

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

ProtCool 2.0: um modelo cliente/servidor para um gerador de protocolos de ancoragens e simulações de dinâmica molecular em complexos proteína-ligante

Moisés Pinheiro Souza

Itabira, abril de 2023

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Moisés Pinheiro Souza

ProtCool 2.0: um modelo cliente/servidor para um gerador de protocolos de ancoragens e simulações de dinâmica molecular em complexos proteína-ligante

Dissertação submetida ao Programa de Pós-Graduação em Ciência e Tecnologia da Computação como parte dos requisitos para obtenção do Título de Mestre em Ciência e Tecnologia da Computação

Área de Concentração: Matemática Computacional

Linha de pesquisa: Inteligência Artificial

Orientador: Prof. Dr. Carlos Henrique Silveira

Abril de 2023
Itabira – MG

AGRADECIMENTOS

Agradeço a Deus pela vida e pelo privilégio de alcançar mais essa conquista que Ele me preparou, além de ter aprendido Dele que tudo tem um tempo determinado para todas as coisas.

Expresso minha profunda gratidão à minha família, especialmente à minha esposa e companheira de vida, Stephanie, pelo suporte, paciência e amor dedicado a mim. Ela tem sido uma verdadeira guerreira, enfrentando momentos difíceis em que, muitas vezes, não pude estar presente devido a este trabalho. Também agradeço ao meu filho Lucas por sua torcida incansável, sendo uma fonte constante de inspiração para mim no dia a dia.

Não poderia deixar de mencionar meus pais, que sempre oraram por mim e acreditaram neste projeto. Dedico este trabalho ao meu pai, Sr. José Antônio de Souza, que sempre foi e será minha maior referência como homem. Também dedico à minha mãe, Sra. Vilma de Jesus Pinheiro Souza, pelo seu amor incondicional e carinho. Agradeço ainda ao meu sobrinho, Filipe Souza Silva, que me auxiliou durante todo o desenvolvimento deste trabalho. Parableno-o por sua dedicação, comprometimento e trabalho, e tenho muito orgulho por ele ter escolhido a mesma carreira que a minha.

Confesso que sem sua ajuda, este trabalho seria muito mais difícil de ser concretizado. Desejo-lhe todo o sucesso.

Quero expressar minha gratidão à UNIFEI por me acolher tão bem como mestrando de outra instituição, em especial ao meu orientador, Prof. Carlos da Silveira. Agradeço pela oportunidade, confiança e aprendizado que obtive com ele. Sua liderança é admirável e aprendi a respeitá-lo pela sua experiência e orientação.

Também gostaria de agradecer a toda a equipe do INSILICO, em especial à Profa. Fabiana Guedes, pela atenção e disponibilidade sempre demonstradas. Agradeço também ao meu amigo e colega de IC, Levi Magny, por sua amizade e suporte ao longo desta jornada. Desejo-lhe muito sucesso.

Por fim, agradeço a todos que me apoiaram de forma direta ou indireta, bem como a todos os amigos e parceiros que encontrei ao longo desta caminhada. É com um agradecimento especial que concluo, expressando minha gratidão à Vale S.A pelo apoio financeiro e ao IFSULDEMINAS pelo incentivo.

*'Em ti confiarão os que conhecem o teu nome; porque tu, Senhor,
nunca desamparaste os que te buscam.'*

Salmos 9:10

RESUMO

A pandemia de COVID-19 evidenciou a alta demanda por sistemas computacionais que acelerem a indicação de novos fármacos. Nesse sentido, compreender o comportamento dinâmico de complexos biomoleculares é algo fundamental. As técnicas que envolvem simulações de dinâmica molecular desses complexos vêm sendo crescentemente utilizadas como forma de agilizar a indicação de melhores candidatos à fármacos. No entanto, a preparação de tais simulações é de grande complexidade, e seus inúmeros detalhes nem sempre são suficientemente destacados, comprometendo sua reprodutibilidade e reusabilidade. Para tanto, foi proposta a ferramenta ProtCool – um gerador de protocolos, focado na integração entre ancoragem e dinâmica molecular de complexos proteína-ligantes. Em sua primeira versão, essa ferramenta era de uso restrito ao ambiente local do usuário. Este trabalho apresenta a versão 2.0 do ProtCool, desenvolvida sob um modelo cliente-servidor com interface WEB. O objetivo é preencher as lacunas deixadas pela versão anterior, aprimorando o software em três aspectos fundamentais: torná-lo multiplataforma, possibilitar o acesso a múltiplos usuários e tornar a ferramenta mais intuitiva. O desenvolvimento de uma interface amigável permite que essa nova versão expanda o domínio de uso também para pesquisadores inexperientes ou novatos em química computacional. O ProtCool 2.0 não executa as dinâmicas após preparação nem opera análises dos resultados, ele é construído para ser um *expert* na preparação baseado em *workflows*, percorrendo o *workflow* que lhe foi programado, gerando todos os arquivos de configuração necessários para uma execução confiável da dinâmica molecular no aparato computacional do usuário. Como todo o processo fica devidamente registrado, isso permite maior reprodutibilidade e reusabilidade das preparações. Boa parte de suas funcionalidades estão fundamentadas na adaptação das ferramentas mais conhecidas pela literatura no campo de simulações de dinâmica molecular. O ProtCool 2.0 foi desenvolvido utilizando as melhores práticas e processos de engenharia de software. Sua arquitetura cliente-servidor implementada sob o padrão web permite que ele seja multiplataforma e multiusuário, característica que traz ganhos em disponibilidade e desempenho. Ele possui uma interface gráfica minimalista que oferece recursos interativos que concedem segurança ao usuário para o correto preenchimento de seus parâmetros de estudo, evitando erros. Para demonstrar seu uso, o ProtCool 2.0 foi submetido a um estudo de caso na simulação da interação entre acetilcolinesterase e galantamina, usada no tratamento do Alzheimer, o que possibilitou uma validação por replicação de uma simulação certificada por pares em publicação internacional. Foi possível demonstrar ao final que a preparação da simulação de fato permitiu a execução de dinâmicas moleculares confiáveis, reproduzindo os resultados esperados. Espera-se que esta ferramenta possa não somente trazer maior agilidade, reprodutibilidade e reusabilidade às preparações de dinâmica molecular, como contribuir para suavizar a curva de aprendizagem dessas simulações em química computacional.

ABSTRACT

The COVID-19 pandemic has made it clear the high demand for computational systems that expedite the discovery of new drugs. In this regard, understanding the dynamic behavior of biomolecular complexes is crucial. Techniques involving molecular dynamics simulations of these complexes have increasingly been used to accelerate the identification of better drug candidates. However, the preparation of such simulations is highly complex, and their numerous details are not always adequately emphasized, compromising their reproducibility and reusability. To address this, the ProtCool tool was proposed—a protocol generator focused on integrating docking and molecular dynamics of protein-ligand complexes. In its initial version, this tool was restricted to the user's local environment. This work presents version 2.0 of ProtCool, developed under a client-server model with a web interface. The aim is to fill the gaps left by the previous version, enhancing the software in three fundamental aspects: making it multi-platform, enabling access to multiple users, and making the tool more intuitive. The development of a user-friendly interface allows this new version to expand its scope of use to inexperienced or novice researchers in computational chemistry. ProtCool 2.0 does not execute the dynamics or perform result analyses; it is designed to be an expert in preparation based on workflows, following the programmed workflow and generating all the necessary configuration files for reliable execution of molecular dynamics on the user's computational setup. With the entire process being properly recorded, this allows for greater reproducibility and reusability of the preparations. Many of its functionalities are based on the adaptation of well-known tools from the literature in the field of molecular dynamics simulations. ProtCool 2.0 was developed using best practices and software engineering processes. Its client-server architecture implemented under the web standard enables it to be cross-platform and multi-user, providing benefits in availability and performance. It features a minimalist graphical interface with interactive resources that ensure user safety in correctly filling out their study parameters, thus preventing errors. To demonstrate its use, ProtCool 2.0 underwent a case study on the simulation of the interaction between acetylcholinesterase and galantamine, used in the treatment of Alzheimer's disease, which allowed for validation through replication of a simulation certified by peers in an international publication. The preparation of the simulation successfully enabled the execution of reliable molecular dynamics, reproducing the expected results. It is expected that this tool will not only bring greater speed, reproducibility, and reusability to molecular dynamics preparations but also contribute to smoothing the learning curve for these simulations in computational chemistry.

LISTA DE FIGURAS

Figura 1. Exemplo de execução do MDAnalysis	20
Figura 2. Tela inicial do gRINN.....	22
Figura 3. Exemplos de telas do HeroAnalysis.....	24
Figura 4. Tela principal do MDANSE.	25
Figura 5. Exemplo de execução do HOOMD-blue	26
Figura 6. Exemplo de código de execução do LOOS.	27
Figura 7. Tela principal do QwikMD.....	29
Figura 8. Ciclo de vida de processo evolucionário de software prototipação.....	32
Figura 9. Representação da arquitetura de uma API (Adaptado de ALTEXSOFT, 2020).....	34
Figura 10. Exemplo de um arquivo confidinamica.txt mostrando as principais tags de controle a serem executadas.	36
Figura 11. Representação da arquitetura client-servidor (Adaptado de “Client-Server Model Definition”, 2016).	37
Figura 12. Encadeamento das etapas do ProtCool regidos pelo script ‘Protcool_Dynamic.py’	38
Figura 13. Tela inicial do ProtCool 2.0	39
Figura 14. Arquitetura do ProtCool 2.0.....	40
Figura 15. Tela de opções iniciais da etapa ‘General Info’.....	46
Figura 16. Opção ‘Model’ na tela inicial do ProtCool 2.0.....	47
Figura 17. Tela para informar os parâmetros referentes aos ligantes.....	49
Figura 18. Tela para informar os parâmetros referentes ao <i>docking</i>	52
Figura 19. Demonstração da validação dos campos vazios na tela ‘Docking’	53
Figura 20. Tela para preenchimento dos parâmetros referentes a protonação.....	56
Figura 21. Tela para preenchimento dos parâmetros referentes a solvatação e ionização.....	57
Figura 22. Parte da tela para preenchimento dos parâmetros referentes a simulação. .	60
Figura 23. Parte final da tela para preenchimento dos parâmetros referentes a simulação.....	67
Figura 24. Execução final da preparação da dinâmica molecular.....	67
Figura 25. Preenchimento dos campos da etapa ‘General Info’ (4EY6 - GNT).....	69
Figura 26. Preenchimento dos campos da etapa ‘Ligands’ (4EY6 - GNT).....	70
Figura 27. Preenchimento dos campos da etapa ‘Docking’ (4EY6 - GNT).....	71
Figura 28. Preenchimento dos campos da etapa ‘Protonation’ (4EY6 - GNT).....	72
Figura 29. Preenchimento dos campos da etapa Solvation/Ionization (4EY6 - GNT). 72	
Figura 30. Preenchimento dos campos da etapa ‘Simulation’ (4EY6 - GNT).....	75
Figura 31. Preenchimento de todos os campos de todas as simulações - de 1 a 9 na etapa ‘Simulation’ (4EY6 - GNT).....	79
Figura 32. Arquivos gerados e preparado pelo ProtCool 2.0 para ser realizado a dinâmica (4EY6 - GNT).....	80
Figura 33. Demonstração do processamento do servidor realizando a dinâmica molecular em 24 processadores - saída do comando ‘htop’.....	81
Figura 34. Representação gráfica gerada pelo software VMD da dinâmica molecular já realizada envolvendo a acetilcolinesterase e a galantamina.....	81

LISTA DE TABELAS

Tabela 1. Comparativo das características entre versões.....	31
Tabela 2. Tabela de das etapas do workflow do ProtCool (GUEDES, 2021).....	42

LISTA DE ABREVIATURAS

AMBER	<i>Assisted Model Building with Energy Refinement</i>
API	Interface de Programação de Aplicação (<i>Application Programming Interface</i>)
ASN	Asparagina
BSU	<i>Berkeley Software Distribution</i>
BPTI	<i>Bovine Pancreatic Trypsin Inhibitor</i>
CHARMM	<i>Chemistry at HARvard Molecular Mechanics</i>
COVID-19	<i>Coronavirus Disease 2019</i>
CPU	Unidades de processamento central (<i>Central Processing Units</i>)
CUDA	<i>Compute Unified Device Architecture</i>
GLN	Glutamina
GNU	GNU Não é Unix! (<i>GNU's Not Unix!</i>)
GPL	Licença Pública Geral (<i>General Public License</i>)
GPU	Unidades de processamento gráfico (<i>Graphics Processing Units</i>)
GROMACS	<i>GRoningen MACHine for Chemical Simulations</i>
GTK	<i>GIMP Toolkit</i>
GUI	Interface Gráfica de Usuário (<i>Graphical User Interface</i>)
HD	Alta Resolução (<i>High Definition</i>)
HIS	Histidina
JSON	<i>JavaScript Object Notation</i>
MD	Dinâmica Molecular (<i>Molecular Dynamic</i>)
LAMMPS	<i>Large-scale Atomic/Molecular Massively Parallel Simulator</i>
LOSS	<i>Lightweight Object-Oriented Structure-analysis</i>
MDANSE	<i>Molecular Dynamics and Lattice Dynamics to Analyse Neutron Scattering Experiments</i>
MDS	Simulações de Dinâmica Molecular (<i>Molecular Dynamic Simulations</i>)
MMTK	<i>Memory Management Toolkit</i>
NAMD	<i>Nanoscale Molecular Dynamics</i>
NPH	<i>Constant number (N), pressure (P), and enthalpy (H)</i>
NPT	<i>Constant number (N), pressure (P), and temperature (T)</i>
NVE	<i>Constant number (N), volume (V), and energy (E)</i>
NVT	<i>Constant number (N), volume (V), and temperature (T)</i>
PDB	<i>Protein Data Bank</i>
RCSB	<i>Research Collaboratory for Structural Bioinformatics</i>

RMN	Resonancia Magnetica Nuclear
RMSD	<i>Root Mean Square Deviation</i>
RMSF	<i>Rocky Mountain Spotted Fever</i>
SASA	<i>Solvent-Scccessible Aurface Area</i>
TNS	<i>Theory of Thermal Neutron Scattering</i>
VMD	<i>Visual Molecular Dynamics</i>

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 ESTRUTURA DO TRABALHO	16
2 OBJETIVOS.....	17
2.1 OBJETIVO GERAL.....	17
2.2 OBJETIVOS ESPECÍFICOS	17
3 REVISÃO DA LITERATURA	17
3.1 SOFTWARES DE DINÂMICAS MOLECULARES.....	18
3.2 EVOLUÇÃO DOS SOFTWARES DE SIMULAÇÕES DE MD.....	19
3.2.1 MDAnalysis.....	20
3.2.2 gRINN	21
3.2.3 HeroMDAnalysis.....	22
3.2.4 MDANSE	24
3.2.5 HOOMD-blue.....	25
3.2.6 LOOS.....	26
3.2.7 QwiKMD	28
3.2.8 BioExcel Building Blocks	29
3.2.9 ProtCool 1.0.....	30
4 METODOLOGIA.....	31
4.1 DESENVOLVIMENTO DO PROTCOOL 2.0.....	31
4.1.1 Etapas do Processo de Software	32
4.2 IMPLEMENTAÇÃO DO PROTCOOL 2.0.....	34
4.2.1 Backend	34
4.2.2 Frontend.....	38
4.2.3 Estrutura de pastas e arquivos de saída	40
4.2.4 Logs	41
5 FUNCIONALIDADES DO PROTCOOL 2.0	41
5.1 WORKFLOWS	41
5.2 FUNCIONALIDADES	44
5.2.1 Etapa General Info.....	45
5.2.2 Etapa Ligands	48
5.2.3 Etapa Docking	50
5.2.4 Etapa Protonation	53
5.2.5 Etapa Solvation/Ionization	56
5.2.6 Simulation.....	57
6 ESTUDO DE CASO	68

6.1 REALIZANDO A DINAMICA	80
7 CONCLUSÃO.....	82
REFERÊNCIAS	83

1 INTRODUÇÃO

Desde as primeiras simulações com intuito de estudar o comportamento dinâmico de proteína globular realizado por (MCCAMMON, 1977), já se sabia que o comportamento real dos átomos e suas principais partículas eram regidos pelas leis da química quântica. Porém, devido à sua alta complexidade, principalmente dos sistemas multiátomo, observou-se que o tratamento desses sistemas, incluindo os sistemas biológicos, são extremamente difíceis de se resolver computacionalmente.

Entretanto, uma alternativa seria a possibilidade de aproximar as interações quânticas utilizando campos de força. Os campos de força são combinações de fórmulas matemáticas e parâmetros associados usados para descrever a energia da proteína em função de suas coordenadas atômicas, bem como uma combinação de van der Waals, ligações de hidrogênio e forças Coulômbicas para interações não covalentes (GUVENCH, 2008). Para compor um ambiente realista e convincente de comportamentos dinâmicos, faz-se necessário ainda que o modelo computacional proposto obedeça às leis de movimento de Newton e da termodinâmica de Boltzmann (BRINI, 2020).

De posse dessas informações, Karplus et al em um feito histórico concedeu movimento realista a centenas de átomos de BPTI (*Bovine Pancreatic Trypsin Inhibitor*) em aproximadamente 1 dúzia de picos (picossegundos) de tempo de simulação (MCCAMMON, 1977). Atualmente, com a evolução da química computacional e da supercomputação é possível realizar simulações fidedignas de milhões de átomos em escalas de micro ou milissegundos contemplando proteínas inteiras ou grandes complexos proteicos em solventes explícitos ou embutidos em bicamadas lipídicas (BRINI, 2020).

Se por um lado os enormes avanços da química computacional, em teorias, algoritmos e hardware, resultaram numa maior abrangência, eficiência e acurácia dos sistemas simulados, por outro lado, o planejamento, preparação, execução, análise e reprodução de tais simulações tornaram-se bem mais complexos (Purawat et al, 2017). As dificuldades advêm principalmente dos inúmeros detalhes, parâmetros e passos exigidos pelos protocolos de simulação, quase sempre insuficientemente relatados na metodologia dos artigos publicados. Tudo isso contribui para tornar mais tortuosa e íngreme a curva de aprendizagem das técnicas e experimentos utilizados em simulações de dinâmica molecular.

Essas dificuldades afetam especialmente a reprodutibilidade dos experimentos. A reprodutibilidade na computação entre outras áreas consiste em obter resultados válidos usando a

mesma fonte de entrada de dados, passos computacionais, métodos e códigos para o mesmo ambiente de análise (*National Academies of Sciences, Engineering, and Medicine and others*, 2019).

Atualmente, a execução dos procedimentos geralmente é feita por meio de scripts que necessitam por diversas vezes serem operados manualmente tornando o processo árduo e difícil de entender, principalmente por aqueles que não participaram de suas implementações e uso. Dessa forma, lidar com particularidades computacionais como: parâmetros de sistemas operacionais, compiladores, bibliotecas e suas dependências exige do usuário um alto nível de conhecimento prévio e técnico que foge dos limites de sua área de atuação se tornando, portanto, uma grande barreira à viabilidade da pesquisa. Além disso, existe falta de detalhes padronizados de como os fluxos de trabalho são testados, selecionados, executados, a sequência exigida, número de repetições e combinações de parâmetros (PURAWAT, 2017). É percebido que mesmo os próprios autores podem se confundir sob esse emaranhado de detalhes e parâmetros exigidos. Conseqüentemente, ocorre a perda da habilidade de conseguirem reproduzir seus próprios experimentos devida à falta de padronização e não possuir um método amigável e intuitivo, sem contar com poucos detalhes nas documentações disponibilizadas.

Um aspecto importante que deve ser considerado é que todo o complexo processo de preparação das simulações pode se tornar bem desafiante, principalmente para pesquisadores inexperientes e iniciantes. Lidar com programação, processos e ferramentas especializadas requerida por boa parte das simulações de dinâmicas moleculares não é uma tarefa trivial. Além do mais, os protocolos de dinâmicas moleculares não são simples do ponto de vista da sua curva de aprendizado, no qual exige do pesquisador anos de dedicação para dominar todos os detalhes envolvidos, relações e conceitos.

Nesse sentido, enquanto alguns pesquisadores iniciantes podem ser desencorajados a aprender MD (*Molecular Dynamic*), outros por sua vez tornam-se impacientes e temerários em subestimar as dificuldades inerentes ao processo e cometem erros que nem sempre são detectados de maneira simples. Grande parte dessas constatações são observadas a partir de nossa experiência profissional trabalhando com simulações em MD há mais de 10 anos e por meio de troca de informações com outros pesquisadores desta área.

Com intuito de amenizar o problema, percebe-se que ferramentas computacionais intuitivas que simulam ambientes reais facilitam o estudo da química em biologia (TATLI, 2010). A utilização e o desenvolvimento de softwares computacionais na educação têm se tornado cada vez mais comuns como forma de tornar o conhecimento mais acessível aos estudantes, isso graças a grande

interatividade que os recursos de multimídia proporcionam. As pesquisas que envolvem a implantação de softwares que auxiliam o ensino de química e biologia têm mostrado que uma grande quantidade de alunos fica mais interessado e satisfeita com seu aprendizado com essas inovações (PENG, 2019).

A atual pandemia de COVID-19 aumentou a demanda por sistemas que simplifiquem, agilizem e facilitem a complexidade desses processos, sem perder a confiabilidade (PARKS, 2020). Ela ajudou aumentar ainda mais o interesse por ferramentas e softwares para o ensino, dado que boa parte da educação mundial precisou migrar para o ensino remoto como sua única condição. O resultado foi uma repentina transição do modelo presencial para os encontros on-line, realizados pela internet por meio do uso de aplicativos de comunicação (TIGAA, 2020). Existem vários softwares que possuem o objetivo de quebrar as barreiras no ensino da química e biologia, no qual possuem funcionalidades generalistas com foco principal em interatividade para alunos de ensino médio e fundamental (POLIK, 2022).

Com a dupla intenção de facilitar o preparo das simulações para pesquisadores experientes e agregar ainda mais a gama de ferramentas que auxiliam o ensino, criamos o ProtCool 2.0, um gerador de protocolo amigável para simulações MD clássicas de complexos proteína-ligante, com compatibilidade testada e validada para NAMD 2.11 (PHILLIPS, 2007). O foco principal é possibilitar pesquisas e estudos na integração do *docking* molecular com o MD clássico, campos que possuem alto impacto no processo de descoberta de fármacos (PARKS, 2020). Entretanto, não executa MD nem gera análises estatísticas. Essas tarefas são de responsabilidade do usuário de acordo com sua conveniência e infraestrutura computacional.

Assim, na atual versão, o ProtCool 2.0 não é uma suíte completa de simulações moleculares. Sua palavra-chave é **preparação**: percorre todas as etapas de preparação, gerando os arquivos necessários para uma execução adequada da MD. Cada etapa registra um log detalhado para que o pesquisador possa verificar a consistência de todas as preparações parciais. Isso dá aos pesquisadores agilidade e flexibilidade para executar simulações confiáveis com adaptações mínimas por conta própria.

Uma vez preparados, executados, validados e salvos os fluxos de trabalho, todo o processo é devidamente registrado no ProtCool 2.0. Dessa forma, implica em reprodutibilidade, pois a simulação pode ser perfeitamente reexecutada por qualquer outro grupo de pesquisa, bastando compartilhar os arquivos necessários. Também implica em reutilização, já que adaptar protocolos antigos a outro conjunto de ligantes e alvos é uma tarefa relativamente simples de implementar com o ProtCool 2.0.

Como parte das estratégias de engenharia de software, o ProtCool foi intencionalmente projetado para ter um escopo inicial menor (mas bem-feito) para crescer de forma consistente. Nesse sentido, há um número limitado de opções prontas para uso em cada etapa dos fluxos de trabalho, mas são as escolhas mais comuns nos protocolos de MD.

