificuldades. Por todos os dias que compartilhamos da mesma dedicação, realizando junto muitas tarefas do mestrado. E porque ele divide comigo os sonhos e planos futuros. Pelo carinho, companheirismo e amor, sem os quais eu não teria conseguido. Agradeço por ter entrado na minha vida, por fazer ela tão feliz.

*“Dar menos que seu melhor é sacrificar o dom que você recebeu.” (Steve Prefontaine)*

*“As macromoléculas possuem propriedades que não podem ser previstas a partir das propriedades das suas unidades constituintes”... “Aparentemente o único obstáculo para compreender a natureza da vida é a sua fantástica complexidade. A Informática dá nos a esperança que um dia possamos vencer também esta dificuldade”.*  
Hermann Staudinger, The Nobel Prize in Chemistry 1953.  
(H.A. Krebs, Persp. Biol. Med., 1971 apud Krippahl, 1999).

## RESUMO

A predição da estrutura terciária de proteínas é um problema da Bioinformática Estrutural ainda não solucionado pela ciência. O desafio é entender a relação entre a sequência de aminoácidos de uma proteína e sua estrutura tridimensional 3D, que está relacionada à função destas macromoléculas. Dentre os métodos relacionados à predição de estruturas de proteínas está o Método CReF (*Central Residue Fragment-based Method*), proposto por Dorn & Norberto de Souza (2008), para predição da estrutura 3D aproximada de uma proteína ou polipeptídio. Nesta dissertação, apresentamos o wCReF, a interface Web para o método CReF desenvolvida com o enfoque em usabilidade. Com esta ferramenta o usuário informa a sequência de aminoácidos de sua proteína alvo, e como resultado obtém a estrutura 3D aproximada de uma proteína, de forma automatizada, sem a necessidade de instalação das ferramentas necessárias para sua utilização. Para definir os requisitos necessários para seu desenvolvimento foram realizadas avaliações de usabilidade, realizadas por especialistas em Interação Humano-Computador e da área de domínio, a bioinformática, em três servidores de predição de estruturas de proteínas - I-TASSER, QUARK e Robetta - todos participantes do CASP (*Critical Assessment of protein Structure Prediction*). As inspeções foram realizadas através do método de Avaliação Heurística, utilizando as 10 heurísticas de Nielsen. Como resultado, foram encontradas violações em todas as heurísticas e detectados 89 problemas de usabilidade. Eles foram classificados em 5 severidades, 29 pontuados como sendo de alta prioridade de solução e 25 problemas de resolução imediata. Os resultados das avaliações serviram como orientação norteadora dos principais recursos que o wCReF deveria possuir, compilados em um documento de requisitos de software, para sua implementação. A partir desta etapa foi realizada a prototipagem, vislumbrando a detecção de novos problemas de usabilidade, com os usuários finais, através da adaptação do questionário de satisfação de Ssemugabi. Como produto final, apresentamos o servidor wCReF alicerçado na preocupação com a usabilidade e interação com seus usuários. Além disso, este estudo pode contribuir para melhoria da usabilidade das aplicações de bioinformática já existentes, dos servidores de predição analisados e no desenvolvimento de novas ferramentas científicas.

Palavras chaves: Bioinformática. Servidores de predição de estruturas terciárias de proteínas. Usabilidade. Interface Web. Avaliação Heurística.



## ABSTRACT

The prediction of protein tertiary structure is a problem of Structural Bioinformatics still unsolved by science. The challenge is to understand the relationship between the amino acid sequence of a protein and a three dimensional structure, which is related to the function of these macromolecules. Among the methods related to protein structure prediction is CREF (*Central Residue Fragment-based Method*) proposed by Dorn & Norbert Souza (2008) for prediction of proteins' or polypeptide's approximate 3D structure. In this thesis we present the wCReF, the Web interface for the CREF method developed with a focus on usability. With this tool the users can enter the amino acid sequence of its target protein, and get as a result the approximate 3D structure of a protein in an automated manner without the need to install all the necessary tools for their use. To define the requirements for its development were conducted usability evaluations, guided by experts on both Human-Computer Interaction and bioinformatics domain areas, in three protein structures prediction servers - I-TASSER, QUARK and Robetta - all participants at CASP (*Critical Assessment of Protein Structure Prediction*) competition. The inspections were conducted through the Heuristic Evaluation method using Nielsen's 10 heuristics. As a result, violations were found in all heuristics resulting in 89 usability problems. They were classified into 5 severities, 29 scored as being of high priority solution and 25 as problems to be solved immediately. Assessment results serve as guiding orientation of the key features that wCReF must have compiled in a document software requirements for its implementation. From this step was carried out prototyping and glimpsing the detection of new usability problems with end users by adapting the Ssemugabi satisfaction questionnaire. As a final product we present the wCReF server, protein structure prediction server rooted in concern about the usability and interaction with its users. Furthermore, this study can contribute to improvement of usability of existing bioinformatics applications, the prediction servers analyzed and the development of new scientific tools.

## LISTA DE ILUSTRAÇÕES

<b>Figura 1.</b> Ciclo de vida da interface do wCReF .....	22
<b>Figura 2.</b> O processo de interação humano-computador .....	26
<b>Figura 3.</b> Usabilidade como facilidade de uso.....	27
<b>Figura 4.</b> Ciclo de desenvolvimento de aplicações Web. O ciclo de desenvolvimento de interfaces Web é uma espiral contínua com seis etapas .....	30
<b>Figura 5.</b> Entradas do PDB ( <i>Protein Data Bank</i> ).....	38
<b>Figura 6.</b> Kendrew e colaboradores publicaram a primeira estrutura de uma proteína.....	39
<b>Figura 7.</b> Como podemos determinar a estrutura nativa de uma proteína a partir do conhecimento da sua estrutura primária? .....	41
<b>Figura 8.</b> Esquema dos passos do CReF.....	48
<b>Figura 9.</b> Problemas de usabilidade encontrados pelos avaliadores.....	61
<b>Figura 10.</b> Menu do I-TASSER.....	63
<b>Figura 11.</b> Uma das heurísticas violadas - Auxiliar os usuários a reconhecer, diagnosticar e recuperar erros .....	64
<b>Figura 12.</b> Graus de severidade atribuídos nas avaliações dos servidores .....	64
<b>Figura 13.</b> Esboço da interface feito na ferramenta PENSIL .....	70
<b>Figura 14.</b> Tela inicial da interface do protótipo desenvolvido (Home) .....	72
<b>Figura 15.</b> Tela inicial do wCReF. Esta é a tela principal do servidor.....	73
<b>Figura 16.</b> Respostas dos questionários.....	78
<b>Figura 17.</b> Total de respostas Positivas, Negativas e Neutras por tópico.....	79
<b>Figura 18.</b> Número de questões negativas avaliadas na concepção da interface.....	80
<b>Figura 19.</b> Questões avaliadas como negativas em <i>Web design</i> .....	84
<b>Figura 20.</b> As questões avaliadas como negativas em bioinformática. ....	85
<b>Figura 21.</b> Parte do código do wCReF que mostra que as telas da interface são adaptáveis..	88
<b>Figura 22.</b> A mensagem de confirmação de uma submissão .....	93
<b>Figura 23.</b> O wCReF. O usuário se comunica com a aplicação através da interface, submetendo sua requisição por meio de um formulário. ....	94
<b>Figura 24.</b> O protótipo deu origem a interface Web.....	95
<b>Figura 25.</b> Arquitetura da interface Web do wCReF.....	96
<b>Figura 26.</b> Descrição Hierárquica do Sistema. ....	100
<b>Figura 27.</b> Template da interface de Apresentação. ....	102
<b>Figura 28.</b> Home ou Tela de Apresentação. ....	104

<b>Figura 29.</b> Interface do servidor do I-TASSER.....	105
<b>Figura 30.</b> O login pode ser realizado de qualquer parte da interface.....	105
<b>Figura 31.</b> Tela de submissão .....	106
<b>Figura 32.</b> Template da Interface de usuário do wCReF.....	107
<b>Figura 33.</b> O perfil de usuário.....	108
<b>Figura 34.</b> Tela de submissão de predições de proteínas.....	109
<b>Figura 35.</b> Tela de acompanhamento das submissões.....	111
<b>Figura 36.</b> Tela de visualização do wCReF.....	113
<b>Figura 37.</b> Representação da estrutura secundária .....	113
<b>Figura 38.</b> Visualizador do wCReF.....	114
<b>Figura 39.</b> Telas da Interface fale-conosco.....	117
<b>Figura 40.</b> Telas da Interface Equipe.....	117
<b>Figura 41.</b> Exemplo da ajuda da interface do wCReF.....	119
<b>Figura 42.</b> Sobre o CReF.....	120
<b>Figura 43.</b> As referências dos artigos do CReF.....	121
<b>Figura 44.</b> <i>Softwares</i> relacionados com a metodologia do CReF.....	121

## LISTA DE TABELAS

<b>Tabela 1.</b> Fatores pertinentes na aplicação de usabilidade em sistemas.....	34
<b>Tabela 2.</b> Conjunto de Heurísticas de Nielsen.....	35
<b>Tabela 3.</b> Números de servidores participantes do CASP das últimas 8 edições.....	42
<b>Tabela 4.</b> Os Principais servidores de predição participantes do CASP.....	44
<b>Tabela 5.</b> As 8 regiões conformacionais utilizadas do mapa de <i>Ramachandran</i> no método CReF, segundo Thornton e colaboradores. ....	50
<b>Tabela 6.</b> Os servidores de predição avaliados nesta pesquisa.....	59
<b>Tabela 7.</b> Inspectores – Avaliação Heurística .....	59
<b>Tabela 8.</b> Escala de severidade atribuída na Avaliação Heurística .....	60
<b>Tabela 9.</b> Problemas de Usabilidade encontrados na Avaliação Heurística.....	61
<b>Tabela 10.</b> Problemas adicionais encontrados nos servidores de predição .....	66
<b>Tabela 11.</b> Questões de satisfação de usabilidade para a área de bioinformática em servidores de predição de estruturas de proteínas.....	74
<b>Tabela 12.</b> Números atribuídos as respostas segundo a Escala de Likert.....	75
<b>Tabela 13.</b> Perfil dos participantes do questionário de avaliação.....	76
<b>Tabela 14.</b> Perguntas relacionadas a experiência do usuário com a predição de estruturas de proteínas .....	77
<b>Tabela 15.</b> Problemas encontrados nas avaliações dos questionários considerados de Média e Alta prioridade de correção. ....	81
<b>Tabela 16.</b> Características que foram contempladas no servidor wCReF após a análise dos problemas encontrados nos servidores. ....	97
<b>Tabela 17.</b> Parâmetros Opcionais do servidor.....	110

## LISTA DE SIGLAS

3D – Tridimensional

ABNT – Associação Brasileira de Normas Técnicas

AMBER – *Assisted Model Building with Energy Refinement*

BE – Bioinformática Estrutural

CASP – *Critical Assessment of protein Structure Prediction*

CGIs – *Common Gateway Interface*

CReF – *Central Residue Fragment-based Method*

EMBL-EBI – *European Molecular Biology Laboratory - European Bioinformatics Institute*

EUA – *Estados Unidos da América*

HCSE – *Human-Centered Software Engineering*

HTML – *Hypertext Markup Language*

IHC – *Interação Humano-Computador*

ISO – *International Organization Standardization*

LABIO – *Laboratório de Bioinformática, Modelagem e Simulação de Biosistemas*

NCBI – *National Center for Biotechnology Information*

NCSA – *National Center for Supercomputing Applications*

ONU - *Organização das Nações Unidas*

PSP – *Protein Structure Prediction Problem*

SGBD – *Sistema de gerenciamento de banco de dados*

URI – *Universal Resource Identifier*

W3C – *World Wide Web Consortium*

WCAG - *Web Content Accessibility Guidelines*

WEKA – *Waikato Environment for Knowledge Analysis*

WHATWG – *Web Hypertext Application Technology Working Group*

WWW – *World Wide Web*

XML - *Extensible Markup Language*

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
1.1	MOTIVAÇÃO	19
1.2	OBJETIVO GERAL	20
1.3	OBJETIVOS ESPECÍFICOS	21
1.4	METODOLOGIA	21
1.5	ORGANIZAÇÃO DA DISSERTAÇÃO	22
<b>2</b>	<b>REVISÃO DE LITERATURA</b>	<b>24</b>
2.1	PARTE 1 - REVISÃO DOS ASPECTOS REFERENTES AO DESENVOLVIMENTO DA INTERFACE WEB	24
2.1.1	INTERAÇÃO HUMANO-COMPUTADOR	24
2.1.2	INTERFACE	25
2.1.3	INTERFACES DE USUÁRIO E USABILIDADE	27
2.1.4	O DESENVOLVIMENTO DE INTERFACES WEB	29
2.1.5	A AVALIAÇÃO DE USABILIDADE	33
2.2	PARTE 2 - REVISÃO DOS CONCEITOS RELACIONADOS A COMPREENSÃO DO MÉTODO CREF	37
2.2.1	A BIOINFORMÁTICA	37
2.3	PREDIÇÃO DE ESTRUTURAS 3D DE PROTEÍNAS	39
2.4	PROTEIN STRUCTURE PREDICTION PROBLEM - PSP	40
2.5	SERVIDORES DE PREDIÇÃO	41
2.5.1	CASP ( <i>CRITICAL ASSESSMENT OF TECHNIQUES FOR PROTEIN STRUCTURE PREDICTION</i> )	41
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>44</b>
3.1	MÉTODOS COMPUTACIONAIS PARA PREDIÇÃO DE ESTRUTURAS TRIDIMENSIONAIS DE PROTEÍNAS	45
3.1.1	MODELAGEM POR HOMOLOGIA	45
3.1.2	RECONHECIMENTO DE PADRÕES OU <i>FOLD RECOGNITION</i>	45
3.1.3	PREDIÇÃO AB INITIO	46
3.2	O MÉTODO CREF	46
3.2.1	ETAPAS DO CREF	47
3.2.2	UTILIZAÇÃO DO MÉTODO CREF	51
3.3	PESQUISAS EM IHC E BIOINFORMÁTICA	53

<b>4</b>	<b>RESULTADOS.....</b>	<b>58</b>
4.1	CONHECER OS USUÁRIOS E SUAS TAREFAS .....	58
4.2	AVALIAÇÃO HEURÍSTICA .....	58
4.2.1	AVALIAÇÃO ADICIONAL DOS SERVIDORES DE PREDIÇÃO .....	66
4.3	DOCUMENTO DE REQUISITOS .....	69
4.4	DESCRIÇÃO DO PROTÓTIPO .....	70
4.5	AVALIAÇÃO POR QUESTIONÁRIOS .....	73
4.5.1	RESULTADO DA AVALIAÇÃO POR QUESTIONÁRIOS REALIZADAS PELOS USUÁRIOS.....	76
4.5.2	PONTOS CONSIDERADOS NEGATIVOS NA AVALIAÇÃO DE USABILIDADE.....	80
4.5.3	RESPOSTAS DAS QUESTÕES ABERTAS.....	86
4.5.4	ALGUNS PONTOS QUE CONTRIBUÍRAM PARA MELHORIA DA INTERFACE.....	92
<b>5</b>	<b>WCREf .....</b>	<b>94</b>
5.1	A INTERFACE DO WCREf .....	95
5.1.1	CONCEPÇÃO DA INTERFACE .....	96
5.1.2	TELAS DA INTERFACE.....	99
5.1.3	INTERFACE DE APRESENTAÇÃO .....	101
5.1.4	INTERFACE DE USUÁRIO.....	107
5.1.5	TEMPO DE EXECUÇÃO.....	115
5.1.6	A AJUDA E DOCUMENTAÇÃO DISPONIBILIZADA NO WCREf .....	115
<b>6</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>122</b>
6.1	TRABALHOS FUTUROS.....	123
	<b>REFERÊNCIAS .....</b>	<b>125</b>
	<b>APENDICE A – MANUAL DE INSTALAÇÃO DO MÉTODO CREF EM COMPUTADOR LOCAL.....</b>	<b>135</b>
	<b>APENDICE B – ORIENTAÇÕES: AVALIAÇÃO HEURÍSTICA DE SERVIDORES DE PREDIÇÃO DE ESTRUTURAS DE PROTEÍNAS ..</b>	<b>149</b>
	<b>APENDICE C – FORMULÁRIO DE AVALIAÇÃO HEURÍSTICA.....</b>	<b>154</b>
	<b>APENDICE D – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO PARA ESPECIALISTAS.....</b>	<b>155</b>

<b>APENDICE E – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO PARA OS USUÁRIOS .....</b>	<b>156</b>
<b>APENDICE F – DOCUMENTO DE REQUISITOS DE SOFTWARE ....</b>	<b>157</b>
<b>APÊNDICE G – QUESTIONÁRIO DE AVALIAÇÃO DO SERVIDOR DE PREDIÇÃO DE ESTRUTURAS DE PROTEÍNAS (WCREF) .....</b>	<b>183</b>
<b>APENDICE H – TUTORIAL PARA SUBMISSÃO DE PROTEÍNAS NO WCREF .....</b>	<b>198</b>



## 1 INTRODUÇÃO

O surgimento da bioinformática se deve à necessidade de analisar grandes conjuntos de dados complexos, o que impulsionou a mudança das ferramentas usadas nas pesquisas biológicas: o navegador Web é atualmente a ferramenta mais difundida disponível para os pesquisadores, pois proporciona o acesso a uma poderosa ferramenta computacional e estatística (Shaer et al., 2010; Skinner et al., 2009). Além disso, a disponibilização de ferramentas através da Web facilita o acesso aos usuários, pois ferramentas científicas podem ser difíceis de instalar, sendo particularmente útil seu uso através de uma interface Web (Néron et al., 2009).

Essa tecnologia vem então a ser massivamente adotada em uma grande quantidade de *softwares* de bioinformática. Contudo, apesar da grande quantidade de aplicações Web neste domínio, não ocorre necessariamente, a satisfação dos usuários que se confrontam muito frequentemente com problemas de usabilidade. Embora os avanços da área de IHC (Interação Humano-Computador) sejam aplicados em uma vasta gama de domínios de aplicações, incluindo a resolução de problemas, educação e entretenimento, pouco da pesquisa nesta área tem se dedicado a investigar interfaces no contexto da pesquisa científica profissional (Shaer et al., 2010).

Sendo assim, as ferramentas Web existentes na área de bioinformática mostram graves limitações, em termos de usabilidade, suporte e ao alto nível de raciocínio (Bolchini et al., 2009; Mirel e Wright, 2009; Rutherford et al., 2010; Veretnik et al., 2008). Embora a comunidade científica forneça dados precisos e valiosos, os usuários encontram nestes recursos interfaces complicadas de usar e navegar (Pavelin et al., 2012). Não é incomum que os usuários dessas ferramentas experimentem uma acentuada curva de aprendizado, sendo oprimidos pela complexidade da realização de tarefas padrões (Rutherford et al., 2010).

Podemos relacionar esta questão com os perfis dos desenvolvedores deste tipo de software. De um lado estão os bioinformatas que são da área biológica (biólogos, químicos e físicos). Percebe-se que estes tendem a não focar na usabilidade das interfaces, e existem várias razões para isso. Por exemplo, é difícil produzir interfaces amigáveis para os recursos de bioinformática, porque os dados que são apresentados são muito complexos (Pavelin et al., 2012). Também é comum que estes desenvolvedores estejam interessados, principalmente, em obter saídas precisas e significativas como, por exemplo, a estrutura 3D aproximada de uma proteína, não sendo o foco principal o estudo da interação usuário-sistema, envolvendo as questões de usabilidade. Do outro lado, há os desenvolvedores de *softwares* (cientistas da

computação, analistas de sistema, engenheiros, designers) que muitas vezes têm pouca compreensão das necessidades dos pesquisadores da área, não conhecendo mais profundamente os aspectos biológicos que estão envolvidos no processo.

Portanto é fundamental fornecer uma interface adequada e eficiente para que o cientista possa ter acesso ao conteúdo das bases de conhecimento (Bisson e Garreau, 1995), sendo necessário, assim, o estudo dos problemas de usabilidade, que é um critério importante no *design* de *softwares* em bioinformática, sendo algo de profundo valor para os pesquisadores que estão cada vez mais dependentes dessas interfaces (Rutherford et al., 2010).

A presente pesquisa é parte de um estudo que tem como objetivo principal o desenvolvimento da interface Web para o método CReF (*Central Residue Fragment-based Method*). O CReF é um método de predição de estruturas de proteínas aproximadas, proposto por Dorn e Norberto de Souza (Dorn e Norberto de Souza, 2008). Apesar de possuir resultados promissores (Dall'Agno, 2012; Dall'Agno e Norberto de Souza, 2013; Dorn et al., 2008; Dorn e Norberto de Souza, 2010, 2008), esta ferramenta não possui uma interface para comunicação com o usuário, o que dificulta a sua utilização e justifica o desenvolvimento da mesma. Além disso, a opção da interface ser disponibilizada através da Web visa garantir um maior acesso aos usuários, independente do sistema operacional e da instalação dos recursos necessários para o funcionamento da metodologia na máquina local. Desta forma, o wCReF funciona como um servidor para troca de informações: o usuário informa a entrada de dados, no caso a proteína alvo, o wCReF processa e devolve como resultado a estrutura 3D aproximada de uma proteína.

O estudo do desenvolvimento da interface do servidor do wCReF está alicerçado a garantir uma boa usabilidade para seus usuários, buscando em todas as fases do design, desde o levantamento de requisitos de software, a prototipação, até a implementação, encontrar os principais problemas de usabilidade, realizando avaliações em todo o processo. Dessa forma acreditamos que para desenvolver uma interface para o wCReF com qualidade, a análise dos servidores de predição de proteínas atuais, buscando problemas de usabilidade é essencial.

Neste sentido é que foi realizado um estudo dos principais servidores de Predição de Estruturas de Proteínas participantes do CASP (*Critical Assessment of Techniques for Protein Structure Prediction*) visando a especificação dos requisitos de software do wCReF. Foram analisadas as interfaces dos Servidores, em busca de problemas de usabilidade. O método de avaliação escolhido foi a Avaliação Heurística (Nielsen, 1994a), realizada por especialistas da área de IHC e da área de domínio (bioinformática). Para isto foram selecionados três servidores I-TASSER (Roy et al., 2010; Zhang, 2008), QUARK (Zhang, 2014) e Robetta (Kim et al., 2004).

Os resultados dessas avaliações serviram para definir as características do novo servidor, que foram descritas no documento de requisitos de software para construção de um protótipo. Posteriormente à implementação, testes com os usuários finais foram realizados, utilizando uma adaptação do questionário de Ssemugabi (Ssemugabi, 2009), cujo enfoque é avaliar a satisfação de usuários, visando obter um feedback sobre a aplicação e encontrar novos erros de usabilidade, que foram evitados na concepção do produto final. Contemplamos assim, como resultado final desta pesquisa, a partir do feedback dos usuários, a interface do wCReF que ficará disponível para uso da comunidade científica através da Web.

## 1.1 Motivação

Apesar do grande progresso na determinação experimental de estruturas tridimensionais de proteínas, o número de estruturas preditas não acompanhou o ritmo de crescimento explosivo de informação sobre sequências (Marks et al., 2012), pois a elucidação das sequências é uma tarefa relativamente mais simples comparada a predição de estruturas tridimensionais (3D) de proteínas (Filho et al., 2003). Isto nos leva ao fato que há 1000 vezes mais sequências que estruturas, sendo que não há uma variedade de métodos que podem prever as estruturas tridimensionais de proteínas (Dill e MacCallum, 2012). Dessa forma, a diferença entre o número de sequências e de proteínas com estruturas ou funções conhecidas continua a aumentar, em um ritmo exponencial (Roche et al., 2011).

Então, o desenvolvimento de métodos computacionais para prever estruturas 3D a partir de sequências é provavelmente o único caminho para preencher a lacuna entre a quantidade de sequências e resolução destas estruturas (Lee et al., 2009). O método CReF (Dorn e Norberto de Souza, 2008) demonstrou bons resultados na predição de estruturas de proteínas, demonstrando potencial científico para mais estudos e aplicações.

A execução do CReF é realizada localmente, em plataforma *Linux*, necessitando por parte do usuário conhecimento para sua instalação, configuração e utilização. Toda a metodologia empregada para se utilizar o método necessita de instalação de pacotes da linguagem *python* no computador local e instalação de *softwares* externos (*Torsions* - Dr. Andrew C.R. Martin's Group, 2014), *Weka* (Frank et al., 2005, 2004), *Amber* (Kollman et al., 1995), dentre outros). Outro fator importante, são as alterações que precisam ser feitas no código fonte (indicando o caminho para diretórios e pastas). Tudo isso torna dispendiosa sua execução. Estes fatores fazem com que o método se torne difícil de ser utilizado por usuários menos experientes, além de sua aplicação ficar restrita ao local de instalação.

Portanto, a automatização do método CReF é adequada, pois devemos “*simplificar a acesso a recursos computacionais, proporcionando um ambiente gráfico mais familiar para usuários inexperientes, salvando-os de instalar software no seu próprio computador*”(Néron et al., 2009). Além disso, é importante disponibilizar esta metodologia ao alcance dos usuários. Atualmente, podemos disponibilizar o método por meio de uma interface Web, como um servidor de predição de estruturas de proteínas.

Podemos observar essa tendência ao analisar o CASP (*Critical Assessment of Techniques for Protein Structure Prediction*) (Moult et al., 2014, 2005, 1997, 1995), evento mundial de avaliação de estruturas de proteínas, que possui em uma de suas categorias, os servidores de predição, que realizam simulações computacionais buscando prever a estrutura 3D aproximada de proteínas. Estes servidores realizam as predições de forma mais rápida que em métodos experimentais, de forma automatizada, e estão obtendo cada vez mais modelos de estruturas tridimensionais com boa qualidade e precisão.

Os usuários poderão utilizar o método em diferentes sistemas operacionais, não necessitando a instalação de nenhuma ferramenta, precisando de apenas de um navegador Web e de acesso à internet. Isso permitirá que o método seja acessado de qualquer lugar, onde haja acesso à rede. O wCReF também tem potencial de se tornar um dos servidores de predição utilizados por pesquisadores no CASP. Outra vantagem prevista é a facilidade para futuros melhoramentos e refinamentos no método, sem a necessidade de que a cada atualização seja instalada individualmente em cada computador host que é utilizado.

Por fim, além das justificativas apresentadas até aqui, este estudo possui duas contribuições gerais. No campo das ciências biológicas por disponibilizar em nível mundial, esta metodologia computacional que pode contribuir com os esforços experimentais para predição de proteínas buscando diminuir a lacuna entre sequências e estruturas. E no campo da Ciência da computação, pelo estudo da usabilidade das ferramentas em bioinformática, podendo vir a contribuir para a melhoria da interface dos servidores de predição analisados quanto como guia para o desenvolvimento de novas aplicações.

## **1.2 Objetivo geral**

O objetivo geral desde trabalho é o desenvolvimento de uma interface Web para o método CReF, tornando assim o método automatizado para os usuários, e disponível para acesso público através da internet. Em sua criação serão aplicados os conceitos relacionados à área da Interação Humano-Computador (IHC).

### 1.3 Objetivos específicos

- Realizar o levantamento de requisitos de software para o desenvolvimento da interface web do wCReF, através da avaliação de usabilidade em servidores de predição de estruturas de proteínas. Este documento poderá ser utilizado para o desenvolvimento de outros servidores de predição de estruturas de proteínas e aplicações da área da bioinformática.
- Elaborar uma ferramenta para a avaliação da interface do wCReF a ser utilizada pelos usuários finais, para apontar os problemas de usabilidade que possam existir na interface do servidor e para propor melhorias. Os problemas encontrados serão corrigidos na versão final. Esta ferramenta também poderá ser utilizada para avaliação de usabilidade de outros servidores de predição de estruturas de proteínas.

### 1.4 Metodologia

A partir dos objetivos dessa pesquisa foram propostas as seguintes atividades:

- Analisar os principais aspectos relacionados a usabilidade no desenvolvimento de interfaces Web, com uma constante revisão de literatura;
- Definir qual o método de avaliação de usabilidade será utilizado para o levantamento de requisitos.
- Especificar os requisitos da interface wCReF, conforme as orientações obtidas nas avaliações, respeitando os aspectos da área de IHC para o desenvolvimento de interfaces Web.
- Definir todos os recursos necessários para o desenvolvimento da interface Web.
- Desenvolver um protótipo Web do wCReF segundo os requisitos definidos e integrar com o método CReF.
- Aplicar a avaliação de usabilidade com os usuários finais, e analisar os resultados.
- Desenvolver o produto final, a interface do wCReF, através dos resultados obtidos.

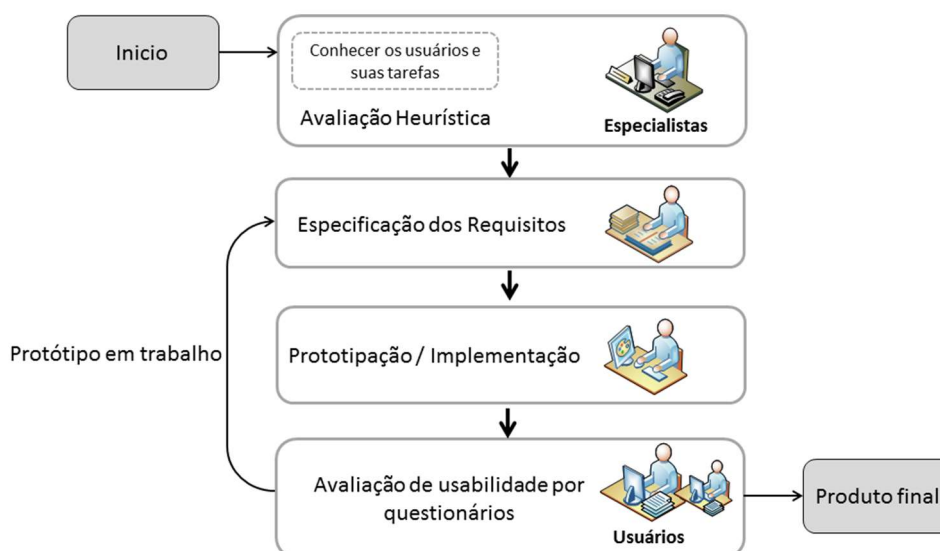
Para a realização das atividades de avaliação de usabilidade, buscamos contemplar a tendência atual de avaliação de interfaces Web, que é, identificar os problemas de usabilidade tão cedo eles possam ser detectados. Dessa forma, propomos os seguintes objetivos para a avaliação de interfaces (Preece et al., 2005; Rex Hartson, 1998):

- 1) Identificar as necessidades dos usuários;
- 2) Identificar problemas de interação ou de interface (Avaliação Heurística);
- 3) Comparar os resultados para definir o documento de requisitos e realizar o protótipo da interface;

- 4) Verificar se o protótipo está em conformidade com um padrão ou conjunto de heurísticas (questionário).

O ciclo de desenvolvimento da interface Web do método CReF (Figura 1) foi baseado no ciclo de desenvolvimento de interfaces apresentado por Winckler e Pimenta (Winckler e Pimenta, 2002).

O ciclo inicia pela definição dos usuários das tarefas que são comumente realizadas pelos bioinformatas que utilizam os servidores de predição de estruturas de proteínas. Com isto, avaliações heurísticas são realizadas por especialistas. O próximo passo é a construção de um protótipo que, em seguida, é avaliado com relação a sua usabilidade através de questionários. Os problemas de usabilidade, identificados na avaliação, são solucionados. A primeira versão então é apresentada.



**Figura 1.** Ciclo de vida da interface do wCReF. O Ciclo inicia-se pela avaliação de usabilidade por especialistas, passando para a especificação dos requisitos de software como resultado das avaliações. Após a implementação de um protótipo de alta fidelidade, seguido da avaliação dos usuários que gerou melhoria. Baseado no ciclo apresentado em Winckler e Pimenta (2002).

## 1.5 Organização da dissertação

Esta dissertação está organizada em 6 Capítulos. O Capítulo 1 introduz esta pesquisa, com a motivação para sua realização, os objetivos propostos e a metodologia. O Capítulo 2 contém a revisão de literatura. Para uma melhor compreensão dos temas pertinentes a esta dissertação, a revisão foi dividida em duas partes:

- Parte 1 – Revisão dos aspectos referentes ao desenvolvimento da interface Web: é a primeira parte da revisão e aborda conceitos relacionados à usabilidade e ao desenvolvimento de interfaces Web.

- Parte 2 – Revisão dos conceitos relacionados a compreensão do método CReF: traz o estado da arte para o entendimento teórico do método CReF.

Os trabalhos relacionados com a proposta em questão estão dispostos no Capítulo 3.

O Capítulo 4 apresenta os resultados desta pesquisa. Inicialmente descrevemos os resultados das avaliações heurísticas realizadas pelos especialistas. A partir dos problemas encontrados é definido e apresentado o documento de requisitos de software do wCReF, que dá origem a um protótipo de alta fidelidade da interface. O protótipo foi avaliado por um grupo de usuários finais, através de questionários. Com estes resultados, a interface foi melhorada, dando origem a interface do wCReF, apresentada no Capítulo 5. Por fim, no Capítulo 6 são feitas as considerações finais, com as principais contribuições da dissertação, seguido dos trabalhos futuros. No final dessa dissertação encontra-se as referências utilizadas para escrita deste volume e o material suplementar (Apêndices).

## 2 REVISÃO DE LITERATURA

Este capítulo descreve os principais conceitos referentes a este trabalho de mestrado e o referencial teórico envolvido com o método CReF. Para distribuir o conteúdo de acordo com a área relacionada, dividimos esta revisão em duas partes:

- Parte 1 – Revisão dos aspectos referentes ao desenvolvimento da interface Web: é a primeira parte da revisão e aborda conceitos relacionados à usabilidade e ao desenvolvimento de interfaces Web. Iniciamos com a Interação Humano-Computador, seguido da definição do termo interface e a relação entre interface de usuário, usabilidade e acessibilidade. Para finalizar essa sessão, definimos a avaliação de interfaces, que é uma das metodologias principais deste estudo.
- Parte 2 – Revisão dos conceitos relacionados a compreensão do método CReF: A segunda parte é relativa aos temas referentes a compreensão sobre o método CReF e traz a definição da Bioinformática, seguido da predição de estruturas de proteínas, englobando o problema PSP (*Protein Structure Prediction*), o evento mundial de avaliação de estruturas de proteínas (CASP) e os servidores de predição.

### 2.1 Parte 1 – Revisão dos aspectos referentes ao desenvolvimento da interface Web

Buscamos contemplar aqui a revisão sobre os principais aspectos de desenvolvimento de interfaces Web. Para isto, primeiramente na seção 2.1.1 são apresentados alguns fundamentos sobre a Interação Humano Computador e interfaces (seção 2.1.2). Após, explicamos os conceitos de Interface de usuário, usabilidade e acessibilidade (2.2.3). A seção 2.1.4 aborda sobre o desenvolvimento de interfaces Web descrevendo um ciclo de desenvolvimento de aplicações Web, explicando as características de cada etapa do ciclo. Por fim a avaliação de usabilidade na seção 2.1.6 juntamente com a Avaliação Heurística e avaliação por questionários, o que finaliza a primeira parte desta revisão.

#### 2.1.1 Interação Humano-Computador

O estudo de como os sentidos e as capacidades motoras permitem às pessoas utilizarem máquinas e ferramentas complexas já era objeto de estudo da disciplina Fatores humanos, como é chamada nos Estados Unidos da América, ou Ergonomia, na Europa, durante a segunda guerra mundial. Estas disciplinas, historicamente, abordam o desempenho das pessoas no uso de máquinas em geral. Como computadores são máquinas destinadas ao processamento e uso de informações, tornou-se necessário estender esta disciplina para se estudar também a capacidade



mental que possibilita às pessoas produzirem informações processáveis por computadores, recuperá-las e compreendê-las. Esta extensão surgiu nos anos 80 com rótulo de Interação entre Seres Humanos e Sistema Computacionais (Leite, 1998).

Interação Humano-Computador (IHC) é uma área que não envolve apenas ergonomia, análise de sistemas e engenharia de software, disciplinas mais tradicionalmente associadas ao desenvolvimento de sistemas computacionais, mas também disciplinas que estudam o comportamento e a capacidade humana, tais como as disciplinas da psicologia e ciências cognitivas, bem como disciplinas que abordem os fatores sociais deste comportamento (Leite, 1998).

A IHC se preocupa com o design, a avaliação e implementação de sistemas de computacionais interativos para uso humano e com o estudo dos principais fenômenos que os cercam (Preece et al., 2005). Tem por objetivo principal fornecer aos pesquisadores e desenvolvedores de sistemas explicações e previsões para fenômenos de interação usuário-sistema e resultados práticos para o *design* da interface de usuário (Leite, 1998).

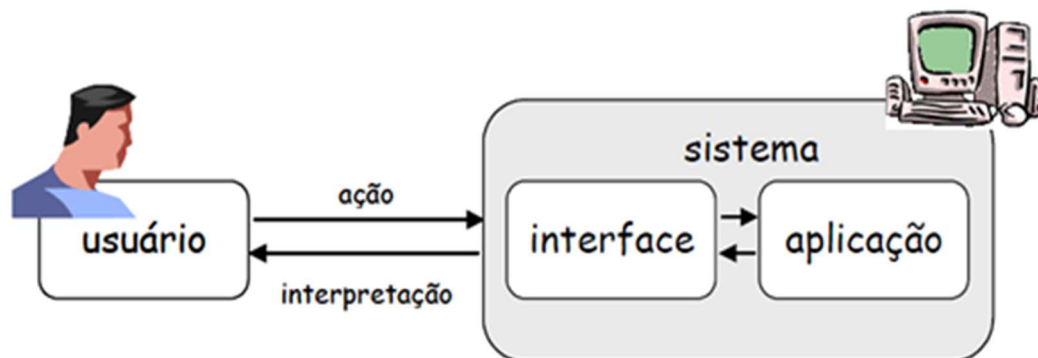
Dessa forma com a evolução da tecnologia, vê-se a disseminação do uso de sistemas interativos nas mais diversas áreas e por variados perfis de usuários. Esta diversidade de perfis contribui para a necessidade de uma maior facilidade de interação entre os usuários e os diferentes sistemas interativos que estes precisam utilizar. E, neste âmbito que surge a IHC (Preece et al., 1994). Portanto, a área de IHC desponta com esta evolução, com a necessidade da criação de sistemas que apoiem as pessoas que os utilizam.

### 2.1.2 Interface

A partir dos anos 1990, com o surgimento do fenômeno Web, os conceitos de usabilidade que até então eram voltados apenas ao desenvolvimento de aplicativos de software em ambientes desktop, passaram por um processo de evolução. Desde então, usuários com os mais variados níveis sociais, profissionais e educacionais passaram a utilizar com maior frequência as interfaces Web (Campos e Matias, 2012).

Considerando a interação como um processo de comunicação, interface pode ser vista como o sistema de comunicação utilizado neste processo (Figura 2) (Prates e Barbosa, 2003). Uma definição usada frequentemente para interfaces é a proposta por Moran:

“A interface de usuário deve ser entendida como sendo a parte de um sistema computacional com a qual uma pessoa entra em contato - física, perceptiva ou conceitualmente” (Moran, 1981).



**Figura 2.** O processo de interação humano-computador (Prates e Barbosa, 2003).

Esta definição de Moran caracteriza uma perspectiva para a interface de usuário como tendo um componente físico, que o usuário percebe e manipula, e outro conceitual, que o usuário interpreta, processa e raciocina. A interface é tanto um meio para a interação usuário-sistema, quanto uma ferramenta que oferece os instrumentos para este processo comunicativo. Desta forma, a interface é um sistema de comunicação (Leite, 1998). Quando se considera a aplicação como máquina (s) virtual (is), a interface pode ser considerada ainda como um ambiente virtual para ações. Ela é composta por uma coleção de dispositivos através dos quais o usuário pode trocar informações com o sistema (através de estruturas de interações tais como menus, janelas, ícones, linguagens de comandos, formulários, perguntas e respostas em linguagem natural) (Leite, 1998).

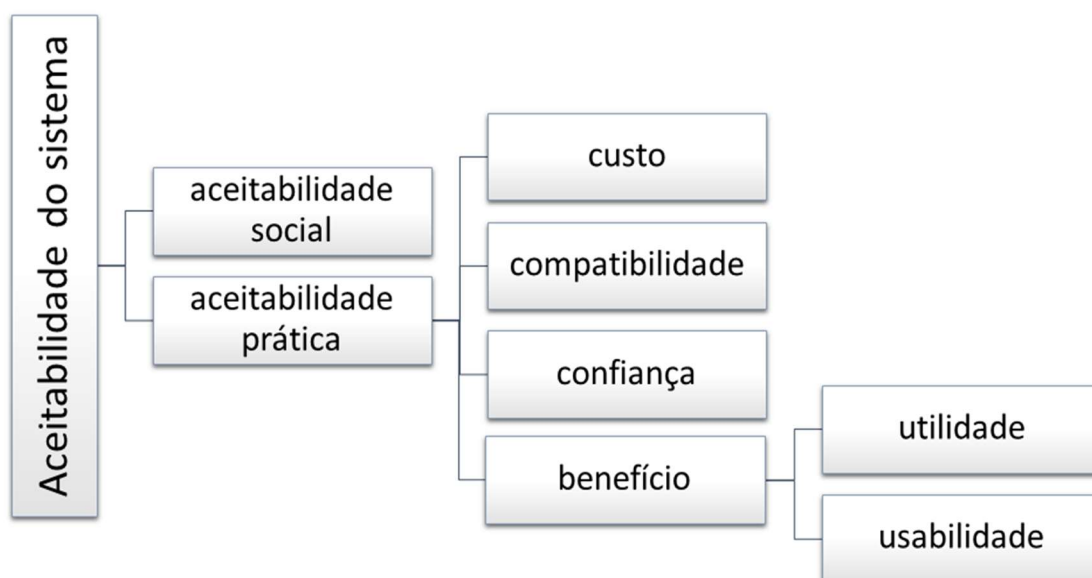
A interface do usuário de um programa de computador é a parte que exibe ao usuário janelas, menus, ícones, etc., o que o programa faz, como ele pode utilizá-lo, quais as informações podem ser solicitadas, como devem ser fornecidos os dados que o programa necessita, etc., aceitando as entradas fornecidas pelo usuário, analisando-as, estabelecendo diálogos e exibindo objetos da aplicação ou da interação com o usuário (Orth, 2005). Pelo exposto acima, podemos definir interface como um sistema de comunicação que possui dois componentes, um componente físico, composto por hardware e software, no qual o usuário percebe e manipula, e outro conceitual, onde interpreta, processa e raciocina (Orth, 2005).

### 2.1.3 Interfaces de usuário e usabilidade

Um dos fatores mais críticos na avaliação de interfaces é a definição dos critérios de avaliação (Tavares e Cavalcanti, 2001). Um dos critérios mais comumente adotado é a usabilidade (Nielsen, 1994b).

As características de qualidade de uso são: usabilidade, experiência do usuário, acessibilidade e comunicabilidade. Dessa forma, compreender esses princípios e aplicá-los na fase do desenvolvimento de novas aplicações é útil para a construção de ambientes interativos, garantindo projetos interessantes, eficientes e fáceis de serem utilizados (Winckler, 2002).

Usabilidade teve suas origens acadêmicas na psicologia e ergonomia. O que faz usabilidade se diferenciar das demais é que ela se concentra nas questões humanas, complementando outros objetivos tais como a funcionalidade, eficiência (ou seja, velocidade de execução) e confiabilidade. Neste sentido é que a usabilidade está intimamente relacionada com a facilidade de uso (Figura 3), que é provavelmente a forma mais comum da utilização deste termo (Bevan, 1995). Ela é um fator que se aninha no interior da aceitabilidade de um sistema, dentro da aceitabilidade prática, dos benefícios para o usuário, estando junto a utilidade do sistema (Nielsen, 1994a).



**Figura 3.** Usabilidade como facilidade de uso (Nielsen, 1994a).

As normas da *International Organization Standardization* (ISO) trazem várias definições para o termo usabilidade. A usabilidade é definida pela ISSO/IEC (ISO/IEC, 1991) como um conjunto de atributos relacionados com o esforço necessário para o uso de um sistema interativo, e relacionado com a avaliação individual de tal uso, por um conjunto específico de

usuários. Já a norma ISO 9241 (2008) define usabilidade como sendo a medida na qual um produto pode ser usado por um determinado perfil de usuário para alcançar objetivos com eficácia, eficiência e satisfação em um contexto específico de uso. É um fator importante pois interfaces com usabilidade aumentam a produtividade dos usuários, diminuem as ocorrências de erros (ou a sua importância) e, não menos importante, contribuem para a satisfação dos usuários.

Segundo Nielsen (Nielsen, 1994b) usabilidade é um conjunto de propriedades de uma interface que reúne os seguintes atributos: fácil aprendizado; eficiência; capacidade de memorização; baixo índice de erros; satisfação e prazer ao uso. Já Preece, Rogers e Sharp (Preece et al., 2005) descrevem a usabilidade como um fator que assegura que os produtos sejam fáceis de usar, eficientes e agradáveis.

Usabilidade também é o termo técnico usado para descrever a qualidade de uso de uma interface (Bevan, 1995). Quando a usabilidade é levada em conta durante o processo de desenvolvimento de interfaces Web, vários problemas podem ser eliminados como, por exemplo, pode-se reduzir o tempo de acesso à informação, tornar informações facilmente disponíveis aos usuários e evitar a frustração de não encontrar informações no site (Winckler e Pimenta, 2002). Em longo prazo, o benefício é uma interface fácil de usar, com menor custo de produção e mais fácil de se manter (Pavelin et al., 2012).

Então a usabilidade é pré-requisito a ser observado quando se fala em desenvolver qualquer nova aplicação. Ela refere-se à facilidade com que os usuários realizam determinadas tarefas quando interagem com uma ferramenta ou objeto através de sua interface (Nielsen, 1994a) e é um aspecto fundamental da qualidade de uma aplicação interativa (Bolchini et al., 2009).

Mirel e Wright (Mirel e Wright, 2009) afirmam que assegurar a utilidade e usabilidade é um desafio constante e particularmente complicado em qualquer projeto de software na pesquisa de bioinformática e no contexto do desenvolvimento. Mas apesar da usabilidade ser um aspecto crucial para o desenvolvimento de interfaces, pouco tem sido feito para analisar as características de *design* dos recursos Web em bioinformática, que podem levar a potenciais problemas de usabilidade. Identificar os problemas de usabilidade na sua origem ajuda a prevenir o aparecimento de problemas nas aplicações atuais e futuras (Bolchini et al., 2009).

Considera-se que a interface tem um problema de usabilidade se um determinado usuário ou um grupo de usuários encontram dificuldades para realizar uma tarefa com a interface. Tais dificuldades podem ter origens variadas e ocasionar diversos problemas, como a perda de dados, a diminuição da produtividade e mesmo a total rejeição do software por parte dos

usuários (Winckler, 2002). Consequentemente, as barreiras de usabilidade podem representar obstáculos significativos para a experiência satisfatória do usuário e forçar os pesquisadores a gastar tempo e esforço desnecessários para concluir suas tarefas (Bolchini et al., 2009).

#### 2.1.4 O Desenvolvimento de interfaces Web

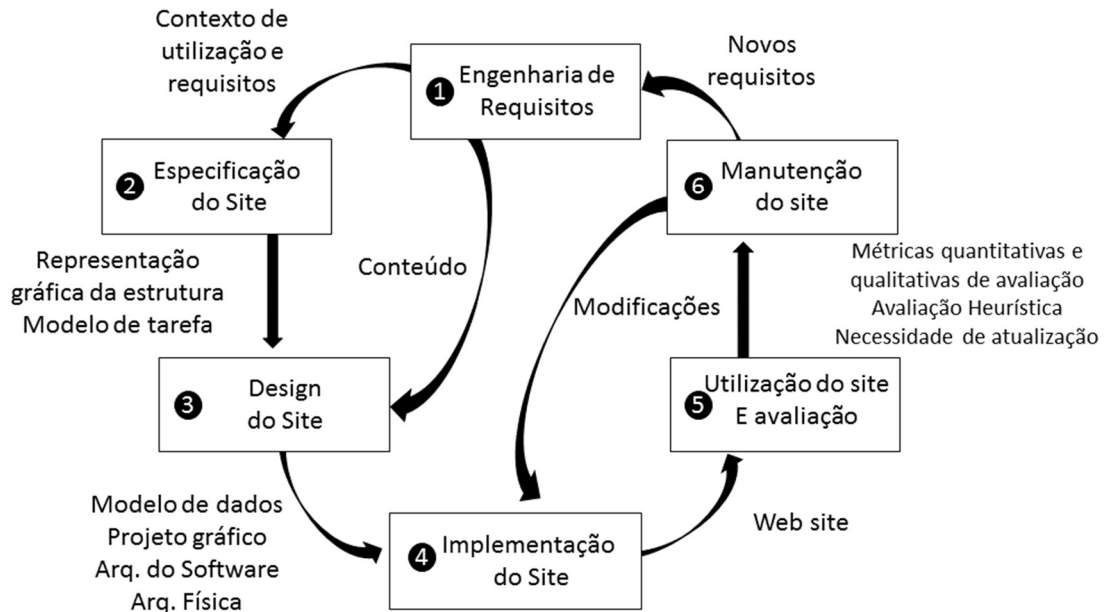
Desde meados da década de 1980, as normas formais e diretrizes publicadas para interfaces de usuário e a IHC têm aumentado em importância à medida que o uso de computadores se tornou difundido em ambientes de trabalho e outros.

Um dos desafios primordiais no desenvolvimento de interfaces Web (interfaces de rede) é melhorar a usabilidade e a acessibilidade. Em busca da melhoria significativa no desenvolvimento dessas interfaces é que os desenvolvedores vêm há algum tempo aplicando o conceito de usabilidade em seu produto, considerando a crescente complexidade das aplicações (Campos e Matias, 2012). Dessa forma, cada vez mais desenvolvedores têm procurado, durante o ciclo de desenvolvimento de suas ferramentas, aplicar conceitos e metodologias que busquem garantir que seu produto final possua uma boa usabilidade, garantindo a satisfação de seus usuários.

Portanto, do ponto de vista do profissional de usabilidade, o processo de desenvolvimento deve possuir as seguintes sete fases (Sears e Lund, 1997):

- 1) Definição da aplicação, conforme o ponto de vista de seus usuários, seus objetivos e suas tarefas.
- 2) O desenho da aplicação e sua interface (as definições).
- 3) Implementação do Protótipo.
- 4) A realização de testes com o protótipo, com a participação dos usuários para identificar falhas de projeto que impeçam os utilizadores de alcançar seus objetivos.
- 5) A correção das falhas do projeto. Aqui se deve redesenhar e reprototipar até que um haja um teste aceitável, onde um nível usabilidade é alcançado.
- 6) Desenvolver a aplicação e fazer um teste definitivo, em seguida, projetar o suporte e a documentação.
- 7) Entregar a aplicação final e avaliar o sucesso do projeto em uso no campo.

Com relação a criação de aplicações Web, Scapin (Scapin et al., 2001) propõe um ciclo para o desenvolvimento de suas interfaces (Figura 4).



**Figura 4.** Ciclo de desenvolvimento de aplicações Web. O ciclo de desenvolvimento de interfaces Web é uma espiral contínua com seis etapas. Scapin (Scapin et al., 2001)

O ciclo de desenvolvimento de interfaces Web é uma espiral contínua, notadamente marcado por sucessivas modificações, que são muito mais frequentes em aplicações Web que em outros tipos de interfaces (Winckler et al., 2001). Dentro deste ciclo espiral, várias etapas se sucedem. O número e a importância de cada uma das etapas variam em função da abordagem utilizada.

- Etapa 1 - Engenharia de requisitos: a estrutura do site e o contexto de utilização são identificados;
- Etapa 2 - Especificação do site: modelos da interface são construídos a partir dos requisitos obtidos durante a análise de requisitos;
- Etapa 3 - *Design* do site: modelos são refinados e o seu conteúdo é definido;
- Etapa 4 - Implementação do site: corresponde a criação de páginas HTML e objetos de som/imagem necessários à aplicação;
- Etapa 5 - Utilização do site e avaliação: são avaliadas a usabilidade da interface e a coerência da interface com relação aos requisitos iniciais;
- Etapa 6 - Manutenção do site: envolve um ciclo de maior duração que envolve a coleta de novos requisitos e planejamento das modificações identificadas durante a etapa de avaliação.

De fato, atualmente é amplamente reconhecido pela comunidade de IHC que é preferível dedicar mais esforço para desenvolver aplicações, de modo que critérios de usabilidade sejam considerados desde o início e durante todo o seu processo de construção, do que uma fase de avaliação a posteriori ao fim deste processo. No entanto, vale a pena reforçar que a avaliação continua a ser essencial sobretudo para validar as escolhas da concepção e confirmar o nível de satisfação dos usuários e deve ser sempre realizada por melhor que seja o processo de concepção que a antecede (Winckler et al., 2001).

#### 2.1.4.1 Usabilidade na Web

A usabilidade é definida pela ISO 9241 (ISO 9241, 2008) como sendo a medida na qual um produto pode ser usado por um determinado perfil de usuário para alcançar objetivos com eficácia, eficiência e satisfação em um contexto específico de uso. Segundo as definições dadas no documento, a eficácia diz respeito a acurácia e completude com as quais usuários alcancem objetivos específicos. A eficiência representa os recursos gastos em relação à acurácia e abrangências com as quais usuários atingem objetivos. Já a satisfação, relaciona o conforto e presença de atitudes positivas para com o uso de um produto, e por fim o contexto de uso que compreende os usuários, tarefas, equipamentos, e o ambiente físico e social no qual um produto será usado.

Segundo a *Associação Brasileira de Normas Técnicas* (ABNT) usabilidade é a medida na qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso (NBR, 2002). Na perspectiva cognitiva, a usabilidade é a vista como sendo a qualidade da interface de proporcionar o menor esforço cognitivo para o usuário interagir com o sistema (Leite, 1998). A usabilidade ajuda o usuário a lidar com determinadas dificuldades em um domínio (Oliveira e Lima, 2013).

Referente à usabilidade Pontes (Pontes, 2008) afirma que:

O processo de interação entre sistemas interativos e seres humanos, devido ao fato de envolver participantes de natureza distinta, requer um estudo multidisciplinar que elabore as necessidades para o desenvolvimento dos sistemas. Mais que desenvolver um sistema, o desafio da usabilidade requer o desenvolvimento da sua aplicação integrada ao ambiente do usuário de maneira a estender sua capacidade.

Para se ter uma ideia dos elementos que devem ser considerados em relação a usabilidade, são apresentados alguns atributos importantes de usabilidade (Nielsen, 1994a):

- Facilidade de aprender

- Eficiência de uso
- Facilidade de lembrar
- Facilidade de usar
- Rapidez com que se conseguir atingir o objetivo
- Flexibilidade
- Poucos erros e não catastróficos
- Satisfação subjetiva

#### 2.1.4.2 Acessibilidade na Web

As novas tecnologias da informação e da comunicação devem ter por objetivo tornar os recursos computacionais mais acessíveis a um conjunto diversificado de atores sociais. A acessibilidade passa a ser entendida como sinônimo de aproximação, um meio de disponibilizar a cada usuário interfaces que respeitem suas necessidades e preferências, e de potencializar a construção de um projeto emancipatório que traga em sua essência a ruptura com um modelo de sociedade que fixa limites, subordina e exclui grupos de homens e mulheres dos coletivos inteligentes (Conforto e Santarosa, 2002).

O conceito de acessibilidade surge ligado a questões físicas relativas a facilidades de acesso (barreiras arquitetônicas). Entre as décadas de 1940 e 1960, o termo Acessibilidade tem uma aplicação direta com questões físicas e funcionais e à reabilitação e posteriormente, é transferido para a informática na questão de acesso (Passerino e Montardo, 2007).

Historicamente a partir da década de 1980, impulsionado pelo *Ano Internacional das Pessoas Deficientes* (1981), que a questão da acessibilidade e eliminação de barreiras arquitetônicas ganham destaque internacional e transforma-se em metas para todos os países desenvolvidos e em vias de desenvolvimento. Nesse período, surge também o conceito de Desenho Universal na concepção de um *design* adaptável às diversas necessidades da população (Passerino e Montardo, 2007).

A Norma Brasileira NBR 9050-1994 adota a seguinte definição de acessibilidade: "*Possibilidade e condição de alcance para utilização, com segurança e autonomia, de edificações, espaço, mobiliário e equipamentos urbanos*". Dessa forma, um objeto acessível é aquele que pode ser alcançado para uso. Uma vez alcançado, supõe-se que seu uso, seguro e autônomo, não seja mais uma questão de acessibilidade, e sim de usabilidade. A definição adotada pela Organização Nações Unidas (ONU) sobre acessibilidade diz que acessibilidade é "*possibilidade de acesso, a que se pode chegar facilmente; que fica ao alcance, o processo de conseguir a igualdade de oportunidades em todas as esferas da sociedade*" (ONU, 2006).



Sendo assim acessibilidade também é definida como *facilidade na aproximação, no trato ou na obtenção* (Ferreira et al., 2010). No contexto de software tem sido traduzida em diversas formas de tratar necessidades especiais do usuário. Dispositivos especiais têm sido incorporados ao *hardware* ou *software* facilitando o acesso de todos e têm sido criadas aplicações especiais para alguns tipos de deficiências do usuário. A meta é: todo mundo deve ter condições de usar o ambiente de maneira independente e igual (Nicholl e Filho, 2001).

A W3C (*World Wide Web Consortium*) (W3C, 2013) possui as seguintes recomendações para acessibilidade:

- WAI - *Web Accessibility Initiative*: Iniciativa do W3C que trabalha em conjunto com organizações de todo o mundo desenvolvendo estratégias, orientações e recursos para ajudar a tornar a Web acessível às pessoas com deficiência.
- WCAG - *Web Content Accessibility Guidelines*: São as Recomendações para a acessibilidade do conteúdo da Web, documentos que explicam como tornar o conteúdo Web acessível a pessoas com deficiências, destinando-se a todos os criadores de conteúdo Web (autores de páginas e projetistas de sites) e aos programadores de ferramentas para criação de conteúdo. Possui duas versões: 1.0: 5 de maio de 1999 e versão 2.0: 11 de dezembro de 2008.

### 2.1.5 A avaliação de usabilidade

A qualidade da interface é fundamental para que sistemas Web possam ser utilizados com sucesso. Para se obter interfaces de alta qualidade é essencial que estas sejam avaliadas durante o processo de design, permitindo assim a identificação e ajustes de problemas de interação (Prates e Barbosa, 2003). A avaliação é uma etapa fundamental no desenvolvimento de *softwares* e, especialmente, no processo de *design* de interfaces (Tavares e Cavalcanti, 2001). Sendo assim, parte do desenvolvimento de qualquer sistema de software que atenda aos padrões de IHC deve possuir uma avaliação de usabilidade (Reed et al., 1999).

A avaliação de usabilidade é um passo importante no desenvolvimento de interfaces interativas de qualidade e desenvolvidas de acordo com seus usuários. Esta etapa é muito importante no desenvolvimento de projeto de um sistema, pois ela permite identificar previamente problemas de usabilidade, evitando assim, transtornos posteriores aos seus usuários (Orth, 2005).

Segundo Nielsen (Nielsen, 1994b) a usabilidade interfere diretamente nos aspectos com os quais o usuário interage com um sistema, e deve ser medida levando em consideração determinados usuários executando determinadas tarefas. Neste sentido que a Tabela 1 mostra

os principais fatores que se devem ser medidos e avaliados na aplicação da usabilidade em sistemas:

**Tabela 1.** Fatores pertinentes na aplicação de usabilidade em sistemas:

Autor:	Princípios que privilegiem
<i>World Wide Web Consortium (W3C)</i> (World Wide Web Consortium, 1999)	Consistência na interface privilegiando a lógica em detrimento à memorização; familiaridade do utilizador com o ambiente; surpresa mínima, ao evitar reações diferentes para ações iguais; facilidade de recuperação em caso de necessidade de reversão a algum estado anterior; confirmações em caso de ações destrutivas ou irreversíveis; <i>checkpointing</i> , onde o utilizador possui versões anteriores para caso tenha necessidade de interromper a navegação que por força de uso ou necessidade própria.
Nielsen (Nielsen, 1994a)	O autor considera cinco tarefas para auxiliar a avaliação e medição da usabilidade. São elas: aprendizagem; eficiência no aprendizado; memorização; erros e satisfação do usuário na utilização do sistema
Netto (Netto, 2004)	Facilidade de aprendizado do sistema; Facilidade de uso; Satisfação do usuário; Flexibilidade; Produtividade; Clareza na arquitetura da informação; Simplicidade; Relevância do conteúdo; Consistência; Utilidade percebida do produto; Adequação a tarefa; Características das tarefas e; Características dos usuários.

Os métodos que estão disponíveis para avaliar a usabilidade são classificados, primeiramente, como métodos de inspeção de usabilidade e testes empíricos com usuários. Métodos de inspeção tem como característica o emprego de especialistas em interfaces, que a utilizam em busca de possíveis problemas de usabilidade (Avaliação Heurística por exemplo). Métodos que contam com a participação de usuários caracterizam-se pela utilização de questionários ou observação direta ou indireta dos usuários durante a utilização da interface (Orth, 2005).

Nesta revisão apresentamos os dois métodos de avaliação de usabilidade que julgamos serem pertinentes ao desenvolvimento desta pesquisa. Um método de inspeção, realizado diretamente com especialistas, a Avaliação Heurística. E outro realizado diretamente com os usuários finais, através de questionários.

### 2.1.5.1 Avaliação Heurística

A Avaliação Heurística (Nielsen, 1995, 1994b; Nielsen e Molich, 1990) é um método de avaliação de usabilidade que busca encontrar problemas de usabilidade numa interface de usuário, como parte de um processo de *design* interativo. Esse tipo de avaliação envolve um pequeno conjunto de inspetores, que irão examinar a interface, avaliando sua conformidade com os princípios de usabilidade reconhecidos (os "heurísticos"). Nielsen (Nielsen, 1995) afirma que o melhor custo/benefício é alcançado quando se utilizam entre 3 e 5 avaliadores.

Quando um problema é encontrado nestas interfaces, significa que o usuário não pode alcançar seu objetivo de forma eficiente, eficaz e satisfatória (Cuperschmid e Hildebrand, 2013). Uma vez identificado, o problema pode ser solucionado ou, ao menos, seus efeitos podem ser minimizados (Winckler e Pimenta, 2002). Para esse fim, Nielsen (Nielsen, 1995) definiu 10 Heurísticas, que são apresentadas na Tabela 2.

**Tabela 2.** Conjunto de Heurísticas de Nielsen

<b>1.</b>	<b>Visibilidade do status do sistema:</b>
O sistema deve sempre manter os usuários informados sobre o que está acontecendo, através de feedback apropriado, em um tempo razoável.	
<b>2.</b>	<b>Compatibilidade entre sistema e mundo real:</b>
O sistema deve utilizar a linguagem do usuário, com palavras, frases e conceitos familiares para ele, ao invés de termos específicos de sistemas. Seguir convenções do mundo real, fazendo com que a informação apareça em uma ordem lógica e natural.	
<b>3.</b>	<b>Controle e liberdade para o usuário:</b>
Os usuários podem escolher funções do sistema por engano e então necessitam de "uma saída de emergência", claramente definida para sair do estado não desejado, sem ter que percorrer um longo diálogo, ou seja, é necessário suporte a <i>undo</i> e <i>redo</i> ( <i>desfazer e refazer</i> ).	
<b>4.</b>	<b>Consistência e padrões:</b>
Referem-se ao fato de que os usuários não deveriam ter acesso a diferentes situações, palavras ou ações representando a mesma coisa. A interface deve ter convenções não ambíguas, seguindo as convenções da plataforma.	
<b>5.</b>	<b>Prevenção de erros:</b>
Ainda melhor do que boas mensagens de erro é um projeto cuidadoso que impede que um problema ocorra em primeiro lugar. O sistema deve eliminar as condições passíveis de erros ou verificá-los, e apresentar aos usuários uma opção de confirmação, antes de se comprometer com	

a ação. Os erros são as principais fontes de frustração, ineficiência e ineficácia durante a utilização do sistema.

---

**6. Reconhecimento em lugar de lembrança:**

---

Minimizar a carga de memória do usuário, fazendo objetos, ações e opções visíveis. O usuário não deve ter de lembrar informações de uma parte do diálogo para outra. Instruções para o uso do sistema devem estar visíveis ou facilmente acessíveis.

---

**7. Flexibilidade e eficiência de uso:**

---

O sistema deve ser adequado tanto para usuários inexperientes quanto para usuários experientes. Deve permitir aos usuários personalizar ações frequentes, como por exemplo, a utilização de aceleradores - invisíveis pelo usuário novato - podem frequentemente acelerar a interação para o usuário experiente.

---

**8. Design estético e minimalista:**

---

Os diálogos não devem conter informações irrelevantes ou que não sejam necessárias. Cada unidade extra de informação em um diálogo acaba competindo com as informações relevantes e diminui a visibilidade das informações importantes para os usuários.

---

**9. Auxiliar os usuários a reconhecer, diagnosticar e recuperar erros:**

---

Mensagens de erro devem ser expressas em linguagem natural/clara (sem códigos), indicando precisamente o erro/problema e sugerir uma solução.

---

**10. Ajuda e documentação:**

---

Mesmo que seja melhor que o sistema possa ser usado sem documentação, pode ser necessário fornecer ajuda e documentação. Tais informações devem ser fáceis de encontrar, centradas na tarefa do usuário, listar passos concretos a serem realizados e não serem muito grande. A ajuda deve estar facilmente acessível e on-line.

(Fonte: Nielsen, 1994, 1995)

Este método de avaliação é baseado no julgamento do avaliador e, normalmente, descobre-se por meio dele 75% dos problemas de usabilidade (Campos e Matias, 2012). Deste modo, cada avaliador interage, individualmente, com a interface e julga a sua adequação comparando-a com princípios de usabilidade reconhecidos, as heurísticas. No final um observador reunirá todas as informações coletadas em um só documento para efetiva identificação e denominação dos problemas (Campos e Matias, 2012).

#### 2.1.5.2 Avaliação por questionários

Questionários são ferramentas muito úteis na avaliação da interação entre o usuário e a interface. São utilizados para coletar informações subjetivas sobre dados, como o perfil dos

usuários, a qualidade da interface e quais problemas são encontrados no seu uso. Estas informações são tão importantes quanto a performance no uso do sistema, e não podem ser obtidas de outra forma senão perguntando aos usuários (Winckler e Pimenta, 2002).

Seu uso é importante para captar a opinião do usuário referente ao sistema e obter um feedback sobre a aplicação que foi desenvolvida. Além disso, é importante que os usuários estejam envolvidos no processo de *design* para melhorar o mapeamento de seus objetivos para o projeto (Sears e Lund, 1997). Uma vantagem é que o uso de questionários dá ao avaliador a possibilidade de aplicar vários testes ao mesmo tempo em locais diferentes.

Dessa forma, questionários podem ser úteis de diferentes maneiras dentro do desenvolvimento de interfaces Web como, por exemplo, para (Winckler e Pimenta, 2002):

- Identificar o perfil dos usuários. O objetivo deste tipo de questionário é basicamente coletar informações sobre os usuários. Tais informações podem ser de origem funcional, pessoal, sobre preferências ou mesmo sobre a utilização de computadores e sistemas. Obviamente outras questões podem ser adicionadas para fins específicos de avaliação.
- Determinar o grau de satisfação dos usuários com relação a interface. Questionários específicos para descobrir a satisfação de usuários vêm sendo pesquisados desde a década de 80.
- Estruturar informações sobre problemas de usabilidade identificados por usuários, na forma de questionários para relato de incidentes críticos.

## **2.2 Parte 2 – Revisão dos conceitos relacionados a compreensão do Método CReF**

### **2.2.1 A Bioinformática**

Os avanços na área da Biologia Molecular trouxeram modificações nos paradigmas biológicos, onde os cientistas sentiram a crescente necessidade de desenvolver programas computacionais que permitissem agrupar, analisar, organizar e relacionar a informação biológica. Assim, surgiu a Bioinformática, que conceitua a biologia em termos de moléculas, no sentido físico-químico, e, por meio de técnicas computacionais (empregando estatística, matemática e ciência da computação) permite armazenar, organizar e compreender, em larga escala, a explosão de dados biológicos (Luscombe et al., 2001).

Consolida-se como uma nova área do conhecimento, que traz uma grande quantidade de dados e ferramentas para manipulação dos mesmos, com o objetivo de: reconhecer as sequências de genes; realizar a predição tridimensional de proteínas; identificar inibidores de

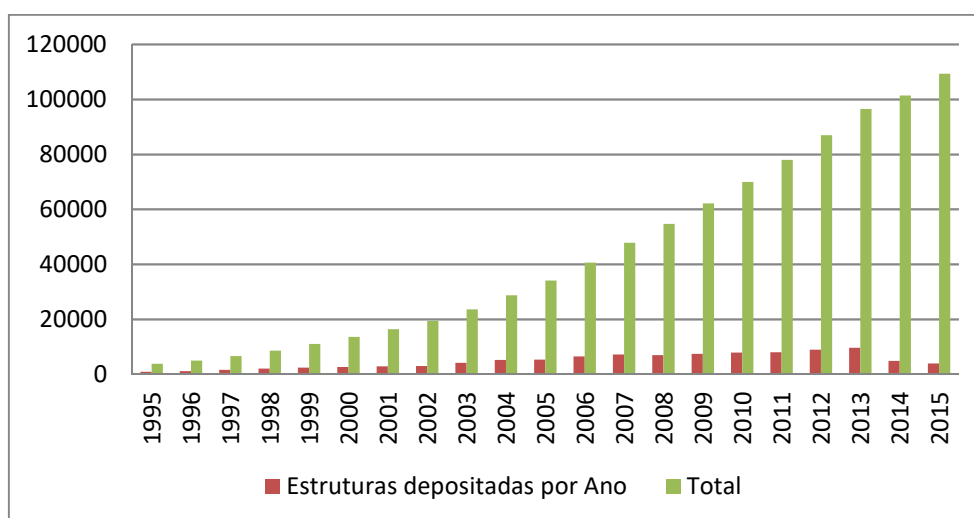
enzimas; organizar e relacionar informação biológica; agrupar proteínas homólogas; estabelecer árvores filogenéticas; analisar experimentos de expressão gênica, entre outras (Griffith, Cattley, Lesk apud Araújo et al., 2008).

Um dos grandes desafios da bioinformática começa a aparecer na era pós-genômica, na qual a proteômica se torna um dos principais alvos de estudo juntamente com o entendimento estrutural e funcional de proteínas (Silveira, 2005). O proteoma em analogia ao genoma é o conjunto de proteínas de um organismo. A proteômica combina identificação, distribuição, interações, dinâmica e padrões de expressão das proteínas de sistemas vivos. É um assunto que envolve um grande volume de informações (Lesk, 2008).

Levando em consideração o aumento significativo desses dados surgiu uma subdisciplina da bioinformática: a Bioinformática Estrutural (BE) (Silveira, 2005). BE é a conceituação da biologia em termos de moléculas, no sentido físico-químico, e a aplicação de técnicas de informática (derivadas de disciplinas como: matemática, ciência da computação e estatística) para entender, organizar e explorar a informação estrutural associada a essas moléculas em grande escala (Dall'Agno, 2012; Luscombe et al., 2001).

Neste sentido que em 1971, para poder armazenar e disponibilizar a quantidade de dados dentro da BE, foi criado o *Protein Data Bank* (PDB) (Berman et al., 2013, 2000), um repositório público de modelos tridimensionais de macromoléculas biológicas.

Desde então, o número, o tamanho e complexidade de modelos estruturais depositados neste banco de dados continuaram a aumentar, o que reflete o crescimento da área da Bioinformática Estrutural (Berman et al., 2013) culminando em mais de 100.000 estruturas depositadas em 2014, chegando em 109.093 estruturas em maio de 2015 (Figura 5).



**Figura 5.** Entradas do PDB (Protein Data Bank) lançadas por ano e o total depositadas por ano. Podemos verificar então o alto crescimento de estruturas depositadas ao longo dos anos. Em 1995 o PDB possuía

3812 estruturas. Dez anos após, no ano 2005 já eram 34.066, e em 2014 ultrapassou as 100.000 estruturas, contabilizando maio de 2015 um total de 109.093 (PDB - Protein Data Bank, 2015).

### 2.2.2 Predição de Estruturas Tridimensionais (3D) de Proteínas

A determinação da estrutura de macromoléculas é uma potente ferramenta para compreender os sistemas biológicos e periodicamente produz resultados marcantes que afetam a comunidade científica em geral (Seringhaus e Gerstein, 2007). A exemplo disso, 2014 foi marcado pelo 52º aniversário do prêmio Nobel, de 1962 em Química, atribuído a Max Perutz (Perutz et al., 1960) e John Kendrew (Kendrew et al., 1958) por seu trabalho pioneiro na determinação da estrutura de proteínas globulares (Figura 6).



**Figura 6.** Imagem obtida de Kendrew e colaboradores publicaram a primeira estrutura de uma proteína globular: Mioglobina em 6.0 Å de resolução (Kendrew et al, 1958). Sua estrutura não possuía a simetria e regularidade esperada. Imagem retirada do artigo de Dill e MacCallum (Dill e MacCallum, 2012).

Desde então, quase um em cada quatro prêmios Nobel de química foram relacionados à determinação de estruturas de macromoléculas, e, na última década, a metade lidaram com trabalhos provenientes. (Seringhaus e Gerstein, 2007).

É nesta perspectiva que historicamente podemos citar alguns trabalhos relativos à questão, que merecem destaque: a) os modelos teóricos postulados por Pauling, Corey e Branson (Pauling et al., 1951); b) o modelo chave-fechadura introduzido por Fischer (Fischer, 1894); c) o primeiro modelo de uma estrutura cristalizada a partir de uma proteína globular, já citado anteriormente (Kendrew et al., 1960, 1958) d) os estudos que mostram a possibilidade de renaturação de uma proteína ao seu estado funcional, partindo de seu estado desnaturado

(Anfinsen, 1973; Anson e Mirsky, 1925); Todos foram citados no trabalho de Torrieri (Torrieri, 2010).

Atualmente, a abordagem mais eficaz na determinação da estrutura 3D de proteínas é aquela que se utiliza de técnicas experimentais como Cristalografia de Raio X (*X-Ray Crystallography*) (Barlow, 1883) ou Ressonância Magnética Nuclear (*Nuclear Magnetic Resonance* - NMR) (Rabi et al., 1938). Entretanto os métodos experimentais são, frequentemente, procedimentos dispendiosos e de difícil execução. Além disso, existem limitações técnicas que dificultam a determinação de várias proteínas. (Prosdocimi, 2003).

Isto se deve às técnicas difíceis para a predição, tempo lento de determinação da estrutura experimental e a custos experimentais caros (Fiser, 2009; Lee et al., 2009). Desta forma em 2005 a revista Science trouxe a questão “*como podemos prever como as proteínas irão se dobrar?*” como um dos 125 maiores problemas não resolvidos da ciência (Science, 2005). É neste sentido que, na bioinformática estrutural, surgem as abordagens computacionais para ajudar a solucionar esse problema.

### 2.2.3 Protein Structure Prediction Problem - PSP

As forças responsáveis pelo enovelamento de proteínas podem estar na sequência de aminoácidos que compõe a sua cadeia e nas suas interações com o meio (Novaes e Scott, 2009). São estas forças que, “*in vivo*”, tornam a busca por esta conformação não aleatória e muito mais espontânea, dado o grande número de conformações possíveis em uma proteína, conhecido como Paradoxo de Levinthal (Levinthal, 1968).

O paradoxo envolve a constatação de que existe tempo insuficiente para pesquisar, aleatoriamente, todo o espaço conformacional disponível para o enovelamento ou dobra de um polipeptídeo de cadeia como as proteínas (Levinthal, 1968).

Há infinitas maneiras possíveis para uma proteína se dobrar, mas ela utiliza somente uma em algumas dezenas de microssegundos. A mesma tarefa levaria cerca de 30 anos de tempo de computador (Kelley, 2009).

A exemplo do Paradoxo de Levinthal, consideramos uma pequena proteína, com 100 aminoácidos, e suponhamos que cada aminoácido só pode estar em um, de dois estados possíveis (por exemplo, só pode tomar duas orientações diferentes). Nestas condições, a proteína tem acesso a um total de  $2^{100} \approx 10^{30}$  conformações, total esse que inclui obviamente a estrutura nativa. Como a molécula não pode passar de uma conformação para a outra em menos de 1 picossegundo (ps), que é o tempo de uma vibração térmica, seriam precisos  $2^{100}$  ps, ou seja,  $3,9 \times 10^{10}$  anos, no mínimo, para explorar exaustivamente todo o espaço conformacional e

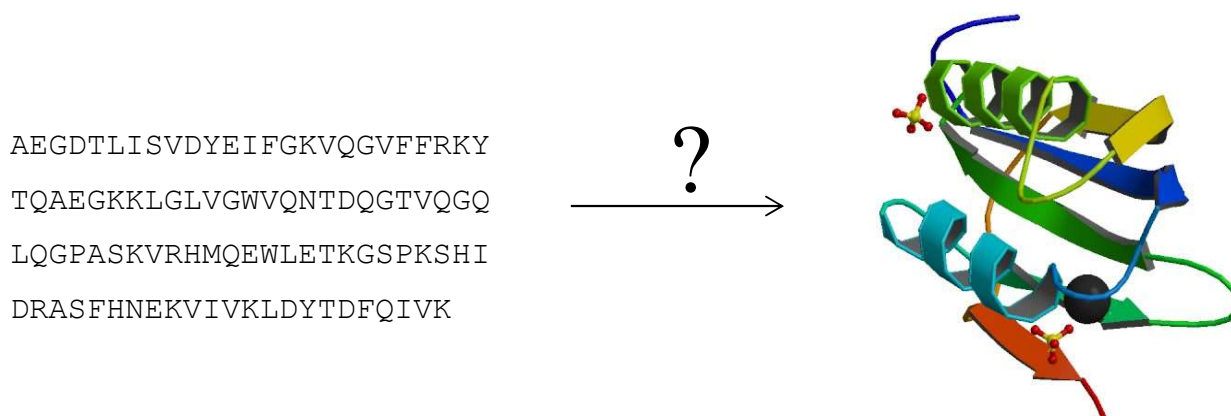


encontrar a conformação correspondente ao estado nativo, tornando-se intratável computacionalmente (Faisca, 2005).

Isto sugere que apenas uma pequena parte do espaço conformacional é explorado durante o processo de enovelamento, implicando assim a existência de vias cinéticas de enovelamento (“atalhos”) (Levinthal, 1968) Este paradoxo de como proteínas enovelam-se rápida e corretamente em sua conformação nativa é conhecido como o problema de predição de estruturas de proteínas (*Protein Structure Prediction Problem - PSP*).

Portanto, dado a grande explosão combinatória dos modos de dobramento entre os aminoácidos, as estruturas tridimensionais de proteínas são de grande complexidade computacional, sendo então problemas NP-completos, difíceis de serem tratados computacional e matematicamente. (Ribeiro, 2010).

Sendo assim um dos principais desafios da Bioinformática Estrutural (Figura 7) é entender como a informação decodificada em uma sequência linear de aminoácidos, ou estrutura primária de uma proteína, possibilita a formação de sua estrutura tridimensional (Dall’Agno, 2012).



**Figura 7.** Como podemos determinar a estrutura nativa de uma proteína a partir do conhecimento da sua estrutura primária? (Adaptada de (Faisca, 2005) A figura traz a sequência de aminoácidos da *Acyl-Phosphatase* e a partir dela sua estrutura terciária. (Código PDB: 2ACY - (Thunnissen et al., 1997).

## 2.2.4 Servidores de predição

### 2.2.4.1 CASP (*Critical Assessment of Techniques for Protein Structure Prediction*)

A tendência do uso de servidores de predição tem sido observada ao analisarmos os CASPs (*Critical Assessment of Techniques for Protein Structure Prediction* programa iniciado em 1994 pelo grupo de J. Moult) (Moult et al., 2014, 2005, 1997, 1995) onde bianualmente, as técnicas de predição são avaliadas, sendo que 2014 ocorreu a última edição (CASP11).

O CASP é uma competição “às cegas” onde a comunidade científica tenta prever a estrutura 3D de proteínas cujas suas estruturas são conhecidas, mas não ainda disponíveis ao público. São disponibilizadas sequências alvo para grupos de pesquisa de todo mundo. Cada grupo participante aplica algum método, algoritmo ou utiliza-se de um servidor de predição objetivando prever as estruturas da proteína alvo, que ao final são reveladas, e assim as predições são avaliadas pelo CASP.

Muitos desses métodos computacionais são disponibilizados via Web como um servidor de predição de estruturas de proteínas. Estes servidores de predição utilizam diferentes abordagens computacionais, para prever a estrutura tridimensional de proteínas a partir da sequência de aminoácidos ou estrutura primária.

Podemos observar (Tabela 3) que a participação de servidores de predição de estruturas 3D de proteínas no CASP aumentou desde as primeiras edições, porém no último ano não houve o surgimento de novas metodologias além do número de participações ter diminuído. Isso reforça a ideia de que é importante investir na pesquisa de novos métodos, na disponibilização de servidores e na predição computacional.

**Tabela 3.** Números de servidores participantes do CASP das últimas 8 edições.

Edição <sup>1</sup>	Ano	Número de servidores de predição registrados
CASP11	2014	84
CASP10	2012	122
CASP9	2010	139
CASP8	2008	122
CASP7	2006	98
CASP6	2004	65
CASP5	2002	72
CASP4	2000	38

<sup>1</sup>O CASP acontece desde 1994, mas é a partir de sua 4ª edição (CASP4), no ano 2000, que em seus dados (CASP em Números) aparece o número de servidores de predição registrados (*Number of prediction servers registered*).

(Fonte: <http://predictioncenter.org/>)

Podemos citar como principais servidores de predição de estruturas de proteínas descritos na literatura e apresentados no CASPs: Rosetta, agora como servidor Robetta (Rohl et al., 2004; Simons et al., 1999), I-TASSER (Wu et al., 2007; Zhang, 2008) e QUARK (Zhang, 2014). Estes métodos utilizam diferentes abordagens computacionais, para prever a estrutura tridimensional a partir da sequência de aminoácidos ou estrutura primária. O usuário fornece

uma entrada, a sequência da proteína alvo e obtém destes servidores o modelo, que são usados para inferir a função da proteína e orientar os esforços experimentais, contribuindo no campo da pesquisa como um todo (Kim et al., 2004).

Informações mais detalhadas sobre os servidores de predição e os métodos computacionais para predição de estruturas de proteínas estão descritas em trabalhos relacionados.

### 3 TRABALHOS RELACIONADOS

Existem diversos servidores de predição de estruturas de proteínas. Um dos passos rumo ao desenvolvimento do wCReF foi realizar um levantamento sobre os principais servidores de predição apresentados nos CASPs. Também, nesta seção, procuramos sintetizar os principais métodos computacionais para a predição de proteínas. A Tabela 4 apresenta os servidores encontrados, seus autores e os endereços Web.

**Tabela 4.** Os Principais servidores de predição participantes do CASP (Moult et al., 2014).

Servidor de predição	Disponível em:	Referência
QUARK	<a href="http://zhanglab.ccmb.med.umich.edu/QUARK/">http://zhanglab.ccmb.med.umich.edu/QUARK/</a>	(Zhang, 2014)
I-TASSER	<a href="http://zhanglab.ccmb.med.umich.edu/I-TASSER/">http://zhanglab.ccmb.med.umich.edu/I-TASSER/</a>	(Roy et al., 2010; Zhang, 2008)
ROBETTA	<a href="http://robeta.bakerlab.org/">http://robeta.bakerlab.org/</a>	(Kim et al., 2004)
ROSETTA	<a href="https://www.rosettacommons.org/">https://www.rosettacommons.org/</a>	(Rohl et al., 2004; Simons et al., 1999)
PMS: A Panoptic Motif Search Tool	<a href="http://www.motifsearch.com/index.php?page=motif_proteinseq">http://www.motifsearch.com/index.php?page=motif_proteinseq</a>	(Dinh e Rajasekaran, 2013)
MUFold	<a href="http://mufold.org/">http://mufold.org/</a>	(Zhang et al., 2010, 2010)
RaptorX server	<a href="http://raptorx.uchicago.edu/">http://raptorx.uchicago.edu/</a>	(Källberg et al., 2014)
SPARKSX	<a href="http://sparks-lab.org/yueyang/server/SPARKS-X/">http://sparks-lab.org/yueyang/server/SPARKS-X/</a>	(Yang et al., 2011)
MULTICOM	<a href="http://casp.rnet.missouri.edu/multicom_3d.html">http://casp.rnet.missouri.edu/multicom_3d.html</a>	(J. Li et al., 2014; Wang et al., 2010)
IntFOLD	<a href="http://www.reading.ac.uk/bioinf/IntFOLD/IntFOLD2_form.html">http://www.reading.ac.uk/bioinf/IntFOLD/IntFOLD2_form.html</a>	(Roche et al., 2011)
Chunk-TASSER	<a href="http://cssb.biology.gatech.edu/skolnick/webservice/chunk-TASSER/index.html">http://cssb.biology.gatech.edu/skolnick/webservice/chunk-TASSER/index.html</a>	(Zhou e Skolnick, 2007)

Esses Métodos computacionais para a previsão da estrutura de proteínas podem ser classificados em três categorias, que descreveremos a seguir.

### 3.1 Métodos computacionais para Predição de Estruturas Tridimensionais de Proteínas

Os métodos computacionais para predição de estruturas 3D de proteínas podem ser classificados em três grupos: (1) modelagem por homologia, (2) reconhecimento de padrões de enovelamento ou *fold recognition* e (3) métodos *ab initio*.

#### 3.1.1 Modelagem por Homologia

Métodos baseados em conhecimento: Tecnologias de modelagem por homologia e segmentação são frequentemente denominados métodos "baseados em conhecimento". Isto significa que os cientistas dependem de um conhecimento prévio estrutural de proteínas, dependem de estruturas 3D que já tenham sido determinadas para inferir a estrutura de proteínas para o qual apenas a sequência é conhecida (Maggio e Ramnarayan, 2001).

Na modelagem por homologia uma sequência de uma proteína (sequência alvo) é alinhada com outras sequências de proteínas ortólogas com estruturas 3D conhecidas e armazenadas no PDB (estrutura molde e suas sequências) (Dall'Agno, 2012).

Se o grau de identidade entre as estruturas primárias das proteínas de estrutura molde e da proteína alvo for igual ou superior a cerca de 25%, quando o número de resíduos é superior a 80, existe grande probabilidade de que estas proteínas tenham estruturas tridimensionais semelhantes e pode-se construir um modelo para a proteína alvo (Filho et al., 2003; Sander e Schneider, 1991). Como a exemplo de métodos com modelagem por homologia temos (Bowie et al., 1991; Fiser et al., 2000; Marti-Renom et al., 2002; Ring e Cohen, 1993; Söding et al., 2005).

#### 3.1.2 Reconhecimento de Padrões ou *Fold Recognition*

Se uma sequência de alta similaridade com estrutura conhecida não pode ser encontrada, uma nova proteína pode ainda ser estruturalmente similar a alguma proteína de estrutura já conhecida. Nesse caso, as proteínas são ditas remotamente homólogas (Lipinski, 2013). Métodos de compatibilidade sequência-estrutura: "*threading*" ou "*3D template matching methods*" (Jones et al., 1992; Lee et al., 2004; Simons et al., 1997; Skolnick et al., 2004; Xu e Xu, 2000) é a melhor maneira de atacar o problema de predição de estruturas de proteínas em casos em que a identidade entre as sequências é inferior a cerca de 25% (Filho et al., 2003). Estes métodos baseiam-se na avaliação da compatibilidade entre as estruturas terciárias da proteína alvo e de potenciais proteínas molde. Isto é, escolhe-se a proteína molde por métodos

empíricos, cuja estrutura terciária é "mais adequada" em relação à sequência da proteína alvo (Filho et al., 2003).

### 3.1.3 Predição *Ab initio*

A predição *Ab initio* ou de novo (Simons et al., 2001, 1999; Srinivasan e Rose, 1995) não se baseia somente em proteínas de estrutura conhecida, permitindo dessa forma a predição de estruturas ainda não resolvidas experimentalmente, é um método que não necessita de modelos (Dall'Agno, 2012). Pesquisas que utilizam métodos *ab initio* podem ser observadas em (Kim et al., 2004; Li e Scheraga, 1987; Ortiz et al., 1998; Simons et al., 2001). Além disso, os servidores que utilizam esse tipo de predição têm-se: Rosetta (Rohl et al., 2004; Simons et al., 1999), I-TASSER (Roy et al., 2010; Zhang, 2014, 2008), LINUS (Srinivasan et al., 2004; Srinivasan e Rose, 2002, 1995) e o QUARK (Zhang, 2014).

O método CReF compreende princípios dos métodos de predição *ab initio* ou de novo para prever novas formas de enovelamento e da modelagem comparativa por homologia para garantir alta acurácia nas predições (Dall'Agno, 2012). Um resumo da sua metodologia é descrito a seguir.

## 3.2 O Método CReF

Em meio às diferentes áreas de estudo da bioinformática, a Bioinformática Estrutural (BE) hoje é um dos campos mais estudados e está em destaque atualmente devido à importância das proteínas para as funções químicas e biológicas essenciais para manutenção da vida. O que se busca é compreender a relação entre a sequência de aminoácidos de uma proteína e sua estrutura tridimensional 3D, que está relacionada à função destas macromoléculas. Esta questão tornou-se um dos maiores desafios de pesquisa na atualidade. Dentre os métodos relacionados à predição de estruturas de proteínas está o método CReF (*Central Residue Fragment-based Method*) que é um algoritmo proposto por Dorn & Norberto de Souza (2008) para predição da estrutura 3D aproximada de uma proteína ou polipeptídeo. Este método já demonstrou ótimos resultados, confirmados pela pesquisa de Dall'Agno (2012).

O método CReF usa técnicas de mineração de dados para agrupar dados de estruturas determinadas experimentalmente, utilizando intervalos de variação angular para representar uma conformação através da manipulação das informações estruturais (Dorn e Norberto de Souza, 2008). Esse método leva em consideração que uma proteína é composta por uma sequência de aminoácidos, que são representadas por uma variável  $x_i$  dentro de um vetor  $V = \{x_1, x_2, \dots, x_n\}$  (Dall'Agno, 2012). Cada  $x_i$  possui ângulos de rotações internas da cadeia

principal, os ângulos de torção  $\varphi$  (phi),  $\psi$  (psi) e  $\omega$  (ômega), que formam um triplete. O princípio que dois átomos não podem ocupar o mesmo lugar no espaço limitam os valores dos ângulos conformacionais. O ângulo  $\omega$  (ômega) geralmente possui os valores de *cis*  $\approx 0^\circ$  (mais raro, na maioria dos casos um resíduo de Prolina) e  $\approx$  *trans*  $180^\circ$ , mais comum em proteínas (Lesk, 2008) e por conseguinte adotado no método (Dorn, 2008). O que varia, portanto, na representação da cadeia principal é somente o duplete dos ângulos de torção  $\varphi, \psi$ .

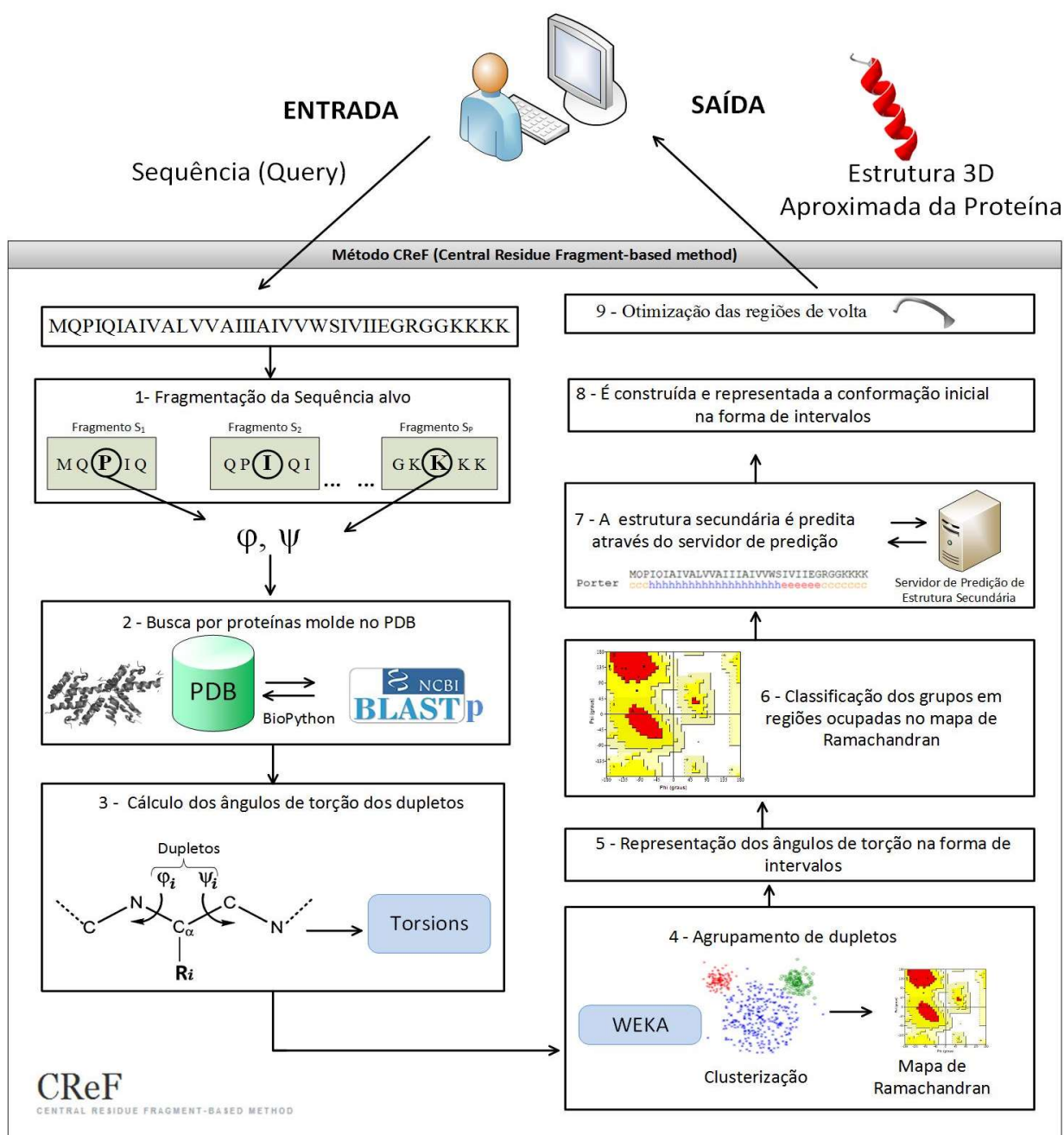
A modelagem no CReF inicia-se pela procura de moldes na base de dados do PDB (<http://www.rcsb.org/pdb>) (Berman et al., 2000), usando-se como parâmetro a entrada, a sequência primária de estrutura não determinada experimentalmente (alvo), realizando a predição de sua estrutura 3D aproximada.

### 3.2.1 Etapas do CReF

O objetivo fundamental do CReF é construir uma conformação aproximada a partir de uma sequência de polipeptídeo de estrutura alvo desconhecida. Seu algoritmo básico consiste de nove etapas, sendo que as etapas dois e sete são executadas remotamente e as demais localmente (Dorn e Norberto de Souza, 2008). A metodologia básica do CReF compreende 9 etapas, que ficarão disponíveis através do wCReF conforme a Figura 8. Um resumo de cada etapa do método pode ser conferido a seguir.

- 1) Fragmentação da sequência alvo:** O CReF inicialmente pega a sequência alvo (aquela que busca-se fazer a predição, através da sequência de aminoácidos) e quebra consecutivamente e continuamente em fragmentos (*si*) com 5 resíduos de aminoácidos, formando um conjunto com todos os possíveis fragmentos  $S = \{si, si+1, \dots, sp\}$ , cada um com os tripletos consecutivos de ângulos de torção  $\{(wi-1, \varphi_i, \psi_i), \dots, (wj-1, \varphi_j, \psi_j)\}$  (Dorn e Norberto de Souza, 2008).
- 2) Busca por Proteínas molde no PDB:** O Método CReF não faz uso de fragmentos inteiros, mas apenas utiliza as informações do ângulo de torção o phi ( $\varphi$ ), psi ( $\psi$ ), do resíduo central de cada fragmento obtido na etapa anterior. Para cada fragmento *si* obtido na etapa anterior é realizada uma busca de fragmentos molde no PDB. Para realizar essa busca é utilizado o *Basic Local Alignment Search Tool* (BLAST) (Altschul et al., 1997) algoritmo de comparação de informações de sequências biológicas primárias, como aminoácidos de diferentes sequências de proteínas ou nucleotídeos de sequências de DNA. No CReF utilizamos o BLASTp, o algoritmo específico para busca de aminoácidos

de proteínas, configurado como padrão para executar com a matriz de substituição BLOSUM62 (Henikoff e Henikoff, 1992).



**Figura 8.** Esquema dos passos do CReF. Primeiramente o usuário informa a *query*, a sequência de aminoácidos da proteína alvo. Nesta figura está representado a submissão da proteína de código PDB 1PJE (Park et al., 2003).

Como resultado esperado pode-se identificar os fragmentos homólogos (molde) ao fragmento submetido *si* (alvo), que não possuam relação evolucionária com a sequência alvo *K* (30% ou menos de similaridade) e que possuam o mesmo tamanho de cada fragmento *si* (Dorn e Norberto de Souza, 2008). Dessa forma é gerada uma lista de



códigos PDB de moldes, sendo que para cada fragmento é armazenado localmente seus arquivos PDB. A conexão automatizada ao PDB é garantida graças à biblioteca *BioPython* (Cock et al., 2009) que analisa automaticamente arquivos XML (*Extensible Markup Language*) resultante das consultas (Dorn, 2008).

- 3) Cálculo dos ângulos de torção dos dupletos:** Com o resultado da etapa anterior podemos calcular os ângulos de torção (representados por  $t_i = (\varphi, \psi)$ ) do aminoácido central de cada fragmento molde obtido na etapa 2. Esse cálculo é realizado pelo software *Torsions* (Dr. Andrew C.R. Martin's Group, 2014).
- 4) Agrupamento dos dupletos:** O agrupamento das tuplas é feito a partir do gráfico de *Sasisekharan-Ramakrishnan-Ramachandran* (Ramachandran e Sasisekharan, 1968), normalmente abreviado como mapa de *Ramachandran*, que define regiões com as faixas permitidas para  $\varphi$  e  $\psi$ , e para  $\omega = 180^\circ$  (Lesk, 2008). O agrupamento é feito a partir das regiões favoráveis do mapa de *Ramachandran*, buscando encontrar como resultado 4 grupos ( $f = 4$ ). No fim desta etapa obtém-se para cada fragmento então um conjunto de quatro grupos (Dorn e Norberto de Souza, 2008), sendo para cada um deles é calculado a média e o desvio padrão estimados para cada ângulo ( $\varphi, \psi$ ), que são utilizados para obter os intervalos de variação angular. Para realização dos agrupamentos é utilizado o programa WEKA (*Waikato Environment for Knowledge Analysis*), que possui uma coleção de algoritmos de aprendizado de máquina para tarefas de mineração de dados (Frank et al., 2005, 2004). Neste programa o algoritmo utilizado para esse fim baseia-se no método probabilístico EM (*Expectation Maximization*) que analisa as diferentes distribuições de probabilidades para cada grupo, buscando identificar o conjunto de *clusters* mais favoráveis em uma coleção de dados (Dorn e Norberto de Souza, 2008).
- 5) Representação dos ângulos de torção na forma de intervalos:** A partir dessa etapa cada fragmento passa ser representado pelos intervalos construídos de ( $\varphi$ ) e ( $\psi$ ), para cada grupo  $k_i$  de *si*. A representação das torções dos ângulos diedro na forma de intervalos de variação reduz drasticamente o espaço de busca conformacional.
- 6) Classificação dos grupos em regiões ocupadas no mapa de *Ramachandran*:** Nesta etapa os valores de todos os clusters dos grupos de  $k_i$  são rotulados para cada representação na forma de intervalo de variação angular (Dorn e Norberto de Souza, 2008). Todo processo é baseado segundo a região do mapa de *Ramachandran* que ele ocupa, conforme biblioteca baseada na pesquisa de Thornton e seus colaboradores (Laskowski et al., 1993; Morris et al., 1992) que divide o mapa de *Ramachandran* em 11 regiões preferenciais. Por simplificação no CReF, essas regiões favoráveis foram

reduzidas para oito (Tabela 5) e foram identificadas por: A, B, L, a, b, l, p e c. As regiões  $\sim a$ ,  $\sim b$ ,  $\sim l$ ,  $\sim p$  e o restante da área não favorável do mapa passam a ser representadas, por c que é chamada de região de volta (*coil*) (Dall'Agno, 2012).

**Tabela 5.** As 8 regiões conformacionais utilizadas do mapa de *Ramachandran* no método CReF, segundo Thornton e colaboradores.

Código da região	Ocorrência	Descrição
A	região mais favorável	hélice- $\alpha$
B	região mais favorável	folha- $\beta$
L	região mais favorável	hélice- $\alpha$ à esquerda
a	região favorável	hélice- $\alpha$
b	região favorável	folha- $\beta$
l	região favorável	hélice- $\alpha$ à esquerda
p	região favorável	extensão de hélice- $\alpha$
c	$\sim a$ , $\sim b$ , $\sim l$ , $\sim p$ : região aceitável	$\sim a$ : hélice- $\alpha$ , $\sim b$ : folha- $\beta$ , $\sim l$ : hélice- $\alpha$ à esquerda, $\sim p$ : extensão de hélice- $\alpha$ e restante da área: exceto para a Glicina
( $\sim a$ , $\sim b$ , $\sim l$ , $\sim p$ e restante da área)	restante da área: região não permitida	

(Fonte: Dall'Agno, 2012)

- 7) A estrutura secundária é predita através do servidor de predição:** O CReF utilizava um consenso sobre a predição da estrutura secundária da sequência alvo K obtido através de três métodos de predição de estrutura secundária DSC (King e Sternberg, 1996), PHD (Rost e Sander, 1993) e PREDATOR (Frishman e Argos, 1996). Eles analisavam qual a região do mapa de *Ramachandran* que os ângulos de torção ( $\varphi$ ,  $\psi$ ) de cada aminoácido possivelmente estariam ocupando (Dorn e Norberto de Souza, 2008). Na pesquisa de (Dall'Agno, 2012) definiu-se o *Porter* (Pollastri e McLysaght, 2005) como método a ser utilizado na predição de estrutura secundária para proteínas com 20 ou mais aminoácidos, e, para proteínas com menos de 20 aminoácidos, optou-se por utilizar o método SAM-T08.
- 8) É construída e representada à conformação inicial na forma de intervalos:** Nesta etapa começa a construção da conformação inicial, baseada nas informações de cada grupo *ki* de cada fragmento *si*. Elege-se um grupo para representar o *-ésimo* aminoácido da sequência K (Dorn, 2008) guiadas pela predição da estrutura secundária.
- 9) Otimização das regiões de volta:** As estruturas irregulares como as voltas e as alças, que são ditas espirais desorganizadas e conectam sucessivas estruturas secundárias regulares

(hélice ou folha) acontecem onde o polipeptídeo muda de direção (Voet e Voet, 2013). Voltas são estruturas secundárias irregulares e, normalmente, possuem de dois a quatro resíduos de aminoácidos. As alças possuem cinco ou mais resíduos de aminoácidos e são denominadas espirais desorganizadas (do inglês *random coils*). Elas possuem influência na determinação da forma do enovelamento das proteínas (Fiser et al., 2000) e são os tipos de estruturas secundárias mais difíceis de serem previstas, pois elas podem aparecer em qualquer área do mapa de *Ramachandran*. Em virtude disso, o método optou por otimizar, pela redução do intervalo da conformação inicial, somente as regiões de volta identificadas na predição da estrutura secundária (Dorn, 2008 apud Dall'Agno, 2012).

### 3.2.2 Utilização do Método CReF

Para se utilizar o método CReF sem a interface web, alguns requisitos são necessários, como a criação de diretórios e o *Download* de *softwares* externos. O método CReF funcionava em computador local, com o Sistema Operacional GNU/Linux (Torvalds, 1999). Como aspectos motivadores pela escolha de uma plataforma Unix Gibas & Jambeck (2001) apontam não apenas o alto grau de confiabilidade e desempenho dessa plataforma, mas também que é possível encontrar uma grande quantidade de ferramentas de bioinformática já bastante consagradas por pesquisadores da área. Apesar dessa vantagem, os usuários de outros Sistemas Operacionais ficavam limitados a utilização do método somente nesta plataforma.

O método CReF foi desenvolvido em linguagem *Python*. O Python é uma linguagem de programação de alto nível, interpretada, orientada a objetos, funcional, de tripagem dinâmica e forte. É um tipo de linguagem usada largamente em Bioinformática, pois. Bibliotecas *Python* já conhecidas são utilizadas por programadores e pela comunidade científica na implementação de algoritmos e métodos. Além disso algumas ferramentas externas ao CReF possuem dependência de certos programas e bibliotecas do Python.

Para poder fazer a configuração do método CReF o usuário precisa fazer alterações em seu código fonte, como por exemplo o caminho para diretórios e pastas. Isso torna sua execução um pouco complexa, pois o usuário tem que alterar diretamente o código fonte, o que pode acarretar em erros não esperados pelos usuários. O que se espera com a interface wCReF é a automatização desse processo, onde o usuário poderá usar diretamente a ferramenta de maneira mais fácil e eficiente.

A instalação de outros pacotes e *softwares* na máquina local também são necessários:

- 1) **BioPython:** *BioPython* é um conjunto de ferramentas disponíveis para bioinformática desenvolvido em linguagem Python, produzido por uma associação internacional que

desenvolvem softwares para biologia molecular (Cock et al., 2009). O site oficial que é <http://www.biopython.org>. Como o *BioPython* é uma fonte de módulos, scripts e links, no método CReF utilizamos para acessar o BLASTp.

- 2) **BLASTp:** Para a busca no PDB – Protein Data Bank (Berman et al., 2000) por fragmentos moldes utilizamos o programa BLASTp (Altschul et al., 1997), programa que se utiliza como entrada uma sequência de aminoácidos e realiza uma busca no banco de dados de proteínas. Esse tipo de BLAST é para comparação de sequências de aminoácidos em bancos de dados de proteínas; No CReF ele permite a identificação de fragmentos homólogos (moldes) ao fragmento submetido si (alvo).
- 3) **Torsions:** O Grupo do Dr. Andrew C. R. Martim ([www.bioinf.org.uk](http://www.bioinf.org.uk)) desenvolveu o Torsions, um programa que faz a leitura de arquivo PDB e calcula ângulos de torção da cadeia principal de uma proteína. Ele calcula os ângulos phi, psi e ômega. No CReF ele calcula os ângulos de torção do dupletos representados pela tupla ( $\phi$ ,  $\psi$ ) do aminoácido central de cada sequência molde obtida do PDB.
- 4) **Weka:** Weka (*Waikato Environment for Knowledge Analysis*) (<http://www.cs.waikato.ac.nz/ml/weka/>) é uma coleção de algoritmos provenientes de diferentes abordagens/paradigmas na subárea da inteligência artificial dedicada ao estudo da aprendizagem de máquinas, aplicadas à mineração de dados (*data-minning*). Os algoritmos podem ser aplicados diretamente a um conjunto de dados ou chamada a partir do seu próprio código Java. Weka contém ferramentas para pré-processamento de dados, classificação, regressão, *clustering*, regras de associação e visualização (Witten et al., 1999). Foi desenvolvido por pesquisadores da Universidade de Waikato (*University of Waikato, Department of Computer Science*).
- 5) **Amber:** Amber (*Assisted Model Building with Energy Refinement*) (<http://ambermd.org/>) foi desenvolvido primeiramente pela equipe de pesquisadores do grupo de Peter Andrew Kollman (Kollman et al., 1995) da Universidade da Califórnia, San Francisco. Refere-se a uma família de campos de força de dinâmica molecular de biomoléculas e também é o nome do pacote de *softwares* de dinâmica molecular que simula estes campos de força, que são de domínio público e são usados em uma variedade de programas de simulação.
- 6) **Jmol:** Jmol é um visualizador de Java open-source para estruturas químicas em 3D (<http://www.jmol.org/>). Ele possui licença open-source, e foi desenvolvido em linguagem Java e por isso, o software roda em Windows, Mac OS, Linux e Unix. Dentro da página desse aplicativo há uma lista de artigos que descrevem Jmol, na seção do Wiki Jmol. O primeiro deles foi escrito por Egon Willighagen (Willighagen, 2001).

Para cada uma dessas aplicações ou pacotes o usuário tem que realizar a instalação e configuração para utilização do método CReF. Os detalhes de cada passo para uso do método são descritos no Manual 2.3 do CReF (APÊNDICE A) como material suplementar.

A interface do wCReF facilitará o uso da metodologia, pois o usuário só necessitará de um navegador web e de conexão com a internet. Além disso a interface busca prover assistência ao usuário, tanto no sentido de explicar para que serve a aplicação, quanto para detalhar os resultados ou mesmo funções específicas, enfim prover informações úteis e ajuda ao usuário, sendo um aspecto muito importante para a usabilidade do sistema.

Espera-se que a partir desta iniciativa, o método CReF esteja mais facilmente ao alcance de pesquisadores, como bioinformatas, biólogos e cientistas da computação, facilitando assim seu uso pela comunidade científica.

### **3.3 Pesquisas em IHC e Bioinformática**

O trabalho de (Shaer et al., 2010) foi desenvolvido por meio de uma pesquisa qualitativa, com um estudo envolvendo 17 participantes. Foram realizadas entrevistas semiestruturadas, de 45-60 minutos, com biólogos computacionais e moleculares, buscando compreender às práticas de trabalho atuais destes pesquisadores. Todos os pesquisadores realizavam pesquisas em genômica. Durante as entrevistas, foi pedido aos participantes relatar seus objetivos de pesquisa, suas práticas de trabalho, e as ferramentas computacionais que eles usam. Foi coletado os dados pelas entrevistas, com fotos e gravação de áudio dos participantes em seu espaço de trabalho, realizado a coleta de amostras de trabalho relevantes, e realizado a captura de tela dos computadores dos participantes para demonstrar a forma como os mesmos executam várias tarefas. Com estas observações, buscou-se identificar os requisitos de projeto para apoiar os pesquisadores a organizar grandes quantidade de informação e obter bons *insights* biológicos. Descobriram que os cientistas têm dificuldade, molecular e computacionalmente, de interpretação, anotação, comparação e compartilhamento relacionados à vasta quantidade de informação biológica. Foi desenvolvida, então, a interface do *G-nome Surfer*, uma interface de usuário para *tabletop* de exploração de dados genômicos, que emprega tecnologias multi-toque e técnicas de interação tangíveis. Por último, um estudo de usabilidade na primeira utilização do G-nome foi realizado, com 16 alunos do curso de graduação em Ciências Biológicas. Foi solicitado para que os participantes realizassem tarefas em duplas, em sessões de 40 minutos. O objetivo era medir a carga de trabalho de cada participante. Na sequência de um breve treinamento, todos os participantes foram capazes de completar a tarefa (e encontrar as respostas corretas para as perguntas biológicas) dentro de um prazo razoável, identificar os

resultados do BLAST e procurar dados genômicos. Os usuários descreveram a experiência como "divertida", "fácil de usar", com "um bom tempo de resposta" e "visualmente estimulante". Mais especificamente, os participantes gostaram da integração das informações, da flexibilidade de mover e redimensionar janelas, e da função de poder procurar genes específicos, os ligando a artigos.

O projeto Mobylye (Néron et al., 2009) fornece uma maneira eficaz de agrupar uma grande quantidade de ferramentas de bioinformática em um único ambiente homogêneo. Estas ferramentas geralmente são acessadas através de linhas de comando. Este sistema foi desenvolvido com a preocupação de satisfazer as exigências dos diferentes públicos de bioinformatas. Além disso, o Mobylye fornece ao cientista uma visão global e integrada de todos os elementos necessários para o desempenho de suas análises. Em uma só página, o usuário pode ver quais programas estão disponíveis e as análises que já foram executadas. O desenvolvimento foi baseado em um processo de *design* centrado no usuário, envolvendo numerosas entrevistas e uma série de oficinas com usuários. Foram identificados anteriormente as principais preocupações dos usuários finais antes da concepção da interface, foram definidos os requisitos técnicos e que esta seria uma interface de usuário Web, permitindo uma fácil navegação e o gerenciamento dos dados dentro de uma única área de trabalho. Durante um período de 4 anos que antecederam a primeira versão do Mobylye, uma série de cerca de 30 entrevistas com usuários e oficinas de *design* participativo, envolvendo sessões de brainstorming e prototipagem de vídeo, foram realizadas. Estes testes foram utilizados para obter uma compreensão mais profunda de como um portal da web pode facilitar a utilização de ferramentas de bioinformática por biólogos. Estes estudos foram descritos por Letondal e Amanatian (Letondal e Amanatian, 2004). Os resultados mostraram as principais necessidades dos biólogos: (1) um conjunto de ferramentas conhecidas, integradas em um portal destinado a utilizadores inexperientes; (2) uma síntese de suas análises e uma organização cuidadosa dos resultados; (3) recursos de reutilização para reexecutar comandos anteriores; e (4) pipelines de análise prontas para uso, semelhante aos protocolos definido pelo usuário.