Desde sua primeira versão publicada no trabalho de (GUEDES, 2021), o ProtCool passou por diversas atualizações e melhorias. Na versão anterior o objetivo inicial era oferecer um *workflow* automatizado visando a unificação de várias ferramentas na execução de todo o processo (GUEDES, 2021). A estrutura do ProtCool foi projetada para conter vários softwares instalados na máquina do pesquisador. Por meio de scripts obedecendo um *workflow* proposto, dispara sequencialmente todas as tarefas e etapas exigidas no processamento para preparação da MD. Assim sendo, a entrada dos dados era realizada por preenchimento manual de arquivo “txt” de forma criteriosa, e após o preenchimento eram executados scripts diversos de busca de arquivos PDB em comunicação com API 's externas. Também realizava tratamentos de arquivos e execução dos softwares especialistas para cada etapa. Para facilitar a entrada de dados, foi criada uma interface preliminar, que basicamente apresentava um formulário web preenchendo os campos exigidos. Porém, essa versão deixava a desejar nos elementos de interatividade e dinamismo tão necessários nos aspectos de usabilidade. Operações mais avançadas de execução, aconteciam diretamente por meio de consoles, o que exigia do pesquisador conhecimento prévio no uso do sistema operacional.

Outra questão que se pode destacar é referente a infraestrutura, conforme dito anteriormente, o ProtCool era um pacote de vários softwares especialistas gerenciados por um workflow por meio de scripts que eram instalados na máquina dos pesquisadores. A grande desvantagem dessa arquitetura é que toda responsabilidade do processamento e do hardware em questão fica com o usuário. Com isso, toda a escalabilidade, ajustes, atualizações e manutenções se tornam difíceis ou inviáveis. Os conceitos de reprodutibilidade também são afetados, pois experiências executadas em um computador podem gerar resultados distintos se executados em outro computador partindo dos mesmos dados de entrada, devido alguns detalhes de ambientes diferentes. Entretanto, o ProtCool alcançou sucesso em sua proposta inicial, principalmente quanto à criação da estrutura de *workflows* para preparação em MD, no qual foi totalmente mantida por se tratar do núcleo do sistema, com pequenas atualizações.

Na versão atualizada, apresentada neste trabalho, o ProtCool 2.0 busca atender 3 questões fundamentais: ser multiusuário, multiplataforma e oferecer melhor usabilidade. Multiusuário porque toda sua estrutura foi migrada para o padrão web rodando sob a arquitetura *client-server* (BERSON,

1996). Ou seja, o pesquisador não precisa instalar mais nada em sua máquina, basta apenas acessar um endereço na internet e imediatamente começar sua pesquisa. Dessa forma, mitigamos a questão de manutenção e reprodutibilidade, no qual todas as atualizações da ferramenta são percebidas por todos os usuários no mesmo instante. Multiplataforma porque seu desenvolvimento segue o padrão web de responsividade, o pesquisador pode acessar o ProtCool 2.0 de qualquer dispositivo conectado à internet e que possua um navegador. A responsividade permite utilizar a ferramenta com mínimos impacto no seu uso, independentemente do dispositivo. Usabilidade porque a interface de preenchimento de parâmetros para a preparação da MD foi construída para ser minimalista, versátil e interativa. Ela conversa com o pesquisador com intuito de não permitir que campos essenciais possam ser preenchidos de forma errada, ou que campos importantes possam estar vazios, garantindo ao *backend* informações previamente checadas e validadas. A usabilidade garante ao usuário, mesmo aos iniciantes, segurança e conformidade para reproduzir sua pesquisa, evitando perda de tempo desnecessária com execução de comandos de sistema operacional entre outras questões, consequentemente permitindo agilidade e foco total no objeto de pesquisa.

Como estudo de caso de uso do ProtCool 2.0, é mostrada a preparação de um protocolo de ancoragem e dinâmica molecular para o complexo envolvendo a proteína acetilcolinesterase (4EY6) com o ligante galantamina (GNT), utilizado no tratamento do Alzheimer (Rocha, 2017). A galantamina liga-se à acetilcolinesterase, inibindo sua ação e elevando os níveis do neurotransmissor acetilcolina no cérebro, com efeitos positivos na capacidade cognitiva e de memória em pacientes com Alzheimer. Em (Rocha, 2017), foi feita uma simulação de dinâmica molecular desse complexo, cuja etapa preparatória envolveu scripts ad hoc dos próprios autores do artigo que culminaram num complexo protocolo de 21 passos. Será demonstrado como montar esse mesmo workflow utilizando o ProtCool.

1.1 ESTRUTURA DO TRABALHO

Esta dissertação de mestrado está organizada em seis capítulos. Além do capítulo 1 que compõe esta introdução, tem-se:

- O Capítulo 2: apresenta os objetivos de trabalho.
- O Capítulo 3: uma revisão de literatura sobre o tema abordado.
- O Capítulo 4: são apresentadas as metodologias e métodos aplicados na condução da pesquisa.

- O Capítulo 5: descreve o estudo de caso envolvendo o complexo acetilcolinesterase e galantamina.
- Capítulo 6: apresenta conclusão e estudos futuros.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Desenvolver e validar uma arquitetura cliente/servidor com *frontend* web para o ProtCool, um gerador de protocolos de simulações de dinâmica molecular em complexos proteína-ligante

2.2 OBJETIVOS ESPECÍFICOS

1. Adequar o *backend* da 1ª versão para uma arquitetura cliente/servidor.
2. Montar e configurar um servidor *backend* web.
3. Projetar e implementar uma interface *frontend* web.
4. Promover a integração *frontend* e *backend*.
5. Testar e validar a nova arquitetura, com um estudo de caso envolvendo a acetilcolinesterase e a galantamina.

3 REVISÃO DA LITERATURA

Simulações de dinâmica molecular têm um profundo impacto na biologia celular molecular e na biomedicina. Ela perpassa por questões que envolvem desde as propriedades físicas e químicas ao desenvolvimento de fármacos até a fabricação de novos biomateriais e criação de fontes de energia renováveis de base biológica. Assim sendo, ela é fundamental para o melhor entendimento de organismos vivos sendo uma das principais ferramentas de pesquisa para atacar problemas biomoleculares (PERILLA, 2015).

Utilizando recursos computacionais é possível realizar a simulação das interações e comportamentos das dinâmicas de moléculas em nível atômico. Portanto, torna-se uma ferramenta computacional poderosa para ser usada em estudos que complementam as análises experimentais de bancada *in vitro* e *in vivo* (ARANDA-GARCIA, 2022).

Tem como objetivo principal investigar o amplo espectro de fenômenos que inclui o enovelamento de proteínas, desenvolvimento de fármacos e propriedades de materiais biológicos (SALO-AHEN, 2020) (ALDER, 1959).

Uma das principais vantagens das simulações em MD é sua habilidade de oferecer detalhes atômicos que não são acessíveis por técnicas experimentais. A simulação pode revelar os mecanismos moleculares principais que governam o comportamento bioquímico, e oferecer percepções nas relações das estruturas. Além disso, as simulações em MD permitem aos pesquisadores explorar as propriedades em condições estritas que são difíceis ou impossíveis de alcançar experimentalmente (RAPAPORT, 2004).

Portanto, simulações em MD são uma ferramenta valiosa para estudo de comportamentos moleculares que têm contribuído bastante no entendimento de um largo espectro de fenômenos (Brini & Dill, 2020). Com o poder computacional em contínuo crescimento, o escopo e a eficiência das simulações em MD tendem a expandir ainda mais, trazendo novas oportunidades e possibilidades em descobertas científicas.

3.1 SOFTWARES DE DINÂMICAS MOLECULARES

Como dito anteriormente, os softwares de simulação de dinâmicas moleculares são largamente utilizados no campo da computação e química para simular os comportamentos de moléculas em nível atômico.

Há um vasto escopo de softwares de simulações de MD disponíveis, em que cada um possui suas próprias características, vantagens e limitações. Alguns dos mais populares e conhecidos pacotes de software, temos: GROMACS, AMBER, NAMD, CHARMM e LAMMPS.

- GROMACS (***G**ROningen **M**Achine for **C**hemical **S**imulations*) é um pacote de software largamente usado em simulações de MD muito conhecido por sua velocidade e escalabilidade. Ele é particularmente adequado para simulações em larga escala de biomoléculas e tem sido usado de forma abrangente em diversas aplicações, que inclui o enovelamento de proteínas e desenvolvimento de fármacos (HESS, 2008).
- AMBER (***A**ssisted **M**odel **B**uilding with **E**nergy **R**efinement*) é outro pacote de software de simulação de MD que é muito usado. Ele é conhecido principalmente por sua acurácia preditiva e flexibilidade. Tem sido utilizado em várias aplicações e estudos que envolvendo complexos proteínas-ligantes (CASE, 2005).

- NAMD (*NANoscale Molecular Dynamics*) é um pacote de software desenvolvido especificamente para simulações de grandes sistemas biomoleculares. Sua principal característica é sua escalabilidade e performance em arquiteturas de computação paralela, ele tem sido usado em boa parte das ferramentas e aplicações para estudos de membranas de proteínas e canais de íons (PHILLIPS, 2007).
- CHARMM (*Chemistry at HARvard Molecular Mechanics*) este sistema foi desenvolvido na Universidade de Harvard. Conhecido também por ter boa capacidade preditiva e versatilidade. Também inclui funcionalidades referentes as interações proteínas-proteínas (BROOKS, 1983).
- LAMMPS (*Large-scale Atomic/Molecular Massively Parallel Simulator*) é um pacote de software de simulações de MD desenvolvido pela *Sandia National Laboratories*. Possui versatilidade e alta performance na utilização de arquitetura de computação paralela. Como sua principal utilização destaca-se os estudos de polímeros, ciências de materiais e química de superfície (PLIMPTON, 1995).

Portanto, os softwares de simulações de dinâmicas moleculares desempenham um papel crucial no campo da química computacional e por meio deles vários pesquisadores estão aptos a escolher entre as opções que mais lhe agradam e melhor atendem no objetivo de pesquisa. Outro fator, é que geralmente todos os pacotes de software são livres ou de código-fonte aberto, o que permite que possam ser adaptados e até mesmo serem evoluídos por diversos pesquisadores. Atualmente, há uma gama de softwares que derivam desses, oferecendo mais velocidade, capacidade preditiva e robustez, contemplando diversas funcionalidades que auxiliam ainda mais os pesquisadores, tais como facilidade no uso, interfaces gráficas intuitivas, adequação do seu uso em supercomputadores, clusters de máquinas, computação distribuída, etc. (GALEAZZI, 2009).

3.2 EVOLUÇÃO DOS SOFTWARES DE SIMULAÇÕES DE MD

Ao realizar a revisão da literatura, com o intuito de encontrar ferramentas similares ao ProtCool 2.0, não foi encontrada nenhuma com mesma generalidade de escopo. Porém, encontrou-se diversas ferramentas que de alguma forma buscam o mesmo objetivo, que é desenvolver e evoluir ferramentas em MD com foco em facilidade, intuitividade e economicidade nos processos que

envolvem MD. É notório que mesmo que as ferramentas encontradas não cubram o mesmo escopo do ProtCool 2.0, percebe-se que há muitas interseções de funcionalidades com várias delas. A seguir é apresentado algumas ferramentas relevantes encontradas na literatura que possuem propósito similar ao ProtCool 2.0.

3.2.1 MDAnalysis

O MDAnalysis (<http://mdanalysis.org>) é uma biblioteca que foi desenvolvida para realizar análises estruturais e temporais de trajetórias dinâmica molecular e de estrutura de proteínas individuais (GOWERS, 2016). Teve seu início inspirado pelo MDTools para Python e MMTK (HINSEN, 2000). Geralmente as simulações são realizadas utilizando de pacotes de software altamente otimizados para serem executados em computadores de alto desempenho. Entretanto, um problema encontrado nessa opção é a falta de padronização em que a maioria dos códigos gerados possuem seus próprios formatos, não alcançando a universalidade de uso na comunidade.

```
In [1]: import nglview as nv
import MDAnalysis as mda
from MDAnalysisTests.datafiles import PSF, DCD

In [2]: u = mda.Universe(PSF, DCD)
protein = u.select_atoms('protein')

In [4]: w = nv.show_mdanalysis(protein)
w
```

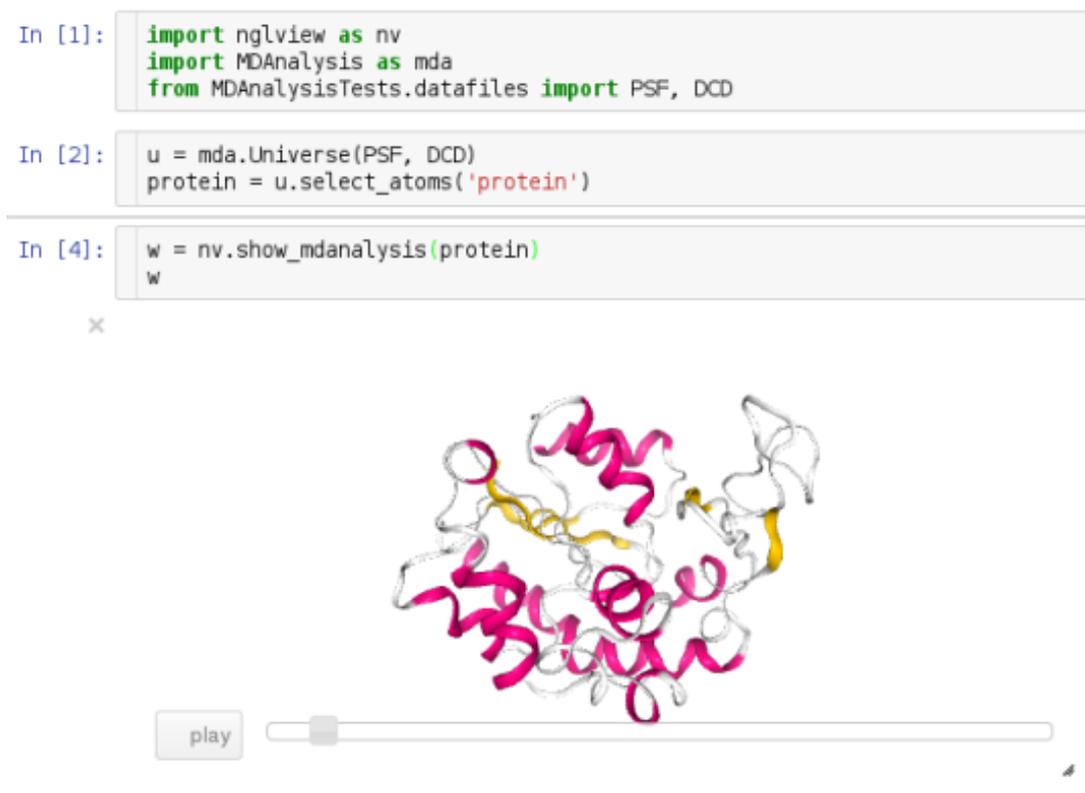


Figura 1. Exemplo de execução do MDAnalysis

A fim de abordar esse problema o MDAnalysis foi criado com objetivo de abstrair o acesso aos dados brutos da simulação e apresentar uma interface Python orientada a objeto uniforme para facilitar a vida do usuário. Isso permite que os usuários ganhem agilidade no desenvolvimento por meio de códigos portáteis e funcionais, podendo ser usados por todas as comunidades de simulação biomolecular como base de seus projetos. Ela apresenta uma interface de usuário com design modular que permite trabalhar muito bem com scripts e workflows. Além disso, sua dinâmica e interatividade permite usufruir da prototipação utilizando ferramentas como iPython/Jupyter Notebook, possui interação com a biblioteca nglview que fornece visualização molecular e com Pandas para trabalhar com análise de séries temporais.

Por ser escrito em Python e Cython também usa vetores NumPy que são largamente conhecidos pela comunidade científica, essas características permitem que a biblioteca sirva de base para outras ferramentas sendo regida pela GNU *General Public License v2*.

3.2.2 gRINN

O gRINN é uma ferramenta que visa identificar as interações entre os resíduos de aminoácidos individuais e o papel desempenhado por cada um deles no contexto das simulações de dinâmica molecular (SERÇINOĞLU, 2018). Ela possui tanto uma interface gráfica como linha de comando. Realiza a geração e análise de energias de interação de emparelhamento de resíduos e correlações de energia de trajetórias em simulações de MD. O gRINN suporta trajetórias em simulações geradas pelo software GROMACS (VAN DER SPOEL, 2005), NAMD (NELSON, 1986) ou CHARMM (BROOKS, 1983). As tarefas são executadas por scripts por meio de workflows simplificando bastante o trabalho para o usuário final, com exposição limitada pelos passos do sistema. Os cálculos de energia de interação de resíduos são completamente automatizados utilizando os softwares acima, permitindo que pesquisadores mantenham o foco em sua pesquisa. Além disso, ele pode computar correlações entre as séries de energia de interação também.

O gRINN é designado para análise pós-simulação, ele é direcionado a usuários que já concluíram uma simulação MD de uma proteína usando NAMD ou GROMACS. Portanto, NAMD ou GROMACS devem ser pré-instalados no sistema como pré-requisito.

Sua GUI possui um visualizador molecular embutido, em que fornece a visualização de séries temporais de energia de interação, distribuições, uma matriz de energia de interação, correlações de energia de interação e uma matriz de correlação de resíduos. Ele também oferece construção e análise de Rede de Energia Proteica, fornecendo métricas baseadas em resíduos, como graus, centralidades

intermediárias, centralidades de proximidade, bem como análise de caminho mais curto. O gRINN foi desenvolvido na linguagem de programação Python (versão 2.7) sendo gratuito e aberto a todos os usuários sem necessidade de login (SERÇINOĞLU, 2018).

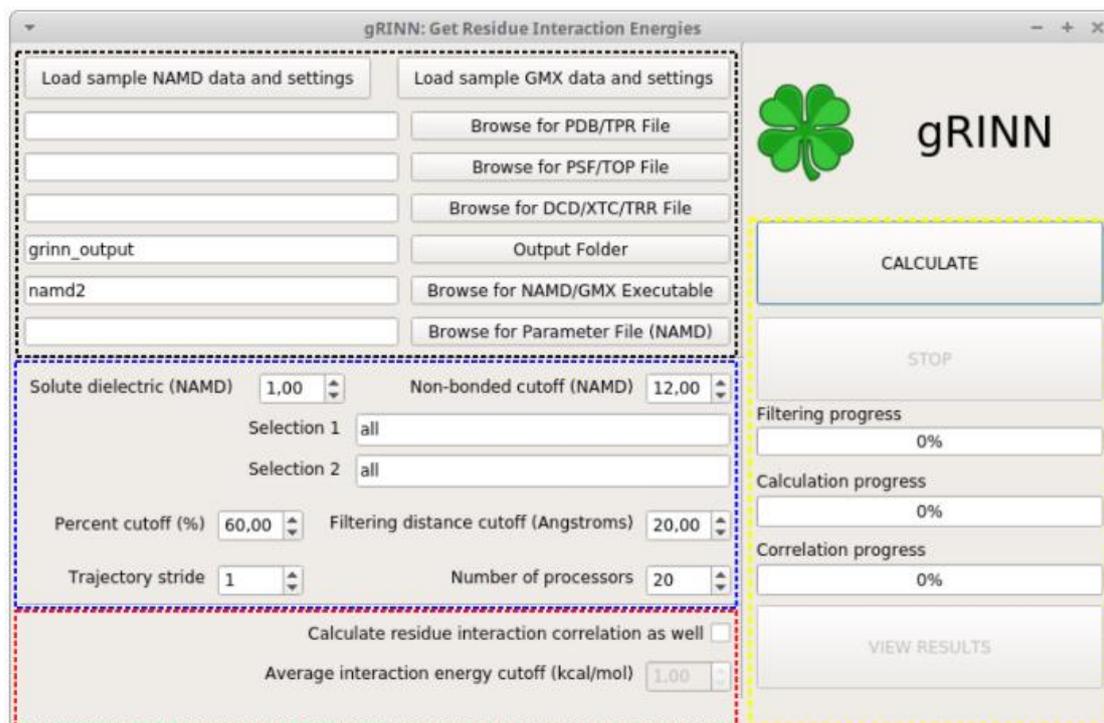


Figura 2. Tela inicial do gRINN.

3.2.3 HeroMDAnalysis

Um projeto chamado HeroMDAnalysis desenvolvido em 2021 baseado no GROMACS (VAN DER SPOEL, 2005), serve para analisar as trajetórias de MDS e gerar gráficos como imagens de alta resolução para vários parâmetros. O intuito foi herdar as características de popularidade e velocidade de execução do GROMACS. Ele foi construído usando a programação *bash shell* e a interface gráfica do usuário foi construída usando o framework Zenity¹. Além disso, umas das preocupações dos desenvolvedores deste projeto era de criar uma interface mais acessível para químicos sem experiência no uso de linha de comando, aos quais enfrentam dificuldades na análise dessas simulações. Segundo seus autores, a interface de linha de comando costuma ser um processo complexo, manual e sujeito a erros tornando-se uma tarefa difícil.

¹ <https://wiki.gnome.org/Projects/Zenity>

Portanto, o HeroMDAnalysis, foi desenvolvido para ser um sistema simples, semiautomático, relativamente rápido e fácil na realização de análises de simulação MD (RAWAT, 2021). Ele foi escrito usando um editor de texto padrão na linguagem de programação *bash shell* (RAWAT, 2021).

O Zenity¹ é um software livre e um programa multiplataforma disponível por padrão no ambiente Linux que permite a execução de caixas de diálogo GTK em linha de comando e scripts de *shell* (RAWAT, 2021). Dentre suas opções de uso, o sistema é capaz de fornecer diversos parâmetros envolvendo complexos de proteína-ligante, tais como: RMSD, SASA, energia livre de ligante de solvatação, contatos ligante-proteína, ligação de H, etc.

Depois de coletar a lista desejada de parâmetros do usuário, a ferramenta executa os comandos de análise GROMACS *gmx* para realizar os ajustes necessários antes de iniciar qualquer análise estatística. Após é realizada a compactação do arquivo de trajetória. Durante a análise dos parâmetros, é exibida uma barra de tarefas que indica o progresso ao usuário. Dessa forma, os arquivos de dados brutos são organizados em estruturas de arquivos e pastas usando os comandos de manipulação de arquivos e diretórios nativos do Linux. Em seguida, as caixas de diálogo são criadas para solicitar ao usuário opções para recusar ou aceitar plotar os arquivos de dados brutos em arquivos de imagem de qualidade HD. Se o usuário recusar, os dados brutos serão armazenados com segurança e estarão disponíveis para uso posterior. O programa termina com sucesso mostrando a tela de saída (RAWAT, 2021). Atualmente, a ferramenta suporta aproximadamente 52 parâmetros pertencentes a várias categorias a partir de uma trajetória de simulação MD. Ela é distribuída gratuitamente através da web <http://heromdanalysis.wordpress.com>, lá pode-se encontrar mais informações como: compatibilidade, guia passo a passo, guia de instalação e tutorial de uso (RAWAT, 2021).

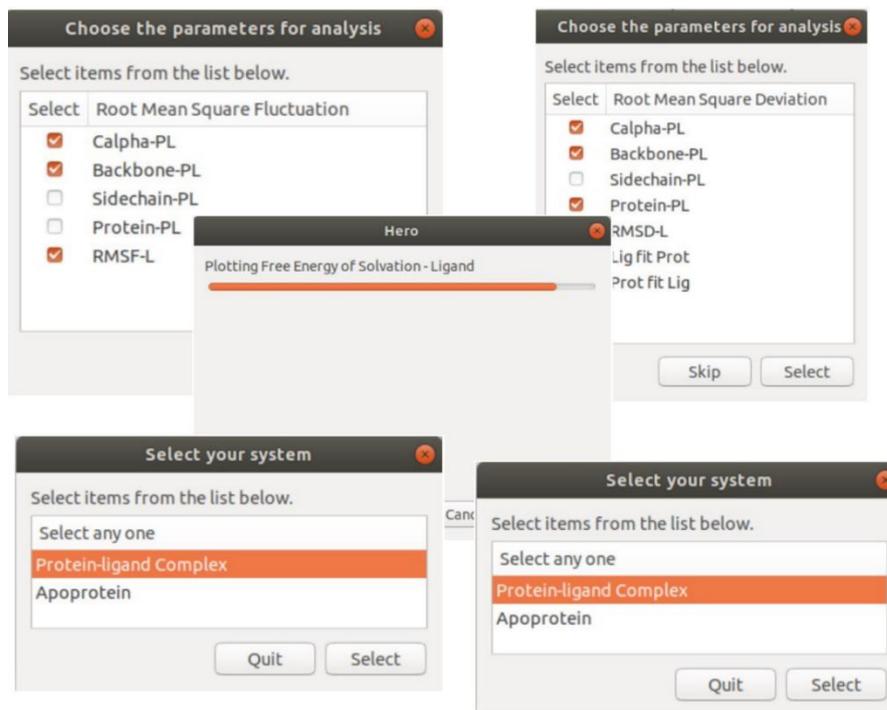


Figura 3. Exemplos de telas do HeroAnalysis.