Para testar o novo portal Mobylye, usuários foram convidados a participar em sessões de teste. Eles selecionaram 16 usuários, entre os usuários do portal anterior (desde iniciantes a utilizadores frequentes). Utilizou-se o método de pensamento em voz alta. Os participantes foram informados de que poderiam trazer os seus próprios dados. A maioria deles tentou realizar uma análise de estrutura 2D, alinhamento de sequência e análise filogenética. Vários usuários perceberam que o novo sistema era mais complexo, e, mas mais completo. Eles apreciaram a capacidade de recuperar resultados. Além disso problemas relatados no portal

anterior, foram resolvidos como: (A) ter que avançar e retornar para acessar os resultados; (B) a contabilização de usuários; (C) a não compreensão de alguns termos em inglês (como alinhamento de 'pares') utilizado para classificação dos programas; e (D) as ambiguidades relativas armazenamento de dados.

Também relacionado é o artigo de Pavelin (Pavelin et al., 2012) denominado “*Bioinformatics Meets User-Centred Design: A Perspective*”, do grupo de pesquisadores do Instituto de Bioinformática Europeu - EMBL-EBI (*European Molecular Biology Laboratory/ European Bioinformatics Institute*). Neste trabalho é explorado a necessidade de estratégias de *design* centrado no usuário (UCD) ao se projetar recursos de bioinformática, demonstrado através de relatos dos trabalhos do grupo. UCD é uma filosofia de *design*, onde as exigências dos usuários são levadas em conta em todas as fases do processo de *design* para um produto ou serviço. O objetivo da UCD é produzir uma ferramenta eficaz e utilizável, que é trabalhada para utilização por determinados tipos de pessoas (Pavelin et al., 2012). O objetivo é apresentar ao leitor como técnicas de UCD podem ser aplicadas com sucesso no *design* de *softwares* para bioinformática. No artigo, é descrito dois projetos do EMBL-EBI que adotaram uma abordagem UCD. O primeiro projeto, buscou redesenhar o principal serviço de busca da EMBL-EBI, EB-eye (Valentin et al., 2010), com foco inicial de informações sobre gene e proteínas. Entrevistas com usuários foram conduzidas para descobrir como as pessoas recuperam a informação de que necessitam a partir da Internet. Os resultados mostraram que os cientistas costumam usar Wikipédia e motores de busca como o Google. Usuários disseram que gostariam que o site da EBI fornecesse um resumo de busca, de modo que se o usuário procurou um gene ou proteína em particular, os resultados resumiriam todas as informações disponíveis ao lado cada um dos recursos em um único lugar. Em resposta a isso, os desenvolvedores criaram um protótipo interativo, analisado por sessões de testes de usabilidade *one-to-one* com mais de 70 pesquisadores de 20 instituições em toda a Europa. Os participantes eram dos setores acadêmico e industrial. Após várias rodadas para recolhimento de feedback e aperfeiçoamento do serviço, foi lançada uma nova pesquisa, em janeiro de 2011. A resposta dos usuários foi tão positiva que a EBI decidiu expandir o serviço para cobrir outros domínios científicos e torná-lo disponível ao público para qualquer organização que necessite de reuso como um recurso em seus próprios sites. Em um projeto paralelo, foi desenvolvido um portal de informações sobre enzimas. O objetivo do trabalho era fornecer um portal único para todos os recursos relacionados com as enzimas do EMBL-EBI, incluindo as bases de dados e os serviços essenciais da web. Os pesquisadores organizaram workshops e entrevistas com usuários para descobrir o que estes desejavam a partir do site. Depois disso, testaram os protótipos com os usuários antes de

escrever qualquer código, para garantir que a navegação e funcionalidades fossem ideais. Outra vantagem é que o desenvolvedor participou ativamente aos testes de usabilidade, o que o ajudou a conhecer desde muito cedo a tecnologia e as ferramentas de que precisaria para criar o recurso. O *Enzyme Portal* a foi lançado em fevereiro de 2012.

O estudo de Shyr (2014) analisa a usabilidade de interfaces de usuário para software clínico de análise exome. Há dois objetivos do estudo, determinar as principais características de sucesso em interfaces de usuário para o software de análise exome clínico com base na perspectiva de geneticistas clínicos especialistas e avaliar as interações de usuário do sistema, a fim de revelar os pontos fortes e fracos dos softwares existentes. Os Métodos utilizados foram pesquisas, entrevistas e análise cognitiva de tarefas, para a avaliação de dois pacotes de software de análise de sequência exome. A amostra foi de dez geneticistas clínicos que interagiram com os pacotes de software. Foram detectadas 193 questões de usabilidade, a maioria das quais referentes a interface e layout de navegação, e a resolução de relatórios. O estudo destaca as lacunas em recursos de software específicos típicos dentro análise exome. Os clínicos têm melhor desempenho quando o fluxo do sistema está estruturado em camadas bem definidas, ainda personalizáveis para incorporação dentro do fluxo de trabalho clínico. Foi apresentado a primeira aplicação de métodos de usabilidade para avaliar interfaces de software no contexto de análise exome. Os resultados destacam como o estudo de respostas de usuário podem levar à identificação de problemas de usabilidade e desafios, e revelar oportunidades de reengenharia de software para melhorar a análise de sequenciamento. Como a análise do genoma de grande escala torna-se cada vez mais comum na área da saúde, é fundamental que as interfaces de software eficientes e eficazes sejam fornecidas para acelerar a adoção clínica desta tecnologia. São discutidas implicações para a melhoria da concepção das tais aplicações.

O trabalho de Javahery et al. (2004) é parte do Projeto *BioUse*, do laboratório de usabilidade do Grupo HCSE (*Human-Centered Software Engineering*) da Universidade de Concordia, Montreal. Nele foram realizadas entrevistas etnográficas e estudos de usabilidade com 16 biólogos e três peritos de interface de usuário, usando o site do NCBI (*National Center for Biotechnology Information*), que é um dos principais portais de informação em bioinformática, com acesso a ferramentas especializadas (tais como para alinhamento de sequências e visualização molecular). Usuários podem acessar, interagir e perseguir metas, que vão desde coleta de informações simples (tais como buscar artigos no PubMed) ou mais específicas para resolver problemas de biologia molecular (como alinhamentos com o BLAST). O projeto do portal Web *BioUse* tem o objetivo de ser uma interface fácil de usar, customizável para uma variedade de usuários. É um *front-end* para um número de ferramentas, incluindo



*BLAST*, e *ClustalW*. A informação obtida dos resultados foi utilizada para entender o domínio do problema (bioinformática), objetivos do usuário e as expectativas para as ferramentas relacionadas. Foram entrevistados um total de 19 participantes de grupos norte-americanos e europeus de investigação, incluindo biólogos, médicos e cientistas da computação (Javaheery et al., 2004). Os usuários foram divididos em duas personas: novatos e especialistas. O conceito de personagens ajuda a colocar um utilizador específico, de uma aplicação particular, no centro do processo de concepção. Personas são baseadas nas características do usuário (como hábitos de trabalho, experiência e especialização). Os estudos de usabilidade encontraram diferenças consideráveis na experiência relatada pelo persona especialista vs. novato. Por exemplo, o percentual de usuários satisfeitos com o site do NCBI era uniformemente maior entre os especialistas, que entre os novatos, quando testados contra nove heurísticas de usabilidade diferentes. As entrevistas etnográficas confirmaram que os usuários novatos encontram uma curva de aprendizado mais íngreme para esses sites, do que eles preferem. Javaheery et al. (2004) cita ainda as equipes de IHC estão engajadas com pesquisas envolvendo IHC e bioinformática: bioinformática visualização e usabilidade (Ben Shneiderman, da Universidade de Maryland); Interfaces com zoom, para navegação do genoma e comparação de múltiplos genomas e proteomas (William Ribarsky do Georgia Tech); protocolos centrados no usuário na pesquisa em bioinformática e recuperação da informação (Joan Bartlett na Universidade McGill); e análise e classificação de tarefas em bioinformática (Robert Stevens na Universidade de Manchester) (Javaheery et al., 2004).

## 4 RESULTADOS

### 4.1 Conhecer os usuários e suas tarefas

Para conhecer os usuários e suas tarefas buscou-se entender as funcionalidades de um servidor de predição de estrutura 3D de proteínas, qual o perfil dos usuários finais e quais as tarefas comumente executadas.

Primeiramente consultou-se especialistas da equipe do Laboratório de Bioinformática, Modelagem e Simulação de Biosistemas – LABIO, para saber quais são as tarefas que são frequentemente realizadas por um usuário que desejasse submeter uma predição em um servidor.

O resultado serviu para definir quais as atividades que deveriam ser executadas pelos especialistas durante a avaliação. Isto serviu para facilitar o uso dos servidores de predição, principalmente por aqueles avaliadores que não são especialistas em bioinformática. Além disso, orientações da avaliação foram passadas pelo condutor da avaliação e através de e-mail, em formato de tutorial. Foram definidas 9 tarefas, como um cenário comum aos usuários que utilizam este tipo de sistema:

- 1) Cadastrar-se como usuário (Os servidores de predição geralmente exigem cadastro para seu uso).
- 2) Efetuar Login com as credenciais criadas. Alterar senha, ou algum dado do cadastro já efetuado.
- 3) Verificar qual os dados de entrada para realizar a predição.
- 4) Enviar a sequência de uma proteína alvo para predição.
- 5) Verificar a fila de espera da predição.
- 6) Modificar os parâmetros para predição (se disponível) e realizar um novo envio.
- 7) Conferir se o servidor retornou os dados esperados.
- 8) Verificar se o mesmo possui documentação, ajuda e referências bibliográficas.
- 9) Ver se os resultados são de fácil acesso e compreensão.

### 4.2 Avaliação Heurística

Existe uma série de métodos de avaliação, que podem ser utilizados em diferentes etapas do desenvolvimento de interfaces Web (Winckler, 2002). Acreditamos que a Avaliação Heurística é um método eficiente para este propósito, pois é de baixo custo e rápido para a identificação de problemas de usabilidade, aplicável também a ferramentas de bioinformática (Mirel e Wright, 2009).

Portanto, este estudo realizou a avaliação de usabilidade em três dos principais servidores de predição de estruturas 3D de proteínas participantes do CASP. Os servidores avaliados são apresentados na Tabela 6.

**Tabela 6.** Os servidores de predição avaliados nesta pesquisa

<b>Servidor de predição</b>	<b>Disponível em:</b>	<b>Referência</b>
QUARK	<a href="http://zhanglab.ccmb.med.umich.edu/QUARK/">http://zhanglab.ccmb.med.umich.edu/QUARK/</a>	(Zhang, 2014)
I-TASSER	<a href="http://zhanglab.ccmb.med.umich.edu/I-TASSER/">http://zhanglab.ccmb.med.umich.edu/I-TASSER/</a>	(Roy et al., 2010; Zhang, 2008)
ROBETTA	<a href="http://robetta.bakerlab.org/">http://robetta.bakerlab.org/</a>	(Kim et al., 2004)

Cada inspetor fez a avaliação dos três servidores, buscando identificar, na sua interface, problemas que violassem qualquer uma das 10 Heurísticas de Nielsen. As orientações que foram recebidas, assim como o formulário para Avaliação Heurística (também disponibilizado via web) encontram-se no APÊNDICE B e C.

Durante a sessão de avaliação, seguimos as orientações de Nielsen (Nielsen, 2001). Os avaliadores percorreram a interface várias vezes, inspecionando os vários elementos de diálogo, comparando com a lista de Heurísticas. Recomendamos que a interface fosse inspecionada pelo menos duas vezes, e sugerimos que o primeiro passo fosse à análise geral do sistema, em que o avaliador analisasse livremente a interface buscando conhecê-la.

Em seguida, realizasse uma segunda passagem, que deveria permitir que o avaliador se concentrasse em elementos de interface específicos, para saber como eles se encaixam no todo. Foram escolhidos 4 avaliadores especialistas (Tabela 7) e um condutor da avaliação. Todos os avaliadores receberam o termo de consentimento livre e esclarecido (APÊNDICE C).

**Tabela 7.** Inspetores – Avaliação Heurística

<b>Avaliador</b>	<b>Qualificação dos especialistas</b>	<b>Área de estudo</b>
A	Doutor em Ciências Físicas e Biomoleculares	Bioinformática
B	Doutor em Informática na Educação	IHC e Informática na Educação
C	Doutoranda em Ciência da Computação	Bioinformática - Desenvolvimento de Software
D	Doutoranda em <i>Design</i> e Tecnologias	IHC e Tecnologias Assistivas

<sup>1</sup> Além da avaliação, este último inspetor foi responsável por conduzir as avaliações e juntar em um único documento. Com relação a IHC e este avaliador participou de disciplinas nesta área no mestrado, não sendo especialista nesta área.

Como comumente aplicações na área da Bioinformática são desenvolvidas tanto por cientistas da computação, programadores, biólogos e cientistas, foi definido que os avaliadores seriam de diferentes áreas do conhecimento, tanto especialistas na área da Interação Humano-Computador, quanto da área de domínio, para verificar se os problemas encontrados seriam diferentes.

Só depois de todas as avaliações concluídas, os resultados foram agregados. Esse processo é importante a fim de assegurar avaliações independentes e imparciais de cada avaliador (Nielsen, 2001). No final, todos os problemas encontrados na Avaliação Heurística foram reunidos em um único documento.

Todos os avaliadores receberam um formulário com as orientações de como realizar a Avaliação Heurística, com apoio para os especialistas de IHC entenderem as tarefas, assim como um formulário eletrônico via Web para preenchimento dos erros de usabilidade que violem os princípios. Como os servidores de predição não fazem parte do domínio dos especialistas em IHC foi necessário, à criação de um tutorial de como se utilizar um servidor de predição de estruturas de proteínas.

Cada problema encontrado e especificado também deveria conter um grau de severidade (de 0 até 4) conforme a classificação de Nielsen (Tabela 8).

**Tabela 8.** Escala de severidade atribuída na Avaliação Heurística.

<b>Severidade</b>	<b>Tipo</b>	<b>Significado</b>
0	Sem-importância	Não é considerado, totalmente, um problema de usabilidade.
1	Cosmético	Problema apenas estético: não necessita ser consertado a menos que haja tempo extra, disponível no projeto.
2	Simple	Problema menor de usabilidade: o conserto desse problema deverá ter baixa prioridade.
3	Grave	Problema maior de usabilidade: é importante consertá-lo, para isso deverá ser dada alta prioridade.
4	Catastrófico	Catástrofe de usabilidade: é obrigatório consertá-lo, antes do produto ser divulgado.

Fonte: Nielsen (Nielsen, 1994b).

Os quatro avaliadores participaram da Avaliação Heurística. Foram encontradas no total 89 heurísticas violadas, nos três servidores avaliados, conforme Tabela 9.

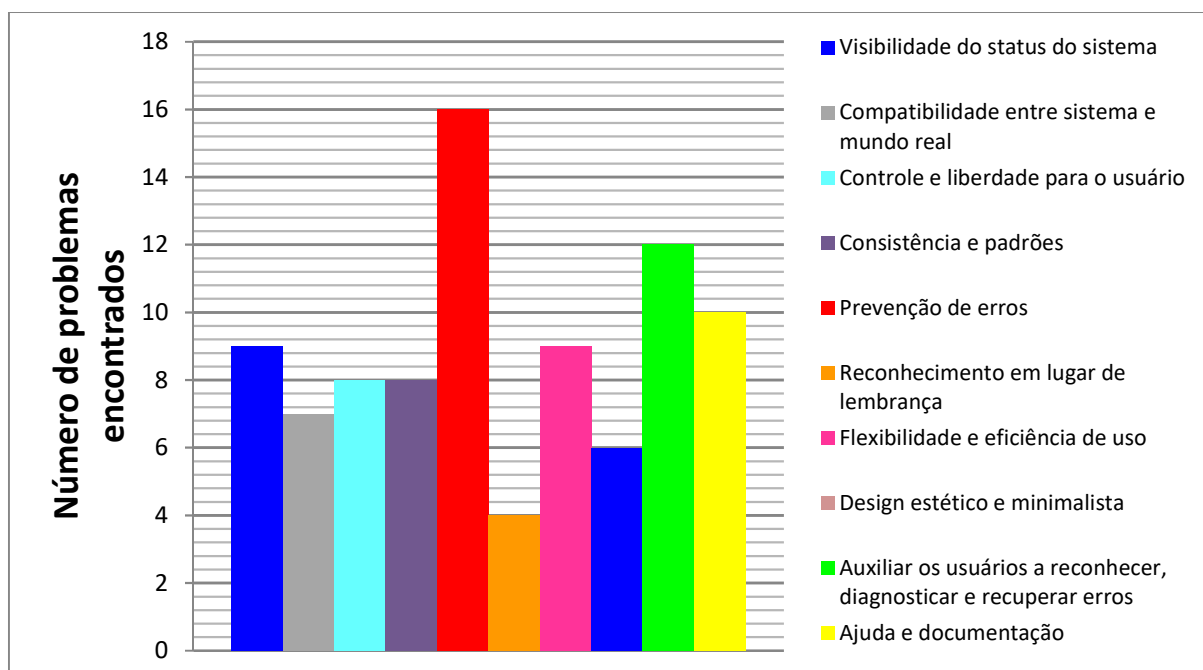
**Tabela 9.** Problemas de Usabilidade encontrados na Avaliação Heurística.

Avaliador	Problemas de Usabilidade				Total
	A	B	C	D <sup>1</sup>	
I-TASSER	3	12	3	x	18
QUARK	3	16	5	21	24 + 21=45
ROBETTA	2	13	11	x	26
TOTAL	8	41	19	21	89

<sup>1</sup>Os resultados da avaliação do inspetor D foram parciais, pois é relativa somente a avaliação do servidor QUARK.

O servidor I-TASSER foi o que apresentou menos erros, com 18 no total, o Robetta com 26 problemas apontados pelos especialistas e o servidor QUARK, que foi avaliado pelos inspetores (A, B e C), com 24 erros, que foram adicionados aos do inspetor D (21 erros), totalizando 45. O Inspetor D não conseguiu concluir a avaliação dos outros dois servidores a tempo de participar do agrupamento das informações. Porém sua opinião com relação ao servidor QUARK, como especialista da área de IHC, foi importante, pois detectou 21 erros de usabilidade nesta interface.

Podemos observar na Figura 9 que as duas heurísticas que mais foram citadas nas avaliações foram prevenção de erros e auxiliar os usuários a reconhecer, diagnosticar e recuperar erros.



**Figura 9.** Problemas de usabilidade encontrados pelos avaliadores. As heurísticas com mais violações foram Prevenção de erros, Prevenção de erros e auxiliar os usuários a reconhecer, diagnosticar e recuperar erros.

Foram encontrados erros de usabilidade para todas as Heurísticas definidas por Nielsen. O Condutor da avaliação compilou os resultados apresentados nas avaliações em um único documento, que resultou no documento de requisitos de software que definiu o protótipo, ambos apresentados nas próximas subseções.

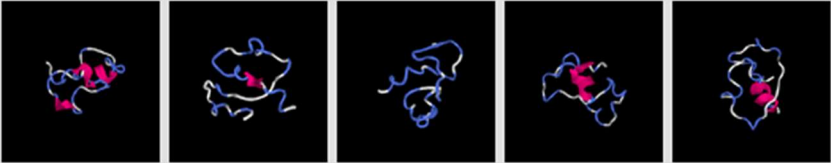
Além das duas heurísticas citadas anteriormente referentes aos erros de sistema, os problemas de usabilidade que mais ocorreram estavam relacionados com:

- 1) **Visibilidade do status do sistema:** Faltavam informações sobre onde o usuário se encontrava na interface. Não apareciam informações de navegação como o menu selecionado ou mapa de navegação.
- 2) **Flexibilidade e eficiência de Uso:** Erros referentes ao cadastro do usuário no servidor, na submissão de uma proteína e também quanto a forma de salvar os resultados da predição que foi enviada.
- 3) **Reconhecimento ao invés de lembrança:** Ocorreram problemas como por exemplo, links com instrução sobre o sistema abriam em outras páginas do navegador, e não ficavam visíveis na mesma página para o usuário.
- 4) **Controle e liberdade do usuário:** o usuário deve poder retornar ao estado inicial depois de estar em um estado não desejado (*desfazer e refazer*). Isso não ocorreu. Apesar de os sistemas apresentarem essa possibilidade, as funcionalidades relativas não estavam funcionando ou ocorreram erros durante sua execução.
- 5) **Consistência e padrões:** erros relativos ao *design* do sistema não seguir padrões básicos de interfaces seguindo os padrões do sistema. Em um dos servidores, por exemplo, o menu em diferentes partes do sistema troca de local (Figura 10).

Um dos erros mais frequentes encontrados nas heurísticas relacionava-se as mensagens de erro apresentadas nas interfaces dos servidores avaliados, como observado nos comentários:

- *“As mensagens de erro devem ser apresentadas em destaque, no formato padrão (janela) junto ao erro ocorrido. O que ocorriam eram mensagens em vermelho, mas que podem ser confundidas com outras mensagens de destaque na página”.*
- *“Há mensagens em vermelho no texto. Apesar de serem importantes, avisos poderiam ser apresentados de outra forma, para não distrair os usuários”.*
- *“Padronizar os avisos de erros. Os erros aparecem em diferentes formas na interface, em outras janelas, ou em formato texto igual ao de outras informações, como campos obrigatórios”.*
- *“Em caso de erros não há a possibilidade de se voltar a o estado inicial”.*

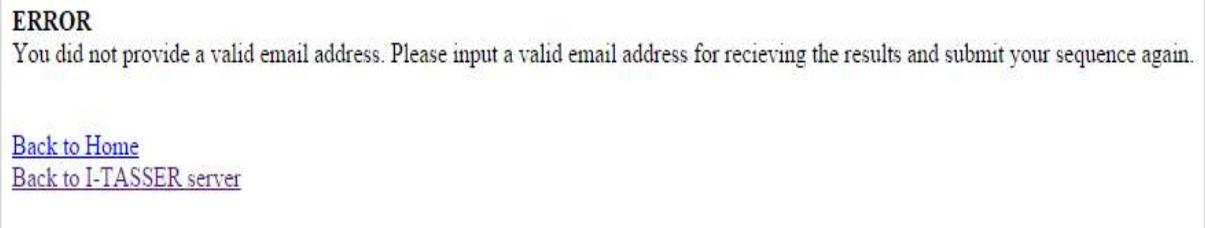
- “Os erros não apresentam cores ou símbolos diferenciados a fim de chamar atenção do usuário”.
- “Novamente a predição enviada deve ter a possibilidade de cancelamento. Em uma avaliação o usuário enviou uma submissão errada, além do sistema não detectar, ainda o usuário não conseguiu enviar outra até que essa fosse processada”.
- “As mensagens de erro não seguem um padrão. Algumas abrem em uma nova janela do navegador, em formato de texto, sem botões para conformação ou retornar ao estado anterior. As mensagens deveriam aparecer na mesma tela que ocorreu o erro, como padrão em qualquer sistema operacional”.

ID	Protein Name	Length	C-score	Estimated TM-score	Estimated RMSD(Å)	Submissio
<a href="#">S195328</a>	your_protein	504	NA	NA	NA	2014-10
This job is running and should be completed in approximately 50hr						
ID	Protein Name	Length	C-score	Estimated TM-score	Estimated RMSD(Å)	Submissio
<a href="#">S195327</a>	STR	59	-2.69	0.40±0.14	8.6±4.5	2014-10
						
ID	Protein Name	Length	C-score	Estimated TM-score	Estimated RMSD(Å)	Submissio
<a href="#">S195326</a>	your_protein	254	NA	NA	NA	2014-10
This job is running and should be completed in approximately 25hr						
ID	Protein Name	Length	C-score	Estimated TM-score	Estimated RMSD(Å)	Submissio
<a href="#">S195325</a>	D4R	467	NA	NA	NA	2014-10
This job is running and should be completed in approximately 50hr						
<a href="#">&lt;&lt;&lt;</a> <a href="#">1</a> <a href="#">2</a> <a href="#">3</a> <a href="#">4</a> <a href="#">5</a> <a href="#">6</a> <a href="#">7</a> <a href="#">8</a> <a href="#">9</a> <a href="#">10</a> <a href="#">11</a> <a href="#">12</a> <a href="#">13</a> <a href="#">14</a> <a href="#">15</a> <a href="#">16</a> <a href="#">17</a> <a href="#">18</a> <a href="#">19</a> <a href="#">20</a> <a href="#">21</a> <a href="#">&gt;&gt;&gt;</a>						
<a href="#">Home</a> <a href="#">Server</a> <a href="#">Queue</a> <a href="#">About</a> <a href="#">Statistics</a> <a href="#">Remove</a> <a href="#">Search</a>						

**Figura 10.** Menu do I-TASSER. O menu é apresentado em vários locais da interface e em diversos padrões diferentes. Nesta janela o mesmo foi apresentado na parte inferior da interface, o que não é comum. O ideal seria apresentá-lo sempre da mesma forma nas diferentes partes da interface.

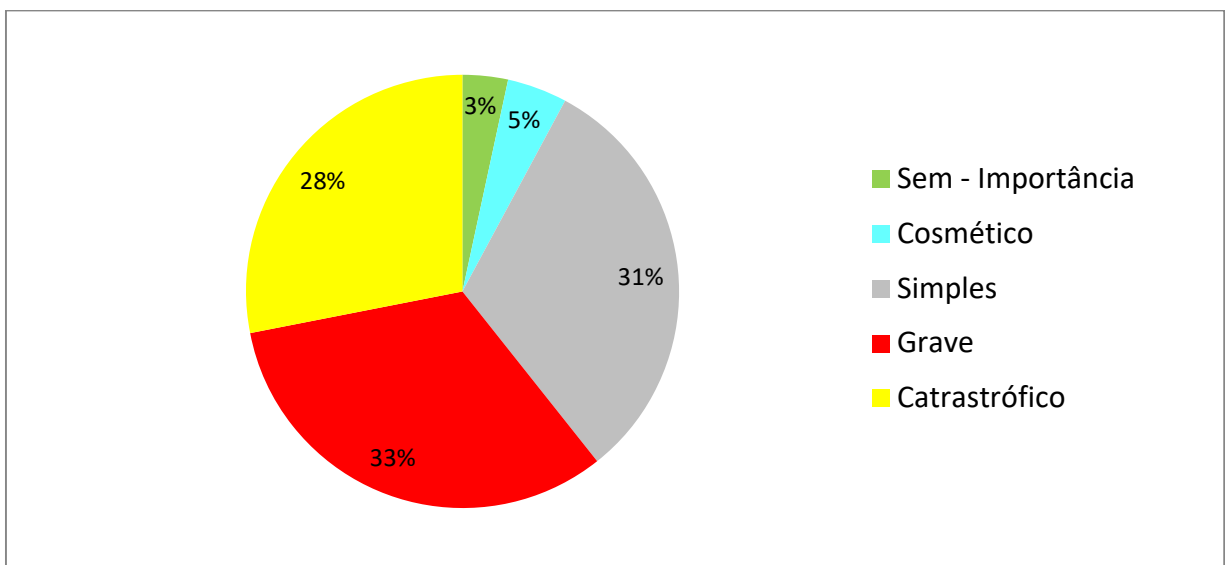
A exemplo disso apresentamos um problema encontrado em uma das interfaces (Figura 11). A mensagem de erro é aberta em outra página do navegador, sendo apresentada fora do local onde ocorreu o erro, não há um botão de confirmação para o usuário para saber se o mesmo compreendeu que ocorreu um erro, tão pouco para retornar ao estado anterior sem perder as informações. Além disso quando o usuário clica no link “*Back to Home*” o mesmo retorna a

*home page* do grupo de pesquisadores, e não para página inicial do servidor e isso pode confundir o usuário.



**Figura 11.** Uma das heurísticas violadas - Auxiliar os usuários a reconhecer, diagnosticar e recuperar erros. A mensagem de erro foi apresentada em uma nova janela do navegador, longe do local em que ocorreu (servidor I-TASSER).

Além das heurísticas encontradas, cada avaliador definiu o grau de severidade para cada problema de usabilidade (Figura 14). Quanto maior o número e o grau de severidade dos problemas encontrados menor a usabilidade da interface, e vice-versa (Winckler, 1999). Como podemos observar na Figura 14 mais da metade, 61% dos erros foram considerados graves e catastróficos, que na opinião dos avaliadores deveriam ser corrigidos, pois são extremamente importantes para o funcionamento da interface. Na opinião dos especialistas, 31% dos erros são simples e não precisam ser ajustados imediatamente, 33% foram considerados graves (alta prioridade para concerto) e 28% catastróficos, com isso demonstra-se o quanto a usabilidade destes sistemas precisa ser estudada e melhorada. Os problemas classificados como cosméticos e sem-importância ficaram em 8%. Apesar dos sistemas serem reconhecidos, funcionais e extremamente úteis para comunidade científica, os elementos de usabilidade precisam ser aprimorados.



**Figura 12.** Graus de severidade atribuídos nas avaliações dos servidores.



Na avaliação Heurística das interfaces dos servidores de predição as diferenças entre os especialistas em Informática/ IHC e os que são cientistas de outras áreas (biólogos, químicos, físicos) na forma como compreendem estas interfaces ficaram evidentes, conforme o esperado. Podemos perceber que os usuários de ferramentas de bioinformática são usuários extremamente focados nos resultados a serem obtidos das ferramentas, então, tendem a relevar certas deficiências das interfaces, desde que consigam obter seus resultados.

Outro detalhe a salientar é que os desenvolvedores de tais ferramentas muitas vezes não são da área da computação e, portanto, não tem formação voltada ao desenvolvimento de um software com qualidade no que tange a interface. Para demonstrar isso vamos comparar as respostas dos avaliadores A, B e D sobre os problemas apresentados na submissão da sequência. O especialista A da área da bioinformática classificou os erros como cosmético e simples:

*“O servidor não permite a modelagem de sequências de aminoácidos menores do que 20 e maiores do que 200. Ao preencher o campo correspondente à sequência de forma indevida, o servidor deveria verificar e indicar qualquer irregularidade. Atualmente essa verificação só é feita após a submissão do processo.”* – Simples (especialista A, bioinformática)

*“Ao preencher o campo correspondente à sequência de forma indevida, o servidor deveria verificar o tamanho e formatação da mesma e indicar qualquer irregularidade. Atualmente essa verificação só é feita após a submissão.”* – Cosmético (especialista A, bioinformática)

Já os avaliadores da área da IHC, classificou este problema como catastrófico:

*“Altere a sequência de aminoácidos, retirando caracteres especiais e o sistema se limitou a validar o e-mail, mesmo tendo colocado um e-mail diferente e em outro navegador. Atividade não realizada.”* – Catastrófico (especialista B, IHC)

*“Não há mensagens de erro ao ser colada uma sequência de proteína incorreta.”* – Catastrófico (especialista D, IHC)

Na avaliação do servidor QUARK por exemplo, com relação a heurística “auxiliar os usuários a reconhecerem, diagnosticarem e recuperar erros” os especialistas da área de bioinformática A e C não detectaram nenhum problema. Já a especialista B detectou 4 violações:

*“No campo relacionado ao e-mail é informado que deve ser incluído um e-mail acadêmico, entretanto, inventei um endereço de e-mail e este foi aceito...resubmeti colocando um e-mail @hotmail, ele detectou. Ou seja, esse controle parece existir para uma*

*relação de servidores de e-mail conhecidos. Se não há como definir o que é acadêmico e o que não é, não deveria haver essa restrição.* ” Catastrófico (especialista B, IHC)

*“Altere a sequência de aminoácidos, retirando alguns símbolos, e o sistema se limitou a retornar informações a respeito do e-mail.”* – Catastrófico (Especialista B, IHC)

*“Falta informação de que há uma limitação de gerar um Job por vez, até que a pendência desse seja solucionada”.* – Catastrófico (especialista B, IHC)

*“Falta informação do por que é necessário utilizar um e-mail acadêmico. Se o usuário não tem, não pode utilizar esse servidor.”* – Catastrófico (especialista B, IHC)

Dessa forma podemos perceber que os usuários de ferramentas de bioinformática são usuários extremamente focados nos resultados a serem obtidos das ferramentas, então, tendem a relevar certas deficiências das interfaces, desde que consigam obter seus resultados. Esse comportamento é visível nos resultados obtidos, visto que os avaliadores mais intimamente envolvidos com a área de bioinformática acharam poucos erros e os consideraram não tão relevantes, ao passo que especialistas em usabilidade e IHC encontraram muito mais erros, considerados graves e catastróficos.

#### 4.2.1 Avaliação adicional dos servidores de predição

Juntamente com a avaliação dos servidores de predição pelos especialistas, foi realizado a busca de problemas de usabilidade que podem ser iguais ou complementares aos especialistas. Esta avaliação adicional foi realizada por aluna de Pós-graduação em Ciência da Computação (Mestrado) com formação em Sistemas de Informação e Ciências Biológicas, que realizou disciplinas de Interação Humano-Computador e Bioinformática. Apesar de não ser especialista nas áreas relacionadas, sua formação nas duas áreas pode contribuir com este estudo. Sendo assim a Tabela 10 traz os problemas encontrados.

**Tabela 10.** Problemas adicionais encontrados nos servidores de predição.

Problema	Servidor
Os botões de voltar na página são somente do navegador.	Robetta
Os botões de atalho só acontecem em algumas telas.	Robetta
O campo de nome do Job é obrigatório e não há nenhum aviso quanto a isso.	Robetta
Há um Campo a ser marcado para escolher o tipo de predição, mas só há uma caixa de seleção.	Robetta

Para evitar erros de login deveria ser obrigatório primeiramente o login no servidor antes de submeter a predição, pois o usuário fica sem saber se está agindo corretamente.	Robetta
O sistema pede a proteína no formato FASTA e ao lado pede somente AA. O usuário pode ficar na dúvida se coloca os aminoácidos somente ou no formato fasta.	Robetta
O sistema pede no cadastro uma senha, mas na hora de submeter e utilizar, essa senha não é utilizada.	Robetta
A página dos resultados é cheia de campos, textos, não há um botão de busca, há vários, sendo que o usuário tem que lembrar muita coisa, sendo que fica difícil para o usuário achar sua predição e entender o que se trata cada coisa.	Robetta
Cadastrei um e-mail que já estava no sistema. O sistema acusou que o cadastro existia, mas não deu a opção de recuperar a senha e o cadastro.	Robetta
Se o usuário carrega a proteína no formato FASTA, pela escolha de arquivo, o sistema não aceita.	Robetta
Apesar da documentação estar bem completa e bem explicada, faltam elementos de desing, como botões de voltar, e a formatação do texto é um texto corrido.	Robetta
Não é informado para o usuário quanto tempo será necessário para predição.	QUARK
Há determinadas telas na interface que, ao serem acessadas como ajuda, não possuem botão de retornar ao estado anterior.	QUARK
Mensagens de erro em outra janela, não padronizada.	QUARK
Faltam botões ou os botões com funções diferentes são muito similares.	QUARK
A mensagem de erro quando se coloca um e-mail não acadêmico abre em outra página, sem botão voltar, sem padronização.	QUARK
É possível carregar qualquer arquivo como proteína na previsão e enviar ao servidor.	QUARK
Sequência FASTA informada erroneamente, sendo que o servidor não reconhece que a sequência está errada.	QUARK
Problema de erro com o visualizador das estruturas preditas, dando erro no navegador e não conseguindo visualizar as estruturas.	QUARK
O usuário ao entrar no fórum de ajuda, não tem a opção de retornar ao servidor. Além disso esse fórum é compartilhado com outros servidores do grupo, sendo necessário uma busca por problemas no Quark.	QUARK
O sistema não possui uma personalização de acordo com seus usuários. Possui uma tela única, que aceita como entrada a sequência FASTA da proteína, e gera uma saída, sem personalização da submissão.	QUARK
Muita informação na mesma tela, ajuda, documentação, mensagem de erro, tudo misturado para o usuário.	QUARK

Mensagens de campos obrigatórios, iguais aos de aviso de erros.	QUARK
Os textos de ajuda estão distribuídos ao longo de toda interface, sendo orientações em texto, e algumas delas não explicativas e confusas.	QUARK
A maioria das páginas da interface não possui botões ou ícones simbolizando ações familiares ao usuário como botão voltar, ir, home, salvar etc. Somente possui botões na página inicial do servidor.	I-TASSER
O sistema permite que o usuário cole sua sequência FASTA e preencha seu e-mail, e submeta mesmo sem ser um usuário cadastrado. A mensagem de erro abre em uma nova janela. O usuário a partir da mensagem pode se cadastrar, mas se resolver voltar a tela inicial caso seja um usuário cadastrado e continuar a fazer sua solicitação, clicando na opção Back to I-TASSER server perdendo o que já foi preenchido. Não existe botão voltar para o mesmo estado do sistema.	I-TASSER
A página da interface que contém o download do pacote de aplicativos da ferramenta, está confusa. Há palavras sublinhadas que podem levar o usuário pensar que são links, e não são, e outras que são. Não há botões de retornar a página anterior, e o menu que é apresentado é a do grupo, e não do servidor.	I-TASSER
Tanto para submeter a predição quanto para submeter alguma coisa para o fórum tem-se que realizar um cadastro. Nos dois casos, o usuário tem que cadastrar seu e-mail, em cada situação, e em cada uma delas recebe uma senha diferente, e um nome de usuário diferente. Em um dos casos a senha pode ser recadastrada, e no outro não.	I-TASSER
O usuário recebe a senha para cadastro por e-mail, mas não tem a possibilidade de trocá-la ou cadastrá-la de acordo com sua vontade. No e-mail recebido a senha ainda não é destacada do texto, podendo a vir confundir o usuário.	I-TASSER
A mensagem de erro ao não se colocar no formulário o e-mail do usuário, abre em uma nova janela do navegador, em formato de texto, não seguindo o padrão do sistema de mensagens de erro.	I-TASSER
O menu superior do servidor muda constantemente dependendo da página que o usuário encontra, não mantendo um padrão. O mesmo até vem parar na parte inferior da página em determinado momento.	I-TASSER
O home em várias telas do servidor, volta para home do grupo e não para a home do servidor, vindo a confundir o usuário.	I-TASSER
O menu da interface não é único, padronizado, possuindo vários formatos, localização e é diferente conforme se percorre a interface. O tamanho das fontes, títulos muda conforme a página da interface.	I-TASSER
Há uma opção para os usuários entrarem num fórum do servidor. Ao clicar neste menu, o usuário é redirecionado para uma outra página, com padrão bem diferentes das demais, e para participar do fórum pede para fazer login. Não há uma explicação sobre se o usuário deve utilizar o login cadastrado anteriormente para o acesso ao servidor, o qual o mesmo recebeu uma senha, ou deve realizar um novo cadastro. Isso pode fazer com que o usuário erre em tentar acessar o fórum.	I-TASSER
O usuário quando realiza o cadastro no site do servidor recebe uma mensagem de confirmação, em uma nova janela, que confirma que o usuário foi cadastrado com	I-TASSER

sucesso, porém não há opção de voltar a página anterior, e nenhum outro botão ou ícone para que o usuário possa sair da página. Não há botões acessíveis. O campo instituição ficou em branco.	
Na tela inicial o usuário precisa digitar o e-mail e uma senha para enviar a predição. Mas se o mesmo colocar a sequência e submeter, abre uma mensagem de erro em outra página, solicitando as informações necessárias, mas só tem dois botões para voltar, a home do grupo ou a tela inicial do I-TASSER, sendo que o usuário perde aquilo que foi preenchido.	I-TASSER
O sistema permite que o usuário cole sua sequência FASTA e preencha seu e-mail, e submeta mesmo sem ser um usuário cadastrado. A mensagem de erro novamente abre em uma nova janela.	I-TASSER
Na tela inicial só é avisado para o usuário que ele deve ter um cadastro prévio de que tem que ser um usuário cadastrado após o mesmo submeter algo, ou como um pequeno aviso de que tem que ter senha em um pedaço do formulário, não ficando claro inicialmente que precisa ser um usuário cadastrado.	I-TASSER

### 4.3 Documento de requisitos

Os requisitos de usabilidade de qualquer produto, incluindo aplicações baseadas na Web, devem ser identificados, se o produto a ser projetado tem o objetivo de atender às necessidades do usuário (Ssemugabi e de Villiers, 2007). Através da aplicação de testes de usabilidade é possível definir um conjunto de requisitos de usabilidade que potencializam a qualidade do sistema (Tavares e Cavalcanti, 2001).

Com este enfoque, é que foram realizados testes de usabilidade para o levantamento de requisitos de software e definição do protótipo. Dessa forma, os requisitos levantados levaram em conta as funcionalidades do sistema quanto aos aspectos referentes a uma aplicação interativa e com boa usabilidade para seus usuários.

Preece et. al (2005) definem que os requisitos consistem em uma declaração sobre o produto pretendido que especifica o que ele deve fazer ou como deve operar. Requisitos funcionais definem o que o sistema deve fazer. Já os Requisitos não funcionais indicam quais são as limitações do sistema em seu desenvolvimento (Preece et al., 2005).

Além dos requisitos obtidos nas avaliações heurísticas, foram realizadas reuniões de complementação da Especificação dos Requisitos do Software com membros da equipe LABIO para o desenvolvimento do projeto, contribuindo também com esta pesquisa.

Estes requisitos foram compilados e agrupados em um único documento que se encontra no material complementar deste artigo (APÊNDICE F). O relatório final, além de sugerir melhorias a serem realizadas nos servidores de predição de estruturas 3D de proteínas, buscando

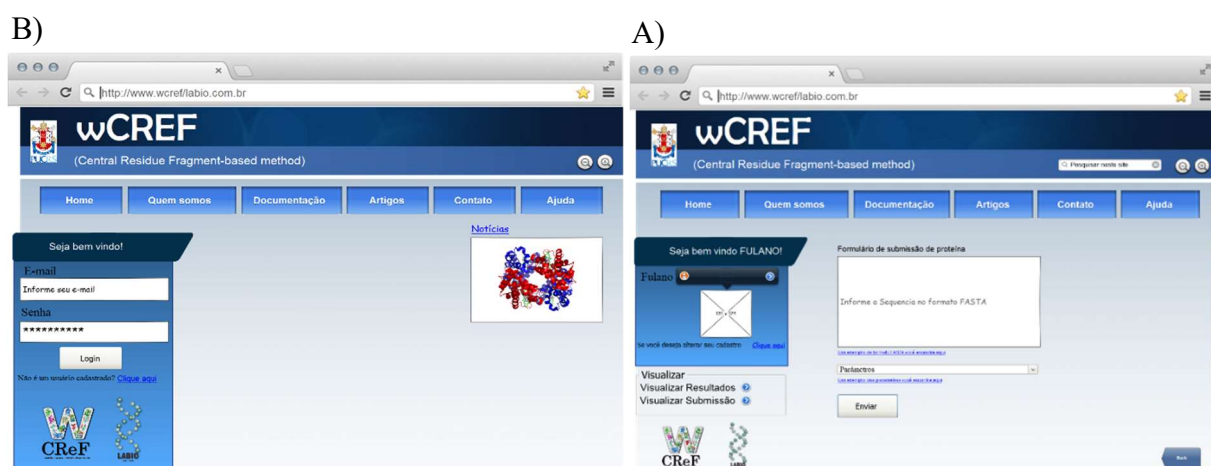
melhorar sua usabilidade, tem como objetivo principal descrever as funcionalidades para a construção da interface do wCReF.

#### 4.4 Descrição do protótipo

Na modelagem de interfaces existem duas fases do seu desenvolvimento imprescindíveis para um projeto, que são: a sua especificação de requisitos e o seu protótipo (Oliveira e Lima, 2013). Definido os requisitos na etapa anterior, então partimos para a modelagem do protótipo.

Segundo Preece et. al. (2005), um protótipo é uma representação limitada de um *design* que permite aos usuários interagir e explorar sua conveniência. Um protótipo pode ser desde um *storyboard* de papel a uma parte complexa de um software, podendo ser feito manualmente ou através de ferramentas de software (Preece et al., 2005). Este trabalho desenvolveu um protótipo de alta fidelidade para o wCReF, a partir dos requisitos de software obtidos nas avaliações.

Primeiramente, um desenho do Protótipo que foi desenvolvido com ferramenta Pencil (<http://pencil.evolus.vn/>) considerando os requisitos do documento de software (Figura 13). Este primeiro esboço serviu como guia, juntamente com o documento de requisitos de software para o desenvolvimento do protótipo. Cabe deixar claro aqui, que o desenho na ferramenta Pencil apenas serviu como um primeiro modelo para o desenvolvimento do protótipo, mas não foi utilizado nas avaliações.



**Figura 13.** Esboço da interface feito na ferramenta Pencil, a partir das definições do documento de requisitos de software. A) Página principal com Login. B) Área de submissão de proteínas.

Foram utilizados os seguintes recursos para o desenvolvimento do protótipo:

- Linguagem de programação: A aplicação do wCReF foi desenvolvida em linguagem HTML (HyperText Markup Language) na versão 5. O HTML é uma linguagem de marcação de texto utilizada para produzir páginas na Web (Graham, 1995). Os Documentos HTML podem ser interpretados por navegadores da internet. Para aplicar um estilo padrão em cada página, com designer agradável ao usuário, ao invés de formatar o próprio documento, cria-se um link (atalho) para uma página que contém os estilos, o que facilita futuras modificações no layout da página, provendo assim a separação entre o formato e o conteúdo de um documento Web. Cascading Style Sheets (CSS) é uma linguagem de folha de estilo utilizada para controlar a apresentação visual em páginas da Web, desenvolvidas por uma linguagem de marcação, como HTML ou XML (Meyer, 2000). O CSS permite a página com um estilo padrão e agradável ao usuário.
- Microframework: Como o Método CReF foi desenvolvido em linguagem Python (método) e a interface Web em HTML/CSS (aplicação), sendo necessário a utilização de uma ferramenta para fazer a ligação entre o método e o aplicativo. Para este fim, se definiu o Flask (Ronacher et al., 2010) um microframework um framework minimalista para desenvolvimento de aplicações Web. O Flask é um aplicativo alimentado por banco de dados. Então tudo que for submetido ao wCReF fica armazenado em um banco de dados. Sua base é formada por Werkzeug e Jinja 2. Além de possuir um código com alta qualidade nos quesitos de legibilidade, ele proporciona uma liberdade para a estruturação de aplicações da maneira que o desenvolvedor desejar, com padrões de projeto e extensões que dão a certeza de a ferramenta pode crescer sem problemas.
- Bancos de dados: o que for submetido ao wCReF fica armazenado em um banco de dados para a consulta, processamento e devolução dos resultados, aos usuários do método. Um banco de dados é um conjunto de arquivos relacionados entre si (Chu, 1983). É uma coleção de dados operacionais armazenados sendo usados pelos sistemas de aplicação de uma determinada organização (Date, 1985). O sistema que permite a definição, construção e manipulação de um banco de dados é o sistema de gerencia de banco de dados SGBD (Sistema de gerenciamento de banco de dados).
- O MySQL (Oracle Corporation, 1995) é um sistema de gerenciamento de banco de dados (SGBD), que utiliza como interface a linguagem de consulta estruturada (SQL - Structured Query Language). MySQL é um SGBD relacional que é largamente utilizado em aplicações para a Internet. É o mais popular entre os bancos de dados com

código-fonte aberto. O MySQL é uma alternativa atrativa porque, mesmo possuindo uma tecnologia complexa de banco de dados, seu custo é bastante baixo. Tem como destaque suas características de velocidade, escalabilidade e confiabilidade, o que faz com que ele seja adotado por departamentos de TI (Tecnologia da Informação), desenvolvedores Web e vendedores de pacotes de softwares (Niedeauer, 2006).

As Figuras 14 e 15 mostram as telas iniciais do protótipo desenvolvido. A Figura 14 apresenta a tela com página inicial do sistema (home). Nesta página encontram-se as informações iniciais sobre o wCReF.

The image shows the home page of the wCReF web application. At the top, there is a navigation bar with icons and labels for Home, Cadastro, Submissões, Documentação, Equipe, and Fale Conosco. Below this is a blue banner with the wCReF logo and the text 'Central Residue Fragment-based Method'. A 'Login' button is located in the top right corner of the banner. The main content area is divided into two columns. The left column contains a 'Login' section with a form for entering an email address (pre-filled with 'fulanodetal@email.com') and a password, a 'Login' button, and links for 'Esqueceu sua senha?' and 'Registre-se'. Below the login form is a 'Links Úteis' section with logos for RCSB PDB, NCBI GenBank, and CASP. The right column contains the title 'CReF(Central Residue Fragment-based method)' and a welcome message 'Bem vindo ao wCReF!'. It includes a paragraph about the challenges of structural bioinformatics and the CReF method, followed by a paragraph about the server's development and its participation in CASP events. At the bottom of the right column are three links: 'Saiba como utilizar', 'Precisa de ajuda?', and 'Referências'.

**Figura 14.** Tela inicial da interface do protótipo desenvolvido (Home). O usuário realiza login no servidor e é direcionado para a tela de submissão.

Para que o usuário possa submeter uma predição, o mesmo deve realizar o cadastro no sistema, e realizar login, sendo direcionado para página inicial do servidor para submissão (Figura 15).



**Figura 15.** Tela inicial do wCReF. Esta é a tela principal do servidor, onde o usuário submete sua predição.

Cabe destacar que o protótipo desenvolvido já apresentava as características principais da interface, definidas a partir do documento de requisitos. Então todos os recursos para seu desenvolvimento, são os mesmos aplicados na versão final da interface web. Isto foi importante para a análise da interface realizada nos testes com os usuários finais.

Foi possível, portanto, encontrar erros de usabilidade no protótipo, aprimorar o que havia sido desenvolvido, e com isto obter a primeira versão da interface web. Este protótipo foi utilizado para os testes com os usuários finais por meio dos questionários. A descrição completa da interface do wCReF, com as correções após as avaliações por questionários, encontra-se disponível no próximo capítulo.

#### 4.5 Avaliação por questionários

Foi adotado a avaliação de questionários para perceber a opinião dos usuários no uso do protótipo, visando a realização de melhorias e correção de erros.

Para estruturar as avaliações por questionários utilizamos como modelo o questionário de Ssemugabi (Ssemugabi, 2009). O questionário de Ssemugabi é utilizado para medir a usabilidade em interfaces Web, sendo baseado nas heurísticas de Nielsen. Ele se divide em duas

partes: uma parte de questões baseadas nas 10 Heurísticas de Nielsen, e outra questões voltadas para a área de *E-learning*. Como neste estudo o contexto de aplicação é a bioinformática, realizamos uma adaptação na segunda parte do questionário, cujo o foco é a usabilidade dos servidores de predição. O questionário aplicado neste trabalho encontra-se no APÊNDICE G.

O questionário foi apresentado aos participantes da seguinte forma:

- Apresentação da pesquisa: importante para motivar o usuário através de uma prévia descrição da pesquisa e os benefícios de sua contribuição.
- Solicitação de cooperação: através do termo de consentimento livre-esclarecido.
- Instruções: para o preenchimento e o cenário de uso.

O conteúdo do questionário referente ao levantamento das informações foi dividido em cinco partes:

- 1) Perfil do usuário:** Como a bioinformática é uma área interdisciplinar, buscamos saber qual o perfil do usuário que realizou os testes e sua experiência na área, abrangendo 7 questões.
- 2) Concepção da interface:** 50 questões fechadas do questionário de Ssemugabi, baseadas nas heurísticas de Nielsen (Ssemugabi e de Villiers, 2007), mais uma questão para cada das 10 heurísticas livre para apontamentos de problema de usabilidade, totalizando 60 questões.
- 3) Web design:** 8 questões (7 fechadas e 1 aberta) envolvendo a simplicidade de navegação do site, organização e estrutura, também baseadas no questionário de Ssemugabi.
- 4) Questões da área da bioinformática:** Foram criadas aqui novas questões, buscando analisar a usabilidade com enfoque em bioinformática, e também mais especificamente em servidores de predição de estruturas de proteínas, totalizando 19 questões fechadas e 2 abertas. Essa categoria e seus respectivos critérios de usabilidade criados para este presente estudo são apresentados na Tabela 11.
- 5) Conclusão:** 5 questões de fechamento da pesquisa e 2 abertas para apontamentos de aspectos positivos e negativos encontrados no sistema.

Para compilação dos resultados, ficou definido que seriam atribuídos a cada resposta dos questionários valores segundo a Escala de Likert (Likert, 1972) (Tabela 12).

Portanto, o questionário possui 96 questões fechadas que podem ser respondidas com a Escala de Likert e 15 questões abertas onde os usuários poderão relatar sua opinião sobre o sistema, as vantagens, desvantagens e sugestões, para a melhoria do protótipo.

**Tabela 11.** Questões de satisfação de usabilidade para a área de bioinformática em servidores de predição de estruturas de proteínas.

<b>Relevância do conteúdo do site para a área e Bioinformática</b>	
1	O conteúdo do site segue os padrões de aplicações em bioinformática.
2	As informações são relevantes para o desenvolvimento das atividades no servidor.
3	O conteúdo está no nível apropriado para minha compreensão.
4	Está claro quais os materiais que tem direitos autorais e quais não possuem.
5	O conteúdo do servidor cita as devidas referências do material que é utilizado.
<b>Clareza de metas, objetivos e resultados</b>	
6	Eu compreendi quais os objetivos do servidor.
7	Eu compreendi a metodologia utilizada no wCReF.
8	Consigo realizar uma submissão de uma proteína-alvo satisfatoriamente.
9	Se sou um usuário avançado, consigo compreender os parâmetros opcionais na submissão do wCReF.
10	Os parâmetros opcionais do wCReF estão de acordo com os utilizados por aplicações relacionadas que eu conheço.
11	Consigo encontrar com clareza as informações sobre os resultados.
12	O tempo de resposta da minha submissão está de acordo com o que eu esperava.
13	Os resultados estão apresentados de maneira clara.
14	Os resultados estão apresentados de maneira concisa.
15	Os resultados estão de acordo com aquilo que eu esperava.
16	Consigo salvar meus resultados.
17	Os resultados da submissão são comunicados a mim sem eu estar conectado ao servidor, por outros meios, como por exemplo e-mail.
18	As estatísticas me fornecem informações úteis.
19	Como pesquisador da área posso enviar sugestões, reclamações e tirar dúvidas com a equipe responsável pelo servidor.

**Tabela 12.** Números atribuídos as respostas segundo a Escala de Likert.

Respostas	Grau	
Concordo plenamente	5	Afirmação positiva
Concordo	4	
Talvez	3	
Discordo	2	
Discordo Plenamente	1	Afirmações negativa

#### 4.5.1 Resultado da avaliação por questionários realizadas pelos usuários

Devemos lembrar que o objetivo principal de uma avaliação é melhorar a interface e não apenas estimar o quanto ela é boa ou ruim. Pode-se dizer, então, que uma boa avaliação de usabilidade não é aquela que apenas identifica os problemas de usabilidade, mas que auxilia a equipe de desenvolvimento a solucioná-los e a melhorar a interação do usuário com a aplicação (Campos e Matias, 2012)

Tendo isso em mente, após o desenvolvimento do protótipo, foram realizadas avaliações de usabilidade com os usuários finais. Para isto, foram convidados a participar deste estudo alunos da disciplina de Bioinformática dos cursos de Pós-Graduação em Ciência da Computação (PPGCC), Pós-Graduação em Pediatria e Saúde da Criança, Pós-Graduação em Zoologia, Pós-Graduação em Medicina e Ciências da Saúde, Pós-Graduação em Biologia Celular e Molecular, Mestrado Profissional em Biotecnologia Farmacêutica, e também alunos da Graduação em Ciências Biológicas, todos cursos pertencentes a PUCRS. Além disso, um dos especialistas (Doutoranda em Ciência da Computação) que realizou a Avaliação Heurística, também respondeu ao questionário (identificado como usuário D).

Todos os avaliadores receberam o termo de consentimento livre e esclarecido (APÊNDICE E).

Cabe salientar que a busca de perfis de diferentes usuários é importante, pois se trata da avaliação de uma ferramenta para o uso em bioinformática, área possui pesquisadores vindos de diferentes áreas de conhecimento. Assim foram realizados testes por 11 usuários com o perfil apresentado na Tabela 13.

**Tabela 13.** Perfil dos participantes do questionário de avaliação

<b>Usuário</b>	<b>Área de estudo</b>	<b>Experiência na área de Bioinformática</b>	<b>Uso de aplicações Web em bioinformática</b>
A	Pós-Graduação em Medicina e Ciências da Saúde; Graduação em Farmácia	Menos de 6 meses.	Semanalmente
B	Pós-Graduação em Ciência da Computação (PPGCC); Graduação em Engenharia da Computação	Menos de 6 meses.	Não utiliza com muita Frequência
C	Pós-Graduação em Biologia Celular e Molecular; Licenciatura em Biologia	Entre 2 e cinco anos	Utiliza diariamente

D	Pós-Graduação em Ciência da Computação (PPGCC); Graduação em Ciência da Computação	Entre um ou 2 anos	Semanalmente
E	Pós-Graduação em Ciência da Computação (PPGCC); Bacharel em Sistemas de Informação	Menos de 6 meses.	Semanalmente
F	Pós-Graduação em Zoologia;	Menos de 6 meses	Não utiliza com muita Frequência
G	Pós-Graduação em Pediatria e Saúde da Criança Graduação em Ciências Biológicas	Menos de 6 meses	Semanalmente
H	Pós-Graduação em Ciência da Computação (PPGCC); Bacharel em Matemática	Menos de 6 meses	Semanalmente
I	Pós-Graduação em Zoologia; Graduação em ciências biológicas	Entre 1 e dois anos	Mensalmente
J	Mestrado Profissional em Biotecnologia Farmacêutica; Graduação em Farmácia		
K	Graduação em Ciências Biológicas	Menos de 6 meses	Semanalmente
L	Graduação em Ciências Biológicas	Menos de 6 meses	

Também foi solicitado para que cada um dos participantes respondesse qual sua experiência como usuário na área da bioinformática estrutural, mais especificamente na área de predição de estruturas de proteínas (Tabela 14).

**Tabela 14.** Perguntas relacionadas a experiência do usuário com a predição de estruturas de proteínas

Pergunta	A	B	C	D	E	F	G	H	I	J	K	L
Você sabe o que é predição de estruturas de Proteínas?	S	S	S	S	S	S	S	S	S	S	S	S
Você conhece algum servidor on-line de predição de estruturas de proteínas	S	N	S	S	N	S	S	N	S	S	S	N

S = Sim, N= Não

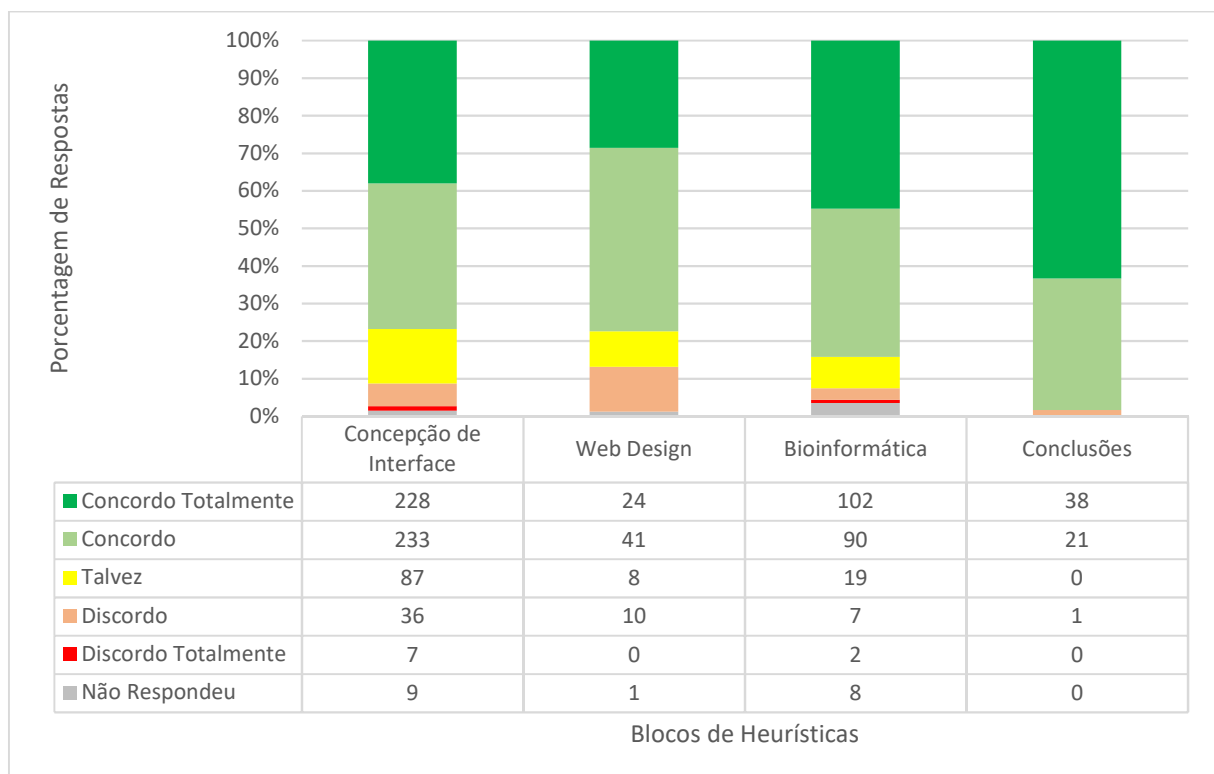
Todos os usuários afirmaram, saber o que é a predição de estruturas de proteínas, sendo que 4 destes não conheciam um servidor online de predição de estruturas de proteínas. Dos 12 participantes, apenas o usuário K não citou alguma interface Web de seu conhecimento que utiliza e que conhece para se trabalhar em bioinformática.

As aplicações conhecidas citadas nos questionários foram: PDB, Swiss-Model, NCBI-Blast, ClustalW, Net Primer, Proscan, Gene Ontology, MAFFT, PRAMK, Data Monkey e os servidores I-TASSER, QUARK, Swiss-Model, Robetta, Phyre.

A avaliação de usabilidade foi realizada pelos usuários, preenchendo o questionário de satisfação de Ssemugabi (adaptado), contendo 96 questões definidas como aplicáveis ao experimento conforme inspeção do avaliador.

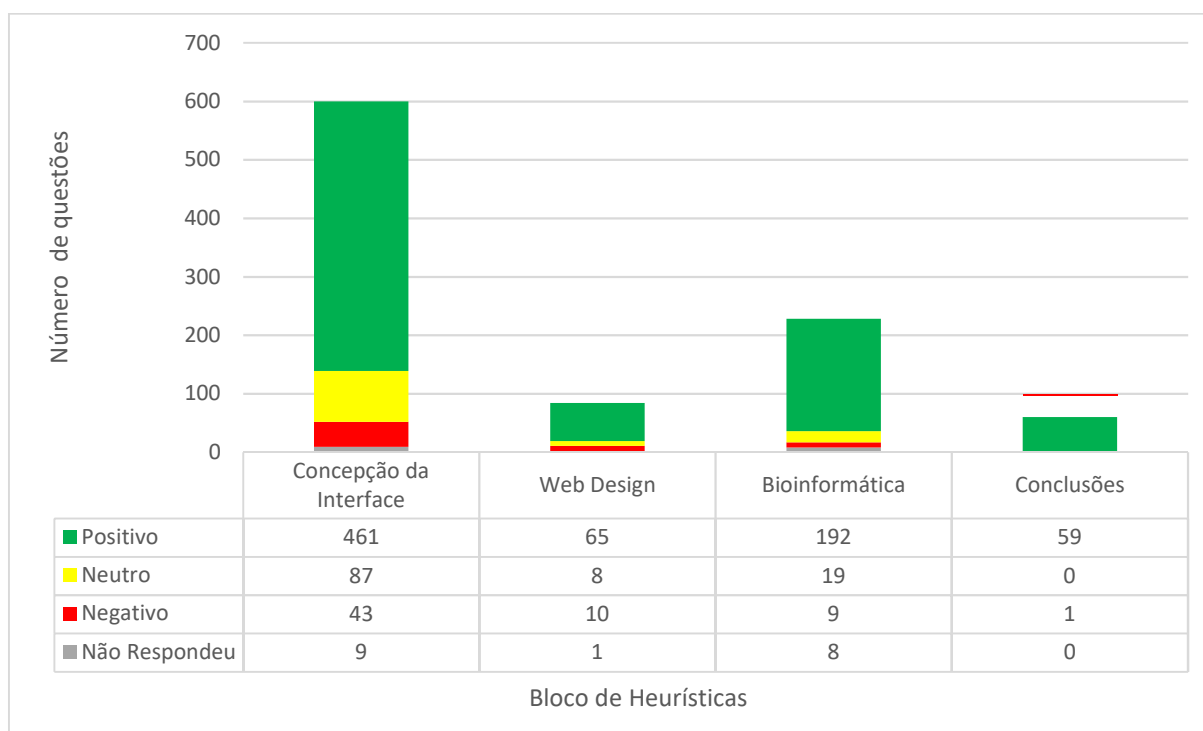
Através do valor atribuído a cada questão foi possível quantificar a concordância ou discordância dos usuários quanto aos aspectos específicos de usabilidade. Quanto maior for o valor em cada questão, maior é a concordância com a questão de usabilidade.

O gráfico a seguir (Figura 16) contém as 972 respostas das questões fechadas, incluindo aquelas que foram deixadas em branco. As respostas foram separadas segundo os blocos que foram apresentadas no questionário de satisfação (a concepção da interface traz as perguntas referentes às dez Heurísticas de Nielsen. *Web Design*, questões referentes a interface conforme o questionário de Ssemugabi. Bioinformática as questões criadas para avaliar a satisfação de acordo com a área do domínio e as conclusões).



**Figura 16.** Respostas dos questionários. As questões estão separadas por blocos segundo as Heurísticas: Concepção da Interface (Heurísticas de Nielsen), *Web design*, bioinformática e conclusões.

Pela opinião dos usuários podemos verificar que as questões classificadas com score mais alto, ficaram em torno de 80% do percentual de respostas, chegando no total de 99% das respostas positivas na conclusão. Do total das 972 respostas, 392 foram classificadas com melhor score (concordo plenamente) e 385 Concordo, 114 respostas foram neutras (Talvez). As negativas, as quais abordaremos a seguir, 54 foram marcadas como discordo e 9 discordo plenamente. Respostas em branco foram 18. O Total de pontos positivos, negativos e neutros são descritos na Figura 17.



**Figura 17.** Total de respostas Positivas, Negativas e Neutras por tópico. As questões positivas foram as classificadas com “Concordo Plenamente” e “Concordo”. As neutras com “Talvez”. As negativas com “Discordo” e “Discordo Plenamente”, e as sem respostas marcadas.

Analisamos que a construção da interface do wCReF respeitando os requisitos levantados na Avaliação Heurística foi o que contribuiu para a maioria das respostas positivas por parte dos usuários. Porém, como o nosso objetivo é disponibilizar uma interface desenvolvida para garantir a satisfação do usuário através da usabilidade, os pontos negativos foram importantes para corrigir esses erros e diminuir os problemas de interação.

Muitas respostas foram também consideradas neutras, principalmente na concepção da interface 87 de 600. Nas avaliações percebeu-se que os usuários utilizaram as neutras para classificar uma resposta que o mesmo não se considerava capaz de responder, ou porque não ocorreu nenhum fato que justificasse sua resposta.

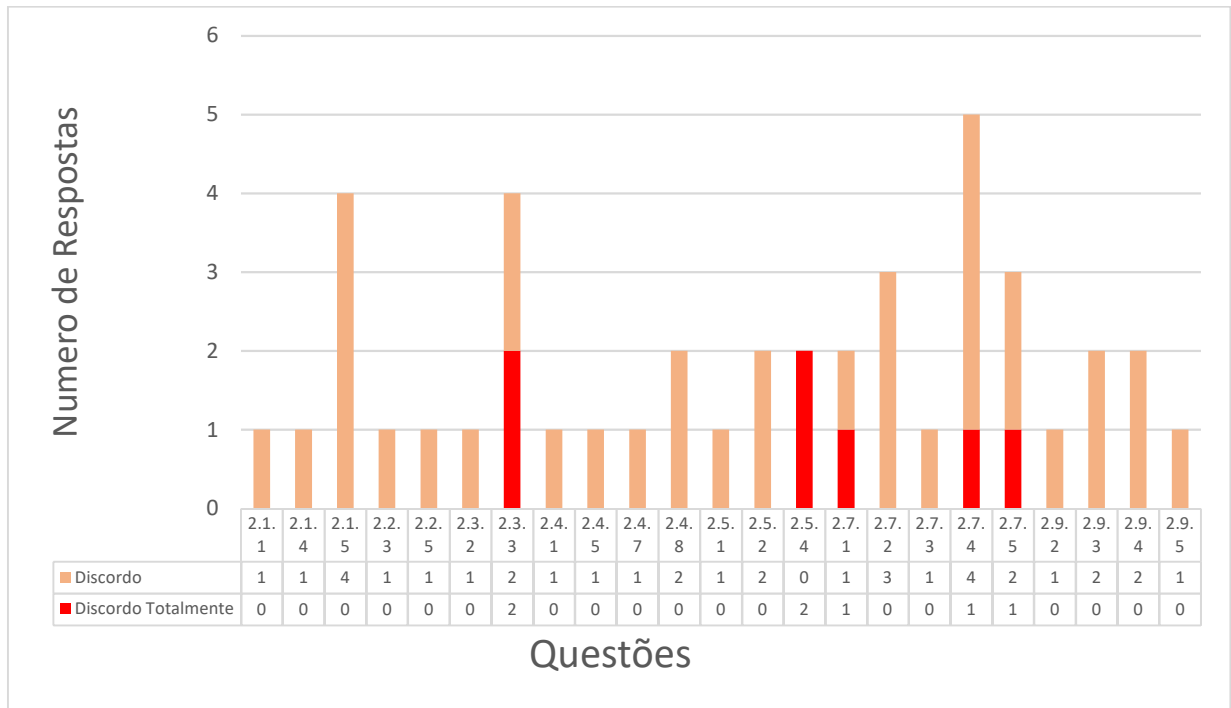
Vamos utilizar o exemplo do usuário A: O mesmo marcou toda a sessão de reconhecimento, diagnóstico e recuperação de erros com talvez (5 respostas) e relatou que como não ocorreu nenhum erro durante seu teste, o mesmo não se julgava capaz de responder. O correto neste caso seria ter uma opção para o usuário como “Não se aplica”, a qual não existe na lista de opções do questionário de satisfação que foi utilizado. Sugerimos que para novas aplicações deste questionário, uma sexta opção seja inserida:

- Concordo plenamente    Concordo    Talvez    Discordo    Discordo plenamente  
 Não se aplica

#### 4.5.2 Pontos considerados negativos na avaliação de usabilidade

Os pontos considerados negativos na avaliação de usabilidade foram analisados para identificar problemas de usabilidade na interface do wCReF. Os pontos considerados negativos são os que possuem valor entre 1 e 2 na Escala de Likert, e que correspondem às respostas Discordo Totalmente (1 ponto), Discordo Parcialmente (2 pontos).

Pelos valores atribuídos, foram separadas as questões classificadas como negativas de acordo com cada bloco de Heurísticas do questionário de avaliação. As primeiras Heurísticas apresentadas são as com relação as 10 Heurísticas de Nielsen, que tem haver a com a concepção da interface (Figura 18).



**Figura 18.** Número de questões negativas avaliadas na concepção da interface. Esse bloco de questões é referente as 10 heurísticas de Nielsen que foram consideradas com problemas de usabilidade.



Prioridade de correção dos problemas do protótipo foi definida da seguinte forma:

- Uma resposta: baixa prioridade e verificação (3).
- Duas respostas: média prioridade e correção (2).
- Três respostas ou mais: alta prioridade e correção imediata (1).

Os problemas apresentados foram corrigidos. A categorização foi realizada para definir a prioridade das correções a serem feitas na interface web. Dessa forma um *checklist* com os problemas que foram corrigidos são apresentados na Tabela 15 com sua respectiva solução. Os problemas relatados foram os que foram descritos pelos usuários na questão aberta ao final de cada seção.

**Tabela 15.** Problemas encontrados nas avaliações dos questionários considerados de Média e Alta prioridade de correção.

<b>Heurística: Visibilidade do Status do sistema</b>		
Número	Questão	Grau
2.1.5	O sistema não reage de forma que me surpreenda ou não faz nada inesperado.	1
Problema:	<i>“O sistema não apresentou erro quando submetido uma sequência fora do formato FASTA” (Usuário I).</i>	
Solução:	Correção do script de validação do formato FASTA.	
Problema:	<i>“O sistema não me redireciona para as submissões depois de uma submissão” (L).</i>	
Solução:	Foi adotado a sugestão de que após uma submissão o usuário é redirecionado para aba de submissões para acompanhar o status da predição.	
Problema:	<i>“O visualizar carregou a proteína exemplo e não meu resultado” (A).</i>	
Solução:	Correção do erro de visualização.	
<b>Heurística: Controle e liberdade para o usuário</b>		
Número	Questão	Grau
2.3.3	Quando eu cometo um erro eu não posso sair do sistema usando um botão de saída de emergência claramente sinalizado.	1
Análise	As mensagens de erros possuem um botão de fechar. Como a aplicação do wCReF é uma aplicação Web, o próprio navegador também possui o botão de fechar a aplicação.	
<b>Heurística: Consistência e Padrões</b>		

Número	Questão	Grau
2.4.8	Há consistência no uso do tipo e tamanho de fontes	2
Problema:	<i>“A interface tem muitos tipos e tamanhos de fontes”</i> (Usuário F).	
Solução:	As fontes dos textos foram padronizadas em fonte: P – Parágrafos: verdana, sans-serif, tamanho 14px (Pixel).	
Problema:	<i>“Em visualização de resultados algumas coisas estão em negrito e outras não, não seguindo uma sequência de relevância ao meu ver”</i> (E).	
Solução:	Os títulos e fontes da visualização foram revisados e padronizados: H1 – Títulos Principais: fonte - normal 20px (Pixel) <i>“OpenSans-ExtraBoldItalic”</i> ; H2, H3 – Títulos Secundários: fonte - normal 1.8px (Pixel) <i>“OpenSans-ExtraBoldItalic”</i> ;	

### Heurística: Prevenção de erros

Número	Questão	Grau
2.5.4	Sou requisitado a confirmar minhas entradas antes de levar adiante ações potencialmente perigosas como a de apagar	2
Problema:	<i>“O sistema não verifica se uma sequência no formato fasta foi submetida (H). Além da opinião deste usuário, com relação a validação do formato fasta os usuários B, E, D e I nas respostas das questões abertas relataram ter problema com a validação da sequência, problema considerado de alta prioridade e correção imediata”</i> .	
Solução:	O Script de verificação do formato FASTA foi corrigido.	
Problema:	<i>“Não encontrei onde apagar minhas submissões”</i> (E).	
Solução:	Uma coluna de apagar foi acrescentada na submissão e em resultados.	

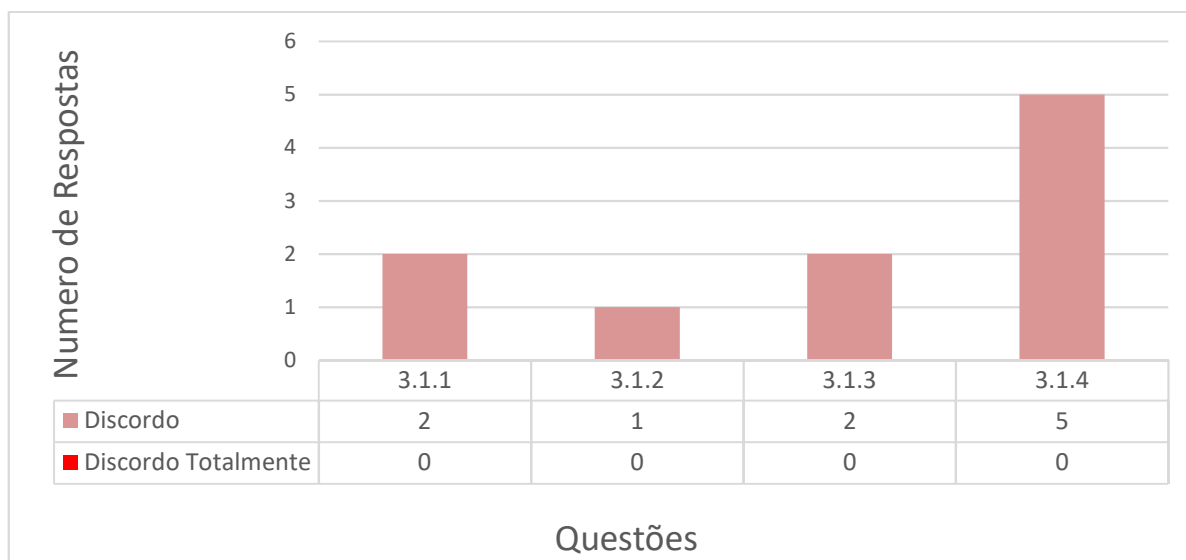
### Heurística: Flexibilidade e eficiência de uso

Número	Questão	Grau
2.7.1	O site oferece diferentes níveis para os usuários, desde o novato até mais o experiente.	2
2.7.2	Atalhos em forma de abreviações, questões especiais, macros e comandos escondidos estão disponíveis para usuários experientes.	1
2.7.5	O sistema é bastante flexível para permitir que os usuários ajustem as configurações adequando-as, isto é, personalizando o sistema.	1
Análise:	O wCReF nesta versão apenas possui os campos personalizáveis da submissão, conforme foi definido na Avaliação Heurística realizada pelos especialistas, que o sistema deve ter uma “customização” do processo na submissão para usuários	

experientes e inexperientes. Para os usuários novatos os padrões de envio da proteína alvo são padrões (Default) e para os usuários mais experientes podem ser personalizados de acordo com parâmetros opcionais de envio. Como o propósito da interface é somente uma, gerar a estrutura 3D aproximada de uma proteína, no formato PDB não necessita uma maior personalização do sistema. Geralmente as análises por usuários mais experientes são realizadas com a estrutura, que é o resultado final da predição, e não diretamente no sistema. A interface Web do CReF segue os padrões dos demais servidores de predição. O que sugerimos quando a personalização é que, para os usuários menos experientes uma explicação do passo a passo e dos parâmetros opcionais sejam demonstradas juntamente no formulário de submissão da proteína, e não somente quando o mesmo clica nos botões de ajuda.

2.7.4	Há opção de usar somente o teclado para realizar tarefas.	1
Solução:	A navegação por teclado ocorria somente através da tecla TAB. Foi acrescentado a navegação por teclado através do atributo <i>accesskey</i> do html5. O atributo <i>accesskey</i> é utilizado para criar uma tecla de atalho para dar foco em um elemento HTML, facilitando assim a navegação do usuário pelo teclado. Os comandos para utilizá-la são <i>shift + alt + tecla</i> e os atalhos estão especificados ao longo da interface.	
<b>Reconhecimento, diagnóstico e recuperação de erros.</b>		
Número	Questão	Grau
2.5.4	Cada mensagem proporciona um procedimento para correção do erro.	2
Solução:	As mensagens de erros foram revisadas e acrescentadas uma maior explicação do erro ocorrido. Quando cabível também a explicação de como o usuário pode evitá-la.	
2.9.5	O site proporciona uma rápida mudança de ação pela qual é possível, por exemplo, disponibilizar ambos, desfazer (Undo) e refazer (Redo).	2
Análise	Há opções de voltar, apagar, limpar. Não há um formulário que necessite os botões de desfazer e refazer.	

O próximo gráfico apresenta as questões negativas referentes a sessão de *Web design* do questionário de satisfação de usabilidade, que são referentes a simplicidade de navegação do site, organização e estrutura (Figura 19).

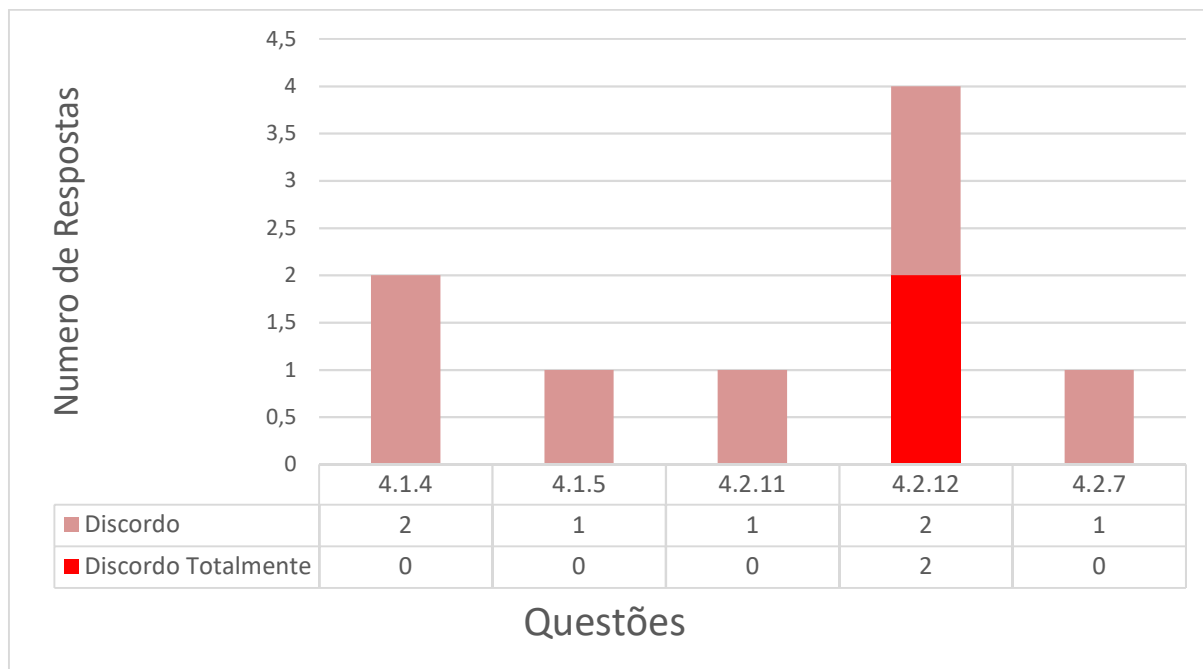


**Figura 19.** Questões avaliadas como negativas em Web *design*. As questões eram referentes a problemas de usabilidade de links da interface e do mapa de navegação.

A questão 3.1.1 era referente a apresentação de um mapa de curso, que tem a mesma finalidade de um menu, apresentando aonde o usuário encontra-se na interface. Para corrigir o problema o mapa foi adicionado nas páginas, ficando na parte superior da interface após o cabeçalho. A questão 3.1.2, marcada com uma resposta negativa (Discordo), perguntava se os links sempre apontavam para os documentos ou páginas corretas. Dessa forma uma revisão nos links da interface foi realizada para garantir que não ocorressem erros neste sentido. Já a questão 3.1.3 pergunta se existem *links* (Atalhos) para mesma sessão dentro de uma mesma página. Dois usuários responderam que discordam. Este fato deve-se que os links de atalho estão disponíveis somente nas páginas que apresentam um conteúdo extenso, para ajudar na navegação da interface, como por exemplo em ajuda. E a questão mais citada, a 3.1.4 com 5 respostas, as cores para os links visitados são consistentes com os padrões da Web, ou seja, as ligações não visitadas estão em azul e aquelas visitadas estão em verde ou roxo, foi resolvida acrescentando a pseudoclassee do CSS3 *:visited* nos links atribuindo a cor roxa para os links visitados. Os links não visitados já eram apresentados em azul em toda a interface.

As questões que abrangem a área de domínio, a bioinformática (Figura 20), foram classificadas em duas questões com mais de uma resposta negativa, e três com uma resposta. Três usuários marcaram que discordam na questão 4.1.4 e 4.1.5 - Está claro quais os materiais que contêm direitos autorais e quais não possuem (duas respostas discordo) e o conteúdo do servidor cita as devidas referências do material que é utilizado (uma resposta). Como a interface do wCReF e o método CReF são resultados de pesquisas científicas, dentro de um ambiente acadêmico, existe uma preocupação com a atribuição dos devidos direitos autorais de qualquer

material, fonte, software que é utilizado em sua concepção. Embora todas as referências tenham sido citadas, alguns links com a referência completa não estavam funcionando. Uma revisão das referências disponibilizadas então foi realizada de forma a resolver o problema.



**Figura 20.** As questões avaliadas como negativas em bioinformática. Ao todo foram 9 avaliações, 7 marcadas com discordo e duas com discordo totalmente. Todos esses problemas foram solucionados.

O problema da questão 4.2.11, consigo salvar meus resultados foi resolvida dando um destaque ao botão de salvar, com um ícone com uma metáfora de disquete, apresentado inicialmente na página de visualização. O usuário A não encontrou a opção de salvar quando analisou essa tela.

A próxima questão é referente aos resultados da submissão serem comunicados ao usuário sem o mesmo estar conectado ao servidor (4.2.12). Dessa forma ficou definido que:

- 1) Quando o usuário submete uma proteína ao servidor o mesmo recebe em seu e-mail um resumo de sua solicitação, com o nome da proteína submetida, a sequência informada, os parâmetros opcionais, a data que a mesma foi enviada ao sistema juntamente com o local que a mesma se encontra na fila de espera.
- 2) Quando a predição da proteína alvo é concluída outro e-mail é enviado informado o fim da predição juntamente com o link para a visualização do resultado e a proteína no formato PDB como anexo.

Com relação ao último tópico das questões houve a ocorrência de uma resposta negativa, a da questão 5.2.2 referente o quão rápido é de se trabalhar no sistema. O usuário relatou que

durante o teste enviou uma proteína de tamanho grande, conseqüentemente seu tempo de resposta foi mais demorado que dos demais, que enviaram uma proteína pequena.

Outro ponto importante a salientar é a avaliação do usuário D quanto a interface do wCReF. Como mencionamos o usuário D é um dos especialistas que participou da Avaliação Heurística. Na avaliação do wCReF realizada com o questionário adaptado de Ssemugabi, não houve respostas negativas (discordo e discordo plenamente) no bloco que abrange as 10 heurísticas de Nielsen. Isto sugere que este especialista considerou que a interface desenvolvida não apresentava problemas que violassem as heurísticas nas questões fechadas. Somente uma questão, foi marcada com a opção discordo, mas com relação a *Web design*, a questão 4.2.12 (Os resultados da submissão são comunicados a mim sem eu estar conectado ao servidor, por outros meios, como por exemplo por e-mail), a qual já foi proposta uma solução para a versão final. Não estamos comparando aqui, as duas avaliações. O objetivo é que uma avaliação de usabilidade complemente a outra, para concepção da interface do wCReF. A opinião deste especialista nas questões abertas está descrita na próxima subseção.

#### 4.5.3 Respostas das questões abertas

No final de cada bloco de questões foi deixado um espaço para que os usuários descrevessem sua opinião quanto aos pontos negativos. Observando as respostas dos usuários estes utilizaram esse espaço para realizar comentários positivos e negativos, e também sugerir alterações na interface do wCReF. Algumas das respostas dos usuários das questões abertas foram disponibilizadas como problemas relatados na Tabela 15 (anterior).

##### 1) Usuário A

O usuário A afirmou que “*no menu inicial o visualizar carrega outra proteína (a proteína exemplo) e não a sua submetida*”. O erro ocorrido foi considerado grave, pois somente a proteína correta apareceu após o usuário atualizar a página. A forma de se carregar a proteína na página foi alterada (código-fonte) e o erro não foi mais observado. O usuário sugeriu que na visualização dos resultados no formato PDF fosse disponibilizado uma cópia da imagem da estrutura na posição desejada pelo usuário.

##### 2) Usuário B

O usuário B fez vários comentários positivos e negativos sobre a concepção da interface:

*“Lendo a documentação do método, entendi o funcionamento parcialmente. A meu ver, falta um pouco de estudo da minha parte. Todavia seria mais interessante descrever o método como um workflow (fluxo de trabalho). Acho mais entendível.”*

Esta sugestão foi aceita, em primeiro momento uma descrição através de imagem do fluxo de trabalho do CReF foi disponibilizada na interface.

*“Muito boa a interface de visualização da estrutura predita, com todos os dados que definimos para a execução do algoritmo.”*

*“Acho que poderia ser melhor avaliado se eu tivesse melhor conhecimento do assunto. Talvez uma explicação de como o parâmetro influência no resultado da estrutura, caso seja de simples entendimento”.*

Os parâmetros opcionais estão descritos na ajuda da página de envio e também na ajuda da documentação, e no tutorial. O conteúdo das explicações não foi suficiente. Definiu-se que uma melhor explicação destes parâmetros que foram colocados na interface final.

*“É uma Interface Web. Não acho necessário somente o uso de teclado, basta apertar tab para ir a outros campos”.*

Como ponto positivo o usuário B disse que o sistema é *“de fácil utilização, muito simples”*.

### **3) Usuário C**

O usuário C mencionou que o conteúdo da interface se altera conforme o mesmo minimiza a tela. Isto se deve a interface do wCReF ter sido implementada de forma a ser adaptativa de acordo com o tamanho da tela, tendo um *Design Responsivo*; Um *Design Responsivo* é aquele em que a interface se adapta os diferentes tipos de telas, dependendo do tipo de dispositivo o qual a mesma usa para acessar, como telas grandes, notebooks, netbooks, tablets e celulares.

A Interface Web do wCReF buscou seguir esta proposta. Foram adaptadas as telas para dispositivos com telas grandes, como por exemplo desktops com mais de 19 polegadas, assim como para notebooks. A adaptação para dispositivos mobile como tablets e celulares serão realizadas. Apresentamos na Figura 21 um exemplo de código de estilo (CSS) diferente para tamanhos diferentes de tela.

*“Nas citações da documentação sobre o CReF alguns links não fazem nada, além disso ficaria bom que a caixa da citação feche com o ESC ou em outro sítio que não seja o X.”*

A sugestão para fechar as janelas com a tecla ESC foi adotada. Porém acreditamos que seja necessário o botão de fechar juntamente com o “X” quando se trata de uma janela de aviso que necessite da confirmação do usuário.

```

/*.....Telas maiores que 1382 px*/
@media only screen and (min-width: 1382px) {
nav#menu_horizontal {
margin:0px auto;
width: 70%;
height:6.000em;
}/*cabecalho*/

img.Logo_cref{
float:left;
display:block;
margin-top:1%;
margin-left:9%;
width:7%;
height:13.000em;
padding:0.5em;
max-width: 100%;
position: absolute;
z-index: 999;
}/*icones topo*/

/*.....Notebook*/
@media (min-width: 1200px) and (max-width: 1382px) {
nav#menu_horizontal {
margin:0px auto;
width: 100%;
height:6.000em;
}/*cabecalho*/

img.Logo_cref{
float:right;
display:block;
margin-top:0.5em;
margin-left:15%;
width:10%;
height:13.000em;
padding:0.5em;
max-width: 100%;
position: absolute;
z-index: 999; /* número máximo é 9999 */
}/*icones topo*/

```

**Figura 21.** Parte do código do wCReF que mostra que as telas da interface são adaptáveis. Foram aplicados diferentes estilos (CSS) dependendo do tamanho da tela em que o usuário visualiza. Os parâmetros da *nav* (menu) e da imagem do logo são aplicados conforme o tamanho da tela que é visualizado.

Os pontos positivos descritos foram “a facilidade do sistema para principiantes”, “os resultados que são demonstrados de uma forma consequente” e “a ajuda que fica pronta em cada passo”. Como ponto negativo o usuário C diz que o link de cadastro teria que mudar para perfil, modificação que foi realizada.

#### 4) Usuário D

Na opinião do especialista D, o wCReF não apresentou problemas negativos de usabilidade nas questões fechadas. Quanto as questões abertas o mesmo fez algumas sugestões, para a melhoria da interface.



*“A mensagem de progresso poderia ser mais explicativa, onde poderia se fazer uma barra de progresso depois que a predição começar”.*

Como a interface do wCReF funciona independentemente da predição enviada, uma barra de progresso neste caso não seria necessária. Geralmente barras de progresso são apresentadas quando um usuário fica esperando alguma resposta do sistema, como por exemplo a transferência de um arquivo ou *download* de uma imagem. Optamos por mostrar qual passo do CReF está sendo executado e uma previsão aproximada do término da predição.

*“Ao esquecer a senha não fica claro que o e-mail para recuperação foi enviado”.*

Foi adicionada uma mensagem de aviso, que informa ao usuário que foi enviado um e-mail para recuperação de sua senha.

*“Fiquei um pouco em dúvida sobre a sessão visualizar. Talvez devesse estar sempre dentro de resultados. ”*

Concordamos que um botão adicional no menu, somente para visualização, como apresentado no protótipo poderia causar uma confusão por parte dos usuários de qual local o mesmo poder ver seus resultados. Então ficou definido que a visualização estaria disponível somente na guia de resultados.

*“Poderia ser colocada uma sessão com a referência para quem usar o wCReF deve incluir no seu artigo”.*

As referências referentes ao método estão todas disponibilizadas na página. Assim que forem publicados artigos sobre a interface Web, estas serão adicionadas.

*“É preciso documentar os parâmetros diretamente na página de submissão”*

Estes parâmetros já estavam disponíveis na página de submissão, pela ajuda embutida. Foi destacado esta informação na página de submissão, assim como a presença da descrição dos parâmetros na documentação.

Também relatou a ocorrência de erro na validação do script do formato FASTA, um link na página do usuário que o levava novamente a home, e a cor dos links visitados na documentação, que deveriam trocar de cor. Todos esses aspectos foram corrigidos para a versão final.

No espaço referente aos pontos negativos, o mesmo diz que deve haver “uma melhor descrição das mensagens de erro da sequência FASTA inválida e um progresso da predição depois que ela começar”.

Como pontos positivos destaca que “o sistema tem uma interface simples e clara”. “O entendimento do método e submissões são fáceis”.

## 5) Usuário E

*“Ao submeter a sequência não recebo nenhuma mensagem referente ao tempo que levará a simulação “esperado”. Não tenho onde visualizar nos resultados ou em outro lugar as sequencias que submeti na página inicial. Há a aba submissões, mas só a vi posteriormente. ”*

Este problema foi solucionado com uma mensagem explicativa após a submissão de que o usuário pode visualizar a submissão clicando em submissões (atalho na mensagem) ou retornar a página de envio. Quanto ao tempo a seção 5.1.3 refere-se à limitação técnica apresentada para determinar o tempo de execução.

*“Em algumas telas como a de visualização dos resultados há termos em inglês misturados com termos em pt-br”.* Foi padronizado todos os botões em português, sendo que também será disponibilizada uma versão da interface em inglês, pois é importante para o uso do servidor em âmbito internacional em pesquisas científicas.

O usuário E também sugeriu que o nome do cadastro quando o usuário estivesse logado fosse trocado para Perfil ou Conta do usuário. Também observou a padronização dos títulos em visualizar, onde algumas coisas estavam em negrito, e outras não.

Foi relatado que o botão de apagar submissão não tinha sido encontrado. Isso porque na implementação este campo só aparecia após uma submissão for concluída. Foi então disponibilizado a opção de cancelar um trabalho desde o início.

*“Não existe caixa de diálogo apontando senha/login incorretos. ”* Esta mensagem existia, mas era apresentada junto com a janela de login, o que pode ter levado ao usuário não perceber. A solução encontrada foi colocar a mensagem em uma nova janela informando o erro.

Os pontos positivos destacados por este usuário é que *“a interface é fácil de usar, bonita e simples”*. E como ponto negativo *“o sistema em pt-br (português do Brasil) pode dificultar o uso em âmbito internacional”*. Isto já foi comentado nos parágrafos acima, há a clara necessidade da interface ser apresentada também primeiramente em inglês, e posteriormente em outras linguagens.

## 6) Usuário F

Na avaliação do usuário F não ocorreu nem um tipo de erro durante a execução do teste, o que fez o mesmo colocar este fato na resposta de seu questionário. Também relatou que a seção “Ajuda” o mesmo encontrou um link que não apontava corretamente para o local do mesmo.

Como ponto positivo em sua avaliação ele citou o menu de preferências de visualização e que a interface era “light”.

### **7) Usuário G**

O usuário G achou que não deveria responder as questões abertas porque trabalhava com bioinformática há menos de 6 meses, portanto não se considerava apto.

### **8) Usuário H**

A sugestão do usuário H quanto ao uso da interface é de mostrar a porcentagem concluída da predição, enquanto a mesma estiver rodando. Isto será uma sugestão para trabalho futuro, pois o método CReF está passando por modificações, visando melhorar a qualidade de suas predições, juntamente com ganho de velocidade. O que podemos mostrar atualmente em qual parte a predição se encontra, isto é, qual o passo do algoritmo do método está sendo executado.

Como qualificação positiva o usuário H mencionou a “*facilidade de uso da interface*”, e um “*ambiente limpo, clean, sem poluições.*” Os pontos negativos foi a demora para receber os resultados e a figura não abriu após o resultado concluído. Na avaliação do usuário H, houve um período maior para receber os resultados porque o mesmo submeteu sequências de proteínas com tamanhos grandes, devido ao processamento do BLAST e do preditor de estrutura secundária.

### **9) Usuário I**

Quanto aos erros ocorridos “*A opção de utilizar somente o teclado só funciona conjuntamente com a utilização do mouse*”, “*As páginas visitadas não ficam de cor diferentes das não visitadas*” e “*Não houve notificação por e-mail dos resultados obtidos*” já foram verificados nas respostas questões fechadas, corrigidos na versão final.

A descrição positiva deste usuário é que “*o wCReF é um sistema Web utilizado para a predição de estrutura 3D de proteínas. O sistema segue os padrões básicos necessários para usabilidade de novos usuários e usuários avançados respondendo de forma satisfatória ao fim para o qual foi desenvolvido*”.

O principal problema encontrado na interface segundo o usuário I é quanto ao retorno de mensagem de erro, especificando qual foi o erro encontrado, e que em suas configurações poderia ter a opção de notificar os resultados via e-mail.

### **10) Usuário J**

A opinião do usuário J é que, na mensagem que aparece quando é submetida uma sequência de aminoácidos para predição, seja informado o local onde o mesmo pode acompanhar os resultados, sugestão também acrescentada. Como não ocorreram erros durante

seu teste, o mesmo marcou a resposta. Talvez nas questões abertas que tinha relação a erros, e comentou isso nas questões fechadas.

*“Depois de clicar em visualizar ele continua azul”*. A cor padrão para links visitados foi modificada para cor roxa.

O ponto positivo destacado foi que o servidor é *“fácil de utilizar”*. Os pontos negativos: *“a proteína que foi demonstrada em visualizar foi a exemplo”* e sugeriu que deveria *“Destacar o botão salvar PDF”*. Os dois pontos negativos foram com relação ao menu visualizar. Este problema foi solucionado modificando o script de chamada da proteína a ser visualizada e seu conteúdo, para que os ícones de salvar e de gerar PDF ficassem mais visíveis para o usuário.

### **11) Usuário K**

O usuário K não conseguiu visualizar a interface no navegador Firefox, somente no Chrome. Com os outros usuários, em outros computadores funcionou normalmente em diferentes navegadores. Então foi colocado na interface, e nesta dissertação quais os navegadores compatíveis e as versões que são compatíveis com as tecnologias utilizadas na implementação. O mesmo erro da proteína carregada erroneamente ocorreu no teste do usuário K.

### **12) Usuário L**

O usuário L recebeu uma janela de erro que não estava no padrão das demais. A referida caixa de diálogo foi referente ao formato da Sequência FASTA, e seguia o formato padrão de mensagens Java Script, e não como as demais apresentadas na interface. Um padrão de mensagens de erro, em toda interface foi então adotado, para que não haja diferença na apresentação desta informação. Como ponto positivo o mesmo diz que *“a interface é simples e leve (sem sobrecarga de cores, imagens) e que possui “menus objetivos”*.

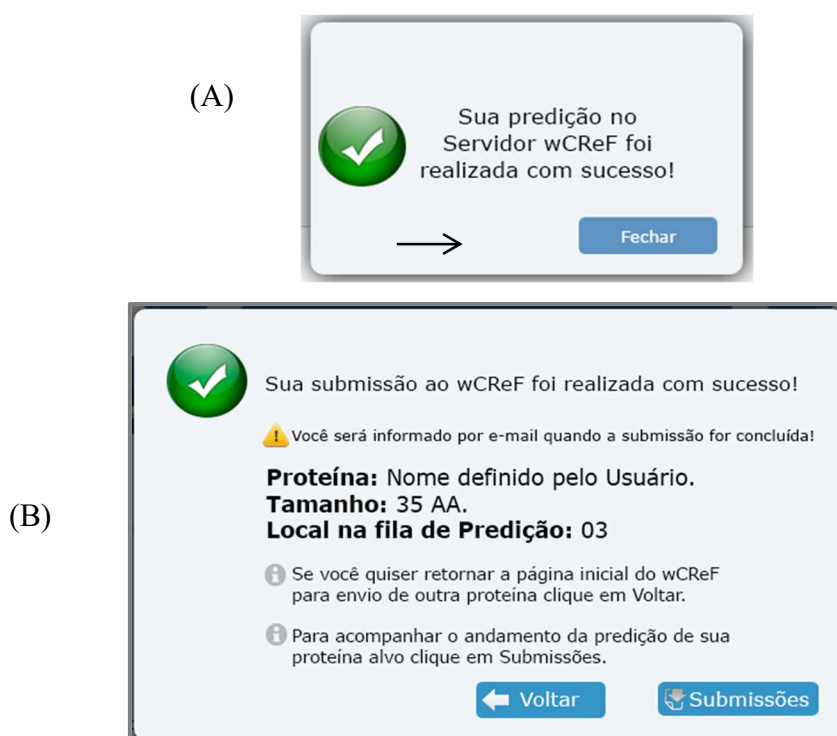
#### **4.5.4 Alguns Pontos que contribuíram para melhoria da interface**

Foi importante a realização das avaliações no início do projeto da interface, mais especificamente na definição dos requisitos de software, pois dessa forma conseguimos captar as características que o servidor deveria possuir. Assim na fase de implementação já estava definido como e o quê o servidor deveria fazer, o que garantiu que ocorressem poucos erros de usabilidade na opinião dos usuários.

Os aspectos apontados pelos usuários foram relevantes para melhoria da usabilidade da interface Web do wCReF, principalmente pelos erros que ocorreram, dois deles considerados essenciais para sua utilização, a validação da sequência no formato FASTA e a visualização da proteína, através da mudança dos scripts de controle dessas duas funções.

A validação do formato FASTA foi apontada como um erro de usabilidade nas Avaliações Heurísticas dos especialistas dos servidores de predição (prevenção de erros). Dessa forma foi criado um *script* para sua validação no protótipo, o que não foi o suficiente para garantir a correção de todos os erros no formato, o que ocasionou as respostas negativas. A avaliação pelos usuários finais foi relevante para testar as possibilidades de erros, e neste caso, permitiu a correção deste *script*.

Outro aspecto importante a considerar foi como abordar a questão de como apresentar a informação após a submissão de uma proteína. Os usuários sugeriram que após a submissão, fossem encaminhados para a página onde poderiam acompanhar as submissões. Desta forma, apresentamos o antes e depois da janela de aviso na Figura 22.



**Figura 22.** A mensagem de confirmação de uma submissão, sugerida nas avaliações de questionário pelos usuários. (A) - Como era apresentado a mensagem de submissão aos usuários. (B) - Mensagem após as sugestões dos usuários.

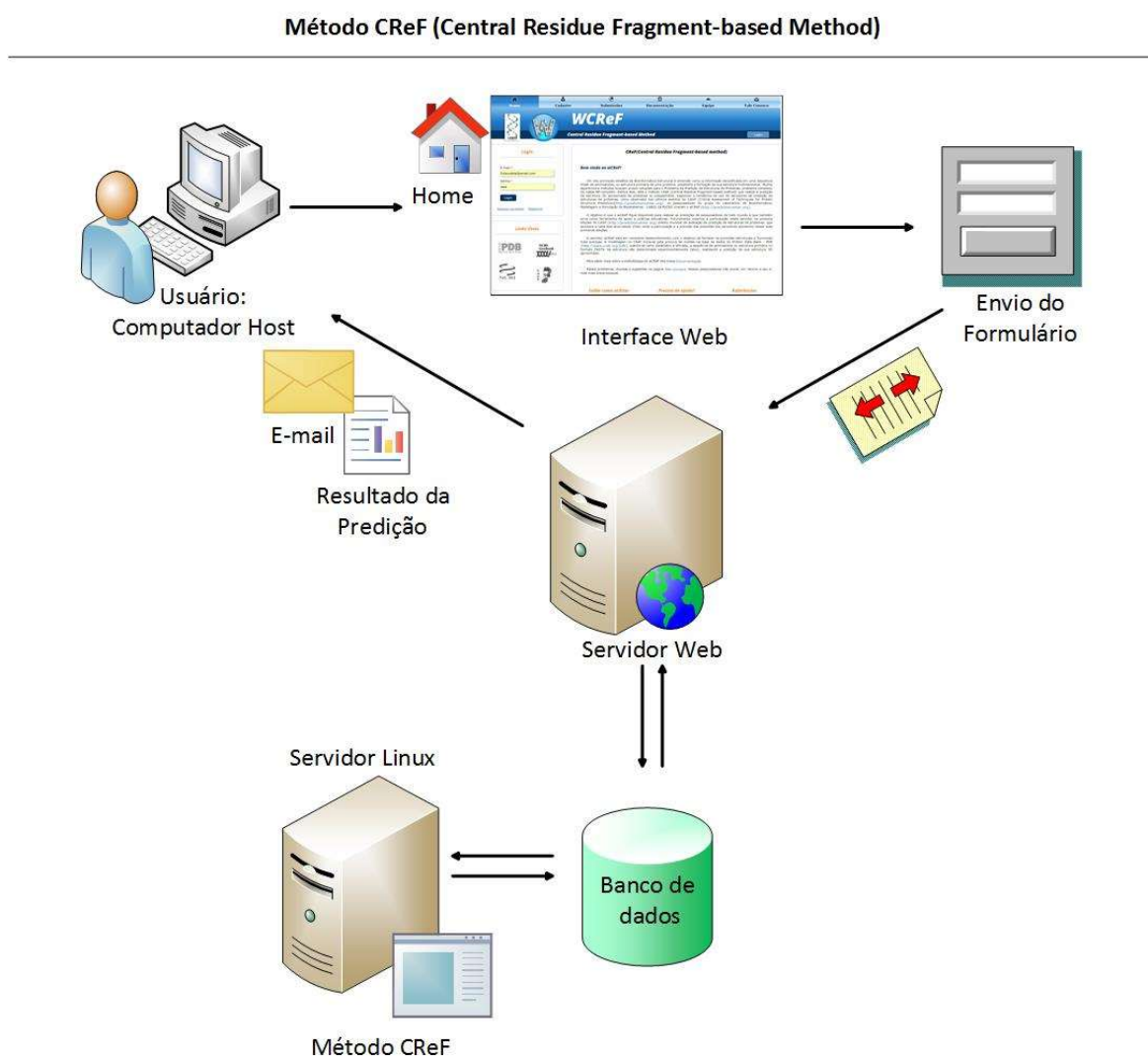
Além destes pontos que contribuiriam para a melhoria da qualidade da interface desenvolvida, também foram realizadas as correções dos erros de usabilidade descritas ao longo desta seção (5.4).

## 5 WCREF

O wCReF é um servidor de estruturas de proteínas que utiliza o método CReF para realizar suas predições computacionais. Este servidor tem o objetivo de facilitar o trabalho de pesquisadores da área de Bioinformática Estrutural para predição computacional de proteínas, fornecendo uma ferramenta Web com uma interface amigável.

O servidor wCReF constitui uma ferramenta útil para a atuação do profissional de bioinformática e biologia molecular, fornecendo um ambiente para a predição de estruturas de proteínas.

Por meio da interface Web do wCReF pesquisadores e cientistas poderão enviar sua proteína alvo para predição aproximada. A sequência é enviada utilizando um formulário web, armazenada em um banco de dados, para que o servidor com o método CReF implementado possa processar e devolver os resultados (Figura 23).



**Figura 23.** O wCReF. O usuário se comunica com a aplicação através da interface, submetendo sua requisição por meio de um formulário. O conteúdo é armazenado em um banco de dados, sendo

processado pelo servidor que retorna para o usuário as informações sobre a predição aproximada da estrutura de proteínas.

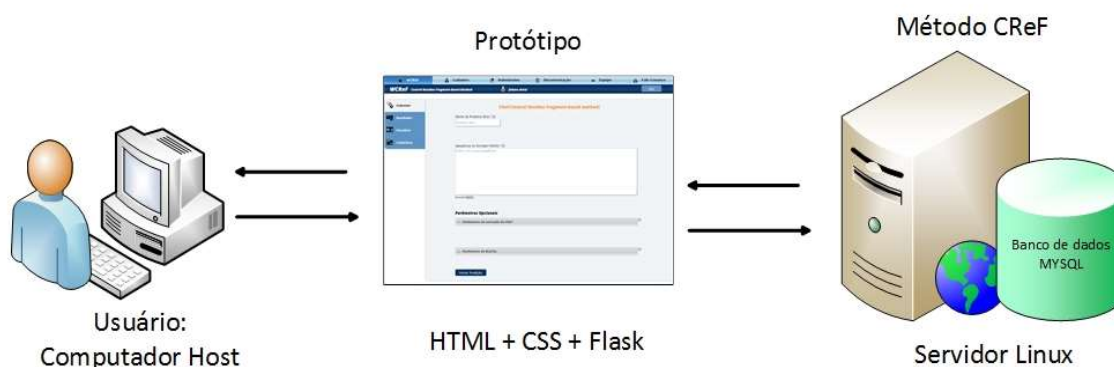
Como saída o usuário obtém a predição da estrutura 3D aproximada no formato padrão do *Protein Data Bank* (extensão .pdb), que contém informações sobre a estrutura 3D aproximada da sequência alvo, com as coordenadas de cada átomo dos aminoácidos da predição, podendo ser observada em programas de visualização e modelagem molecular, como o *Swiss PDB Viewer* (Guex, 1996).

Os resultados são devolvidos para o usuário através do próprio sistema do wCReF e também é enviado um e-mail avisando que o resultado se encontra disponível no sistema.

### 5.1 A interface do wCReF

A interface do wCReF foi definida a partir do melhoramento do protótipo realizado através dos resultados obtidos a partir da execução do teste com o protótipo pelos usuários (Figura 24). Por meio desses testes foi possível avaliar e validar os objetivos propostos neste trabalho. Como resultado descrevemos aqui a interface final do servidor de predição.

#### Método CReF (Central Residue Fragment-based Method)

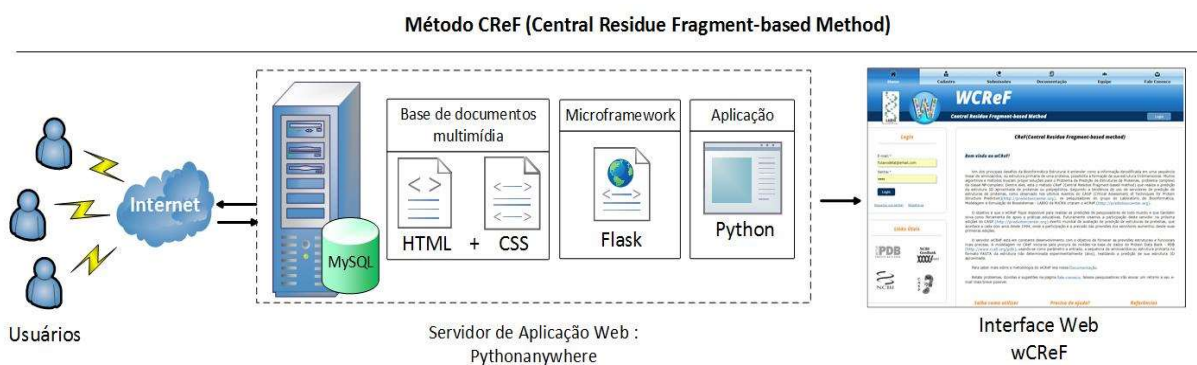


**Figura 24.** O protótipo deu origem a interface Web, pela melhoria implementada após as avaliações. Os usuários realizaram as avaliações através de questionários. A interface que foi desenvolvida utilizando linguagens de programação Web (HTML, CSS, *Flask*, Python) a interface se comunica com o método CReF realizando a interação do usuário com o sistema.

Os recursos utilizados no desenvolvimento do wCReF são os mesmos que foram utilizados na implementação do Protótipo (apresentado no capítulo anterior, seção 5.3). Na verdade, a versão apresentada neste capítulo, nada mais é que o protótipo acrescentado das correções dos erros e as sugestões dos usuários, levantadas através do questionário de satisfação

usabilidade, juntamente com os requisitos do sistema. A interface final apresenta, além dos recursos descritos no protótipo, um servidor para hospedagem na Web. A hospedagem foi realizada através do servidor Web *Pythonanywhere* (<https://www.pythonanywhere.com>) que é baseado na linguagem *Python*.