3.2.4 MDANSE

Ainda no mesmo objetivo de facilitar o uso de software para MD, temos o MDANSE um aplicativo Python para pós-processamento de simulações MD (GORET, 2017). Ele concentra em fornecer ao usuário uma interface gráfica amigável (GUI) permitindo que as análises sejam iniciadas, seu foco operacional está na utilização da técnica experimental TNS (*theory of thermal neutron scattering*) (SQUIRES, 1996).

A ferramenta pretende abranger todos os tipos de simulações MD, cobrindo todo o espectro de sistemas abordados pelo TNS. Uma das principais características do MDANSE é a interface gráfica que permite visualizar uma trajetória, interagir com ela antes de executar uma análise ou exibir dados 2D ou 3D resultantes de análises de forma amigável e integrada. Em sua versão atual houve a inclusão do visualizador molecular para facilitar análises e permitir análises qualitativas como: trajetória molecular, o nível de sofisticação de plotagem e gestão dos cálculos simultâneos utilizados na execução e que são iniciados a partir da interface. Sua arquitetura possui um *logger* do aplicativo por meio de um console Python que dá acesso à linha de comando para a estrutura MDANSE interativamente, além de possuir um gerenciador de tarefas, mostrando o status dos trabalhos em execução. Ele permite que várias tarefas possam ser executadas simultaneamente e mesmo independentes da interface sendo executadas em segundo plano.

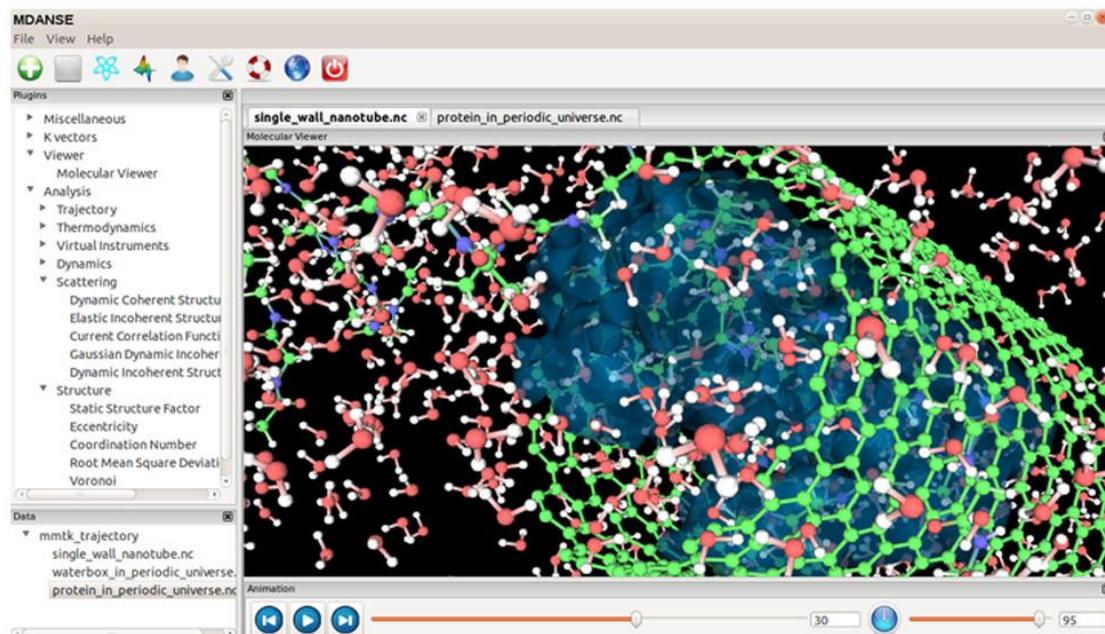


Figura 4. Tela principal do MDANSE.

Outra particularidade é que suas análises são frequentemente realizadas em átomos em vez de todo o sistema, pode ser muito prático a capacidade de selecionar os átomos interativamente com o mouse. Essa funcionalidade permite também a gravação da seleção em um *script* Python que pode ser editado se necessário e reutilizado. Dessa forma, permite a conveniência de o usuário realizar a sobreposição a trajetória de uma ou mais moléculas em todo o modelo com intuito de entender melhor as vias de difusão antes ou após a execução.

3.2.5 HOOMD-blue

O HOOMD-blue é um mecanismo de simulação de partículas projetado para dinâmica molecular em escala nano e coloidal, e de simulações de Monte Carlo (MOONEY, 1997). Desenvolvido em março de 2007 pela Universidade do Estado de Iowa - EUA, está disponível desde agosto de 2008 (ANDERSON, 2020). HOOMD-blue é um pacote Python de alta performance que trabalha em *backend* com C++/CUDA para otimização de GPU. A interface Python permite que os usuários combinem HOOMD-blue com outros pacotes no ecossistema Python para criar simulação e fluxos de trabalho de análise em MD. O objetivo do seu desenvolvimento é obter uma ferramenta única de uso geral em MD e simulações biomoleculares. A fim de alcançar esse objetivo a implementação desse pacote Python permite fácil interoperabilidade com outras ferramentas do ecossistema científico. Sua interface de alto nível abstrai um *backend* de alto desempenho que executa

simulações em uma ou várias GPUs ou CPUs.

O HOOMD-blue possui integradores MD para muitos conjuntos termodinâmicos diferentes, incluindo NVE, NVT, NPH, NPT, dinâmica de Langevin, dinâmica Browniana e dinâmica de partículas dissipativas, e também suporta a minimização de *FIRE energy* (BITZEK, 2006). HOOMD-blue também implementa o solvente de dinâmica de colisão de modelo multi-partícula (HOWARD, 2018).

Seu desenvolvimento teve rígidas observações de boas práticas de engenharia de software como: desenvolvimento, teste, manutenção e escalabilidade, principalmente visando a colaboração de outros desenvolvedores. Em seu histórico de versionamento, o projeto já contou com mais de 10.000 *commits* por mais de 60 desenvolvedores.

HOOMD-blue está disponível em código aberto sob a licença BSD. Para implementar mudanças, os usuários podem baixar o código HOOMD-blue e adicionar novas funcionalidades diretamente, ou criar um plug-in em um repositório de código separado e vincular ao HOOMD-blue em tempo de construção. Muitos usuários publicaram artigos e estruturas de software usando HOOMD-blue com extensões que desenvolveram (ERVIK, 2017).

```

1 from hoomd import *
2 from hoomd import hpmc
3 # place particles
4 context.initialize('--mode=cpu')
5 unitcell=lattice.sc(a=1.2, type_name='A')
6 system = init.create_lattice(unitcell, n=7)
7 # hard particle Monte Carlo
8 mc = hpmc.integrate.convex_polyhedron(
9     d=0.1, a=0.1, seed=2)
10 cube_verts = \
11     [[-0.5,-0.5,-0.5], [0.5,-0.5,-0.5],
12      [-0.5,-0.5, 0.5], [0.5,-0.5, 0.5],
13      [-0.5, 0.5,-0.5], [0.5, 0.5,-0.5],
14      [-0.5, 0.5, 0.5], [0.5, 0.5, 0.5]]
15 mc.shape_param.set('A',
16     vertices=cube_verts)
17 # run the simulation
18 run(50e3)

```

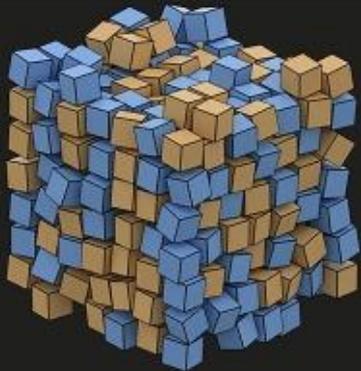


Figura 5. Exemplo de execução do HOOMD-blue

3.2.6 LOOS

LOOS (*Lightweight Object-Oriented Structure-analysis*) é uma biblioteca C++ projetada para facilitar criação de novas ferramentas para analisar simulações de dinâmica molecular, seu foco é eliminar tarefas repetitivas, permitindo que os desenvolvedores se concentrem na parte cientificamente relevante do problema. LOOS suporta entrada usando os formatos de arquivo nativos dos pacotes de simulação biomolecular mais comuns, incluindo CHARMM, NAMD, Amber, Tinker

e GROMACS. Além disso, O LOOS vem com mais de 120 ferramentas pré-criadas, incluindo suítes de ferramentas para análise de simulação convergente, histogramas 3D e modelos de redes elásticas.

```
#!/usr/bin/env python3
import sys
import loos
import loos.pyloos
import math
header = " ".join(sys.argv)
print("# ", header)
system_file = sys.argv[1]
traj_file = sys.argv[2]
sel_string1 = sys.argv[3]
sel_string2 = sys.argv[4]
# create the system and trajectory
system = loos.createSystem(system_file)
traj = loos.pyloos.Trajectory(traj_file, system)
# apply selections to get atoms
sel1 = loos.selectAtoms(system, sel_string1)
sel2 = loos.selectAtoms(system, sel_string2)
for frame in traj:
    # compute distance
    centroid1 = sel1.centroid()
    centroid2 = sel2.centroid()
    diff = centroid2 - centroid1
    distance = diff.length()
    # compute angle between principal axes
    vectors1 = sel1.principalAxes()
    axis1 = vectors1[0]
    vectors2 = sel2.principalAxes()
    axis2 = vectors2[0]
    angle = math.acos(axis1 * axis2) * 180/math.pi
    # compute torsion between principal axes
    p1 = centroid1 + axis1
    p2 = centroid2 + axis2
    tors = loos.torsion(p1, centroid1, centroid2, p2)
    # write output
    print(traj.index(), distance, angle, tors)
```

Figura 6. Exemplo de código de execução do LOOS.

O LOOS é distribuído gratuitamente como código-fonte sob a licença GNU GPLv3 ² através do Source Forge (<http://loos.sourceforge.net>). A compatibilidade de LOOS é testada na maioria dos

² <http://www.gnu.org/licenses/gpl.html>

principais Linux distribuições, incluindo Debian, Ubuntu e Fedora, juntamente com várias versões de cada distribuição. Além disso, o MacOS é compatível, assim como o Windows (via cygwin).

Segundo os autores (ROMO, 2014), LOOS não é apenas uma biblioteca, ela contempla uma série de ferramentas que têm aplicabilidade geral que são consideradas úteis para pesquisa. O intuito dela é combinar ferramentas que são voltadas para uma única tarefa, serem unificadas por meio de um workflow em um sistema de script como o *shell* Unix ou Python. Diferentemente de programas monolíticos, como CHARMM e VMD³, as ferramentas LOOS são, de alguma forma, uma continuação do “Unix Filosofia” de kit ferramentas focadas em uma única tarefa.

3.2.7 QwikMD

A QwikMD, que é uma ferramenta que possibilita a preparação, execução e análise de dinâmicas moleculares, utilizando para isso os sistemas VMD e NAMD (GUEDES, 2021). Ele trabalha de forma semelhante ao ProtCool 2.0 no que se refere ao *workflow* da dinâmica. Entretanto, o QwikMD realiza apenas preparação de proteínas, enquanto o ProtCool 2.0 também realiza a preparação de proteínas complexadas a ligantes.

O principal objetivo do QwikMD é permitir que novatos e especialistas abordem questões biomédicas relevantes, onde muitas vezes apenas simulações de dinâmica molecular podem fornecer respostas. Ele é baseado na integração de dois programas de modelagem amplamente utilizados, NAMD e VMD (RIBEIRO, 2016).

O QwikMD é apto a executar simulações MD simples e avançadas de forma interativa, sua arquitetura permite automatizar quantos passos forem necessários para preparar, realizar e analisar simulações, permitindo a reprodutibilidade.

³ <https://www.ks.uiuc.edu/Research/vmd/>

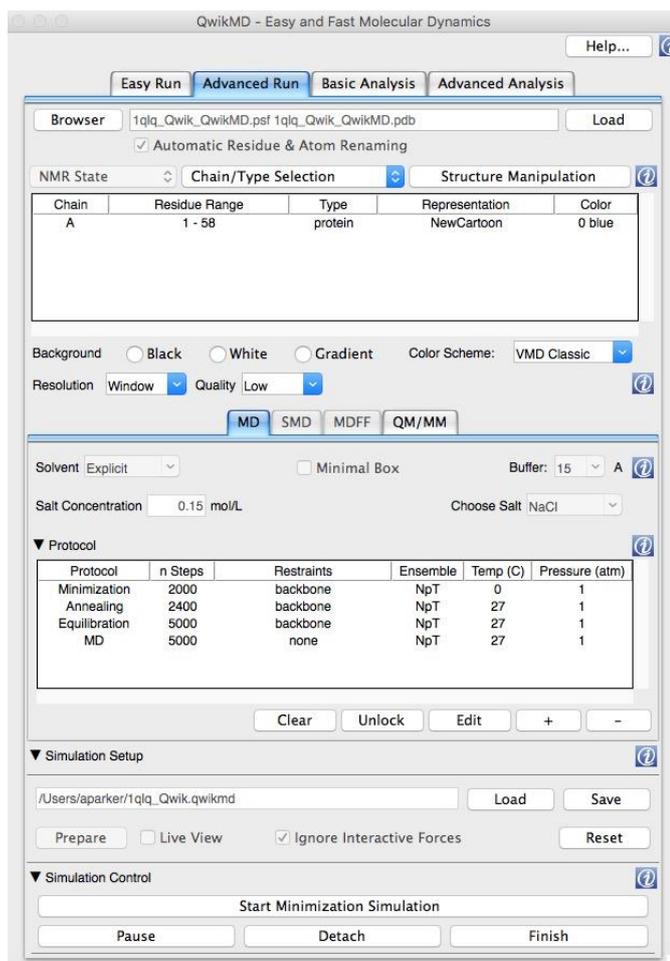


Figura 7. Tela principal do QwikMD.

Possui uma interface gráfica intuitiva que fornece um ambiente robusto e fácil de usar para um usuário não especialista, mas também deixa a liberdade de adotar variáveis para casos específicos para facilitar e agilizar o trabalho de usuários avançados. Portanto, possui ampla versatilidade no qual é fornecido para o usuário acesso a todas as estruturas subjacentes e arquivos de configuração NAMD e oferece uma oportunidade de ajustá-los individualmente quando necessário.

Em suma, o QwikMD é um software disponível gratuitamente que permite aos novatos responder uma ampla gama de questões sobre sistemas moleculares, com aplicações que vão desde a biologia celular até a ciências da saúde.

3.2.8 BioExcel Building Blocks

Construído com por meio de wrappers Python que fornece uma arquitetura interoperável, o BioBB's é inserido na categoria de ferramentas de gestão de software que é usado para geração dados ontológicos, documentação, instalação de pacotes, containers de software e disponibiliza integração

de workflows biomoleculares, o que torna útil na maioria de ambientes computacionais (ANDRIO, 2019).

Sua biblioteca oferece uma camada de interoperabilidade entre as ferramentas ao em torno, nas quais trazem compatibilidade e preparo na interconectividade para construir workflows biomoleculares complexos.

Uma de suas propostas através do pacote MDWeb (HOSPITAL, 2012), incluso no projeto é oferecer um ambiente de trabalho unificado que permite configurar um sistema para simulação de dinâmicas moleculares atômicas em complexo de proteínas, que seja apto a trabalhar com GROMACS, NAMD e Amber. Entre suas características e funcionalidades, os workflows são construídos usando componentes (blocos) de cada bibliotecas que são executados em diversos ambientes computacionais, incluindo computadores pessoais, sistemas de virtualização, nuvens públicas e sistemas HPC. Os componentes são documentados usando CWL e openAPI o que torna possível executá-lo a partir de gerenciadores de workflow de reclamações CWL. Atualmente, a proposta está em uso por mais de 3000 usuários registrados.

3.2.9 ProtCool 1.0

A versão 1.0 do ProtCool foi inicialmente proposta por (GUEDES, 2021) focado na integração entre ancoragem e dinâmica molecular de complexos proteína-ligantes. Em sua primeira versão, essa ferramenta era de uso restrito ao ambiente local do usuário.

Sua característica principal era oferecer um workflow automatizado visando a unificação de várias ferramentas na execução de todo o processo. Sendo projetado para conter vários softwares instalados na máquina do pesquisador, em que por meio de scripts obedecendo um workflow proposto, ela dispara sequencialmente todas as tarefas e etapas exigidas no processamento para preparação da MD. A entrada dos dados era realizada por preenchimento manual de arquivo “txt” de forma criteriosa, e após o preenchimento eram executados scripts diversos de busca de arquivos PDB em comunicação com API 's externas. Também realizava tratamentos de arquivos e execução dos softwares especialistas para cada etapa. Para facilitar a entrada de dados, foi criada uma interface preliminar, que basicamente apresentava um formulário web preenchendo os campos exigidos. Porém, essa versão deixava a desejar nos elementos de interatividade e dinamismo tão necessários nos aspectos de usabilidade. As operações mais avançadas de execução, aconteciam diretamente por meio de consoles, o que exigia do pesquisador conhecimento prévio no uso do sistema operacional.

A infraestrutura do ProtCool era um pacote de vários softwares especialistas que eram

instalados na máquina dos pesquisadores. A desvantagem dessa arquitetura é que toda responsabilidade do processamento e do hardware em questão fica com o usuário. Entretanto, o ProtCool alcançou sucesso em sua proposta inicial, principalmente quanto à criação da estrutura de workflows para preparação em MD, no qual foi totalmente mantida por se tratar do núcleo do sistema, com pequenas atualizações, o que permitiu dar origem ao ProtCool 2.0.

ProtCool 1.0	ProtCool 2.0
<ul style="list-style-type: none"> • Monousuário. 	<ul style="list-style-type: none"> • Multiusuário.
<ul style="list-style-type: none"> • Versão apenas para Linux. 	<ul style="list-style-type: none"> • Multiplataforma.
<ul style="list-style-type: none"> • Entrada de dados manual por arquivos de texto. 	<ul style="list-style-type: none"> • Interface web responsiva para entrada de dados.
<ul style="list-style-type: none"> • Validação de dados pós-execução. 	<ul style="list-style-type: none"> • Validação de dados pré-execução (tempo real).
	<ul style="list-style-type: none"> • Escalável.
	<ul style="list-style-type: none"> • Arquitetura cliente-servidor.

Tabela 1. Comparativo das características entre versões.

4 METODOLOGIA

4.1 DESENVOLVIMENTO DO PROTCOOL 2.0

Com o principal objetivo de alcançar reprodutibilidade por meio de *workflows* dos procedimentos *in silico* e sua representação sistêmica, foi possível desenvolver uma aplicação homogênea e padrão que atenda aos requisitos necessários para as atividades apresentados neste trabalho.

O processo de desenvolvimento foi baseado nas melhores práticas de engenharia de software que contempla todas as fases de projeto de desenvolvimento de software como a fase de comunicação, planejamento, modelagem, construção e entrega (PRESSMAN, 2005).

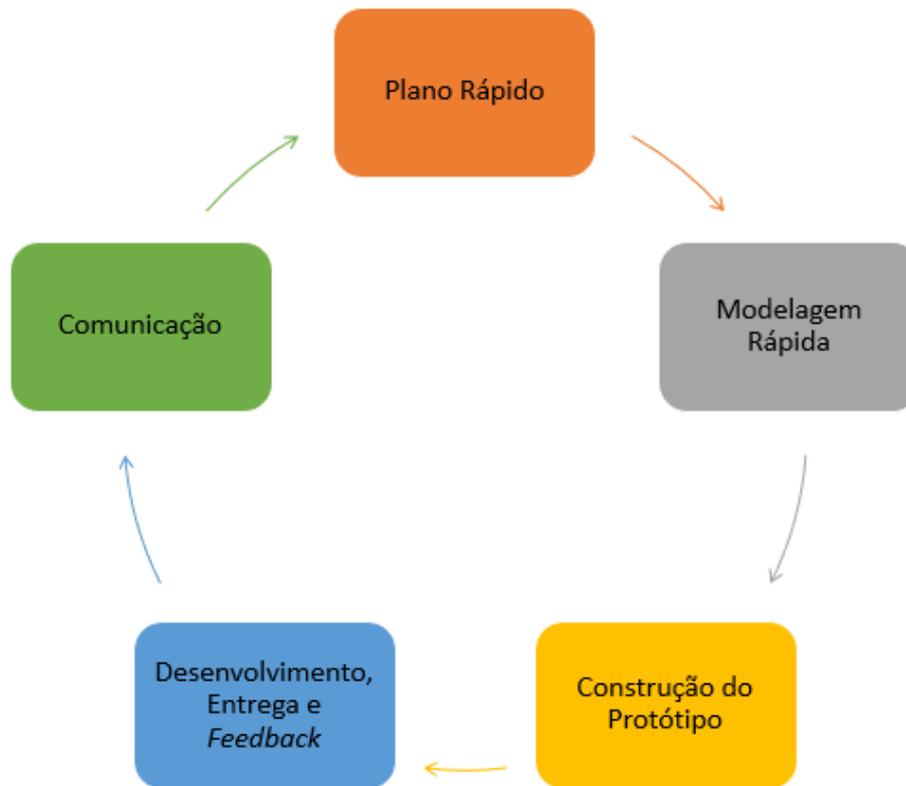


Figura 8. Ciclo de vida de processo evolucionário de software prototipação.

A abordagem de desenvolvimento escolhida foi a prototipação. Essa abordagem é interessante quando a equipe sabe muito bem o que a aplicação precisa entregar, porém não tem detalhes específicos que envolvem a produção, como a comportamento da eficiência da aplicação, a adaptabilidade do sistema operacional, ou a forma da interação homem máquina (PRESSMAN, 2005). No caso do ProtCool 2.0, o cenário apresentou-se propício à essa abordagem, uma vez que o projeto partia de um sistema legado funcional, porém ainda não possuía uma interface gráfica adequada que contemplava todo o escopo do software. Partindo dessa premissa, cada tela/componente e funcionalidade foi construída estaticamente para serem levados para aprovação e conseqüentemente ao desenvolvimento, operação e testes, correspondendo ao modelo evolutivo de processo de software (SOMMERVILLE, 2011).

4.1.1 Etapas do Processo de Software

Fase de Comunicação: o processo iniciou com a coleta dos requisitos vindo dos usuários em potencial principalmente através de entrevistas e feedback dos membros grupo de pesquisa do

ProtCool. Outra fonte foi a experiência de usuário resgatada a partir de aplicações softwares similares que possuem o mesmo objetivo.

Fase de Planejamento: surgiu a partir da adoção da abordagem do processo de prototipação na definição das atividades, tarefas e cronograma de desenvolvimento, sob responsabilidade da equipe. O desenvolvimento do ProtCool ficou dividido em 2 equipes, sendo uma responsável pelas funcionalidades do *backend* e outra pelo *frontend*, uma vez que arquitetura definida foi a arquitetura *client-server* (BERSON, 1996).

Fase de Modelagem: de posse dos requisitos, coube a decisão da equipe de desenvolvimento escolher qual seria a arquitetura do sistema implementada, assim como os competentes de usabilidade, adoção de ferramentas, plataformas de desenvolvimento, as integrações e comunicações API de *backend* e *frontend*, para suprir o escopo do projeto tendo como foco principal o desenvolvimento da interface gráfica.

Fase de Desenvolvimento: por meio da divisão do projeto entre as equipes de *backend* e *frontend*, a metodologia seguida pela equipe permitia transitar nos campos das metodologias ágeis quando se pensa na periodicidade das entregas dos produtos (BECK, 2001). A cada semana era reavaliada as atividades e entregas, eram desenhados novos protótipos de tela, sugeriam-se alterações e no caso de aprovação, partia para construção de novas funcionalidades. A divisão de desenvolvimento do ProtCool era simples de ser lida, pois bastava seguir o *workflow core* proposto da aplicação. Durante cada entrega, a equipe escrevia novos códigos, realizava testes e revisava o trabalho um do outro. A contínua integração e práticas foi usada para garantir que a aplicação fosse testada e entregue automaticamente por meio de compartilhamento de código na plataforma de versionamento github, para que fosse de fácil identificação e resolução dos problemas encontrados no processo de desenvolvimento.

Portanto, por meio da adoção dos princípios de engenharia de software, foi permitido ao ProtCool 2.0 alcançar o desenvolvimento de uma ferramenta de qualidade com características peculiares por contemplar modularidade, escalabilidade e manutenibilidade. O código foi escrito e estruturado para ser reutilizável, focado em componentização, o que torna fácil a manutenção e evolução para aplicações futuras.