Podemos conferir na Figura 25 a arquitetura da interface do wCReF:



**Figura 25.** Arquitetura da interface Web do wCReF. Os usuários acessam o wCReF através da internet. O servidor da aplicação Web gerencia a aplicação, que foi desenvolvida utilizando linguagens de programação Web (HTML, CSS, *Flask*).

#### 1) Quais as alterações do CReF alteram o wCReF?

Como a interface do wCReF foi feita independentemente do método CReF, as alterações em sua metodologia não alteram a interface web. O banco de dados, o código HTML e CSS e seus templates flask, não sofrem alterações se o método passar por mudanças. O código desenvolvido em python faz a chamada do método, envia a proteína (onde o método CReF realiza a predição) e devolve a mesma, na interface depois realizada a predição. Sendo assim a interface web não sofre alterações se o método é modificado, o que facilita as atualizações e a manutenção. O que pode gerar alterações na interface apresentada é se, os parâmetros de entrada e saída do método CReF forem alterados, ou se novas funcionalidades forem adicionadas ao servidor de predição, o que pode acontecer em trabalhos futuros.

#### 5.1.1 Concepção da interface

A interface do wCReF é resultado das avaliações realizadas pelos Especialistas na avaliação Heurística e dos usuários na avaliação por questionários. Podemos perceber que as heurísticas de Nielsen serviram para detectar os problemas de usabilidade nos servidores de predição (I-TASSER, QUARK e Robetta), e assim evitar que na concepção da interface web do CReF estes mesmos erros ocorressem. Dessa forma, compilamos na Tabela 16.



Características que foram contempladas no servidor wCReF após a análise dos problemas encontrados nos servidores. As características que a interface do wCReF possui, e que estão relacionadas aos problemas relatados pelos avaliadores especialistas, que foram contemplados na concepção do wCReF.

**Tabela 16.** Características que foram contempladas no servidor wCReF após a análise dos problemas encontrados nos servidores.

	<b>Característica</b>	<b>wCReF</b>
	<b>Visibilidade do sistema:</b>	
1	Retorno ou visibilidade que a proteína está sendo gerada.	X
2	Demonstração clara de que o servidor está sendo utilizado.	X
3	Apresenta um caminho demarcado para que o usuário visualize onde ele se encontra em um determinado momento.	X
4	Apresenta informação visível de que o usuário precisa realizar um cadastro antes da submissão da predição da proteína, e que o mesmo precisa realizar o login no servidor para enviar seu trabalho.	X
5	Há exibição da fila de Jobs e de trabalhos enviados.	X
6	Há uma estimativa de tempo para término da predição	X
	<b>Design Minimalista</b>	
7	Menu reduzido e prioridade a área de envio da sequência para predição da estrutura de proteína.	X
8	Opções de menu mostradas somente quando solicitadas.	X
9	As mensagens de aviso ou erro são mostradas em janelas separadas com confirmação, ficando bem visíveis ao usuário, evitando a sobrecarga de informações na mesma página.	X
10	São apresentadas somente as informações necessárias em cada página, evitando a sobrecarga de informações.	X
	<b>Erros ocorridos e mensagens de erro:</b>	
11	Apresenta mensagens de erro ao ser colada uma sequência de proteína incorreta.	X
12	Apresenta mensagens de erro na mesma página em que o erro ocorre.	X
13	Em caso de erro há um retorno a página principal.	X

14	Erros são apresentados com cores ou símbolos diferentes a fim de chamar a atenção.	X
15	Os erros na entrada de dados são exibidos de forma clara na interface, através de caixas de mensagens e aviso e pedem confirmação do usuário.	X
	<b>Prevenção de erros</b>	
16	Há indicação de quais dados são obrigatórios e quais são opcionais.	X
17	A função de limpar um campo só é realizada através da confirmação do usuário.	X
	Envio e resultados da predição	
18	Apresenta um exemplo de sequência (na página principal) para direcionar o usuário.	X
19	Apresenta um direcionamento do que pode estar incorreto na sequência fornecida pelo usuário.	X
20	Há um Método de busca para sequência ou proteínas no servidor.	X
21	Há uma página com os resultados que pode ser lida de tempos em tempos para determinar se a predição foi concluída.	X
22	Possibilidade de apagar um job ou predição de forma direta (com uso de um botão, sem ser necessário várias ações) e com confirmação do usuário.	X
23	Envio de mais de um job por vez ao servidor.	X
24	Apresenta validação do formato da sequência de aminoácidos da proteína a ser predita.	X
25	Informa ao usuário como acompanhar o andamento de sua predição.	X
26	Há uma mensagem clara que a predição foi enviada ao servidor.	X
27	Há a possibilidade de customizar a predição, fazendo distinção entre usuários experientes e inexperientes.	X
28	Os parâmetros opcionais possuem uma documentação.	
	<b>Perfil e cadastro do usuário:</b>	
29	Há uma conta e, portanto, as sequências submetidas podem ser facilmente obtidas posteriormente.	X
30	E-mail para o envio da predição não necessita ser acadêmico.	X
31	Pode-se alterar o perfil do usuário cadastrado e também excluir a conta do usuário.	X
32	Há opção de recuperação de senha.	X

33	Há opção de recuperação de dados de login.	X
	<b>Ajuda ao usuário:</b>	
34	Há documentação sobre os parâmetros opcionais e ajuda rápida sobre seu significado.	X
35	Ajuda rápida a Termos ou funcionalidades, abrindo na mesma página do navegador.	X
36	Há uma ajuda contextual para cada campo de entrada.	X
	<b>Controle e liberdade para o usuário</b>	
37	Há opção de sair do servidor.	X

A opção de escolha de idioma também foi um dos requisitos detectado por meio da avaliação heurística. Este recurso será adicionado a interface do wCReF pois há a necessidade de tornar o uso do servidor o mais flexível possível, possibilitando o acesso a pesquisadores e usuários de outros países.

Além destas, a avaliação dos usuários quanto a concepção da interface também foi importante. Com ela, podemos verificar a ocorrência de erros no protótipo, que foram corrigidos, para acrescentar sugestões de funcionalidades ao servidor, além de melhorar o *design* de acordo com a opinião dos usuários (Seção 4.5).

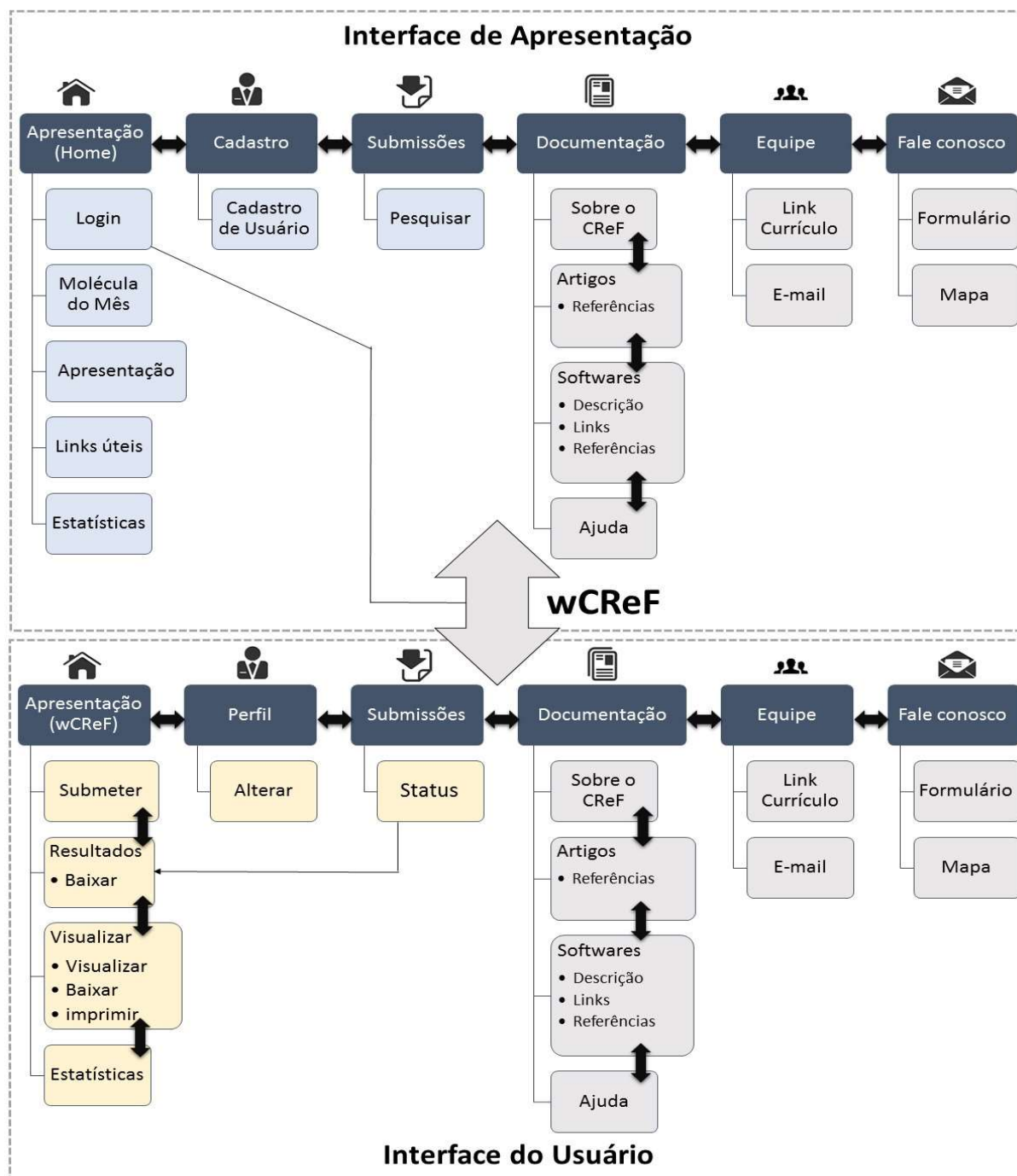
### 5.1.2 Telas da Interface

Antes de apresentar as telas da interface do wCReF apresentamos uma descrição hierárquica com as informações desta ferramenta, conforme Figura 26. Este modelo descreve os blocos de informações contidas no servidor, apresentando uma lista completa das informações que fazem parte desta aplicação e de forma a organizar as informações apresentada nesta seção.

Ao acessar a interface do wCReF o usuário conta com dois templates distintos, o qual chamaremos neste trabalho de:

- 1) **Interface de Apresentação:** Neste caso, a interface é de uso comum. Qualquer usuário que esteja na Web pode acessar essas informações. Neste caso a interface está dividida nas seguintes partes:
  - Tela de Apresentação: Home
  - Tela de Cadastro: Para o cadastro de novos usuários no sistema.

- Tela de Consulta de Submissões: Para acompanhamento dos resultados fora do sistema. Aqui apresenta todas as submissões realizadas no wCReF (apenas consulta do status da submissão).
- Tela Documentação: Traz a documentação do wCReF.
- Tela da Equipe: Apresenta a equipe com os pesquisadores do LABIO.
- Tela do Fale-Comosco: Para contato com a equipe do wCReF.



**Figura 26.** Descrição Hierárquica do Sistema. O usuário tem dois tipos de interface principais. A interface de apresentação de uso comum, e aquela que o usuário acessa ao realizar login no servidor.

**2) Interface do usuário:** O visual (aparência) da interface é a mesma para cada usuário, porém apresenta diferenças no que tange o preenchimento do cadastro, as submissões e os resultados, que para cada usuário é única. É acessada através do Login no sistema do wCReF:

- Tela principal do wCReF: Para a submissão da sequência da proteína alvo a ser predita. Aqui o usuário pode: Submeter uma predição, acessar e visualizar os resultados, e acessar as estatísticas.
- Tela de Perfil: Apresenta os dados cadastrados do usuário.
- Tela de Consulta de Submissões: Aqui o usuário pode acompanhar suas submissões.
- Tela Documentação: Traz a documentação do wCReF.
- Tela da Equipe: Apresenta a equipe com os pesquisadores do LABIO.
- Tela do Fale-Comosco: Para contato com a equipe do wCReF.

O critério para seleção do template foi o login do usuário. Para isto foi importante a definição do microframework *Flask* na aplicação, pois permitiu o uso de templates na criação do servidor. O uso do template é importante para garantir uma uniformidade em todas as telas da interface. O que varia é o conteúdo de cada tela, permanecendo no mesmo padrão o cabeçalho do sistema (na parte superior) e o rodapé da página (no final da página). Assim se mantém a consistência do menu horizontal que dá acesso as demais telas da interface.

Cada tela aqui apresentada foi desenvolvida segundo os aspectos relacionados no documento de requisitos de software vislumbrando a usabilidade e também pelas sugestões e correções de erros definidas pelos usuários finais. Em cada parte da interface aqui descrita, relaciona-se com os problemas de usabilidade encontrados nos servidores de predição, através das Heurísticas de Nielsen.

### 5.1.3 Interface de Apresentação

É a parte do sistema de uso comum a todos os usuários. É preciso estar conectado ao servidor do wCReF através da Web para poder acessar essas telas. São as informações que estão disponíveis para acesso público, sem necessidade de um cadastro prévio ou realização de login no servidor. O template da interface de Apresentação é demonstrado na Figura 27 e contém as seguintes partes:



**Figura 27.** Template da interface de Apresentação. O cabeçalho e o rodapé permanecem iguais durante a navegação entre as páginas.

- **Menu Horizontal:** A parte superior apresenta um menu horizontal. É através dele que o usuário consegue navegar pelas diferentes partes da interface.
- **Cabeçalho:** O cabeçalho é a parte em destaque no topo da página, que fica logo abaixo do menu principal. Nele há as informações de identificação do servidor, as logo marcas e títulos, assim como também tem um botão de login. Este atalho então aparece em todas as telas da interface, para o usuário realizar o login de qualquer local.
- **Rodapé:** Contém um menu vertical de navegação, a documentação, os links da instituição e as logo marcas do grupo de pesquisa.

As funções do template então são de navegação pelas partes da interface (Menu), Informação (Título, logo marcas, links) e o botão de acesso ao login no site. Isso foi importante para garantir a padronização e evitar os problemas de usabilidade que ocorreram nas Heurísticas:

- **Compatibilidade entre o sistema e o mundo real:** Ícones simbolizando ações familiares ao usuário, como por exemplo limpar (borracha), home (casa) e salvar (disquete).
- **Consistência e Padrões:** Os menus da interface são únicos, padronizados. Houve a ocorrência nos servidores de menus diferentes em cada tela, com vários formatos, e posições diferentes conforme se percorria a interface. Em outro caso os botões apresentados eram somente os do navegador. Os botões tanto do template quanto da

parte do conteúdo da interface do wCReF buscou manter um padrão, mas serem diferenciados através da adição de ícones para não serem muito similares. Por exemplo botão de enviar possui uma seta para a direita, enquanto que o botão de limpar o conteúdo possui uma borracha (ações familiares ao usuário como descrito no item anterior).

Além disso, foi encontrado nos servidores avaliados, muito texto para repassar informações aos usuários, geralmente diferenciados somente por títulos. Não há uma padronização nas telas, ficando o conteúdo confuso e com muita informação reunida. Buscamos priorizar o uso de atalhos, botões, mensagens, distribuição das informações de acordo com interesse do usuário, reunidas em blocos ou novas telas, separadas por menus de acordo com o interesse ou conteúdo relacionado, para contemplar a heurística de *design* estético e minimalista.

O menu fica separado do restante do conteúdo também seguindo as orientações definidas nesta heurística. Outro ponto apresentado nos servidores foi o uso do menu de acesso as páginas junto a ações a serem executadas pelo usuário. Isto polui bastante a interface. Por isso as opções são mostradas seguindo um padrão, e algumas delas apresentadas somente quando o ponteiro do mouse é colocado sobre itens. Como por exemplo em Documentação, as opções “*Sobre o CReF*”, “*artigos*”, “*softwares*” e “*ajuda*” aparecem com o passar do mouse, efeito chamado *hover* em CSS.

A primeira tela da Interface de Apresentação é a Tela de apresentação, ou Home do sistema (Figura 28). É ela que o usuário vê primeiramente quando acessa o servidor do wCReF. Por isso esta área da interface tem basicamente conteúdo informativo e de ajuda ao usuário para se contextualizar no ambiente que o mesmo virá a utilizar.

Todas as páginas apresentadas além do Menu superior (Cabeçalho) e inferior (Rodapé) possuem um mapa de navegação, para situar o usuário onde o mesmo encontra-se na interface, fornecendo uma informação de visibilidade. É um caminho demarcado para saber onde o usuário encontra-se em determinado momento. Isto foi uma das sugestões definidas pelos usuários finais.

Sugerimos que para enviar a sequência primeiramente o usuário tem que realizar login no sistema, através de um cadastro prévio. Problemas de usabilidade encontrados foram relacionados à questão de não ficar claro a necessidade de login antes do usuário submeter a predição. A Heurística relacionada a isto nas avaliações foi a prevenção de erro. Dessa forma, o usuário deve realizar login para poder enviar a predição, ou isso deve ser avisado previamente para o usuário, antes de se utilizar a ferramenta.

Home Cadastro Submissões Documentação Equipe Fale Conosco

**wCReF**  
Central Residue Fragment-based Method

Você encontra-se em: [Home](#)

**Login**

E-mail: \*  
email\_usuario@mail.com

Senha: \*  
\*\*\*\*

Login

[Esqueceu sua senha?](#) [Registre-se](#)

**Links Úteis**

RCSB PDB PROTEIN DATA BANK

NCBI GenBank XXXX01011

NCBI

CASP

**Molécula do Mês de Maio**

Receptor for Advanced Glycation End Products

June/2015

© David S. Goodsell and RCSB PDB

**CReF(Central Residue Fragment-based method)**

**Bem vindo ao wCReF!**

Um dos principais desafios da Bioinformática Estrutural é entender como a informação decodificada em uma sequência linear de aminoácidos, ou estrutura primária de uma proteína, possibilita a formação de sua estrutura tridimensional. Muitos algoritmos e métodos buscam propor soluções para o Problema da Predição de Estruturas de Proteínas, problema complexo da classe NP-completo. Dentre eles, está o método CReF (Central Residue Fragment-based method) que realiza a predição da estrutura 3D aproximada de proteínas ou polipeptídeos. Seguindo a tendência de uso de servidores de predição de estruturas de proteínas, como observado nos últimos eventos do CASP (Critical Assessment of Techniques for Protein Structure Prediction)(<http://predictioncenter.org>), os pesquisadores do grupo do Laboratório de Bioinformática, Modelagem e Simulação de Biosistemas - LABIO da PUCRS criaram o wCReF.(<http://predictioncenter.org>).

O objetivo é que o wCReF fique disponível para realizar as predições de pesquisadores de todo mundo e que também sirva como ferramenta de apoio a práticas educativas. Futuramente visamos a participação deste servidor na próxima edição do CASP (<http://predictioncenter.org>) evento mundial de avaliação de predição de estruturas de proteínas, que acontece a cada dois anos desde 1994, onde a participação e a precisão das previsões dos servidores aumentou desde suas primeiras edições.

O servidor wCReF está em constante desenvolvimento com o objetivo de fornecer as previsões estruturais e funcionais mais precisas. A modelagem no CReF inicia-se pela procura de moldes na base de dados do Protein Data Bank - PDB (<http://www.rcsb.org/pdb>), usando-se como parâmetro a entrada, a sequência de aminoácidos ou estrutura primária no formato FASTA da estrutura não determinada experimentalmente (alvo), realizando a predição de sua estrutura 3D aproximada.

Para saber mais sobre a metodologia do wCReF leia nossa [Documentação](#).

Relate problemas, dúvidas e sugestões na página [fale-conosco](#). Nossos pesquisadores irão enviar um retorno a seu e-mail mais breve possível.

**Saiba como utilizar**  
Leia nosso tutorial para utilização do wCReF.

**Precisa de ajuda?**  
Leia nosso material de ajuda.

**Referências**  
Veja as referências dos artigos publicados.

**Estatísticas**

O servidor completou 72 predições para proteínas submetidas por 16 usuários de 1 país.  
Clique aqui para saber mais.

**Navegação**

- Home
- Cadastro
- Submissões
- Documentação
- Fale conosco

**Documentação**

- Sobre o CReF
- Artigos
- Softwares
- Ajuda

**Links**

- Laboratório de Bioinformática, Modelagem e Simulação de Biosistemas - LABIO
- Programa de Pós-Graduação em Ciência da Computação - PPGCC
- Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS

Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS  
© Copyright 2007-2008 PUCRS. Todos os direitos reservados.

**Figura 28.** Home ou Tela de Apresentação. É a primeira tela que é apresentada quando se acessa o servidor do wCReF.

Em alguns dos servidores estudados, a área de login era apresentada depois que o usuário envia a predição, o que facilitava a ocorrência de erro, ou era apresentada somente na hora da submissão da sequência. Como exemplo, mostramos a interface do I-TASSER (Figura 29). O campo de login é apresentado após o formulário de submissão. As opções de envio da



submissão, os Parâmetros Opcionais não são disponibilizados junto ao formulário de envio da predição.

**Figura 29.** Interface do servidor do I-TASSER. A área de login é apresentada após o formulário de envio da predição.

Portanto, a área de login no wCReF fica em destaque na página inicial, e o botão de login é apresentado em todas as páginas da interface (Figura 30). Também foi considerado que, quando o sistema pedir uma senha no cadastro, a mesma deve ser utilizada para login no Sistema. Há servidor que apresenta essa possibilidade, porém ela está desabilitada ou não funciona.

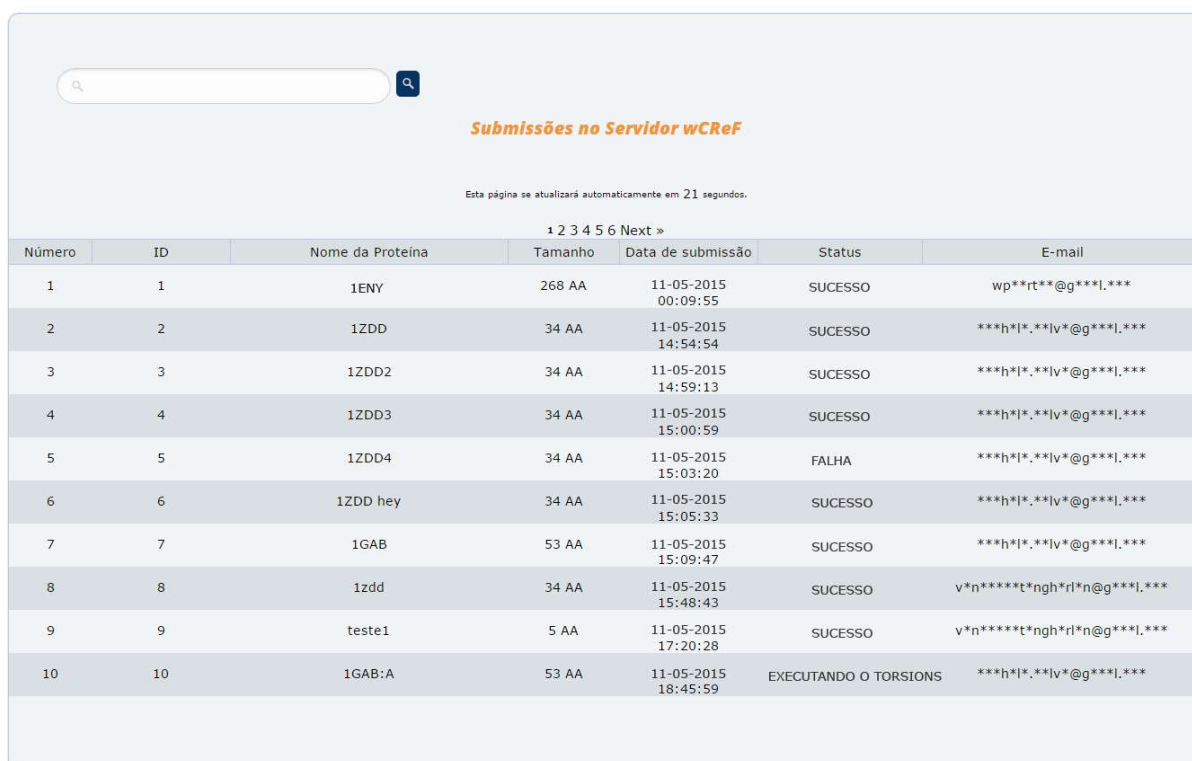
**Figura 30.** O login pode ser realizado de qualquer parte da interface. Aqui o usuário estava na página Fale Conosco e solicitou o login.

Outra orientação quanto ao cadastro é que o mesmo deve ser único (erros ocorreram como, por exemplo, o usuário tinha um cadastro para enviar a predição e outro para participar

do fórum de discussão). Outro ponto a considerar, foi a possibilidade de recuperação dos dados cadastrados, o que também violou a heurística nos servidores Auxiliar, reconhecer e diagnosticar erros. Para isso foi implementado que o usuário possa recuperar sua senha e a identificação se o mesmo já é um usuário cadastrado no sistema.

Não foram definidos os tipos de e-mail a ser utilizado no cadastro do servidor wCReF. Os especialistas sugeriram que o e-mail não deveria ser estritamente acadêmico. Ocorreram testes nos servidores com e-mails não acadêmicos onde o sistema aceitou login, e outros que aceitou e-mail não existentes. A mudança nas permissões poderá permitir que os servidores de predição sejam abertos a outro tipo de público, como estudantes e professores, para ser utilizado como ferramenta educativa.

Seguindo o menu, após o cadastro encontramos as submissões, onde o usuário pode consultar a fila de execução das predições, com o status do andamento da mesma (Figura 31).



Número	ID	Nome da Proteína	Tamanho	Data de submissão	Status	E-mail
1	1	1ENY	268 AA	11-05-2015 00:09:55	SUCESSO	wp**rt**@g***.***
2	2	1ZDD	34 AA	11-05-2015 14:54:54	SUCESSO	***h** *.** v*@g*** .***
3	3	1ZDD2	34 AA	11-05-2015 14:59:13	SUCESSO	***h** *.** v*@g*** .***
4	4	1ZDD3	34 AA	11-05-2015 15:00:59	SUCESSO	***h** *.** v*@g*** .***
5	5	1ZDD4	34 AA	11-05-2015 15:03:20	FALHA	***h** *.** v*@g*** .***
6	6	1ZDD hey	34 AA	11-05-2015 15:05:33	SUCESSO	***h** *.** v*@g*** .***
7	7	1GAB	53 AA	11-05-2015 15:09:47	SUCESSO	***h** *.** v*@g*** .***
8	8	1zdd	34 AA	11-05-2015 15:48:43	SUCESSO	v*n*****t*ngh*r *n@g*** .***
9	9	teste1	5 AA	11-05-2015 17:20:28	SUCESSO	v*n*****t*ngh*r *n@g*** .***
10	10	1GAB:A	53 AA	11-05-2015 18:45:59	EXECUTANDO O TORSIONS	***h** *.** v*@g*** .***

**Figura 31.** Tela de submissão. Aqui o usuário pode acompanhar o andamento de sua predição.

Há um campo para pesquisa, onde pode se realizar uma busca pela data que foi submetida a predição, pelo nome que foi atribuído a proteína ou pelo número de identificação (ID). Um dos requisitos definidos para garantir a visibilidade foi esta possibilidade, de informar ao usuário a lista de Jobs (trabalhos) que estão sendo executados.

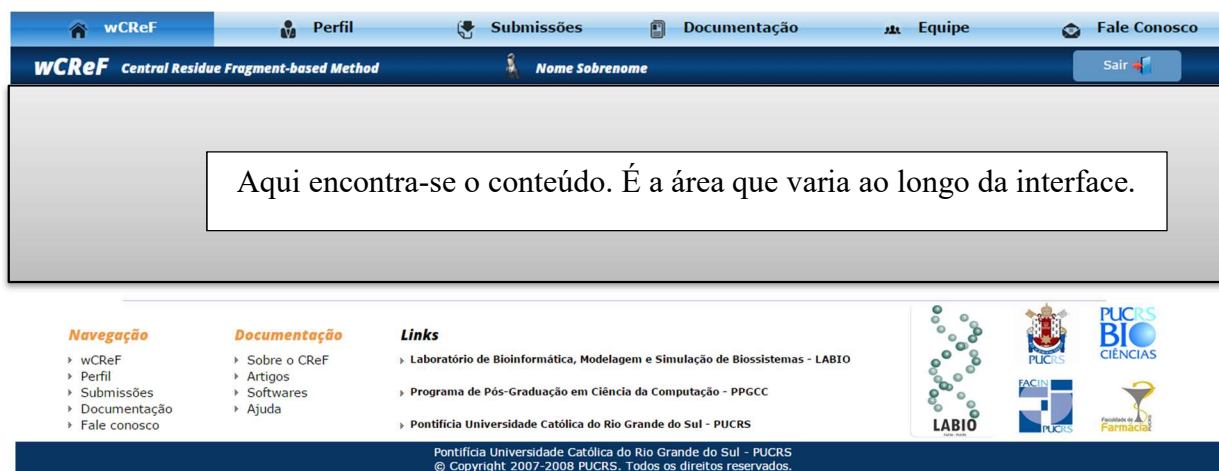
Por questões éticas, o usuário somente pode acompanhar as submissões por essa tela e consultar o status. Para visualizar o resultado da predição, ele precisa estar no seu ambiente de trabalho. Isso é importante pois garante que somente cada usuário poderá ver seus resultados. Além disso o e-mail do usuário não é apresentado em sua totalidade:

- E-mail cadastrado: fulanodetal@email.com
- Forma de apresentação: **f\*1\*n\*d\*t\*1@\*\*\*\*1.\*\*\***

Seguindo o menu principal, a próxima página é a da documentação. Ela é descrita em uma seção posterior, a 6.1.4, juntamente com a ajuda, equipe e fale conosco.

### 5.1.4 Interface de Usuário

É a parte do sistema que é específica para cada usuário, sendo apresentada quando o usuário realiza login. Neste ambiente, o usuário pode enviar sua predição para processamento e acompanhar os resultados. Ele é redirecionado primeiramente para tela inicial do wCReF. O Template da interface de usuário (Figura 32) segue o mesmo padrão do exibido na interface de apresentação, ocorrendo uma pequena modificação, no tamanho do cabeçalho.



**Figura 32.** Template da Interface de usuário do wCReF.

A parte do cabeçalho foi reduzida com intuito de aumentar e dar mais destaque a área de trabalho, ou seja, aquela que o usuário entra em contato enquanto envia, acompanha e manipula seus resultados. Além disso, é acrescentado na barra superior, o nome do usuário que está utilizando o servidor, assim como uma foto de identificação (opcional), e o botão de *logon* ou sair (definido na avaliação). Um dos erros encontrados na Avaliação Heurística foi referente ao excesso de menu e de informações, sem categorização. Isso atrapalha o usuário a focar-se na

predição. Por isso, a redução da área de navegação da ferramenta, dando mais prioridade de visualização ao conteúdo.

Este ambiente é carregado com as ações definidas no cadastro de usuário. Como foi sugerido pelos usuários na avaliação de questionários, ficou definido como perfil o local para os mesmos modificarem seus dados (Figura 33). Pela heurística o controle e liberdade do usuário, foi definido que o usuário pode então modificar seus dados, trocando a senha cadastrada, e modificando seu perfil, além de ter a opção de cancelar sua inscrição no servidor mediante confirmação de exclusão de cadastro.

Você encontra-se em: [Perfil](#)

**Dados de identificação:**

Nome:\*  Sobrenome:\*

Sexo:  
 Masculino  Feminino

Instituição:\*

País:\*

**Dados de login:**

E-mail\*   
 ⚠ Neste campo são aceitos até 50 caracteres.

Imagem de perfil:  
 ⚠ Nenhum arquivo selecionado!

Senha:\*  Confirme sua senha:\*   
 ⚠ Neste campo são aceitos até 8 caracteres alfanuméricos.

ⓘ Nesta página você pode alterar os dados do seu perfil. Mantenha seu e-mail atualizado, pois o wCReF o utiliza para o envio de informações, como avisos, notícias e envio de resultados.

Se você não deseja mais ter uma conta no servidor wCReF clique em excluir. A exclusão será executada após sua confirmação. Lembre-se que a exclusão do seu cadastro é automática, e acarretará a perda dos dados do perfil e das predições realizadas.

Se tiveres alguma dúvida sobre o servidor wCReF pode consultar a ajuda ou utilizar o fale conosco para dúvidas, reclamações e sugestões.

**Figura 33.** O perfil de usuário é carregado automaticamente na interface de usuário. Podem ser realizadas alterações mediante confirmação.

Os campos de entrada para a submissão de um trabalho no servidor wCReF é apresentada na primeira tela (principal) da interface de usuário, denominada wCReF, onde o usuário informa a sequência da proteína alvo a ser predita (Figura 34). As orientações seguidas nesta parte da interface definida na Avaliação Heurística é que o sistema deve:

- Informar qual o tempo que será necessário para a predição.
- Informar ao usuário quando submetida uma proteína que a mesma está sendo gerada.
- A mensagem que a sequência da proteína foi enviada deve ser destacada para que o usuário tenha certeza que submeteu a predição com sucesso.

- Quando há um botão de *clean form* (apagar) como o apresentado nesta tela, deve-se pedir a confirmação do usuário através de mensagem.
- Os campos obrigatórios utilizam uma simbologia usual que indica este tipo de campo. A ajuda rápida quanto aos campos obrigatórios traz a seguinte mensagem: “Preencha o formulário com os dados para a predição da estrutura 3D aproximada. Os campos marcados com \* são de preenchimento obrigatório.”
- O sistema deve apresentar um exemplo do formato FASTA correto na página inicial para direcionar o usuário.

Você encontra-se em: [wCReF](#) > [Submeter](#)

**Submeter**

**Resultados**

**Estatísticas**

**CReF(Central Residue Fragment-based Method)**

Nome do trabalho: \*

Sequência no formato FASTA: \*

Entre com a sua sequência

Exemplo FASTA

**Parâmetros Opcionais**

▶ **Parâmetros de execução do CReF**

▶ **Parâmetros do BLASTp**

**Info** Ao submeter uma predição ao servidor wCReF você recebe como resultado a Estrutura 3D Aproximada de uma Proteína no formato .pdb.

Você receberá um e-mail com os parâmetros de entrada no servidor (Nome da proteína, Sequência submetida e os Parâmetros Opcionais), o local da fila que encontra-se o seu trabalho e o tempo médio para conclusão. Para saber mais sobre os Parâmetros Opcionais, o formato .pdb e o formato FASTA consulte a ajuda em nossa Documentação ou clique no botão de ajuda rápida disponível [?](#) no formulário acima.

Os envios dos trabalhos podem ser acompanhados em [Submissões](#) e depois de concluídas em [Resultados](#). Os resultados também serão enviados para o e-mail cadastrado.

**Navegação**

- ▶ [wCReF](#)
- ▶ [Perfil](#)
- ▶ [Submissões](#)
- ▶ [Documentação](#)
- ▶ [Fale conosco](#)

**Documentação**

- ▶ [Sobre o CReF](#)
- ▶ [Artigos](#)
- ▶ [Softwares](#)
- ▶ [Ajuda](#)

**Links**

- ▶ [Laboratório de Bioinformática, Modelagem e Simulação de Biosistemas - LABIO](#)
- ▶ [Programa de Pós-Graduação em Ciência da Computação - PPGCC](#)
- ▶ [Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS](#)

**LABIO**

**PUCRS**

**FACIN**

**PUCRS**

**PUCRS BIO CIÊNCIAS**

**Faculdade de Farmácia**

Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS  
© Copyright 2007-2008 PUCRS. Todos os direitos reservados.

**Figura 34.** Tela de submissão de predições de proteínas. É nesta parte da interface onde o usuário envia suas predições para o servidor wCReF realizar o processamento. É composta por um formulário com os dados da submissão: Nome da proteína Alvo, sequência no formato FASTA e os Parâmetros Opcionais.

Foi definido um padrão do que é necessário para o envio de um trabalho. Por vezes, a entrada nos servidores analisados era atrelada a conferência do e-mail, senha ou a sequência informada. Também ocorre que mesmo informando a sequência da proteína fora do formato definido, o sistema avaliado preocupa-se mais com a questão do e-mail do que a submissão correta. Neste sentido que foram definidos os parâmetros de entrada para a submissão de um trabalho no formulário do servidor wCReF:

- **Nome da proteína alvo:** O usuário informa o nome da proteína a ser predita.
- **Sequência no formato FASTA:** É um formato usado para dados de sequências derivado das convenções do FASTA, programa de alinhamento rápido (*FAST Alignment*), desenvolvido por W. R. Pearson (Pearson e Lipman, 1988). Muitos programas utilizam o formato FASTA para a leitura de sequências ou para a informação de seus resultados. Foi o principal recurso modificado em relação ao o protótipo através da avaliação de questionários pelos usuários finais foi a validação do formato FASTA. Este erro também ocorreu na avaliação heurística dos servidores, foi apontado na avaliação do wCReF e corrigido.
- **Parâmetros opcionais.** Podem ser de dois tipos: os parâmetros relacionados com a execução do wCReF ou os parâmetros utilizados na execução do BLASTp ((Altschul et al., 1997), descritos na Tabela 17.

**Tabela 17.** Parâmetros Opcionais do servidor.

Nome do Parâmetro	Definição
<b>Parâmetros do CReF:</b>	
Tamanho do fragmento	Define quantos aminoácidos o fragmento irá possuir: 5(padão), 7 ou 9.
Número de clusters a ser usado para os ângulos Phi e Psi	Define o número de clusters do agrupamento é realizado no WEKA ( <i>Waikato Environment for Knowledge Analysis</i> ) buscando as regiões mais favoráveis para formação das estruturas secundárias no mapa de <i>Ramachandran</i> .
Excluir homólogos com similaridade acima	Podem ser excluídas da busca de templates no PDB os fragmentos de proteínas que possuem similaridade acima de 100%, 90%, 80% ou 70%.
Excluir homólogos baseados no código PDB:	Exclui da busca de templates as proteínas que forem informadas através do código PDB: Ex: 1ZDD, 1ENY, 1BVR.
<b>Parâmetros gerais do BLAST:</b>	
<i>Expect threshold</i>	É probabilidade de achar essa sequência num modelo aleatório.
Word size:	O tamanho da sequência somente para encontrar os pontos da sequência em que o BLAST usa para iniciar o alinhamento. Os valores padrão são 2 ou 3.
<b>Parâmetros de pontuação do BLAST:</b>	

Matrix:	Associa uma pontuação para pares de resíduos alinhados e determina a pontuação geral. Deve ser escolhido de acordo com o tipo de sequência que está sendo pesquisada.
GAP Costs:	O custo de um gap (espaço) encontrado no alinhamento de acordo com a Matriz selecionada.
Compositional Adjustment:	Método de ajuste da matriz para compensar a composição de sequências de aminoácidos.

Em visibilidade do sistema foi recomentado que o resultado da predição não deve ser enviado somente por e-mail. Isto dificulta a utilização do servidor como parte de um workflow. O ideal é oferecer uma página com os resultados que podem ser lidos em tempos e tempos para determinar se a predição foi concluída. Neste sentido que foi criada a tela de submissões (Figura 35). A página é atualizada automaticamente a cada 30 segundos. Nesta tela da interface encontram-se somente as predições realizadas pelo usuário. O usuário pode acompanhar também em qual o status da sua predição, em que passo das etapas do CReF a mesma se encontra, e quando finalizada, é apresentada uma nova guia, a de resultados para visualizar, e também há a opção de excluir a predição.

Você encontra-se em: [Submissões](#)

**Submissões no Servidor wCReF**

Esta página se atualizará automaticamente em 19 segundos.

1 2 Next »

Número	ID	Nome da Proteína	Tamanho	Data de submissão	Status	Resultado
1	56	1ZDD	268 AA	2015-06-16 20:57:13	Submissão realizada com sucesso!	Visualizar Excluir
2	57	1ZDD	268 AA	2015-06-16 20:58:19	Submissão realizada com sucesso!	Visualizar Excluir
3	58	1ZDD	35 AA	2015-06-16 20:59:44	Submissão realizada com sucesso!	Visualizar Excluir
4	59	1ZDD	35 AA	2015-06-16 21:00:04	Submissão realizada com sucesso!	Visualizar Excluir
5	60	1ZDD	35 AA	2015-06-16 21:00:09	Submissão realizada com sucesso!	Visualizar Excluir
6	61	1ZDD	35 AA	2015-06-16 21:00:13	Submissão realizada com sucesso!	Visualizar Excluir
7	62	1ZDD	35 AA	2015-06-16 21:00:18	Submissão realizada com sucesso!	Visualizar Excluir
8	63	1ZDD	35 AA	2015-06-16 21:00:29	Submissão realizada com sucesso!	Visualizar Excluir
9	64	1K43	0 AA	2015-06-16 21:39:19	Sua Submissão Falhou!	Enviar novamente Excluir
10	65	1K43	0 AA	2015-06-16 21:40:08	Passo do CReF - 2 Busca por Proteínas Molde.	

**Figura 35.** Tela de acompanhamento das submissões. Conforme o usuário vai enviando seus trabalhos eles vão sendo acrescentados, e apresentados pelo número da submissão, ID, nome da proteína, tamanho (número de aminoácidos) e data de submissão. Também é informado o status de andamento da predição

no servidor (qual passo da metodologia do CReF está sendo executado) e por fim, o Resultado, com a opção de excluir.

Na conclusão da predição os resultados podem ser acessados em Resultados. O resultado encontra-se disponível para *download* ou para visualização na Interface do servidor.

Quando ocorre um erro de envio de predição, este deve ser comunicado ao usuário. Da mesma forma que, se a submissão falhar esta informação também deve ser disponibilizada.

Além disso, uma parte importante definida nas avaliações heurísticas é que, o sistema deve informar como será o retorno da predição. Isso já havia sido contemplado na descrição e na ajuda pela interface, mas também foi acrescentado um exemplo de saída na página de resultados, e essa informação também é enviada por e-mail ao usuário no momento da submissão, e na página inicial de envio.

A predição da proteína aproximada é disponibilizada no formato padrão do Protein Data Bank, com a extensão “.pdb” que é o formato padrão de arquivos do Banco de dados PDB (*Protein Data Bank*), e funciona na maioria dos visualizadores de macromoléculas biológicas. Este formato proporciona um padrão para a representação de dados de estrutura macromolecular derivados de difração por raio-X (*X-Ray Crystallography*) (Barlow, 1883) e estudos de Ressonância Magnética Nuclear (*Nuclear Magnetic Resonance - NMR*) (Rabi et al., 1938). Esta representação foi criada na década de 1970 e uma grande quantidade de *softwares* em bioinformática utilizam esse padrão, principalmente para visualização de estruturas de proteínas. A documentação que descreve o formato PDB está disponível a partir do wwPDB em <http://www.wwpdb.org/documentation/file-format.php>.

Os resultados também podem ser visualizados diretamente na interface. Eles devem ser apresentados de forma clara na ferramenta. As próximas figuras, Figura 36, Figura 37 e Figura 38, demonstram como os resultados são apresentados no servidor wCReF. Está representada a submissão da proteína de código PDB 1ZDD (Starovasnik et al., 1997). Na parte superior da visualização dos resultados encontra-se os dados da predição, ou seja, aqueles informados pelo usuário quando o mesmo enviou sua proteína alvo. Também há a opção de imprimir o resultado em PDF ou realizar o Download no formato PDB (Figura 36).

Logo após os dados da predição na mesma tela há a representação da estrutura secundária (2D). O significado de cada uma das estruturas secundárias é disponibilizado assim como orientado (Figura 37). O software utilizado para a geração da imagem contendo a estrutura secundária foi o POLYVIEW-2D (Porollo et al., 2004). O wCReF envia a predição da estrutura



secundária realizada durante o processamento do método CReF para esta ferramenta e que a devolve em forma de imagem. Esta imagem pode ser visualizada na página de resultados.

Você encontra-se em: [wCReF](#) > [Resultados](#) > [Visualizar](#)

AAO Submeter
[Voltar](#)

Salvar Arquivo PDB
Ver o resultado em PDF

### Resultado da submissão id 58

**Nome da proteína:** 1ZDD Data da submissão: 16-June-2015

**Tamanho:** 35 AA

Sequência submetida no formato **FASTA:**

```
>1ZDD:A|PDBID|CHAIN|SEQUENCE
FNMQCQRRFY EALHDPNLN EEQRNAK IKS IRDDCX
```

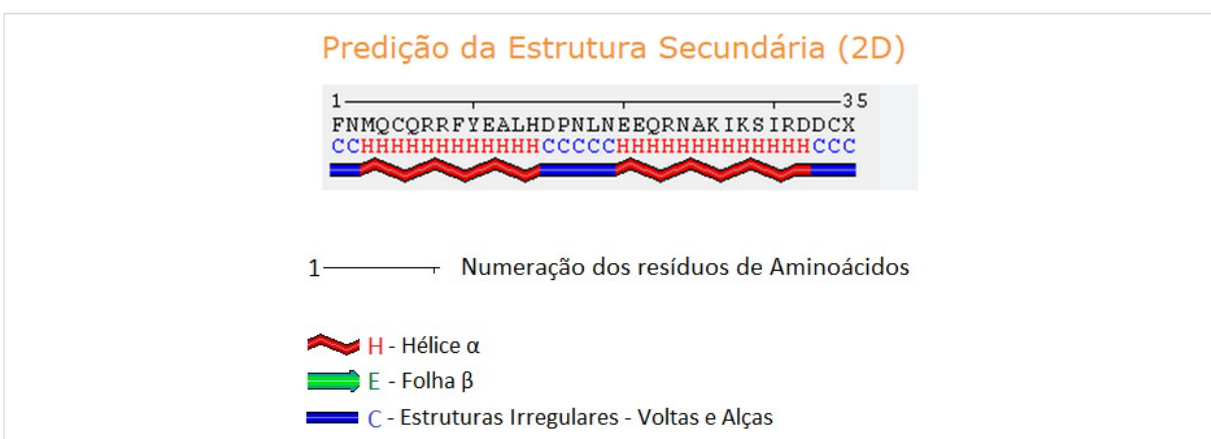
**Parâmetros de execução do CReF:**

Tamanho do fragmento:	Número de clusters:	Exclusão por Similaridade	Homólogos excluídos por código PDB
5	4	100	

**Parâmetros do BLAST:**

Expect threshold:	Word size:	Matrix:	GAP Costs:	Compositional Adjustment:
200000	2	PAM30	ungapped	2

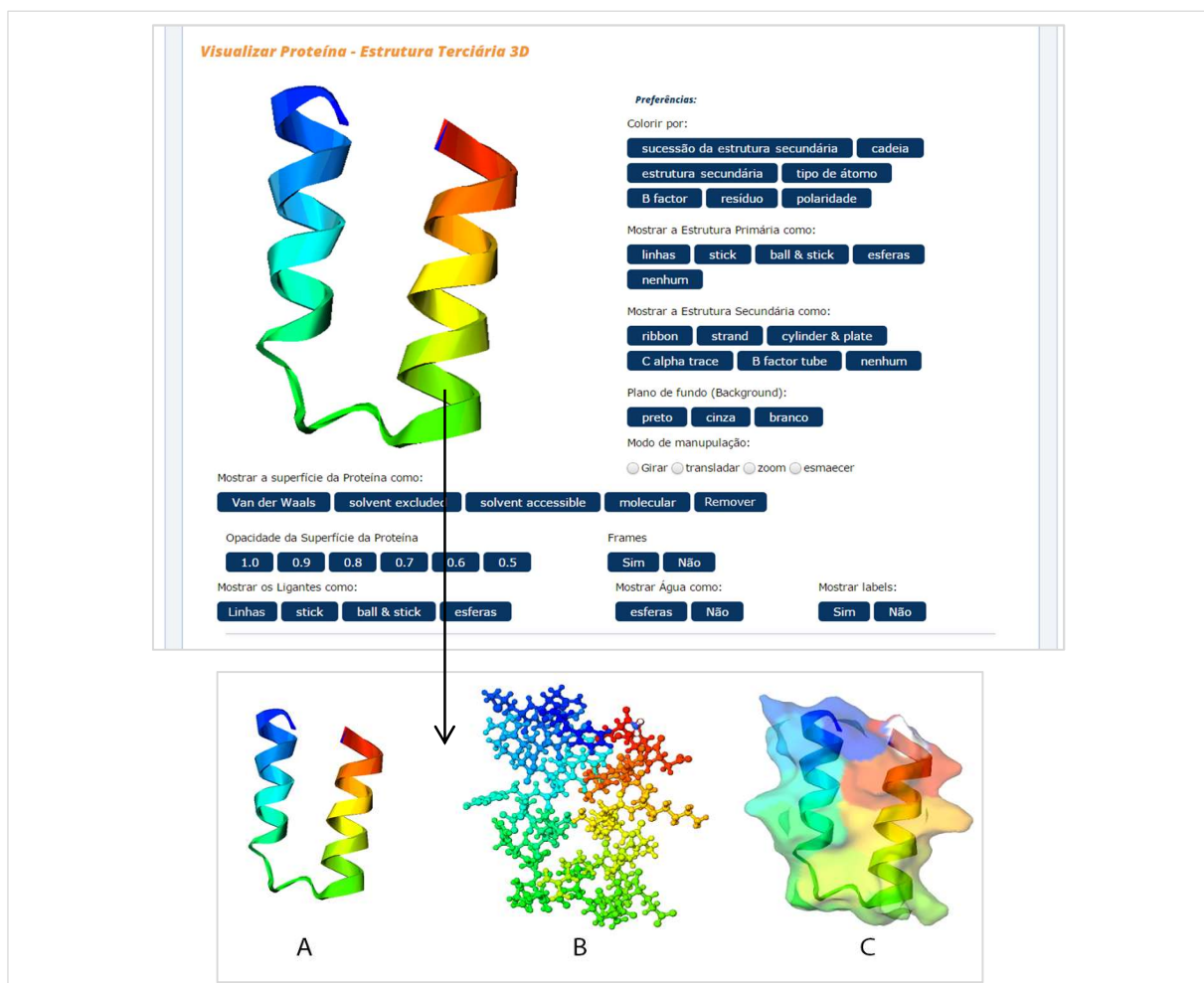
**Figura 36.** Tela de visualização do wCReF. A parte inicial contém os parâmetros que foram informados pelo usuário no momento da submissão.



**Figura 37.** Representação da estrutura secundária. Os resíduos de aminoácidos encontram-se alinhados com a predição da estrutura secundária. As hélices  $\alpha$  são coloridas em vermelho, as folhas  $\beta$  são representadas em verde e as voltas e alças em Azul.

E ao final da página de visualização do wCReF, a Estrutura 3D Aproximada pode ser visualizada de diferentes formas (Figura 38). O software de visualização utilizado no wCReF foi o *iview*, um visualizador interativo de estruturas tridimensionais de moléculas baseado no

WebGL (H. Li et al., 2014). Seus componentes estão ligados a interface do wCReF através de código HTML.



**Figura 38.** Visualizador do wCReF. Aqui está representado a proteína 1ZDD. Na imagem A está representada por *Ribbon* e colorida por Sucessão da Estrutura Secundária. Em B sua representação é do tipo Ball e Stick e em C é demonstrada a superfície molecular.

A página de visualização do wCReF gera a imagem da estrutura secundária com o POLYVIEW-2D e apresenta visualização da estrutura 3d aproximada com o *iview* automaticamente, não necessitando, desta forma, a instalação ou a configuração destes componentes pelo usuário.

Além das páginas descritas até aqui na Interface de Usuário, ela também possui as páginas de documentação, equipe e fale conosco, apresentadas anteriormente na interface de apresentação. Elas seguem o mesmo padrão, para manter a consistência durante a navegação da interface, independentemente da área que o usuário se encontra.

### 5.1.5 Tempo de execução

A apresentação do tempo aproximado de execução de uma predição foi um dos pontos que apareceu nos questionários tanto como sugestão de melhoria quanto como problema encontrado pelos usuários. Existe um requisito especificado no Documento de Requisitos de Software para que a interface wCReF apresente esta informação para o usuário ao submeter a estrutura, porém, o método CReF ainda está sofrendo alterações e não é capaz de prover esta informação, impossibilitando a mesma de ser mostrada na interface. A informação que o wCReF possui a respeito do tempo de execução é imprecisa, pois considera o tempo de espera na fila, que é pode ser variável dependendo da quantidade de submissões sendo feitas concorrentemente e, portanto, deve ser calculada assim que, o método obtiver a consistência necessária para medir cada passo, e apontar o tempo médio da predição.

### 5.1.6 A ajuda e documentação disponibilizada no wCReF

A utilização de sistemas computacionais pelos mais diversos perfis de usuários e a complexidade de muitas das tarefas a serem por eles realizadas com o uso destes sistemas, faz com que seja necessário oferecer-lhes algum tipo de auxílio (Silveira e Leite, 2009). Com base nesta necessidade apresentamos a ajuda e documentação do wCReF.

Na Avaliação Heurística realizada pelos especialistas nos servidores, alguns problemas foram apontados, a saber:

- *“A documentação dos arquivos de entrada e saída dos servidores são muito superficiais e em formato de texto (.txt), o que dificulta o foco de quem somente tem interesse em configurar e visualizar”* – Na interface do wCReF o usuário pode visualizar seus resultados diretamente na interface, sem precisar baixar arquivos no formato txt ou a proteína no formato PDB.
- *“Falta esclarecimento ao usuário o do porquê utilizar e-mail acadêmico e porque ele pode enviar somente um job por vez na documentação”* – Não foi colocada a restrição de e-mail acadêmico no wCReF e há a possibilidade de envio de mais de um job, porém recomenda-se enviar um por vez para não sobrecarregar o servidor.
- *“Poderia ter uma ajuda contextual, para cada campo de entrada”* - O wCReF possui ajuda para os campos de entrada.
- *“Não há ajuda contextuais. Dessa forma resta ao usuário acessar a documentação e tentar localizar o que deseja, sem a ajuda do sistema”*. O wCReF possui ajuda contextual ao logo da interface.

- “Os parâmetros apresentados nos resultados devem ser explicados, através de links e documentação adequada. Há muitos parâmetros como o que significa E ou H na predição da estrutura secundária, ou o que significa SignalP, C-Score, S-Score ou Y-Score, ou do parâmetro TM”. Os parâmetros foram descritos na documentação e também na ajuda rápida ao lado de cada campo de entrada no wCReF.
- “Alguns termos têm links associados, mas é necessário abrir uma outra página para uma explicação de seu significado. Melhor seria ter uma ajuda rápida ao passar o mouse, ou símbolo de ajuda que explicasse um dado em uma linha”. O wCReF possui ajuda rápida e explicações para os links.
- Portanto, os problemas encontrados nos servidores de predição na avaliação heurística, serviram para que na concepção da interface web do CReF estes problemas fossem evitados.

#### 5.1.6.1 Apresentação da ajuda na interface

Silveira e Leite (Silveira e Leite, 2009) trazem em seu artigo as ajudas que são geralmente utilizadas (Houghton Jr, 1984; Patrick e McGurgan, 1993; Silveira, 2002). Baseado em sua descrição, foram definidas as do wCReF:

- **Ajuda embutida:** instruções rotuladas diretamente na interface do usuário. São chamadas na interface do servidor através de um ícone com um ponto de interrogação, representando dúvida;
- **Assistência contextual:** apresentação de ajuda determinada pela localização do usuário no ambiente da aplicação. Em cada parte da interface, se necessário, há uma explicação sobre a determinada página ou o conteúdo referente;
- **Descrição da aplicação:** apresentação de detalhes sobre as funcionalidades, modos de operação e domínio da aplicação (encontrada na documentação);
- **Perguntas frequentes (*Frequently Asked Questions* - FAQs):** apresentação de uma lista de perguntas comumente realizadas pelos usuários; Estas perguntas são disponibilizadas dentro da documentação, no menu ajuda.
- **Tutoriais on-line:** detalhamento do ambiente, com demonstrações de uso do mesmo. Um exemplo de tutorial do servidor encontra-se no APÊNDICE H.
- **Notas:** passos, dicas ou referências a explicações detalhadas sobre determinada funcionalidade da aplicação;
- **Diálogo:** discussão de dúvidas diretamente com a equipe de suporte da aplicação. Pode ser realizada através do envio de dúvidas, reclamações e/ou sugestões pelo fale-

conosco (Figura 39) ou diretamente pelo e-mail acadêmico da equipe de pesquisadores do LABIO (Figura 40);

**Figura 39.** Telas da Interface fale-conosco. O usuário pode obter ajuda com o envio de suas dúvidas, sugestões ou reclamações por formulário.