4.2 IMPLEMENTAÇÃO DO PROTCOOL 2.0

Neste momento, é apresentado toda estrutura de desenvolvimento do ProtCool 2.0, bem como sua arquitetura e tecnologias utilizadas nesse processo. O foco central é a sua interface gráfica web que está hospedado em servidor utilizando a estrutura de API desenvolvidas nas linguagens como: Python⁴, JavaScript⁵, em que Python foca na estrutura de *backend* e o JavaScript focado no *frontend*. Html⁶ e CSS⁷ fazem a estética do sistema inteiramente responsiva. A ferramenta ProtCool 2.0 foi desenvolvida pela interatividade de uma API com a camada de interface gráfica construída em padrão web, em que todas as informações são geradas pela API (*backend*) e visualizadas na camada de interface (*frontend*).

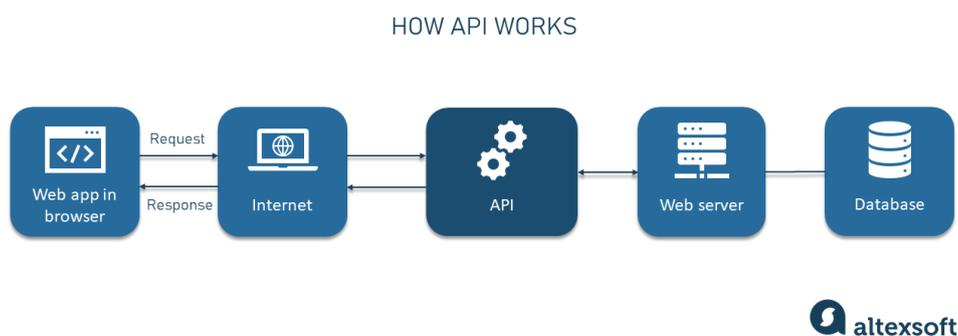


Figura 9. Representação da arquitetura de uma API (Adaptado de ALTEXSOFT, 2020).

4.2.1 Backend

O *backend* do ProtCool 2.0 foi construído totalmente na linguagem de programação Python. O Python foi a linguagem escolhida por oferecer recursos e bibliotecas robustas para tratamentos de arquivos de texto, vindo ao encontro (conforme já visto) da necessidade de tratar arquivos extensos e prepará-los para dinâmica molecular. Outra característica é que essa linguagem possui uma sintaxe simples, com uma boa curva de aprendizado, o que permite abranger de forma eficiente o potencial do projeto. Além de ter uma comunidade ativa de desenvolvedores em que facilita o suporte. Também possui uma vasta biblioteca para utilização na área de inteligência artificial e computação paralela (PYTHON, 2019). O ProtCool 2.0 é baseado em uma série de scripts isolados que formam um

⁴ <https://www.python.org>

⁵ <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

⁶ <https://developer.mozilla.org/pt-BR/docs/Web/HTML>

⁷ <https://developer.mozilla.org/pt-BR/docs/Web/CSS>

conjunto integrado, sob supervisão de um script controlador representando o *workflow in silico*, esse script é responsável por toda integração dos demais scripts, suas validações e a execução, garantindo a coesão do sistema.

O módulo chamado *Protcool_Dynamic*, responsável pela preparação de dinâmica molecular dos complexos, é representado pelo arquivo *Protcool_Dynamic.py*. Uma de suas características principais é a execução de tarefas que são programadas para serem realizadas em paralelo.

Na figura 10, as *tags* são preenchidas pelos valores ‘YES’ ou ‘NO’. Essa é a forma que representa e indica ao módulo dinâmica quais serão as fases do processo que serão executadas. Isso permite ao pesquisador escolher a execução completa das etapas ou apenas aquela que deseja, oferecendo maior flexibilidade e abertura para que ele possa optar por outras ferramentas externas que não fazem parte do *workflow* do ProtCool 2.0.

Todos os parâmetros necessários para execução do workflow com os valores de cada etapa estão contidos no arquivo *confdinamica.txt*. É baseado nesse arquivo que foi construída uma interface gráfica para o correto preenchimento, no qual as validações foram implementadas na camada de *frontend* para evitar ao máximo que erros de entrada venham acarretar o processamento incorreto da aplicação.

Na figura 10, trata-se de um arquivo básico de texto no formato ‘txt’ que pode ser acessado e editado por qualquer editor de texto padrão. Ele é formado por *tags* iniciadas e finalizadas por ‘[]’ colchetes que representam as instruções, e as linhas posteriores são os parâmetros. Para indicar o fim de uma instrução e o início de outra, os parâmetros são colocados linha a linha até que se inicie a próxima instrução (*tags*). Este procedimento é repetido até o fim do arquivo onde se encontra a *tag* ‘[END]’ que indica a finalização das instruções e do arquivo.

Diferentemente da proposta de (GUEDES, 2021), em que o ProtCool deveria ser instalado em um computador pessoal e seu funcionamento se dava apenas por operação monousuário, na versão 2.0, o ProtCool foi totalmente reestruturado para executar em ambiente *cliente-servidor* (BERSON, 1996). Nesse modelo, os parâmetros são preenchidos e validados pelo usuário por meio de um *frontend* (navegador web) e enviado ao *backend* para processamento. Após esse processamento, é retornado ao *frontend* o produto final em um arquivo compactado, para *download* do usuário. Esse arquivo compactado possui todos os arquivos e diretórios de configuração necessários para a pronta execução da dinâmica, conforme conveniência da infraestrutura computacional do usuário. O arquivo que controla a execução dessa dinâmica é o *dinamica.py*.

```

[Researcher]
Moisés Pinheiro Souza
[Email]
moises.souza@ifsuldeminas.edu.br
[Language]
Portugues
[Prepare]
No
[Docking]
No
[MolecularModel]
No
[Protonation]
Yes
[ExecutionForceField]
Yes
[PrepareNamd]
Yes
[Protein]
4EY6

```

Figura 10. Exemplo de um arquivo confdinamica.txt mostrando as principais tags de controle a serem executadas.

A versão de desenvolvimento e teste do ProtCool 2.0 atualmente está instalada em um computador HP Deskpro G6 com 8GB de memória RAM, SSD de 256GB e um processador Intel(R) Core (TM) i3-9100 CPU @ 3.60GHz. A versão do sistema operacional foi o Ubuntu 20.04.4 LTS (Focal Fossa) - 64Bits e os programas instalados foram: BioNimbuZ 2⁸, Namd 2.11 for Linux-x86_multicore-CUDA⁹; VMD¹⁰; PyMOL¹¹; WORDOM¹² v.0_22-rc3.x86-64; AmberTools¹³,

⁸ <https://github.com/bionimbuz/Bionimbuz>

⁹ <https://www.ks.uiuc.edu/Research/namd/>

¹⁰ <https://www.ks.uiuc.edu/Research/vmd/>

¹¹ <https://pymol.org/2/>

¹² <https://wordom.sourceforge.net/>

¹³ <https://ambermd.org/AmberTools.php>

OpenBabel¹⁴, RDKit¹⁵; Pacote Request do Python; Vina¹⁶; Smina¹⁷; MGLTools¹⁸; Modeller¹⁹; Selenium²⁰; Geckodriver²¹; Pacote pyPDB²²; CUDA²³; R²⁴; Python (2.7 e 3)²⁵; GNU (gcc, g++, gfortran); NodeJS²⁶; VueJS²⁷; Bibliotecas Python de compactação, download e upload de arquivos.

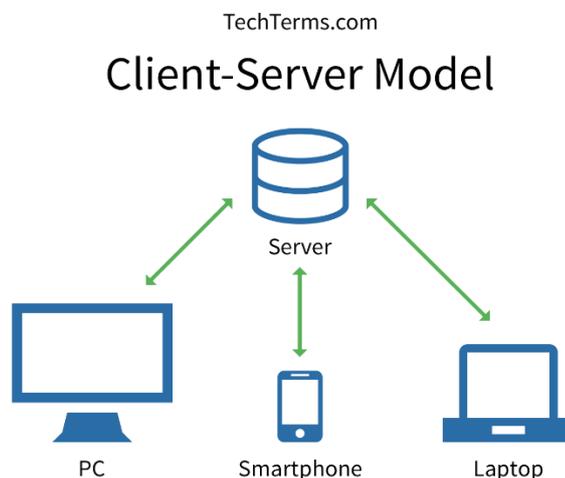


Figura 11. Representação da arquitetura cliente-servidor (Adaptado de “Client-Server Model Definition”, 2016).

Essa adaptação trouxe benefícios se comparado com a necessidade de se criar um instalador com todas essas ferramentas conforme vemos na versão desktop de (GUEDES, 2021). Entre os ganhos, o principal destaque é a arquitetura para operação multiusuário em que uma vez hospedado em um servidor *web* na internet, toda a comunidade acadêmica pode utilizar de qualquer lugar ou dispositivo sem a necessidade de realizar qualquer instalação. Também poupa o usuário de disponibilizar recursos computacionais próprios, e não se preocupar com upgrades de hardware, manutenção e suporte, no que possibilita destinar o foco integralmente a pesquisa da simulação em si.

¹⁴ https://openbabel.org/wiki/Main_Page

¹⁵ <https://www.rdkit.org/>

¹⁶ <https://vina.scripps.edu/>

¹⁷ <https://vina.scripps.edu/>

¹⁸ <https://ccsb.scripps.edu/mgltools/>

¹⁹ <https://salilab.org/modeller/>

²⁰ <https://salilab.org/modeller/>

²¹ <https://github.com/mozilla/geckodriver/releases>

²² <https://github.com/williamgilpin/pypdb>

²³ <https://developer.nvidia.com/cuda-downloads>

²⁴ <https://www.r-project.org/>

²⁵ <https://www.python.org/>

²⁶ <https://nodejs.org/en>

²⁷ <https://vuejs.org/>

Na figura 12 são listados os objetivos, encadeamento e os momentos em que cada etapa é chamada à execução.

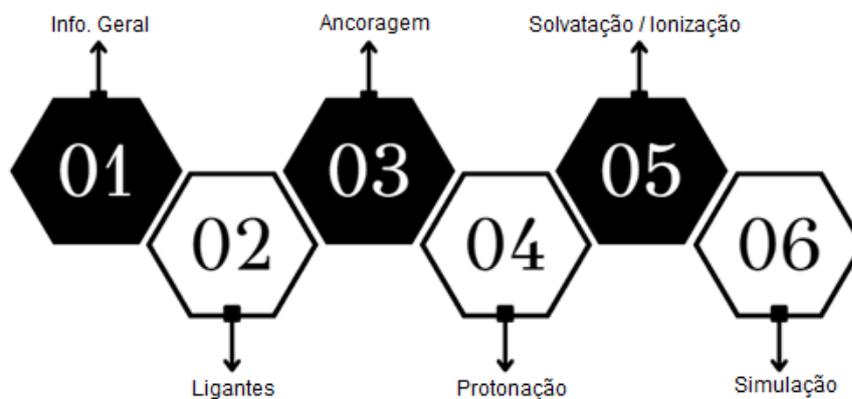
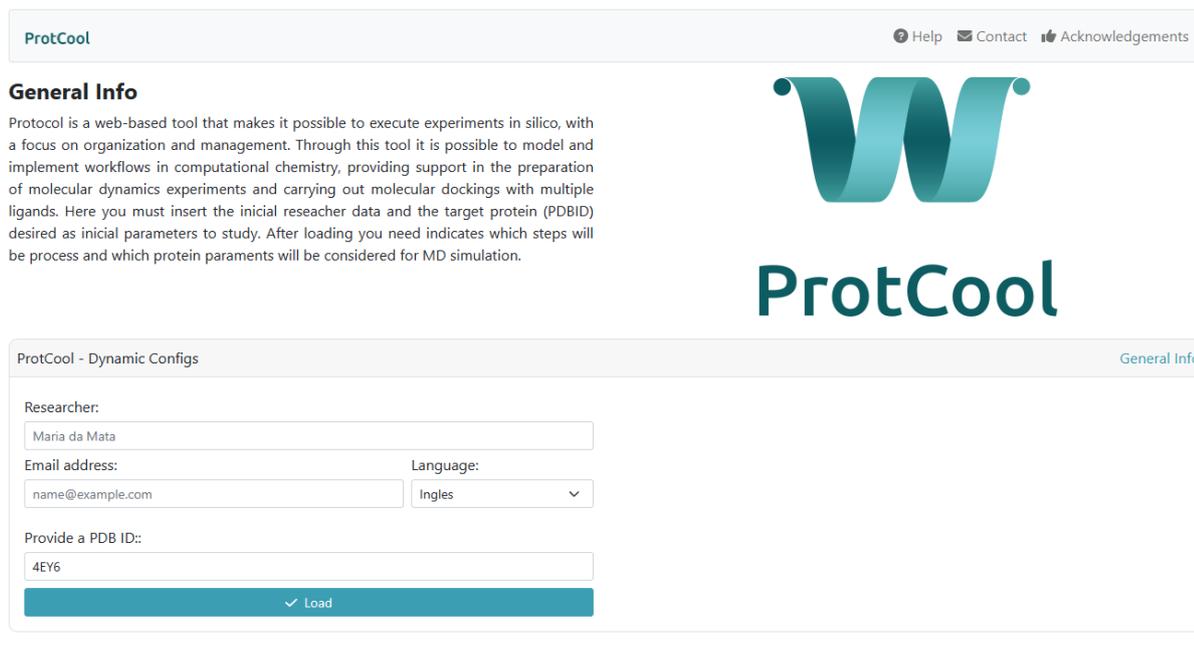


Figura 12. Encadeamento das etapas do ProtCool regidos pelo script ‘Protcool_Dynamic.py’.

O ProtCool 2.0 foi dividido em passos, como: Informações Gerais da Dinâmica, Ancoragem, Ligantes, Protonação, Solvatação e Ionização, Simulação. O foco desta dissertação é apresentar a construção da ferramenta pela óptica do *frontend*, ou seja, apresentar os aspectos da construção da interface gráfica e as funcionalidades adicionadas no auxílio de filtros e validações do processamento na geração dos arquivos da dinâmica molecular. Para mais detalhes sobre os scripts e infraestrutura de funcionamento completo, contendo a construção do *backend* e toda a interação/integração com os demais softwares e conjunto de ferramentas, juntamente com exemplo de uso, consultar o trabalho de (GUEDES, 2021).

4.2.2 Frontend

O objetivo da interface gráfica construída neste trabalho é auxiliar o correto preenchimento do *confdinamica.txt*, apresentado na subseção anterior. Após realizar a chamada de execução dos scripts de *backend*, ela disponibiliza para *download* os arquivos gerados de preparação da dinâmica molecular. Portanto, foi desenvolvida a interface *web* que permite ao pesquisador preencher formulários em que cada campo guarda relação com as *tags* do *confdinamica.txt*, divididos em etapas como abstração do *workflow* original.



ProtCool Help Contact Acknowledgements

General Info

Protocol is a web-based tool that makes it possible to execute experiments in silico, with a focus on organization and management. Through this tool it is possible to model and implement workflows in computational chemistry, providing support in the preparation of molecular dynamics experiments and carrying out molecular dockings with multiple ligands. Here you must insert the inicial researcher data and the target protein (PDBID) desired as inicial parameters to study. After loading you need indicates which steps will be process and which protein paraments will be considered for MD simulation.

ProtCool

ProtCool - Dynamic Configs General Info

Researcher:

Email address: Language:

Provide a PDB ID:



Figura 13. Tela inicial do ProtCool 2.0

Dentre as tecnologias para desenvolvimento web, foi utilizado o framework Javascript Vue.js, escolhido devido sua versatilidade em conciliar um desenvolvimento que adota as melhores práticas do mercado com funcionalidades simples de implementação (“Vue.js”, [s.d.]). O Vue possui uma estrutura voltada à componentização, permitindo forte reutilização de código. Inclui toda a interface e elementos personalizados criados no ProtCool 2.0 como: Botões, Caixa de Textos, *Selects*, *Divs*, *Inputs* etc. assim como os eventos, propriedades e validações. Com isso, uma vez construído, esses elementos podem ser facilmente acoplados ao código, aumentando a velocidade do *designer* da interface e permitindo ajustes de responsividade, podendo ser utilizado em qualquer dispositivo.

Dessa forma, o *frontend* do ProtCool 2.0 consiste basicamente em: receber as informações no formulário pelo usuário, realizar tratamento/validação das informações quando couber, preencher o arquivo *confdinaminca.txt* e fazer a chamada de execução. Todo esse processo de comunicação da API (Python) e a interface é feito pelo padrão JSON (“JSON”, [s.d.]). Ele envia e recebe informações por JSON entre API e a interface web, ao fim transforma um objeto JSON em *'txt'*, faz a chamada de execução da preparação dos arquivos pelo lado do servidor e gera o andamento do processamento em tela de forma transparente ao usuário.

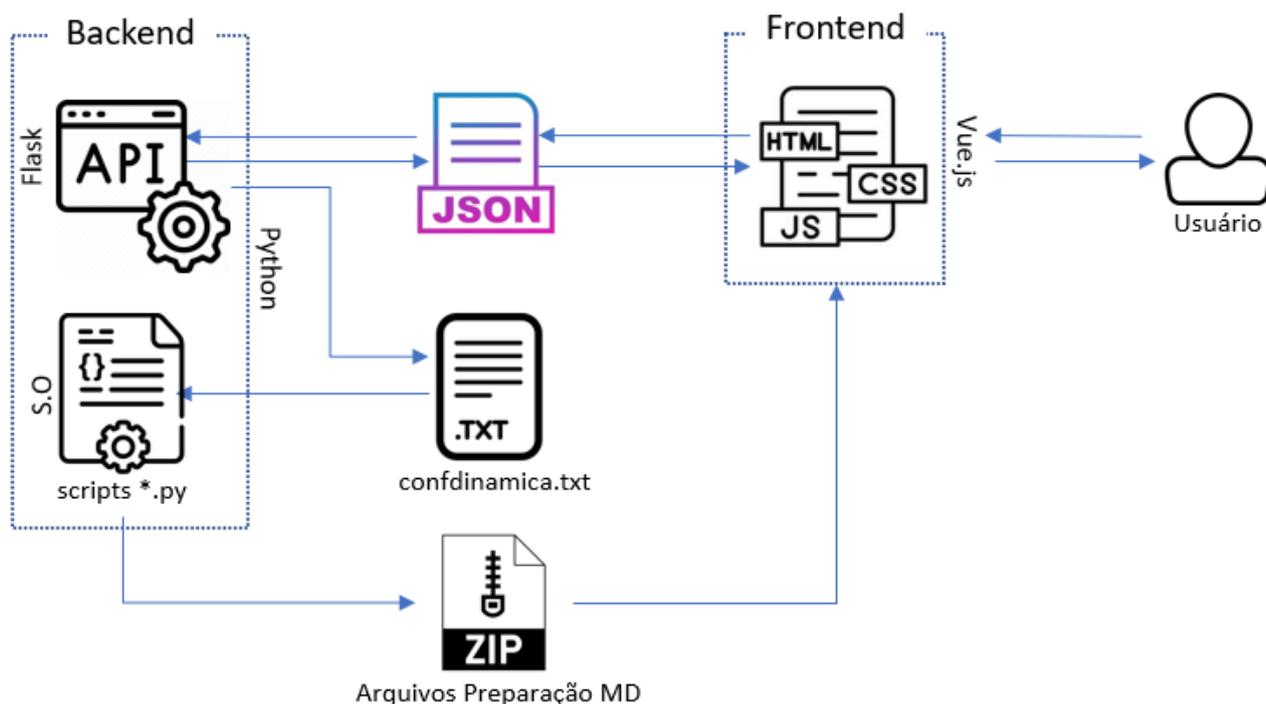


Figura 14. Arquitetura do ProtCool 2.0

Para acessar a versão de testes do ProtCool 2.0 basta acessar o endereço <http://200.131.57.242:8080> de qualquer navegador e de qualquer dispositivo conectado à internet. O fluxo e as etapas da ferramenta são apresentados conforme a figura 14.

4.2.3 Estrutura de pastas e arquivos de saída

O ProtCool 2.0 organiza todos os arquivos necessários para a execução da simulação, além dos logs de todo o processo. Os logs permitem ao pesquisador analisar como foi feita toda a preparação. Como exemplo, pode-se citar os arquivos que representam os ligantes, onde para cada complexo (proteína-ligante) é gerada uma pasta, e dentro da pasta os arquivos PDB com as restrições definidas.

Como já dito, um dos arquivos mais importantes é *'confdinamica.txt'*, ele é o resumo de todas as informações de todas as etapas do ProtCool 2.0. É neste processo que a interface captura todos os campos, transforma as informações em um JSON e reescreve no arquivo *'confdinamica.txt'* conforme pede a preparação. Na dinâmica, a tarefa de preparação da simulação gera boa parte dos arquivos de saída como os arquivos PDB, prmtop, inpcrd, todos os arquivos de restrição, todos os arquivos de configuração e um arquivo de execução chamado *dinamica.py*.

4.2.4 Logs

No arquivo compactado de saída do ProtCool 2.0 estão anexados 2 arquivos de log. O primeiro chamado '*logWorkflow.log*' apresenta o passo a passo como ocorreu toda a execução do processamento. Já o segundo arquivo de log chamado '*logchamada.log*' que apresenta todas as informações referentes às chamadas realizadas. Sua função é mostrar se todas elas foram devidamente executadas, obedecendo a ordem imposta no *workflow*.

Portanto, por meio desses arquivos é possível saber se tudo ocorreu de forma adequada e gerou os arquivos corretamente, ou se apresentou erros durante a execução. Vale ressaltar que todos os scripts foram desenvolvidos para realizar uma validação coerente, no qual seu objetivo é descobrir se todos os campos foram preenchidos de maneira correta. Caso exista uma entrada não condizente, é abortado o processo e lançado um erro no arquivo '*Errorlog.txt*', registrando detalhes da etapa onde ocorreu.

5 FUNCIONALIDADES DO PROTCOOL 2.0

5.1 WORKFLOWS

Conforme mencionado anteriormente, o ProtCool 2.0 trabalha com a definição de protocolos por meio de *workflows*. Segundo (GUEDES, 2021) existe um modelo que descreve a ordem das etapas de modelagem, projeto, inserção, execução e análise de *workflows*.

- i. Hipóteses, Experimentos e Objetivos – Estabelecer os *workflows* a serem desenvolvidos;
- ii. Projeto do *workflow*
 - a. Modelagem do *workflow*
 - i. Definir Tarefas do *workflow*
 - ii. Definir arquivos de entrada
 - iii. Validar *workflow*
 - iii. Instanciação do *workflow*
 - a. Definir parâmetros do *workflow*
 - b. Desenvolver scripts
 - c. Preparar arquivos de entrada
 - d. Preparar ambiente de execução do *workflow*

- iv. Execução do *workflow*
- v. Análise pós-execução
 - a. Analisar resultados do *workflow*
 - b. Definir próximos passos

Tabela 2. Tabela de das etapas do workflow do ProtCool (GUEDES, 2021).

i- Hipóteses, Experimentos e Objetivos.:

O principal objetivo dessa etapa é desenvolver o *workflow* de preparação de dinâmicas moleculares baseada nas hipóteses de criação do mesmo, de forma automatizada liberando o pesquisador para dedicar mais tempo para a etapa de análise de resultados.

ii- Projeto do *Workflow*.:

A partir da etapa de hipótese é feito o desenvolvimento do projeto do *workflow*, por meio da definição de tarefas, arquivos de entrada e a validação do *workflow*.

Essa etapa é formada por várias tarefas, divididas em seis partes: Configuração Inicial das Moléculas; Ancoragem Molecular (*docking*); Modelagem Molecular; Protonação; Campo de força, Solvatação e Ionização; Geração de Arquivos para a execução da Dinâmica (Simulação). Sendo que, cada uma das tarefas apresentadas é formada por uma série de scripts que foram desenvolvidos para executar todos os passos necessários do sistema (GUEDES, 2021).

Configuração Inicial das Moléculas: é o momento que o pesquisador seleciona a molécula objeto de estudo, no qual é realizado por meio de script a recuperação do arquivo com as coordenadas atômicas, sendo sempre uma proteína nesta versão do ProtCool. O script desenvolvido carrega automaticamente arquivos oriundos do principal repositório de coordenadas de biomoleculares chamado PDB (*Protein Data Bank*) (BERMAN, 2000). Entre outras tarefas o script realiza a limpeza do arquivo tornando mais organizado e enxuto, de forma a possibilitar que o arquivo esteja adequado para as demais fases do processo de preparação da dinâmica molecular.