**Figura 40.** Telas da Interface Equipe. O usuário pode realizar contato por e-mail. O formulário de e-mail é aberto automaticamente ao clicar no ícone do e-mail. O usuário também pode consultar o currículo Lattes da equipe do LABIO.

Como ainda não há em nossa base de consultas uma lista de perguntas frequentes submetidas pelos usuários, foram criadas 15 perguntas referentes ao uso do servidor do wCReF, que serão atualizadas conforme forem surgindo as dúvidas:

- 1) Onde posso buscar ajuda em cada página?
- 2) Como posso enviar uma predição?
- 3) Por que preciso definir o nome da minha proteína?
- 4) O que é o formato FASTA?
- 5) O que é o formato PDB?
- 6) O que são os parâmetros opcionais?
- 7) Quais são os parâmetros do CReF?
- 8) Quais são os parâmetros do BLAST?
- 9) Por que só consigo ver meus resultados quando estou logado no sistema?
- 10) Por que a página visualizar só apresenta a última predição?
- 11) Posso salvar ou imprimir meus resultados?
- 12) Por quanto tempo minha predição fica disponível?
- 13) Quanto tempo demora para se realizar uma predição?
- 14) Como posso alterar meu cadastro ou excluí-lo?
- 15) Como entro em contato com a equipe?

A ajuda embutida está presente em toda interface, com textos curtos com a indicação da tarefa que o usuário deve realizar ou com material informativo, como por exemplo:

O usuário não sabe o que é o formato FASTA para o envio de sua predição. Ele clica no botão de ajuda e abre o seguinte texto:

*“O formato FASTA é um formato padrão usado para informar dados de sequências de nucleotídeos e aminoácidos. Uma sequência em formato FASTA começa com uma descrição de linha única, seguida por linhas de dados da sequência, usando códigos de uma letra. A linha de descrição distingue-se dos dados de sequência por um símbolo de maior no início (“>”). Para saber mais sobre este formato, clique em formato FASTA.”*

Suponhamos que o mesmo não saiba o que é o formato, ele então clica no link com a palavra FASTA. Abre-se então a seguinte janela de orientação (Figura 41):

<< Voltar

## O formato FASTA

É um formato usado para dados de sequências derivado das convenções do FASTA, programa de alinhamento rápido (FAST Alignment), desenvolvido por W. R. Pearson. Muitos programas utilizam o formato FASTA para a leitura de sequências ou para a informação de seus resultados.

Uma sequência no formato FASTA inicia com uma única linha de descrição. O sinal > aparece na primeira coluna. O conteúdo do restante da linha título ou de identificação é arbitrário, mas deve ser informativo. A linha título ou de identificação contém as seguintes informações:

- O sinal > obrigatório na coluna 1.
- A palavra seguinte o símbolo ">" é o identificador de sequência de seu banco de dados. Para cada sequência depositada em um banco de dados existe um identificador único. Veja o [exemplo](#) do formato FASTA da proteína com o código PDB 1ENY.③
- O restante da linha é a descrição (ambos são opcionais).
- As linhas subsequentes contêm a sequência, um caractere por resíduo.

O formato FASTA utiliza o código de uma letra especificado pela União Internacional de Bioquímica e pela União Internacional de Química Pura e Aplicada (IUB/IUPAC) para representar nucleotídeos ou aminoácidos. Saiba mais:

- › IUB-IUPAC: Comissão em Nomenclatura Bioquímica - Abreviaturas e símbolos para Ácidos Nucléicos, Polinucleotídeos e seus constituintes
- › Nomenclatura e simbolismo para aminoácidos e peptídeos

Após o término da linha de identificação, seque a sequência com o código de uma letra dos aminoácidos:

### Um exemplo de formato FASTA

```
>1ENY: A | PDBID | CHAIN | SEQUENCE
AGLLDGKRIILVSGIITDSSIAFHARVAQEQQGQLVLTGFDRRLRIQRITDRLP
AKAPLLELDVQNEEHLASLAGRVTEAIGAGNKLDGVVHSGIFMPQTGMGINPFF
DAPYADVSKGIHISAYSASYASMAKALLPIMNPGGSIVGMDFDPSRAMPAYNWMTV
AKSALESVNRVAREAGKYGVRSNLVAAGPIRTLAMSAIVGGALGEEAGAQIQLE
LEEGWDQRAPIGWNMKDATPVAKTVCALLSDWLPATTDGDIYADGGAHTQLL
```

---

**Sobre a 1ENY:**

- › Veja a 1ENY no PDB
- › Referências

**Figura 41.** Exemplo da ajuda da interface do wCReF. Parte da Tela contendo uma breve explicação sobre o Formato FASTA.

### 5.1.6.2 Documentação do Servidor

Finalizamos a apresentação da interface do wCReF com a documentação, que está dividida nas seguintes partes:

- Sobre o CReF: É um resumo da metodologia do método CReF, com as 9 etapas do algoritmo que é utilizado para sua execução. Há a descrição das 9 etapas, referenciadas com os devidos autores (Figura 42). Como sugestão descrita na avaliação por questionários, além da descrição de todos os passos do wCReF, uma imagem representando todas as etapas foi adicionada.
- Referências: São as referências para os artigos publicados do método. Para cada artigo, há sua referência assim como o link para o *journal* ou revista que o mesmo foi publicado para consulta (Figura 43). Além disso, as referências encontram-se disponíveis para cópia, no formato utilizado para referências: Elsevier (Elsevier, 2015), Nature (Nature, 2015), IEEE (IEEE, 2015), Norma da ABNT NBR6023

(ABNT, 2002) e para download no formato dos editores de referências BibTeX (BibTeX, 2006) e EndNote (EndNote, 2014).

- Softwares: Informações sobre todos os recursos utilizados na execução do wCReF, desde a linguagem de programação, as bibliotecas e as aplicações. Há também um link para as páginas oficiais ou grupo de pesquisa que estão relacionados com cada um destes recursos computacionais (Figura 44).
- O último item disposto na documentação é a ajuda, onde encontram-se as perguntas frequentes, relacionadas ao método. Este e outros pontos relacionados a ajuda foram explicadas na seção anterior.
- Com a descrição da documentação nesta subseção, dar-se por finalizada a apresentação da interface do servidor wCReF.

The image shows the wCReF website interface. The main content area displays the 'CReF - Central Residue Fragment-based Method' documentation. It includes a navigation menu (Home, Cadastro, Submissões, Documentação, Equipe, Fale Conosco) and a search bar. The documentation text describes the method's purpose and workflow. A flowchart on the right side of the page illustrates the 9 steps of the algorithm: 1) Fragmentação da sequência alvo, 2) Busca por Proteínas molde no PDB, 3) Cálculo dos ângulos de torção dos dupletos, 4) Agrupamento de dupletos, 5) Representação dos ângulos de torção na forma de intervalos, 6) Classificação dos grupos em regiões ocupadas no mapa de Ramachandran, 7) A estrutura secundária é predita através do servidor de predição, 8) É construída e representada à conformação inicial na forma de intervalos, 9) Otimização das regiões de volta. A list of steps is also provided on the right side of the page, and a citation for the method is shown at the bottom.

**Figura 42.** Sobre o CReF. Aqui encontram-se as 9 etapas do algoritmo de funcionamento do CReF. Todo material citado aqui é referenciado, basta o usuário clicar na referência que uma janela se abre com os dados referentes de autoria.



The screenshot shows the 'Artigos' (Articles) section of the CReF website. A sidebar on the left contains navigation links: 'Sobre o CReF', 'Artigos', 'Softwares', and 'Ajuda'. The main content area displays two article titles: 'CReF: A central-residue-fragment-based method for predicting approximate 3-D polypeptides structures' and 'A Hybrid Method for the Protein Structure Prediction Problem'. A 'Citar Link' button is visible next to the first article. An inset window titled 'Citar' is open, showing citation information for the first article in various formats: Elsevier, Nature, IEEE, and NBR6023. At the bottom of the window are buttons for 'BibTex' and 'EndNote'.

**Figura 43.** As referências dos artigos do CReF. O usuário pode copiar ou baixar as referências no formato Bibtext ou EndNote. Há também um link para o local onde encontra-se publicado o artigo.

The screenshot shows the 'Softwares' section of the CReF website. A sidebar on the left contains navigation links: 'Sobre o CReF', 'Artigos', 'Softwares', and 'Ajuda'. The main content area lists various software-related resources under the heading 'Softwares'. These include 'Sistema Operacional' (Linux), 'Linguagem de programação e bibliotecas' (Python, NumPy, BioPython), 'Banco de Dados e Ferramentas' (BLASTp, PDB), and 'Softwares' (Torsions, WEKA, AMBER, Jmol). A 'Linux' article is highlighted, showing a detailed description of GNU-Linux, its history, and links to official websites and a Wikipedia page.

**Figura 44.** Softwares relacionados com a metodologia do CReF. Para cada recurso uma janela adicional é apresentada, com a referência, os links para página oficial ou grupo de pesquisa. Nesta imagem foi selecionado o sistema operacional Linux.

## 6 CONSIDERAÇÕES FINAIS

O primeiro passo na aplicação de metodologias e princípios para garantir a usabilidade de interfaces deve ser obter uma melhor compreensão do comportamento dos usuários, suas experiências, tarefas e contexto de trabalho. Para o desenvolvimento da interface Web do wCReF foi muito importante, desde sua concepção, iniciando na fase do levantamento de requisitos de software e até as etapas finais de desenvolvimento, que avaliações de usabilidade fossem realizadas.

Para isso, o levantamento de requisitos de software foi feito através da Avaliação Heurística, onde os especialistas apontaram os principais problemas de usabilidade encontrados nas interfaces dos servidores de predição de estruturas de proteínas avaliados (Robetta, QUARK E I-TASSER). Com a descrição dos erros e a observação destas interfaces, ficou claro quais as características que o novo servidor deveria possuir, e quais problemas deveriam ser evitados durante sua criação.

O próximo passo foi a implementação do protótipo seguindo as orientações dos especialistas. Optamos, na avaliação do protótipo, realizar testes com os usuários finais para agregar sua opinião a dos especialistas. Dessa forma, conseguimos captar a opinião do sujeito mais importante na concepção deste produto, o usuário. Também foi possível apontar os problemas de usabilidade da interface do wCReF.

A avaliação do protótipo, por meio da adaptação do questionário de Ssemugabi, foi positiva. Do total das 972 questões avaliadas, aproximadamente 80% ficaram avaliadas com *scores* positivos, o que sugere que o protótipo já contemplava uma boa usabilidade em sua concepção. Podemos notar que o principal ponto positivo, citado nas questões abertas, é que a interface é “fácil de usar”. Como uma das características da usabilidade é a facilidade de uso, acreditamos que conseguimos atingir nosso objetivo nesta etapa.

Com os apontamentos realizados, a interface do wCReF foi melhorada, sendo complementada com as sugestões dos usuários e a correção dos problemas de usabilidade. Como resultado final então, a interface do wCReF foi apresentada.

Isso permitiu que a interface Web atingisse os objetivos propostos nesta dissertação, priorizando a usabilidade como foco principal da aplicação. Campos e Matias (Campos e Matias, 2012) afirmaram que técnicas como análise heurística e pequenos testes com usuários podem ser muito reveladores para qualquer desenvolvedor. E neste caso, tanto a realização da Avaliação Heurística quanto as avaliações feitas com usuários finais foram o diferencial que contribuiu para um desenvolvimento satisfatório desta interface. Os resultados apoiam a

hipótese inicial, que o estudo da usabilidade desde a concepção até o produto final, por meio da avaliação de usabilidade, pôde garantir a interface do wCReF uma boa usabilidade, focada nos seus usuários.

Além disso, outros pontos a considerar são:

- A simplicidade da interface, com o objetivo de fornecer funcionalidades avançadas sem impor a curva de aprendizado acentuada geralmente associada a sistemas complexos. Por exemplo, os dados referentes à predição encontram-se dispostos em uma única janela, não contendo outras informações que podem atrapalhar o usuário, sendo enviadas diretamente para o servidor wCReF e nem necessidade de informar o e-mail e a senha a cada nova submissão.
- O wCReF permite a execução da metodologia CReF em uma ambiente Web, sem a necessidade da instalação local de todos os pacotes e programas relacionados com o seu uso, facilitando a utilização do método pelo usuário e reduzindo os custos associados à manutenção de ambiente computacional para todos os programas. Isto também contribui para as novas atualizações sendo implementadas no CReF através de um trabalho de doutorado, visando aperfeiçoar a metodologia. Aliás, os resultados desta nova abordagem já são visíveis, pois graças a este trabalho, realizado em conjunto com os integrantes do LABIO, a integração da interface Web com os recursos do método tornou-se possível. E isto, é claro, sempre visando uma melhoria da qualidade da predição da proteína alvo, ganho de velocidade na execução da metodologia e uso de tecnologias mais atuais, e possibilitando a participação nas próximas edições do CASP.

Acreditamos este estudo resultou em uma aplicação fácil de interagir, de usar e que poderá ser utilizada tanto por usuários mais experientes, em suas pesquisas científicas, quanto por usuários menos experientes, servindo inclusive como uma ferramenta de apoio ao ensino de bioinformática.

O wCReF será publicado oficialmente na Web no momento em que as modificações do método CReF estiverem concluídas.

## **6.1 Trabalhos futuros**

Esta ferramenta fará parte de um pacote de aplicativos em Bioinformática disponibilizados através da Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS, pelo Laboratório de Bioinformática, Modelagem e Simulação de Biosistemas – LABIO. Com a ferramenta implementada, poderá ser um dos servidores de predição avaliados nas edições

futuras do CASP. Além disso, novas avaliações de usabilidade poderão ser aplicadas ao wCReF. Fica a sugestão da avaliação heurística na interface final do wCReF, e a comparação com outros servidores de predição. Outro ponto a ser considerado é como criar metodologias ou heurísticas para avaliação de *softwares* em bioinformática, mais especificamente na área da bioinformática estrutural e os servidores de predição de proteínas. Por último, a agregação de outras ferramentas e funcionalidades, através de trabalhos que estão sendo realizados pela equipe de pesquisadores do LABIO, como refinamento da estrutura através de dinâmica molecular, diretamente na interface do servidor.

## REFERÊNCIAS

- ABNT, A.B.N.T., 2002. *NBR 6023: Informação e documentação - Referência - Elaboração*. Associação Brasileira de Normas Técnicas, Rio de Janeiro, Brasil, 24 p.
- Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J., 1997. *Gapped BLAST and PSI-BLAST: a new generation of protein database search programs*. *Nucleic Acids Research*, v. 25, n. 17, p. 3389-3402.
- Anfinsen, C.B., 1973. *Principles that govern the folding of protein chains*. *Science*, v. 181, n. 4096, p. 223-230.
- Anson, M.L., Mirsky, A.E., 1925. *On some general properties of proteins*. *The Journal of General Physiology*, v. 9, n. 2, p. 169-179.
- Araújo, N.D., Farias, R.P., Pereira, P.B., Figueiredo, F.M., Morais, A.M.B., Saldanha, L.C., Gabriel, J.E., 2008. *A era da Bioinformática: seu potencial e suas implicações para as ciências da saúde*. *Estudos de Biologia*, v. 30, n. 70-72, p. 143-148.
- Barlow, W., 1883. *Probable nature of the internal symmetry of crystals*. *Nature*, v. 29, n. 186-188, p. 205-207.
- Berman, H.M., Kleywegt, G.J., Nakamura, H., Markley, J.L., 2013. *How community has shaped the Protein Data Bank*. *Structure*, v. 21, n. 9, p. 1485-1491.
- Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E., 2000. *The Protein Data Bank*. *Nucleic Acids Research*, v. 28, n. 1, p. 235-242.
- Bevan, N., 1995. *Usability is quality of use*. *Advances in Human Factors Ergonomics*, v. 20, p. 349-354.
- BibTeX, 2006. *Your BibTeX resource*. Disponível em: <http://www.bibtex.org>. Acessado em: 09/06/2015.
- Bisson, G., Garreau, A., 1995. *APIC: a generic interface for sequencing projects*. In: *Third International Conference on Intelligent Systems for Molecular Biology (III ISMB)*, Cambridge, England, UK, p. 57-65.
- Bolchini, D., Finkelstein, A., Perrone, V., Nagl, S., 2009. *Better Bioinformatics through usability analysis*. *Bioinformatics* v. 25, n. 3, p. 406-412.
- Bowie, J.U., Lüthy, R., Eisenberg, D., 1991. *A method to identify protein sequences that fold into a known three-dimensional structure*. *Science*, v. 253, n. 5016, p. 164-170.
- Campos, P.S, Matias, M., 2012. *Avaliação de usabilidade de sites web*. *Revista Caminhos*, v. 3, n. 5, p. 189-203.
- Chu, S.Y., 1983. *Banco de dados: organização, sistemas e administração*. 3ª edição. Editora Atlas, São Paulo, Brasil, 398 p.
- Cock, P.J.A., Antao, T., Chang, J.T., Chapman, B.A., Cox, C.J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., Hoon, M.J.L., 2009. *Biopython: freely available python tools for computational molecular biology and Bioinformatics*. *Bioinformatics*, v. 25, n. 11, p. 1422-1423.
- Conforto, D., Santarosa, L.M., 2002. *Acessibilidade à web: internet para todos*. *Informática na Educação: Teoria & Prática*, v. 5, n. 2, p. 87-102.

- Cuperschmid, A.R.M., Hildebrand, H.R., 2013. *Heurísticas de jogabilidade: usabilidade e entretenimento em jogos digitais*. 1ª edição. Editora Marketing Aumentado, São Paulo, Brasil, 123 p.
- Dall'Agno, K.C.M., 2012. *Um estudo sobre a predição da estrutura 3D aproximada de proteínas utilizando o método CReF com refinamento*. Dissertação de Mestrado. Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre, Rio Grande do Sul, Brasil, 132 p.
- Dall'Agno, K.C.M., Norberto de Souza, O., 2013. *An expert protein loop refinement protocol by molecular dynamics simulations with restraints*. Expert Systems with Applications, v. 40, n. 7, p. 2568-2574.
- Date, C.J., 1985. *Banco de dados: fundamentos*. 1ª edição. Editora Campus, Rio de Janeiro, Brasil, 214 p.
- Dill, K.A., MacCallum, J.L., 2012. *The Protein-Folding Problem, 50 years on*. Science, v. 338, n. 6110, p. 1042-1046.
- Dinh, H., Rajasekaran, S., 2013. *PMS: A Panoptic Motif Search Tool*. PLoS One, v. 8, n. 12, p. e80660.
- Dorn, M., Norberto de Souza, O., 2010. *Mining the Protein Data Bank with CReF to predict approximate 3-D structures of polypeptides*. International Journal of Data Mining and Bioinformatics, v. 4, n. 3, p. 281-299.
- Dorn, M., Breda, A., Norberto de Souza, O., 2008. *A hybrid method for the protein structure prediction problem*. In: Advances in Bioinformatics and Computational Biology (BSB 2008), Santo André, São Paulo, Brazil, p. 47-56.
- Dr. Andrew C.R. Martin's Group, 2014. *Welcome to Andrew C.R. Martin's Bioinformatics Group at UCL*. Disponível em: <http://www.bioinf.org.uk/index.html>. Acessado em: 01/06/2014.
- Elsevier, 2015. *Journals*. Disponível em: <http://www.elsevier.com/journals/title/a>. Acessado em: 09/06/2015.
- EndNote, 2014. *EndNote*. Disponível em: <http://endnote.com>. Acessado em: 09/06/2015.
- Faisca, P.F.N., 2005. *O mistério da forma das proteínas*. Gazeta de Física, v. 29, p. 34-39.
- Ferreira, A.S., Silva, E.P., Barros, R.A., 2010. *Interfaces da informação e comunicação: possibilidades de inclusão digital do aluno com deficiência intelectual*. Em: V Encontro de Pesquisa em Educação de Alagoas (V EPEAL), Maceió, Alagoas, Brasil, p. 1-16.
- Filho, S., Andrade, O., Alencastro, R.B., 2003. *Protein homology modeling*. Química Nova, v. 26, n. 2, p. 253-259.
- Fischer, E., 1894. *Einfluss der configuration auf die wirkung der enzyme*. European Journal of Inorganic Chemistry, v. 27, n. 3, p. 2985-2993.
- Fiser, A., 2009. *Comparative protein structure modelling*. Rigden, D.J. (Ed.). From Protein Structure to Function with Bioinformatics. Springer, Dordrecht, Netherlands, p. 57-90.
- Fiser, A., Do, R.K., Sali, A., 2000. *Modeling of loops in protein structures*. Protein Science, v. 9, n. 9, p. 1753-1773.
- Frank, E., Hall, M., Trigg, L., Holmes, G., Witten, I.H., 2004. *Data mining in Bioinformatics using Weka*. Bioinformatics, v. 20, n. 15, p. 2479-2481.

- Frank, E., Witten, I.H., Hall, M., Holmes, G., Kirkby, R., Pfahringer, B., Trigg, L., 2005. *Weka*. Maimon, O., Rokach, L. (Eds.). *Data Mining and Knowledge Discovery Handbook*. Springer, New York, USA, p. 1305-1314.
- Graham, I.S., 1995. *The HTML sourcebook*. 2nd edition. John Wiley & Sons, New York, USA, 432 p.
- Guex, N., 1996. *Swiss-PdbViewer: a new fast and easy to use PDB viewer for the macintosh*. *Experientia*, v. 52, p. A26.
- Henikoff, S., Henikoff, J.G., 1992. *Amino acid substitution matrices from protein blocks*. *Proceedings of the National Academy of the United States of America*, v. 89, n. 22, p. 10915-10919.
- Houghton Jr, R.C., 1984. *Online help systems: a conspectus*. *Communications of the ACM*, v. 27, n. 2, p. 126-133.
- IEEE, 2015. *IEEE Advancing technology for humanity*. Disponível em: <https://www.ieee.org/index.html>. Acessado em: 09/06/2015.
- ISO 9241, 2008. *ISO 9241-151:2008 - Ergonomics of human-system interaction - Part 151: guidance on World Wide Web user interfaces*. Disponível em: <https://www.iso.org/standard/37031.html>. Acessado em: 27/11/2014.
- ISO/IEC, S., 1991. *ISO/IEC 9126: Information technology - Software product evaluation - Quality characteristics and guidelines for their use*. International Organization for Standardization, Geneva, Switzerland, 28 p.
- Javahery, H., Seffah, A., Radhakrishnan, T., 2004. *Beyond power: making Bioinformatics tools user-centered*. *Communications of the ACM*, v. 47, n. 11, p. 58-63.
- Jones, D.T., Taylor, W.R., Thornton, J.M., 1992. *A new approach to protein fold recognition*. *Nature*, v. 358, n. 6381, p. 86-89.
- Källberg, M., Margaryan, G., Wang, S., Ma, J., Xu, J., 2014. *RaptorX server: a resource for template-based protein structure modeling*. *Protein Structure Prediction*, v. 1137, p. 17-27.
- Kelley, L.A., 2009. *Fold Recognition*. Rigden, D.J. (Ed.). *From Protein Structure to Function with Bioinformatics*. Springer, Dordrecht, Netherlands, p. 27-55.
- Kendrew, J.C., Bodo, G., Dintzis, H.M., Parrish, R.G., Wyckoff, H., Phillips, D.C., 1958. *A three-dimensional model of the myoglobin molecule obtained by x-ray analysis*. *Nature*, v. 181, n. 4610, p. 662-666.
- Kendrew, J.C., Dickerson, R.E., Strandberg, B.E., Hart, R.G., Davies, D.R., Phillips, D.C., Shore, V.C., 1960. *Structure of myoglobin: a three-dimensional fourier synthesis at 2 Å resolution*. *Nature*, v. 185, n. 4711, p. 422-427.
- Kim, D.E., Chivian, D., Baker, D., 2004. *Protein structure prediction and analysis using the Robetta server*. *Nucleic Acids Research*, v. 32, n. 2, p. W526-W531.
- King, R.D., Sternberg, M.J., 1996. *Identification and application of the concepts important for accurate and reliable protein secondary structure prediction*. *Protein Science*, v. 5, n. 11, p. 2298-2310.
- Kollman, P., Pearlman, D.A., Case, D.A., Caldwell, J.W., Ross, W.S., Cheatham III, T.E., Debolt, S., Ferguson, D., Seibel, G., 1995. *AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy*

*calculations to simulate the structural and energetic properties of molecules.* Computer Physics Communications, v. 91, n. 1-3, p. 1-41.

Laskowski, R.A., MacArthur, M.W., Moss, D.S., Thornton, J.M., 1993. *PROCHECK: a program to check the stereochemical quality of protein structures.* Journal of Applied Crystallography, v. 26, n. 2, p. 283-291.

Lee, J., Kim, S.Y., Joo, K., Kim, I., Lee, J., 2004. *Prediction of protein tertiary structure using PROFESY, a novel method based on fragment assembly and conformational space annealing.* Proteins: Structure, Function, and Bioinformatics, v. 56, n. 4, p. 704-714.

Lee, J., Wu, S., Zhang, Y., 2009. *Ab Initio protein structure prediction.* Rigden, D.J. (Ed.). From Protein Structure to Function with Bioinformatics. Springer, Dordrecht, Netherlands, p. 3-25.

Leite, J.C., 1998. *Modelos e formalismos para a engenharia semiótica de interfaces de usuário.* Tese de Doutorado. Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), Rio de Janeiro, Rio de Janeiro, Brasil, 194 p.

Lesk, A.M., 2008. *Introdução à Bioinformática.* 2º edição. Editora Artmed, Porto Alegre, Brasil, 384 p.

Letondal, C., Amanatian, O., 2004. *Participatory design of pipeline tools and web services in Bioinformatics.* In: Requirements Capture for Collaboration in eScience, Edimburgo, Scotland, UK, p. 1-4.

Levinthal, C., 1968. *Are there pathways for protein folding?* Journal de Chimie Physique, v. 65, p. 44-45.

Li, H., Leung, K.S., Nakane, T., Wong, M.H., 2014. *Iview: an interactive WebGL visualizer for protein-ligand complex.* BMC Bioinformatics, v. 15, n. 1, p. 56.

Li, J., Bhattacharya, D., Cao, R., Adhikari, B., Deng, X., Eickholt, J., Cheng, J., 2014. *The MULTICOM protein tertiary structure prediction system.* Methods in Molecular Biology, v. 1137, p. 29-41.

Li, Z., Scheraga, H.A., 1987. *Monte Carlo-minimization approach to the multiple-minima problem in protein folding.* Proceedings of the National Academy of Sciences, v. 84, n. 19, p. 6611-6615.

Likert, R., 1972. *Likert technique for attitude measurement.* Sahakian, W.S. (Ed.). Social Psychology: Experimentation, Theory, Research. Intext Educational, Scranton, USA., p.101-119.

Lipinski, T.P., 2013. *Pro-Smart: predição de estruturas terciárias de proteínas utilizando sistemas multiagente.* Dissertação de Mestrado. Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre, Rio Grande do Sul, Brasil, 133 p.

Luscombe, N.M., Greenbaum, D., Gerstein, M., 2001. *What is Bioinformatics? A proposed definition and overview of the field.* Methods of Information in Medicine, v. 40, n. 4, p. 346-358.

Maggio, E.T., Ramnarayan, K., 2001. *Recent developments in computational proteomics.* Drug Discovery Today, v. 6, n. 19, p. 996-1004

Marks, D.S., Hopf, T.A., Sander, C., 2012. *Protein structure prediction from sequence variation.* Nature Biotechnology, v. 30, n. 11, p. 1072-1080.



- Marti-Renom, M.A., Yerkovich, B., Sali, A., 2002. *Comparative protein structure prediction*. Current Protocols in Protein Science, n. 2, p. 2-9.
- Meyer, E.A., 2000. *Cascading Style Sheets: the definitive guide*. 1st edition. O'Reilly Media, Sebastopol, USA, 172 p.
- Mirel, B., Wright, Z., 2009. *Heuristic evaluations of Bioinformatics tools: a development case*. In: Human-Computer Interaction (HCI 2009), San Diego, California, USA, p. 329-338.
- Moran, T.P., 1981. *The command language grammar: a representation for the user interface of interactive computer systems*. International Journal of Man-machine Studies, v. 15, n. 1, p. 3-50.
- Morris, A.L., MacArthur, M.W., Hutchinson, E.G., Thornton, J.M., 1992. *Stereochemical quality of protein structure coordinates*. Proteins: Structure, Function, and Bioinformatics, v. 12, n. 4, p. 345-364.
- Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., Tramontano, A., 2014. *Critical assessment of methods of protein structure prediction (CASP): round x*. Proteins: Structure, Function, and Bioinformatics, v. 82, n. S2, p. 1-6.
- Moult, J., Hubbard, T., Bryant, S.H., Fidelis, K., Pedersen, J.T., 1997. *Critical assessment of methods of protein structure prediction (CASP): round II*. Proteins, v. 23, n. 1, p. 2-6.
- Moult, J., Kryshtafovych, A., Venclovas, C., Fidelis, K., 2005. *Progress over the first decade of CASP experiments*. Proteins, v. 61, n. 7, p. 225-236.
- Moult, J., Pedersen, J.T., Judson, R., Fidelis, K., 1995. *A large-scale experiment to assess protein structure prediction methods*. Proteins: Structure, Function, and Bioinformatics, v. 23, n. 3, p. 2-4.
- Nature, 2015. *Nature*. Disponível em: <http://www.nature.com>. Acessado em: 10/05/2015.
- NBR, A., 2002. *NBR 9241-11 - Requisitos ergonômicos para trabalho de escritórios com computadores: parte 11 - Orientações sobre usabilidade*. Associação Brasileira de Normas Técnicas, Rio de Janeiro, Brasil, 21 p.
- Néron, B., Ménager, H., Maufrais, C., Joly, N., Maupetit, J., Letort, S., Carrere, S., Tuffery, P., Letondal, C., 2009. *Mobylye: a new full web Bioinformatics framework*. Bioinformatics, v. 25, n. 22, p. 3005-3011.
- Netto, O., 2004. *Interação Humano-Computador - Modelagem e gerência de interface com o usuário*. 1º edição. Editora VisualBooks, Florianópolis, Brasil, 120 p.
- Nicholl, A.J.R., Filho, J.B., 2001. *Acessibilidade e usabilidade de equipamento telemático*. Revista Ação Ergonômica (ABERGO), v. 1, n 1, p. 1-3.
- Niedeauer, J., 2006. *MySQL 5 - Guia de consulta rápida*. 1ª edição. Editora Novatec, São Paulo, Brasil, 110 p.
- Nielsen, J., 2001. *How to conduct a heuristic evaluation: article by Jakob Nielsen*. Disponível em: <https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation>. Acessado em: 02/03/2015.
- Nielsen, J., 1995. *10 heuristics for user interface design: article by Jakob Nielsen*. Disponível em: <http://www.nngroup.com/articles/ten-usability-heuristics>. Acessado em: 09/08/2014.

- Nielsen, J., 1994a. *Usability engineering. 1st edition*. Elsevier, San Francisco, USA, 362 p.
- Nielsen, J., 1994b. *Usability inspection methods*. In: ACM Conference on Human Factors in Computer Systems (CHI 1994), Boston, Massachusetts, USA, p. 413-414.
- Nielsen, J., Molich, R., 1990. *Heuristic evaluation of user interfaces*. In: ACM Conference on Human Factors in Computing Systems (CHI 1990), Seattle, Washington, USA, p. 249-256.
- Novaes, B.C.S., Scott, L.P.B., 2009. *Modelagem molecular e docking de proteína - ligante*. Em: II Simpósio de Iniciação Científica da Universidade Federal do ABC (II UFABC), Santo André, São Paulo, Brasil, p.1-3.
- Oliveira, E. S., Lima, C.R.B., 2013. *Realce das normas e padrões: a usabilidade como fator primordial para a boa interatividade do usuário*. Caderno de Ciências Humanas e Sociais Aplicadas, n. 1, p. 1-17.
- ONU, O. N. U., 2006. *Convention on the rights of persons with disabilities*. Disponível em: <http://www.un.org/disabilities/default.asp?id=150>. Acessado em: 11/12/2013.
- Oracle Corporation, 1995. *MySQL: the world's most popular open source database*. Disponível em: <http://www.mysql.com>. Acessado em: 06/02/2014.
- Orth, A.I., 2005. *Interface homem-máquina*. 1ª edição. Editora AIO, Porto Alegre, Brasil, 280 p.
- Ortiz, A.R., Kolinski, A., Skolnick, J., 1998. *Fold assembly of small proteins using Monte Carlo simulations driven by restraints derived from multiple sequence alignments*. Journal of Molecular Biology, v. 277, n. 2, p. 419-448.
- Park, S.H., Mrse, A.A., Nevzorov, A.A., Mesleh, M.F., Oblatt-Montal, M., Montal, M., Opella, S.J., 2003. *Three-dimensional structure of the channel-forming trans-membrane domain of virus protein "u" (Vpu) from HIV-1*. Journal of Molecular Biology, v. 333, n. 2, p. 409-424.
- Passerino, L.M., Montardo, S.P., 2007. *Inclusão social via acessibilidade digital: proposta de inclusão digital para pessoas com necessidades especiais*. Em: XI Colóquio Internacional sobre a Escola Latino Americana de Comunicação (XI CELACOM), Pelotas, Rio Grande do Sul, Brasil, p. 1-17.
- Patrick, A., McGurgan, A., 1993. *One proven methodology for designing robust online help systems*. In: 11th Annual International Conference on Systems Documentation (SIGDOC 1993), Kitchener, Ontario, Canada, p. 223-232.
- Pauling, L., Corey, R.B., Branson, H.R., 1951. *The structure of proteins*. Journal of the American Chemical Society, v. 61, n. 7, p. 1860-1867.
- Pavelin, K., Cham, J.A., Matos, P., Brooksbank, C., Cameron, G., Steinbeck, C., 2012. *Bioinformatics meets user-centred design: a perspective*. PLoS Computational Biology, v. 8, n. 7, p. e1002554.
- PDB - Protein Data Bank, 2015. *RCSB PDB - Content growth report*. Disponível em: <http://www.rcsb.org/pdb/statistics/contentGrowthChart.do?content=total&seqid=100>. Acessado em: 11/11/2013.
- Pearson, W.R., Lipman, D.J., 1988. *Improved tools for biological sequence comparison*. Proceedings of the National Academy of Sciences, v. 85, n. 8, p. 2444-2448.

- Perutz, M.F., Rossmann, M.G., Cullis, A.F., Muirhead, H., Will, G., North, A.C.T., 1960. *Structure of hemoglobin: a three-dimensional Fourier synthesis at 5.5-Å resolution, obtained by X-ray analysis*. Nature, v. 185, n. 4711, p. 416-422.
- Pollastri, G., McLysaght, A., 2005. *Porter: a new, accurate server for protein secondary structure prediction*. Bioinformatics, v. 21, n. 8, p. 1719-1720.
- Pontes, P.E., 2008. *Especificação de Requisitos para comunicabilidade em websites na Engenharia Semiótica*. Trabalho de Conclusão de Curso. Centro Universitário Ritter dos Reis (UNIRITTER), Porto Alegre, Rio Grande do Sul, Brasil, 108 p.
- Porollo, A.A., Adamczak, R., Meller, J., 2004. *POLYVIEW: a flexible visualization tool for structural and functional annotations of proteins*. Bioinformatics, v. 20, n. 15, p. 2460-2462.
- Prates, R.O., Barbosa, S.D.J., 2003. *Avaliação de interfaces de usuário - Conceitos e métodos*. Em: XXIII Congresso Nacional da Sociedade Brasileira de Computação (XXIII CSBC), XXII Jornadas de Atualização em Informática (XXII JAI), Campinas, São Paulo, Brasil, p. 1-49.
- Preece, J., Rogers, Y., Afiada, H., Benyon, D., Holland, S., Carey, T., 1994. *Human-Computer Interaction*. ACM SIGCHI Bulletin, v. 26, n. 4, p. 82-85.
- Preece, J., Rogers, Y., Sharp, H., 2005. *Design de interação: além da Interação Humano-Computador*. 2ª edição. Editora Bookman, Porto Alegre, Brasil, 548 p.
- Prosdocimi, F., 2003. *Bioinformática: manual do usuário*. Biotecnologia Ciência & Desenvolvimento, v. 29, p. 12-25.
- Rabi, I.I., Zacharias, J.R., Millman, S., Kusch, P., 1938. *A New Method of Measuring Nuclear Magnetic Moment*. Physical Review, v. 53, n. 4, p. 318-318.
- Reed, P., Holdaway, K., Isensee, S., Buie, E., Fox, J., Williams, J., Lund, A., 1999. *User interface guidelines and standards: progress, issues, and prospects*. Interacting with Computers, v. 12, n. 2, p. 119-142.
- Rex Hartson, H., 1998. *Human-computer interaction: interdisciplinary roots and trends*. Journal of Systems and Software, v. 43, n. 2, p. 103-118.
- Ribeiro, T., 2010. *Nanomedicina: uma abordagem para futuras construções de proteínas utilizando neurocomputação*. In: XVIII Seminário de Iniciação Científica da PUC-Rio, Rio de Janeiro, Rio de Janeiro, Brasil, p. 1-9.
- Ring, C.S., Cohen, F.E., 1993. *Modeling protein structures: construction and their applications*. The FASEB Journal, v. 7, n. 9, p. 783-790.
- Roche, D.B., Buenavista, M.T., Tetchner, S.J., McGuffin, L.J., 2011. *The IntFOLD server: an integrated web resource for protein fold recognition, 3D model quality assessment, intrinsic disorder prediction, domain prediction and ligand binding site prediction*. Nucleic Acids Research, v. 39, n. 2, p. W171-W176.
- Rohl, C.A., Strauss, C.E., Chivian, D., Baker, D., 2004. *Modeling structurally variable regions in homologous proteins with ROSETTA*. Proteins: Structure, Function, and Bioinformatics, v. 55, n. 3, p. 656-677.
- Ronacher, A., Brandl, G., Zapletal, A., Afshar, A., Edgemon, C., Grindstaff, C., Grebs, C., Neuhäuser, D., Xicluna, F., Brandl, G., Reitz, K., Sigler, M., Campell, M., Frazier, M., Tellingén, M. van, Duplain, R., Estienne, S., Sapin, S., Wirtel, S., Schranz, T., Xiaohong, Z., Burnett, E., 2010. *Flask (a Python microframework)*. Disponível em: <http://flask.pocoo.org>. Acessado em: 06/02/2014.

- Rost, B., Sander, C., 1993. *Prediction of protein secondary structure at better than 70% accuracy*. Journal of Molecular Biology, v. 232, n. 2, p. 584-599.
- Roy, A., Kucukural, A., Zhang, Y., 2010. *I-TASSER: a unified platform for automated protein structure and function prediction*. Nature Protocols, v. 5, n. 4, p. 725–738.
- Rutherford, P., Abell, W., Churcher, C., McKinnon, A., McCallum, J., 2010. *Usability of navigation tools for browsing genetic sequences*. In: 11th Australasian User Interface Conference (AUIC 2010), Brisbane, Queensland, Australia, p. 33-41.
- Sander, C., Schneider, R., 1991. *Database of homology-derived protein structures and the structural meaning of sequence alignment*. Proteins: Structure, Function, and Bioinformatics, v. 9, n. 1, p. 56-68.
- Scapin, D., Vanderdonckt, J., Farenc, C., Bastide, R., Bastien, C., Leulier, C., Mariage, C., Palanque, P., 2001. *Transferring knowledge of user interfaces guidelines to the web*. Vanderdonckt, J., Farenc, C. (Eds.). Tools for Working with Guidelines. Springer, London, England, p. 293–304.
- Science, 2005. *So much more to know*. Science, v. 309, n. 5731, p. 78-102.
- Sears, A., Lund, A.M., 1997. *Creating effective user interfaces*. IEEE Software, v. 14, n. 4, p. 21.
- Seringhaus, M., Gerstein, M., 2007. *Chemistry nobel rich in structure*. Science, v. 315, n. 5808, p. 40-41.
- Shaer, O., Kol, G., Strait, M., Fan, C., Grevet, C., Elfenbein, S., 2010. *G-nome surfer: a tabletop interface for collaborative exploration of genomic data*. In: ACM Conference on Human Factors in Computing Systems (CHI 2010), Atlanta, Georgia, USA, p. 1427-1436.
- Silveira, M.S., 2002. *Metacomunicação designer-usuário na Interação Humano-Computador: design e construção do sistema de ajuda*. Tese de Doutorado. Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), Rio de Janeiro, Rio de Janeiro, Brasil, 133 p.
- Silveira, N.J.F., 2005. *Bioinformática estrutural aplicada ao estudo de proteínas alvo do genoma do Mycobacterium tuberculosis*. Tese de Doutorado. Universidade Estadual Paulista (UNESP), São José do Rio Preto, São Paulo, Brasil, 117 p.
- Silveira, M.S., Leite, L.L., 2009. *Alternativas de ajuda on-line para ambientes de aprendizagem colaborativa*. In: XX Simpósio Brasileiro de Informática na Educação (XX SBIE), Florianópolis, Santa Catarina, Brasil, p. 1-10.
- Silveira, N.J.F., 2005. *Bioinformática estrutural aplicada ao estudo de proteínas alvo do genoma do Mycobacterium tuberculosis*. Tese de Doutorado. Universidade Estadual Paulista (UNESP), São José do Rio Preto, São Paulo, Brasil, 117 p.
- Simons, K.T., Bonneau, R., Ruczinski, I., Baker, D., 1999. *Ab initio protein structure prediction of CASP III targets using ROSETTA*. Proteins: Structure, Function, and Bioinformatics, v. 37, n. S3, p. 171-176.
- Simons, K.T., Kooperberg, C., Huang, E., Baker, D., 1997. *Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions*. Journal of Molecular Biology, v. 268, n. 1, p. 209-225.
- Simons, K.T., Strauss, C., Baker, D., 2001. *Prospects for ab initio protein structural genomics*. Journal of Molecular Biology, v. 306, n. 5, p. 1191-1199.

- Skinner, M.E., Uzilov, A.V., Stein, L.D., Mungall, C.J., Holmes, I.H., 2009. *JBrowse: a next-generation genome browser*. *Genome Research*, v. 19, n. 9, p. 1630-1638.
- Skolnick, J., Kihara, D., Zhang, Y., 2004. *Development and large-scale benchmark testing of the PROSPECTOR\_3 threading algorithm*. *Proteins: Structure, Function, and Bioinformatics*, v. 56, n. 3, p. 502-518.
- Söding, J., Biegert, A., Lupas, A.N., 2005. *The HHpred interactive server for protein homology detection and structure prediction*. *Nucleic Acids Research*, v. 33, n. 2, p. W244-W248.
- Srinivasan, R., Fleming, P.J., Rose, G.D., 2004. *Ab initio protein folding using LINUS*. *Methods in Enzymology*, v. 383, p. 48-66.
- Srinivasan, R., Rose, G.D., 2002. *Ab initio prediction of protein structure using LINUS*. *Proteins: Structure, Function, and Bioinformatics*, v. 47, n. 4, p. 489-495.
- Srinivasan, R., Rose, G.D., 1995. *LINUS: a hierarchic procedure to predict the fold of a protein*. *Proteins: Structure, Function, and Bioinformatics*, v. 22, n. 2, p. 81-99.
- Ssemugabi, S., 2009. *Usability evaluation of a web-based e-learning application: a study of two evaluation methods*. Doctoral Dissertation. University of South Africa (UNISA), Pretoria, Gauteng Province, South Africa, 336 p.
- Ssemugabi, S., Villiers, R., 2007. *A comparative study of two usability evaluation methods using a web-based e-learning application*. In: Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries (SAICSIT 2007), Port Alfred, Eastern Cape, South Africa, p. 132-142.
- Starovasnik, M.A., Braisted, A.C., Wells, J.A., 1997. *Structural mimicry of a native protein by a minimized binding domain*. *Proceedings of the National Academy of Sciences*, v. 94, n. 19, p. 10080-10085.
- Tavares, T.A., Cavalcanti, J.L., 2001. *A utilização do ambiente POIROT como ferramenta de apoio ao método de avaliação*. Em: IV Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais (IHC 2001), Florianópolis, Santa Catarina, Brasil, p. 1-13.
- Thunnissen, M.M., Taddei, N., Liguri, G., Ramponi, G., Nordlund, P., 1997. *Crystal structure of common type acylphosphatase from bovine testis*. *Structure*, v. 5, n. 1, p. 69-79.
- Torrieri, R., 2010. *Identificação e caracterização computacional de proteínas do tipo IUP no proteoma predito de Schistosoma mansoni*. Tese de Doutorado. Fundação Oswaldo Cruz (FIOCRUZ), Belo Horizonte, Minas Gerais, Brasil, 116 p.
- Valentin, F., Squizzato, S., Goujon, M., McWilliam, H., Paern, J., Lopez, R., 2010. *Fast and efficient searching of biological data resources - using EB-eye*. *Briefings in Bioinformatics*, v. 11, n. 4, p. 375-384.
- Veretnik, S., Fink, J.L., Bourne, P.E., 2008. *Computational biology resources lack persistence and usability*. *PLoS Computational Biology*, v. 4, n. 7, p. e1000136.
- Voet, D., Voet, J.G., 2013. *Bioquímica*. 4ª edição. Editora Artmed, Porto Alegre, Brasil, 1512 p.
- W3C, 2013. *About World Wide Web Consortium*. Disponível em: <http://www.w3.org/Consortium>. Acessado em: 26/06/2015.

- Wang, Z., Eickholt, J., Cheng, J., 2010. *MULTICOM: a multi-level combination approach to protein structure prediction and its assessments in CASP8*. *Bioinformatics*, v. 26, n. 7, p. 882-888.
- Willighagen, E.L., 2001. *Processing CML conventions in Java*. *Internet Journal of Chemistry*, v. 4, n. 4, p. 1099-8292.
- Winckler, M.A.A., Farenc, C., Palanque, P., Pimenta, M.S., 2001. *Avaliação da navegação de interfaces web a partir de modelos*. In: IV Workshop sobre Fatores Humanos em Sistemas Computacionais, IV Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais (IHC 2001), Florianópolis, Santa Catarina, Brasil, p. 48-60.
- Winckler, M., Pimenta, M.S., 2002. *Avaliação de usabilidade de sites web*. Disponível em: <https://www.irit.fr/~Marco.Winckler/2002-winckler-pimenta-ERI-2002-cap3.pdf>. Acessado em: 02/05/2015.
- World Wide Web Consortium, 1999. *Web Content Accessibility Guidelines 1.0*. Disponível em: <http://www.w3.org/TR/WAI-WEBCONTENT>. Acessado em: 20/11/2013.
- Wu, S., Skolnick, J., Zhang, Y., 2007. *Ab initio modeling of small proteins by iterative TASSER simulations*. *BMC Biology*, v. 5, n. 1, p. 17.
- Xu, Y., Xu, D., 2000. *Protein threading using PROSPECT: design and evaluation*. *Proteins: Structure, Function, and Bioinformatics*, v. 40, n. 3, p. 343-354.
- Yang, Y., Faraggi, E., Zhao, H., Zhou, Y., 2011. *Improving protein fold recognition and template-based modeling by employing probabilistic-based matching between predicted one-dimensional structural properties of the query and corresponding native properties of templates*. *Bioinformatics*, v. 27, n. 15, p. 2076-2082.
- Zhang, J., Wang, Q., Barz, B., He, Z., Kosztin, I., Shang, Y., Xu, D., 2010. *MUFOLD: a new solution for protein 3D structure prediction*. *Proteins: Structure, Function, and Bioinformatics*, v. 78, n. 5, p. 1137-1152.
- Zhang, Y., 2014. *Interplay of I-TASSER and QUARK for template-based and ab initio protein structure prediction in CASP10*. *Proteins: Structure, Function, and Bioinformatics*, v. 82, n. S2, p. 175-187.
- Zhang, Y., 2008. *I-TASSER server for protein 3D structure prediction*. *BMC Bioinformatics*, v. 9, n. 1, p. 1-8.
- Zhou, H., Skolnick, J., 2007. *Ab initio protein structure prediction using chunk-TASSER*. *Biophysical Journal*, v. 93, n. 5, p. 1510-1518.

# APENDICE A – MANUAL DE INSTALAÇÃO DO MÉTODO CREF EM COMPUTADOR LOCAL

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL  
FACULDADE DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

# CReF

*(Central Residue Fragment-based method)*

---

*Manual do usuário*  
*User's manual*

*Versão: 2.2*

Porto Alegre  
2013

## 1. Introdução

### a. Objetivo:

Este manual tem como finalidade orientar e informar o usuário quanto a uma correta instalação e operação do Método CReF (Central Residue Fragment-based Method) para predição aproximada de estruturas de proteínas.

As orientações que são apresentadas neste documento constituem-se em indicações básicas para o melhor entendimento de como se utilizar o Método CReF, constituindo-se mais um passo de um trabalho já reconhecido, e que se encontra em processo de aperfeiçoamento. Neste manual encontram-se as orientações para a instalação do método CReF em máquina local, antes do desenvolvimento de sua interface web, apresentada neste trabalho.

## 2. Orientações

Para um melhor entendimento deste manual, filtramos as informações com algumas diretrizes, separada conforme a legenda:

**Borda pontilhada:** Comandos para serem digitados no terminal do Linux.

**Borda em ziguezague:** Exemplo de comando a ser digitado no terminal do Linux.

**Palavra sublinhada:** Variável que é para ser trocada pelo usuário.

## 3. Criando os diretórios padrões do CReF:

### 4. Em modo terminal:

- Abra o terminal do Linux: **Ctrl+F1**

- Defina o diretório base do CReF:

```
mkdir /home/fulano/cref
```

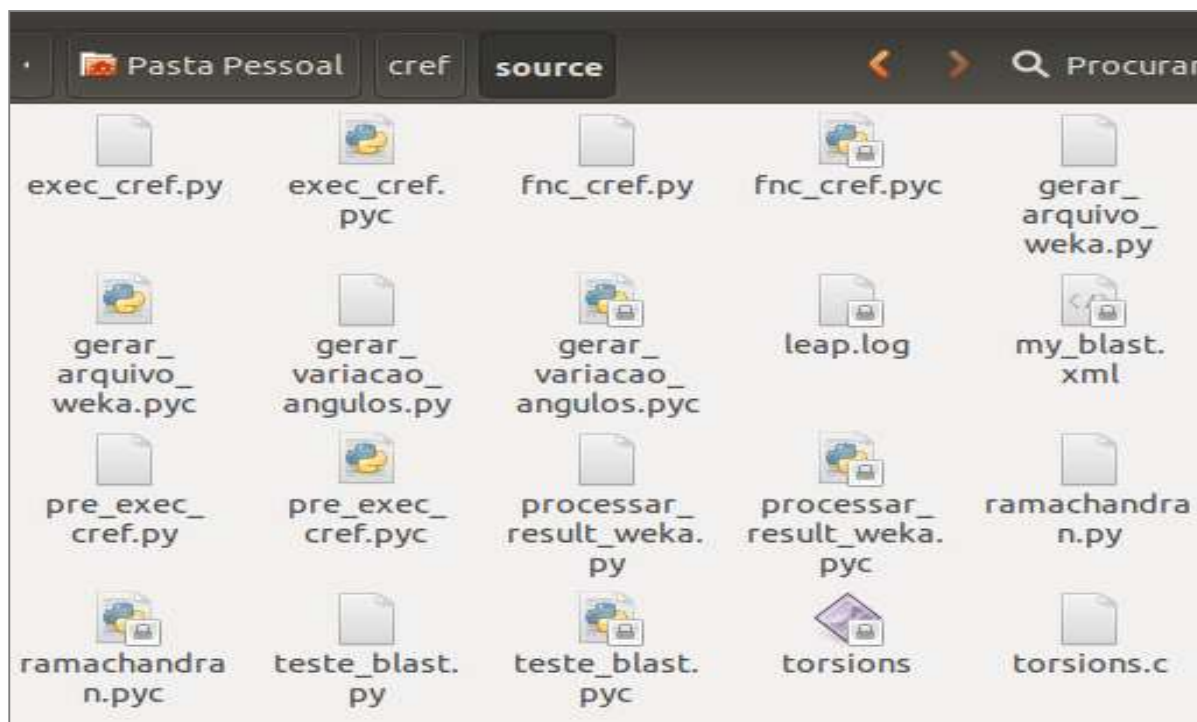
\* Aqui sugerimos colocar o nome da pasta em caixa baixa para facilitar a digitação dos comandos no terminal.



- Defina o diretório do CReF onde seu código fonte será armazenado.

```
mkdir /home/fulano/cref/source
```

- Cole o código fonte do CReF dentro da pasta source definida no passo anterior.



- Defina um diretório onde serão armazenados os arquivos pdbs que serão baixados pelo CReF após o processamento do BLASTp.

```
mkdir /home/fulano/cref/pdbs
```

- Para verificar se as pastas estão criadas corretamente no diretório corrente, dê um **ls -l** na pasta CReF.

```
cd /home/fulano/cref
```

```
ls -l
```

- Para fechar o terminal: **Ctrl+F7**

### a. Em modo gráfico:

- Entre na pasta home, e após na pasta pessoal (fulano), crie uma pasta **cref** e dentro dela crie as pastas **source** (para o código fonte do CReF) e a pasta **pdb**s. A pasta arquivos é gerada automaticamente na execução do método.

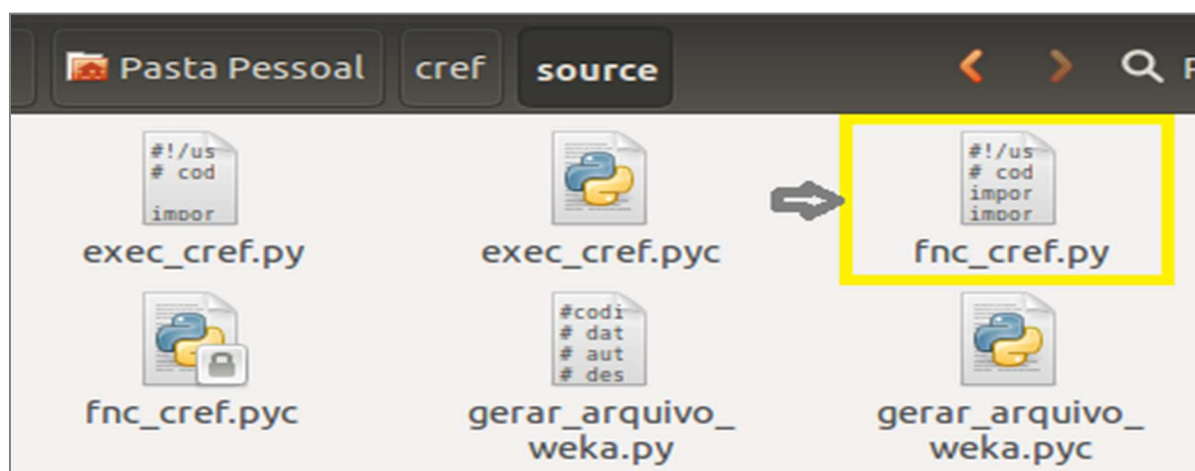


## 5. Configuração do CReF

### a. Alteração do código fonte:

No código fonte do CReF original algumas alterações são necessárias. Faça as modificações como a seguir:

- Abra o arquivo **fnc\_cref.py** no diretório Source:



- Altere o valor da variável **path\_def** para receber o endereço da pasta **arquivos**:

```
path_def = "/home/fulano/cref/arquivos/"
```

```
*fnc_cref.py x
#!/usr/bin/env python
# coding: utf-8
import os
import sys
import datetime
import urllib2
import platform
from Bio.SeqUtils import seq3

#DEFINA O DIRETORIO BASE
path_def = "/home/fulano/cref/arquivos/|
```

- Altere o valor da variável **path\_pdb**s para receber o caminho da pasta **pdb**s:

```
path_pdb = "/home/fulano/cref/pdb/"
```

```
#COLOQUE O DIRETORIO DOS ARQUIVOS PDBS EM UM LUGAR SEPARADO
path_pdb = "/home/fulano/cref/pdb/"
```

- Altere o valor da variável **email\_pred** colocando um e-mail que seja utilizado para receber o resultado da busca da predição da estrutura secundária. Foi criada uma rotina que acessa o e-mail e busca o resultado da predição de estrutura secundária gerada pelo Porter, a qual é enviada para este e-mail.

```
email_pred = emailfulano@gmail.com
```

```
num_grupos_weka = 6 # ou 4
email_pred = "emailfulano@gmail.com"
```

Ao final destes passos deve-se instalar os *softwares* externos ao CReF necessários para seu correto funcionamento.

## 6. Pré-instalação das ferramentas externas ao CReF:

Algumas das ferramentas externas ao CReF possuem dependência de certos programas e bibliotecas do Python. Para baixá-las execute os seguintes comandos:

3.1 Abra o terminal do Linux: **Ctrl+F1**

3.2 Execute os comandos como administrador: **sudo -i** e digite seu nome de usuário e senha.

3.3 Execute **apt-get update** para atualizar as listas de repositórios do Linux.

```
Ign http://security.ubuntu.com quantal-security/restricted Translation-pt_PT
Ign http://security.ubuntu.com quantal-security/restricted Translation-pt
Ign http://security.ubuntu.com quantal-security/restricted Translation-pt_BR
Ign http://security.ubuntu.com quantal-security/universe Translation-pt_PT
Ign http://security.ubuntu.com quantal-security/universe Translation-pt
Ign http://security.ubuntu.com quantal-security/universe Translation-pt_BR
A ler as listas de pacotes... Pronto
root@vanessa:~#
```

3.4 Instale as dependências através dos comandos a seguir:

```
apt-get install csh
apt-get install flex
apt-get install bison
apt-get install patch
apt-get install default-jre
apt-get install openjdk-6-jdk
apt-get install csh flex gfortran g++ xorg-dev zlibg-dev libbz2-
dev
```

#### i. NumPy:

O NumPy é um pacote básico da linguagem Python que permite trabalhar com arranjos, vetores e matrizes de N dimensões. Para fazer sua instalação basta dar o comando:

```
apt-get install python-numpy
```

#### ii. BioPython:

O projeto BioPython é um conjunto de ferramentas disponíveis para bioinformática desenvolvido em linguagem Python, produzido por uma associação internacional que desenvolvem *softwares* para biologia molecular. Para maiores informações visite o site oficial que é <http://www.biopython.org>. Como o BioPython é uma fonte de módulos, scripts e links, no método CReF utilizamos para acessar o BLASTp.

```
apt-get install python-biopython
apt-get build-dep python-biopython
```

```

root@vanessa:~# apt-get install python-biopython
A ler as listas de pacotes... Pronto
A construir árvore de dependências
A ler a informação de estado... Pronto
Os seguintes pacotes extra serão instalados:
  python-biopython-doc python-support
Pacotes sugeridos:
  python-tk wise muscle clustalw mafft emboss blast2
Serão instalados os seguintes NOVOS pacotes:
  python-biopython python-biopython-doc python-support
0 pacotes actualizados, 3 pacotes novos instalados, 0 a remover e 418 não actualizados.
É necessário obter 9.727 kB de arquivos.
Após esta operação, serão utilizados 16,4 MB adicionais de espaço em disco.
Deseja continuar [Y/n]? y

```

- Aceite a instalação dando o comando Y no terminal e aguarde o fim da instalação.

## 7. Configurar variáveis de ambiente no Linux

- No terminal digite o seguinte comando:

```
gedit /etc/profile
```

- Ao final deste arquivo devem ser colocados os comandos utilizados para gerar as variáveis de ambiente que foram definidas. Segue abaixo:

```

export AMBERHOME=/home/<caminho>/amber<numero-da-versao>

export PATH="$AMBERHOME/bin:<caminho>/amber<versao>/bin:$PATH"

export JMOL_HOME="<caminho>/Jmol/"

export CLASSPATH="<caminho>/WEKAINSTALL/weka.jar:$CLASSPATH"

```

## 8. Instalação de ferramentas externas

### a. Torsions:

- Faça o download do software em: <http://www.bioinf.org.uk/software/swreg.html>.

Obs.: é necessário preencher um formulário.

Overview Antibodies Servers Mutations Software Group Sitemap

## Software Registration

Index

Installation Before downloading my software, I ask you to fill in this simple registration form. This costs you nothing and simply enables me to keep a record of who is using my software (useful for grant applications as well as for my ego) to keep you informed of bug-fixes and updates.

Download

Name:

Address:

E-Mail:

Software packages you wish to download:

- ProFit V3.1** Protein least squares fitting
- ProFit V3.1 Linux Binary Distribution** As above but contains an Linux executable instead of source code.
- ProFit V3.1 Windows Binary** As above but contains a Win32 executable instead of source code.
- QTree** Graphics rendering
- MINT** Interface to Andrej Sali's MODELLER
- CDoc** Generate HTML documentation from C source
- QLite** A simple queueing system for a farm of machines
- NW** An implementation of the Needleman and Wunsch sequence alignment algorithm
- Torsions** A program to calculate backbone torsion angles in a PDB file
- torsions.exe** A Windows binary of the Torsions program
- Cluster** Cluster analysis

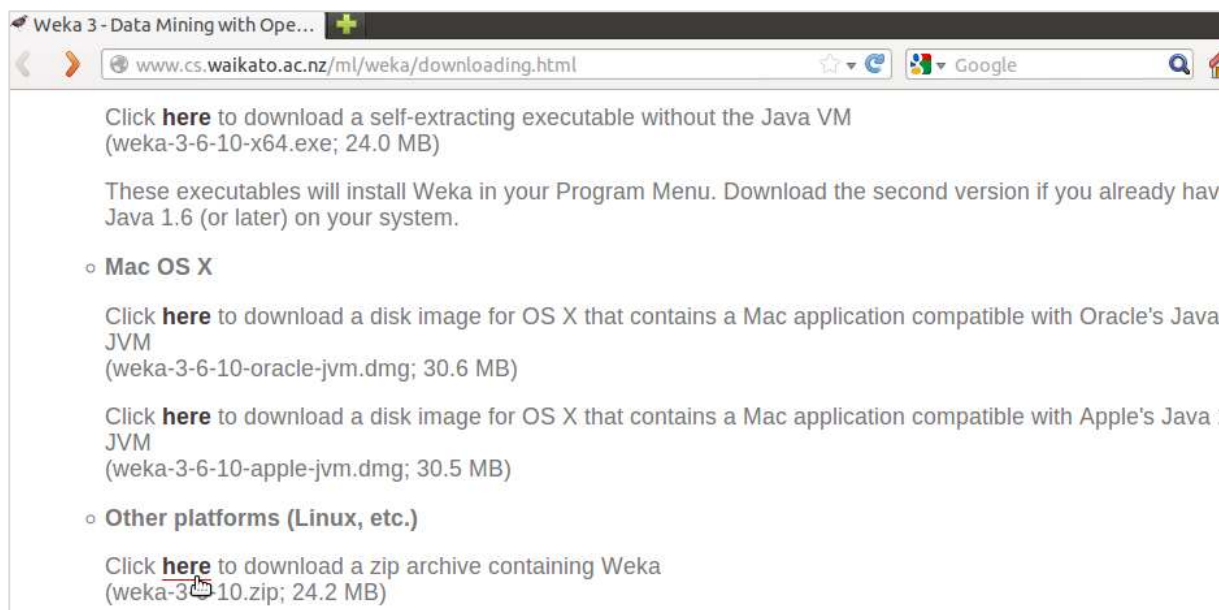
- Salve o programa no mesmo diretório onde estão armazenados os fontes do CReF.
- Abra o terminal, vá para a pasta onde está localizado o fonte do CReF e execute o seguinte comando:

```
root@vanessa:/home/vanessa/cref/source# cc -o torsions torsions.c -lm
root@vanessa:/home/vanessa/cref/source# _
```

\*É necessário utilizar o parâmetro `-lm`, pois, do contrário, o arquivo não irá compilar.

### b. Weka:

- Faça o download da versão mais atual do Weka através do link: <http://www.cs.waikato.ac.nz/ml/weka/downloading.html>



- Extraia o fonte do Weka para dentro do diretório raiz do CReF. Importante: Abra o terminal em modo gráfico (painel inicial – pesquisar – terminal – console, sem estar no modo de administrador) e execute os seguintes comandos:

```
cd /home/fulano/CReF
mv weka<numero-da-versao> WEKAINSTALL
```

**Ex: mv weka3-6-10 WEKAINSTALL**

\* Obs: Segundo diretrizes na página do WEKA a pasta deve ser renomeada para WEKAINSTALL.

- Prossiga com a instalação:

```
cd WEKAINSTALL
java -Xmx1000M -jar weka.jar
```

Após executar esses 2 comandos a aplicação Weka irá ser executada automaticamente aparecendo então uma tela com sua interface gráfica. Feche a janela para continuar:



```
java -jar weka.jar weka.classifiers.j48.J48
```

```
export CLASSPATH="<caminho>/WEKAINSTALL/weka.jar:$CLASSPATH"
```

Ex: `export CLASSPATH=/home/fulano/cref/WEKAINSTALL/weka.jar:$CLASSPATH`

```
root@vanessa:/home/vanessa/cref/WEKAINSTALL# export CLASSPATH=home/vanessa/cref/WEKAINSTALL/weka.jar:$CLASSPATH
root@vanessa:/home/vanessa/cref/WEKAINSTALL#
```

Para ter certeza de que o weka foi instalado corretamente execute o seguinte comando:

```
java weka.clusterers.EM -t
```

```
vanessa@vanessa: ~/cref/WEKAINSTALL
clusters evaluation.

Options specific to weka.clusterers.EM:

-N <num>
  number of clusters. If omitted or -1 specified, then
  cross validation is used to select the number of clusters.
-I <num>
  max iterations.
  (default 100)
-V
  verbose.
-M <num>
  minimum allowable standard deviation for normal density
  computation
  (default 1e-6)
-o
  Display model in old format (good when there are many clusters)
-S <num>
  Random number seed.
  (default 100)

vanessa@vanessa:~/cref/WEKAINSTALL$
```

### c. Amber:

- a. Faça o download do programa Ambertools zipado através do link <http://ambermd.org/AmberTools-get.html>

\* Obs.: é necessário preencher um formulário:

**Note:** Installation instructions (which have changed slightly from previous versions) are in Section 1.2 of the *AmberTools13 Reference Manual*. More detailed instructions for many machines may be found at *Jason Swails' blog page*.

Name:	<input type="text" value="fulano"/>
Institution:	<input type="text" value="instituição"/>
City:	<input type="text" value="cidade"/>
State or Province:	<input type="text" value="Other"/>
Country:	<input type="text" value="Brazil"/>



- Extraia o conteúdo do arquivo de preferência para o diretório pai do CReF.
- No terminal digite os seguintes comandos:

```
export AMBERHOME=/home/<caminho>/amber<numero-da-versao>
```

**EX: export AMBERHOME=/home/fulano/CReF/amber12**

```
cd $AMBERHOME
```

```
./update_amber --check-updates
```

```
./update_amber --update
```

```
vanessa@vanessa: ~/amber12
vanessa@vanessa:~$ export AMBERHOME=/home/vanessa/amber12
vanessa@vanessa:~$ cd $AMBERHOME
vanessa@vanessa:~/amber12$ ./update_amber --check-updates
Checking for available patches online. This may take a few seconds...

Available AmberTools 13 patches:
update.1, update.2, update.3, update.4, update.5, update.6, update.7, update.8,
update.9, update.10,
update.11, update.12, update.13, update.14, update.15, update.16
vanessa@vanessa:~/amber12$
```

- Executar o comando acima até receber aviso de que o amber está atualizado.
- Prosseguir com o comando:

```
./configure gnu
```

```
make install
```

```
make test
```

```
export PATH="$AMBERHOME/bin:<caminho>/amber<versao>/bin:$PATH"
```

**Ex: export PATH="\$AMBERHOME/bin:/home/fulano/CReF/amber12/bin:\$PATH"**

\* Obs. Os dois procedimentos acima levam um certo tempo para finalizar.

#### d. Jmol:

Jmol: um visualizador de Java open-source para estruturas químicas em 3D (fonte página oficial: <http://www.jmol.org/>). Para fazer sua instalação:

- Baixe o Jmol usando o link: <http://jmol.sourceforge.net/download/>



The screenshot shows the Jmol website's download page. At the top left is the Jmol logo with flags for UK, France, Spain, and a globe. At the top right are navigation links: Home, Demonstration pages, Websites, Documentation, Wiki, History, FAQs, Browser check, Download, and Project pages. The main heading is "Download Jmol". Below it are links for "Downloading Jmol", "Requirements", "Installing and Running Jmol", and "Subversion Access". A sub-heading "Downloading Jmol" is followed by the text: "The current official release is version 13.0. It can be downloaded from SourceForge at this direct [download link](#) (.zip file)."

- Extrair o arquivo compactado (de preferência para o diretório base do CReF) e renomear a pasta extraída para “Jmol”.

- No terminal executar os seguintes comandos:

```
cd <caminho>/Jmol
```

**Ex: cd /home/fulano/cref/Jmol**

```
chmod u+x jmol.sh
```

```
ln -s <caminho>/Jmol/jmol.sh /usr/local/bin/jmol.sh
```

**Ex: ln -s /home/fulano/cref/Jmol/jmol.sh /usr/local/bin/jmol.sh**

- Para ter certeza de que o Jmol foi instalado corretamente execute o seguinte comando:

```
java -jar Jmol.jar
```

```
export JMOL_HOME="<caminho>/Jmol/"
```

**Ex: export JMOL\_HOME="/home/fulano/cref/Jmol/"**

```
source $HOME/.profile
```

## 9. Utilizando o método CReF

### a. Arquivos:

Para cada execução do CReF é necessário à criação de 2 arquivos:

#### a. Resultado da predição da estrutura secundária:

**<codigo\_pdb>.pred**

Neste arquivo deve estar contido em uma linha o resultado da predição da estrutura secundária.

Nome do arquivo: 2EVQ.pred  
 Conteúdo do arquivo: CCCEECCHH

**b. Lista de códigos PDB:**

**<codigo\_pdb>.exc**

Neste arquivo deve ser colocada a lista de códigos pdbs, sendo um por linha, que possuem alguma relação de homologia com a proteína a ser predita, caso se saiba. No mínimo o código pdb da própria proteína deve ser informado no arquivo

Ex: 2EVQ.exc  
 Conteúdo do arquivo: 2EVQ

**b. Execução do CReF**

O CReF pode ser executado de 3 maneiras diferentes, sendo 2 delas através do terminal:

**c. Na primeira maneira, o método será executado desde o começo:**

```
cd /home/fulano/cref/source

python pre_exec_cref.py codigo_pdb sequencia

diretorio_base/codigo_pdb.exc
```

Ex: `python pre_exec_cref.py 2EVQ KTWNPATCKWTE`  
`/home/fulano/CReF/arquivos/codigo_pdb.exc`

```
root@vanessa:/home/vanessa/cref/source# python pre_exec_cref.py 2EVQ KTWNPATGKWT
E /home/vanessa/cref/arquivos/2EVQ.exc
Argumento1: 2EVQ
Argumento2: KTWNPATGKWT
```

**d. Na segunda maneira, pode-se reexecutar o CReF a partir de uma etapa que se tenha parado:**

```
cd /home/fulano/CReF/source
```

```
python exec_cref.py <num_etapa>
```

**Ex: python exec\_cref.py 5**

- e. **Na terceira maneira, deve ser utilizada a interface web que foi desenvolvida para se comunicar diretamente com o CReF.**

Nela devem ser informados:

- Código PDB da proteína.
- Sequência.
- E-mail para onde os resultados devem ser enviados.
- Lista de códigos pdbs homólogos ou arquivo com a extensão (.txt ou .exc) contendo os códigos PDBs, sendo 1 por linha.

## APENDICE B – ORIENTAÇÕES: AVALIAÇÃO HEURÍSTICA DE SERVIDORES DE PREDIÇÃO DE ESTRUTURAS DE PROTEÍNAS

### Descrição da avaliação:

Avaliação Heurística (Nielsen, 1995, 1994a, 1994b, 2001; Nielsen e Molich, 1990) é um método de avaliação de usabilidade que busca encontrar problemas de usabilidade numa interface de usuário, como parte de um processo de *design* interativo. Esse tipo de avaliação envolve um pequeno conjunto de inspetores, que irão examinar a interface, avaliando sua conformidade com os princípios de usabilidade reconhecidos (os "heurísticos").

Este estudo pretende realizar avaliação de usabilidade nos principais dos servidores de predição de estruturas tridimensionais de proteínas participantes do CASP (*Critical Assessment of Techniques for Protein Structure Prediction*) (Moult et al., 2014, 2005, 1997, 1995) objetivando o levantamento de requisitos de software para desenvolvimento de um novo servidor de predição (wCReF) para o método CReF (*Central Residue Fragment-based Method*). Os servidores avaliados são apresentados na Tabela 1.

**Tabela 1.** Os servidores de predição avaliados nesta pesquisa

Servidor de predição	Disponível em:	Referência
QUARK	<a href="http://zhanglab.ccmb.med.umich.edu/QUARK/">http://zhanglab.ccmb.med.umich.edu/QUARK/</a>	(Zhang, 2014)
I-TASSER	<a href="http://zhanglab.ccmb.med.umich.edu/I-TASSER/">http://zhanglab.ccmb.med.umich.edu/I-TASSER/</a>	(Roy et al., 2010; Zhang, 2008)
ROBETTA	<a href="http://rosetta.bakerlab.org/">http://rosetta.bakerlab.org/</a>	(Kim et al., 2004)

Cada inspetor fará a avaliação dos três servidores, buscando achar na sua interface problemas que violem qualquer uma das 10 Heurísticas de Nielsen.

Durante a sessão de avaliação o avaliador deverá percorrer a interface várias vezes, inspecionar os vários elementos de diálogo e compará-los com a lista de Heurísticas. A recomendação geral é que a interface seja inspecionada pelo menos duas vezes. É sugerido que o primeiro passo seja a análise geral do sistema, em que o avaliador irá percorrer livremente a interface buscando conhecê-la. Em seguida, é realizada uma segunda passagem, que deve permitir que o avaliador se concentre em elementos de interface específicos, para saber como eles se encaixam no todo (Nielsen, 2001).

Só depois de todas as avaliações concluídas os avaliadores são autorizados a comunicar e ter seus resultados agregados. Esse processo é importante a fim de assegurar avaliações independentes e imparciais de cada avaliador (Nielsen, 2001).

No final todos os problemas encontrados na Avaliação Heurística serão reunidos em um único documento. Esse relatório além de sugerir melhorias a serem realizadas nos servidores de predição

de proteínas, buscando melhorar sua usabilidade, tem como objetivo principal o levantamento de requisitos de software para a construção da interface do wCReF.

Os procedimentos de como se realizar a avaliação neste trabalho e o formulário para realização da Avaliação Heurística encontram-se a seguir.

## Condução da Avaliação Heurística:

### Orientações:

1. Primeiramente, leia e analise as dez Heurísticas (Tabela 2).

**Tabela 2.** Conjunto de Heurísticas de Nielsen (1994a,1995)

<b>1.</b>	<b>Visibilidade do status do sistema:</b> O sistema deve sempre manter os usuários informados sobre o que está acontecendo, através de feedback apropriado, em um tempo razoável.
<b>2.</b>	<b>Compatibilidade entre sistema e mundo real:</b> O sistema deve utilizar a linguagem do usuário, com palavras, frases e conceitos familiares para ele, ao invés de termos específicos de sistemas. Seguir convenções do mundo real, fazendo com que a informação apareça em uma ordem lógica e natural.
<b>3.</b>	<b>Controle e liberdade para o usuário:</b> Os usuários frequentemente escolhem as funções do sistema por engano e então necessitam de "uma saída de emergência", claramente definida para sair do estado não desejado, sem ter que percorrer um longo diálogo, ou seja, é necessário suporte a <i>undo</i> e <i>redo</i> ( <i>desfazer e refazer</i> ).
<b>4.</b>	<b>Consistência e padrões:</b> Referem-se ao fato de que os usuários não deveriam ter acesso a diferentes situações, palavras ou ações representando a mesma coisa. A interface deve ter convenções não ambíguas, seguindo as convenções da plataforma.
<b>5.</b>	<b>Prevenção de erros:</b> Ainda melhor do que boas mensagens de erro é um projeto cuidadoso que impede que um problema ocorra em primeiro lugar. O sistema deve eliminar as condições passíveis de erros ou verificá-los, e apresentar aos usuários uma opção de confirmação, antes de se comprometer com a ação. Os erros são as principais fontes de frustração, ineficiência e ineficácia durante a utilização do sistema.
<b>6.</b>	<b>Reconhecimento em lugar de lembrança:</b> Minimizar a carga de memória do usuário, fazendo objetos, ações e opções visíveis. O usuário não deve ter de lembrar informações de uma parte do diálogo para outra. Instruções para o uso do sistema devem estar visíveis ou facilmente acessíveis.
<b>7.</b>	<b>Flexibilidade e eficiência de uso:</b> O sistema deve ser adequado tanto para usuários inexperientes quanto para usuários experientes. Deve permitir aos usuários personalizar ações frequentes, como por exemplo, a utilização de aceleradores - invisíveis pelo usuário novato - podem frequentemente acelerar a interação para o usuário experiente.

**8. Design estético e minimalista:**

Os diálogos não devem conter informações irrelevantes ou raramente necessárias. Cada unidade extra de informação em um diálogo compete com unidades relevantes e diminui sua visibilidade relativa.

**9. Auxiliar os usuários a reconhecer, diagnosticar e recuperar erros:**

Mensagens de erro devem ser expressas em linguagem natural/clara (sem códigos), indicando precisamente o erro/problema e sugerir uma solução.

**10. Ajuda e documentação:**

Mesmo que seja melhor que o sistema possa ser usado sem documentação, pode ser necessário fornecer ajuda e documentação. Tais informações devem ser fáceis de encontrar, centradas na tarefa do usuário, listar passos concretos a serem realizados e não serem muito grande. A ajuda deve estar facilmente acessível e on-line.

2. Cada Heurística definida anteriormente deve ser respeitada. Desta forma você deverá percorrer a interface buscando encontrar erros de usabilidade que violem esses princípios.

O primeiro momento é reservado para que se conheça a interface, analisando a mesma livremente. Após este primeiro contato, deve-se analisar a interface em busca de problemas de usabilidade que não estejam de acordo com as Heurísticas, sendo que algumas tarefas são sugeridas para esta avaliação (Tabela 3). Estas tarefas geralmente são as atividades que os usuários fazem com este tipo de sistema.

**Tabela 3.** Lista de tarefas para ser realizadas na Avaliação Heurística.

Lista de tarefas	
1.	Cadastrar-se como usuário (se disponível).
2.	Efetuar <i>Login</i> com as credenciais criadas.
3.	Alterar senha, ou algum dado do cadastro já efetuado.
4.	Enviar a sequência de uma proteína alvo para predição.
5.	Modificar os parâmetros para predição (se disponível) e realizar um novo envio.
6.	Verificar a fila de espera da predição.
7.	Analisar se a interface possui documentação ou tutorial para usuário.
8.	Conferir se o servidor retornou os dados esperados.
9.	Percorrer o Menu do servidor analisando as opções possíveis e as páginas correspondentes.
10.	Verificar se o mesmo possui referências bibliográficas.
11.	Analisar se o usuário consegue obter ajuda, através de mensagens, fóruns, documentação etc.
12.	Ver se os resultados são de fácil acesso e compreensão.

3. Quando um problema qualquer for detectado, classifique-o em uma das dez Heurísticas de Nielsen, anotando o problema na tabela correspondente no formulário de avaliação (cada Heurística possui seu respectivo formulário). Para cada Heurística poderão ser utilizados quantos formulários forem necessários para identificação dos problemas de usabilidade.

4. Cada problema encontrado e especificado deverá conter um **grau de severidade** (de 0 até 4) conforme a classificação (Tabela 3) definida por Nielsen (1994).

**Tabela 3.** Escala de severidade atribuída na Avaliação Heurística. Fonte: Nielsen (Nielsen, 1994a)

Severidade	Tipo	Significado
0	Sem importância	Não é considerado, totalmente, um problema de usabilidade.
1	Cosmético	Problema apenas estético: não necessita ser consertado a menos que haja tempo extra, disponível no projeto.
2	Simple	Problema menor de usabilidade: o conserto desse problema deverá ter baixa prioridade.
3	Grave	Problema maior de usabilidade: é importante consertá-lo, para isso deverá ser dada alta prioridade.
4	Catastrófico	Catástrofe de usabilidade: é obrigatório consertá-lo, antes do produto ser divulgado.

5. Esse processo deverá se repetir até que não se encontrem mais problemas de usabilidade.

6. Normalmente o tempo dedicado para cada avaliação é de uma a duas horas.

7. A avaliação poderá ser feita de duas formas, através do formulário a seguir, encaminhado através de e-mail, ou através do preenchimento do formulário on-line.

8. Ao final, o condutor da Avaliação Heurística que deverá reunir os dados coletados, em um documento único, que será utilizado para a montagem da interface do wCReF.

## REFERÊNCIAS

- Moult, J., Fidelis, K., Kryshatovych, A., Schwede, T., Tramontano, A., 2014. *Critical assessment of methods of protein structure prediction (CASP): round x*. Proteins: Structure, Function, and Bioinformatics, v. 82, n. S2, p. 1-6.
- Moult, J., Hubbard, T., Bryant, S.H., Fidelis, K., Pedersen, J.T., 1997. *Critical assessment of methods of protein structure prediction (CASP): round II*. Proteins, v. 23, n. 1, p. 2-6.
- Moult, J., Kryshatovych, A., Venclovas, C., Fidelis, K., 2005. *Progress over the first decade of CASP experiments*. Proteins, v. 61, n. 7, p. 225-236.
- Moult, J., Pedersen, J.T., Judson, R., Fidelis, K., 1995. *A large-scale experiment to assess protein structure prediction methods*. Proteins: Structure, Function, and Bioinformatics, v. 23, n. 3, p. 2-4.
- Nielsen, J., 1994a. *Usability engineering. 1st edition*. Elsevier, San Francisco, USA, 362 p.



- Nielsen, J., 1994b. *Usability inspection methods*. In: ACM Conference on Human Factors in Computer Systems (CHI 1994), Boston, Massachusetts, USA, p. 413-414.
- Nielsen, J., 1995. *10 heuristics for user interface design: article by Jakob Nielsen*. Disponível em: <http://www.nngroup.com/articles/ten-usability-heuristics>. Acessado em: 09/08/2014.
- Nielsen, J., 2001. *How to conduct a heuristic evaluation: article by Jakob Nielsen*. Disponível em: <https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation>. Acessado em: 02/03/2015.
- Nielsen, J., Molich, R., 1990. *Heuristic evaluation of user interfaces*. In: ACM Conference on Human Factors in Computing Systems (CHI 1990), Seattle, Washington, USA, p. 249-256.
- Rohl, C.A., Strauss, C.E., Chivian, D., Baker, D., 2004. *Modeling structurally variable regions in homologous proteins with ROSETTA*. Proteins: Structure, Function, and Bioinformatics, v. 55, n. 3, p. 656-677.
- Roy, A., Kucukural, A., Zhang, Y., 2010. *I-TASSER: a unified platform for automated protein structure and function prediction*. Nature Protocols, v. 5, n. 4, p. 725–738.
- Simons, K.T., Bonneau, R., Ruczinski, I., Baker, D., 1999. *Ab initio protein structure prediction of CASP III targets using ROSETTA*. Proteins: Structure, Function, and Bioinformatics, v. 37, n. S3, p. 171-176.
- Zhang, Y., 2008. *I-TASSER server for protein 3D structure prediction*. BMC Bioinformatics, v. 9, n. 1, p. 1-8.
- Zhang, Y., 2014. *Interplay of I-TASSER and QUARK for template-based and ab initio protein structure prediction in CASP10*. Proteins: Structure, Function, and Bioinformatics, v. 82, n. S2, p. 175-187.

## APENDICE C - FORMULÁRIO DE AVALIAÇÃO HEURÍSTICA

Inspetor: \_\_\_\_\_

Instituição: \_\_\_\_\_

Inspeção: \_\_\_\_\_ Servidor: \_\_\_\_\_

Data da avaliação: \_\_\_ / \_\_\_ / \_\_\_ Hora inicial: \_\_\_\_\_ Hora final: \_\_\_\_\_

### Formulário 1 - Heurísticas de Nielsen - Sessão de avaliação

1.1	Visibilidade do <i>status</i> do sistema*	
<b>Definição:</b>	O sistema deve sempre manter os usuários informados sobre o que está acontecendo, através de feedback apropriado, em um tempo razoável.	<b>Grau de severidade</b>
<b>Identificação:</b> (Ex.: nº 1, 2, 3...)	<b>Problema:</b>	<input type="checkbox"/> 0 - Sem importância <input type="checkbox"/> 1 - Cosmético <input type="checkbox"/> 2 - Simples <input type="checkbox"/> 3 - Grave <input type="checkbox"/> 4 - Catastrófico
	<b>Problema:</b>	<input type="checkbox"/> 0 - Sem importância <input type="checkbox"/> 1 - Cosmético <input type="checkbox"/> 2 - Simples <input type="checkbox"/> 3 - Grave <input type="checkbox"/> 4 - Catastrófico
	<b>Problema:</b>	<input type="checkbox"/> 0 - Sem importância <input type="checkbox"/> 1 - Cosmético <input type="checkbox"/> 2 - Simples <input type="checkbox"/> 3 - Grave <input type="checkbox"/> 4 - Catastrófico
	<b>Problema:</b>	<input type="checkbox"/> 0 - Sem importância <input type="checkbox"/> 1 - Cosmético <input type="checkbox"/> 2 - Simples <input type="checkbox"/> 3 - Grave <input type="checkbox"/> 4 - Catastrófico
	<b>Problema:</b>	<input type="checkbox"/> 0 - Sem importância <input type="checkbox"/> 1 - Cosmético <input type="checkbox"/> 2 - Simples <input type="checkbox"/> 3 - Grave <input type="checkbox"/> 4 - Catastrófico
	<b>Problema:</b>	<input type="checkbox"/> 0 - Sem importância <input type="checkbox"/> 1 - Cosmético <input type="checkbox"/> 2 - Simples <input type="checkbox"/> 3 - Grave <input type="checkbox"/> 4 - Catastrófico

\* Este quadro se repete para todas as Heurísticas de Nielsen.

## APENDICE D- TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO PARA ESPECIALISTAS

Prezado (a) participante:

Sou estudante do curso de Pós-Graduação em Ciência da Computação da Pontifícia Universidade Católica do Rio Grande do Sul. Estou realizando uma pesquisa sob supervisão do professor Osmar Norberto de Souza, cujo objetivo é o desenvolvimento de uma interface Web para o método CReF (*Central Residue Fragment-based Method*). Dessa forma este estudo pretende realizar avaliação de usabilidade nos principais dos servidores de predição de estruturas de proteínas participantes do CASP (*Critical Assessment of Techniques for Protein Structure Prediction*) objetivando o levantamento de requisitos de software para o desenvolvimento da interface Web desse servidor.

Sua participação envolve a realização de avaliação de usabilidade, através da Avaliação Heurística de três servidores de predição. O procedimento leva em torno de uma a duas horas para cada atividade. A definição de como se realizar a avaliação, os formulários e toda a explicação necessária para que possa realizar a tarefa satisfatoriamente será repassada pelo condutor da avaliação.

A participação nesse estudo é voluntária e se você decidir não participar ou quiser desistir de continuar em qualquer momento, tem absoluta liberdade de fazê-lo.

Na publicação dos resultados desta pesquisa, sua identidade será mantida no mais rigoroso sigilo. Serão omitidas todas as informações que permitam identificá-lo(a).

Mesmo não tendo benefícios diretos em participar, indiretamente você estará contribuindo para a compreensão do fenômeno estudado e para a produção de conhecimento científico.

Quaisquer dúvidas relativas à pesquisa poderão ser esclarecidas pela estudante pesquisadora Vanessa Stangherlin Machado, do Laboratório de Bioinformática, Modelagem e Simulação de Biosistemas (LABIO), fone 51 33203611 ext.8602.

Atenciosamente

---

Vanessa Stangherlin Machado  
Matrícula: 1329006221

---

Local e data

---

Osmar Norberto de Souza

**Consinto em participar deste estudo e declaro ter recebido uma cópia deste termo de consentimento.**

---

Nome e assinatura do participante

---

Local e data

## APENDICE E - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO PARA OS USUÁRIOS

Prezado (a) participante:

Sou estudante do curso de Pós-Graduação em Ciência da Computação da Pontifícia Universidade Católica do Rio Grande do Sul. Estou realizando uma pesquisa sob supervisão do professor Osmar Norberto de Souza, cujo objetivo é o desenvolvimento de uma interface Web para o método CReF (*Central Residue Fragment-based Method*). A interface do wCReF foi desenvolvida utilizando requisitos obtidos por avaliações de usabilidade nos principais servidores de predição de estruturas de proteínas participantes do CASP (*Critical Assessment of Techniques for Protein Structure Prediction*). Dessa forma queremos agora avaliar a usabilidade da primeira versão dessa interface.

Sua participação envolve a realização de avaliação de usabilidade, através do uso de um questionário. A definição de como se realizar a avaliação, o questionário e toda a explicação necessária para que possa realizar a tarefa satisfatoriamente será repassada pelo condutor da avaliação.

A participação nesse estudo é voluntária e se você decidir não participar ou quiser desistir de continuar em qualquer momento, tem absoluta liberdade de fazê-lo.

Na publicação dos resultados desta pesquisa, sua identidade será mantida no mais rigoroso sigilo. Serão omitidas todas as informações que permitam identificá-lo (a).

Mesmo não tendo benefícios diretos em participar, indiretamente você estará contribuindo para a compreensão do fenômeno estudado e para a produção de conhecimento científico.

Quaisquer dúvidas relativas à pesquisa poderão ser esclarecidas pela estudante Vanessa Stangherlin Machado, do Laboratório de Bioinformática, Modelagem e Simulação de Biosistemas (LABIO), fone 51 33203611 ext.8602.

Atenciosamente

---

Vanessa Stangherlin Machado

Matrícula: 1329006221

---

Local e data

---

Osmar Norberto de Souza

**Consinto em participar deste estudo:**

---

Nome

---

Assinatura do Participante

---

## APENDICE F – DOCUMENTO DE REQUISITOS DE SOFTWARE



LABIO-PUCRS

*wCReF: Uma Interface Web para o Método CReF de Predição da Estrutura 3D Aproximada de Proteínas*

*wCReF: A Web Interface for CReF Method for Prediction of Protein Structure Approximate 3D*

---

# *Documento de Requisitos de Software*

Versão 1.0

### Histórico de Alterações

Data	Versão	Descrição	Autor
05/11/2014	1.0	Versão inicial do documento	Vanessa Machado
05/12/2014	1.0	Especificação dos Requisitos	Vanessa Machado
09/12/2014	1.0	Formatação do Documento	Vanessa Machado
15/12/2014	1.0	Revisão do documento	Osmar Norberto de Souza
16/12/2014	1.0	Definição da Versão Final	Vanessa Machado

---

*“Qualidade de software começa na especificação.”*  
- Rafael Helm

## 1 INTRODUÇÃO

### 1.1 Propósito

Este trabalho especifica os requisitos de software que serão utilizados para a construção da interface Web para o método CReF (*Central Residue Fragment-based Method*), denominada *wCReF*, fornecendo aos desenvolvedores as informações necessárias para sua implementação, assim como para a realização dos testes e homologação do sistema.

O documento especifica todos os requisitos funcionais e não funcionais do sistema e foi preparado levando-se em conta as funcionalidades levantadas em outros servidores de predição de estruturas de proteínas, participantes do CASP (*Critical Assessment of Techniques for Protein Structure Prediction*).

### 1.2 Âmbito

O trabalho foi conduzido pela Aluna Vanessa Stangherlin Machado, da *Pontifícia Universidade Católica do Rio Grande do Sul* (PUCRS), como parte da sua Dissertação de Mestrado, do programa de Pós-Graduação em Ciência da Computação (PPGCC) sobre a orientação do Professor Dr. Osmar Norberto de Souza.

No decorrer do escopo serão mostrados os principais problemas que motivam a realização desse trabalho, os requisitos principais, a equipe inicial e cronograma completo para o projeto.

#### 1.2.1 Escopo do produto

Servidores de predição de estruturas de proteínas são ferramentas cruciais para pesquisas em bioinformática. O objetivo geral deste trabalho é definir quais os requisitos de software a serem utilizados para o desenvolvimento de uma interface Web para o método o CReF (*Central Residue Fragment-based Method*). Este trabalho faz parte da automatização do método CReF, dispondo o mesmo para comunidade científica através da web, baseado em preceitos de usabilidade investigados na fase de avaliação\*.

## 1.2.2 Identificação do produto

### *Nomes, componentes e missão do produto*

Nome do produto	wCReF 1.0
Componentes principais	wCReF (Componente único)
Missão do produto	Automatizar o método CReF através de sua interface Web que ficará disponível para comunidade científica. O sistema irá funcionar como uma ferramenta para predição de estruturas de proteínas baseado no método CReF.

### *Limites do produto*

Número	
1	O wCReF será um sistema a ser utilizado somente através da Web necessitando de um navegador browser para acesso.
2	O usuário precisa estar conectado a uma rede mundial de computadores (internet).
3	O sistema será desenvolvido na língua nativa do Brasil – Português (versão 1.0)
4	Para utilizar o sistema será necessário um cadastro prévio pelo usuário.

### *1.2.2.1 Benefícios esperados do produto*

Número		Valor para o cliente
1	Predição da estrutura 3D Aproximada através de um novo método.	Essencial
2	Disponibilização do método CReF em rede mundial.	Importante
3	Sistema automatizado.	Importante
4	Servidor de estruturas de proteínas aproximada, desenvolvido no Brasil.	Desejável
6	Sistema que pode ser utilizado como ferramenta educativa.	Desejável

## 1.2.3 Problema Identificado

A predição da estrutura terciária de proteína é um dos problemas mais importantes ainda não solucionados pela ciência. O desafio é entender a relação entre a sequência de aminoácidos de uma proteína e sua estrutura tridimensional 3D, que está relacionada à função destas macromoléculas. Este problema pode ser verificado comparando-se as 16,8 milhões de sequências depositadas no banco de dados *GenBank* (Benson et al., 2012) as aproximadas 94 mil estruturas resolvidas no *Protein Data Bank* (PDB) (Berman et al., 2000), dados estes referentes a novembro de 2013. Para diminuir esta lacuna é que surge a bioinformática estrutural, no sentido de tratar esses dados e transformá-los em informação útil, tal como a caracterização da estrutura e função das proteínas.



Dentre os métodos relacionados à predição de estruturas de proteínas está o método CReF (Central Residue Fragment-based Method) que é um algoritmo proposto por Dorn & Norberto de Souza (2008) para predição da estrutura 3D aproximada de uma proteína ou polipeptídeo. Este método já demonstrou excelentes resultados, confirmados pela pesquisa de Dall’Agno (2012).

O desenvolvimento da interface Web do wCReF envolveu um levantamento de aplicações de bioinformática atuais, realizando uma análise heurística sobre os servidores de predição de estruturas de proteínas participante do CASP. Este estudo avaliou estas interfaces em busca de problemas de usabilidade. Com base nisso, um conjunto de requisitos de software foram definidos, que servirão como especificações para o desenvolvimento desta interface, assim como para o desenvolvimento de novas ferramentas na área da bioinformática.

#### 1.2.4 Objetivo

O objetivo deste documento é descrever os requisitos de software para o wCReF.

### 1.3 Definições, acrónimos e abreviaturas;

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

#### 1.3.1 Identificação dos requisitos

Por convenção, a referência a requisitos é feita através do nome da subseção onde eles estão descritos, seguidos do identificador do requisito, de acordo com a especificação a seguir:

[nome da subseção/identificador do requisito]

Número		Valor para o cliente
<i>Feature</i>	Característica principal do sistema	<b>FEAT</b>
<i>Scenario</i>	Característica secundária de uma Feature	<b>SCE</b>
Requisitos Funcionais	Correspondem à listagem de todas as coisas que o sistema deve fazer	<b>RF</b>
Requisitos Não-Funcionais	São restrições que se coloca sobre como o sistema deve realizar seus requisitos funcionais	<b>NF</b>

Por exemplo, o requisito funcional [RF01/Cadastro] deve estar descrito em uma subseção Feature Administrar Servidor [FEAT-01/Administrar Servidor], que pode estar no bloco de uma SCE - Configurar Método “[SCE-001/ Configurar Método]”, identificado pelo número [RF01]. Já o

requisito não-funcional [NF01/Desempenho] deve estar descrito na seção de requisitos não-funcionais de Desempenho, em um bloco identificado por [NF01].

Os requisitos devem ser identificados com um identificador único. A numeração inicia com o identificador [RF01] ou [NF01] e prossegue sendo incrementada à medida que forem surgindo novos requisitos.

### 1.3.2 Prioridades dos requisitos

Para estabelecer a prioridade dos requisitos, foram adotadas as denominações “essencial”, “importante” e “desejável”.

**Essencial** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.

**Importante** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.

**Desejável** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

### 1.3.3 Definições e siglas

Número	Sigla	Definição
1	CREF	Central Residue Fragment-based Method

## 1.4 Referências

Número	Sigla	Definição
1	Documentação	Gonçalves, A., Martins, F., Carreira, P., 1998. <i>IEEE Std 830 Prática Recomendada Para Especificações de Exigências de Software 9 Standard Internacional</i> . Disponível em: <a href="http://www.urisan.tc.br/~pbtencourt/engsoft/IEEE830/">http://www.urisan.tc.br/~pbtencourt/engsoft/IEEE830/</a> . Acessado em: 25/10/2014.
2	Livro	Pressman, R.S., 2009. <i>Engenharia de software</i> . 7ª edição. McGraw Hill, Rio de Janeiro, Brasil, 720 p.
3	Livro	Sommerville, I., Melnikoff, S.S.S., Arakaki, R., Andrade, B.E., 2003. <i>Engenharia de software</i> . 6ª edição. Addison Wesley, São Paulo, Brasil, 552 p.

4	Padrão	IEEE, 1998. <i>IEEE Std. 830 - 1998: IEEE Recommended Practice for Software Requirements Specifications</i> . IEEE Software Standards, New York, USA, 39 p.
5	Documentação	Ronacher, A., Brandl, G., Zapletal, A., Afshar, A., Edgemon, C., Grindstaff, C., Grebs, C., Neuhäuser, D., Xicluna, F., Brandl, G., Reitz, K., Sigler, M., Campell, M., Frazier, M., Tellingen, M. van, Duplain, R., Estienne, S., Sapin, S., Wirtel, S., Schranz, T., Xiaohong, Z., Burnett, E., 2010. Flask (a Python microframework). Disponível em: <a href="http://flask.pocoo.org">http://flask.pocoo.org</a> . Acessado em: 06/02/2014.

## 1.5 Organização

A organização deste documento está de acordo com a norma IEE 830, que define as Especificações de Exigências de Software (EES) (Gonçalves et al., 1998), definindo assim este documento:

### **1. Introdução** (Secção 1 do documento de EES)

A introdução do documento de EES deve providenciar uma visão geral sobre o documento inteiro. Deve por isso conter as seguintes subsecções:

- **Propósito;** (Secção 1.1 do documento de EES)
- **Âmbito;** (Secção 1.2 do documento de EES)
- **Definições, acrónimos e abreviaturas;** (Secção 1.3 do documento de EES)
- **Referências;** (Secção 1.4 do documento de EES)
- **Organização;** (Secção 1.5 do documento de EES)

Além da secção introdutória, as secções seguintes estão organizadas como descrito abaixo.

### **2. Descrição geral** (Secção 2 do documento de EES)

Esta secção do documento descrevem os fatores gerais que afetam o produto e as suas exigências. Esta secção não enumera exigências específicas. Ao invés, fornece um contexto para essas exigências, e facilita a sua compreensão). Esta secção consiste habitualmente em 4 subsecções:

- **Perspectiva do produto;** (Secção 2.1 do documento de EES)
- **Funções do produto;** (Secção 2.2 do documento de EES)
- **Características do utilizador;** (Secção 2.3 do documento de EES)
- **Restrições;** (Secção 2.4 do documento de EES)

### **3. Requisitos do Software**

Descreve os requisitos funcionais e não funcionais do sistema.

### **4. Barreiras de Usabilidade**

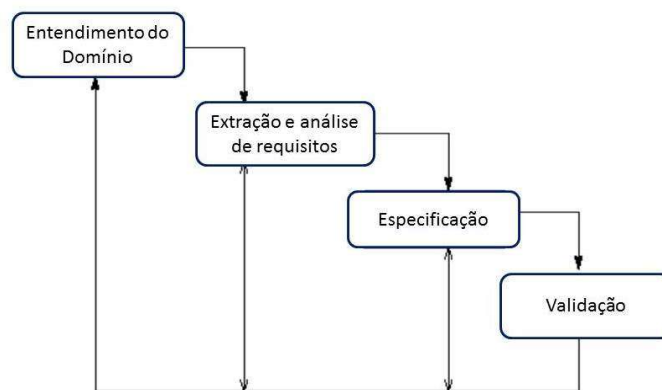
### **5. Sugestões**

## 2 DESCRIÇÃO GERAL

### 2.1 Perspectiva do produto

Esse documento tem a finalidade de apresentar aos requisitos de software para wCReF, o escopo será construído através da descrição dos requisitos, descrevendo suas funcionalidades, facilitando o trabalho dos desenvolvedores, para atender os requisitos para os clientes. Este documento também se propõe a dar uma melhor descrição do software, para transparecer ao cliente todas as funcionalidades que irão compor o sistema.

A Especificação de Requisitos de Software (ERS) concentra-se na coleta e na organização de todos os requisitos que envolvem o projeto. Segundo Pressman (2003) e Sommerville (2009) existem os seguintes passos para o processo de extração de requisitos de software:



O propósito deste documento é apresentar a descrição dos serviços e funções que um sistema a ser desenvolvido deve prover, bem como as suas restrições de operação e propriedades gerais, a fim de ilustrar uma descrição detalhada do sistema do servidor, e que sirva de auxílio durante as etapas de análise, projeto e testes. O documento especifica os requisitos funcionais e não funcionais do sistema e foi preparado levando-se em conta as funcionalidades levantadas através de avaliações heurísticas em servidores de predição de estruturas de proteínas.

Dessa forma foram realizados para este documento:

- Entendimento do domínio – Realizado através de avaliações heurísticas;
- Extração e análise de requisitos: compilação dos resultados.
- Especificação: Definição dos requisitos de software
- Validação: Utilização deste documento para criação de um protótipo.

## 2.1.1 Criação do Documento de Requisitos

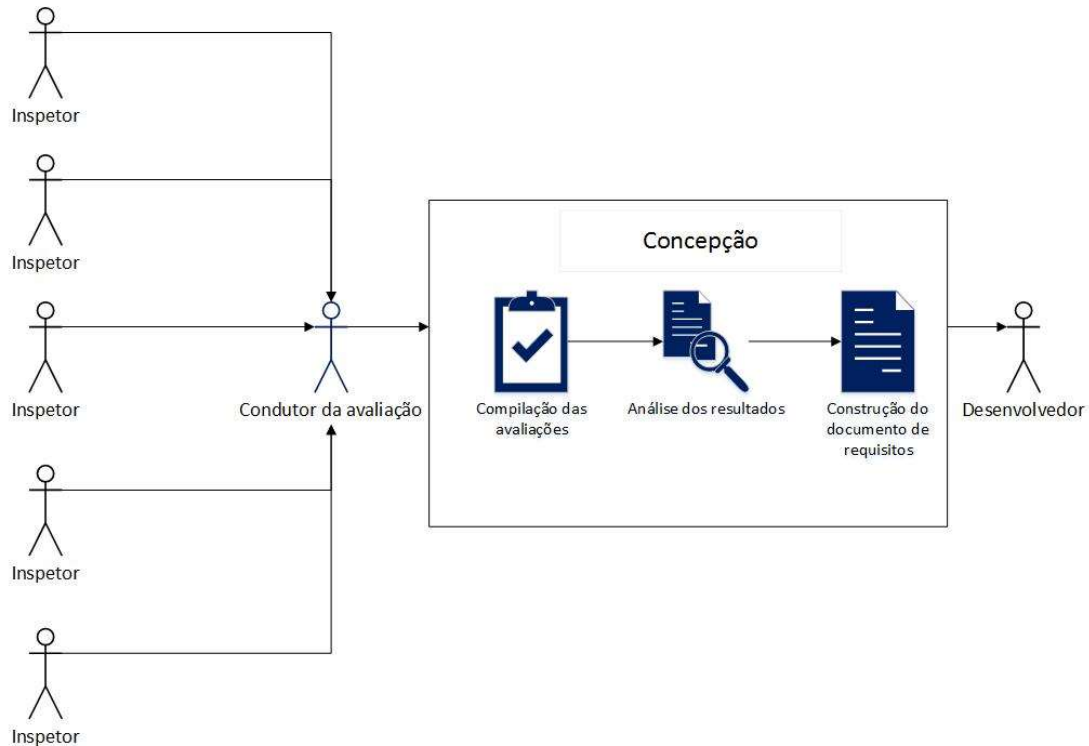
### 2.1.1.1 Metodologia

Foi usado o método de inspeção de usabilidade de Avaliação Heurística para que pudessem ser obtidos os requisitos do sistema. A Avaliação Heurística (Nielsen, 1995, 1994; Nielsen e Molich, 1990) é um método de avaliação de usabilidade que busca encontrar problemas de usabilidade numa interface de usuário, como parte de um processo de *design* interativo. Esse tipo de avaliação envolve um pequeno conjunto de inspetores, que irão examinar a interface, avaliando sua conformidade com os princípios de usabilidade reconhecidos (os "heurísticos"). A partir dos resultados dessas avaliações, um único documento com os erros de usabilidade encontrado nos servidores de predição avaliados foi compilado, e usado como referência para criação do documento de requisitos.

### 2.1.1.2 Consulta com especialista

Avaliador	Qualificação	Área de estudo
A	Doutor em Ciências Físicas e Biomoleculares - Especialista	Bioinformática
B	Doutor em Informática na Educação - Especialista	IHC e Informática na Educação
C	Doutoranda em Ciência da Computação - Especialista	Bioinformática - Desenvolvimento de Software
D - parcial	Doutoranda em <i>Design</i> e Tecnologias - Especialista	IHC e Tecnologias Assistivas

2.1.1.3 Diagrama de criação do documento



2.1.2 Sistemas similares

No cenário mundial podemos usar como referência de sistemas similares os servidores de predição observados nas últimas edições do CASP (*Critical Assessment of Techniques for Protein Structure Prediction*) (Moult et al., 2014, 2005, 1997, 1995). O CASP é uma competição “às cegas” onde a comunidade científica tenta prever a estrutura 3D de proteínas cujas suas estruturas são conhecidas, mas não ainda disponíveis ao público. São disponibilizadas sequências alvo para grupos de pesquisa de todo mundo. Cada grupo participante aplica algum método, algoritmo ou utiliza-se de um servidor de predição objetivando prever as estruturas da proteína alvo, que ao final são reveladas, e assim as predições são avaliadas pelo CASP.

Servidor de predição	Disponível em:	Referência
QUARK	<a href="http://zhanglab.ccmb.med.umich.edu/QUARK/">http://zhanglab.ccmb.med.umich.edu/QUARK/</a>	(Zhang, 2014)
I-TASSER	<a href="http://zhanglab.ccmb.med.umich.edu/I-TASSER/">http://zhanglab.ccmb.med.umich.edu/I-TASSER/</a>	(Roy et al., 2010a; Zhang, 2008)
ROBETTA	<a href="http://robetta.bakerlab.org/">http://robetta.bakerlab.org/</a>	(Kim et al., 2004)
ROSETTA	<a href="https://www.rosettacommons.org/">https://www.rosettacommons.org/</a>	(Rohl et al., 2004; Simons et al., 1999)

PMS: A Panoptic Motif Search Tool	<a href="http://www.motifsearch.com/index.php?page=motifproteinseq">http://www.motifsearch.com/index.php?page=motifproteinseq</a>	(Dinh e Rajasekaran, 2013)
MUFold	<a href="http://mufold.org/">http://mufold.org/</a>	(Zhang et al., 2010, 2010)
RaptorX server	<a href="http://raptorx.uchicago.edu/">http://raptorx.uchicago.edu/</a>	(Källberg et al., 2014)
SPARKSX	<a href="http://sparks-lab.org/yueyang/sparks-x">http://sparks-lab.org/yueyang/sparks-x</a>	(Yang et al., 2011)
MULTICOM	<a href="http://casp.rnet.missouri.edu/multicom_3d.html">http://casp.rnet.missouri.edu/multicom_3d.html</a>	(J. Li et al., 2014; Wang et al., 2010)
IntFOLD	<a href="http://www.reading.ac.uk/bioinf/IntFOLD/IntFOLD2_form.html">http://www.reading.ac.uk/bioinf/IntFOLD/IntFOLD2_form.html</a>	(Roche et al., 2011)
Chunk-TASSER	<a href="http://cssb.biology.gatech.edu/skolnick/webservice/chunk-TASSER/index.html">http://cssb.biology.gatech.edu/skolnick/webservice/chunk-TASSER/index.html</a>	(Zhou e Skolnick, 2007)

## 2.2 Funções do produto

### 2.2.1 Interfaces de usuário

Número	Nome	Definição
1	<i>Tela de Usuários - Cadastro</i>	Interface on-line para cadastro no sistema, inclusão, consulta, alteração e exclusão de usuários.
2	<i>Tela de Submissão</i>	Interface on-line para submissão das proteínas alvo a serem preditas.
3	<i>Tela de acompanhamento das submissões</i>	Interface on-line para acompanhamento dos pedidos submetidos.
4	<i>Tela de apresentação dos resultados.</i>	Interface onde o usuário acompanha os resultados finais de sua predição.
5	<i>Ajuda e Documentação</i>	Interface on-line apresentando a documentação referente ao sistema, e ajuda ao usuário.
6	<i>Tela dos softwares utilizados</i>	Interface apresentando os <i>softwares</i> utilizados no método e suas respectivas referências.
7	<i>Tela de submissão de dúvidas, questões e sugestões</i>	Interface para o usuário enviar questões referentes ao sistema.
8	<i>Tela de login.</i>	Tela para o usuário se logar no sistema e prosseguir com a submissão.
9	<i>Tela sobre o grupo de pesquisa</i>	Interface on-line apresentando o grupo de pesquisa do Método CReF.
10	<i>Tela de Referências</i>	Tela com as referências sobre o método CReF.

### 2.2.2 Interfaces de hardware

Número	Nome	Definição
1	<i>Computador servidor</i>	

### 2.2.3 Interface de software

Número	Nome	Definição
1	Sistema Operacional	O método CReF é executado em sistema operacional LINUX. Sua interface Web porém poderá ser acessada de qualquer sistema operacional que possua um navegador padrão e conexão com internet.
2	<i>Softwares</i> externos	Torsions (Dr. Andrew C.R. Martin's Group, 2014), Weka (Frank et al., 2005, 2004), Amber (Kollman et al., 1995)
3	Linguagem de programação	A aplicação do wCReF será desenvolvida em linguagem HTML que pode ser interpretada por navegadores da internet. Para aplicar o visual ( <i>design</i> da interface) nas páginas HTML, irá se utilizar a linguagem de folha de estilo ( <i>Cascading Style Sheets</i> ou simplesmente CSS) deixando a página com um estilo padrão e agradável ao usuário. O Método CReF foi desenvolvido em linguagem Python e sua plataforma Web será HTML/CSS.
4	Publicação	A hospedagem foi realizada através do servidor Web <i>Pythonanywhere</i> ( <a href="https://www.pythonanywhere.com">https://www.pythonanywhere.com</a> ) que é baseado na linguagem Python.
5	Bancos de dados	Tudo que for submetido ao wCReF é armazenado em um banco de dados MySQL (Oracle Corporation, 1995) para sua consulta, processamento e devolução dos resultados para os usuários do método.

### 2.2.1 Interfaces de comunicação

Número	Nome	Definição
1	Framework Flask	Interface para comunicação da interface Web com a linguagem python. O Flask é uma ferramenta para fazer a ligação/comunicação entre o método e o aplicativo. Para isso foi usado o microframework Flask (Armin Ronacher et al., 2010).

## 2.3 Funções do produto

Número	Característica	Definição
1	Predição da estrutura 3D de proteínas	Predição da estrutura 3D de proteínas aproximada de proteínas, a partir da execução do método CReF.



2	Predição da estrutura 3D de proteínas	O usuário através de um formulário Web irá informar a sequência da estrutura alvo a ser predita, e o sistema retornará com a estrutura 3D aproximada da proteína no formato PDB (extensão .pdb)
---	---------------------------------------	---

## 2.4 Características do utilizador

### 2.4.1 Identificação dos envolvidos com o sistema

O wCReF é documento de requisitos de software destinado a descrever requisitos de software que sirvam para o desenvolvimento da interface Web para o método CReF e está destinado a vislumbrar profissionais e usuários de diferentes contextos. Na área científica (computacional e biológica) está relacionada com a disponibilização de um novo servidor de predição de estruturas de proteínas, apoiando a pesquisa científica na área da bioinformática estrutural, apoiando também o desenvolvimento deste tipo de software, servindo de aporte de orientação para programadores, designers, desenvolvedores e demais profissionais de Bioinformática. Para educadores, por prover subsídios necessários ao desenvolvimento de ferramentas que possam servir de apoio às suas práticas pedagógicas. E, para estudantes de bioinformática uma vez que será uma ferramenta de predição de estruturas implementada no Brasil. Além disso além de ser um guia a ser considerado para usabilidade de software durante a fase de desenvolvimento, poderá servir de um método de avaliação para *softwares* existentes e também pós desenvolvimento que permita a identificação de pontos falhos a serem adaptados para melhorar a interatividade de um produto final.