Docking Molecular: visa buscar o encaixe mais perfeito possível entre duas moléculas. (LENGAUER, 1996) O usuário informa previamente os arquivos do alvo e do ligante, adicionando as coordenadas do sítio ativo ou pocket onde o ligante será inserido. Também deve informar o algoritmo que será usado para busca das melhores poses (com os melhores scores) de inserção no

pocket. Modelagem Molecular: nesta etapa, o objetivo é resolver lacunas na sequência dos resíduos, em função da sua indeterminação estrutural, por meio da modelagem por homologia. através dos seguintes passos: identificação do homólogo como modelo; alinhamento entre as sequências; geração do *backbone*; modelagem dos loops existentes; modelagem das cadeias laterais, otimização do modelo; e validação do modelo (CARVALHO, 2003).

Protonação: em função de limitações experimentais, nem todo arquivo PDB contém o posicionamento dos hidrogênios. Portanto é necessário a busca dos estados de protonação nessas estruturas, que levará em conta também o contexto físico-químico de cada átomo a ser hidrogenado existem ferramentas computacionais que fazem este trabalho. O ProtCool 2.0 utiliza o H++ (ONUFRIEV, 2005). Por meio do H++ é feito a inserção de hidrogênios ausentes na estrutura de acordo com o pH da solução e estimativas de pKa de cada resíduo (GUEDES, 2021).

Solvatação e Ionização: compreende a inserção de águas e íons explícitos ao redor do complexo proteína-ligante. Entra também com dados referentes à solvatação e ionização que serão utilizados no campo de força. Nesta versão do ProtCool, aceitam-se apenas parâmetros e campos de força da *suite* AMBER (CASE, 2008).

Geração Arquivos da Dinâmica: é a última etapa do ProtCool, na qual o pesquisador deve informar os parâmetros e configurações necessárias para a execução da simulação da dinâmica molecular em si. Atual versão do ProtCool trabalha com o sistema NAMD (NELSON, 1996). Esta etapa pode ser subdividida em outras, conforme desejo do usuário. Em geral, alocam-se uma ou mais etapas para a minimização de energia do sistema, seguida por outras etapas de relaxamento atômico de forma gradual, definindo tempos de simulação para cada uma. Após o relaxamento e equilíbrio do sistema, vem a simulação da dinâmica molecular per se, por um tempo mais longo.

iii. Instanciação do workflow

Uma vez que o *workflow* esteja definido é possível executar a instanciação. Essa etapa indica quais as tarefas serão executadas e tratadas por scripts que trabalham de forma isolada em cada passo do sistema. O objetivo é rodar apenas uma única atividade e que essa tarefa possa ser usada em outros *workflows* como forma de garantir a reprodutibilidade. Apesar dos scripts terem sido desenvolvidos de forma modular, suas entradas e saídas estão integradas, garantindo o encadeamento do workflow como um todo.

Dentre as tarefas, a definição do *workflow* preocupa em resolver as questões do fluxo do sistema como: o que é o trabalho da ferramenta, o que deve ser gerado pelas tarefas modeladas, quais tarefas devem ser automatizadas do *workflow*, e quais precisam de auxílio do usuário, e quais são unicamente realizadas pelo usuário. Logo, compete à instanciação do *workflow* a responsabilidade da preparação e adequação dos arquivos de entrada e saída, além da preparação do ambiente de execução, que em algumas condições pode ser processado de forma paralela.

iv. Execução do workflow

Uma vez que as etapas de instanciação foram concluídas, a execução ocorre por meio do script de gestão do *workflow* denominado “controlador”. É ele o responsável pelo gerenciamento de todo o processo. Sua função é controlar toda execução, possibilitando o pesquisador ter conhecimento do status do processo a qualquer momento, disponibilizando o produto final por meio de arquivos nos quais ele poderá pôr para execução toda a simulação. O pesquisador faz o download desses arquivos gerados no fim de execução. Porém, para que tudo funcione conforme desejado, a preocupação do *workflow* é disponibilizar e garantir que os arquivos iniciais estejam presentes no sistema.

Uma série de scripts foram desenvolvidos com esta função de validar o arquivo de configuração gerado, em que são checados se todas as *tags* obrigatórias para o perfeito funcionamento estão de acordo junto com os valores aceitáveis. Há checagens de validação tanto no lado *frontend* quanto *backend*.

v. Análise pós execução

Conforme afirmado anteriormente, o ProtCool apenas faz a preparação para execução das simulações, ele não as realiza. Mas, um usuário especialista pode checar e analisar o arquivo *confdinamica.txt* gerado, se as informações ali registradas condizem com o esperado para a simulação.

5.2 FUNCIONALIDADES

Seguem abaixo as principais funcionalidades implementadas a versão web do ProtCool. O nome das funcionalidades está em inglês, por conta que na versão web somente este idioma está habilitado. Outros idiomas serão disponibilizados em versões futuras.

5.2.1 Etapa General Info

A tela ‘General Info’ é responsável por receber as informações iniciais para execução da ferramenta, sob domínio do script `General_Info.py`.

Os primeiros campos envolvem: nome do pesquisador, e-mail e idioma, no momento disponível apenas ‘Inglês’. O campo ‘Provide a PDB ID’ o usuário entra com o código PDB.

Ao apertar o botão ‘Load’ uma série de validações são disparadas, a primeira delas é não permitir que avance se os campos deste formulário estiverem vazios. No decorrer do projeto essa validação ocorre em vários momentos. Outra validação, é certificar que um e-mail seja válido. A próxima validação é realizar a busca do arquivo PDB diretamente no site RCSB²⁸ por meio da API que utiliza a biblioteca ‘pypdb’ escrita em Python. A API busca o arquivo conforme informado e se encontrado retorna várias propriedades e informações que são pertinentes para o avanço das próximas etapas do ProtCool 2.0. Entre elas, as atividades que deverão ser executadas pelo *workflow*.

Segue descrições de cada opção apresentada ainda nesta etapa inicial:

i - [Prepare]: indica que será considerada a primeira etapa de preparação, e que será realizada a adequação inicial das biomoléculas alvos. Entre essas adequações estão a eliminação de águas estruturais e certos tipos de heteroátomos.

ii - [Docking]: indica que será realizado a ancoragem ou docking de ligantes; uma vez ativada, é possível selecionar as ferramentas de docking entre ‘Vina’ e ‘Smina’. Na versão web, há somente a opção ‘Vina’ implementada.

iii - [MolecularModel]: indica que será realizada a modelagem molecular por homologia para correção de *gaps*.

iv - [Protonation]: indica que será realizada a protonação, conforme descrito anteriormente.

v - [ExecutionForceField]: indica que a etapa de definição de campo de força, solvatação e ionização que serão realizadas, conforme descrito anteriormente.

²⁸ <https://www.rcsb.org/>

vi - [PrepareNamd]: indica que a etapa de preparação final dos arquivos de configuração será realizada para a execução no padrão NAMD.

Além disso, temos as opções não default que podem ser ativadas pelo usuário como ‘Hydrogen’, ‘Residues’, ‘Water’.

vii - [Hydrogen]: indica se o usuário deseja utilizar os hidrogênios da estrutura fornecida pelo arquivo PDB, se presentes.

viii - [Heteroatom]: indica se devem ser preservados certos compostos ligados a heteroátomos. Neste ponto uma função é disparada na API para verificar se a molécula selecionada possui heteroátomos. Caso encontre ele retornará automaticamente com todas as opções encontradas e organizadas por suas respectivas cadeias.

ix - [Water]: indica se o usuário deseja manter as águas cristalográficas da estrutura, também chamadas águas estruturais. A API busca e trata os arquivos PDB, existe uma função que realiza a leitura completa e verifica se existem moléculas de água na proteína informada. Ela consiste em buscar nas linhas e colunas determinadas a *string* ‘HOH’ em conjunção com a *string* ‘HETA’. Caso encontre alguma molécula ele retornará automaticamente com a opção [Water] ativada.

ProtCool - Dynamic Configs General Info

Researcher:

Email address: Language:

Provide a PDB ID:

Define workflow steps:

- Prepare
- Protonation
- Prepare Namd
- Hydrogen
- Molecular Model
- Execution Force Field
- Docking
- Vina
- Smina
- Heteroatom
- Water

Model:
 Model: Not Found

Chains:
 Chain: A Chain: B

Disulfide Bridges:

- CYS A 69 - CYS A 96
- CYS A 257 - CYS A 272
- CYS A 409 - CYS A 529
- CYS B 69 - CYS B 96
- CYS B 257 - CYS B 272
- CYS B 409 - CYS B 529

Figura 15. Tela de opções iniciais da etapa ‘General Info’.

Outras funcionalidades associadas à proteína escolhida são retornadas com as informações de ‘Model’, ‘Chains’ e ‘Bridges’.

x - [Model]: permite selecionar qual dos modelos quando as estruturas são resolvidas por RMN e outras técnicas. A função na API busca e verifica se há 'MODELS' no arquivo PDB, se encontrado informa a interface por meio de JSON e essa gera os componentes gráficos dinamicamente em tela. Neste caso, trata-se de *radio buttons*, pois só é permitida uma seleção dentre todas apresentadas.

xi - [Chains]: permite selecionar as cadeias. Como pode haver mais de uma cadeia, este campo é obrigatório, não é permitido que o sistema continue sem que o usuário informe pelo menos um 'CHAIN'. Esses componentes gráficos são gerados dinamicamente pela comunicação entre API e o *frontend*.

Figura 16. Opção 'Model' na tela inicial do ProtCool 2.0

xii - [Disulfide Bridges]: permite selecionar quais pontes dissulfeto o usuário deseja utilizar, se existirem no arquivo PDB. Pontes dissulfeto são ligações covalentes entre átomos de enxofre relacionados aos resíduos cisteínas, distantes sequencialmente um do outro, mas próximas estruturalmente. Essas cisteínas podem inclusive estar em cadeias diferentes. Um ponto interessante é que foi desenvolvido uma lógica em que as pontes dissulfeto geradas dinamicamente retornadas pela API são atreladas obrigatoriamente à opção anterior CHAIN. Logo, só é possível selecionar as pontes desejadas se elas são associadas às CHAINS selecionadas. Nesse processo de retorno existe uma função que verifica no arquivo PDB existente e retorna a lista de pontes encontradas conforme figura 16.

Após todos os campos devidamente preenchidos pelo usuário na tela ‘General Info’, ele pode avançar apertando o botão ‘Next’. Neste momento são realizadas as validações citadas anteriormente, e se tudo estiver ‘OK’, o sistema avança para tela referente às informações dos ligantes. Importante ressaltar que a interface foi construída pensando na fluidez do processo, ou seja, ao passar para uma tela seguinte, os dados da tela anterior permanecerão intactos, permitindo ao usuário tanto avançar quanto retroceder nos passos. Isso facilita na alteração de alguma informação no meio do processo e evita retrabalho.

5.2.2 Etapa Ligands

A etapa ‘Ligands’ é responsável por receber a informação dos ligantes para realização da ancoragem ou *docking* na proteína alvo. É possível dar entrada desses arquivos de 2 formas diferentes, sendo a primeira delas preenchendo o campo ‘ZINC’, e a segunda por meio de *upload* de arquivos *.mol2 fornecidos pelo pesquisador. O ZINC é uma das bibliotecas de ligantes mais usadas na área, por ser curada e só envolver compostos comercializáveis (STERLING; IRWIN, 2015). Para a recuperação automática do ZINC, o pesquisador deve fornecer o código completo do ligante no ZINC, por exemplo (ZINC00491073) e um código de 3 caracteres associado como GNT para a Galantamina. Para realizar a busca do ligante direto no site do ZINC²⁹ a API utiliza o módulo *requests* do Python. Essa tarefa é realizada pelo *backend*. Como a versão web do ProtCool é dedicada às simulações de complexos proteína-ligante, informar pelo menos um ligante é obrigatório.

Portanto, uma vez preenchido corretamente os campos, é obrigatório que o usuário adicione ligantes na tabela por meio do botão ‘Increase’. Conforme mencionado é possível que o usuário adicione quantos ligantes desejar, seja ele por meio da busca automática via script (*backend*), ou realizando diretamente o *upload* de arquivo local. Esta etapa só avança se houver pelo menos uma informação na tabela. O ProtCool irá realizar *docking* e simulações de dinâmica separadamente para cada par ligante-alvo.

²⁹ <https://zinc15.docking.org/>

ProtCool - Dynamic Configs Ligands

ZINC File/MOL2

Ligand: Name Ligand: Ligand: Name Ligand:

Ligand Code: NumDoc: Ligand Code hyb: NumDoc:

Force Field: Force Field:

#	Ligands	Name Ligand	Code Ligand	Force Field	Num Doc	Action
0	ZINC00491073	gnt	GNT	~/amber18/dat/leap/cmd/oldff/leaprc.ff99SB	1	<input type="button" value="Delete"/>
1	ZINC00491073	gnt	GNT	~/amber18/dat/leap/cmd/oldff/leaprc.ff99SB	1	<input type="button" value="Delete"/>

Figura 17. Tela para informar os parâmetros referentes aos ligantes.

Abaixo é descrito os campos e suas respectivas finalidades:

- i - [ZINC]: abre os campos para inserir informações dos ligantes do tipo ZINC que serão utilizados.
- ii - [Ligand]: permite inserir o código para busca do ligante. No caso de haver demanda de mais de um ligante deverá ser buscado e adicionado na tabela separadamente por meio do botão ‘Increase’. No campo, é necessário colocar o prefixo “ZINC” na frente do número do ligante, conforme o exemplo “ZINC4102420”.
- iii - [Ligand Name]: permite inserir o código dado ao ligante pelo usuário. Deve ser escrito com 3 caracteres entre letras e números, pois será utilizado como código do resíduo em arquivos ‘PDBs’ que vierem a ser gerados.
- iv - [Ligand Code]: *tag* que contém o código do ligante formado por três caracteres maiúsculos, segue a mesma ordem do [Ligand]. - Difere maiúscula de minúscula.
- v - [ForceField]: permite informar o campo de força que será utilizado. Nesta versão, só está permitido o ff19SB do AMBER.

vi - [NumDoc]: permite informar o valor numérico correspondente à quantidade de vezes que um docking rodará para seleção de poses do sistema. Para cada ligante pode-se ter um número.

No caso de inserção por meio de upload de arquivo *.mol2, esses campos se repetem e possuem as mesmas funções.

5.2.3 Etapa Docking

Na versão web, essa etapa trabalha apenas com a ancoragem ou Autodock Vina (TROTT, 2010) A seleção da pose é aferida por função de pontuação (score), que opera sobre cálculos de energias de interações aproximados, para um melhor desempenho computacional, ainda que isso implique em menor precisão. A seguir, uma descrição mais detalhada dos campos exigidos no *docking*:

i - [center_x]: permite inserir o valor do centro geométrico do pocket no eixo x. Para facilitar a inserção dos valores, foram disponibilizadas na interface duas formas, a primeira digitando diretamente o valor no 'input' e a segunda ajustando o valor pelo componente 'slider'.

ii - [center_y]: permite inserir o valor do centro geométrico do pocket no eixo y. Para facilitar a inserção dos valores, foram disponibilizadas na interface duas formas, a primeira digitando diretamente o valor no 'input' e a segunda ajustando o valor pelo componente 'slider'.

iii - [center_z]: permite inserir o valor do centro geométrico do pocket no eixo z. Para facilitar a inserção dos valores, foram disponibilizadas na interface duas formas, a primeira digitando diretamente o valor no 'input' e a segunda ajustando o valor pelo componente 'slider'.

iv - [size_x]: permite inserir o valor do tamanho geométrico do pocket no eixo x. Para facilitar a inserção dos valores, foram disponibilizadas na interface duas formas, a primeira digitando diretamente o valor no 'input' e a segunda ajustando o valor pelo componente 'slider'.

v - [size_y]: permite inserir o valor do tamanho geométrico do pocket no eixo y. Para facilitar a inserção dos valores, foram disponibilizadas na interface duas formas, a primeira digitando diretamente o valor no 'input' e a segunda ajustando o valor pelo componente 'slider'.

vi - [size_z]: permite inserir o valor do tamanho geométrico do pocket no eixo z. Para facilitar a inserção dos valores, foram disponibilizadas na interface duas formas, a primeira digitando diretamente o valor no 'input' e a segunda ajustando o valor pelo componente 'slider'.

OBSERVAÇÃO: Importante destacar que os centros e tamanhos em x, y, z acima encapsulam a região de *docking* num prisma tridimensional.

vii - [Consider Hydrogens In The Protein]: permite informar se serão considerados ou não os hidrogênios na proteína para o *docking*.

viii - [Consider Hydrogens In The Ligand]: permite informar se serão considerados ou não os hidrogênios nos ligantes.

ix - [Exhaustiveness]: permite determinar o esforço computacional que será empregado na exploração do espaço conformacional do ligante durante o *docking*. Caso desativado, será utilizado o valor padrão do sistema de *docking*. Caso ativado, será apresentado o input para inserção de valor.

x - [Energy Range]: permite determinar a diferença máxima de energia aceita entre as conformações em kcal/mol. Caso desativado, será utilizado o valor padrão do sistema de *docking*. Caso ativado, será apresentado o input para inserção de valor.

xi - [Flex]: permite informar se será realizado o *docking* flexível. Nessa modalidade, aceita-se que certos resíduos na proteína alvo escolhido pelo usuário possa ter liberdade de movimentos. Caso ativado, será apresentado novos campos na tela, para serem inseridos os resíduos da proteína que deseja tornar flexível, no formato código de resíduo de 3 letras e id, tipo "TRP86" (um triptofano na posição 86 na sequência primária). Uma vez ativada, a informação dos resíduos é obrigatória, e a interface garante por meio de validação se o campo foi devidamente preenchido ou não. Além disso, é necessário informar a cadeia (CHAIN) vinculada, ela já pré-definida a partir da seleção feita na etapa 'General_Info'. Após o preenchimento, deve-se confirmar a inserção por meio do botão 'Increase', a informação de resíduos será colada em uma tabela linha a linha.

xii - [Num Modes]: permite informar o valor do número de poses máximas que serão geradas

pelos sistemas de *docking*. Caso desativado, será utilizado o valor padrão do sistema de *docking*. Caso ativado, será apresentado o input para inserção de valor.

xiii - [CPU]: permite informar o número de CPUs que será utilizado no processo de *docking*. O valor default é 1.

The screenshot shows the 'ProtCool - Dynamic Configs' interface. At the top, it is titled 'ProtCool - Dynamic Configs' on the left and 'Docking' on the right. Below the title, there are three sliders for 'Center X', 'Center Y', and 'Center Z'. The values are: Center X: -14,041; Center Y: -42,654; Center Z: 28,419. Below these are three sliders for 'Size X', 'Size Y', and 'Size Z', all set to 25. On the left, there are several toggle switches: 'Consider Hydrogens In The Protein' (off), 'Consider Hydrogens In The Ligand' (off), 'Exhaustiveness' (on), 'Energy Range' (on), 'Flex' (on), and 'Num Modes' (on). In the center, there are input fields for 'Exhaustiveness' (1), 'Energy Range' (1), 'Num Modes' (10), and 'CPU Number' (1). On the right, there are dropdown menus for 'Chains' (A) and 'Flex Docking' (TRP86), along with an 'Increase' button. At the bottom, there is a table with columns '#', 'Flex Docking', and 'Action'. The table lists five entries with their respective chain IDs and 'Delete' buttons. At the very bottom, there are 'Prev' and 'Next' navigation buttons.

#	Flex Docking	Action
0	A:TRP86	Delete
1	A:TYR124	Delete
2	A:SER203	Delete
3	A:TYR337	Delete
4	A:HIS447	Delete

Figura 18. Tela para informar os parâmetros referentes ao *docking*.

ProtCool - Dynamic Configs Docking

Center X: -14,041 Center Y: -42,654 Center Z: 28,419

Size X: 25 Size Y: 25 Size Z: 25

Consider Hydrogens In The Protein
 Consider Hydrogens In The Ligand
 Exhaustiveness
 Energy Range
 Flex
 Num Modes

Exhaustiveness: Exhaustiveness cannot be empty!

Chains: A Flex Docking: TRP86

Energy Range: Energy Range cannot be empty!

Num Modes: Num Modes cannot be empty!

CPU Number: CPU Number cannot be empty!

#	Flex Docking	Action
0	A:TRP86	<input type="button" value="Delete"/>
1	A:TYR124	<input type="button" value="Delete"/>
2	A:SER203	<input type="button" value="Delete"/>
3	A:TYR337	<input type="button" value="Delete"/>
4	A:HIS447	<input type="button" value="Delete"/>

Figura 19. Demonstração da validação dos campos vazios na tela 'Docking'.

Ao avançar para a próxima tela por meio do botão 'Next' as validações dos campos na interface em JavaScript serão disparadas, não é permitido que avance sem que todos os campos estejam devidamente preenchidos. Se caso houver algum erro de preenchimento, algumas informações de erros serão apresentadas ao usuário, mostrando os campos que devem ser corrigidos.

Se todos os campos foram devidamente preenchidos, a ferramenta avança para a próxima tela (etapa) chamada 'Protonation'.

5.2.4 Etapa Protonation

A etapa consiste em executar o processo de protonação, ou seja, a inserção de hidrogênios, conforme já explicado antes.

Para realização da protonação foi escolhido o sistema H++ e seu site web (ONUFRIEV, 2005). Porém, toda interatividade é feita pelo site e até o presente momento que se confecciona este trabalho eles não possuem softwares e API's desenvolvidas para facilitar a troca de informações com ferramentas externas. Portanto, o ProtCool 2.0 implementa uma *web crawler* (ou *spider bot*) que

extraí informações desejadas do site H++ Para tanto, utilizam-se os aplicativos Selenium (“WebDriver”, [s.d.]) e Geckodriver (“Geckodriver”, 2023). Esses softwares, normalmente, são utilizados para realizar teste automático de sistemas em ambientes web. Dessa forma, a partir de dados fornecidos de entrada, acessa-se remotamente a interface sem interação humana, realiza-se o preenchimento dos dados e dispara-se um comando de execução.

Após o preenchimento dos campos desta etapa é disparado a execução da *web crawler* no *backend*, no qual ele acessa o navegador (servidor), entra com os dados, aguarda o processamento e ao final retorna o PDB gerado. Esse arquivo fica armazenado em uma pasta temporária no servidor que será abordado mais à frente. Entre os campos a serem preenchidos, temos:

- i - [Salinity]: permite informar a salinidade do sistema no H++, sendo o valor padrão no H++: “0.15”.
- ii - [Internal Dielectric]: permite informar o valor da constante ‘Internal Dielectric’ no H++, sendo o valor padrão no H++: “10”.
- iii - [External Dielectric]: permite informar valor da constante ‘External Dielectric’ no H++, sendo o valor padrão: “80”.
- iv - [pH]: permite informar o pH do sistema ao H++, sendo o valor padrão no H++: 6.5.
- v - [Correct Orientation]: permite corrigir a orientação dos grupos ASN, GLN e HIS. Essa opção adiciona os átomos de H, com base nos contatos de van der Waals.

Na versão ProtCool 2.0, os parâmetros abaixo são fixos no valor default:

- vi - [Box]: permite informar o tipo de caixa onde o solvente se encontra. Entre os valores possíveis temos.: ‘No’ (default), ‘Cubic’ ou ‘Octahedral’.
- vii - [Water Model]: permite informar o modelo de água do sistema. Entre os valores possíveis temos.: ‘OPC’ (default) e ‘TIP3P’.
- viii - [Dist Solvente]: permite informar a distância do solvente na caixa. Default em 10.

ix - [Contra Ions]: permite informar se serão adicionados íons. Entre os valores possíveis temos.: “No” (default), “Neutralize using Na⁺/Cl⁻”, “Add the following ions”, “Neutralize and Add ions”.

x - [Ions Add 1]: permite informar qual íon será adicionado. Entre os valores temos.: -- (nenhum, o default), Na⁺, Cl⁻, Ca²⁺, K⁺, Li⁺, Mg²⁺, Mn²⁺.

xii - [Number Add 1]: permite informar o número de íons que serão adicionados. Default em 0.

xiii - [Ions Add 2]: permite informar qual íon será adicionado. Entre os valores temos.: -- (nenhum, o default), Na⁺, Cl⁻, Ca²⁺, K⁺, Li⁺, Mg²⁺, Mn²⁺.

xiv - [Number Add 2]: permite informar o número de íons que serão adicionados. Default em 0.

xv - [Ions Add 3]: permite informar qual íon será adicionado. Entre os valores temos.: -- (nenhum, o default), Na⁺, Cl⁻, Ca²⁺, K⁺, Li⁺, Mg²⁺, Mn²⁺.

xvi - [Number Add 3]: permite informar o número de íons que serão adicionados. Default em 0.

Como toda tela até aqui, ao avançar por meio do botão ‘Next’ é disparado a validação dos campos para garantir o correto preenchimento. Uma vez correto parte-se, para a próxima tela "Solvation and Ionization".

ProtCool - Dynamic Configs Protonation

Salinity: 0.15	Box Solvent: No	Dist Solvent: 10
Internal Dielectric: 10	Water Model: OPC	Contra Ions: No
External Dielectric: 80	Ions Add 1: --	Number Add 1: 0
PH 7	Ions Add 2: --	Number Add 2: 0
<input checked="" type="checkbox"/> Correct Orientation	Ions Add 3: --	Number Add 3: 0

← Prev Next →

Figura 20. Tela para preenchimento dos parâmetros referentes a protonação.

5.2.5 Etapa Solvation/Ionization

Seguindo o *workflow* de preparação da dinâmica, esta etapa é o momento em que o pesquisador informa dos dados referentes à solvatação e ionização. Dentre os campos necessários, o pesquisador deve informar:

i - [Water Box]: permite informar o modelo de água a ser utilizada durante a dinâmica. A versão web do ProtCool aceita apenas o modelo AMBER TIP3PBOX 12, representando águas com 3 pontos de carga (TIP3P) com condições de borda periódica (BOX), em caixas de 12 Å de tamanho. Isso implica numa célula unitária que se repete a cada 12 Å ao longo de cada dimensão.

ii - [Ionize]: permite informar o valor utilizado para realizar a ionização do sistema (fração). O valor default é “0.15”, emulando condições fisiológicas.

iii - [Ions]: permite informar quais íons serão utilizados para ionizar o sistema. Na versão ProtCool 2.0, são default Na⁺ e Cl⁻.

ProtCool - Dynamic Configs Solvation Ionization

Water Box:

Ionize Percentage:

Ions:

#	Ions	Action
0	Na+	<input type="button" value="Delete"/>
1	Cl-	<input type="button" value="Delete"/>

Figura 21. Tela para preenchimento dos parâmetros referentes a solvatação e ionização.

5.2.6 Simulation

A última etapa do ProtCool 2.0 conforme o *workflow* é referente à preparação da simulação. Nela o pesquisador deve informar os parâmetros e as configurações necessárias para a execução a dinâmica molecular pelo NAMD (NELSON, 1996).

Esta etapa pode ser subdividida em outras, conforme desejo do usuário. Em geral, alocam-se uma ou mais etapas para a minimização de energia do sistema, seguida por outras etapas de relaxamento atômico de forma gradual, definindo tempos de simulação para cada uma. Por exemplo: primeiro pode-se restringir todos os átomos da proteína e ligante, relaxando apenas o resto, como água e íons; depois, continuar restringindo os átomos da proteína, mas liberando o ligante; agora, já libera alguns átomos da proteína, mas apenas aqueles mais próximos do ligante, como num raio de 5Å, que não sejam do *backbone*; o mesmo que antes, mas ampliando o raio para 10 Å; liberando todos os átomos, mas restringindo ainda o *backbone*; liberando geral, mas por um certo tempo limitado; se uma análise prévia indicar equilíbrio, libera todo o sistema para rodar a dinâmica molecular efetiva e produtiva num tempo longo.

As subetapas de relaxamento gradual compõem a fase de equilíbrio. Essa liberação gradual dos movimentos visa trazer o sistema para um estado de equilíbrio térmico, eliminando inconsistências nas posições e velocidades do sistema, algo essencial para gerar uma simulação mais realista e confiável. Atingida a equilíbrio, a simulação pode entrar num estado produtivo, num tempo mais longo, que costuma ser o objetivo principal de toda dinâmica molecular. É da fase de produção que vão emergir os principais resultados e análises do comportamento dinâmico do sistema,

que no ProtCool 2.0 representa o complexo proteína-ligante imerso em solvente. Nesta versão do ProtCool, são 14 possibilidades para o campo [Free]:

i - Água livre: – permite liberar ou deixar sem restrição as moléculas de água do sistema. Usa-se o valor.: “Water”.

ii - Íons livres - permite liberar ou deixar sem restrição os íons Na⁺ e Cl⁻ durante a simulação. Usa o valor.: "Íons"

iii - Ligante livre - permite liberar ou deixar sem restrição os ligantes durante a simulação. Usa o valor.: "Ligand”.

iv- Liberar apenas hidrogênios nos ligantes - nesse caso, libera os hidrogênios do ligante, de forma que eles ficam livre e os demais átomos continuam restritos. Usa o valor.: “LigandH”.

v - Cadeia lateral do ligante - informa à simulação que toda cadeia lateral do ligante deve ficar livre. Usa o valor.: “SCLigand”.

vi - Cadeia lateral da proteína - informa à simulação que toda cadeia lateral da proteína deve ficar livre. Usa o valor.: “SCProtein”.

vii - Proteína livre - informa à simulação que a proteína inteira ficará livre. Usa o valor.: “Protein”.

viii - Cadeia da proteína - informa que a cadeia especificada ficará livre. Usa o valor.: “ProteinChain A” ou “ProteinChain X”; no qual X é a cadeia informada.

ix - Cadeia lateral de uma cadeia da proteína - informa que todos os átomos da cadeia lateral de uma das cadeias ficarão livres. Usa o valor: SCProteinChain A ou SCProteinChain A; no qual X é a cadeia informada.

x - Cadeia lateral de um dado resíduo - informa que a cadeia lateral daquele resíduo

especificado ficará livre. Usa o valor: SC TRP A 86.

xi - Resíduo livre - informa que o resíduo especificado fica livre. Usa o valor.: “TYR A 72”.

xii - Liberar N resíduos no início de uma cadeia da proteína - informa a quantidade de resíduos desejado pelo pesquisador no início da cadeia da proteína estará livre. Usa o valor: “ProteinBegin A 5”.

xiii - Liberar N resíduos no fim de uma cadeia da proteína - informa a quantidade de resíduos desejados pelo pesquisador no fim da cadeia da proteína ficará livre. Usa o valor: “ProteinEnd A 10”.

xiv - Tudo livre - todo o PDB fica livre (proteína, ligante, íons e água). Usa o valor: “ALL”.

Nesse ponto, é importante ter atenção, pois é necessário respeitar uma hierarquia no momento de inserir as restrições na tabela ‘Free’, para que não haja sobrescrita de funcionalidades. Como exemplo, se o pesquisador colocar LigandH na 1ª ordem e Ligand na 2ª, o ligante ficará todo livre e não apenas os hidrogênios. Segue hierarquia das restrições.:

1º: Water e Ions – só tem a possibilidade de deixar livre ou não. Não são sobrescritos por nenhum outro a não ser quando coloca a opção “ALL”.

2º: Ligante – As opções são “LigandH”, “SCLigand” e “Ligand”. Nessa ordem de sobrescrita, lembrando que o “SCLigand” só ocorre no caso de ligantes do tipo peptídico. Sobrescrito quando seleciona “ALL”.

3º: Resíduo – SC TRP A 86, TRP A 72. Nessa ordem de sobrescrita. Lembrando que se colocar proteína toda livre sobrescreve os dois, porém, se colocar proteína “SC TRP A 72” e “TRP A 72”, a proteína ficará com a cadeia lateral livre mais o resíduo desejado completamente livre. Sobrescrito quando seleciona ALL. Ficar atento ao selecionar junto qualquer opção de cadeia e proteína, pois pode afetar o resultado.

4º: Cadeia da Proteína – “SCProteinChain A”, “ProteinBegin A 5”, “ProteinEnd A 10”, “ProteinChain A”. Considerando os itens que levam em consideração as cadeias das proteínas, esta é a ordem se sobrescrita dentro da cadeia. Porém, como são parte da proteína, no caso de selecionar a restrição da proteína inteira, ela é sobrescrita, ou seja, mesmo falando que queira a restrição apenas da cadeia lateral da cadeia A da proteína, se colocar também a opção Protein, toda a proteína ficará livre. Sobrescrito quando seleciona “ALL”. Ficar atento ao selecionar junto qualquer opção de proteína e resíduo, pois pode afetar o resultado.

5º: Proteína – “SCProtein”, “Protein”. Nessa ordem de sobrescrição. Sobrescrito quando seleciona ALL. Ficar atento ao selecionar junto qualquer opção de cadeia e resíduo, pois pode afetar o resultado.

The screenshot shows the 'ProtCool - Dynamic Configs' interface. At the top right, there is a 'Simulation' button. Below the title bar, there are several configuration sections:

- NAMD CPU Cluster:** A dropdown menu set to '24'.
- Toggle switches:** 'Cuda' and 'Start' are both turned on.
- Temperature:** A text input field containing '300'.
- Free:** A dropdown menu set to 'Water'.
- First Time Step:** A text input field containing '0'.
- Buttons:** An 'Increase' button with a plus icon is located between the 'Free' and 'First Time Step' fields.

Below these fields is a table with three columns: '#', 'Free', and 'Action'. The table contains three rows:

#	Free	Action
0	Water	Delete
1	Ligand	Delete
2	SCProtein	Delete

Below the table, there are several more configuration fields arranged in a grid:

- Scnb:** 2
- Consexp:** 2
- Conskcol:** 8
- Exclude:** scaled1-4
- Scaling:** 1
- Cutoff:** 12
- Switchdist:** 10
- Pairlistdist:** 13.5
- Timestep:** 2.0 ;# 2.0 fs/step
- RigidBonds:** all ;# needed for 2fs steps
- NonbondedFreq:** 1
- FullElectFrequency:** 2
- Stepspercycle:** 10
- Langevin:** on ;# do langevin dynamics
- LangevinDamping:** 2 ;# damping coefficient (gamma) of 2/ps
- LangevinHydrogen:** off ;# don't couple langevin bath to hydrogens
- Use Group Pressure:** yes ;# needed for rigidBonds
- langevinPistonTarget:** 1.01325 ;# in bar -> 1 atm

Figura 22. Parte da tela para preenchimento dos parâmetros referentes a simulação.

Após o preenchimento da tabela ‘Free’ o pesquisador deve informar o restante dos parâmetros referente à simulação em geral. Alguns campos já são apresentados preenchidos com valores default, sendo alguns valores fixos na versão ProtCool 2.0. Geralmente, trata-se de parâmetros que serão usados na maioria das simulações. Vale ressaltar que as propriedades, formatações e correto

preenchimento desses campos segue rigorosamente os padrões informados na documentação da ferramenta NAMD (NELSON, 1996), ou seja, para maiores detalhes sobre o correto preenchimento e opções permitidas, deve-se consultar a documentação desta ferramenta. A seguir é apresentado as instruções para preenchimento dos campos.:

i - [Temperature] – permite informar a temperatura inicial (em Kelvin) para o sistema. Esta opção irá gerar uma distribuição de velocidade aleatória para as velocidades iniciais de todos os átomos de modo que o sistema esteja na temperatura desejada. Exemplo: ‘300’.

ii - [First Time Step] - permite informar o intervalo de tempo em femtossegundos a ser utilizado no início das simulações. Constitui o passo de tempo no qual posições e velocidades dos átomos são integradas. Para assegurar maior estabilidade do sistema durante equilíbrio, os passos iniciais podem ser menores que os das fases de produção. Se for zero, assume-se direto a fase de produção. O valor default é zero, e não deve ser maior que o *Timestep*.

iii - [Scnb] - permite informar o fator de escala aplicado às forças de van der Waals e Coulombicas. Valor padrão: ‘2.0’.

iv - [Consexp] - permite informar o valor usado como expoente na equação de força da restrição harmônica. O valor padrão para a função de restrição de energia harmônica é ‘2’, ou seja, um expoente quadrático.

v - [Conskcol] - permite informar em que coluna buscar a restrição harmônica no arquivo PDB. Valores aceitáveis são X, Y, Z, O ou B, indicando as coordenadas (x,y,z), ocupâncias ou beta-coupling (um fator relacionado à temperatura ou grau de agitação do átomo). Valor default é “B”.

vi - [Exclude] - permite informar quais pares de átomos ligados devem ser excluídos das interações não ligadas. Nesse campo, temos várias possibilidades que serão relatadas a seguir.

- ‘none’: nenhum par de átomos ligados será excluído.
- ‘1-2’: todos os pares de átomos diretamente conectados por meio de uma ligação linear serão excluídos.

- ‘1-3’: todos os pares 1-2 serão excluídos junto com todos os pares de átomos que estão ligados a um terceiro átomo comum.
- ‘1-4’: todos os pares 1-3 serão excluídos junto com todos os pares de átomos conectados por um conjunto de duas ligações.
- ‘scaled1-4’: todos os pares 1-3 são excluídos e todos os pares que correspondem aos critérios 1-4 são modificados.

Valor default: ‘scaled1-4’.

vii - [Scaling] - permite informar o parâmetro que contém o fator de escala para interações eletrostáticas 1-4. Este fator é usado apenas quando o parâmetro 'Excluído' é definido como scaled1-4. Valor default é ‘1.0’.

viii - [Cutoff] - permite informar a distância máxima a partir do qual interações serão negligenciadas para ambos os cálculos eletrostáticos e de van der Waals. Valor default é 12 Å.

vix - [Switchdist] - permite informar a distância na qual a função *switching* vai a zero. Essa função gradualmente decresce as forças de van der Waals até atingir zero em “switchdist”. Portanto, o valor de “switchdist” deve ser sempre menor que o de “Cutoff”. Valor default é 10.

x - [Pairlistdist] - Informa a distância a partir da qual todos os pares de átomos dentro dessa distância formarão uma lista para cálculos de interações. Pares de átomos com distâncias maiores serão ignorados. Valor default: 13.5 Å.

xi - [Timestep] - Constitui o passo de tempo no qual posições e velocidades dos átomos são integradas num regime em produção. Valor default: 2.0 femtossegundos.

xii - [rigidBonds] - Parâmetro que define limitações ao movimento de hidrogênios ligados à certos grupos de átomos. Valor default é “all”, o que faz que fiquem rígidos os hidrogênios das águas e de outros átomos aos quais possam estar ligados.

xiii - [NonbondedFreq] - permite informar o número de *TimeSteps* que interações não-ligadas de curto alcance devem ser calculadas. Default: 1.

xiv - [FullElectFrequency] - permite informar o número de *TimeSteps* que interações eletrostáticas devem ser calculadas. Default: 2.

xv - [Stepspercycle] - Define o número de *TimeSteps* depois do qual a lista de pares de átomos será atualizada. Default: 10.

xvi - [Langevin] - permite informar se a dinâmica de Langevin está ativa ou não. Essa dinâmica inclui termos estocásticos nas equações do movimento no intuito de simular flutuações térmicas e dissipações no sistema. Default: on.

xvii - [LangevinDamping] - É um valor associado ao ajuste da “damping force” nas dinâmicas de *Langevin*. Confere viscosidade ao sistema, atuando como uma força friccional, contrária ao movimento. Default: 2.

xviii - [LangevinHydrogen] - permite informar se a dinâmica de Langevin será aplicada nos átomos de hidrogênio ou não. Default: ‘off’.

xix - [Use Group Pressure] - permite informar se a pressão será calculada utilizando-se a energia atômica virial e cinética (o padrão), ou se será usada uma energia virial pseudo-molecular baseada em grupos de hidrogênio. Default: “yes”.

xx - [langevinPistonTarget] - permite informar o valor da pressão desejada para o “pistão” de Langevin, técnica usada para manter a pressão constante variando volumes de uma câmara virtual envolta ao sistema. Uma pressão típica seria 1.01325 bar, que corresponde à pressão atmosférica ao nível do mar.

xxi - [LangevinPistonPeriod] - Controla a frequência com que o volume da câmara virtual será atualizado em simulações de pressão constante. Default: 100 (fs).

xxii - [LangevinPistonDecay] - Determina quão rápida a alteração de volume da câmara

virtual retorna ao seu valor de equilíbrio. Geralmente, é um valor igual ou menor que 'LangevinPistonPeriod'. Default: 50 (fs).

xxii - [RestartFreq] - permite informar a frequência com que as simulações serão salvas em disco como um arquivo de reinicialização. Esse arquivo pode ser usado para reiniciar a simulação a partir daquele ponto, evitando sua reexecução a partir do início de novo. Default: 1000 *TimeSteps*.

xxiii - [dcdfreq] - permite informar a frequência com que as simulações serão salvas em disco como um arquivo binário formato DCD (*Double-precision Coordinate Dump*). Esses arquivos são úteis para as análises de trajetórias. Default: 2000 *TimeSteps*.

xxiv - [outputEnergies] - permite informar a frequência com que informações envolvendo as energias da simulação serão salvas. Default: 1000 *TimeSteps*.

xxv - [outputPressure] - permite informar a frequência com que informações envolvendo as pressões da simulação serão salvas. Default: 1000 *TimeSteps*.

xxvi - [Minimize] - permite informar será feita minimização de energia ou não. Para configurar como sim, basta informar o número de passos na minimização. Para não operar minimizações, coloque "No". Default: 1000.

xxvii - [Reinitvels] - Informa se as velocidades iniciais deverão ser reinicializadas randomicamente ou não a cada retomada de uma simulação já em curso. Default: "No".

xxviii - [Simulation]: permite informar o número de *TimeSteps* total de uma simulação em produção. Se esse campo estiver vazio, não terá tempo determinado de fim.

Os campos seguintes da tela são todos "*toggle buttons*", eles foram criados para permitirem informações booleanas como 'On' e 'Off' ou 'Yes' ou 'No', conforme se pede na documentação do NAMD. Esses campos coordenam se os campos já informados acima serão utilizados ou não.

xxix - [Structure] - permite informar se os arquivos “.psf”. terão os nomes colocados automaticamente pelo sistema. Default: “On”.

xxx - [Restart] - permite informar se a simulação vai reiniciar a partir de uma simulação anterior. Default: “On”.

xxxi - [Amber] - permite informar se deve usar o campo de força de formato AMBER (CASE, 2008). Default: “On”.

xxxii - [Ambercoord] - permite informar o parâmetro que contém o arquivo de coordenadas no formato AMBER. Esse componente ligado indica que ele será preenchido automaticamente pelo ProtCool 2.0. Default: “Off”.

xxxiii - [Constraints] - permite informar o valor usado na restrição harmônica. Componente ligado define se será utilizada a restrição harmônica. Default: “On”.

xxxiv - [Read Exclusions] - permite informar se as exclusões do arquivo PARM serão lidas. O arquivo PARM fornece explicitamente informações de exclusão completa (incluindo 1-4 exclusões). Quando ‘Read Exclusions’ estiver ativado, o NAMD lerá todas as exclusões do arquivo PARM e não adicionará mais nenhuma; alternativamente, se “Read Exclusions” estiver desativado, o NAMD ignorará as exclusões no arquivo PARM e as gerará automaticamente de acordo com a política de exclusão especificada em ‘Exclude’. Default: “On”.

xxxv - [Para Type Charmm] - permite informar se os arquivos de parâmetro estão ou não no formato CHARMM (BROOKS, 1983). O formato X-PLOR é o padrão para arquivos de parâmetros. Se tiver desativado, não será considerado. Default: “Off”.

xxxvi - [Parameters] - permite ao Protcool 2.0 colocar o arquivo de parâmetros no arquivo de configuração NAMD. Se tiver desativado, não será considerado. Se tiver ativado, o sistema coloca o valor do nome do arquivo. Default: “Off”.

xxxvii - [Simulation Temperature] - permite informar se a temperatura será habilitada por meio da ativação do componente. Normalmente é ativado na primeira simulação e desativado na última, indicando como o NAMD tratará de onde recuperará a informação na simulação. Default: “On”.

xxxviii - [Switching] - Se esse parâmetro estiver ativado, funções de suavização serão aplicadas às forças eletrostática e de van der Waals. Default: “On”.

xxxixL - [PME] - permite informar se a Malha ou Gride de Partículas Ewald (*Particle Mesh Ewald*) para cálculo de forças eletrostáticas será usada como técnica de condição periódica de controle. Default: “On”.

xL - [Use Flexible Cell] - permite informar se a célula periódica poderá flutuar independentemente ao longo de suas três posições ortogonais. Default: “Off”.

xLi - [Use Constant Area] - permite informar se o NAMD deverá manter as dimensões da célula unitária constantes enquanto permite flutuações ao longo do eixo z. Default: “Off”.

xLii - [LangevinPiston] - permite informar se o controle de pressão do pistão de Langevin deve estar ativo ou não. Default: “On”.

xLiii - [wrapAll] - permite envolver todas as coordenadas em torno de limites periódicos. Esta opção aplica uma translação ao centro de massa de cada molécula ou sob aglomerado contíguo de átomos ligados para mantê-lo dentro da célula unitária. Default: “On”.

#	Simulation	Free	Minimize	Action
1	100000 ; #200 ps	Water; Ligand; SCProtein;	1000	Delete
2	100000 ; #200 ps	Water;	1000	Delete
3	100000 ; #200 ps	Ligand; Water;	1000	Delete

Figura 23. Parte final da tela para preenchimento dos parâmetros referentes a simulação.

Com todos os campos da etapa ‘Simulation’ preenchidos, o pesquisador pode agora guardar todas estas informações ao pressionar o botão ‘Simulation’. Uma instância de simulação será adicionada em tabela. Pode-se inserir quantas simulações desejar com diferentes parâmetros selecionados conforme as estratégias de minimização, de etapas de equilíbrio e de produção planejadas pelo pesquisador.

Figura 24. Execução final da preparação da dinâmica molecular.

Como ação final do ProtCool 2.0 o botão ‘Execute’ ficará ativo. Basta o pesquisador executá-lo e todo processamento de preparação da dinâmica molecular será disparado (*backend*). Durante o

processamento, é possível acompanhar em tempo real todas as etapas de execução em uma janela modal. No final é disponibilizado para download o arquivo compactado com todos os arquivos de preparação necessários para execução da dinâmica molecular efetiva.

6 ESTUDO DE CASO

Neste capítulo, será demonstrado um estudo de caso do uso do ProtCool 2.0. Foi realizada a preparação de um protocolo de ancoragem e dinâmica molecular para o complexo envolvendo a proteína acetilcolinesterase (4EY6) com o ligante galantamina (GNT), utilizado no tratamento do Alzheimer. A escolha dessas moléculas deveu-se ao fato de possibilitar uma validação do ProtCool 2.0 por replicação de uma simulação certificada por pares em publicação internacional (ROCHA, 2017).

A galantamina liga-se à acetilcolinesterase, inibindo sua ação e elevando os níveis do neurotransmissor acetilcolina no cérebro, com efeitos positivos na capacidade cognitiva e de memória em pacientes com Alzheimer. Em (ROCHA, 2017), foi feita uma simulação de dinâmica molecular desse complexo, cuja etapa preparatória envolveu scripts ad hoc dos próprios autores do artigo que culminaram num complexo protocolo de 21 passos.

Aqui serão apresentados como foi feito o preenchimento do ProtCool 2.0, detalhando todo o *workflow* de preparação da dinâmica molecular para o complexo Acetilcolinesterase-Galantamina. Posteriormente serão realizadas análises em cima das dinâmicas efetuadas a partir dos arquivos de preparação gerados pelo ProtCool 2.0. Será possível demonstrar ao final que a preparação dos arquivos de fato permitiu a simulação de dinâmicas computacionais confiáveis, reproduzindo os resultados de (Rocha, 2017)

1. General Info:

- i – [Researcher]: ‘Moisés Souza’
- ii – [Email Address]: ‘moises.souza@ifsuldeminas.edu.br’
- iii - [Language]: ‘Inglês’
- iv – [PDB]: ‘4EY6’ (acetilcolinesterase)
- v – [Define workflow steps]:
 - Prepare (On),
 - Protonation (On),
 - Prepared Namd (On),

- Molecular Model (On),
- Execution Force Field (On),
- Docking (On),
- Vina (On).

vi – [Chains]: ‘A’

vii – [Disulfide Bridges]: ‘CYS A 69 - CYS A 96’, ‘CYS A 257 – CYS A 272’, ‘CYS A 409 – CYS A 529’.

Figura 25. Preenchimento dos campos da etapa ‘General Info’ (4EY6 - GNT).

2. Ligands:

i – [ZINC]: ‘On’

ii – [Ligand]: ‘ZINC00491073’

iii – [Name Ligand]: ‘gnt’ (galantamina)

iv - [Ligand Code]: ‘GNT’ (galantamina)

v – [NumDoc]: ‘1’

vi – [Force Field]: ‘~/amber18/dat/leap/cmd/oldff/leaprc.ff99SB’

ProtCool - Dynamic Configs Ligands

ZINC File/MOL2

Ligand: Name Ligand:

Ligand Code: NumDoc:

Force Field:

#	Ligands	Name Ligand	Code Ligand	Force Field	Num Doc	Action
0	ZINC00491073	gnt	GNT	~/amber18/dat/leap/cmd/oldff/leaprc.ff995B	1	<input type="button" value="Delete"/>

Figura 26. Preenchimento dos campos da etapa ‘Ligands’ (4EY6 - GNT).

3. Docking:

- i – [Center X]: ‘-14.041’
- ii – [Center Y]: ‘-42.654’
- iii – [Center Z]: ‘28.419’
- iv - [Size X]: ‘25’
- v - [Size Y]: ‘25’
- v – [Size Z]: ‘25’
- vi – [Flex]: ‘On’
- vii - [Num Modes]: ‘On’ e ‘10’
- viii - [CPU Number]: ‘1’
- ix - [Flex Docking]: ‘A:TRP86’, ‘A:TYR124’, ‘A:SER203’, ‘A:TYR337’, ‘A:HIS:447’

ProtCool - Dynamic Configs Docking

Center X: -14,041 Center Y: -42,654 Center Z: 28,419

Size X: 25 Size Y: 25 Size Z: 25

Consider Hydrogens In The Protein
 Consider Hydrogens In The Ligand
 Exhaustiveness
 Energy Range
 Flex
 Num Modes

Num Modes:
 CPU Number:

Chains:

#	Flex Docking	Action
0	A:TRP86	Delete
1	A:TYR124	Delete
2	A:SER203	Delete
3	A:TYR337	Delete
4	A:HIS447	Delete

Figura 27. Preenchimento dos campos da etapa ‘Docking’ (4EY6 - GNT).

4. Protonation:

- i – [Salinity]: ‘0.15’
- ii – [Internal Dielectric]: ‘10’
- iii – [External Dieletrica]: ‘80’
- iv - [pH]: ‘7’
- v - [Box Solvent]: ‘No’
- v – [Water Model]: ‘OPC’
- vi – [Ions Add 1]: Não informado
- vii - [Ions Add 2]: Não informado
- viii – [Ions Add 3]: Não informado
- ix - [Dist Solvente]: ‘10’
- x – [Contra Ions]: ‘No’
- vi – [Number Add 1]: ‘0’
- vii - [Number Add 2]: ‘0’
- viii – [Number Add 3]: ‘0’

Figura 28. Preenchimento dos campos da etapa ‘Protonation’ (4EY6 - GNT).

5. Solvation / Ionization:

- i – [Water Box]: ‘TIP3BOX 12’
- ii – [Ionize Percentage]: ‘0.15’
- iii – [Ions]: ‘Na+’, ‘Cl-’

#	Ions	Action
0	Na+	Delete
1	Cl-	Delete

Figura 29. Preenchimento dos campos da etapa Solvation/Ionization (4EY6 - GNT).

6. Simulation:

Essa etapa foi inserida várias configurações para cada etapa da simulação desejada, ao todo 9 etapas. Na figura 29 os campos destacados em vermelho são os parâmetros que permaneceram comuns à todas as simulações. Ao passo que os campos destacados em verde houve alterações de simulação para simulação.

1º - Simulação - Minimização

Nesta etapa é feita apenas a minimização de energia, em 1000 passos. Essa minimização faz

com que possíveis falhas nos posicionamentos dos átomos que estavam no PDB original ou sejam decorrentes das ações do *workflow* das etapas anteriores possam ser mitigadas. Átomos anormalmente próximos geram energias altas, e a minimização dessas energias pode reposicionar esses átomos em condições mais realistas.

Abaixo, os parâmetros desta etapa:

i – [NAMD CPU Cluster]: ‘24’

ii – [Cuda]: ‘On’

iii – [Start]: ‘On’

iv - [Temperature]: ‘300’

v - [Free]: Para cada simulação foi inserida parâmetros diferentes. Na 1ª simulação, foram apenas inseridos a liberação de águas e íons.: ‘Water’, ‘Ions’

vi - [First Time Step]: ‘0’

vii - [Scnb]: ‘2’

viii - [Conexp]: ‘2’

ix - [Conskcol]: ‘12’

x - [Exclude]: ‘scaled1-4’

xi - [Scaling]: ‘1’

xii - [Cutoff]: ‘12’

xiii - [Switchdist]: ‘10’

xiv - [Pairlistdist]: ‘13,5’

xv - [Timestep]: ‘2.0 ;#2.0 fs/step’

xvi - [RigidBonds]: ‘all ;# needed for 2fs steps’

xvii - [NobondedFreq]: ‘1’

xviii - [FullElectFrequency]: ‘2’

xix - [Stepspercycle]: ‘10’

xx - [Langevin]: ‘on ;# do langevindynamics’

xxi - [LangevinDamping]: ‘2 ;# damping coefficient (gamma) of 2/ps’

xxii - [LangevinHydrogen]: ‘off ;# don’t couple langevin bath to hydrogens’

xxiii - [Use Group Pressure]: 'yes ;# needed for rigidBonds'
xxiv - [langevinPistonTarget]: '1.01325 ;# in bar -> 1 atm'
xxv - [LangevinPistonPeriod]: '100'
xxvi - [LangevinPistonDecay]: '50'
xxvii - [RestartFreq]: '1000 ;# 1000steps = every 2 ps'
xxviii - [dcdfreq]: '2000'
xxix - [outputEnergies]: '1000'
xxx - [outputPressure]: '2000'
xxxi - [Minimize]: '1000'
xxxii - [Reinitvels]: 'No'
xxxiii - [Simulation]: 'No'
xxxiv - [Restart]: 'Off'
xxxv - [Amber]: 'On'
xxxvi - [Constrainst]: 'On'
xxxvii - [Read Exclusions]: 'On'
xxxviii - [Simulation Temperature]: 'On'
xxxix - [Switching]: 'On'
xL - [PME]: 'On'
xLi - [Langevin Piston]: 'On'
xLii - [wrapAll]: 'On'

ProtCool - Dynamic Configs Simulation

NAMD CPU Cluster:

 Cuda
 Start

Temperature: Free: First Time Step:

#	Free	Action
0	<input type="text" value="Water;ions"/>	<input type="button" value="Delete"/>

Scnb: Consexp: Conskcol:
 Exclude: Scaling: Cutoff:
 Switchdist: Pairlistdist: Timestep:
 RigidBonds: NonbondedFreq: FullElectFrequency:
 Stepspercycle: Langevin: LangevinDamping:
 LangevinHydrogen: Use Group Pressure: langevinPistonTarget:
 LangevinPistonPeriod: LangevinPistonDecay: RestartFreq:
 dcdfreq: outputEnergies: outputPressure:
 Minimize: Reinitvels: Simulation:
 Structure Restart Amber AmberCoord Constraints
 Read Exclusions Para Type Charmm Parameters Simulation Temperature Switching
 PME Use Flexible Cell Use Constant Area Langevin Piston wrapAll

Figura 30. Preenchimento dos campos da etapa ‘Simulation’ (4EY6 - GNT).

Para os parâmetros da 2ª simulação até a última e 9ª simulação apenas as opções descritas abaixo são alteradas como:

2º - Simulação - águas e íons livres

Esta etapa dá início ao processo de equilibração do sistema. Todos os átomos do sistema são mantidos fixos, exceção às águas e íons na caixa que engloba o complexo proteína-ligante. Dado que cada *TimeStep* foi configurado para 2 femtossegundos, foram executados 100000 *TimeSteps*, ou 200 picossegundos (ps) de águas e íons livres. O “Restart” em “On” garante que serão mantidas as posições e velocidades da etapa anterior.

Abaixo, os parâmetros desta etapa:

i – [Free]: ‘Ions, Water’

- ii - [Minimize]: 'No'
- iii - [Restart]: 'On'
- iv - [Simulation]: '100000 ; #200 ps'

3º - Simulação - ligantes, águas e íons livres

Agora, além de águas e íons, ficam irrestritos também os átomos do(s) ligante(s), tomando outros 200 ps de simulação.

- i - [Free]: 'Ligand, Ions, Water'
- ii - [Minimize]: 'No'
- iii - [Restart]: 'On'
- iv - [Simulation]: '100000 ; #200 ps'

4º - Simulação - átomos da proteína a 5A do ligante

Nessa etapa, além dos átomos da etapa anterior, são liberados átomos da cadeia lateral dos resíduos da proteína alvo que estejam a menos de 5A de distância do ligante, ou seja, os átomos mais próximos. Mais 300 ps de simulação.

i - [Free]: 'SC ASP A 74, SC TRP A 86, SC GLY A 120, SC GLY A 121, SC GLY A 122, SC TYR A 124, SC SER A 125, SC TYR A 133, SC GLU A 202, SC SER A 203, SC ALA A 204, SC TRP A 236, SC PHE A 295, SC PHE A 297, SC TYR A 337, SC PHE A 338, SC TYR A 341, SC HIS A 447, SC GLY A 448, SC ILE A 451, Ligand, Ions, Water' - (Resíduos a 5 Å de distância do ligante.)

- ii - [Minimize]: 'No'
- iii - [Restart]: 'On'
- iv - [Simulation]: '150000 ; #300 ps'

5º - Simulação - átomos da proteína a 10A do ligante

Abre-se mais o raio do que ficará irrestrito. Além dos átomos da etapa anterior, são liberados átomos da cadeia lateral dos resíduos da proteína alvo que estejam a menos de 10A de distância do ligante. Outros 300 ps de simulação.

- i - [Free]: 'SC GLN A 71, SC TYR A 72, SC VAL A 73, SC ASP A 74, SC THR A 75, SC

LEU A 76, SC TYR A 77, SC PHE A 80, SC GLU A 81, SC GLY A 82, SC THR A 83, SC GLU A 84, SC MET A 85, SC TRP A 86, SC ASN A 87, SC PRO A 88, SC TRP A 117, SC ILE A 118, SC TYR A 119, SC GLY A 120, SC GLY A 121, SC GLY A 122, SC PHE A 123, SC TYR A 124, SC SER A 125, SC GLY A 126, SC ALA A 127, SC LEU A 130, SC VAL A 132, SC TYR A 133, SC TYR A 151, SC VAL A 153, SC GLY A 201, SC GLU A 202, SC SER A 203, SC ALA A 204, SC GLY A 205, SC ALA A 206, SC ALA A 207, SC GLN A 228, SC SER A 229, SC GLY A 230, SC ALA A 231, SC GLY A 234, SC PRO A 235, SC TRP A 236, SC ALA A 237, SC GLU A 285, SC TRP A 286, SC VAL A 294, SC PHE A 295, SC ARG A 296, SC PHE A 297, SC SER A 298, SC VAL A 300, SC VAL A 331, SC GLU A 334, SC GLY A 335, SC TYR A 337, SC PHE A 338, SC VAL A 340, SC TYR A 341, SC VAL A 402, SC ASN A 406, SC VAL A 407, SC TRP A 439, SC PRO A 446, SC HIS A 447, SC GLY A 448, SC TYR A 449, SC GLU A 450, SC ILE A 451, SC GLU A 452, Ligand, Ions, Water' - (Resíduos a 10Å distancias do ligante).

ii - [Minimize]: 'No'

iii - [Restart]: 'On'

iv - [Simulation]: '150000 ; #300 ps'

6º - Simulação - resíduos da proteína a 10 A do ligante

A etapa anterior deixou irrestrito todos os átomos da proteína a 10A do ligante. Agora, estende-se para todos os átomos dos resíduos que tenha pelo menos um átomo a 10A do ligante, incluindo *backbone*. Também 300 ps de simulação.

i – [Free]: 'GLN A 71, TYR A 72, VAL A 73, ASP A 74, THR A 75, LEU A 76, TYR A 77, PHE A 80, GLU A 81, GLY A 82, THR A 83, GLU A 84, MET A 85, TRP A 86, ASN A 87, PRO A 88, TRP A 117, ILE A 118, TYR A 119, GLY A 120, GLY A 121, GLY A 122, PHE A 123, TYR A 124, SER A 125, GLY A 126, ALA A 127, LEU A 130, VAL A 132, TYR A 133, TYR A 151, VAL A 153, GLY A 201, GLU A 202, SER A 203, ALA A 204, GLY A 205, ALA A 206, ALA A 207, GLN A 228, SER A 229, GLY A 230, ALA A 231, GLY A 234, PRO A 235, TRP A 236, ALA A 237, GLU A 285, TRP A 286, VAL A 294, PHE A 295, ARG A 296, PHE A 297, SER A 298, VAL A 300, VAL A 331, GLU A 334, GLY A 335, TYR A 337, PHE A 338, VAL A 340, TYR A 341, VAL A 402, ASN A 406, VAL A 407, TRP A 439, PRO A 446, HIS A 447, GLY A 448, TYR A 449, GLU A 450, ILE A 451, GLU A 452, Ligand, Ions, Water' - (Resíduos a 10A distancias do ligante).

ii - [Minimize]: 'No'

iii - [Restart]: 'On'

iv - [Simulation]: '150000 ; #300 ps'

7º - Simulação - cadeia lateral dos resíduos da proteína a 10 A do ligante

Volta a restringir os *backbones*, liberando apenas a cadeia lateral dos resíduos a 10 A do

ligante. 300 ps de simulação.

i – [Free]: ‘SCProtein, GLN A 71, TYR A 72, VAL A 73, ASP A 74, THR A 75, LEU A 76, TYR A 77, PHE A 80, GLU A 81, GLY A 82, THR A 83, GLU A 84, MET A 85, TRP A 86, ASN A 87, PRO A 88, TRP A 117, ILE A 118, TYR A 119, GLY A 120, GLY A 121, GLY A 122, PHE A 123, TYR A 124, SER A 125, GLY A 126, ALA A 127, LEU A 130, VAL A 132, TYR A 133, TYR A 151, VAL A 153, GLY A 201, GLU A 202, SER A 203, ALA A 204, GLY A 205, ALA A 206, ALA A 207, GLN A 228, SER A 229, GLY A 230, ALA A 231, GLY A 234, PRO A 235, TRP A 236, ALA A 237, GLU A 285, TRP A 286, VAL A 294, PHE A 295, ARG A 296, PHE A 297, SER A 298, VAL A 300, VAL A 331, GLU A 334, GLY A 335, TYR A 337, PHE A 338, VAL A 340, TYR A 341, VAL A 402, ASN A 406, VAL A 407, TRP A 439, PRO A 446, HIS A 447, GLY A 448, TYR A 449, GLU A 450, ILE A 451, GLU A 452, Ligand, Ions, Water’

ii - [Minimize]: ‘No’

iii - [Restart]: ‘On’

iv - [Simulation]: ‘150000 ; #300 ps’

8º - Simulação - pré-produção

Libera geral, numa pré-produção de 2 milhões de *TimeSteps*, ou 4 nanossegundos (ns). A intenção aqui é verificar se o sistema atingiu uma equilibrção adequada.

i – [Free]: ‘ALL’

ii - [Minimize]: ‘No’

iii - [Restart]: ‘On’

iv - [Simulation]: ‘2000000 ; #4 ns’

9º - Simulação - produção

Agora sim, vem a preparação da simulação produtiva efetiva num tempo mais longo, no caso, 10 milhões de *TimeSteps* ou 20 ns.

i – [Free]: ‘ALL’

ii - [Minimize]: ‘No’

iii - [Restart]: ‘On’

iv - [Simulation]: ‘10000000 ; #20 ns’

#	Simulation	Free	Minimize	Action
1	No	Water; Ions;	1000	Delete
2	100000 ; #200 ps	Ions; Water;	No	Delete
3	100000 ; #200 ps	Ligand; Ions; Water;	No	Delete
4	150000 ; #300 ps	SC ASP A 74; SC TRP A 86; SC GLY A 120; SC GLY A 121; SC GLY A 122; SC TYR A 124; SC SER A 125; SC TYR A 133; SC GLU A 202; SC SER A 203; SC ALA A 204; SC TRP A 236; SC PHE A 295; SC PHE A 297; SC TYR A 337; SC PHE A 338; SC TYR A 341; SC HIS A 447; SC GLY A 448; SC ILE A 451; Ligand; Ions; Water;	No	Delete
5	150000 ; #300 ps	SC GLN A 71; SC TYR A 72; SC VAL A 73; SC ASP A 74; SC THR A 75; SC LEU A 76; SC TYR A 77; SC PHE A 80; SC GLU A 81; SC GLY A 82; SC THR A 83; SC GLU A 84; SC MET A 85; SC TRP A 86; SC ASN A 87; SC PRO A 88; SC TRP A 117; SC ILE A 118; SC TYR A 119; SC GLY A 120; SC GLY A 121; SC GLY A 122; SC PHE A 123; SC TYR A 124; SC SER A 125; SC GLY A 126; SC ALA A 127; SC LEU A 130; SC VAL A 132; SC TYR A 133; SC TYR A 151; SC VAL A 153; SC GLY A 201; SC GLU A 202; SC SER A 203; SC ALA A 204; SC GLY A 205; SC ALA A 206; SC ALA A 207; SC GLN A 228; SC SER A 229; SC GLY A 230; SC ALA A 231; SC GLY A 234; SC PRO A 235; SC TRP A 236; SC ALA A 237; SC GLU A 285; SC TRP A 286; SC VAL A 294; SC PHE A 295; SC ARG A 296; SC PHE A 297; SC SER A 298; SC VAL A 300; SC VAL A 331; SC GLU A 334; SC GLY A 335; SC TYR A 337; SC PHE A 338; SC VAL A 340; SC TYR A 341; SC VAL A 402; SC ASN A 406; SC VAL A 407; SC TRP A 439; SC PRO A 446; SC HIS A 447; SC GLY A 448; SC TYR A 449; SC GLU A 450; SC ILE A 451; SC GLU A 452; Ligand; Ions; Water;	No	Delete
6	150000 ; #300 ps	GLN A 71; TYR A 72; VAL A 73; ASP A 74; THR A 75; LEU A 76; TYR A 77; PHE A 80; GLU A 81; GLY A 82; THR A 83; GLU A 84; MET A 85; TRP A 86; ASN A 87; PRO A 88; TRP A 117; ILE A 118; TYR A 119; GLY A 120; GLY A 121; GLY A 122; PHE A 123; TYR A 124; SER A 125; GLY A 126; ALA A 127; LEU A 130; VAL A 132; TYR A 133; TYR A 151; VAL A 153; GLY A 201; GLU A 202; SER A 203; ALA A 204; GLY A 205; ALA A 206; ALA A 207; GLN A 228; SER A 229; GLY A 230; ALA A 231; GLY A 234; PRO A 235; TRP A 236; ALA A 237; GLU A 285; TRP A 286; VAL A 294; PHE A 295; ARG A 296; PHE A 297; SER A 298; VAL A 300; VAL A 331; GLU A 334; GLY A 335; TYR A 337; PHE A 338; VAL A 340; TYR A 341; VAL A 402; ASN A 406; VAL A 407; TRP A 439; PRO A 446; HIS A 447; GLY A 448; TYR A 449; GLU A 450; ILE A 451; GLU A 452; Ligand; Ions; Water;	No	Delete
7	150000 ; #300 ps	SCProtein; GLN A 71; TYR A 72; VAL A 73; ASP A 74; THR A 75; LEU A 76; TYR A 77; PHE A 80; GLU A 81; GLY A 82; THR A 83; GLU A 84; MET A 85; TRP A 86; ASN A 87; PRO A 88; TRP A 117; ILE A 118; TYR A 119; GLY A 120; GLY A 121; GLY A 122; PHE A 123; TYR A 124; SER A 125; GLY A 126; ALA A 127; LEU A 130; VAL A 132; TYR A 133; TYR A 151; VAL A 153; GLY A 201; GLU A 202; SER A 203; ALA A 204; GLY A 205; ALA A 206; ALA A 207; GLN A 228; SER A 229; GLY A 230; ALA A 231; GLY A 234; PRO A 235; TRP A 236; ALA A 237; GLU A 285; TRP A 286; VAL A 294; PHE A 295; ARG A 296; PHE A 297; SER A 298; VAL A 300; VAL A 331; GLU A 334; GLY A 335; TYR A 337; PHE A 338; VAL A 340; TYR A 341; VAL A 402; ASN A 406; VAL A 407; TRP A 439; PRO A 446; HIS A 447; GLY A 448; TYR A 449; GLU A 450; ILE A 451; GLU A 452; Ligand; Ions; Water;	No	Delete
8	2000000 ; #4 ns	ALL;	No	Delete
9	10000000 ; #20 ns	ALL;	No	Delete

← Prev
9 Simulation
▶ Execute

Figura 31. Preenchimento de todos os campos de todas as simulações - de 1 a 9 na etapa ‘Simulation’ (4EY6 - GNT).

Ao terminar o processamento do ProtCool 2.0, será gerado o arquivo ‘4EY6.zip’ (uma referência ao código PDB alvo selecionado), nele estará todos os arquivos gerados e tratados pelo ProtCool 2.0 para realizar a dinâmica pelo NAMD, conforme figura 32.

Name	Size	Packed	Type	Modified	CRC32
..			Pasta de arquivos		
log	10.427.043	1.555.717	Pasta de arquivos	21/04/2023 16:56	
confNAMDVina.tar	40.960	1.912	WinRAR archive	21/04/2023 16:57	357A743B
frcomod.tar	10.240	337	WinRAR archive	21/04/2023 16:56	34893923
GeometricCenterVina.tar	10.240	329	WinRAR archive	21/04/2023 16:56	0606B7DA
inpcrdVina.tar	2.662.400	789.643	WinRAR archive	21/04/2023 16:56	CFC78E4D
lib.tar	10.240	1.819	WinRAR archive	21/04/2023 16:56	E191D9A8
ligand1.tar	10.240	1.396	WinRAR archive	21/04/2023 16:49	8C084ECA
ligandconf.tar	10.240	255	WinRAR archive	21/04/2023 16:49	47F3E27B
ligandmol2.tar	10.240	1.448	WinRAR archive	21/04/2023 16:56	3AA9A08F
ligandpdb.tar	10.240	933	WinRAR archive	21/04/2023 16:49	9C5BA5BF
ligandpdbqt.tar	10.240	815	WinRAR archive	21/04/2023 16:49	95A6F3DD
ligandposesVina.tar	71.680	8.589	WinRAR archive	21/04/2023 16:52	AF3857CC
logLeapVina.tar	10.240	966	WinRAR archive	21/04/2023 16:56	AB54130C
NAMDFilesVina.tar	66.631.680	12.897.414	WinRAR archive	21/04/2023 16:57	A57C5C0E
othersTopologyVina.tar	23.920.640	2.844.395	WinRAR archive	21/04/2023 16:56	595C8301
PoseSelectedVina.tar	337.920	85.380	WinRAR archive	21/04/2023 16:52	720D1F50
PosesVina.tar	40.960	4.108	WinRAR archive	21/04/2023 16:52	B9A425B4
PosesVina1.tar	40.960	4.100	WinRAR archive	21/04/2023 16:52	5D9025F1
PosesVinaPDB.tar	30.720	3.668	WinRAR archive	21/04/2023 16:52	2065C0A8
prmtopVina.tar	13.701.120	960.672	WinRAR archive	21/04/2023 16:56	FD7A15D7
proteinlonizeVina.tar	5.027.840	1.113.050	WinRAR archive	21/04/2023 16:56	D41D711F
proteinlonizeVina1.tar	5.027.840	1.113.055	WinRAR archive	21/04/2023 16:56	D4B74AB7
ProteinLigandVina.tar	337.920	78.043	WinRAR archive	21/04/2023 16:56	4EFA0A2
restrictionVina.tar	45.219.840	10.021.573	WinRAR archive	21/04/2023 16:57	398A583A
4EY6.ali	1.175	437	Arquivo ALI	21/04/2023 16:52	B23B3D02
4EY6.fasta	594	394	Arquivo FASTA	21/04/2023 16:52	A6DD7238
4EY6.pdb	784.809	192.911	Arquivo PDB	21/04/2023 16:49	CDBB76C6
4EY6.pdbqt	327.280	88.983	Arquivo PDBQT	21/04/2023 16:49	5CCB4C0F
4EY6.seq	595	385	Arquivo SEQ	21/04/2023 16:52	F3DF5A0A
4EY6_1.pdb	348.500	90.157	Arquivo PDB	21/04/2023 16:49	CB6FEB81
4EY6_2.pdb	330.093	85.133	Arquivo PDB	21/04/2023 16:49	FE22F35F
4EY6_3.pdb	326.301	84.108	Arquivo PDB	21/04/2023 16:49	E1D29ECF
4EY6_4.pdb	332.147	77.437	Arquivo PDB	21/04/2023 16:53	6C3D5B28
4EY6_5.pdb	332.147	77.347	Arquivo PDB	21/04/2023 16:53	C85E771A
4EY6_6.pdb	332.147	77.347	Arquivo PDB	21/04/2023 16:53	C85E771A
4EY6_7.pdb	332.146	77.368	Arquivo PDB	21/04/2023 16:56	7EBC876A
4EY6flex.pdbqt	4.494	1.299	Arquivo PDBQT	21/04/2023 16:49	D9C4B6B9
4EY6H.pdb	671.901	153.842	Arquivo PDB	21/04/2023 16:56	44AB999D
confdinamica.txt	13.552	1.852	Documento de Tex...	21/04/2023 16:49	0AE34D78
confdinamica_1.txt	13.552	1.854	Documento de Tex...	21/04/2023 16:53	F0C06EEA
confdinamica_2.txt	13.415	1.801	Documento de Tex...	21/04/2023 16:56	A0D0668D
pontes4EY6.txt	147	71	Documento de Tex...	21/04/2023 16:49	C4B4A8B7
pontes4EY6_1.txt	135	70	Documento de Tex...	21/04/2023 16:53	34D24F97

Figura 32. Arquivos gerados e preparado pelo ProtCool 2.0 para ser realizado a dinâmica (4EY6 - GNT).

6.1 REALIZANDO A DINAMICA

De posse dos arquivos de preparação gerados, pode ser executada a dinâmica em qualquer máquina que tenha o NAMD devidamente instalado. Neste estudo de caso, o arquivo foi enviado para ser executado em uma lâmina do Laboratório do Cluster no Campus da UNIFEI em Itabira, com sistema operacional Linux, que contendo 24 CPUs, GPU Tesla K40M e 64GB de RAM.

Para executar a dinâmica, foi descompactado o arquivo ‘4EY6.zip’, acessado o diretório ‘/NAMDFilesVina’ e executado o script Python chamado ‘dinamica.py’. Todo o workflow levou em torno de 96 horas (4 dias) de execução ininterruptas, ocupando em torno de 4,5Gb de espaço em disco.

```

1 [||||| 100.0%] 7 [||||| 100.0%] 13 [||||| 100.0%] 19 [||||| 100.0%]
2 [||||| 100.0%] 8 [||||| 100.0%] 14 [||||| 100.0%] 20 [||||| 100.0%]
3 [||||| 100.0%] 9 [||||| 100.0%] 15 [||||| 100.0%] 21 [||||| 100.0%]
4 [||||| 100.0%] 10 [||||| 100.0%] 16 [||||| 100.0%] 22 [||||| 100.0%]
5 [||||| 100.0%] 11 [||||| 100.0%] 17 [||||| 100.0%] 23 [||||| 100.0%]
6 [||||| 100.0%] 12 [||||| 100.0%] 18 [||||| 100.0%] 24 [||||| 100.0%]
Mem | 1.05G/62.8G | Tasks: 34, 77 thr; 24 running
Swap | 1.50M/65.0G | Load average: 23.98 23.99 24.00
                                           Uptime: 45 days, 21:47:08

PID USER   PRI  NI  VIRT   RES   SHR  S CPU% MEM%   TIME+  Command
19755 default 20   0 86.3G 453M 115M R 2396  0.7 2155:52S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19757 default 20   0 86.3G 453M 115M R 99.0  0.7 2155:26S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19763 default 20   0 86.3G 453M 115M R 100.  0.7 2155:24S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19765 default 20   0 86.3G 453M 115M R 99.7  0.7 2155:15S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19762 default 20   0 86.3G 453M 115M R 100.  0.7 2155:02S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19776 default 20   0 86.3G 453M 115M R 99.7  0.7 2154:58S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19773 default 20   0 86.3G 453M 115M R 99.0  0.7 2154:58S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19772 default 20   0 86.3G 453M 115M R 99.7  0.7 2154:53S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19766 default 20   0 86.3G 453M 115M R 99.7  0.7 2154:38S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19767 default 20   0 86.3G 453M 115M R 100.  0.7 2154:38S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19768 default 20   0 86.3G 453M 115M R 100.  0.7 2154:37S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19770 default 20   0 86.3G 453M 115M R 99.7  0.7 2154:36S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19761 default 20   0 86.3G 453M 115M R 99.7  0.7 2154:33S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19778 default 20   0 86.3G 453M 115M R 99.7  0.7 2154:23S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19758 default 20   0 86.3G 453M 115M R 100.  0.7 2154:23S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19775 default 20   0 86.3G 453M 115M R 100.  0.7 2154:15S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19769 default 20   0 86.3G 453M 115M R 99.7  0.7 2153:48S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19759 default 20   0 86.3G 453M 115M R 100.  0.7 2153:54S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19774 default 20   0 86.3G 453M 115M R 99.0  0.7 2153:33S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19769 default 20   0 86.3G 453M 115M R 99.7  0.7 2153:33S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19756 default 20   0 86.3G 453M 115M R 99.7  0.7 2153:15S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19764 default 20   0 86.3G 453M 115M R 100.  0.7 2153:09S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19777 default 20   0 86.3G 453M 115M R 99.7  0.7 2152:54S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
19771 default 20   0 86.3G 453M 115M R 98.4  0.7 2154:10S /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
1261 root    20   0 108M 3176 2680 S 0.0  0.0 15:00.19 /usr/sbin/lrbalance --foreground
19950 default 20   0 11920 5552 3560 S 0.0  0.0 3:22.13 http
1 root    20   0 1220M 7640 8072 S 0.0  0.0 3:10.30 /lib/systemd/systemd --system --deserialize 39
20729 systemd-t 20   0 138M 9484 2884 S 0.0  0.0 1:09.17 /lib/systemd/systemd-timesyncd
20705 systemd-r 20   0 70620 3656 8056 S 0.0  0.0 0:55.45 /lib/systemd/systemd-resolved
3501 root    20   0 2740M 35040 13304 S 0.0  0.1 0:45.39 /usr/lib/snapd/snapd
1282 root    20   0 279M 1328 1324 S 0.0  0.0 0:44.30 /usr/lib/accounts-service/accounts-daemon
1318 root    20   0 279M 1328 1324 S 0.0  0.0 0:44.11 /usr/lib/accounts-service/accounts-daemon
19791 default 20   0 86.3G 453M 115M S 0.0  0.7 0:35.55 /home/default/enthalpy/NAMD_2.12.Linux-x86_64-multicore-CLD/namd2 +idlepoll +p24 ./!Rodada 4EYegnt_9.conf
20747 systemd-t 20   0 138M 9484 2884 S 0.0  0.0 0:33.65 /lib/systemd/systemd-timesyncd
1395 root    20   0 302M 2864 1588 S 0.0  0.0 0:21.13 /usr/bin/xcfs /var/lib/xcfs/
15596 default 20   0 109M 4692 4024 S 0.0  0.0 0:21.04 sdris -defaultgpts/0
1438 root    20   0 302M 2864 1588 S 0.0  0.0 0:05.87 /usr/bin/xcfs /var/lib/xcfs/
1437 root    20   0 302M 2864 1588 S 0.0  0.0 0:05.81 /usr/bin/xcfs /var/lib/xcfs/
2435 root    20   0 302M 2864 1588 S 0.0  0.0 0:05.81 /usr/bin/xcfs /var/lib/xcfs/
1320 root    20   0 70670 8608 4232 S 0.0  0.0 0:05.51 /lib/systemd/systemd-logind
1 help  ?  Setup  Search  Edit  View  Help  Home  ?  Exit

```

Figura 33. Demonstração do processamento do servidor realizando a dinâmica molecular em 24 processadores - saída do comando 'htop'.

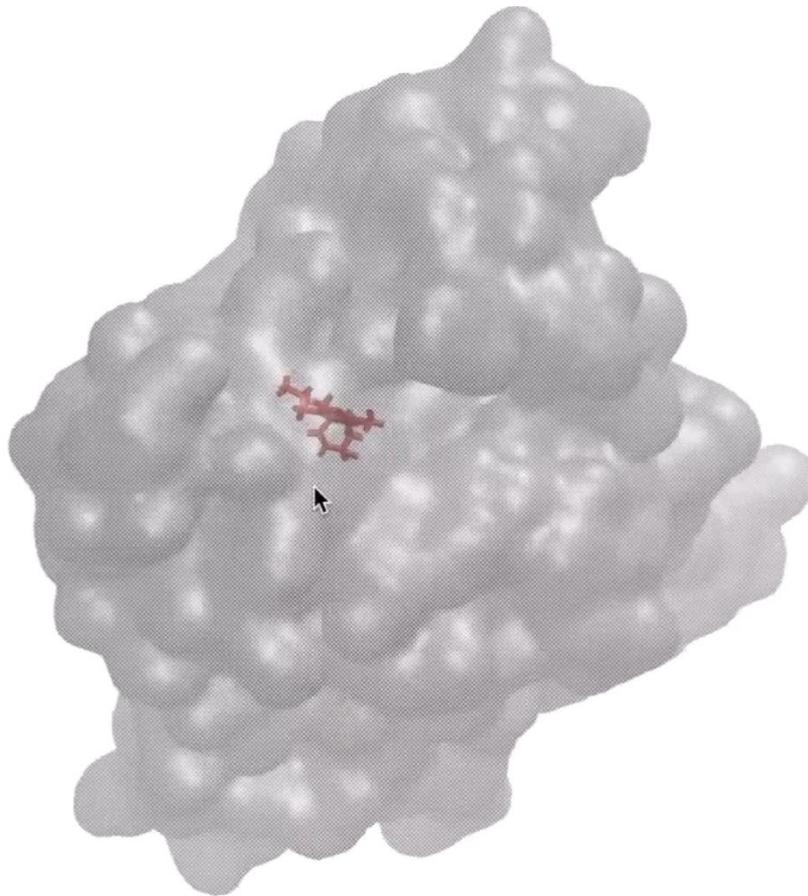


Figura 34. Representação gráfica gerada pelo software VMD da dinâmica molecular já realizada envolvendo a acetilcolinesterase e a galantamina.

7 CONCLUSÃO

Ante tudo que foi exposto, fica evidente a facilidade do ProtCool 2.0 em implementar a preparação de complexos e variados *workflows* de simulações de dinâmica molecular. E faz tudo isso através de uma interface amigável e intuitiva que consegue alcançar pesquisadores de diferentes níveis de experiência. Foi projetado para ser uma ferramenta fácil e ao mesmo tempo útil com intuito de minimizar a barreira complexa de usabilidade quando envolve pesquisas de MD. Da mesma forma, ele torna-se uma ferramenta que pode ser utilizada para melhorar os aspectos didáticos e educacionais, em consequência aumentar o aprendizado em disciplinas que envolvam o conteúdo de química computacional, além de proporcionar avanços no processo de desenvolvimento de novos fármacos.

Além do mais, uma vez gerado um *workflow*, o mesmo pode ser reproduzido por diferentes pesquisadores em variados ambientes de computação. Fica fácil também a adaptação e reutilização de *workflows* antigos em novos contextos. Com o ProtCool2.0, ganha-se então reprodutibilidade e reusabilidade em experimentos de simulação de dinâmica molecular

No entanto, ProtCool 2.0 tem suas limitações, e para cada uma delas, abrem-se novas perspectivas para implementações futuras. Em sua versão atual, oferece um escopo limitado de opções pré-configuráveis, ainda que estejam disponibilizados as etapas e os parâmetros mais comumente utilizados nos protocolos de dinâmica molecular descritos na literatura. Mais opções poderão ser integradas à ferramenta conforme a demanda dos usuários, em atualizações futuras. Seja como for, usuários com demandas específicas e com alguma habilidade de programação não teriam dificuldades em customizá-lo de acordo com suas necessidades. Os passos envolvendo os *workflows* produzidos pelo ProtCool 2.0 são modularizados, de modo que a adequação de alguma etapa tende a ter poucos efeitos colaterais nas demais.

REFERÊNCIAS

ADCOCK, Stewart A.; MCCAMMON, J. Andrew. Molecular dynamics: survey of methods for simulating the activity of proteins. **Chemical reviews**, v. 106, n. 5, p. 1589-1615, 2006.

ALDER, Berni J.; WAINWRIGHT, Thomas Everett. Studies in molecular dynamics. I. General method. **The Journal of Chemical Physics**, v. 31, n. 2, p. 459-466, 1959.

ALTEXSOFT. What is API: Definition, Types, Specifications, Documentation. Disponível em: <<https://www.altexsoft.com/blog/engineering/what-is-api-definition-types-specifications-documentation/>>.

ANDERSON, Joshua A.; GLASER, Jens; GLOTZER, Sharon C. HOOMD-blue: A Python package for high-performance molecular dynamics and hard particle Monte Carlo simulations. **Computational Materials Science**, v. 173, p. 109363, 2020.

ANDRIO, Pau et al. BioExcel Building Blocks, a software library for interoperable biomolecular simulation workflows. **Scientific data**, v. 6, n. 1, p. 169, 2019.

ARANDA-GARCIA, David et al. Simulating Time-Resolved Dynamics of Biomolecular Systems. 2022.

BERMAN, Helen M. et al. The protein data bank. **Nucleic acids research**, v. 28, n. 1, p. 235-242, 2000.

BERSON, Alex. Client/server architecture. McGraw-Hill, Inc., 1996.

BECK, Kent et al. Manifesto for agile software development. 2001.

BITZEK, Erik et al. Structural relaxation made simple. **Physical review letters**, v. 97, n. 17, p. 170201, 2006.

BRINI, Emiliano; SIMMERLING, Carlos; DILL, Ken. Protein storytelling through physics. *Science*, v. 370, n. 6520, p. eaaz3041, 2020.

BROOKS, Bernard R. et al. CHARMM: a program for macromolecular energy, minimization, and dynamics calculations. **Journal of computational chemistry**, v. 4, n. 2, p. 187-217, 1983.

CARVALHO, Ivone et al. Introdução a modelagem molecular de fármacos no curso experimental de química farmacêutica. **Química Nova**, v. 26, p. 428-438, 2003.

CASE, David A. et al. Amber 10. University of California, 2008.

CASE, David A. et al. The Amber biomolecular simulation programs. **Journal of computational chemistry**, v. 26, n. 16, p. 1668-1688, 2005.

Client-Server Model Definition. Disponível em: <https://techterms.com/definition/client-server_model>.

ERVIK, Åsmund; SERRATOS, Guadalupe Jiménez; MÜLLER, Erich A. raaSAFT: A framework enabling coarse-grained molecular dynamics simulations based on the SAFT- γ Mie force field. **Computer Physics Communications**, v. 212, p. 161-179, 2017.

GALEAZZI, Roberta. Molecular dynamics as a tool in rational drug design: current status and some major applications. **Current Computer-Aided Drug Design**, v. 5, n. 4, p. 225-240, 2009.

Geckodriver. Disponível em: <<https://github.com/mozilla/geckodriver>>. Acesso em: 11 abr. 2023.

Gnu general public license. URL <http://www.gnu.org/licenses/gpl.html>

GORET, G.; AOUN, B.; PELLEGRINI, Eric. MDANSE: An interactive analysis environment for molecular dynamics simulations. **Journal of chemical information and modeling**, v. 57, n. 1, p. 1-5, 2017.

GOWERS, Richard J. et al. MDAnalysis: a Python package for the rapid analysis of molecular dynamics simulations. In: **Proceedings of the 15th python in science conference**. Austin, TX: SciPy, 2016. p. 105.

GUEDES, Fabiana Costa et al. Protcool: um gerador de protocolos para ancoragens e simulações de dinâmica molecular em complexos proteína-ligante. 2021.

GUVENCH, Olgun; MACKERELL, Alexander D. Comparison of protein force fields for molecular dynamics simulations. *Molecular modeling of proteins*, p. 63-88, 2008.

HESS, Berk et al. GROMACS 4: algorithms for highly efficient, load-balanced, and scalable molecular simulation. **Journal of chemical theory and computation**, v. 4, n. 3, p. 435-447, 2008.

HINSEN, Konrad. The molecular modeling toolkit: a new approach to molecular simulations. **Journal of Computational Chemistry**, v. 21, n. 2, p. 79-85, 2000.

HOSPITAL, Adam et al. MDWeb and MDMoby: an integrated web-based platform for molecular dynamics simulations. **Bioinformatics**, v. 28, n. 9, p. 1278-1279, 2012.

HOWARD, Michael P.; PANAGIOTOPOULOS, Athanassios Z.; NIKOUBASHMAN, Arash. Efficient mesoscale hydrodynamics: Multiparticle collision dynamics with massively parallel GPU acceleration. **Computer Physics Communications**, v. 230, p. 10-20, 2018.

JSON. Disponível em: <<http://json.org/json-pt.html>>. Acesso em: 11 abr. 2023.

KARPLUS, Martin; MCCAMMON, J. Andrew. Molecular dynamics simulations of biomolecules. **Nature structural biology**, v. 9, n. 9, p. 646-652, 2002.

KOES, David Ryan; BAUMGARTNER, Matthew P.; CAMACHO, Carlos J. Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise. **Journal of chemical information and modeling**, v. 53, n. 8, p. 1893-1904, 2013.

LENGAUER, Thomas; RAREY, Matthias. Computational methods for biomolecular docking. *Current opinion in structural biology*, v. 6, n. 3, p. 402-406, 1996.

MCCAMMON, J. Andrew; GELIN, Bruce R.; KARPLUS, Martin. Dynamics of folded proteins. *Nature*, v. 267, n. 5612, p. 585-590, 1977.

MOONEY, Christopher Z. Monte carlo simulation. Sage, 1997.

NATIONAL ACADEMIES OF SCIENCES, ENGINEERING, AND MEDICINE et al. Reproducibility and replicability in science. **National Academies Press**, 2019.

NELSON, Mark T. et al. NAMD: a parallel, object-oriented molecular dynamics program. **The International Journal of Supercomputer Applications and High Performance Computing**, v. 10, n. 4, p. 251-268, 1996.

ONUFRIEV, A. et al. H++. <http://biophysics.cs.vt.edu>. 2005.

PARKS, Jerry M.; SMITH, Jeremy C. How to discover antiviral drugs quickly. **New England Journal of Medicine**, v. 382, n. 23, p. 2261-2264, 2020.

PENG, Zhe; JIMENEZ, Jose L. KinSim: a research-grade, user-friendly, visual kinetics simulator for chemical-kinetics and environmental-chemistry teaching. 2019.

PERILLA, Juan R. et al. Molecular dynamics simulations of large macromolecular complexes. *Current opinion in structural biology*, v. 31, p. 64-74, 2015.

PETTITT, B. Montgomery; KARPLUS, Martin. The potential of mean force surface for the alanine dipeptide in aqueous solution: a theoretical approach. **Chemical physics letters**, v. 121, n. 3, p. 194-201, 1985.

PHILLIPS, James C. et al. Scalable molecular dynamics with NAMD. **Journal of computational chemistry**, v. 26, n. 16, p. 1781-1802, 2005.

PLIMPTON, Steve. Fast parallel algorithms for short-range molecular dynamics. **Journal of computational physics**, v. 117, n. 1, p. 1-19, 1995.

POLIK, William F.; SCHMIDT, J. R. WebMO: Web-based computational chemistry calculations in education and research. Wiley Interdisciplinary Reviews: **Computational Molecular Science**, v. 12, n. 1, p. e1554, 2022.

PRESSMAN, Roger S. **Software engineering: a practitioner's approach**. Palgrave macmillan, 2005.

PURAWAT, Shweta et al. A Kepler workflow tool for reproducible AMBER GPU molecular dynamics. **Biophysical journal**, v. 112, n. 12, p. 2469-2474, 2017.

PYTHON SOFTWARE FOUNDATION. Python Language Site: Homepage, 2023. Página principal. Disponível em: <<https://www.python.org>>. Acesso em: 04 de mar. de 2023.

RAPAPORT, Dennis C.; RAPAPORT, Dennis C. Rapaport. The art of molecular dynamics simulation. Cambridge university press, 2004.

RAWAT, Ravi et al. HeroMDAnalysis: an automagical tool for GROMACS-based molecular dynamics simulation analysis. **Future Medicinal Chemistry**, v. 13, n. 05, p. 447-456, 2021.

RIBEIRO, João V. et al. QwikMD—integrative molecular dynamics toolkit for novices and experts. **Scientific reports**, v. 6, n. 1, p. 1-14, 2016.

ROCHA, Rafael Eduardo Oliveira. Estudos de modelagem molecular dos mecanismos de afinidade relativa para quatro galantamínicos com potencial anti-Alzheimer. 2017.

ROMO, Tod D.; LEIOATTS, Nicholas; GROSSFIELD, Alan. Lightweight object oriented structure analysis: tools for building tools to analyze molecular dynamics simulations. **Journal of computational chemistry**, v. 35, n. 32, p. 2305-2318, 2014.

SALO-AHEN, Outi MH et al. Molecular dynamics simulations in drug discovery and pharmaceutical development. *Processes*, v. 9, n. 1, p. 71, 2020.

SERÇINOĞLU, Onur; OZBEK, Pemra. gRINN: a tool for calculation of residue interaction energies and protein energy network analysis of molecular dynamics simulations. **Nucleic acids research**, v. 46, n. W1, p. W554-W562, 2018.

SINGH, Sakshi; BAKER, Qanita Bani; SINGH, Dev Bukhsh. Molecular docking and molecular dynamics simulation. In: **Bioinformatics**. Academic Press, 2022. p. 291-304.

SOMMERVILLE, Ian. **Software Engineering, 9/E**. Pearson Education India, 2011.

SQUIRES, Gordon Leslie. Introduction to the theory of thermal neutron scattering. **Courier Corporation**, 1996.

Standard ECMA-262. Homepage 2023. Página principal. Disponível em: <ecma-international.org>. Acesso em 22 de julho de 2022.

STERLING, Teague; IRWIN, John J. ZINC 15—ligand discovery for everyone. **Journal of chemical information and modeling**, v. 55, n. 11, p. 2324-2337, 2015.

TATLI, Zeynep; AYAS, Alipaşa. Virtual laboratory applications in chemistry education. **Procedia-Social and behavioral sciences**, v. 9, p. 938-942, 2010.

TAVAKOL, Mahdi et al. Mechanical properties of graphene oxide: The impact of functional groups. **Applied Surface Science**, v. 525, p. 146554, 2020.

TIGAA, Rodney A.; SONAWANE, Swapnil L. An international perspective: teaching chemistry and engaging students during the COVID-19 pandemic. **Journal of Chemical Education**, v. 97, n. 9, p. 3318-3321, 2020.

TROTT, Oleg; OLSON, Arthur J. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. **Journal of computational**

chemistry, v. 31, n. 2, p. 455-461, 2010.

VAN DER SPOEL, David et al. GROMACS: fast, flexible, and free. **Journal of computational chemistry**, v. 26, n. 16, p. 1701-1718, 2005.

Vue.js. Disponível em: <<https://br.vuejs.org/index.html>>. Acesso em: 11 abr. 2023.

W3C. CSS Site: Homepage, 2023. Página principal. Disponível em: <<https://www.w3.org/Style/CSS/>>. Acesso em: 04 de mar. de 2023.

W3C. HTML Site: Homepage, 2023. Página principal. Disponível em: <<https://www.w3.org/html/>>. Acesso em: 04 de mar. de 2023.

WebDriver. Disponível em: <<https://www.selenium.dev/documentation/webdriver>>. Acesso em: 11 abr. 2023.

Zenity GUI Interactivity (2023).Página do projeto. Disponível em <<https://wiki.gnome.org/Projects/Zenity>>. Acesso em 04 de mar de 2023.