### 2.4.2 Pessoas envolvidas no projeto

Número	Característica	Definição
1	Vanessa Stangherlin Machado	Este projeto faz parte de sua dissertação de Mestrado em Ciência da Computação.
2	Osmar Norberto de Souza	Orientador

### 2.4.3 Perfil do usuário

Existem basicamente 4 tipos de usuários que podem ter acesso às funcionalidades do sistema:

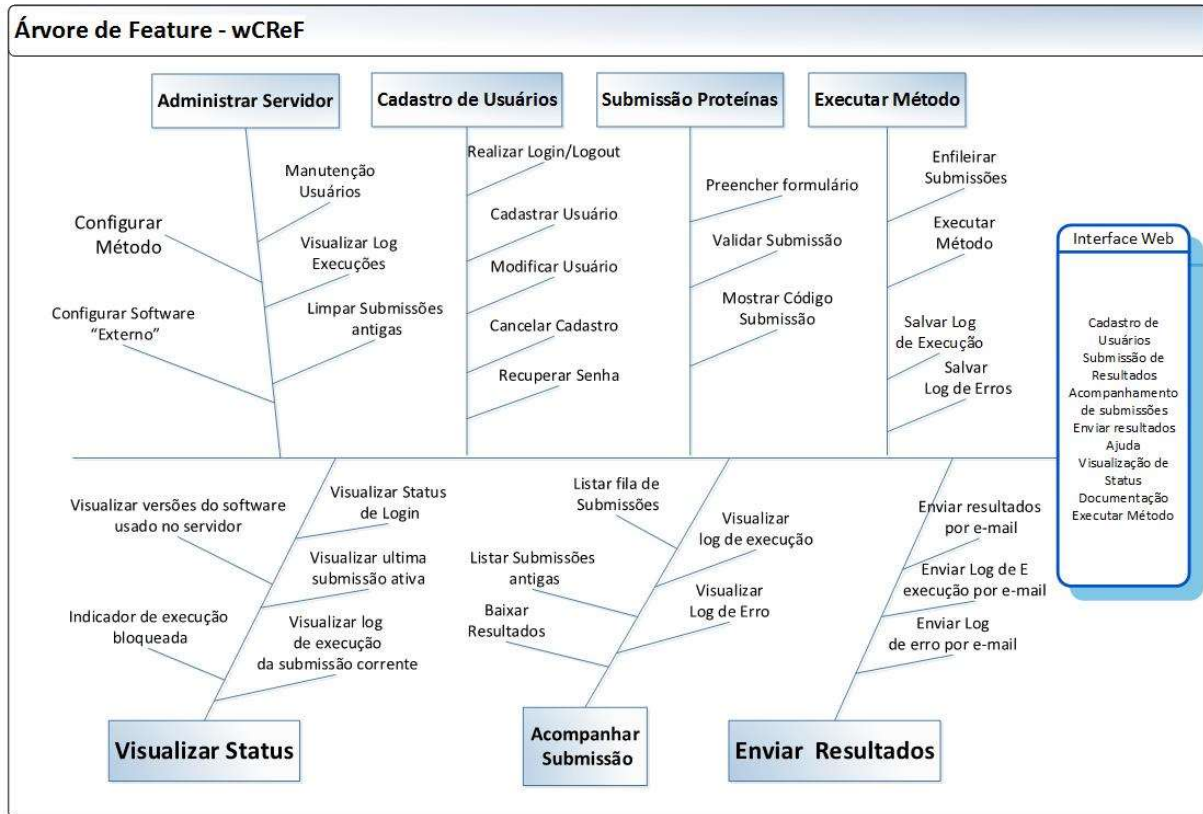
Número	Característica	Definição
1	Cientistas, Bioinformatas, Pesquisadores	Usuário Final

2	Pesquisadores	Autores do Documento de requisitos e também pesquisadores da comunidade científica
3	Profissionais da Computação	Programadores, Projetistas, Designers que utilizem o Documento de Requisitos para desenvolver <i>Softwares</i> para Bioinformática
4	Profissionais da Educação e da Saúde	Biólogos, Pedagogos, Professores, e demais profissionais que trabalhem diretamente com educação e usuários finais

## 2.5 Restrições

Número	Restrição	Descrição
1	Ambiente	Este trabalho não define sistema operacional específico, pois se trata de uma interface Web, podendo ser acessada de qualquer SO que possua um navegador Web e conexão com internet.
1	Ambiente	Este trabalho não define a linguagem de programação a ser utilizada, pois isso depende da escolha da equipe de TI e também do contexto do software que será desenvolvido a partir desses requisitos.
2	Ambiente	Alguns pontos devem ser considerados no desenvolvimento, quanto ao não uso de determinadas funções e características, e estão descritas nos requisitos específicos na próxima sessão.
3	Usabilidade	Os produtos desenvolvidos a partir desse documento devem levar em conta usabilidade
4	Legal	Os produtos desenvolvidos a partir desse documento deverão estar de acordo com as leis e regulamentos vigentes na época de seu desenvolvimento.
5	Segurança	Os produtos desenvolvidos devem estar de acordo com questões de segurança e tolerância a erros, levando-se em consideração os requisitos descritos abaixo.

### 3 REQUISITOS DO SOFTWARE



#### 3.1 Requisitos funcionais (casos de uso)

Os requisitos funcionais do sistema são apresentados a seguir:

##### 3.1.1 Administrar Servidor

Identificador	Nome	Descrição	Casos de uso relacionados	Prioridade
[FEA-001]	[FEA-001/Administrar Servidor]	O sistema terá um Administrador que será responsável por manter os recursos funcionando, realizar as atualizações do software, corrigir os erros e acompanhar o funcionamento do sistema. Toda a configuração deve ser realizada por um administrador do sistema.	[SCE-001] [SCE-002] [SCE-003] [SCE-004] [SCE-005] [SCE-006]	Essencial
[SCE-001]	[SCE-001/Configurar Método]	O Administrador do Sistema deverá configurar os softwares, programas e pacotes pertencentes ao método CReF.	[RF-01] [RF-02] [RF-03] [RF-04]	Essencial

[RF-01]	[RF-01/Configurar Software “Externo”]	O Administrador deve baixar e configurar todos os <i>softwares</i> que são utilizados no método CReF.		<b>Essencial</b>
[RF-02]	[RF-02/Atualizar pacotes]	Os pacotes Linux e Python devem ser atualizados conforme as versões mais recentes.		<b>Importante</b>
[RF-03]	[RF-03/Atualizar <i>Software</i> “Externo”]	O Administrador deve atualizar as versões mais recentes dos <i>softwares</i> que são utilizados no CReF.		<b>Desejável</b>
[RF-04]	[RF-04/ “path” caminho do software]	Os caminhos para diretórios e pastas, e o “path” devem ser configurados pelo administrador.		<b>Essencial</b>
[SCE-002]	[SCE-002/Execuções Paralelas]	<b><i>O Sistema deve permitir a execução de mais de uma predição simultaneamente.</i></b>		<b>Importante</b>
[SCE-003]	[SCE-003/ Nome de arquivos – resultados]	<b><i>O Sistema deve nomear cada job submetido.</i></b>	[RF-05] [RF-06]	<b>Importante</b>
[RF-05]	[RF-05/Nome das execuções]	Os arquivos receberão um nome obrigatório, com a data da submissão.		<b>Importante</b>
[RF-06]	[RF-06/Diretório dos resultados]	Os resultados ficarão armazenados em uma pasta com o nome da submissão e a data.		<b>Desejável</b>
[SCE-004]	[SCE-004/ Manutenção Usuários]	<b><i>O Administrador será responsável pela manutenção dos usuários.</i></b>	[RF-07][RF-08]	<b>Importante</b>
[RF-07]	[RF-07/Exclusão de Usuário]	O Administrador poderá excluir um usuário que esteja indevido no sistema.		<b>Importante</b>
[RF-08]	[RF-08/Mensagem de Exclusão-Usuário]	Um usuário cadastrado e retirado do banco de dados deverá ser informado do procedimento.		<b>Desejável</b>
[SCE-005]	[SCE-005/ Visualizar Log Execuções]	<b><i>O administrador deve ser capaz de visualizar os logs de execução.</i></b>	[RF-09]	<b>Importante</b>
[RF-09]	[RF-09/ Visualizar Log Execuções]	<b><i>O Sistema deve listar todos os logs de execução, mostrando os seguintes campos: Nome, Data, e Situação.</i></b>		<b>Desejável</b>
[SCE -006]	[SCE -006/ Limpar Submissões antigas]	<b><i>As submissões antigas já entregues para os usuários deverão ser excluídas do sistema.</i></b>		<b>Importante</b>
[RF-10]	[RF-10/Exclusão de submissão antigas]	As predições deverão permanecer no sistema durante duas semanas depois excluídas do banco de dados.	[RF-10]	<b>Desejável</b>

### 3.1.2 Cadastro de Usuários

Identificador	Nome	Descrição	Casos de uso relacionados	Prioridade
[FEA-002]	<b>[FEA-002/Cadastro de Usuários]</b>	<b>O sistema manterá o cadastro dos usuários em um banco de dados.</b>	<i>[SCE-007] [SCE-008] [SCE-009] [SCE-010]</i>	<i>Importante</i>
<i>[SCE-007]</i>	<i>[SCE-007/ Realizar Login/Logout]</i>	<i>O usuário poderá se logar no sistema se estiver cadastrado.</i>	<i>[RF-11] [RF-12] [RF-13]</i>	<i>Importante</i>
[RF-11]	[RF-11/ Informar e-mail cadastrado]	Campo e-mail deve ser obrigatório para o login, não ultrapassando 50 caracteres.		<i>Importante</i>
[RF-12]	[RF-12/ Informar senha cadastrada]	Campo senha deve ser obrigatório para o login, contendo até 8 caracteres alfanuméricos.		<i>Importante</i>
[RF-13]	[RF-13/ Mensagem de e-mail não cadastrado]	Mensagem de erro informando que o usuário não é cadastrado.		<i>Desejável</i>
<i>[SCE-008]</i>	<i>[SCE-008/ Cadastrar Usuário]</i>	<i>O sistema deverá permitir o cadastro de usuários para sua utilização.</i>	<i>[RF-14] [RF-15] [RF-16] [RF-17] [RF-18] [RF-19]</i>	<i>Importante</i>
[RF-14]	[RF-14/ Informar e-mail]	Campo e-mail deve ser obrigatório para o cadastro, não ultrapassando 50 caracteres.		<i>Importante</i>
[RF-15]	[RF-15/ Informar senha]	Campo senha deve ser obrigatório para o cadastro, contendo até 8 caracteres alfanuméricos.		<i>Importante</i>
[RF-16]	[RF-16/ Repetir senha]	Campo repetir senha deve ser obrigatório para o cadastro, sendo igual a primeira senha informada, contendo até 8 caracteres alfanuméricos.		<i>Importante</i>
[RF-17]	[RF-17/ Mensagem de erro formulário]	Mensagem de erro informando que o usuário esqueceu de preencher algum campo do cadastro, e informando o campo.		<i>Desejável</i>
[RF-18]	[RF-18/ Mensagem de confirmação]	Mensagem de confirmação do cadastro realizado com sucesso.		<i>Desejável</i>
[RF-19]	[RF-19/ Mensagem de confirmação por e-mail]	Mensagem de confirmação do cadastro enviado por e-mail.		<i>Desejável</i>
<i>[SCE-009]</i>	<i>[SCE-009/ Recuperar Senha]</i>	<i>O usuário poderá recuperar a senha cadastrada.</i>	<i>[RF-20][RF-21]</i>	<i>Importante</i>
[RF-20]	[RF-20/ Definição de lembrete de senha]	A definição da nova senha deverá ser encaminhada por e-mail.		<i>Desejável</i>
[RF-21]	[RF-21/ Esqueceu a senha]	Campo para recuperação de senha.		<i>Importante</i>

<i>[SCE-010]</i>	<b><i>[SCE-010/Modificar Usuário]</i></b>	<b><i>O usuário poderá alterar seu cadastro.</i></b>	<b><i>[RF-23][RF-24]</i></b>	<b><i>Importante</i></b>
[RF-23]	[RF-22/ Modificar e-mail cadastrado]	Campo para alteração de e-mail. O usuário cadastrado poderá alterar seu e-mail de cadastro.		<i>Desejável</i>
[RF-24]	[RF-23/ Modificar senha]	A senha informada poderá ser modificada pelo usuário.		<b><i>Importante</i></b>
<i>[SCE-011]</i>	<b><i>[SCE-011/Cancelar Cadastro]</i></b>	<b><i>O usuário poderá solicitar a exclusão do seu cadastro.</i></b>	<b><i>[RF-25]</i></b>	<i>Desejável</i>
[RF-25]	[RF-24/ Exclusão do cadastro]	Mensagem informada ao usuário se o mesmo deseja excluir seu cadastro. A mesma deverá ter um botão para confirmação (sim) e um para cancelar o pedido. Se marcado como sim, o sistema irá retirar o usuário do sistema.		<b><i>Importante</i></b>

### 3.1.3 Submissão Proteínas

Identificador	Nome	Descrição	Casos de uso relacionados	Prioridade
<b>[FEA-003]</b>	<b>[FEA-003/Submissão Proteínas]</b>	<b>O sistema permitirá que o usuário submeta para a predição uma proteína alvo.</b>	<b><i>[SCE-012]</i> <i>[SCE-013]</i> <i>[SCE-014]</i></b>	<b><i>Importante</i></b>
<i>[SCE-012]</i>	<i>[SCE-012/ Preencher formulário]</i>	<b><i>O usuário deverá informar todos os campos para submissão da proteína.</i></b>	<b><i>[RF-26] [RF-27] [RF-28]</i> <i>[RF-29] [RF-30]</i></b>	<i>Essencial</i>
[RF-26]	[RF-26/ Formulário da proteína alvo]	O campo de submissão da proteína deverá ser obrigatório.		<i>Essencial</i>
[RF-27]	[RF-27/ Formato FASTA]	O campo de submissão da proteína deverá aceitar a proteína alvo no formato FASTA.		<b><i>Importante</i></b>
[RF-28]	[RF-28/ Tamanho da proteína alvo]	O tamanho da proteína deverá ser até 200AA.		<i>Desejável</i>
[RF-29]	[RF-29/ Formulário da proteína alvo]	O formulário deve aceitar apenas caracteres do alfabeto e números.		<b><i>Importante</i></b>
[RF-30]	[RF-30/Mensagem de erro da submissão]	Se formulário de submissão da proteína estiver vazio ou em um formato não adequado o sistema terá que recusar a submissão, informando ao usuário.		<b><i>Importante</i></b>
<i>[SCE-013]</i>	<i>[SCE-013/ Validar Submissão]</i>	<b><i>Informar ao usuário que a submissão foi enviada corretamente.</i></b>	<b><i>[RF-31] [RF-32]</i> <i>[RF-33]</i> <i>[RF-34] [RF-35]</i></b>	<b><i>Importante</i></b>

[RF-31]	[RF-31/ Mensagem de resposta]	A mensagem de que a sequência da proteína foi enviada deve ser destacada para que o usuário tenha certeza de que submeteu a predição com sucesso.		<i>Desejável</i>
[RF-32]	[RF-32/ Tempo de resposta]	O sistema deve informar para o usuário o tempo que será necessário para predição.		<i>Importante</i>
[RF-33]	[RF-33/ Aviso de envio]	O servidor precisa informar ao usuário que a proteína está sendo gerada.		<i>Desejável</i>
[RF-34]	[RF-34/ Envio da predição para o servidor]	A submissão deve ser enviada ao servidor e ficar armazenada no banco de dados, respeitando a fila de execuções.		<i>Essencial</i>
[RF-35]	[RF-35/ Confirmação da predição]	Mensagem contendo a confirmação da predição, juntamente com o código da mesma.		<i>Desejável</i>
<i>[SCE-014]</i>	<i>[SCE-014/ Mostrar Código Submissão]</i>	<b><i>Informar o código da submissão.</i></b>	<b><i>[RF-36]</i></b>	<i>Desejável</i>
[RF-36]	[RF-03/ Código da Submissão]	Deverá ser um código numérico, de até 6 caracteres juntamente com a data da submissão.		<i>Desejável</i>

### 3.1.4 Executar Método

Identificador	Nome	Descrição	Casos de uso relacionados	Prioridade
[FEA-004]	<b>[FEA-004/Executar Método]</b>	<b>O sistema fará a execução do método CReF, recebendo a proteína alvo e retornando a estrutura predita.</b>	<b><i>[SCE-015] [SCE-016] [SCE-017] [SCE-018]</i></b>	<b><i>Essencial</i></b>
<i>[SCE-015]</i>	<i>[SCE-015/ Enfileirar submissões]</i>	<b><i>O sistema deverá enfileirar as submissões de acordo com a ordem do pedido da submissão e armazenar no banco de dados.</i></b>	<b><i>[RF-37]</i></b>	<b><i>Importante</i></b>
[RF-37]	[RF-37/Executar submissões]	As execuções das submissões deverão respeitar a fila de chegada das submissões.		<b><i>Importante</i></b>
<i>[SCE-016]</i>	<i>[RF-016/ Executar o método]</i>	<b><i>O sistema deverá executar a submissão solicitada, tendo como entrada a sequência FASTA da proteína e ao final a proteína no formato PDB</i></b>	<b><i>[RF-38]</i></b>	<b><i>Essencial</i></b>
[RF-38]	[RF-38/Saída Padrão]	O formato de saída padrão é a proteína no formato PDB		<b><i>Essencial</i></b>
<i>[SCE-017]</i>	<i>[SCE-017/ Salvar Log de Execução]</i>	<b><i>O sistema irá manter um log com as execuções realizadas.</i></b>		<b><i>Importante</i></b>
<i>[SCE-018]</i>	<i>[SCE-018/ Salvar Log de Erros]</i>	<b><i>O sistema irá manter um log com os erros ocorridos durante o método para correções do sistema.</i></b>	<b><i>[RF-39]</i></b>	<b><i>Importante</i></b>

[RF-39]	[RF-39/ Corrigir erros]	Os erros nas submissões e na execução do método devem ser corrigidos pelo administrador do sistema.		<i>Importante</i>
---------	-------------------------	---	--	-------------------

### 3.1.5 Visualizar Status

Identificador	Nome	Descrição	Casos de uso relacionados	Prioridade
[FEA-005]	<b>[FEA-005/Visualizar Status]</b>	<b>O sistema deverá informar ao usuário e ao administrador informações sobre o status do sistema.</b>	<i>[SCE-019] [SCE-020] [SCE-021] [SCE-022]</i>	<i>Importante</i>
<i>[SCE-019]</i>	<i>[SCE-019/ Visualizar versões do software usados no servidor]</i>	<i>O sistema deverá informar ao usuário as versões dos softwares que são estão sendo utilizadas na execução do método.</i>	<i>[RF-40] [RF-41]</i>	<i>Importante</i>
[RF-40]	[RF-40/Download do software]	Link para redirecionamento para página do software e seu Download.		<i>Desejável</i>
[RF-41]	[RF-41/Referencias bibliográficas]	Cada software utilizado deverá acompanhar suas referências bibliográficas, assim como Download do artigo citado.		<i>Desejável</i>
<i>[SCE-020]</i>	<i>[SCE-020/ Visualizar status de login]</i>	<i>O sistema deverá informar ao usuário seu status no sistema.</i>	<i>[RF-42]</i>	<i>Importante</i>
[RF-42]	[RF-42/Mensagem de login]	Uma mensagem visível ao usuário deverá demonstrar que o mesmo está logado.		<i>Desejável</i>
<i>[SCE-021]</i>	<i>[SCE-021/ Visualizar a última submissão ativa]</i>	<i>O Sistema deverá informar ao administrador a última submissão que foi efetuada.</i>	<i>[RF-41]</i>	<i>Desejável</i>
[RF-43]	[RF-43/Jobs em execução]	O sistema deve mostrar ao administrador a lista de Jobs.		<i>Desejável</i>
<i>[SCE-022]</i>	<i>[SCE-022/ visualizar log de submissão corrente]</i>	<i>O Sistema deverá manter para o Administrador a informação da submissão corrente que o servidor está realizando</i>		<i>Desejável</i>

### 3.1.6 Acompanhar Submissão

Identificador	Nome	Descrição	Casos de uso relacionados	Prioridade
[FEA-006]	<b>[FEA-006/Acompanhar Submissão]</b>	<b>O sistema permitirá que o usuário acompanhe sua submissão.</b>	<i>[SCE-023] [SCE-024] [SCE-025] [SCE-026] [SCE-027]</i>	<i>Importante</i>
<i>[SCE-023]</i>	<i>[SCE-023/ Listar fila de Submissões]</i>	<i>O sistema permitirá visualizar a lista de jobs.</i>	<i>[RF-44]</i>	<i>Importante</i>



[RF-44]	[RF-44/]	O sistema deve manter os dados pelo período de 15 dias.		<i>Desejável</i>
<i>[SCE-024]</i>	<i>[SCE-024/ Baixar Resultados]</i>	<b>O sistema permitirá Download dos resultados.</b>	<i>[RF-45] [RF-46]</i>	<b><i>Importante</i></b>
[RF-45]	[RF-45/ Formato do arquivo]	Os resultados serão fornecidos em formato PDF.		<i>Desejável</i>
[RF-46]	[RF-46/ Formato do arquivo]	A proteína predita para ser visualizada estará no formato padrão PDB		<b><i>Importante</i></b>
<i>[SCE-025]</i>	<i>[SCE-025/ Visualizar Log de Erro]</i>	<b><i>O sistema informará ao usuário se ocorreu um erro na predição</i></b>	<i>[RF-47]</i>	<b><i>Importante</i></b>
[RF-47]	[RF-47/ Mensagem de erro]	Uma mensagem de erro será informada ao usuário através do e-mail cadastrado.		<i>Desejável</i>
<i>[SCE-026]</i>	<i>[SCE-026/ Listar Submissões antigas]</i>	O sistema irá apresentar a lista de Jobs para usuário.	<i>[RF-48] [RF-49] [RF-50]</i>	<i>Desejável</i>
[RF-48]	[RF-48/ Informação da retirada dos Resultados]	O sistema irá informar ao usuário por quanto tempo o resultado da submissão ficará ativo.		<i>Desejável</i>
[RF-49]	[RF-03/ Tempo de retirada dos Resultados]	O resultado permanecerá disponível para Download por 14 dias.		<i>Desejável</i>
[RF-50]	[RF-03/ Lista de Jobs do usuário]	Um histórico dos Jobs de cada usuário ficará disponível.		<i>Desejável</i>
<i>[SCE-027]</i>	<i>[SCE-027/ Visualizar log de execução]</i>	O administrador poderá acompanhar o log de execução.		<b><i>Importante</i></b>

### 3.1.7 Enviar Resultados

Identificador	Nome	Descrição	Casos de uso relacionados	Prioridade
[FEA-007]	<b>[FEA-007/Enviar Resultados]</b>	<b>O sistema deverá enviar aos usuários o resultado das predições</b>	<i>[SCE-028] [SCE-020] [SCE-021] [SCE-022]</i>	<b><i>Essencial</i></b>
<i>[SCE-028]</i>	<i>[RF-028/Enviar resultados por e-mail]</i>	<b><i>O sistema deverá fornecer os resultados para o usuário por e-mail.</i></b>	<i>[RF-51] [RF-52]</i>	<b><i>Importante</i></b>
[RF-51]	[RF-01/ Resultados por E-mail]	O usuário receberá em seu e-mail a proteína no formato PDB e os dados da submissão em PDF		<b><i>Importante</i></b>
[RF-52]	[RF-01/ Aviso dos resultados]	O sistema deve informar como será o retorno da predição solicitada.		<i>Desejável</i>
<i>[SCE-029]</i>	<i>[SCE-029/ Enviar Log de execução por e-mail]</i>	<b><i>O sistema deverá fornecer o log de execução para o usuário por e-mail.</i></b>		<i>Desejável</i>

[SCE-030]	[SCE-030/ Enviar Log de erro por e-mail]	<b>O sistema deverá enviar o log de erro para o usuário por e-mail.</b>		<i>Desejável</i>
[SCE -031]	[SCE -31/ Configurar intervalo de envio de resultados]	<b>O sistema deve enviar diariamente as predições que terminaram sua execução.</b>	[RF-53]	<i>Importante</i>
[RF-53]	[RF-01/ Horário dos resultados]	O sistema deve enviar diariamente os resultados até a meia noite 00:00 diariamente.		<i>Desejável</i>

### 3.2 Requisitos Não-funcionais

Os requisitos que descrevem os aspectos não-funcionais do sistema são apresentados a seguir:

#### 3.2.1 Requisitos de Segurança

Identificador	Descrição	Prioridade
[RNFSEG01]	Somente o administrador do sistema poderá realizar configurações, alterações no método e exclusão de usuários.	Desejável

#### 3.2.2 Requisitos de Usabilidade

Identificador	Descrição	Prioridade
[RNFUSA01]	Interface WEB: A interface com o usuário é de vital importância para o sucesso desse sistema. Para verificação da usabilidade do sistema novos testes de usabilidade no produto deverão ser realizados.	Importante
[RNFUSA02]	Na sessão barreiras de usabilidade, encontram-se os principais problemas encontrados nos servidores de predição avaliados, e, portanto, essas recomendações devem ser seguidas.	Importante

#### Requisitos de Operacionais

Identificador	Descrição	Prioridade
[RNFOPE01]	O sistema deve ser desenvolvido em HTML/CSS, Python.	Importante
[RNFOPE02]	O sistema deve ser desenvolvido em uma arquitetura em camadas.	Desejável
[RNFOPE03]	A camada de aplicação para Web compatível com browsers de mercado (Internet Explorer, Firefox, Internet Explorer).	Importante

### 3.2.3 Requisitos de Confiabilidade

Identificador	Descrição	
[RNFCON01]	O sistema deve estar disponível 24 horas por dia durante os 7 dias da semana. Por não se tratar de um sistema crítico, o sistema poderá ficar fora do ar até que seja corrigida alguma falha que possa ocorrer.	Essencial

### 3.2.4 Requisitos de Desempenho

Identificador	Descrição	
[RNFDDES01]	Embora não seja um requisito essencial ao sistema, deve ser considerada por corresponder a um fator de qualidade de software, pois geralmente usuários de bioinformática tem necessidade de resultados rápidos em suas pesquisas.	Essencial

### 3.2.5 Hardware e Software

Identificador	Descrição	
[RNFDDES01]	Como se trata de uma aplicação Web ela deve ser acessada por qualquer navegador, que esteja conectado na rede (internet), não ficando definidos aqui o hardware e software necessários.	Importante

## • Barreiras de Usabilidade

Nesta sessão descrevemos uma compilação dos problemas encontrados nos servidores de predição estudados para a definição deste documento, separados pela Heurística Violada.

1.	<b>Visibilidade do status do sistema:</b>
	O sistema deve informar para o usuário o tempo que será necessário para predição.
	O servidor precisa informar ao usuário que a proteína está sendo gerada.
	O sistema deve mostrar ao usuário a lista de Jobs.
	O sistema deve informar como será o retorno da predição solicitada.
	A mensagem de que a sequência da proteína foi enviada deve ser destacada para que o usuário tenha certeza de que submeteu a predição com sucesso.
	O resultado da predição deve ser enviado por e-mail. Isso dificulta a utilização do servidor como parte de um workflow. O ideal seria oferecer uma página com os resultados que pode ser lida em tempos e tempos para determinar se a predição foi concluída.
	A parte de registro no site não fica disponível até a hora da submissão da sequência. Isto deveria estar disponível na interface de uma forma mais evidente.
2.	<b>Compatibilidade entre sistema e mundo real:</b>
	A maioria das páginas não possuem botões ou ícones simbolizando ações familiares ao usuário como botão voltar, avançar, home, salvar etc., simbolizando ações familiares ao usuário.

	Em uma situação real, o usuário pode desejar realizar mais de uma predição, o que não é permitido. Isso torna os sistemas limitados. No que se refere a sistemas, no mundo real espera-se que eles agilizem os serviços.
	Excesso de menu e de informações, sem categorização. Isso atrapalha o usuário a focar-se na predição.
3.	<b>Controle e liberdade para o usuário:</b>
	Os sistemas não apresentarem botões de avançar, voltar, <i>undo</i> , <i>redo</i> (desfazer e refazer) dentre outros (se apresentam não são padronizados, os atalhos acontecem somente em algumas telas da interface). A maioria dos botões eram somente aqueles apresentados pelo navegador.
	Há perda de dados de formulários quando ocorre um erro quando o usuário clica em botão voltar.
	O usuário não consegue trocar a senha cadastrada, ou o modificar seu cadastro. Deve se dar a opção do mesmo poder modificar os dados cadastrados.
	O sistema deve ser uma opção de sair do sistema (logon).
	Mensagens de aviso ao usuário realizar uma ação devem ser priorizadas, a fim de evitar erros. Exemplo o botão “ <i>clean form</i> ” ao ser pressionado deve pedir uma confirmação ao usuário, pois o mesmo apaga as informações digitadas pelo usuário.
4.	<b>Consistência e padrões:</b>
	As mensagens de erro não seguem um padrão. Algumas abrem em uma nova janela do navegador, em formato de texto, sem botões para conformação ou retornar ao estado anterior. As mensagens deveriam aparecer na mesma tela que ocorreu o erro, como padrão em qualquer sistema operacional.
	Faltam botões ou botões com funções muito diferente são muito similares.
	Os menus das interfaces não são únicos, padronizados, possuindo vários formatos, localização diferente conforme se percorre a interface. Manter um menu padrão durante toda interface.
	Campos obrigatórios poderiam utilizar a simbologia usual que indica campo deste tipo. Isso agiliza a utilização por parte do usuário.
	Em algumas telas mensagens de usuário é incluindo links de navegação e outras não. Comportamentos diferentes para mesmas ações.
	Definir um padrão do que é necessário o usuário informar previamente para submissão de seu <i>job</i> . Por vezes é atrelada a conferência do e-mail, outras, que falta a senha, outras relacionadas a sequência informada.
	Os campos obrigatórios devem ser sinalizados.
5.	<b>Prevenção de erros:</b>
	Os sistemas pedem a entrada no formato FASTA. Porém testes foram realizados em outros formatos, e mesmo assim o sistema aceitou ou a submissão ou o carregamento de um arquivo não compatível.
	Mesmo com a sequência da proteína errada, o sistema está mais preocupado com a testagem do e-mail do que a submissão correta.
	Não há mensagens de erro ao se colar uma sequência de proteína no formato errado.
	O sistema deve ter que realizar o login para poder enviar a predição, ou avisar o usuário previamente do mesmo antes de utilizar a ferramenta. Muitos erros aconteceram por causa do login do usuário.
	O sistema deve dar ao usuário a escolha do tipo de predição da estrutura, modificando parâmetros de entrada. Há servidor que apresenta essa possibilidade, porém a funcionalidade ou está desabilitada ou não funciona.
	Quando o sistema pedir uma senha para cadastro, o sistema deverá utilizar para quando o usuário quiser logar no sistema. Houve caso de que a senha cadastrada não foi utilizada em nenhum caso.
	O cadastro do usuário deve ser único para todo sistema (erros ocorreram como, por exemplo, o usuário tinha um cadastro para enviar a predição e outro para participar do fórum de discussão).
	Os avaliadores acham que o fato de que o sistema só aceita e-mail acadêmico não é uma boa forma de controlar o uso do sistema. Ocorreram testes com e-mails não acadêmicos onde o sistema aceitou o login, outros que aceitou e-mails não existentes (como fulano@xxx.com). A mudança nas permissões de e-mail poderá permitir que o sistema fosse aberto a outros estudantes ou professores, sendo utilizado também como ferramenta educativa.
	O sistema deve apresentar um exemplo da sequência FASTA correta a ser utilizada na página principal para direcionar o usuário.

	Ao preencher o campo correspondente a sequência de forma indevida, o servidor deveria verificar o tamanho e a formatação da mesma e indicar a irregularidade. Isto só é feito depois de enviar a submissão.
	As mensagens de erro devem ser apresentadas em destaque, no formato padrão (janelinha), na mesma janela, junto ao erro. O que ocorre muitas vezes é uma mensagem em vermelho, mas que pode ser confundida com outras mensagens de destaque na página.
6.	<b>Reconhecimento em lugar de lembrança:</b>
	Não há um caminho demarcado para saber onde o usuário se encontra em determinado momento.
	Alguns menus apresentados com algumas funcionalidades como “ <i>Fragment libraries</i> ” ou Structure Prediction não tem explicações como são implementadas essas funcionalidades ou documentação adequada. O usuário pode não reconhecer o que significa ou a parte que ele está na interface.
	Alguns termos têm links associados, mas é necessário abrir outra página para uma explicação de seu significado. Melhor seria ter uma ajuda rápida ao passar o mouse, ou símbolo de ajuda que explicasse o dado em uma linha.
7.	<b>Flexibilidade e eficiência de uso:</b>
	Somente usuários cadastrados podem utilizar dos sistemas.
	O servidor solicita que o usuário apague seus Jobs depois de serem completados, de uma forma de economizar espaço em disco. No entanto apagar uma <i>job</i> é um processo composto por um conjunto de ações. Isto deveria ser uma ação única do sistema, integrada ao sistema de Jobs.
	O usuário deve ter a opção de cancelar a predição enviada, pois muitos servidores só aceitam uma predição por usuário.
	O sistema não permite customização do processo submetido. Não fazendo a distinção entre usuários experientes e inexperientes.
8.	<b>Design estético e minimalista:</b>
	É utilizado muito texto para repassar informações aos usuários, geralmente diferenciados somente por títulos. Não há uma padronização nas telas, ficando o conteúdo confuso e com muita informação reunida. Priorizar o uso de atalhos, botões, mensagem, distribuição das informações de acordo com interesse do usuário, reunidas em blocos ou novas telas.
	Menu com as opções de acesso as páginas, e junto ações para serem executadas pelo usuário. Isso polui bastante a interface. Opções poderiam ser mostradas somente quando clicado ou quando o ponteiro do mouse fosse colocado sobre itens.
	Há informações em vermelho no meio do texto. Apesar de serem importantes, avisos poderiam ser apresentadas de outra forma, para não distrair os usuários.
9.	<b>Auxiliar os usuários a reconhecer, diagnosticar e recuperar erros:</b>
	Possibilitar ao usuário a recuperação de senha, nome de usuário.
	Identificar se o usuário já é um usuário cadastrado no sistema.
	Padronizar os avisos de erros. Os erros aparecem em diferentes formas na interface, em outras janelas, ou em formato texto igual ao de outras informações, como campos obrigatórios.
	Em caso de erros não há a possibilidade de se voltar a o estado inicial.
	Os erros não apresentam cores ou símbolos diferenciados a fim de chamar atenção do usuário.
	Novamente a predição enviada deve ter a possibilidade de cancelamento. Em uma avaliação o usuário enviou uma submissão errada, além do sistema não detectar, ainda o usuário não conseguiu enviar outra até que essa fosse processada.
10.	<b>Ajuda e documentação:</b>
	A documentação dos arquivos de entrada e saída é muito superficial e em formato .txt, o que dificulta o foco somente em que tem interesse em configurar e visualizar.
	Falta esclarecimento ao usuário o do porquê utilizar e-mail acadêmico e/ou que se pode enviar só um <i>job</i> por vez na documentação.
	Poderia haver uma ajuda contextual, para cada campo de entrada.
	Não há ajuda contextuais. Dessa forma resta o usuário acessar a documentação e tentar localizar o que deseja, sem a ajuda do sistema.

Os parâmetros apresentados nos resultados devem ser explicados, através de links e documentação adequada. Há muitos parâmetros como o que significa E ou H na predição da estrutura secundária, ou o que significa SignalP, C-Score, S-Score ou Y-Score, ou do parâmetro TM.
--

## • Sugestões

Este documento está em sua versão 1.0. Ao desenvolvimento desta pesquisa, novas funcionalidades poderão ser adicionadas ao conteúdo aqui descrito.

# APÊNDICE G – QUESTIONÁRIO DE AVALIAÇÃO DO SERVIDOR DE PREDIÇÃO DE ESTRUTURAS DE PROTEÍNAS (WCREF)

## Descrição da avaliação

Este estudo pretende realizar uma avaliação de usabilidade do servidor de predição de estruturas de proteínas do método CReF (*Central Residue Fragment-based Method*) (wCReF). O objetivo é que cada participante percorra a interface do wCReF buscando conhecer, avaliar e validar os recursos, para que futuramente suas funcionalidades sejam aprimoradas.

Sua avaliação é importante, pois através dela buscamos melhorias na interface do servidor wCReF para sua disponibilização através da Web como uma ferramenta para pesquisas no campo da bioinformática.

## Condução da avaliação

Para realização deste questionário primeiramente você deve percorrer toda a interface para conhecê-la. Após o contato inicial, em um segundo momento, deverá percorrer novamente toda interface buscando responder a esse questionário (Baseado nas heurísticas de Nielsen, adaptado de Ssemugabi e Villiers, 2007).

O questionário está dividido em blocos, com a definição do que se está avaliando e suas respectivas perguntas. O primeiro bloco da avaliação traz a definição do perfil do usuário. A segunda parte é relativa a avaliação do servidor. São dois tipos de questões: fechadas, onde você terá que marcar a opção conforme sua opinião; e abertas que são as quais você fica livre para responder.

Para avaliar o nível de satisfação do usuário definimos 5 opções de acordo com a sua opinião:

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

## Cenário de uso:

Você é um pesquisador e deseja utilizar um servidor de predição de estruturas de proteínas para determinar um modelo de sua proteína alvo para depois aplicar um protocolo de refinamento. Para isso escolheu a proteína 1ZDD. Para iniciar copiou a sequência da proteína, no formato FASTA, do Protein Data Bank (<http://www.rcsb.org/pdb>). Ao acessar a Web encontrou o wCReF. Percorreu a página inicial, e percebeu que era necessário um cadastro para sua utilização. Ao concluir o cadastro realizou login no site do servidor. Sendo assim pode enviar a sequência. Após o término da predição verificou os resultados. Achou o método interessante e para buscar mais informações sobre ele visualizou a documentação, da mesma forma que quando teve dúvidas procurou ajuda\*

\*Este cenário é um exemplo para seguir. Você pode realizar as atividades que achar necessárias para avaliação, conforme achar necessário.

# QUESTIONÁRIO DE AVALIAÇÃO

## 1 PERFIL DO USUÁRIO

### 1.1 Área de estudo/curso

1.1.1 Qual é o curso de pós-graduação que você está vinculado?

---

1.1.2 Qual é a sua formação? (Ex. Graduação em ciências biológicas, farmácia, etc.?)

---

### 1.2 Experiência do usuário com a área de domínio - bioinformática

1.2.1 Há quanto tempo você trabalha e/ou estuda bioinformática?

- Menos de 6 meses                       Entre 1 e 2 anos                       Mais de 5 anos  
 Entre 6 meses e 1 ano                       Entre 2 e 5 anos

1.2.2 Você utiliza aplicações Web para suas atividades em bioinformática?

- Sim                       Não                       Às vezes

a) Se sim, qual a frequência que você utiliza essas aplicações?

- Utilizo diariamente                       Mensalmente  
 Semanalmente                       Não utilizo com muita frequência

b) Cite exemplo (s) de aplicações Web que você utiliza para este fim:

---

### 1.3 Experiência do usuário na bioinformática estrutural

1.3.1 Você sabe o que é Predição de Estruturas de Proteínas?

- Sim     Não

1.3.2 Você conhece algum servidor on-line de Predição de Estruturas de Proteínas?

- Sim    Não

1.3.3 Se sim, qual (is)?

- |   |                                       |  |
|---|---------------------------------------|--|
| <input type="checkbox"/> Robetta        | <input type="checkbox"/> SPARKSX      | <input type="checkbox"/> PMS: A Panoptic Motif Search Tool |
| <input type="checkbox"/> I-TASSER       | <input type="checkbox"/> MULTICOM     | <input type="checkbox"/> Outro (s). Especifique:           |
| <input type="checkbox"/> QUARK          | <input type="checkbox"/> IntFOLD      | _____  |
| <input type="checkbox"/> RaptorX server | <input type="checkbox"/> Chunk-TASSER | _____  |



## 2 CONCEPÇÃO DA INTERFACE

### 2.1 Visibilidade do status do sistema:

#### Definição:

O sistema deve sempre manter os usuários informados sobre o que está acontecendo através de feedback apropriado, em um tempo razoável.

#### Questões relacionadas:

2.1.1 O Sistema me mantém informado sobre o que está acontecendo, através de feedback.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.1.2 Eu entendo o que significa feedback.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.1.3 Eu obtenho o feedback dentro de um tempo razoável.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.1.4 Posso ver ou ouvir os resultados de cada ação que eu realizo.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.1.5 O sistema não reage de forma que me surpreenda e não faz nada inesperado.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.1.6 Anote o problema( s) ou sugestão (ões) que você julga estar relacionado com esta seção:



### 2.3 Controle e liberdade para o usuário:

Os usuários frequentemente escolhem as funções do sistema por engano e então necessitam de "uma saída de emergência", claramente definida para sair do estado não desejado, sem ter que percorrer um longo diálogo, ou seja, é necessário suporte a *undo* e *redo* (*desfazer* e *refazer*).

2.3.1 O sistema trabalha da forma que eu quero que trabalhe.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.3.2 Cada página apresenta todos os botões de navegação ou hiperlinks necessários, tais como, anterior (voltar), próxima e página inicial (home).

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.3.3 Quando eu cometo um erro eu não posso escolher sair do sistema usando um botão de saída de emergência claramente sinalizado.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.3.4 Anote o problema (s) ou sugestão (ões) que você julga estar relacionado com esta seção:

### 2.4 Consistência e padrões:

Referem-se ao fato de que os usuários não deveriam ter acesso a diferentes situações, palavras ou ações representando a mesma coisa. A interface deve ter convenções não ambíguas, seguindo as convenções da plataforma.

2.4.1 A mesma convenção (normas ou o caminho pelo qual o conteúdo é organizado e apresentado) é usada através do sistema.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.4.2 A convenção usada é similar àquelas em outros sistemas que utilizei.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.4.3 As mesmas palavras, frases, situações ou ações referem-se às mesmas coisas através do sistema.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.4.4 As cores são usadas de forma consistente (mesma forma) através do sistema.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.4.5 Os gráficos, ícones e imagens são consistentemente (mesma forma) usados através do sistema.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.4.6 Há consistência no Layout do sistema.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.4.7 Há consistência no uso dos menus.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.4.8 Há consistência no uso do tipo e tamanhos das fontes.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.4.9 Os links das páginas são consistentes com os títulos das páginas as quais estão vinculadas.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.4.10 Anote o problema (s) ou sugestão (ões) que você julga estar relacionado com esta seção:

## 2.5 Prevenção de erros

Ainda melhor do que boas mensagens de erro é um projeto cuidadoso que impede que um problema ocorra em primeiro lugar. O sistema deve eliminar as condições passíveis de erros ou verificá-los, e apresentar aos usuários uma opção de confirmação, antes de se comprometer com a ação. Os erros são as principais fontes de frustração, ineficiência e ineficácia durante a utilização do sistema.

2.5.1 O sistema me dá suporte de forma que se torna difícil cometer erros graves.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.5.2 A qualquer hora que um erro é cometido uma mensagem de erro é apresentada.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.5.3 O sistema mostra uma interface de usuário gráfica (demonstrando como, por exemplo, listas suspensas (*drop down*), hiperlink ou ícones, que podem ser clicadas por mouse, ao invés de baseada em comandos, onde os comandos têm de ser digitados pelo uso de teclado).

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.5.4 Sou requisitado a confirmar minhas entradas antes de levar adiante ações "potencialmente perigosas" como a de "apagar".

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.5.5 Acho fácil entrar com as informações no sistema.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.5.6 Anote o problema (s) ou sugestão (ões) que você julga estar relacionado com esta seção:

## 2.6 Reconhecimento ao invés de lembrança

Minimizar a carga de memória do usuário, fazendo objetos, ações e opções visíveis. O usuário não deve ter que lembrar informações de uma parte do diálogo para outra. Instruções para o uso do sistema devem estar visíveis ou facilmente acessíveis.

2.6.1 Estão disponíveis instruções de como utilizar o sistema.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.6.2 Há uma relação óbvia entre os controles e suas ações.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.6.3 Os objetos usados, como os gráficos em barras de ferramentas, são fáceis de reconhecer.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.6.4 Ao trabalhar em uma tarefa eu não preciso lembrar as informações de outras tarefas.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.6.5 Anote o problema (s) ou sugestão (ões) que você julga estar relacionado com esta seção:

## 2.7 Flexibilidade e eficiência do uso

O sistema deve ser adequado tanto para usuários inexperientes quanto para usuários experientes. Deve permitir aos usuários personalizar ações frequentes, como por exemplo, a utilização de aceleradores - invisíveis pelo usuário novato - podem frequentemente acelerar a interação para o usuário experiente.

2.7.1 O site oferece diferentes níveis de usuários, desde o novato até o experiente.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.7.2 Atalhos em forma de abreviações, questões especiais, macros e comandos escondidos estão disponíveis para usuários experientes.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.7.3 O site guia os usuários novatos de forma competente.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.7.4 Há a opção de usar somente o teclado para realizar as tarefas.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.7.5 O sistema é o bastante flexível para permitir que os usuários ajustem as configurações adequando-as, isto é, personalizando o sistema.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.7.6 Anote o problema (s) ou sugestão (ões) que você julga estar relacionado com esta seção:

## 2.8 *Design* estético e minimalista

Os diálogos não devem conter informações irrelevantes ou raramente necessárias. Cada unidade extra de informação em um diálogo compete com unidades relevantes e diminui sua visibilidade relativa.

2.8.1 As páginas contêm a informação requisitada.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.8.2 A informação em cada página não é muito grande para confundir-me ou distrair-me.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.8.3 Não há o uso excessivo de gráficos e imagens no site.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.8.4 As caixas de diálogo proporcionam informações adequadas ao desempenho das tarefas.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.8.5 Menus e listas suspensas possuem as opções exigidas para as escolhas.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.8.6 Anote o problema (s) ou sugestão (ões) que você julga estar relacionado com esta seção:

## 2.9 Reconhecimento, diagnóstico e recuperação de erros

Auxiliar os usuários a reconhecer, diagnosticar e recuperar erros: as Mensagens de erro devem ser expressas em linguagem natural/clara (sem códigos), indicando precisamente o erro/problema e sugerir uma solução.

2.9.1 As mensagens de erros são expressas em linguagem simples.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.9.2 As mensagens de erros indicam precisamente qual é o problema.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.9.3 Cada mensagem proporciona um procedimento para a correção do erro.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.9.4 O procedimento para corrigir um erro é específico, rápido e eficiente.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.9.5 O site proporciona uma rápida mudança de ação pela qual é possível, por exemplo, disponibilizar ambos Desfazer (Undo) e Refazer (Redo).

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.9.6 Anote o problema (s) ou sugestão (ões) que você julga estar relacionado com esta seção:

## 2.10 Ajuda e documentação

Mesmo que seja melhor que o sistema possa ser usado sem documentação, pode ser necessário fornecer ajuda e documentação. Tais informações devem ser fáceis de encontrar, ser centradas na tarefa do usuário, listar passos concretos a serem realizados e não ser muito grande. A ajuda deve estar facilmente acessível e on-line.

2.10.1 Eu acho os recursos de ajuda úteis (tais como ajuda online, glossário).

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.10.2 Os recursos de ajuda são fáceis de usar.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.



2.10.3 Eu acho fácil procurar pela ajuda solicitada.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

2.10.4 Anote o problema (s) ou sugestão (ões) que você julga estar relacionado com esta seção:

### 3 WEB DESIGN

#### 3.1 Simplicidade de navegação do site, organização e estrutura

3.1.1 Não há necessidade de usar o Menu uma vez que existe um mapa do curso que tem a mesma finalidade.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

3.1.2 Os links de site sempre apontam para os documentos ou páginas corretas.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

3.1.3 Existem links (atalhos) para as seções dentro da mesma página.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

3.1.4 As cores para os links são consistentes com as convenções da Web, ou seja, as ligações não visitadas estão em azul e aquelas visitadas estão em verde ou roxo.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

3.1.5 A informação relacionada é colocada em conjunto.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

3.1.6 A informação importante é colocada no topo de uma página.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

3.1.7 A barra de rolagem ou *Scrolling* é minimizada, ou seja, eu não tenho para percorrer muitas páginas para encontrar as informações necessárias.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

3.1.8 Anote o problema (s) ou sugestão (ões) que você julga estar relacionado com esta seção:

## 4 BIOINFORMÁTICA

### 4.1 Relevância do conteúdo do site para a Bioinformática

4.1.1 O conteúdo segue os padrões de aplicações em bioinformática.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.1.2 As informações são relevantes para o desenvolvimento das atividades no servidor.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.1.3 O conteúdo está no nível apropriado para minha compreensão.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.1.4 Está claro quais os materiais que tem direitos autorais e quais não possuem.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.1.5 O conteúdo do servidor cita as devidas referências do material que é utilizado.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.1.6 Anote o problema (s) ou sugestão (ões) que você julga estar relacionado com esta seção:

### Resultados

4.2.1 Eu compreendi quais os objetivos do servidor.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.2.2 Eu compreendi a metodologia utilizada no wCReF.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.2.3 Consigo realizar uma submissão de uma proteína-alvo satisfatoriamente.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.2.4 Se sou um usuário avançado, consigo compreender os parâmetros opcionais na submissão do wCReF.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.2.5 Os parâmetros opcionais do wCReF estão de acordo com os utilizados por aplicações relacionadas que eu conheço.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.2.6 Consigo encontrar com clareza as informações sobre os resultados.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.2.7 O tempo de resposta da minha submissão está de acordo com o que eu esperava.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.2.8 Os resultados estão apresentados de maneira clara.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.2.9 Os resultados estão apresentados de maneira concisa.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.2.10 Os resultados estão de acordo com aquilo que eu esperava.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.2.11 Consigo salvar meus resultados.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.2.12 Os resultados da submissão são comunicados a mim sem eu estar conectado ao servidor, por outros meios, como por exemplo e-mail.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.2.13 As estatísticas me fornecem informações úteis.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.2.14 Como pesquisador da área posso enviar sugestões, reclamações e tirar dúvidas com a equipe responsável pelo servidor.

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

4.2.15 Anote o problema(s) ou sugestão(ões) que você julga estar relacionado com esta seção:

## 5 CONCLUSÕES

5.1.1 **Eu achei o sistema fácil de usar.**

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

5.1.2 **É rápido de se trabalhar no sistema.**

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

5.1.3 **O sistema desempenha as tarefas apropriadamente.**

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

5.1.4 **Uma vez que eu aprendi a usar o sistema, será fácil de usá-lo na próxima vez.**

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente.

5.1.5 **Eu fiquei satisfeito com o sistema.**

Concordo plenamente  Concordo  Talvez  Discordo  Discordo plenamente

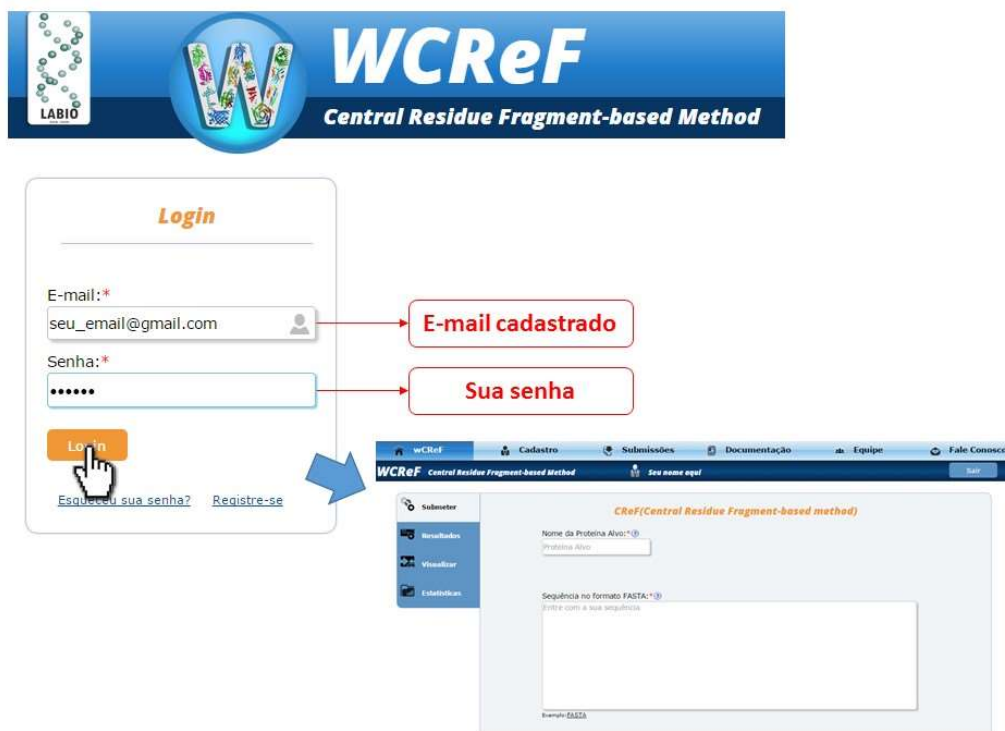
5.1.6 Use o espaço que segue para preencher os aspectos que você considera como os mais positivos do sistema (questão aberta).

5.1.7 Use o espaço que segue para preencher os aspectos que você considera como os mais problemáticos do sistema (questão).

## APENDICE H – TUTORIAL PARA SUBMISSÃO DE PROTEÍNAS NO WCREf

Uma forma de ajudar o usuário a utilizar a interface do wCReF é através do Tutorial. O tutorial com a explicação de como o usuário deve realizar uma predição está descrito a seguir:

2) Para submeter uma proteína você deve primeiramente realizar login no sistema:

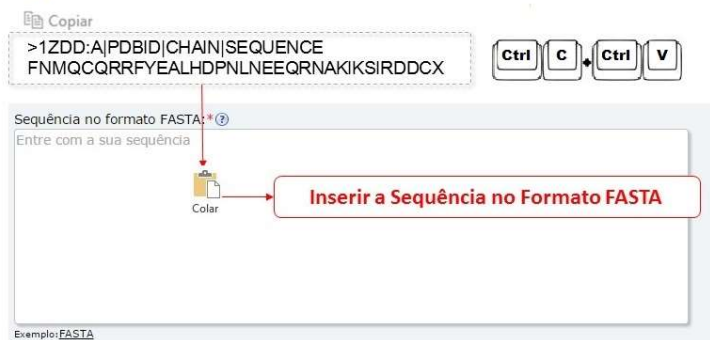


3) Você será direcionado para página principal. Aqui você pode enviar sua proteína alvo. Primeiramente você deve colocar o nome da proteína a ser enviada no primeiro campo do formulário. No segundo campo a sequência no formato FASTA:

1º Nome da Proteína Alvo:



2º Sequência no Formato Fasta:



- Formato FASTA:



- Para saber mais sobre este formato clique em FASTA:

4) O wCReF possui parâmetros opcionais. Se os mesmos não forem modificados receberão os valores default (padrão). Os primeiros parâmetros são os referentes ao wCReF e os segundos ao BLAST (*Basic Local Alignment Search Tool*):

\*Saiba mais sobre o Blast lendo a nossa documentação, na guia softwares.

### 3º Parâmetros Opcionais:

**1º Tamanho do fragmento:** Define quantos aminoácidos o fragmento irá possuir: 5 (padrão), 7 ou 9.

**2º Número de clusters a ser usado para os ângulos Phi e Psi:** O agrupamento é feito no WEKA (Waikato Environment for Knowledge Analysis) buscando as regiões mais favoráveis para formação das estruturas secundárias no mapa de Ramachandran.

**3º Excluir homólogos com similaridade acima:** Podem ser excluídas da busca de templates no PDB para os fragmentos as proteínas que possuem similaridade acima de 100%, 90%, 80% ou 70%.

**4º Excluir homólogos baseados no código PDB:** Colocar os código do PDB das proteínas que você seja que seja excluída na sua busca separada por vírgulas. Ex: 1ZDD, 1ENY, 1BVR...

5) Depois de preencher o formulário basta clicar em enviar:

### 4º Enviar predição:

