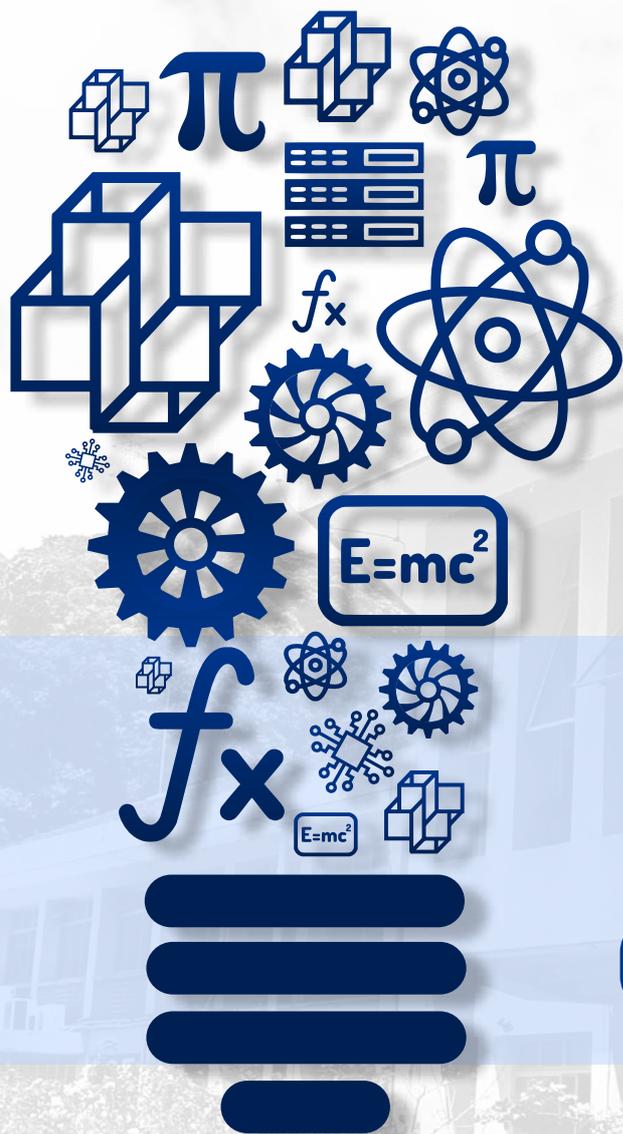


Jornada de Iniciação Científica e Tecnológica

• 2022



PIBIC • PIBITI
LNCC/MCTI

28 de julho

Jornada de Iniciação Científica e Tecnológica do LNCC

Petrópolis, 28 de julho de 2022.

Laboratório Nacional de Computação Científica – LNCC

Diretor

Fabio Borges de Oliveira

Coordenação de Gestão e Administração - COGEA

Sergio Ferreira de Figueiredo

Coordenação de Métodos Matemáticos e Computacionais - COMAC

Pablo Javier Blanco – Coordenador substituto

Coordenação de Modelagem Computacional - COMOD

Laurent Emmanuel Dardenne

Coordenação de Pós-Graduação e Aperfeiçoamento - COPGA

Sandra Mara Cardoso Malta

Coordenação de Tecnologia da Informação e Comunicação - COTIC

Wagner Vieira Léo

Programa Institucional de Bolsas de Iniciação Científica &

Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação

José Karam Filho

Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq

Presidente

Evaldo Ferreira Vilela

Coordenadora Geral do PIBIC/PIBITI

Lucimar Batista de Almeida

Jornada de Iniciação Científica e Tecnológica do LNCC

Comissão Interna do PIBIC/PIBITI-LNCC

José Karam Filho
Antonio Tadeu Azevedo Gomes
Eduardo Lucio Mendes Garcia
Jack Baczynski
Fábio Lima Custódio

Avaliadores Externos

Márcio Antônio de Andrade Bortoloti – UESB
Cristiane Oliveira de Faria UERJ

Apresentação

O LNCC realiza este ano a XIX Edição da Jornada de Iniciação Científica e Tecnológica, que é um fórum de divulgação das pesquisas desenvolvidas no contexto dos Programas Institucionais de Bolsas de Iniciação Científica (PIBIC) e de Bolsas de Iniciação Tecnológica (PIBITI) fomentados pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). No período de setembro de 2021 a julho de 2022, o PIBIC e PIBITI congregaram alunos de várias instituições de ensino e de diversas áreas do conhecimento. Este volume apresenta os resumos dos trabalhos desenvolvidos pelos bolsistas no período. Durante a Jornada, os trabalhos são apresentados pelos bolsistas oralmente e avaliados por um comitê científico externo.

Nesta XIX Edição da Jornada, o Comitê Externo de Avaliação do PIBIC/PIBITI tem a seguinte composição:

Prof. Márcio Antônio de Andrade Bortoloti- UESB

Prof^a. Cristiane Oliveira de Faria (UERJ)

Destacamos o papel relevante do PIBIC/PIBITI do LNCC no desenvolvimento das pesquisas no LNCC e, principalmente, na formação complementar dos bolsistas, promovendo o aprimoramento do conhecimento, espírito criativo, reflexão crítica e ética. Estas características têm contribuído para suas inserções no mercado de trabalho e em programas de pós-graduação, como o PPG em Modelagem Computacional do LNCC. Este é o resultado do esforço e dedicação de todos os participantes.

Agradecimentos

Agradecemos ao CNPq pelas bolsas concedidas, à Direção do LNCC pelo apoio e à Comissão Interna do PIBIC e PIBITI no LNCC.

Agradecemos a disponibilidade e contribuição dos membros do Comitê Externo de Avaliação.

O sucesso desta Jornada, e do Programa como um todo, é o resultado da dedicação e do esforço de toda a comunidade do LNCC. Expressamos em particular nosso reconhecimento ao apoio concedido pela secretaria do PPG-LNCC e, em particular, à Sra. Roberta Machado.

José Karam Filho
Coordenador do PIBIC/PIBITI - LNCC

Índice

Bolsistas PIBIC ativos

Comportamento Estrutural de Dutos Flexíveis no Transporte de Petróleo Offshore

Bolsista: Antônio Arbex Barroso

Orientadores: Elson Magalhães Toledo e Marcos Vinicius Rodrigues

Técnicas de Ciências de Dados Aplicadas a Pesquisas Climáticas

Bolsista: Gabriel Tavares Da Silva Vale

Orientadores: Fábio Andrade Machado Porto e Anderson Chaves Da Silva

Exploração de Módulos Paralelo-Híbrido de Bioinformática para Ambientes GPU de Supercomputação

Bolsista: Guilherme Freire da Silva Dornelas

Orientadores: Carla Osthoff Ferreira de Barros, Kary Ann del Carmen Ocaña, Micaella Coelho Valente de Paula

Arcabouço Computacional para Simulações Numéricas apoiadas por Aprendizado de Máquina

Bolsista: João Victor Caccavo de Araújo Wanderley

Orientadores: Antônio Tadeu Azevedo Gomes e Frédéric Valentin

Coletor de amostras de dossel embarcado em veículo a éreo não tripulado

Bolsista: João Vitor R. Rabello

Orientadores: Jauvane Cavalcante de Oliveira e Luis Claudio Batista da Silva

Previsão meteorológica utilizando métodos de inteligência artificial

Bolsista: Júlia Neumann Bastos

Orientador: Fabio Porto e Rafael Silva Pereira

Projeto e Implementação de Workflows Científicos Reprodutíveis de Alto Desempenho

Bolsista: Lucas da Cruz Silva

Orientadores: Luiz Manoel Rocha Gadelha Júnior, Carla Osthoff e Kary Ocaña

Modelos da anatomia arterial do córtex cerebral gerados por algoritmos de vascularização automáticos

Bolsista: Luiza Carneiro Pizzi

Orientador: Pablo Javier Blanco

Modelagem e Integração de Bancos de Dados Relacionais na Arquitetura do Bioinfo-Portal

Bolsista: Marco Antonio Silva Cabral

Orientador: Kary Ann del Carmen Ocaña Gautherot, Antonio Tadeu Azevedo Gomes e Marcelo Monteiro Galheigo

Otimização do firmware utilizado pelo robô da Very Small Size League para aperfeiçoamento das suas habilidades, do controle e do sistema de comunicação.

Bolsista: Nicolle Vitória Costa de Vasconcelos

Orientador: Jauvane de Oliveira Cavalcante

Simulação Computacional de Parâmetros de Escoamento Sanguíneo

Bolsista: Rodrigo Schmidt

Orientador: Pablo Javier Blanco

Modelagem de vigas de Euler Bernoulli e Timoshenko

Bolsista: Tarsiane Ribeiro da Costa

Orientador: Jaime Ediberto Munoz Rivera

Plataforma móvel Robótica-UV para desinfecção de ambientes: uma solução no combate à COVID-19 e outros agentes patogênicos

Bolsista: Victor Freixinho Nunes

Orientadores: Jauvane Cavalcante de Oliveira e Paulo Fernando Ferreira Ros

Bolsista PIBITI ativo

Gerenciamento de grandes volumes de dados nos sistemas Web do SINAPAD

Bolsista: Lucas Fernandes Calheiros

Orientador: Antônio Tadeu Azevedo Gomes

Comportamento Estrutural de Dutos Flexíveis no Transporte de Petróleo Offshore

1 Objetivos

Os objetivos do presente trabalho visam a busca pelo entendimento do comportamento de dutos flexíveis utilizados na indústria petrolífera sujeitos a carregamentos ambientais que podem levar a falhas catastróficas com grandes consequências financeiras e ambientais.

Pretende-se desta forma, dentro das atividades deste projeto de IC, ampliar o conhecimento sobre a cadeia de exploração e produção de petróleo e gás; aprofundar o entendimento sobre os desafios de engenharia nas atividades desempenhadas pelos sistemas de risers; determinar os modos de falha a que um riser flexível está sujeito durante sua vida em operação e avançar no conhecimento sobre análise de fadiga das armaduras de tração dos risers flexíveis; além de contribuir para a formação na área de Engenharia.

De forma mais detalhada, para se obter um maior conhecimento do comportamento de um duto flexível sujeito ao carregamento de fadiga, foram estudados os seguintes objetivos específicos:

- Estudos sobre os carregamentos a que um riser flexível está sujeito quando conectado a uma plataforma flutuante;
- Estudo sobre a composição das camadas que compõem um duto flexível e a função de cada uma destas camadas;
- Modelagem de um segmento de duto flexível, incluindo-se detalhes da seção transversal que sejam suficientes para se obter a simulação do comportamento do duto quando submetido a carregamentos de fadiga;
- Estudo dos esforços nas armaduras de tração a partir da aplicação de carregamentos axiais, flexionais e torcionais na estrutura .

2 Introdução

Os risers são dutos suspensos utilizados para o transporte de petróleo desde estruturas subaquáticas até estruturas na superfície do oceano utilizadas no armazenamento deste material. Estão sujeitos, principalmente, à carregamentos dinâmicos, tais como tração, torção, flexão, etc.

As seções críticas desses risers estão localizadas nas zonas de acoplamento nas estruturas flutuantes e no touchdown point, onde ocorrem grandes variações de tração (e grandes curvaturas), além da crista e da corcova de sua trajetória, onde também se evidenciam grandes curvaturas (e baixas trações) [1]. São justamente essas variações de tração que são responsáveis pelo seu possível rompimento devido a fadiga do componente.

Esses dutos flexíveis possuem uma estrutura complexa, de várias camadas, que visam garantir confiabilidade ao sistema de transporte e proteção contra carregamentos externos oriundos de intempéries. As camadas consistem em:

- Carcaça interna, responsável pelo revestimento interno protegendo o duto de entrar em colapso;
- Revestimento de pressão polimérico, que mantém o fluido dentro de seu calibre;
- Armaduras de pressão, que provêm resistência à compressão radial do riser;
- Armaduras de tração, oferecem resistência à tração. São duas camadas constituídas por arames de aço dispostos helicoidalmente em sentidos opostos, que oferecem rigidez ao duto;
- Revestimento polimérico externo, que impede a interação da água do mar com as armaduras internas de aço.

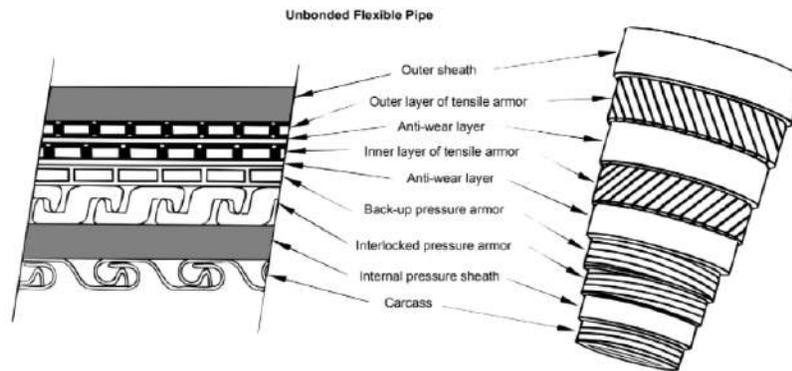


Figura 1: Esquema das secções transversais de um riser. [1]

As armaduras de tração são intercaladas por camadas anti-desgaste que permitem que os arames da estrutura fiquem livres para o deslizamento relativo entre si e assim, fazem com que as tensões axiais provenientes de flexões possam ser liberados. Neste trabalho será feita uma análise mais aprofundada nas armaduras de tração.

3 Metodologia

O presente trabalho consistirá no estudo do comportamento estrutural de dutos flexíveis com armaduras helicoidais e suas respostas quando expostas a carregamentos cíclicos de tração. Para tanto, faremos, uma análise de duas metodologias diferentes para equacionar os dutos helicoidais flexíveis baseadas, respectivamente, nos trabalhos de Svein Sævik [2] e Nils Sødahl [3], as quais nos referiremos como Metodologia 1 e Metodologia 2 a partir de agora.

Em todas as duas abordagens, faz-se uma distinção dos carregamentos presentes em duas categorias [3]:

- Carregamentos axissimétricos constituídos por tração, torção, pressões internas e externas;
- Carregamentos por flexão.

A separação dos modelos de resposta visa gerar um esquema viável de solução computacional eficiente e robusto [3].

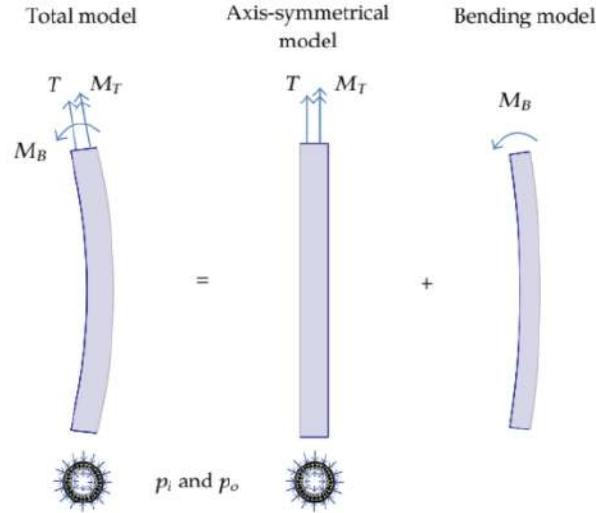


Figura 2: Modelos de resposta para análises transversais. [3]

3.1 Modelos Axissimétricos

3.1.1 Metodologia 1 (Sævik)

Em dutos flexíveis as armaduras de tração são responsáveis por receber a maior parte dos carregamentos axiais. Quando expostos à tração e à torção, os riser do tipo unbonded respondem de maneira linear.

Realiza-se, então, uma análise de um modelo pré-existente de uma armadura com um único arame deslizante. Este modelo será usado para descrever o comportamento do conjunto de arames da armadura durante seu uso. Como os carregamentos axissimétricos (torção, tração, pressão, etc.) não mudam o formato da armadura, o comportamento axissimétrico do riser pode ser descrito por equações relativamente simples [4].

Para descrever um ponto em uma superfície cilíndrica com um raio constante ρ , serão usados eixos de coordenadas Z^I e vetores E_I . Além disso, os pontos da superfície lateral serão identificados por um raio R e por coordenadas w e v [4]:

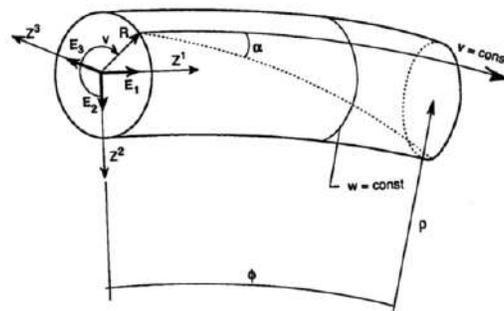


Figura 3: Definição do sistema de coordenadas da tubulação [4].

As coordenadas cartesianas de um ponto arbitrário na superfície circular transversal são expressas por [4]:

$$Z^1 = (\rho - R \cos v) \sin \varphi \quad (1)$$

$$Z^2 = \rho - (\rho - R \cos v) \cos \varphi \quad (2)$$

$$Z^3 = R \sin v \quad (3)$$

Quando o duto é exposto a um grande esforço de tração, pode-se assumir que os principais eixos de coordenadas da seção transversal estão fixados sobre a normal da superfície [4]. Assim, define-se, sobre um ponto arbitrário, um sistema de coordenadas I_I para descrever os principais eixos de torção-flexão ao longo curva, onde I_1 consiste no vetor unitário tangente à curva, I_2 no vetor unitário normal e I_3 no vetor unitário binormal [4]:

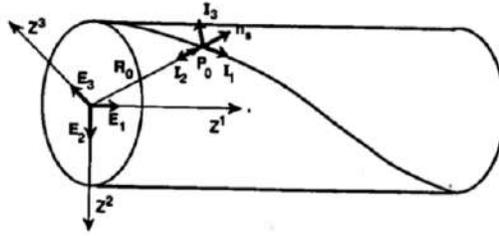


Figura 4: Sistema de coordenadas I_I [4].

3.1.2 Metodologia 2 (Sødahl)

A metodologia 2 faz uso do software Helica [5], um programa dedicado desenvolvido pela DNV para superar os desafios da análise de fadiga nos risers. É feita a separação dos carregamentos presentes em axissimétricos e gerados por flexões [3]. Trabalharemos aqui com os carregamentos axissimétricos.

Descrever cinematicamente as camadas cilíndricas e helicoidais é essencial para entender como a seção transversal responde aos carregamentos. O cilindro é dado parametricamente por [3]:

$$x_c(x, \theta) = x i_x + R \cos \theta i_y + R \sin \theta i_z \quad (4)$$

i_j corresponde ao sistema de coordenadas global. Uma hélice seria definida como uma curva na superfície cilíndrica seguindo uma trajetória fixa definida pelo ângulo de inclinação α [3]. Isso produz também uma relação entre a eixo helicoidal s e o ângulo polar θ . A geometria da hélice é, portanto, um caso particular de um parâmetro do cilindro [3]:

$$x_h(s) = x_c(s \cos \alpha, \frac{s \sin \alpha}{R}), \quad (5)$$

onde

$$\theta = \frac{s \sin \alpha}{R} = \theta_s, \quad x = s \cos \alpha = x_s \quad (6)$$

O cilindro deformado se dá por meio de três componentes independentes de mudança constante de raio, de ângulo e de extensão linear. As geometrias não deformadas e deformadas estão relacionadas por [3]:

$$x' = x + \frac{x}{L}u_x, R' = R + u_R, \theta' = \theta + \frac{x}{L}u_\theta \quad (7)$$

4 Resultados Preliminares e Discussão

Nesse projeto, nos dedicamos, inicialmente, em criar um modelo baseado na metodologia 2. Esta modelagem foi feita com base na equações (4), (5), (6). O modelo consiste primeiramente em uma hélice gerada com base em um parâmetro α de $\pi/4$, $R = 10$ e $L = 500$, em função de x e θ . Posteriormente, é gerada um segunda hélice, deformada, com os parâmetros alterados x' , R' e θ' com base nas equações (7). Para tanto, foram arbitrados valores de $u_x = 10$, $u_R = -0.5$ e $u_\theta = -1$, pelo fato de não ter sido estabelecida a relação entre força e deslocamento. As duas hélices podem ser visualizadas abaixo:

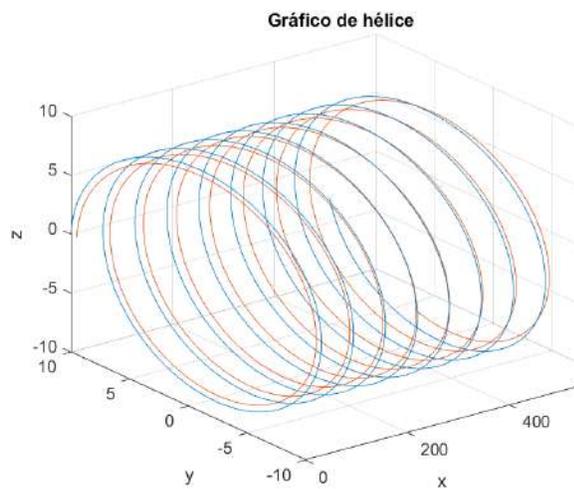


Figura 5: Curva original e deformada.

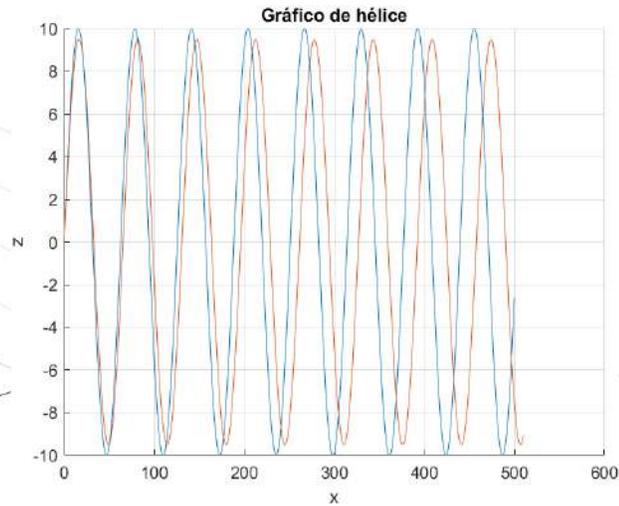


Figura 6: Curva original e deformada.

É possível notar a influência de u_R que age sobre o raio da curva, diminuindo-o. Além disso, os outros dois parâmetros u_x e u_θ contribuem para o “alongamento” da hélice, como é possível verificar acima. Isso seria uma configuração deformada resultado da combinação de carregamentos de tração, torção e pressão.

5 Conclusões

Com este trabalho foi possível iniciar os estudos sobre o complexo tema que se trata os risers flexíveis. Com ele, pôde-se estudar duas abordagens diferentes sobre a geometria das hélices das armaduras de tração, e modelar uma geometria tubular helicoidal seguindo uma das metodologias estudadas. Com isso, foi adquirido conhecimento sobre modelagem de curvas e geometria diferencial além de conhecimentos mais aprofundados sobre a área de produção de petróleo.

Como próximo passo, pretende-se estabelecer as relações entre os parâmetros u_x , u_R e u_θ e os carregamentos que geram essas respectivas deformações, caminhando, assim, para uma análise estrutural de fato, levando em consideração a rigidez do duto como um todo. Posteriormente

pretende-se estudar a relação entre tensão e deformação para obter-se as tensões no arame e ampliar os estudos para carregamentos causados por flexões, abrangendo, assim, geodésicas e curvas loxodrômicas.

Referências

- [1] AMERICAN PETROLEUM INSTITUTE. Recommended Practice for Flexible Pipe: API RECOMMENDED PRACTICE 17B. American Petroleum Institute, [s. l.], ed. 3, Março 2002.
- [2] SÆVIK, Svein. On Stresses and Fatigue in Flexible Pipes. The University of Trondheim: [s. n.], 1992.
- [3] SKEIE, Geir; SØDAHL, Nils; STEINKJER, Oddrun. Efficient Fatigue Analysis of Helix Elements in Umbilicals and Flexible Risers: Theory and Applications. Journal of Applied Mathematics, Det Norske Veritas AS Riser Technology, Høvik, 1363 Oslo, Norway, v. 2012, p. 1-23, 11 abr. 2012.
- [4] LI, Henan. Flexible Pipe Stress and Fatigue Analysis. Norwegian University of Science and Technology: Department of Marine Technology, [s. l.], Junho 2012.
- [5] DET NORSKE VERITAS. Sesam Helica. [S. l.], 6 jul. 2022. Disponível em: <https://www.dnv.com/services/umbilical-analysis-and-flexible-pipe-analysis-helica-69553>. Acesso em: 12 maio 2022.
- [6] FERGESTAD, Dag; LØTVEIT, Svein Are; SINTEF OCEAN; 4SUBSEA (ed.). Handbook on Design and Operation of Flexible Pipes. [S. l.: s. n.], 2017.
- [7] Relatório do Projeto "Análise de Risers Flexíveis à Flexão". CENPES-LNCC, [S. l.], p. 1-55, Setembro 2004.

1. DADOS GERAIS

Nesta Seção serão apresentadas as informações gerais do Trabalho

Título do Projeto: Técnicas de Ciências de Dados Aplicadas a Pesquisas Climáticas

Nome do Bolsista: Gabriel Tavares Da Silva Vale

Nome do Orientador: Fábio Andrade Machado Porto

Nome do Coorientador: Anderson Chaves Da Silva

Tipo de Bolsa: Iniciação Científica

Período do Relatório: 01/11/2021 21/06/2022

2. OBJETIVOS

Nesta seção serão apresentados os objetivos do Trabalho de Iniciação Científica.

Os objetivos do Trabalho Realizado consistem em:

1. Estruturar e Organizar de forma sistemática dados meteorológicos provenientes das diferentes fontes disponibilizadas através de parceiras do Laboratório Dextl, dando ênfase nos dados fornecidos pelo COR (Centro de Operações Rio).
2. Instalar e Compreender o funcionamento do SGBD Não Relacional matricial SAVIME[1], bem como entender das comandos utilizados para a criação de matrizes e inserção dos dados no sistema.
3. Continuar os experimentos realizados no artigo Managing Space Spatio-Temporal Data in SAVIME: an Evaluation of the Ph-Tree Index[4].
4. Analisar o impacto do fenômeno de Concept Drift no desempenho de modelos meteorológicos para aprendizado de máquina.

3. INTRODUÇÃO

Nesta seção será apresentada a informação necessária para a compreensão do Trabalho de Iniciação Científica.

Com o passar dos anos, os dados climáticos tornaram-se mais acessíveis beneficiando as atividades de previsão do tempo, devido ao fato da tecnologia IoT ter um grande impacto nos estudos meteorológicos. Com essa grande quantidade de dados climáticos disponíveis, faz-se necessário encontrar sistemas que consigam armazenar e analisar esses dados de forma precisa, visando obter um desempenho favorável. Sendo assim, o sistema SAVIME[1] se torna uma solução relevante para dados cuja natureza é melhor representada por múltiplas dimensões, pois, consegue armazenar e realizar consultas em Datasets com grandes volumes de dados multidimensionais de forma rápida e precisa comparado a outros modelos como no caso do modelo Relacional.

SAVIME(Simulation Analysis and Visualization in Memory) é um Banco de Dados Não Relacional baseado em array multidimensional desenvolvido no (LNCC) pelo Professor Hermano Lourenço Souza Lustosa. Apesar de ser um Sistema que está em desenvolvimento, o SAVIME já suporta algumas operações básicas, como, projeções, filtros, junções dentre outros. Segundo Lustosa[2020,p.8] “O SAVIME pode promover análise e visualização de dados científicos de forma declarativa, sem necessitar da execução de processos de conversão de dados custosos através do uso de um modelo de dados flexível.” .

O SAVIME possui como estrutura o modelo de dados chamado TARS(Typed Array Schema). TARS são estruturas criadas no SAVIME com o uso da Linguagem de Definição de Dados (DDL) e com o uso da Linguagem de Definição de armazenamento . LOAD SubTAR segundo Lustosa(2020,p.50) “a ponte entre os Datasets e os TARS, funciona como uma linguagem de Definição de Armazenamento permitindo com que os usuários especifiquem o layout de dados para matrizes”. O SAVIME possui três tipos de especificação de dimensão para o carregamento do SubTAR sendo elas:

Especificação de Dimensão Ordenadas: Dataset é associado ao subTAR conforme intervalo especificado, de forma ordenada.

Especificação de Dimensão Parcial: Apenas um subconjunto dos valores de dimensão é preenchido para determinada dimensão, mas este subconjunto permanece coerente com relação às demais.

Especificação de Dimensão Total: Todos os índices para todas as dimensões precisam ser armazenados explicitamente.

Ph-Tree é uma árvore de indexação de dados, que foi desenvolvida por Tilmann Zschke, Christoph Zimmerli e Moira C. Norrie. O experimento da PH-Tree foi publicado no artigo (Herrera, et al. 2021). A PH-Tree é constituída essencialmente em uma quadtree que usa Hypercubus, compartilhamento de prefixos e armazenamentos bit-stream. Com a abordagem da PH-Tree, evita-se a necessidade de tratar vários conceitos que muitas vezes são considerados necessários para se obter um desempenho eficiente em outras árvores multidimensionais de indexação de dados. A motivação para abordar a PH-Tree neste trabalho de Iniciação Científica, consiste no fato da realização da reprodução dos experimentos realizados no artigo Managing Space Spatio-Temporal Data in SAVIME: an Evaluation of the Ph-Tree Index[4].

4. METODOLOGIA DA PESQUISA

Nesta seção será detalhado o passo a passo de como foi feito para atingir os objetivos do Trabalho.

As informações necessárias para realizar a instalação do SAVIME estão disponíveis em: <https://dexllab.github.io/Savime/>. Para este trabalho de Iniciação Científica, o SAVIME foi instalado em uma Máquina Virtual com 20 GB de Armazenamento e 5GB de Memória RAM.

Os dados utilizados nos experimentos deste trabalho de Iniciação Científica são oriundos do Centro de Operações Rio (COR). O COR foi inaugurado em 2010, possui como propósito antecipar soluções, alertando os setores responsáveis sobre os riscos e as medidas urgentes que devem ser tomadas em casos de

emergências, como chuvas fortes, deslizamentos e acidentes de trânsito. O Dataset Cosmo utilizado é armazenado em formato .csv e possui um total de 47 milhões de linhas e consome um total de 3,84 GB de armazenamento. Ele mapeia toda a região da América do Sul, possui um total de 10 colunas sendo elas: Tempo, Estação, Latitude, Longitude, Vento Zonal, Vento Meridional, Temperatura, td, Pressão Reduzida, Precipitação, Radiação Onda_curta. No entanto, nem todos os dados que estão no arquivo serão necessários para a realização deste experimento. Portanto, houve a necessidade do código em PYTHON possuir um método para selecionar somente com as colunas que continham os dados necessários. O principal objetivo da elaboração deste código é, através do arquivo com os dados oriundos do COR, gerar um array multidimensional que armazenará os valores de chuva de acordo com as dimensões do mesmo. O SAVIME aceita a geração de Datasets com dados providos de arquivos, no entanto, o arquivo deve estar em formato binário. Sendo assim, como o arquivo com os dados oriundos do COR está no formato .csv, faz-se necessário gerar um código em PYTHON visando criar um método para gerar um array numpy com os dados do arquivo .csv e salvá-lo no formato binário para ser carregado no SAVIME. Os métodos utilizados para gerar o array numpy bem como os scripts com os comandos utilizados no SAVIME estão disponíveis em: <https://github.com/GabrielTavaresVale/EXPERIMENTO-02>.

5. RESULTADOS OBTIDOS

Nesta seção serão detalhados os resultados obtidos no trabalho de Iniciação Científica até o momento .

Conforme detalhado na seção de objetivos deste relatório, uma das propostas do Trabalho de Iniciação Científica consiste em continuar os experimentos do artigo *Managing Space Spatio-Temporal Data in SAVIME: an Evaluation of the Ph-Tree Index*[4]. No artigo em questão, os dados do arquivo .csv oriundos do COR são inseridos na PH-Tree e no SAVIME utilizando TARs com dimensões Totais explícitas, visando comparar o tempo de execução de três consultas com

os dados. A proposta deste trabalho é carregar os dados do artigo no SAVIME utilizando dessa vez dimensões ordenadas explícitas e implícitas e comparar o tempo de execução. Portanto, fez-se necessário reproduzir os experimentos realizados no artigo. A seguir serão detalhados os experimentos realizados, bem como os resultados.

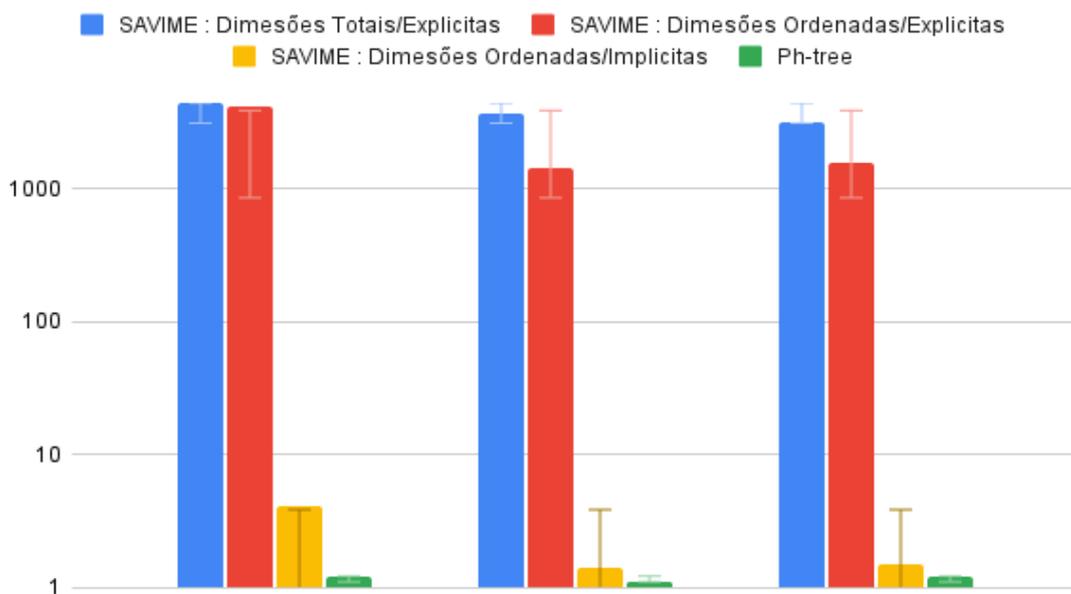
Experimento 1: Medir o tempo médio da execução das seguintes Point Queries

Query a: `subset(COSMO, data, 2001010000, 2001010000, latitude, -32.905, -32.905, longitude, -60.782, -60.782);`

Query b: `subset(COSMO, data, 2001010000, 2001010000, latitude, -22.373, -22.373, longitude, -50.975, -50.975);`

Query c: `subset(COSMO, data, 2012121500, 2012121500, latitude, -14.700, -14.700, longitude, -52.350, -52.350);`

Figura 2. tempo médio das Point-Queries SAVIME vs Ph-Tree



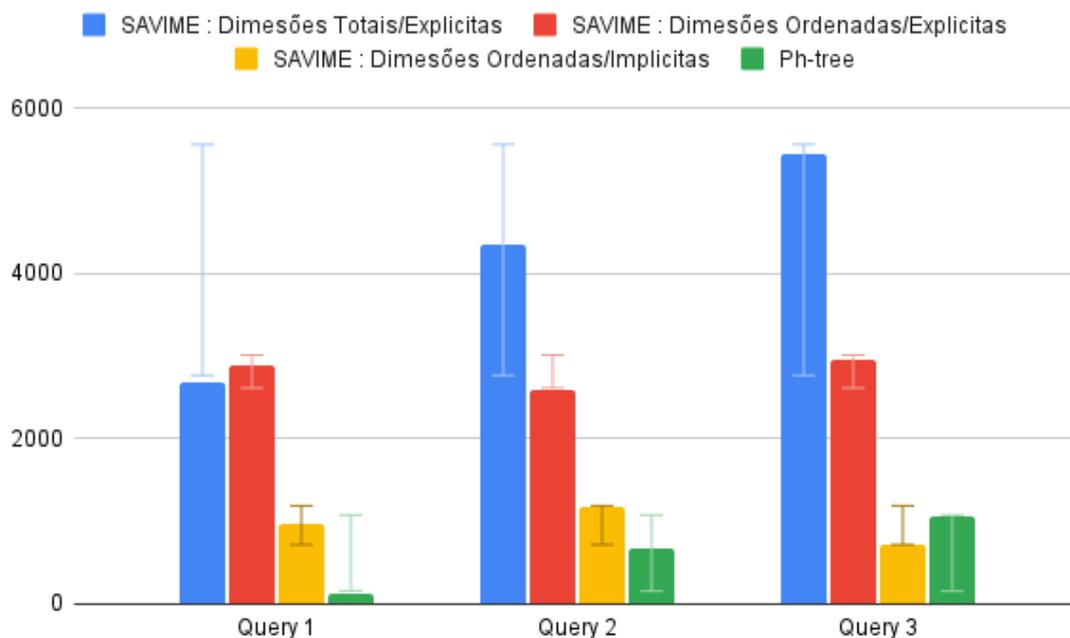
Experimento 2: Medir o tempo médio da execução das seguintes Range Queries

Query 1: subset(COSMO, data, 2001010000, 2003200000);

Query 2: subset(COSMO, data, 2001010000, 2008210000);

Query 3: subset(COSMO, data, 2001010000, 2012120000)

Figura 3. tempo médio das Range-Queries SAVIME vs Ph-Tree



Os resultados demonstram que, o SAVIME utilizando dimensões ordenadas implícitas consegue obter um tempo médio de desempenho comparável ou até superior a Ph-Tree,

6. CONCLUSÃO

Nesta subseção serão descritas as principais conclusões obtidas no trabalho de Iniciação Científica até o momento.

O que pode ser concluído deste trabalho de Iniciação Científica até o momento, consiste que o SAVIME é um Sistema recomendado para lidar com grandes volumes de dados Multidimensionais. Portanto, o carregamento dos dados do COR ocorreu de maneira extremamente rápida.

Os próximos passos deste Trabalho de Iniciação Científica consistem em integrar Modelos de Machine Learning ao SAVIME utilizando o operador Predict, visando obter análises climáticas mais precisas.

7. REFERÊNCIAS

- [1] Lustosa, H. L. S., Savime Enabling Declarative Array Processing in Memory, 2020.
- [2] Lustosa, H. L. S., Silva, A. C., da Silva, D. N. R., Porto, F. A. M., and Valdúriez, P. (2020). SAVIME: An array dbms for simulation analysis and ml models prediction. *Journal of Information and Data Management*, 11(3).
- [3] Herrera, StiW; Miguez Larissa; Reis, Paulo; Silva, Anderson; Porto, Fábio. Managing Sparse Spatio-Temporal Data in SAVIME: an Evaluation of the Ph-Tree Index, 2021.
- [4] Zashke, Tilmann; Zimmerli, Cristoph; Norrie, Moira . The Ph-Tree _ A Space – Efficient Storage Structure and Multi- Dimensional Index, 2014

RELATÓRIO DE ATIVIDADES

Título do Projeto: Exploração de Módulos Paralelo-Híbrido de Bioinformática para Ambientes GPU de Supercomputação

Nome do bolsista:

Guilherme Freire da Silva Dornelas

Nome do orientador:

D.Sc. Carla Osthoff Ferreira de Barros (Tecnologista Sênior – CENAPAD/LNCC)

Nome do coorientador:

D.Sc. Kary Ann del Carmen Ocaña Gautherot (Tecnologista Pleno – LABINFO/LNCC)

B.Sc Micaella Coelho Valente de Paula (Analista de TI – SEPAD/COTIC/LNCC)

Tipo de bolsa: PIBIC

Período do relatório: 01/09/2021 a 01/07/2022

INTRODUÇÃO

As plataformas de Computação de Alto Desempenho (CAD) como o supercomputador Santos Dumont SDumont permitem executar de uma maneira eficiente tarefas com grande demanda de processamento. O SDumont possui uma arquitetura híbrida com CPU *multi-core* e dispositivos com arquitetura *Many core*, GPU e Metodologias de Investigação Científica (MIC), é usado de forma intensiva por vários grupos de pesquisa brasileiros.

O Bioinfo-Portal (<https://bioinfo.lncc.br/>) é um ambiente computacional que gerencia a execução automática de aplicações e dados científicos de bioinformática em larga escala, como apoio às pesquisas da comunidade científica de informática através de uma interface *Web* amigável e interativa. Bioinfo-Portal utiliza, via serviços *Web*, o *middleware* CSGrid como *framework* de integração à arquitetura do SINAPAD. O Bioinfo-Portal está hospedado e gerenciado pelo LNCC e usa recursos e tecnologias de computação de alto desempenho, como o supercomputador Santos Dumont (SDumont, <https://sdumont.lncc.br/>) a fim de diminuir o tempo de processamento das aplicações.

Este projeto faz parte da colaboração entre o CENAPAD e o LABINFO, ambos pertencentes ao LNCC, onde são realizadas diversas pesquisas no apoio às análises computacionais envolvendo bioinformática, biologia computacional e CAD.

OBJETIVOS

- O projeto visa levantar um estudo de desempenho de aplicações de filogenômica e evolução molecular computacional em *multi* GPU no ambiente de computação de alto desempenho (CAD), mais especificamente nos recursos computacionais do SDumont.

- Estabelecer uma análise comparativa do BEAST acoplado a BEAGLE 3, nas suas versões CPU, GPU e *multi- CPU/GPU*.
- Integrar os programas BEAST, BEAST2 e RAXML para o SDumont. Explorar esses programas nos ambientes CPU e GPU do SDumont.
- Realizar análises de desempenho e escalabilidade, visando otimizar o uso do ambiente computacional.
- Efetuar execuções do programa RAXML, com alternância de *threads*, do *bootstrap* e variação dos nós computacionais do SDumont.

METODOLOGIA

Esse projeto está realizando a integração de ferramentas de bioinformática em clusters de supercomputadores nos sistemas de CPU e GPU.

Adicionamos um novo programa ao projeto, a aplicação de bioinformática RAXML, no momento estão sendo realizados novos testes do programa junto ao ambiente do SDumont.

O BEAST é um programa de análise filogenômica baseada em inferência Bayesiana, multiplataforma de sequências moleculares utilizando os métodos de *Markov Chain Monte Carlo* (MCMC).

A BEAGLE é uma biblioteca de alto desempenho, que faz uso de processadores altamente paralelos, como aqueles em placas gráficas (GPUs). Está acoplada ao programa (BEAST) para tornar mais eficiente a paralelização em escala fina de cálculos de probabilidade filogenética.

O RAXML é um programa para inferência sequencial e paralela, baseada em Máxima Verossimilhança de grandes árvores filogenéticas. Pode ser usado para pós-análises de conjuntos de árvores filogenéticas, análises de alinhamentos e colocação evolutiva de leituras curtas.

O BEAST 1.8 e BEAGLE 3 foram acoplados no SDumont e executados em diferentes cenários nas filas CPU, GPU e *multi-CPU/GPU* levando em consideração as características e natureza dos dados e parametrização das aplicações.

Dados moleculares do vírus da Dengue em formato XML foram extraídos do diretório benchmark do BEAST 1.8 e usados nas análises. O parâmetro *chainLength* do BEAST 1.8 está relacionado ao cálculo das cadeias MCMC. O incremento no valor do *chainLength* fornece consistência as análises bayesianas, mostrando-se proporcional ao incremento no tempo computacional. As execuções foram realizadas usando CPU 24 *threads*, CPU 40 *threads*, 1 GPU, CPU 40 *threads*/1 GPU, CPU 40 *threads*/8 GPU, e 8 GPU, conforme a Figura 1.

Para os cálculos foi usado como variabilidade o parâmetro *chainLength* fixado em “100000” e “10000000”. Os experimentos sugerem que as características como tamanho dos dados e configuração de parâmetros no BEAST, como o *chainLength*, influenciam no tempo computacional.

As execuções usando um valor baixo no parâmetro *chainLength* = “100000” é exploratório e requer menos gasto computacional. Quando usado um valor maior de *chainLength* = “10000000” tende a gerar uma maior consistência nos resultados, os cálculos de probabilidade requeridos tornam-se mais exaustivos levando a um gasto computacional maior. Nos resultados, o incremento do *chainLength* influencia no maior tempo computacional obtido, mesmo assim ambas as execuções de *chainLength* apresentaram um comportamento muito similar no uso dos recursos CPU e GPU.

Efetuamos testes com o programa RAXML, utilizando dados de Aminoácido (aminoacido.phylip), conforme a Figura 2, apresenta as execuções com 1 nó computacional; o *bootstrap*

alternando em 100, 500, 1000, 1500 e 2000; as *Threads* selecionadas foram 4, 8, 12, 24. Coletamos os dados relacionados ao tempo total de execução (TTE).

RESULTADOS

Realizamos testes com dados do vírus da Dengue em (XML), com o programa BEAST, conforme a Figura 1, demonstra essa análise, fixamos o parâmetro *chainLength* em “100000” como apresentado na coluna da esquerda e em “10000000” na coluna da direita. Selecionamos as variações entre CPU 24 *threads* (azul), CPU 40 *threads* (laranja), 1 GPU (amarelo), CPU 40 *threads*/1 GPU (verde), CPU 40 *threads*/8 GPU (roxo), e 8 GPU (azul claro). O melhor desempenho do tempo total de execução (TTE) foi obtido com 8 GPU.

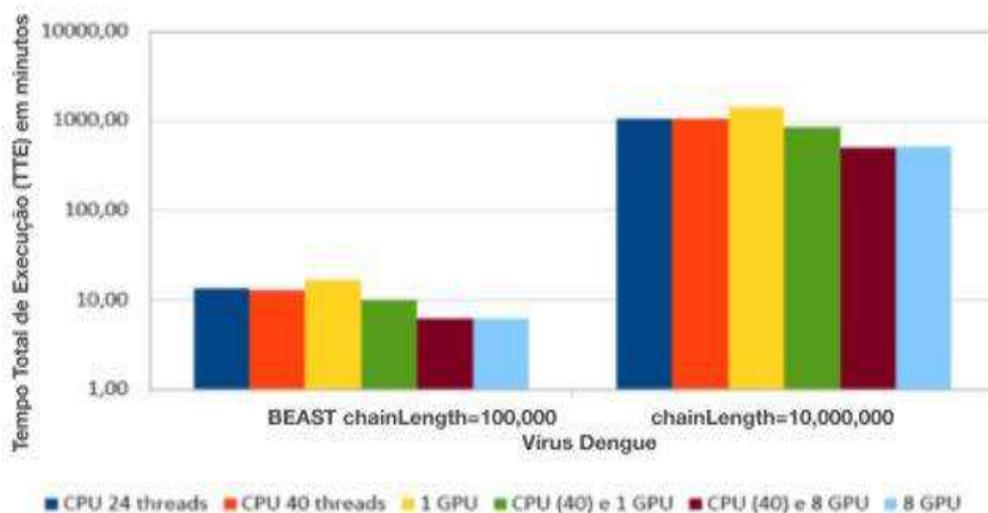


Figura 1. Análise de desempenho do BEAST/BEAGLE no CPU/GPU do SDumont

O *software* RAXML têm funções de suporte híbrido para comunicação de dados MPI e *threads*, iremos apresentar análises dessa ferramenta junto ao ambiente do SDumont. Os testes do RAXML, foram com dados de Aminoácido (*aminoacido.phylip*), a execução em 1 nó conforme, a Figura 2, publicada em [2], através do gráfico apresentamos os dados do tempo total de execução (TTE). O *bootstrap* ganhou alternância entre 100 (linha azul), 500 (linha vermelha), 1000 (linha verde), 1500 (linha roxa) e 2000 (linha azul claro). Foram selecionados para tais execuções 4 *threads*, 8 *threads*, 12 *threads*, 24 *threads*. O melhor tempo total de execução (TTE) foi do parâmetro de 100 *bootstrap* e 24 *threads*.

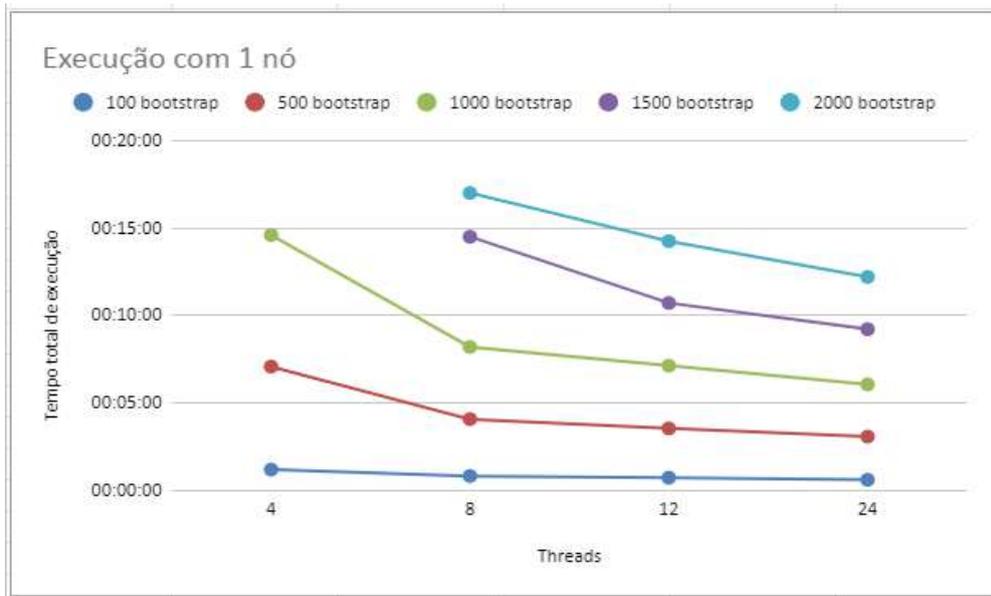


Figura 2. Execução do RAXML com 1 nó

A Figura 3, publicada em [3], temos um gráfico do programa RAXML, utilizando os dados de Aminoácido (aminoacido.phylip), com o *bootstrap* alternando em 100 (linha azul), 500 (linha vermelha), 1000 (linha verde), 1500 (linha roxa) e 2000 (linha azul claro); esses testes foram de execuções em até 4 nós fechados (24 threads); os nós selecionados foram 1, 2 e 4. Obtemos o melhor tempo de execução com o parâmetro de 100 *bootstrap* e 4 nós computacionais.

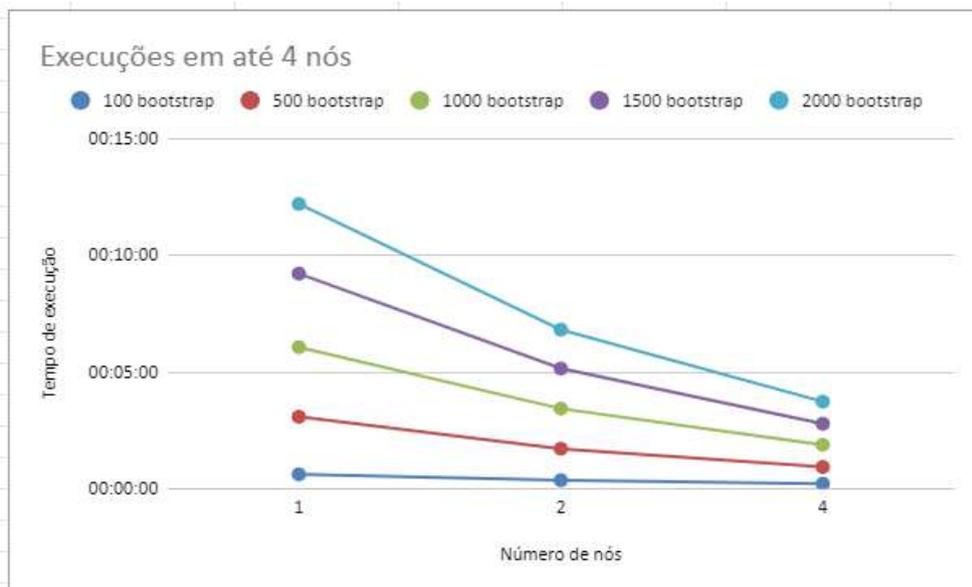


Figura 3. Execução do RAXML em até 4 nós

CONCLUSÕES

O presente estudo viabiliza a exploração e análise de desempenho do BEAST/BEAGLE e do RAXML em ambientes de CAD com a especificação do ambiente computacional que leve a um desempenho mais eficiente. Dessa maneira, permite que usuários possam usufruir dessas informações e realizar execuções garantindo um uso racional do ambiente do SDumont.

Análises de desempenho do BEAST/BEAGLE em múltiplas configurações de CPU/GPU no SDumont sugerem o uso da configuração híbrida CPU 40 threads e 8 GPU como a mais eficiente. Sobre a variabilidade no número de *chainLength* fixados em “100000” e “10000000” como esperado o tempo computacional gasto na execução é incremental, mas o comportamento é muito similar.

Este relatório apresenta estudos de desempenho de aplicações científicas BEAST/BEAGLE, em diversos ambientes computacionais do Santos Dumont: executado em paralelo em 1 nó com CPU *multi-core*, em 1 nó com arquitetura GPU, em um ambiente híbrido executando em paralelo na CPU *multi-core* e na GPU e em um ambiente de sistema distribuído, escalando BEAGLE/BEAST de 1 até 8 nós com arquitetura GPU, conforme Figura 1, publicada em [1]. Os estudos mostram que, para as configurações dos testes realizados para a aplicação BEAST/BEAGLE, a execução de maior eficiência pelo portal é realizada ao utilizar 1 nó CPU *multi-core*.

Os testes do RAXML, estabelecem, conforme Figura 2, publicada em [1], como melhor desempenho, a execução de 100 *bootstrap* e 24 *threads*, estes dados coletados serão aplicados numa ferramenta inteligente (IA), que está sendo desenvolvida para integrar o Portal-Bioinfo.

O BEAST, BEAGLE e RAXML estão integrados ao Portal-Bioinfo (<https://bioinfo.lncc.br/>) como aplicações de bioinformática, apresentando o desenvolvimento de portais científicos verdes e eficientes.

O objetivo deste projeto é o desenvolvimento de estudos de desempenho das aplicações e dos *workflows* científicos do Portal Bioinfo para que os mesmos possam vir a ser executados de forma eficiente, versátil, escalável e inteligente pelo Bioinfo-Portal.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Ocana, K., Coelho, M., Freire, G., and Osthoff, C. (2020). High-performance computing of beast/beagle in bayesian phylogenetics using sdumont hybrid resources. In 14^o BreSci – Brazilian e-Science Workshop.

• C.-L. Hung, Y.-S. Lin, C.-Y. Lin, Y.-C. Chung, e Y.-F. Chung, “CUDA ClustalW: An efficient parallel algorithm for progressive multiple sequence alignment on Multi-GPUs,” *Comput Biol Chem*, vol. 58, pp. 62– 68, May 2015.

• Yin, Z., Lan, H., Tan, G., Lu, M., Vasilakos, A. V., and Liu, W. (2017). Computing platforms for big biological data analytics: Perspectives and challenges. *Computational and Structural Biotechnology Journal*, 15:403–411.

• Jin, Z. and Bakos, J. D. (2013). Extending the beagle library to a multi fpga platform. *BMC Bioinformatics*, 14(1):25.

• J. P. Huelsenbeck e F. Ronquist, “MRBAYES: Bayesian inference of phylogenetic trees,” *Bioinformatics*, vol. 17, no. 8, pp. 754–755, Aug. 2001.

• ERAD RJ - Exploração de Módulos Paralelo Híbrido de Bioinformática para Ambientes GPU de Supercomputação - 1 de dezembro de 2020 Autores: G Dornelas, M Coelho, K Ocaña, C Osthof

• Ocaña K, Coelho M, Terra R, Freire G, Santos M, Cruz L, Galheigo M, Carneiro A, Fagundes B, Carvalho D, Cardoso D, Meneses E, Gadelha L, Osthoff C, DEVELOPING EFFICIENT SCIENTIFIC GATEWAYS FOR BIOINFORMATICS

INSUPERCOMPUTER ENVIRONMENTS SUPPORTED BY ARTIFICIAL INTELLIGENCE. In: (ISC High Performance 2021)

Arcabouço Computacional para Simulações Numéricas apoiadas por Aprendizado de Máquina

Projeto PIBIC/LNCC

Candidato: João Victor Caccavo de Araújo Wanderley (j.caccavo@eq.ufrj.br)

Orientador: Antônio Tadeu Azevedo Gomes (atagomes@lncc.br)

Co-orientador: Frédéric Valentin (valentin@lncc.br)

Palavras chaves: engenharia de software, aprendizado de máquina, métodos numéricos

Área: 1.03.00.00-7 Ciência da Computação

Subárea: 1.03.03.00-6 Metodologia e Técnicas da Computação

Objetivos específicos

O objetivo geral do trabalho é desenvolver uma base computacional que permita realizar uma integração de modelos preditivos elaborados utilizando aprendizado baseado em dados (ABD) e modelos elaborados utilizando equações diferenciais (EDs). Quando se trabalha com técnicas de ABD, como máquinas de vetor de suporte e redes neurais, existem algumas dificuldades que surgem quando há uma escassez de dados, especialmente em modelos muito complexos. Por outro lado, em abordagens baseadas somente em EDs o modelo precisa capturar as interações no problema em questão, as EDs precisam ser aproximadas numericamente utilizando métodos computacionais, como elementos finitos, e tanto as EDs quanto os métodos computacionais precisam ser parametrizadas corretamente, além de apresentarem alto custo computacional em modelos complexos. Portanto, integrar essas técnicas pode trazer muitos benefícios, tanto na precisão dos modelos preditivos, quanto na redução do custo computacional.

Diferentes trabalhos têm mostrado quão útil é a combinação de ABD e EDs. Por exemplo, em [1] e [2] as EDs são incorporadas ao kernel de processos gaussianos; já em [3] e [4] as EDs são incorporadas à função de perda de redes neurais multicamada, criando o que vem sendo denominado na literatura como “redes neurais informadas por física” (*Physics-Informed Neural Networks* – PINNs). Porém, podemos observar que existem muitos trabalhos, incluindo os citados, explorando os aspectos teóricos dessa união, e poucos detalhando os arcabouços computacionais que exemplificam como este potencial

pode ser melhor explorado na prática.

O objetivo específico deste trabalho é investigar como a integração dos mecanismos existentes tanto para predições baseadas em EDs, quanto para simulação numérica de EDs pode ser utilizada na construção de modelos preditivos. Como principal estudo de caso será analisada a integração de PINNs implementadas com a biblioteca de código aberto TensorFlow¹ (ABD) a simuladores numéricos baseados em métodos de elementos finitos (EDs) multiescala implementados na biblioteca MSL, desenvolvida pelo grupo de pesquisa IPES do LNCC.²

Metodologia

A biblioteca TensorFlow é normalmente usada através de requisições à API feitas com a linguagem Python, tanto para o treinamento dos modelos, quanto para realização de previsões. Já a biblioteca MSL é desenvolvida em C++. Por esse motivo, o desafio tecnológico inicial é a integração destas duas tecnologias. O outro desafio, já no longo prazo, é a proposição de um arcabouço genérico que permita que a combinação pretendida entre ABD e EDs possa envolver no futuro diferentes componentes de software além dos aqui citados.

A metodologia para lidar com esses desafios combina atividades de revisão da bibliografia já existente tanto sobre a parte teórica dos ABDs, quanto sobre as técnicas de simulação numérica de EDs e atividades práticas de desenvolvimento de software para explorar o ferramental teórico e computacional disponível.

Atividades Realizadas

A primeira parte do projeto consiste no estudo das técnicas de ABD, que está sendo realizado a partir da leitura de [5]. A parte introdutória do livro discorre sobre a parte histórica, e a base matemática das redes neurais artificiais, e já aborda de maneira superficial o tópico do Deep Learning, tema central do livro.

Junto ao estudo teórico, foram realizadas reuniões semanais para discussão dos tópicos presentes no livro, e acompanhamento do progresso. Além disso, foi elaborado um código para criação de uma classe em Python que pode ser usada para treinar e testar uma rede neural simples, como maneira de aplicar os conhecimentos teóricos aprendidos até o momento.

Referências bibliográficas

- [1] M. Alvarez, D. Luengo e N. Lawrence, “Linear Latent Force Models Using Gaussian Processes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 2693- 2705, 2013.

¹ <https://www.tensorflow.org/>

² <http://ipes.lncc.br/>

- [2] Z. Wang, W. Xing, R. Kirby e S. Zhe, "Physics Regularized Gaussian Processes," arxiv, 2020.
- [3] M. Raissi, P. Perdikaris e G. Karniadakis, "Physics Informed Deep Learning (Part I): Data driven Solutions of Nonlinear Partial Differential Equations," arxiv, 2017.
- [4] S. Mishra e R. Molinaro, "Estimates on the generalization error of Physics Informed Neural Networks (PINNs) for approximating a class of inverse problems for PDEs," 2020.
- [5] Data Science Academy. Deep Learning Book, 2022. Disponível em: <<https://www.deeplearningbook.com.br/>>. Último acesso em: 1 de Julho, 2022.

Coletor de amostras de dossel embarcado em veículo aéreo não tripulado

Bolsa: PIBIC - LNCC, Período do relatório: 2 meses

Aluno: João Vitor R. Rebello¹

Orientadores: Jauvane Cavalcante de Oliveira, PhD² ; Luis Claudio Batista da Silva, D.Sc.¹

¹Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET-RJ) Campus Petrópolis

²Laboratório Nacional de Computação Científica (LNCC) - Petrópolis - RJ

1. Objetivo

O objetivo deste projeto é estudar e desenvolver atuadores e efetadores robóticos aplicados no desenvolvimento de um robô do tipo veículo aéreo não tripulado (VANT) para coleta de amostras de folhagem no topo das árvores, conhecido como dossel. Sendo assim, para dar início a este processo fez-se necessário estudar cuidadosamente os temas relacionados a introdução a robótica e assuntos mais específicos que dizem respeito as tarefas do robô. Dito isso, este relatório tratará de explicar e detalhar os assuntos estudados e aprofundados ao longo de 2 meses que serão a base para prosseguimento do projeto em si.

2. Introdução

A principal ideia do projeto é que estas aeronaves do tipo VANT substituam o uso de trabalho manual para coleta de amostras no topo das arvores, evitando assim riscos acidentais para o ser humano, visto que a prospecção atual de amostras desse tipo é feita manualmente.

Por tratar-se então de um robô, é necessário que sejam feitos estudos sobre temas relevantes na robótica. Portanto, seguindo o cronograma proposto para nosso projeto, os 3 primeiros meses de desenvolvimento seriam para estudar e aprofundar o conhecimento do aluno em relação a robótica geral e específica, que são assuntos essenciais para o projeto.

3. Metodologia

Para estudo dos assuntos introdutórios, primeiramente foi montada uma reunião com os orientadores na qual ficaria acordado que, a cada semana, fosse realizada uma apresentação (em modo online ou presencial) de temas propostos pelos professores em relação a robótica geral, em formato de seminário.

Nesta apresentação semanal, seriam discutidos os assuntos ali apresentados, indicando possíveis falhas do aluno e sugerindo leitura de determinados materiais e revisão de conceitos mostrados, também seriam sanadas possíveis dúvidas que o aluno tenha em relação ao tema da semana.

Logo após, um novo tema seria apresentado pelo orientador, seus conceitos, significado e palavras chave. Também seria indicados livros e revistas nos quais poderiam ser feitas as consultas para criação do material da nova apresentação.

4. Resultados

Portanto, os mais diversos assuntos introdutórios à robótica foram pesquisados e estudados pelo aluno, como por exemplo:

- Termo Robótica: origem do termo, conceito de um robô, grandes nomes na área.
- Componentes de um robô: atuadores, Efetuadores, Controladores, Sensores, Corporalidade.
- Exemplos de robôs no dia a dia: evolução e facilidade de trabalho
- Principais problemas da robótica: aceitação da sociedade, vale da estranheza, segurança, privacidade de informações.

A partir desses temas, foi então proposto ao aluno o estudo de temas mais específicos, todos eles sendo subcategorias dos temas iniciais indicados. São eles:

- Corporalidade: tipos de materiais para construção de robôs, tipos de elos, tipos de juntas.
- Sensoriamento: tipos de sensores, conceitos de mapeamento de local, localização, proporções sensores X efetividade, interpretação dos dados.
- Módulos robóticos: quais tarefas realizam, como podem ser implementados em um projeto.
- Manipuladores: o que são, suas propriedades e como adaptá-los a determinadas tarefas.
- Movimentos robóticos: relações entre elos e juntas, conceito de Graus de liberdade, proporção Graus de liberdade X controlabilidade, uso de atuadores específicos e como isso afeta o movimento.
- Níveis de automação: como são definidos, interação humano-robô, aprendizado de máquina e controladores.

Com estes tópicos estudados, foi então mostrado como são separadas as grandes da robótica: terrestre, marítima, aérea e virtual. Tendo em vista que certas áreas utilizam conceitos, componentes e tipos de abordagem diferentes umas das outras, são então separadas em sub-áreas, na qual foram estudados:

- Robótica industrial: aplicações de manipuladores específicos (pouca interação humana, mais resistentes), relação de robôs com eficiência, padronização e segurança aos empregados.
- Robótica Humanoide: aplicação do vale da estranheza, sugestões de aceitação humana, relação de movimentos humanos com graus de liberdade e complexidade.
- Robótica de serviço: aplicação de sensores, definição de trajetória, conceito localização e mapeamento simultâneo (SLAM), ambiente dinâmico e estático.
- Robótica móvel: relação entre eficiência, diferentes atuadores e corporalidade do robô, robôs de coordenadas diversas (cartesianas, esféricas, cilíndricas) e robôs de articulações verticais e horizontais.

Notou-se também que, essas grandes áreas, mesmo que diferenciando-se em pontos específicos, ainda assim compartilham de inúmeros conceitos e aplicações iguais as outras, fazendo com que um robô qualquer possa pertencer a mais de uma categoria e/ou sub-categoria ao mesmo tempo.

Também foi pesquisado o quão comum é a robótica no presente: quais são as peças e conceitos mais aplicados no mercado, relação entre confiabilidade e valor de atuadores e sensores, como grandes empresas aplicam robôs em seus negócios e previsões de ascensão da robótica na sociedade.

5. Conclusão

Com os resultados de pesquisa e aprendizado, conclui-se que há base bibliográfica e conceitual para dar prosseguimento ao cronograma do projeto. Além de que este aprendizado será aprimorado ao longo do tempo com novos conteúdos. Sendo então os 3 meses seguintes com a tarefa que será de estudo de tecnologias de robótica embarcada.

Referências

- Cocota, J. (2013). *Elementos de Robótica (Apostila) – Engenharia de Controle e Automação*. Universidade Federal de Ouro Preto.
- Craig, J. J. (2013). *Robótica*. São Paulo: Pearson.
- Géron, A. (2019). *Hands-on Machine Learning with scikit-learn, Keras TensorFlow*. O'Reiley.
- Mataric, M. (2014). *Introdução à Robótica*. Editora UNESP, 1ª edição.
- Rosário, J. M. (2005). *Princípios de mecatrônica*. São Paulo: Prentice Hall.
- Russell, S. J. e Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Prentice Hall, 4 edição.
- Simplício, P. V. G., Lima, B. R., e Junkes, J. A. (2016, ISSN 1980-1777). Manipuladores robóticos industriais. Cadernos de graduação.
- Tinwell, A. (Consultado em 12 de Junho de 2022, ISBN 9781466586956). The uncanny valley in games and animation.

Previsão meteorológica utilizando métodos de inteligência artificial

Júlia Neumann Bastos¹, Rafael S. Pereira², Fabio A. M. Porto¹

¹DEXL – Laboratório Nacional de Computação Científica (LNCC)
Av Getúlio Vargas 333 Petrópolis Brasil

²Data Science Department – Just A Little Data
São Paulo, Brazil.

julia@lncc.br, rafael.pereira@just.bi, fporto@lncc.br

1. Objetivos

O objetivo desse trabalho é avaliar a qualidade de modelos preditivos baseados em redes neurais profundas para o problema de previsão meteorológica. Com base nisso, avaliaremos o desempenho do modelo em estudo, para diferentes regiões do espaço. Caso haja regiões onde o desempenho se mostre insatisfatório, investigaremos suas limitações para previsões nessas regiões tentando superar os problemas encontrados.

2. Introdução

As mudanças climáticas se referem a transformações de longo prazo nos padrões de temperatura e clima na terra. Como se tem observado, tais mudanças colocam em risco a vida no planeta. Desta forma, acompanhar tais mudanças e seus impactos no meio-ambiente é uma tarefa muito importantes. A ocorrência destas acarreta no desalojamento de múltiplos indivíduos, assim como perdas em safras agrícolas, afetando financeiramente a população.

Por esses motivos, a previsão meteorológica, principalmente envolvendo eventos extremos, se mostra bastante importante, permitindo o planejamento de ações que minimizem os danos a população, às propriedades e aos ambientes públicos. Tendo em vista que a execução de modelos numéricos de previsão meteorológica atualmente utilizados são extremamente custosos, e requerem tempo para a aquisição de dados [Leticia Braga Berlandi e Analice Costacurta Brandi 2022], existe interesse em modelar esse problema via métodos de aprendizado profundo.

Optamos neste trabalho pela utilização de redes neurais profundas do tipo ConvLSTM como arquitetura para predição de chuva, como adotado em [Souto et al. 2018]. De forma a adequar a distribuição espacial dos dados, alocados de forma irregular na superfície, construímos um algoritmo de pre-processamento de dados que realiza uma interpolação espacial entre os dados de pluviômetros para alcançar uma grade regular a ser utilizada como entrada no treinamento da rede neural.

O restante deste relatório encontra-se estruturado da seguinte forma. Na seção 3 descrevemos a metodologia utilizada na solução deste problema. Na seção 4 apresentamos os resultados experimentais obtidos e na seção 5 terminamos com algumas conclusões.

3. Metodologia

Nesta seção discutimos a metodologia usada para atacar este problema. Primeiramente discutiremos sobre o processo de extração dos dados. Em seguida, apresentaremos a etapa de pré-processamento de dados. Na seção seguinte, discutimos o tratamento dado à natureza esparsa dos dados, permitindo sua utilização em um modelo espaço temporal, e por fim descrevemos o processo de treinamento e avaliação do modelo.

3.1. Extração e natureza dos dados

Os dados utilizados nesse projeto foram cedidos pelo Centro de Operações do Rio (COR), no contexto da parceria de pesquisa que [Prefeitura do Rio de Janeiro 2022], com a participação do Laboratório Nacional de Computação Científica (LNCC) e do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (Cefet/RJ). A parceria tem por objetivo o aprimoramento da qualidade preditiva das condições meteorológicas da cidade. A partir dessa parceria, tivemos acesso aos dados referentes aos registros de precipitação nas estações meteorológicas do COR espalhadas pela cidade do Rio de Janeiro.

O período escolhido para estudo foi do dia 15 de fevereiro de 2022, uma terça-feira, data que marcou o acontecimento de evento extremo na cidade de Petrópolis, acarretando em inúmeras mortes e perdas materiais que assolou a cidade, na região serrana. Os dados começaram a ser extraídos a partir de 13hrs e continuaram até as 23hrs deste mesmo dia. Foram realizadas medições a cada 2 minutos. O evento começou a ocorrer a partir do meio deste período, com duração de aproximadamente 3hrs e registrou mais de 250 milímetros de chuva [Nathan Lopes - UOL Notícias 2022].

Os dados foram disponibilizados por meio de um arquivo no formato csv, que consiste em um tipo de arquivo de texto fundamental para transferência de informações. O arquivo apresenta os seguintes atributos: o código de cada estação meteorológica presente na área geográfica de Petrópolis; a Unidade Federativa (UF); o nome de cada estação; sua localização em latitude e longitude; a data e hora de cada medida; e por fim, o valor de precipitação medido e os respectivos instantes de tempo. Sendo assim, ao todo foram localizadas 21 estações meteorológicas. Destas, nem todas estavam em funcionamento e não mediram em todos os instantes de tempo, não seguindo assim seguir um padrão. Por estes motivos o pré-processamento de dados foi de extrema importância, após o estudo dos dados.

3.2. Pré-processamento

Nesta seção discutiremos processos aplicados nos dados para podermos usá-los no processo de treinamento do modelo.

O primeiro ponto observado no dataset fornecido, é que as diferentes estações presentes na região de Petrópolis não realizam todas suas medidas ao mesmo tempo. Observando a variação entre elas e tendo uma frequência mínima de captura de uma hora, resolvemos adotar esta frequência para os dados. Ao tomar esta decisão, consideramos a coluna que contém a data e hora da medição, e arredondamos para a hora mais próxima, após isto tomamos a média das medidas pelo par (hora, estação), assim padronizando para as estações com frequência de captura maior. Após esta etapa, fazemos um teste de valores inválidos, removendo possíveis valores negativos na medida de chuva, pois estas

indicariam defeito no sensor na hora da medida. Por fim, calculamos o número de estações mínimas que funcionaram durante o período analisado, pois este seria um limitador para o número de vizinhos utilizado no processo de interpolação discutido a seguir

3.3. A natureza esparsa dos dados e o processo de interpolação

Após obter os dados descritos na seção 3.2, temos as medidas associadas a cada estação. Porém, como pode ser observado na figura 1, as estações meteorológicas estão localizadas em apenas para alguns pontos do espaço. Desta forma, para obtermos uma grade regular necessária ao treinamento de modelos convolucionais, utilizamos um algoritmo de interpolação derivado de [Souto et al. 2018] e baseado no algoritmo KNN (K-nearest neighbors). O passo a passo do algoritmo é descrito na figura 2. Foi utilizada uma discretização produzindo uma grade regular de 500 por 500 metros, definida após discussão com especialistas da área.



Figura 1. Grid utilizado, pontos vermelhos demonstram delimitação da cidade e pontos azuis localizações das estações

Fonte: Elaborada pelo autor.

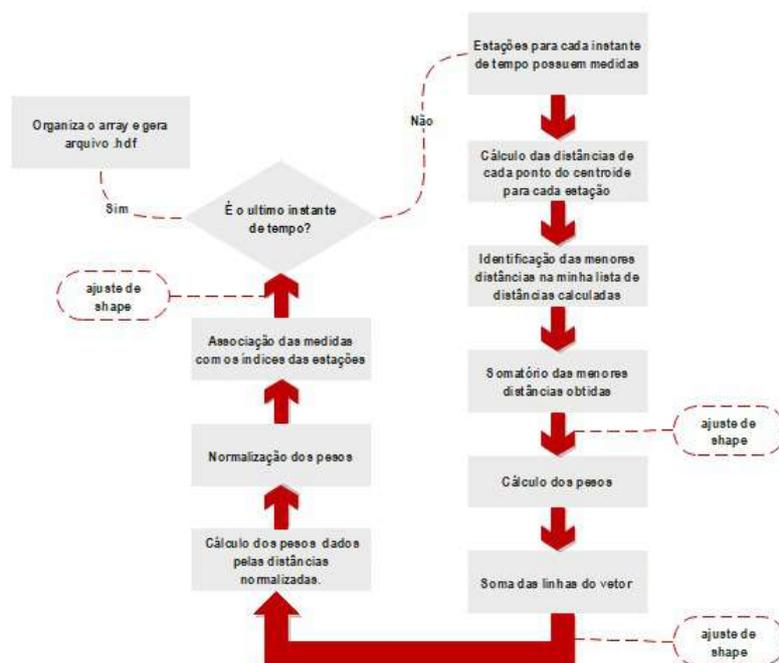


Figura 2. Processo de interpolação baseado em algoritmo *KNN*

Fonte: Elaborada pelo autor.

3.4. Construção do modelo preditivo

Para modelagem, nós utilizamos uma arquitetura baseada na arquitetura de redes neurais profunda ConvLSTM. A adoção desta arquitetura é inspirada no trabalho de [Souto et al. 2018], e tem como objetivo capturar as relações espaciais do fenômeno a partir da convolução enquanto captura a relação temporal utilizando a recorrência das camadas *Long Short Term memory*.

Neste tipo de arquitetura, o modelo processa não apenas o passado de múltiplas series com o modelo recorrente, mas também se utiliza da convolução para associar series localizadas espacialmente próximas para melhor entendimento do fenômeno que estamos modelando. Este tipo de modelo é normalmente utilizado em problemas espaço temporais. Como exemplo, podem-se citar problemas meteorológicos, processamento de vídeos, entre outros. Neste trabalho, utilizamos de uma segunda camada que é a convolução tridimensional. Esta tem como função colapsar as informações temporais adquiridas pela ConvLSTM para o 'espaço de medidas'. Assim obtendo o tensor final na estrutura desejada. No estudo aqui apresentado, utilizamos de um modelo com duas camadas ConvLSTM, com 32 e 64 filtros, assim como, uma camada Conv3D [Tran et al. 2015], com número de filtros igual ao número de atributos modelados. Nós utilizamos de *zero padding* em todas as camadas para não ocorrer redução de dimensionalidade espacial. Uma vez que queremos projetar o futuro de toda esta região. A seguir descrevemos os processos para treinamento do modelo.

3.5. Treinamento e avaliação do modelo

Nesta seção discutimos o processo de treinamento do modelo. Ao final da etapa de interpolação, discutida na seção 3.3, produzimos um arquivo no formato *h5* [The HDF Group]. O tensor assim construído é estruturado nas seguintes dimensões: tempo, espaço, espaço e amostra, já separado em treino, validação e teste. O dado é passado para um processo *gerador* que reajusta o tensor utilizando as 10 últimas medidas para prever a próxima. Para resolver o problema de otimização dos pesos, utilizamos da biblioteca *keras* e otimizamos o modelo por 20 épocas, escolhendo o resultado que minimiza a perda de validação. Reportaremos os resultados iniciais na seção 4. Uma síntese do processo de treinamento do modelo pode ser vista na Figura 3.

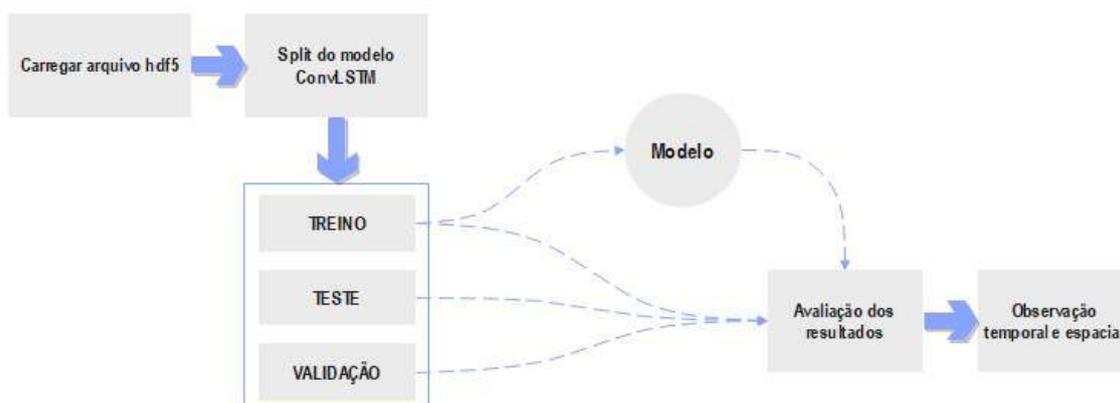


Figura 3. Processo de treinamento e avaliação do modelo

Fonte: Elaborada pelo autor.

4. Resultados e discussão

Nesta seção apresentamos os resultados iniciais obtidos a partir da avaliação do modelo. Na Figura 4 vemos a previsão inicial do modelo comparado ao valor interpolado a partir das medidas reais obtidas pelos pluviômetros utilizando as três estações mais próximas. Podemos observar que no intervalo de tempo analisado para o evento extremo, o modelo não conseguiu capturar e realizar a previsão do momento em que o evento ocorreu. No restante do tempo distribuído em um local do espaço, o modelo conseguiu seguir razoavelmente bem a curva dos valores reais observados. Isso mostra que o uso de apenas um atributo como base (por exemplo, volume de precipitação) para previsão de eventos extremos é insuficiente para se atingir uma previsão com boa qualidade, considerando-se a base de dados com os valores já interpolados e com uma medida de chuva como atributo.

Uma forma de resolver este problema pode ser incluir outros atributos que baseados em outras fontes de dados, como talvez a velocidade do vento. Eventos dessa magnitude muitas das vezes devem ser estudados por modelos multimodais, aos quais conseguiriam ver qual a relação de cada fonte no resultado obtido. A partir dos sinais de diferentes fontes de dados sobre o mesmo fenômeno poder-se-ia utilizar as medidas necessárias que tenham mais interferência sobre o evento extremo. Outra forma de observar o erro de predição se dá quanto à sua variação no espaço, como pode-se visualizar na figura 5. Podemos assim observar as limitações do modelo ao observar concentrações de erro em certas regiões do espaço, que em alguns cortes temporais fica evidente que a concentração do erro foi muito maior e cobriu toda a região, se tratando dos instantes de tempo que o evento extremo aconteceu.

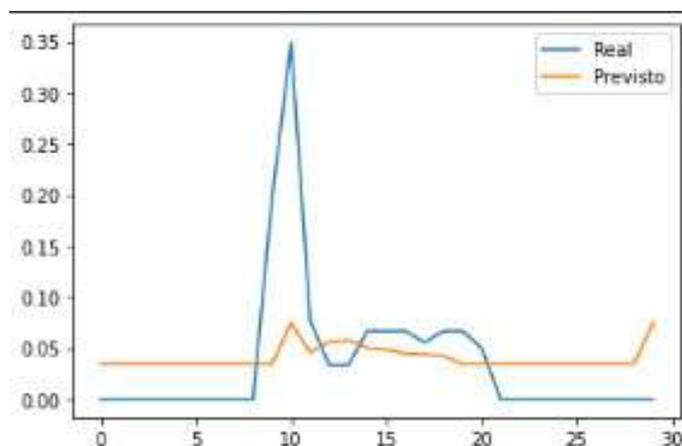


Figura 4. Previsão vs Valor Real em uma região do espaço

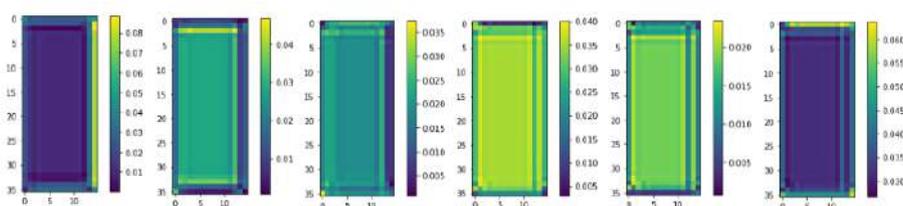


Figura 5. Distribuição espacial do erro em múltiplas janelas temporais

Fontes: Elaborada pelo autor.

5. Conclusão

Neste trabalho, estudamos a utilização de redes neurais convolucionais na predição de eventos extremos. De maneira a preparar o dado para treinamento, uma série de atividades de pré-processamento foram executadas, incluindo o desenvolvimento de um algoritmo de interpolação espacial para dados esparsos, de forma a gerar uma estrutura em grade regular necessária ao treinamento do modelo. De forma a aprender as relações espaço-temporais das séries temporais capturadas pelos pluviômetros, foi utilizada uma arquitetura de redes *ConvLSTM*. Contudo, avaliamos que para obter boas métricas e resultados é preciso a utilização de um modelo multimodal com a inclusão de novos atributos de outras fontes de dados.

A partir disso, é possível concluir que um atributo não é suficiente para obtenção de uma boa predição, assim como uma maior quantidade de dados, uma vez que o experimento foi feito em uma pequena localidade da região serrana, onde a distribuição geográfica é composta por um território muito irregular com inúmeras montanhas formando um vale, não conseguindo ter a difusão do evento, ficando concentrado apenas naquela região. Isso mostra que inúmeras características e fontes de dados vão estar influenciando para conseguir lidar de forma ideal com os dados e eventos extremos, muito localizado.

Em trabalhos futuros, pode-se realizar a combinação de várias fontes de dados e maior estudo focalizado na região onde ocorreu o evento, com o objetivo de extrair o máximo de informações possíveis para estudo do problema, realizando assim, uma análise e auxílio na base de novos trabalhos.

Referências

- Leticia Braga Berlandi e Analice Costacurta Brandi (2022). Comparação entre métodos numéricos computacionais na solução de um problema de valor inicial. Disponível em: <https://www.fc.unesp.br/Home/Departamentos/Matematica/revistacqd2228/v07a01-comparacao-entre-metodos-numericos.pdf>. Acesso em: 22 de junho 2022.
- Nathan Lopes - UOL Notícias (2022). Notícia: Choveu mais que o esperado para fevereiro inteiro em 3 horas em petrópolis. Disponível em: <https://noticias.uol.com.br/cotidiano/ultimas-noticias/2022/02/16/chuva-petropolis-meteorologia.htm>. Acesso em: 18 de junho 2022.
- Prefeitura do Rio de Janeiro (2022). Centro de operações do rio. Disponível em: <http://cor.rio>. Acesso em: 13 de junho 2022.
- Souto, Y. M., Porto, F., Moura, A. M., and Bezerra, E. (2018). A spatiotemporal ensemble approach to rainfall forecasting. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.
- The HDF Group. Arquivo hierárquico formato de dados 5. Disponível em: <https://ficheiros.com.br/extensao/h5/>. Acesso em: 20 de junho 2022.
- Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatio-temporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Relatório de Atividades

1. Dados gerais

Título do projeto: Projeto e Implementação de *Workflows* Científicos Reprodutíveis de Alto Desempenho

Bolsista: Lucas da Cruz Silva

Orientador: Luiz Manoel Rocha Gadelha Júnior

Coorientador: Carla Osthoff e Kary Ocaña

Tipo de bolsa: Iniciação Científica

Período do relatório: 01/12/2019 - 30/07/2022

2. Objetivos

O presente trabalho se desenvolve a partir de um projeto multidisciplinar que envolve o Laboratório de Bioinformática (LABINFO) e o Centro Nacional de Processamento de Alto Desempenho (CENAPAD) do Laboratório Nacional de Computação Científica (LNCC), com o objetivo de acoplar ao Bioinfo-Portal, um portal de Bioinformática hospedado no LNCC, um *workflow* científico de sequenciamento RNA (RNA-Seq) de Alto Desempenho. Dessa forma, através da motivação de facilitar o gerenciamento (definição, execução e monitoração) de atividades extensivas, complexas e computacionalmente custosas, este projeto tem por objetivo geral, modelar, executar e analisar *workflows* científicos com tecnologias e ambientes de Processamento de Alto Desempenho (PAD), tomando como caso de estudo o experimento de bioinformática de sequenciamento RNA.

3. Introdução

Em diversas áreas da ciência existem experimentos que podem ser computacionalmente intensivos, como é o caso dos experimentos de bioinformática de sequenciamento RNA. A técnica de RNA-Seq trata da análise de Expressão Diferencial de Genes (EDG) e sequenciamento genômico, ou seja, ela permite estudar o comportamento de um conjunto de transcritos de uma célula em uma dada condição fisiológica ou de desenvolvimento, tal como o câncer. A tecnologia de RNA-Seq demonstra um grande avanço nos estudos da transcritômica, no entanto, existem ainda muitos desafios relacionados à complexidade, produção e consumo de grande volume de dados e uso de programas e *software* de alto custos computacionais. Tipicamente, podem levar muitas horas ou mesmo dias para que um estudo de caso de RNA-Seq seja processado. Sob essa condição, diversas áreas da ciência têm se apoiado a procedimentos e uso de infraestruturas computacionais no decorrer dos anos [Cordeiro et al. 2013].

Computação de Alto Desempenho (CAD), ciência de dados e aprendizado de máquinas são exemplos de áreas da computação que se aliam a bioinformática visando desenvolver estratégias para prover um baixo tempo e custo computacional. Outra tecnologia que cabe citar são os *workflows* científicos, que desenvolvem uma forma abstrata de representar esses experimentos, modelando, computacionalmente, e encadeando as etapas experimentais, isso permite que a gerência de execução e análise sejam mais bem estabelecidas. E isso leva a reforçar a reprodutibilidade, confiabilidade e escalabilidade do experimento. Para atingir esses três requisitos de experimentos científicos, é necessário utilizar ferramentas que forneçam a infraestrutura para definir, executar e monitorar o *workflow*, permitindo que toda a modelagem abstrata desenvolvida pelo cientista seja traduzida a nível de máquina.

Sistema de Gerência de *Workflows* Científicos (SGWfC) ou mesmo determinadas linguagens de programação são capazes de orquestrar a execução de *workflows* e até acoplar *workflows* computacionalmente intensivos em ambientes de CAD. Nesse âmbito existem SGWfC baseados em sistema *web*, como o Galaxy, e pacotes estatísticos do R e Bioconductor, como DESeq2 e EdgeR, que são usados nos estudos de EDG e para automação das tarefas os SGWfC distribuídos e paralelos Kepler, Pegasus e as linguagens Swift e Python/Parsl.

O desenvolvimento deste trabalho é uma continuação da proposta de [Cruz et al. 2021] para a otimização de desempenho do *workflow* desenvolvido, chamado ParslRNA-Seq. O que levou as execuções a alcançarem um ganho em tempo computacional maior do que 65% em relação a versão

apresentada em [Cruz et al. 2020], com três atividades. A grande proposta passa pela remodelagem do *workflow*, onde ele passaria de três para seis atividades. Computacionalmente, o esperado é que o tempo de processamento seja maior dado a adição de novas atividades ao *workflow*. No entanto, elas foram adicionadas de forma estratégica sugerindo o uso da técnica de divisão e conquista: com o particionamento do dado, processamento e combinação de resultados. Da versão do ParsIRNA-Seq apresentada em [Cruz et al. 2020] até a versão de [Cruz et al. 2021] o *workflow* saiu de um Tempo Total de Execução (TTE) de cerca de 3 dias e foi para cerca de 24 minutos dentro de um ambiente distribuído de alto desempenho. É fato que o gargalo computacional sempre irá existir. Mas, falta uma análise que sugere uma melhora ainda mais significativa no TTE do *workflow*: a de Entrada e Saída (E/S). Portanto, o presente trabalho, apresenta uma análise das operações de E/S das atividades mais custosas do ParsIRNA-Seq, uma análise comparativa acerca do ganho computacional da execução do *workflow* científico ParsIRNA-Seq no Hard Disk Drive (HDD), ou seja, no Lustre, e no Solid State Drive (SSD), além de uma nova proposta na modelagem do *workflow* para o seu processamento em ambientes distribuídos de alto desempenho. As estimativas sugerem um ganho de tempo de processamento para cada pipeline de tarefas do *workflow* em cerca de 2 minutos, o que para um ambiente paralelo e distribuído já é bastante significativo.

Este relatório está organizado da seguinte forma: a Seção 1, apresenta informações gerais sobre o relatório; a Seção 2 traz o resumo e objetivo geral do projeto de pesquisa; na Seção 3 a presente introdução, bem como um breve resumo do que será apresentado; a Seção 4 traz a metodologia adotada na modelagem do *workflow* de RNA-Seq desenvolvido e sua descrição; na Seção 5 são apresentados os resultados e análises experimentais; e, por fim, a Seção 6 traz as considerações finais e os trabalhos futuros.

4. Metodologia

4.1. Modelagem do *workflow* de RNA-Seq

Para modelar computacionalmente um *workflow* científico, inicialmente, é requerido que as atividades ou etapas do experimento sejam bem mapeadas, a fim de fazer uma abstração de todos os procedimentos que o cientista realizaria sem a utilização da automatização. A modelagem requer o conhecimento de quantas atividades são realizadas, qual a ordem dessas atividades e as dependências entre elas. O *workflow* de RNA-Seq inicialmente desenvolvido em [Cruz et al. 2020], denominado ParsIRNA-Seq é composto por três atividades principais: Bowtie, HTSeq e DESeq (Figura 1(a)). Na atual versão do ParsIRNA-Seq, o *workflow* passa a ser composto por seis atividades (Figura 1(b)), as quais foram adicionadas Sort, Split_Picard e Merge_HTSeq [Cruz et al. 2021]. Todas incluídas em função da melhora de desempenho da atividade HTSeq.

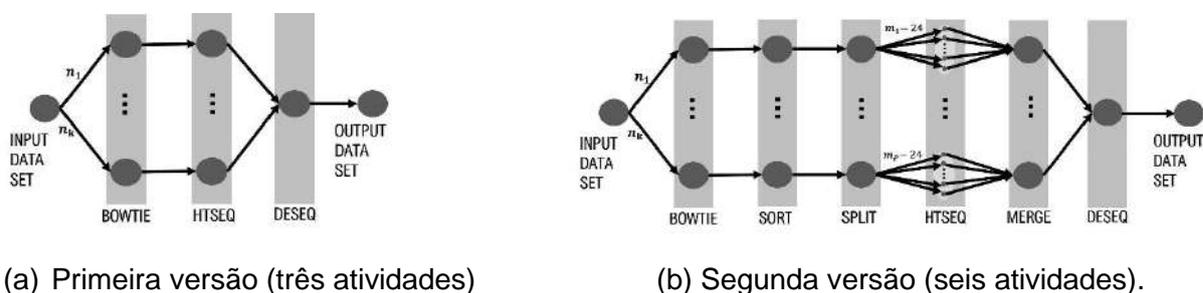


Figura 1. Modelagem conceitual do *workflow* científico de RNA-Seq.

As atividades são executadas na seguinte sequência: A atividade 1, Bowtie, faz o mapeamento das leituras curtas do genoma [Langmead and Salzberg 2012]. A atividade 2, Sort, faz a ordenação das leituras. A atividade 3, Split, faz a divisão dos arquivos de leituras, dividindo cada arquivo de amostra em várias subpartes. A atividade 4, HTSeq, faz a contagem das leituras mapeadas de cada gene [Anders et al. 2014], nesse caso, as contagens serão realizadas em relação a cada subparte da amostra. A atividade

5, Merge, vai combinar todas as subpartes da amostra em uma única; por fim, a atividade 6, DESeq, aplica estatísticas em cima das contagens para a análise da EDG [Love et al. 2014].

A nova proposta para a modelagem do *workflow* se refere a uma desacoplagem e as formas de execução. Que surge com base no fato de que todo usuário do supercomputador Santos Dumont (SDumont) que escala um *job* na fila do Slurm (*Simple Linux Utility for Resource Management*), em geral, usa o Sistema de arquivos paralelos (Lustre) para a escrita e leitura de dados. No entanto, os nós do SDumont são providos, cada um, de um SSD e sabe-se que as operações de escrita e leitura nesses dispositivos chegam a ser cerca de 10 vezes mais rápidas em relação ao HDD (*Hard Drive Disk*), nesse caso o Lustre. Ainda sobre a forma de execução, segundo [Cruz et al. 2021], os melhores desempenhos do ParsIRNA-Seq são alcançados quando para n dados, são alocados, de forma única, n nós para o *job*. Ou seja, um bloco de n nós. Assim, cada pipeline pode ser processada em um nó computacional. No entanto, a alocação única de n nós, pode aumentar o *throughput* da rede o que pode fazer reduzir o TTE do *workflow*. Portanto, quanto a execução, a nova proposta para processamento do *workflow* sugerida no presente trabalho é que para cada dado seja alocado um nó para seu processamento. Assim, temos um *throughput* menor. Além disso, é sugerida também a utilização do SSD munido da seguinte estratégia: os dados de entrada são copiados do Lustre para o SSD e o processamento do *workflow* se segue no SSD até a última atividade (merge), que escreve sua saída diretamente no Lustre. Essa saída é utilizada como dado de entrada pela atividade *deseq*.

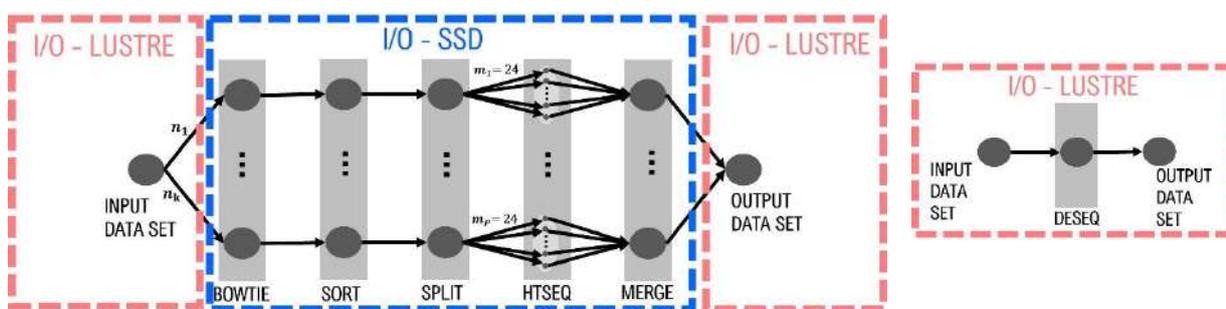


Figura 2. Nova proposta para processamento do ParsIRNA-Seq.

A desacoplagem sugerida, é referida acerca da atividade *deseq*, que necessita de uma sincronização de dados, pois irá utilizar todos os dados de entrada de uma só vez e, sendo de baixo custo computacional, só irá utilizar um único nó computacional. Ou seja, ainda que uma pipeline de tarefas termine essa atividade (*deseq*) só irá inicializar sua execução depois que a última pipeline de tarefas for processada. Isso quer dizer que ela irá impedir a liberação de outros $n - 1$ nós durante algum tempo. Esse fato dentro de um ambiente distribuído é ruim porque um usuário teria uma alocação com n nós, onde $n - 1$ estariam ociosos. O que poderia atrasar a inicialização do *job* de outro usuário. Em ambientes de nuvem, onde o usuário paga pelo tempo de utilização seria de fato um prejuízo. A solução então é desacoplar essa atividade do *workflow* e escalar a dependência da outra parte do *workflow*. O TTE será então definido em duas partes, como apresentado na Figura 2: tempo da primeira parte, definido pelas atividades *bowtie*, *sort*, *split*, *htseq* e *merge*; e, tempo da segunda parte, definido pela atividade *deseq*.

Toda a modelagem computacional do ParsIRNA-Seq foi feita utilizando a biblioteca Parsl, da linguagem de programação Python, que é capaz de agir como um gerenciador de atividades dentro dos mais variados ambientes computacionais (*notebooks*, nuvem, *clusters*, supercomputadores etc.) [Babuji et al. 2019]. Além disso, é capaz de abstrair ao usuário toda a complexidade envolvida para inicialização e processamento de *workflows* dentro desses ambientes, se fazendo dessa forma uma grande ferramenta a cientistas das mais diversas áreas.

4.2. Configuração de ambiente computacional

O ambiente computacional utilizado para as execuções e as análises de desempenho é o supercomputador SDumont, que está entre as 500 máquinas mais poderosas do mundo. Ele possui uma capacidade de processamento de 5.1 Petaflop/s, com 34.688 CPU *multicores* distribuídas em 1.132 nós computacionais que são interligados por uma rede de interconexão *Infiniband FDR/HDR*. Os nós computacionais possuem duas CPUs Ivy Bridge Intel Xeon E5-2695v2 (12c @2.4GHz) e 64GB de memória RAM e uma GPU Nvidia K40. As execuções foram realizadas em nós computacional de duas CPUs Intel Xeon E5-2695v2 Ivy Bridge, 24 núcleos (12 por CPU), 64GB de memória RAM e um SSD de 128GB de memória.

4.3. Dados de entrada

Os dados de entrada pertencem a um experimento real de RNA-Seq, foram extraídos do repositório público *Gene Expression Omnibus* (GEO), são eles: o genoma de referência; o arquivo de características genômicas no formato GTF (*Gene Transfer Format*); os arquivos de sequenciamento no formato *fastq* e as condições experimentais da amostra. Esses dados de RNA-Seq se referem à via metabólica de sinalização transcricional *Wnt*, a que tem sido reportada com um papel regulatório no coração. Um total de seis arquivos de sequenciamento, divididos em dois grupos de três, controle e condição, sequenciadas no *Mus Musculus* (rato) foram usados como dados de entrada pelo ParsIRNA-Seq. Há de se notar que o número de arquivos de sequenciamento e a divisão dos grupos podem variar conforme o experimento. Então, inicialmente, a análise de desempenho está ligada somente a esses seis arquivos, onde dois deles possuem o tamanho de 1.8GB e os outros 1.9GB, 2.0GB, 2.5GB e 3.0GB, totalizando 13GB.

5. Resultados e discussão

As subseções estão organizadas da seguinte forma: a seção 5.1 apresenta uma análise das atividades mais custosas em E/S do *workflow*, *bowtie* e *sort*, utilizando o Lustre e SSD; Tendo observado algum ganho relevante em termos de tempo computacional, a seção 5.2 apresenta uma análise comparativa de execução do *workflow* da modelagem apresentada na Figura 2, usando o Lustre e o SSD.

5.1 Análise das operações de E/S no Lustre e SSD das atividades mais custosas do *workflow*

Atividade *bowtie*. Para a atividade *Bowtie* as operações de escrita são as que mais se destacam. Segundo o Darshan, utilizando o Lustre para as operações de E/S na atividade *Bowtie* o arquivo de maior tamanho leva cerca de 4 minutos e 23 segundos e cerca de 90% desse tempo é consumido em operações de escrita. Em média, a atividade escreveu 11 GB. Já para o arquivo de menor tamanho, o TTE é de 2 minutos e 32 segundos e cerca de 80% desse tempo e gasto para operações de escrita. Quando as execuções dessa atividade passam para o SSD para o arquivo de maior tamanho o TTE cai para cerca de 3 minutos e 47 segundos. Já para o arquivo de menor tamanho o TTE fica em torno de 2 minutos e 10 segundos.

Atividade *sort*. Para a atividade *Sort* as operações de escrita já não estão tão em destaque. Segundo o Darshan, utilizando o Lustre para as operações de E/S na atividade *Sort* o arquivo de maior tamanho, leva cerca de 1 minuto e 31 segundos e cerca de 70% desse tempo é consumido em operações de leitura e cerca de 10% em operações de escrita. Em média são lidos 13 GB e escritos 1.1 GB. Já o arquivo de menor tamanho, o TTE é de 41 segundos, cerca de 5% e consumido em operações de leitura e em torno de 15% em operações de escrita. Quando as execuções dessa atividade passam para o SSD para o arquivo de maior tamanho o TTE cai para cerca de 48 segundos. Já para o arquivo de menor tamanho o TTE fica em torno de 28 segundos.

5.2 Análise comparativa de execuções entre uso de SSD e Lustre

Os TTEs que serão apresentados na presente subseção dizem respeito a execuções completas do *workflow* feitas com o uso do SSD e uso do Lustre de modo a comparar a diferença no TTE do *workflow*. No experimento a seguir, foram realizadas cinco execuções, foi obtido então a média entre as essas execuções e o desvio padrão. Nesse experimento, foi utilizada a estratégia de desacoplagem para

impedir que os nós fiquem ociosos durante tempo demasiado, já que o TTE do *workflow* será sempre definido como o tempo que o maior arquivo leva para ser processado. Dessa forma, o TTE do *workflow* passa a ser o tempo de processamento das atividades bowtie, sort, split, htseq e merge (primeira parte do *workflow*) acrescidos do tempo de processamento da atividade deseque (segunda parte do *workflow*).

Aqui podemos ter como base o tempo do arquivo de maior tamanho, de 3 GB, que levará maior tempo para ser processado, e, portanto, o tempo dele determina o TTE do *workflow*. Pela Tabela 1, usando o Lustre, o tempo médio de execução da primeira parte do *workflow* é de cerca de 17,6 minutos. Já pela Tabela 2, usando o SSD, a execução da primeira parte do *workflow* dura em média cerca de 15,8 minutos. Na Tabela 3 é indicado o tempo médio de execução da atividade deseque, que é em torno de 1,4 minutos. Ou seja, usando o Lustre o *workflow* leva cerca de 19 minutos para finalizar a execução e usando o SSD ele leva cerca de 17 minutos.

É possível ainda notar, pelo tempo de execução do arquivo de menor tamanho, de 1.8GB, que através da alocação de um único nó a liberação de um nó computacional acontece cerca de 6 minutos mais cedo em relação ao arquivo de maior tamanho.

EXECUÇÕES NO LUSTRE							
TAMANHO	TEMPO TOTAL DE EXECUÇÃO (minutos)					MÉDIA	DESVIO PADRÃO
1.8 G	11,6167	12,0833	11,95	12,0667	11,5667	11,85668	0,2479175508
3.0 G	17,6	17,4	17,7667	17,7667	17,6333	17,63334	0,1509378117

Tabela 1. Tempo de execução da primeira parte do *workflow* no Lustre.

EXECUÇÕES NO SSD							
TAMANHO	TEMPO TOTAL DE EXECUÇÃO (minutos)					MÉDIA	DESVIO PADRÃO
1.8 G	10,0661	10,0328	10,0817	10,1988	10,0816	10,0922	0,06284055219
3.0 G	15,8649	15,8978	15,5828	15,6994	16,1159	15,83216	0,2035279661

Tabela 2. Tempo de execução da primeira parte do *workflow* no SSD.

EXECUÇÕES NO LUSTRE (ATIVIDADE DESEQ)						
TEMPOS DE EXECUÇÃO (minutos)					MÉDIA	DESVIO PADRÃO
1,4	1,5167	1,5333	1,4333	1,3333	1,475	0,09168182299

Tabela 3. Tempo da segunda parte do *workflow* (atividade deseque).

6. Conclusão

No presente trabalho, diante da proposta de remodelagem do *workflow*, foram apresentadas as atividades que dentro do *workflow* ParsIRNA-Seq se apresentam como mais custosas em termos de E/S: bowtie e sort. E, comparativamente, vimos uma diferença, ainda que pequena, no tempo de computação para essas duas atividades usando o Lustre e o SSD. Essas atividades tiveram um ganho em cerca de 13 a 36 segundos na execução pela utilização do SSD. Tendo ganho algum tempo em computação, as análises seguiram para as execuções do *workflow*, seguindo a nova proposta apresentada na seção 4 (Figura 2). Infelizmente, não é possível fazer uma comparação de forma justa entre o formato de execução utilizada por [Cruz et al. 2021] e o presente trabalho, pois são diferentes. No entanto, é possível comparar o tempo de computação que de fato é o interesse deste trabalho e a grande importância dentro da área de CAD, redução de tempo computacional. Podemos notar que alocações de nós em bloco único para o ParsIRNA-Seq não compensa, pois há uma ociosidade dos nós durante tempo excessivo, de

modo que outro usuário poderia estar utilizando aquele recurso computacional. Com a alocação equivalente de um nó para um dado e com a desacoplagem entre a primeira parte e segunda parte do *workflow* o tempo em que um nó ficaria ocioso é descartado e temos também um refinamento para o público que utiliza serviços de nuvem. Além disso, há uma melhora no throughput do *workflow*. Como observado, em tempo, o *workflow* sai de uma execução de cerca de 24 minutos, apresentada por [Cruz et al. 2021], para cerca de 19 minutos apresentado no presente trabalho.

Para utilização do supercomputador SDumont, isso significa que para a execução do *workflow* ParsIRNA-Seq não é mais necessário utilizar filas de tempo longo, no caso da modelagem de [Cruz et al. 2020], ou filas de tempo médio, como é o caso da modelagem de [Cruz et al. 2021]. Mas, hoje, com o presente trabalho é possível utilizar filas rápidas, de 20 minutos, para realizar experimentos de RNA-Seq. Mas ainda há mais para avançar, como trabalhos futuros temos que: serão feitas otimizações nessa nova modelagem do ParsI-RNA-Seq, eliminando perdas desnecessárias de processamento; outro fator a ser estudado é sobre a utilização do SSD, pois talvez não tenha sido feita do modo ideal, que seria a atividade bowtie ler direto do Lustre e escrever sua saída no SSD, assim os tempos processando a cópia seria descartado. Mas, ainda é um estudo em andamento; e, por fim, também serão realizados estudos para executar múltiplos experimentos de RNA-Seq em paralelo no SDumont. O ParsRNA-Seq se encontra disponível no GitHub¹ para toda a comunidade científica e está sendo acoplado ao Bioinfo-Portal, um portal hospedado no LNCC, voltado ao fortalecimento das pesquisas envolvendo o uso da bioinformática.

Como apresentado, no desenvolvimento do projeto foram alcançados resultados mais do que satisfatórios o que inclui geração de códigos dos *workflows* disponíveis para a comunidade científica de bioinformática, apresentação oral nos principais fóruns em diversas comunidades de CAD, Ciência de Dados, Bioinformática e e-Ciência, publicação de trabalhos completos e resumos nos principais eventos brasileiros, publicação de pôster em evento internacional, menção honrosa em evento de CAD. Além disso, este projeto está integrado em colaboração a grupos de pesquisa internacionais (INRIA) e projetos multi-institucionais (UNB, Fiocruz, UFRJ), o que demonstra a importância do presente projeto de Iniciação Científica em termos científicos e tecnológicos.

7. Referências bibliográficas

- Anders, S., Pyl, P. T., and Huber, W. (2014). HTSeq—a Python *framework to work with high-throughput sequencing data*. *Bioinformatics*, 31(2):166–169.
- Babuji, Y., Woodard, A., Li, Z., Katz, D. S., Clifford, B., Kumar, R., Lacinski, L., Chard, R., Wozniak, J., Foster, I., Wilde, M., and Chard, K. (2019). Parsl: *Pervasive parallel programming in python*. *In 28th ACM International Symposium on High-Performance Parallel and Distributed Computing (HPDC)*
- Cordeiro, D., Braghetto, K. R., Goldman, A., and Kon, F. (2013). Da ciência à e-ciência: paradigmas da descoberta do conhecimento. *Revista USP*, (97):71–81
- Cruz, L., Coelho, M., Gadelha, L., Ocaña, K., and Osthoff, C. (2020). Avaliação de desempenho de um *workflow* científico para experimentos de rna-seq no supercomputador santos dumont. *In Anais Estendidos do XXI Simpósio em Sistemas Computacionais de Alto Desempenho*, pages 86–93, Porto Alegre, RS, Brasil. SBC.
- Cruz, L., Coelho, M., Terra, R., Carvalho, D., Gadelha, L., Osthoff, C., and Ocaña, K. (2021). *Workflows científicos de rna-seq em ambientes distribuídos de alto desempenho: Otimização de desempenho e análises de dados de expressão diferencial de genes*. *In Anais do XV Brazilian e-Science Workshop*, pages 57–64, Porto Alegre, RS, Brasil. SBC.
- Langmead, B. and Salzberg, S. L. (2012). *Fast gapped-read alignment with bowtie 2*. *Nature methods*, 9(4):357.
- Love, M. I., Huber, W., and Anders, S. (2014). *Moderated estimation offold change and dispersion for rna-seq data with deseq2*. *Genome biology*, 15(12):550.

¹ <https://github.com/lucruzz/RNA-seq>

Modelos da anatomia arterial do córtex cerebral gerados por algoritmos de vascularização automáticos

Bolsista Luiza Carneiro Pizzi

Orientador: Pablo Javier Blanco

Bolsa: Iniciação Científica

Ano: 2022

1) Objetivos

O objetivo geral deste projeto é pesquisar sobre a utilização de algoritmos de vascularização automática visando seu domínio e aplicação na vascularização de domínios anatômicos correspondentes a territórios do cérebro humano.

Os objetivos particulares deste projeto são enumerados a seguir:

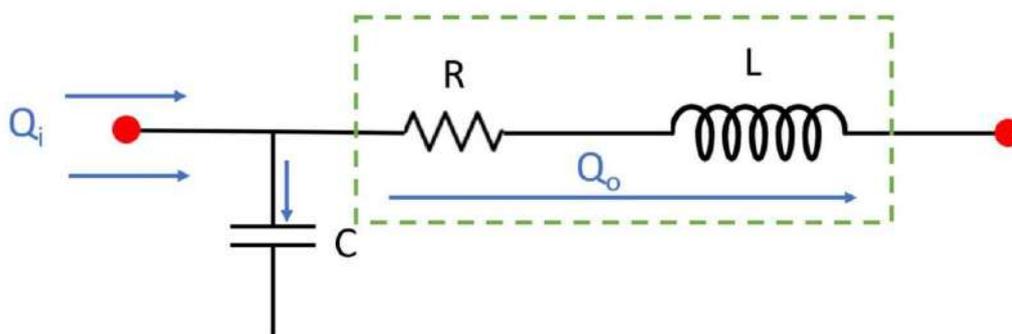
1. Pesquisar sobre os algoritmos de vascularização automática desenvolvidos no grupo de pesquisa HeMoLab.
2. Realizar uma introdução à pesquisa na modelagem do sistema cardiovascular humano, com ênfase na modelagem e simulação em hemodinâmica computacional.
3. Pesquisar acerca da utilização dos algoritmos de vascularização automática.
4. Pesquisar acerca da utilização dos algoritmos de vascularização automática em domínios anatômicos simples, complexos e de grande porte.
5. Estudo do circuito RLC, em pequenos e grandes vasos.

3) Introdução

Para iniciar os estudos dos circuitos RLC, temos a equação Geral da queda de vazão do vaso que será importante para o desenvolvimento do sistema. Ao adicionar um capacitor, conseguimos obter o armazenamento de massa, de acordo com a figura 1.

Figura 1: Introdução ao circuito RLC.

Circuito RLC



$$C \frac{\delta P}{\delta t} = Q_i - Q_o$$

Fonte: A autora, 2022.

A seguir temos as equações utilizadas para cada um dos seguimentos, sendo duas equações de conservação de massa e uma de queda de pressão:

$$P_{PROX} - P_{MED} = \frac{R}{2} Q_i + \delta \frac{Q_i}{\delta t}$$

$$P_{MED} - P_{DIST} = \frac{R}{2} Q_0 + \delta \frac{Q_0}{\delta t}$$

$$C \frac{R}{2} = Q_i - Q_0$$

Ao analisar os conceitos, podemos levar em consideração o módulo de Young, associados a arteríolas, quanto mais rígido, menos deformável.

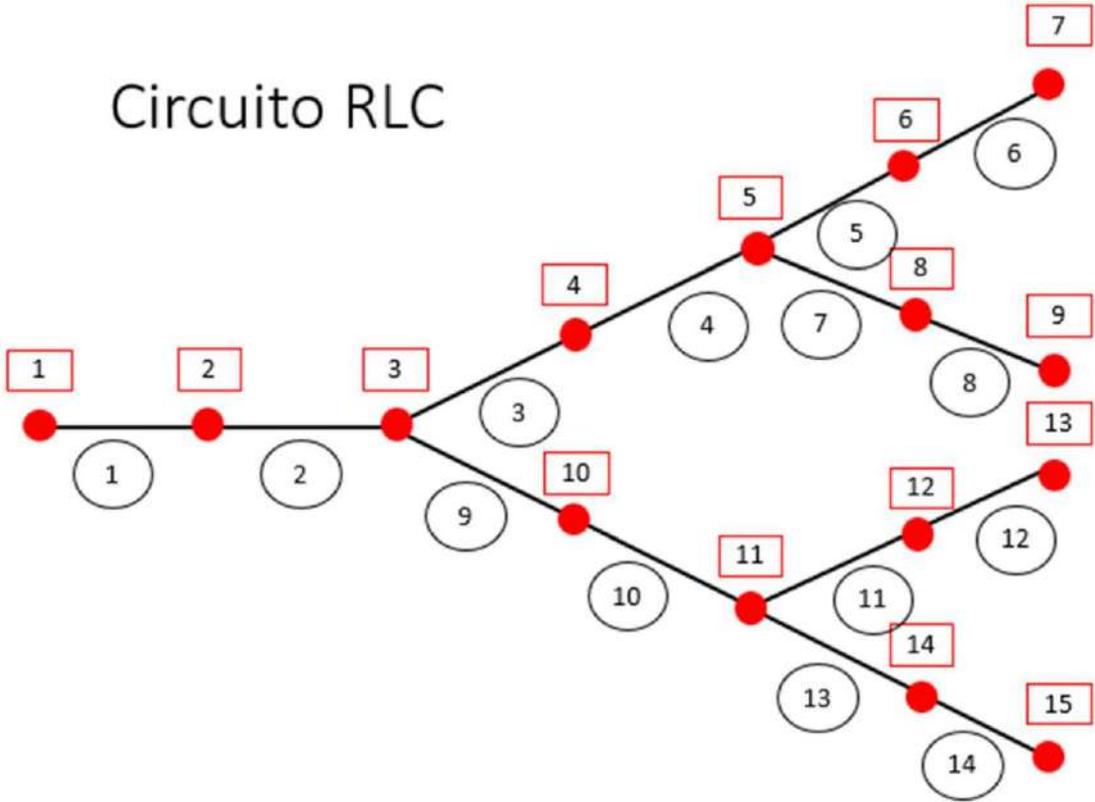
4) Material e Métodos ou Metodologia

O material utilizado para a execução da pesquisa foi pesquisas dos trabalhos desenvolvidos pelo time HeMoLab, aulas e explicações do orientador Pablo Javier.

5) Resultados e Discussão

Podemos associar um circuito RLC ao sistema de acordo com a figura 2:

Figura 2: Representação de um circuito RLC.



A estrutura de dados de entrada pode ser representada por 4 diferentes tipos de elementos, conforme figura 3, e podem ser descritos da seguinte maneira:

Cells: Número que cada um dos nós que compõem um vaso.

Terminals: Representa o número do terminal, seguido por P representando pressões e valor da pressão.

Pipes: Representa o diâmetro e a largura de cada um dos vasos.

Junctions: Contém o número do nó que contém bifurcação e dos nós de entrada e saída de cada um dos vasos.

Figura 3: Exemplo de dados de entrada diferenciados por tipo.

Dados de Entrada

• Cells

1	1, 2, 3
2	3, 4, 5
3	5, 6, 7
4	5, 8, 9
5	3, 10, 11
6	11, 12, 13
7	11, 14, 15
8	

• Terminals

1	1,P, 10, 2
2	7,P, 5, 2
3	9,P, 5, 1
4	13,P, 5, 1
5	15,P, 5, 5

• Pipes

1	D,L
2	2,1
3	8,2
4	4,3
5	1,4
6	2,7
7	3,9
8	4,6

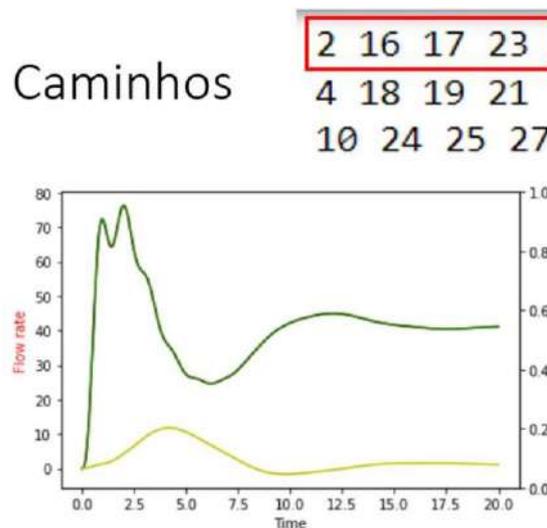
• Junctions

1	3, 2, -3, -9
2	5, 4, -5, -7
3	11, 10, -11, -13

Fonte: A autora, 2022.

Com os dados de entrada acima, conseguimos chegar nos seguintes resultados, plotando o caminho (2, 16, 17, 23), temos o resultado mostrado na figura 4:

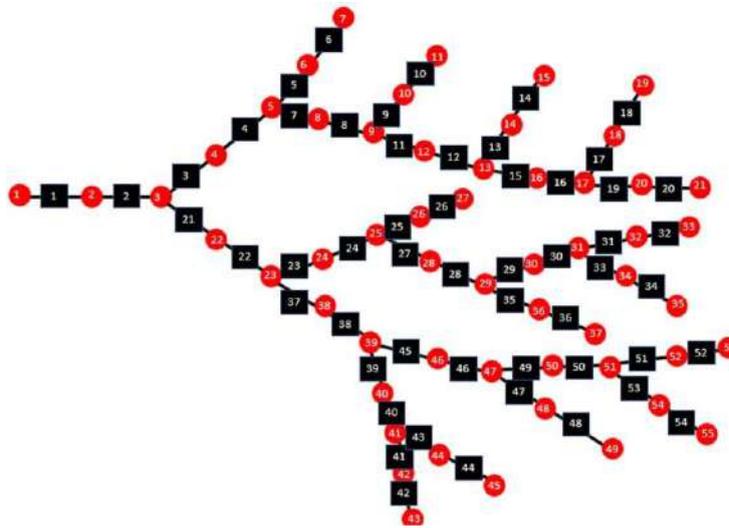
Figura 4: Gráfico de quociente de vazão vs tempo.



Fonte: A autora, 2022.

É possível trabalhar em uma rede maior como no exemplo da figura 5:

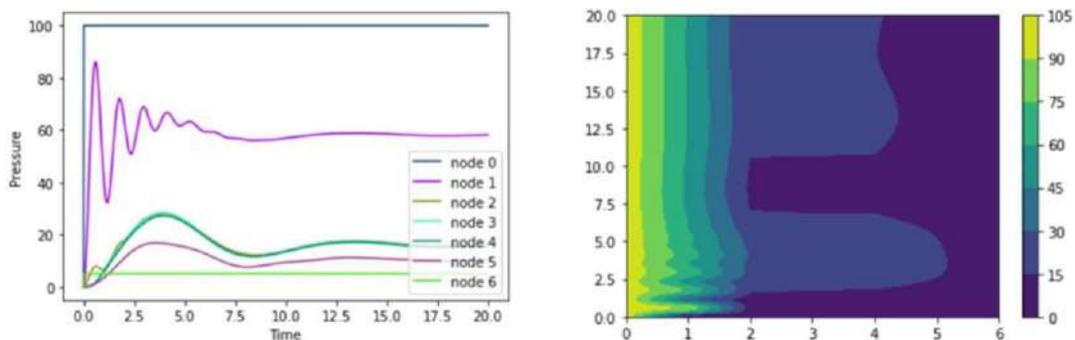
Figura 5: Representação de uma árvore de maior porte.



Fonte: A autora, 2022.

Plotando o caminho (1,2,3,4,5,6,7) temos o seguinte gráfico de pressão vs tempo na figura 6:

Figura 6: Representação da pressão em diferentes nós.



Fonte: A autora, 2022.

6) Conclusões

Após os resultados apresentados no capítulo anterior, pode-se concluir que a metodologia utilizada é capaz de apresentar dados factíveis e realizar a simulação dos parâmetros de escoamento sanguíneo. Após a definição dos dados de entrada, conseguimos adaptar a quantidade de vasos apresentadas como exemplo na rede menor e rede maior. Conseguimos ver a associação do fluxo sanguíneo no circuito RLC, e associar ao circuito sanguíneo humano de acordo com algumas características apresentadas.

7) Referências bibliográficas

1. Blanco, P., Queiroz, R. & Feijóo, R. A computational approach to generate concurrent arterial networks in vascular territories. *Int. J. for Numer. Methods Biomed. Eng.* 29, 601–614 (2013).
2. Maso Talou, G.D. et al. Adaptive constrained constructive optimisation for complex vascularisation processes. *Scientific Reports*, V. 11, p. 6180 (2021).
3. Cury, L.F.M. et al. Parallel generation of extensive vascular networks with application to an archetypal human kidney model, Under review (2021).

Modelagem e Integração de Bancos de Dados Relacionais na Arquitetura do Bioinfo-Portal

Nome: Marco Antonio Silva Cabral

Orientador: DSc. Kary Ann del Carmen Ocaña Gautherot

Coorientador: DSc. Antonio Tadeu Azevedo Gomes

Coorientador: Marcelo Monteiro Galheigo

Instituição de estágio: Laboratório Nacional de Computação Científica (LNCC), Laboratório de Bioinformática (LABINFO), Sistema Nacional de Processamento de Alto Desempenho (SINAPAD).

Bolsa: PIBIC - Programa Institucional de Bolsas de Iniciação Científica

Período: 01/01/2022 - 04/07/22

O Projeto de Iniciação Científica (IC) se desenvolve no nível de estágio supervisionado sob coordenação da orientadora Kary Ocaña e coorientadores Antônio Tadeu Gomes e Marcelo Galheigo do Laboratório Nacional de Computação Científica, com bolsa de IC financiada pelo CNPq. Um objetivo importante da Rede Nacional de Bioinformática (RNBio) em conjunto ao LNCC é o desenvolvimento de ferramentas e arquiteturas para a bioinformática [1]. Fruto dessas pesquisas em colaboração, foi desenvolvido o *gateway* Bioinfo-Portal (<https://bioinfo.lncc.br/>) que visa a execução de aplicações de bioinformática em larga escala, no apoio às pesquisas da comunidade científica de bioinformática [2-4] Bioinfo-Portal está hospedado e gerenciado pelo LNCC e usa recursos e tecnologias de computação de alto desempenho, como o supercomputador Santos Dumont (SDumont, <https://sdumont.lncc.br/>) a fim de diminuir o grande tempo de processamento das execuções. Bioinfo-Portal gerencia a execução automática de aplicações e dados científicos através de uma interface *Web* amigável e iterativa e das diversas camadas de *software* do *gateway*. Bioinfo-Portal utiliza, via serviços *Web* RESTful, o *middleware* CSGrid como *framework* de integração à arquitetura do SINAPAD. Atualizações e otimizações do Bioinfo-Portal na camada de banco de dados e de gerência de execuções irão fornecer uma melhor funcionalidade e escalabilidade de processos de execuções e armazenamento de dados de proveniência, tal que auxiliem na tomada de decisões inteligentes no uso de recursos computacionais [5,6].

OBJETIVOS

- Atualização das camadas de banco de dados e de gerência de execuções das aplicações de bioinformática em ambientes de CAD
- Desenvolvimento de serviços específicos para integrar informações científicas e dados de proveniência das diversas camadas da arquitetura do Bioinfo-Portal.
- Desenvolvimento de sistemas para criar inteligência em análise de coleta de dados do Bioinfo-Portal, melhorando o *gateway* em termos de velocidade, execução e armazenamento.

METODOLOGIA

Na primeira etapa, foi utilizado o PostgreSQL v10 como Sistema de Gerência de Banco de Dados (SGBD) relacional *Open Source* e o pgAdmin v5.2 como plataforma de gerência. O processo de conexão e implementação do banco de dados foi realizado via SSH a partir da liberação VPN do LNCC [7,8].

A segunda etapa envolve a utilização de serviços RESTful para o desenvolvimento de sistemas de inteligência na coleta de dados do *gateway*, sistemas de autenticação de usuário e mapeamento de dados das aplicações presentes na arquitetura do Bioinfo-Portal. A linguagem de programação utilizada é a PHP (*Hypertext Preprocessor*) e o *Visual Studio Code* como o editor de código-fonte usado para o desenvolvimento dos sistemas.

RESULTADOS E DISCUSSÃO

Baseado no referencial bibliográfico sobre gerência e modelos de banco de dados e na arquitetura do Bioinfo-Portal e do SINAPAD, foi implementado o modelo conceitual de banco de dados do Bioinfo-Portal. Iniciou-se a etapa de mapeamento dos dados da arquitetura do *gateway* para a implementação do modelo lógico. A Figura 1 apresenta a estrutura do banco de dados atualizada do Bioinfo-Portal que visa melhorar o armazenamento de dados de proveniência.

Estudo de Caso: Mapeamento do modelo conceitual ER

Atualmente, o banco de dados possui 15 entidades, *Files* e *Executions* são as únicas são as entidades originais Bioinfo-Portal (Figura 1). Todas as demais entidades da nova versão desenvolvido no presente projeto de IC do banco de dados estão disponibilizadas no servidor de desenvolvimento no SINAPAD. No período de desenvolvimento deste relatório, foram desenvolvidas as entidades *Gateway*,

Institution e *Scheduler* (mostrado na Figura 1) para o melhor armazenamento de dados provenientes da estrutura do *gateway* que possuem esses tipos de informações. A partir do acesso aprovado ao *VPN* do LNCC, foi possível acessar a máquina onde o PostgreSQL está instalado via *SSH* e implementar o modelo de banco de dados (mostrado na Figura 1) do Bioinfo-Portal.

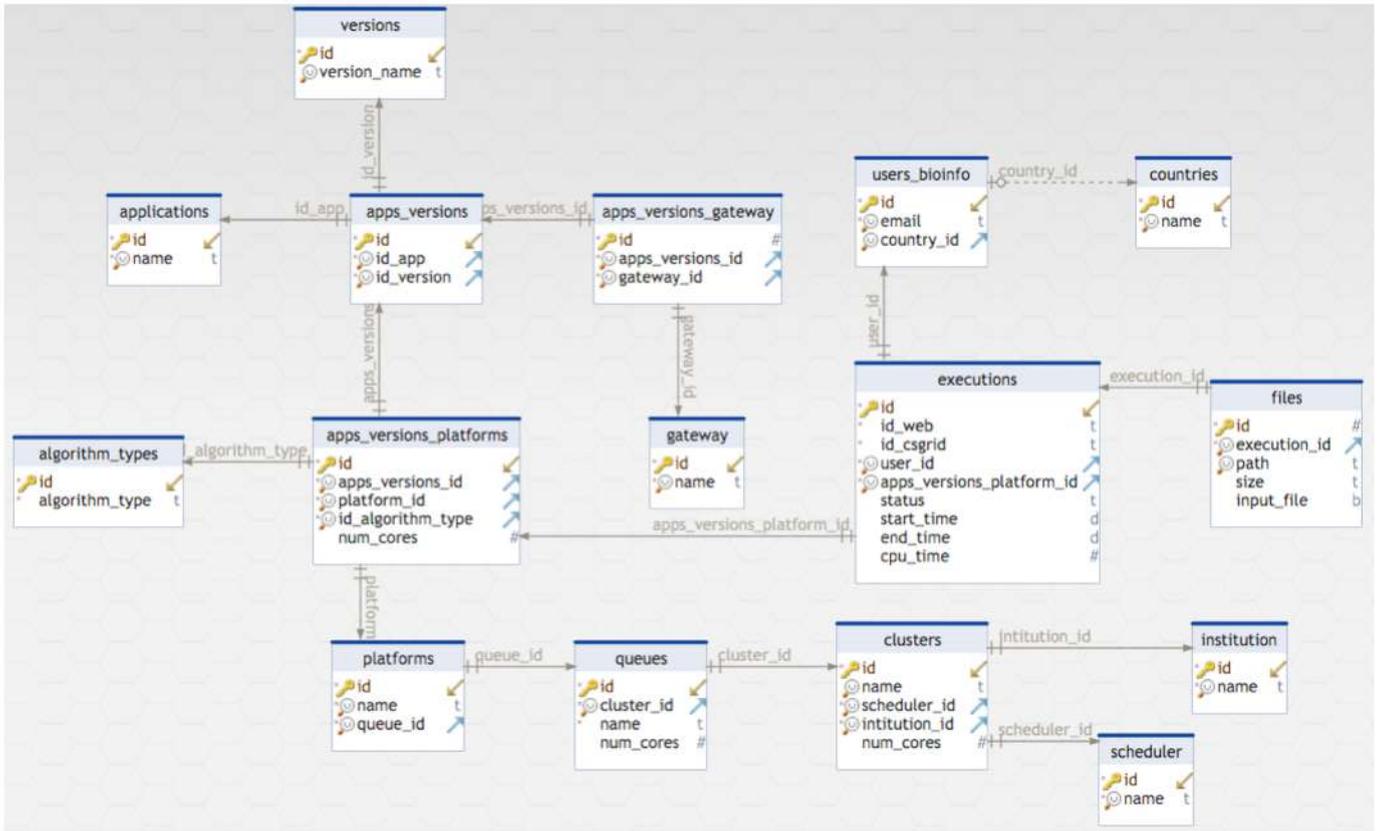


Figura 1. Diagrama Entidade-Relacionamento do Banco de Dados Bioinfo.

Após a implementação da atualização do banco de dados do Bioinfo-Portal, foi realizado o desenvolvimento de sistemas utilizando serviços *Web* RESTful, que interage dinamicamente com o *middleware* CSGrid do SINAPAD.

O primeiro sistema desenvolvido (Algoritmo 1) é o de autenticação, podendo tanto ser um método LDAP (Algoritmo 1A) quanto *RSA* (Algoritmo 1B), armazenando dados de proveniência de usuários como nome e a identificação e servindo como base para futuros desenvolvimentos de outros sistemas.

```

$url= ('http://fenix.***.lncc.br:8080/rest/op***');
$data = array(
'username'=> '****',
'password'=> '*****',
'service'=> '****',
'uuid' => $uuid );

```

```

$headers = array(
'Accept: application/json');
$handle = curl_init();
curl_setopt($handle, CURLOPT_URL, $url);
curl_setopt($handle, CURLOPT_HTTPHEADER, $headers);
curl_setopt($handle, CURLOPT_RETURNTRANSFER, true);
curl_setopt($handle, CURLOPT_SSL_VERIFYHOST, false);
curl_setopt($handle, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($handle, CURLOPT_POST, true);
curl_setopt($handle, CURLOPT_POSTFIELDS, http_build_query($data));
$response = curl_exec($handle);
$obj = json_decode ( $response );
$b = $obj->{'uuid'};

```

Algoritmo 1A: Sistema de autenticação LDAP.

```

$url ="http://fenix.*****.br:8080/rest/op/*****";
    $data = array (
'service' => "****",
'username'=>"*****",
'file'=newCurlFile("/Users/*****/*****.key", 'multipart/form-data'));
    $headers = array (
        'Accept: application/json');
    $handle = curl_init ();
    curl_setopt ( $handle, CURLOPT_URL, $url );
    curl_setopt ( $handle, CURLOPT_HTTPHEADER, $headers );
    curl_setopt ( $handle, CURLOPT_RETURNTRANSFER, true );
    curl_setopt ( $handle, CURLOPT_SSL_VERIFYHOST, false );
    curl_setopt ( $handle, CURLOPT_SSL_VERIFYPEER, false );
    curl_setopt ( $handle, CURLOPT_CONNECTTIMEOUT, 120 );
    curl_setopt ( $handle, CURLOPT_TIMEOUT, 600 );
    curl_setopt ( $handle, CURLOPT_POST, true );
    curl_setopt ( $handle, CURLOPT_POSTFIELDS, $data );
    curl_setopt ( $handle, CURLINFO_HEADER_OUT, true);
    $response = curl_exec ( $handle );
    $obj = json_decode ( $response );
    $b = $obj->{'uuid'};

```

Algoritmo 1B: Sistema de autenticação RSA.

O segundo sistema desenvolvido (Algoritmo 2) se refere ao mapeamento dos dados de nome, fila, plataforma, versões, *gateway*, *cluster*, escalonador das aplicações contidas na arquitetura do gateway. Dados esses que serão inseridos no Banco de dados do Bioinfo-Portal nas entidades *Applications*, *Queues*, *Platforms*, *Versions*, *Gateways*, *Clusters* e *Scheduler* respectivamente. Para serem consultados, analisados e servindo como base para outros sistemas, além de atingir pesquisas multidisciplinares do LNCC de diferentes grupos, dissertações e teses. A integração desse mapeamento de dados permitirá extrair informações de execuções e melhor armazenamento de informações das aplicações do Bioinfo-Portal.

```

foreach ($objN["elements"]["element"] as $valor){
    $banconom=$valor["name"];
    pg_query($conn, "INSERT INTO teste.applications (name) VALUES ('$banconom')");
}

```

```

foreach ( $valor["versions"]["version"] as $j){
    $bancovers=$j["version"];
    pg_query($conn, "INSERT INTO teste.versions (version_name) VALUES
('$bancovers')");
    $f=pg_query($conn, "SELECT id FROM teste.applications WHERE name =
'$banconom'");
    $w=pg_query($conn, "SELECT id FROM teste.versions WHERE version_name =
'$bancovers'");
    $de=pg_fetch_row($f);
    $di=pg_fetch_row($w);
    pg_query($conn, "INSERT INTO teste.apps_versions (id_app, id_version)
VALUES ($de[0], $di[0])");

foreach ($j["queues"] as $chave){
    preg_match_all("/(.)-(.)-(.)-(.)-(.)-(.)$/U", $chave, $out,
PREG_PATTERN_ORDER);

    if ($out[3][0] != 'none' && $out[3][0] != ''){ // Caso o escalonador ou o
cluster apresente 'none', não amostra valores.
    $bancoplat=$out[0][0];
    $bancoinst=$out[2][0];
    pg_query($conn, "INSERT INTO teste.institution (name) VALUES
('$bancoinst')");
    $bancosh=$out[3][0];
    pg_query($conn, "INSERT INTO teste.scheduler (name) VALUES ('$bancosh')");
    $bancoclu=$out[4][0];
    $b=pg_query($conn, "SELECT id FROM teste.scheduler WHERE name =
'$bancosh'");
    $b=pg_fetch_row($b);
    pg_query($conn, "INSERT INTO teste.clusters (name, scheduler_id) VALUES
('$bancoclu', $b[0])");
    $bancogate=$out[5][0];
    pg_query($conn, "INSERT INTO teste.gateway (name) VALUES ('$bancogate')");
    $bancoque=$out[6][0];
    $a=pg_query($conn, "SELECT id FROM teste.clusters WHERE name =
'$bancoclu'");
    $a=pg_fetch_row($a);
    pg_query($conn, "INSERT INTO teste.queues (name, cluster_id) VALUES
('$bancoque', $a[0])");
    $c=pg_query($conn, "SELECT max(id) FROM teste.queues WHERE name =
'$bancoque'");
    $c=pg_fetch_row($c);
    pg_query($conn, "INSERT INTO teste.platforms (name, queue_id) VALUES
('$bancoplat', $c[0])");
    $q=pg_query($conn, "SELECT id FROM teste.platforms WHERE name =
'$bancoplat'");
    $k=pg_query($conn, "SELECT teste.apps_versions.id FROM teste.apps_versions
LEFT JOIN teste.versions on teste.versions.id = teste.apps_versions.id_version
LEFT JOIN teste.applications on teste.applications.id =
teste.apps_versions.id_app WHERE teste.versions.version_name = '$bancovers'
and teste.applications.name = '$banconom'");
    $qe=pg_fetch_row($q);
    $ki=pg_fetch_row($k);
    pg_query($conn, "INSERT INTO teste.apps_versions_platforms (apps_versions_id,
platform_id) VALUES ($ki[0], $qe[0])");
}
}

```

Figura 3: Sistema de mapeamento de dados das aplicações contidas na arquitetura do *Gateway Bioinfo-Portal* e inserção desses dados no banco de dados Bioinfo.

CONCLUSÕES

Apresentamos resultados sobre o desenvolvimento e implementação da atualização do banco de dados acoplado à infraestrutura do Bioinfo-Portal, que permitirá armazenar dados de proveniência, por exemplo, das execuções de todas as aplicações do *gateway*. A integração de sistemas ao banco de dados centralizado e atualizado permitirá obter melhorias no desempenho relacionadas ao gerenciamento de arquivos, envio de trabalhos ou interfaces de contabilidade. Esse estudo irá sustentar o acesso aos dados para estudos realizados em paralelo no Bioinfo-Portal, análises de predição via aprendizado de máquina e etc.

Como passos próximos estão a alimentação do banco de dados com execuções do Bioinfo-Portal, o desenvolvimento de sistemas para o mapeamento de dados de localização do usuário como IP e País de origem e o mapeamento dos dados de submissão de arquivos emitido pelo usuário ao *gateway*. Dados esses que serão armazenados no banco de dados centralizado, garantindo uma maior eficiência ao *gateway* em relação a armazenamentos de dados de usuários que utilizam o Bioinfo-Portal.

REFERÊNCIAS

1. Koboldt DC, Steinberg KM, Larson DE, Wilson RK, Mardis ER. The next-generation sequencing revolution and its impact on genomics. *Cell*. 2013 Sep 26;155(1):27-38. doi: 10.1016/j.cell.2013.09.006. PMID: 24074859; PMCID: PMC3969849.
2. Ocaña, K., Galheigo, M., Osthoff, C., Gadelha, L., Porto, F., Gomes, A., Oliveira, D., Vasconcelos, A. 2020. BioinfoPortal: A scientific gateway for integrating bioinformatics applications on the Brazilian national high-performance computing network. *Future Gener. Comput. Syst.* 107, C (Jun 2020), 192–214. <https://doi.org/10.1016/j.future.2020.01.030>
3. Gesing, S., Dooley, R., Pierce, M., Krüger, J., Grunzke, R., Herres-Pawlis, S., Hoffmann., A.: Gathering requirements for advancing simulations in HPC infrastructures via science gateways. *Future Gener. Comput. Syst.* 82: 544-554 (2018)
4. Gomes, A., Bastos, B. F., Medeiros, V., Moreira, V. M. Experiences of the Brazilian national high-performance computing network on the rapid prototyping of science gateways, *Concurrency and Computation: Practice and Experience*, vol. 27, no. 2, pp. 271– 289. 2015.
5. Ramakrishnan, R., Gehrke, J. *Database management systems*, Third edition, International edition. New York: McGraw-Hill, 2003.
6. Moreau, L. et al., 'Special Issue: The First Provenance Challenge', *Concurrency and Computation: Practice and Experience*, vol. 20, no. 5, pp. 409–418, 2008.
7. *An Introduction to Database Systems*. Eighth Edition. C. J. Date. ISBN 0-321-19784-4. Addison-Wesley. 2003.
8. *A Guide to the SQL Standard*. A user's guide to the standard database language SQL. Fourth Edition. C. J. Date and Hugh Darwen. ISBN 0-201-96426-0. Addison-Wesley. 1997.

RELATÓRIO DE ATIVIDADES

TÍTULO:

Otimização do *firmware* utilizado pelo robô da Very Small Size League para aperfeiçoamento das suas habilidades, do controle e do sistema de comunicação.

ALUNO:

Nicolle Vitória Costa de Vasconcelos

ORIENTADOR:

Jauvane de Oliveira Cavalcante, PhD

Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação
(PIBITI–CNPq/LNCC)

Relatório de Atividades referente ao período de 1^o de maio de 2022 a 4 de julho de 2022

1 Objetivos

O trabalho realizado teve por objetivo, primeiramente, o aprofundamento do conhecimento necessário para desenvolver as atividades propostas. Por meio de revisão bibliográfica e de simples experimentos práticos, seria possível a compreensão dos conceitos e técnicas necessárias para a reformulação e melhoria do *firmware* e do controle do robô da categoria de robótica IEEE Very Small Size (VSS) da equipe RoboIME.

2 Introdução

2.1 Microcontroladores

Microcontroladores estão amplamente presentes em aplicações de eletrônica atualmente, desde automóveis, celulares a complexos sistemas industriais [1]. Podem ser definidos como um sistema multiprocessado, com de diferentes periféricos contidos em um *chip* [2], sendo capaz de funcionar de modo independente ofertando diversas funcionalidades. Estruturados um circuito integrado, *System-on-chip (SoC)*, reunindo os componentes de de um sistema de computador, como CPU e periféricos - Ethernet, USB, UART, SPI, I2C, entre outros - em um só *chip*. Entretanto, um *SoC*, dependendo da aplicação, pode não ser autossuficiente, neste caso, surgindo as Plataformas de Hardware, *System-on-module (SoM)*, circuitos personalizados contendo um SoC e componentes adicionais considerados necessários, como memórias e circuitos de alimentação, por exemplo.

A complexidade das instruções realizadas pela CPU de um microcontrolador pode ser medida pela arquitetura de *hardware* implementada, caracterizada como CISC, *Complex Instruction Set Computer*, ou RISC, *Reduced Instruction Set Computer*. Arquiteturas RISC, devido à simplicidade do seu conjunto de instruções e à rapidez com a qual permite executá-las, dominam atualmente o mercado, especialmente em processadores da família ARM, especificação de arquitetura criada pela ARM Holdings no final do século passado [3, 4].

Em 2006 foi lançada a arquitetura ARM Cortex-M, direcionada para microcontroladores com alto desempenho, baixo consumo e de baixo custo. Sob essa premissa, estão os microcontroladores da família STM32, produtos da empresa ST baseados em ARM Cortex-M, caracterizando um conjunto variado de plataformas de hardware servindo a diversas aplicações [5]. Em especial, a placa de desenvolvimento *BluePill*, na qual encontra-se um microcontrolador SMT32F103C8T6, possui um núcleo de processador ARM Cortex-M3 e é utilizada no desenvolvimento do presente projeto. O Cortex-M3 difere de gerações anteriores de processadores ARM em termos de periféricos de hardware e memória e por toda a sua família de processadores. A arquitetura deste núcleo consiste de um processador de 32 bits com um pequeno conjunto de periféricos principais, com arquitetura Harvard [6], isto é, usa interfaces separadas para acessar dados e instruções simultaneamente, evitando obstáculos no acesso à memória. Os micro-controladores STM32 F1 tem uma variedade de periféricos, como ADC, Conversor Analógico para Digital, GPIO (*General Purpose I/O*), I2C, SPI, vários *Timers*, USART, entre outros. O núcleo interage com o hardware periférico por meio de instruções de *load/store*, lendo e armazenando dados, como documentado nos manuais de referência da família STM32F1 [7].

Antes que qualquer um desses periféricos possam ser utilizados eles precisam ser configurados, como a habilitação do *clock* para cada um, a inicialização dos pinos, entre outras. Todos os produtos da família STM32F1 possuem um *timer* do sistema no núcleo que pode ser usados para fornecer intervalo de tempo regular, *tick*, usado para estabelecer periodização para tarefas conforme necessidade da aplicação. Para executar determinadas rotinas, os periféricos podem solicitar o núcleo, por meio de pedidos de interrupção específicos de cada periférico. Isto é, as interrupções são mecanismos de hardware fundamentais que permitem que os periféricos notifiquem o software de eventos críticos [8]. Interrupções podem ser disparadas por diversos eventos, como do próprio temporizador do sistema, falhas de acesso a memória, *resets* externos e pelos periféricos [8].

Dentre os periféricos contidos em um microcontrolador STM32F1, o UART (Transmissor/receptor assíncrono universal) é o método mais básico para comunicação serial. Em sua forma mais primitiva, é implementada por um par simétrico de fios conectando os dois dispositivos, um para transmissão e outro para recepção, não havendo uma conexão física para referência de relógio, sendo essas propriedades codificadas no fluxo de bits. O transmissor e o receptor mantém, cada um, independentemente, clocks que funcionam em uma frequência concordada, comumente chamada de taxa de transmissão, *baud rate*. O protocolo serial fornece acesso a uma ampla variedade de dispositivos, tal como acontece com as outras interfaces de comunicação, a serem exploradas no decorrer do projeto.

2.2 Sistemas de Controle

As crescentes exigências dos processos técnicos e industriais e as inúmeras possibilidades hoje oferecidas por equipamentos eletrônicos levaram ao aumento da necessidade de se aprofundar o estudo teórico de sistemas de controle [9]. Atualmente, são parte fundamental em sistemas de automóveis, aeronaves, robôs como também na própria indústria [10].

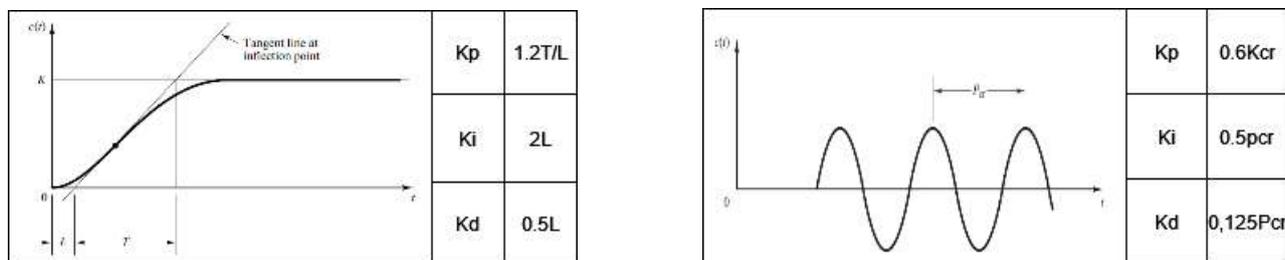
Um sistema de controle comanda ou regula processos para obter a resposta desejada. É composto, basicamente, por sensores, atuadores e controladores. Para medir o comportamento do processo, entre os sensores mais utilizados estão os de temperatura, os de pressão e os *encoders*. Os atuadores, como os motores e os relés, são os componentes responsáveis por fornecer as forças corretivas para a obtenção da saída desejada. Neste contexto, os controladores, analógicos ou digitais, geram os comandos, a partir das observações dos sensores, para os atuadores, garantindo que o sistema permaneça estável sob as condições desejadas de operação [11]. Com o aumento o desempenho e da qualidade dos microprocessadores, os controladores digitais assumiram o lugar dos analógicos em diversas situações, graças as vantagens de serem programáveis, com poucas imprecisões e capazes de implementar algoritmos complexos.

Um sistema de controle pode ser em malha aberta ou fechada. Em qualquer sistema de malha aberta a saída não é comparada com a entrada de referência, desse modo, a cada entrada de referência corresponde uma condição de operação fixa. Em consequência, a precisão do sistema depende de uma calibração. Ocorrendo perturbações, o sistema não executará a ação desejada. Em malha fechada, sinal de erro atuante, que é a diferença entre o sinal de entrada e o sinal realimentado, este podendo ser o próprio sinal de saída ou uma função dele, é introduzido no controlador de modo a reduzir o erro e trazer a saída do sistema a um valor desejado. Possui a vantagem da realimentação tornar a resposta do sistema relativamente insensível a distúrbios externos e a variações internas dos parâmetros do sistema [10, 12].

O sinal realimentado pode ser uma função linearmente proporcional ao erro do sistema, configurando o controle proporcional (P) [13, 14]. Os problemas de instabilidade e de sobressinal apresentados por controladores proporcionais com alto ganho, podem ser minimizados com a adição de um termo derivativo, o qual age no sistema causando um amortecimento, tendo assim o controle proporcional derivativo (PD) [12]. Para minimizar erros no estado estacionário, deve ser introduzido ao controlador PD o termo integral, tendo como um resultado o controlador de três termos PID [14], aumentando a precisão do sistema. Porém, quando o sensor de um sistema é susceptível a ruídos ou à interferência elétrica, ações de controle derivativas podem causar grandes flutuações de saída do sistema. Nestas circunstâncias, a melhor escolha de controlador seria o do tipo PI, proporcional integral [12].

Os parâmetros de ganho proporcional, integral e derivativo, K_P , K_I e K_D , respectivamente, a serem utilizados em um controlador PID podem ser ajustados por tentativa e erro [15] ou empregando algoritmos baseados nas características da resposta transitória de uma determinada planta de controle, como o de Ziegler e Nichols [14]. Há dois métodos chamados regras de ajuste de Ziegler – Nichols. O primeiro consiste em obter a resposta da planta

a um degrau unitário na entrada e calcular os parâmetros em função da observação da curva de saída em forma de "S". O segundo método, ao ser introduzida uma entrada senoidal de amplitude fixa, consiste em aumentar o parâmetro proporcional até que a saída do sistema exiba oscilações para um determinado valor de K_{cr} , em função do qual serão calculados os demais parâmetros.



(a) Primeiro método de Ziegler-Nichols

(b) Segundo método de Ziegler-Nichols

Figura 1: Obtenção dos parâmetros para o controle PID

3 Metodologia

Inicialmente, realizou-se o aprofundamento do conhecimento necessário à implementação de microcontroladores. Primeiramente, foram consolidados conceitos de arquitetura de *hardware* inerentes ao entendimento do funcionamento do microcontrolador STM32F103C8T6 (Figura 3) utilizado no projeto.

Em sequência, outros aspectos envolvendo programação de microcontroladores foram desenvolvidos. Para a programação do microcontrolador, foi utilizada a STM32CubeIDE [16], como plataforma de desenvolvimento C/C++, por meio da qual foram configurados os periféricos necessários e gerado e compilado o código embarcado no microcontrolador (Figura 2). Para isso, o conhecimento a ser adquirido foi segmentado em dois conjuntos de experimentos simples de modo a concretizar conceitos como a configuração de periféricos, implementação de interrupções e de interfaces de comunicação.

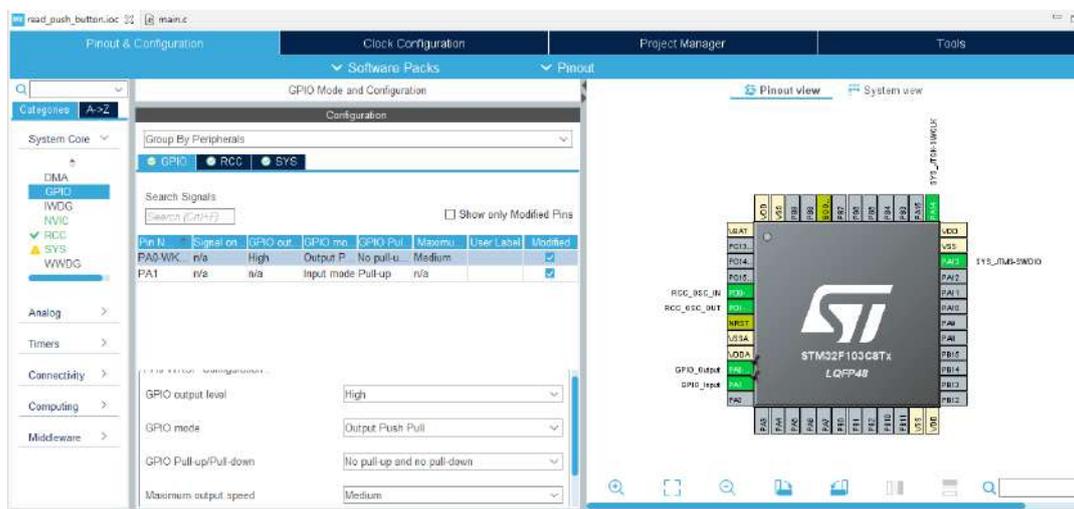


Figura 2: Configuração dos periféricos usando a plataforma de desenvolvimento STM32CubeIDE

No primeiro conjunto, foi utilizada a *BluePill* para fazer um LED acender de quatro formas diferentes:

1. com um período de 0,5 s implementando um *delay*;
2. ao ser pressionado um botão externo, o LED acende;
3. com período fixo, com as interrupções acioandas por um *timer*; e

4. ao ser pressionado um botão externo, uma interrupção é acionada e o LED é aceso.

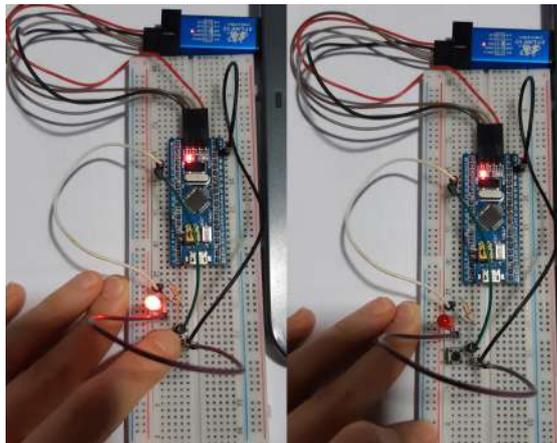


Figura 3: *BluePill* sendo utilizada na segunda prática do primeiro grupo de experimentos

Foi criado um repositório no *GitHub* para o desenvolvimento desta primeira etapa de experimentos [17].

No segundo conjunto de experimentos, foi implementada a comunicação UART entre uma *BluePill* e um segundo microcontrolador - ESP32 [18] - conectado à *Internet* de modo a enviar dados do sensor de temperatura e umidade DHT11 [19] para um servidor WEB, em tempo real [20]. O desenvolvimento referente a esta segunda parte das atividades também pode ser encontrado em um repositório *GitHub* [21]

Paralelamente à prática com microcontroladores, iniciou-se o estudo teórico de Sistemas de Controle. A revisão bibliográfica foi realizada de modo a compreender os diferentes modos de adequação e ajuste do controle implementado, analisando possibilidades e dificuldades nas soluções apresentadas na literatura. Diante dos métodos pesquisados, destaca-se popularmente o controle PID. Sendo este o método de controle que se deseja implementar nos robôs da VSS, foi estudado como os dados dos sensores, *encoders* ópticos [22], são lidos e interpretados pelo microcontrolador, como são obtidos os parâmetros do controle e como são enviados os comandos da velocidade desejada para o robô.

4 Resultados e Discussão

4.1 Experimentos com a *BluePill*

As práticas realizadas apontaram, de forma bastante direta, a aplicação de técnicas de programação de microcontroladores e reforçaram o aprendizado dos conceitos necessários. O primeiro conjunto de experimentos serviram como uma breve introdução à prática com o STM32F103C8T6, uma vez que, de forma progressiva, o conhecimento foi aprofundado. Partindo do uso de simples funções de *delays* até a aplicação mais complexa de interrupções para realizar rotinas específicas, foi possível comparar aspectos referentes à eficiência do *firmware* implementado. O uso de interrupções, embora inicialmente apresente certo nível de dificuldade para implementação, é vantajoso pois permitiu a otimização do código. Ao se criar corretamente várias rotinas de interrupção, torna-se possível executar mais de uma tarefa de modo ágil e sem que haja interferência entre elas.

Como os primeiros experimentos realizados possuíam um baixo nível de complexidade, não houve diferenças muito significativas ao comparar o funcionamento ao substituir os *delays*. Entretanto, ao realizar o segundo conjunto de experimentos, evidenciaram-se as vantagens da implementação dos pedidos de interrupção, uma vez que, inicialmente, o uso de *delays* incorreu em pequenos atrasos, que no decorrer do funcionamento prejudicaram a leitura dos dados do sensor, levando a perda recorrente de informações. Ao fazer o correto uso das rotinas de interrupção, foi possível, além de reestruturar o código de maneira mais eficiente e econômica, reduzir as perdas ao serem feitas as leituras do sensor e minimizar o consumo energético, otimizando o funcionamento geral do *firmware*.

A partir dessa abordagem prática, foi possível avaliar técnicas de otimização a serem implementadas em busca da melhoria da eficácia e reformulação geral do *firmware* do robô da VSS para um código mais eficiente e conciso.

4.2 Controle PID

A velocidade dos robôs da VSS é regulada utilizando o controle proporcional, em um sistema de malha fechada, no qual a velocidade de cada um dos dois motores - responsáveis por movimentar individualmente as duas rodas - pode ser encontrada ao interpretar os dados obtidos do *encoder* óptico relacionado a cada um deles.

Para cada um dos sensores, há um *timer* do microcontrolador que é ativado no modo de *encoder*. Cada *encoder* [22] gera dois sinais senoidais em quadratura, por meio dos quais é possível obter a posição e a direção da rotação do motor. A cada inversão de polaridade de qualquer um desses dois sinais, o contador do *timer* utilizado é automaticamente incrementado. Utilizando um segundo *timer* em modo de captura, é possível, por meio de interrupções temporizadas, ler o valor do contador incrementado. Consultando as especificações técnicas dos *encoders* e do motor [23], pode-se calcular a velocidade de rotação de cada motor, sabendo que a cada incremento do contador, o motor realizou 1/20 de rotação. Para a velocidade das rodas, a razão é de 75 rotações do motor para uma rotação da roda. Conhecendo as dimensões da roda, o valor do contador e o intervalo de tempo no qual foi incrementado, é calculada, então, a velocidade de cada roda.

Atualmente, o controle implementado reduz-se apenas ao proporcional, de modo que é calculado o erro do sistema - conhecendo a velocidade desejada e a imprimida por cada motor - e o comando de velocidade é enviado é linearmente proporcional ao erro. A constante K_P foi calculada utilizando o segundo método de Ziegler-Nichols, encontrando um valor de $K_P = 50$.

Cabe ressaltar a necessidade de serem estendidos os estudos no que tange o controle implementado, uma vez que apenas a constante proporcional não é suficiente para o funcionamento otimizado do robô. É possível perceber problemas de sobressinal e de instabilidade no sistema em regime permanente, evidenciando a necessidade da adição dos termos derivativos e integral ao sistema de controle.

5 Conclusões

Diante das atividades desenvolvidas, foi possível consolidar o aprendizado em diversos aspectos relacionados aos objetivos principais do projeto. As práticas realizadas com o microcontrolador utilizado no projeto permitiu a construção do conhecimento acerca do funcionamento do *firmware* e o esclarecimento de conceitos de programação considerados muitas vezes tidos como abstratos. Além disso, a segmentação e a crescente complexidade relacionada às práticas apontou as principais dificuldades no que tange a eficiência do *firmware* do robô. Sendo assim, foram de extrema utilidade para o desenvolvimento e aprimoramento do projeto.

Ainda, os estudos teóricos acerca de sistemas de controle evidenciaram a sua relevância no contexto da Engenharia. É preciso que haja a adequada regulagem de processos em busca das respostas desejadas. Nesse sentido, foram exemplificadas diversas técnicas empregadas e suas vantagens frente as demais. Sendo assim, percebe-se que o controle implementado no *firmware* do robô da VSS atualmente requer uma reformulação com o objetivo de refinar seus parâmetros a fim de se obter respostas mais próximas das desejadas.

Como meio de aprofundar o aprendizado na área de controle e dar prosseguimento ao projeto, planejam-se alguns experimentos de simulação do motor utilizado e práticas com o STM32F103C8T6. Primeiramente, uso de recursos computacionais para simular o funcionamento do motor será de utilidade para o refinamento dos parâmetros de controle. Além disso, implementações simples usando o microcontrolador deverão ser realizadas para que seja compreendido como são interpretados os dados obtidos dos sensores e como os comandos de controle são enviados para os motores do robô.

Referências

- [1] D. Cortelli, *Introdução a programação de microcontroladores Microchip PIC*. SENAI, 2006.
- [2] C. Lima and M. Villaça, *AVR e Arduino: técnicas de projeto*, 2012.
- [3] D. McCarthy, *Architectures for parallel DSP-based adaptive optics feedback control*. United States: SPIE, 1999.
- [4] C. Fonseca Filho, *História da computação: O Caminho do Pensamento e da Tecnologia*. RS-BRASIL: EDIPURS, 2007.
- [5] G. Brown, *Discovering the STM32 Microcontroller*. Indiana University, 2012.
- [6] A. Francillon and C. Castelluccia, *Code injection attacks on harvard-architecture devices*. ACM, 2008.
- [7] ST Electronics, “RM0008 Reference manual,” https://www.st.com/resource/en/reference_manual/cd00171190-stm32f101xx-stm32f102xx-stm32f103xx-stm32f105xx-and-stm32f107xx-advanced-arm-based-32-bit-mc.pdf, 2021.
- [8] S. Electronics, “STM32 High Performance MCUs,” <https://www.st.com/en/microcontrollers-microprocessors/stm32-high-performance-mcus.html#overview>, 2020.
- [9] R. Isermann, *Digital Control Systems*. Springer, 1989.
- [10] O. Katsuhiko, *Engenharia de Controle Moderno*. Springer, 1993.
- [11] T. Instruments, *Digital control application with the TM5320 family*. Texas Instruments, 1996.
- [12] A. L. Giorgini, *Implementação de um controlador PID digital para robótica baseado em computação reconfigurável*. ICMC-USP, 2020.
- [13] Franklin Gene, “Sistemas de controle para engenharia.”
- [14] N. S. M. M. Oliveira and G. R. A. Neto, *DESENVOLVIMENTO DE SOLUÇÃO PARA AÇÃO CONTINUADA PARA MÚLTIPLOS VEÍCULOS NÃO TRIPULADOS EM TAREFAS COOPERATIVAS*. Instituto Militar de Engenharia, 2019.
- [15] C. Junior and J. Roberto, *Controladores PID industriais com sintonia automática por realimentação a relê*. USP, 1999.
- [16] ST Electronics, “Integrated Development Environment for STM32 ,” <https://www.st.com/en/development-tools/stm32cubeide.html>, 2019.
- [17] N. V. C. Vasconcelos, “pibiti2021,” <https://github.com/nicollevtr/pibiti2021>, 2021.
- [18] Espressif Systems, “ESP32 Technical Reference manual,” https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf, 2021.
- [19] Mouser Electronics, “DHT11 Humidity and Temperature Sensor,” <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>.
- [20] ThingSpeak, “Controle de Temperatura e Umidade,” <https://thingspeak.com/channels/174144>.
- [21] N. V. C. Vasconcelos, “Microcontroladores 2022,” <https://github.com/nicollevtr/Microcontroladores2022>, 2022.
- [22] P. Robotics and Electronics, “Optical encoder pair kit for micro metal gearmotors, 3.3v,” <https://www.pololu.com/product/2591>.
- [23] P. R. . Electronics, “75:1 micro metal gearmotor hpcb 6v with extended motor shaft,” <https://www.pololu.com/product/3074>.

SIMULAÇÃO COMPUTACIONAL DE PARÂMETROS DE ESCOAMENTO SANGUÍNEO

Bolsista: Rodrigo Schmidt

Orientador: Pablo Javier Blanco

Tipo de bolsa: PIBIC

Vigência: 01/08/2021 até 31/07/2022

Sumário

1) OBJETIVOS.....	3
2) INTRODUÇÃO	3
3) METODOLOGIA	4
4) RESULTADOS.....	4
5) CONCLUSÕES.....	7
6) REFERÊNCIAS	7

1) OBJETIVOS

Os principais objetivos do trabalho desenvolvido são:

- Abordar os modelos resistivo e resistivo-complacente que representam parte do sistema cardiovascular;
- Identificar as variáveis atuantes em cada modelo;
- Determinar as variáveis do sistema, a partir de condições iniciais arbitrárias, através de métodos numéricos;
- Utilizar a linguagem de programação Python para resolver o sistema de equações.

2) INTRODUÇÃO

No estudo da dinâmica do escoamento sanguíneo, dois parâmetros são essenciais: a Pressão e a Vazão. No campo da fisiologia, esses parâmetros são importantes pois determinam como o sangue é conduzido até os vasos capilares, com diâmetro inferior a 20 μm , onde ocorrem as trocas de substâncias com as células. Nesses vasos, um aumento na pressão em um determinado ponto e a diminuição da vazão podem significar uma interrupção no escoamento sanguíneo.

Hoppensteadt e Peskin^[2] desenvolveram em 2002 um modelo completo do sistema cardiovascular humano, onde não há variações espaciais das variáveis e parâmetros globais empregados. Este modelo ficou conhecido como 0D (zero dimensional). Blanco e Feijóo^[1] apresentaram em 2009 uma variação do modelo de Hoppensteadt e Peskin^[2], introduzindo o conceito da inércia no funcionamento do coração, das válvulas e dos vasos sanguíneos que compõem o sistema.

A queda de pressão em um vaso sanguíneo pode ser determinada de forma simples através de uma equação algébrica. No entanto, quando se considera uma bifurcação, com um vaso pai e dois vasos filhos, o número de equações passa a ser três, e é necessário resolver o sistema de equações. Conforme o número de vasos e bifurcações aumenta, cresce proporcionalmente o número de equações no sistema a ser resolvido, o que inviabiliza uma solução manual.

Nesse contexto surge a proposta de solução automatizada desse sistema de equações. Condições iniciais são impostas arbitrariamente para que seja possível resolver matematicamente o sistema. A linguagem de programação Python possui recursos suficientes para resolver o problema proposto.

3) METODOLOGIA

Inicialmente, o trabalho foi desenvolvido em modelo puramente resistivo, onde apenas a resistência dos vasos é levada em consideração, e deseja-se conhecer a pressão e a vazão em diferentes pontos do sistema. Este modelo foi desenvolvido para permitir uma análise plena da rede de vasos abordada, bem como das técnicas de solução de um sistema de equações.

Em seguida, foi desenvolvido um modelo resistivo-complacente-inercial, onde se aborda a resistência, a complacência e a inércia dos vasos sanguíneos. Essas variáveis podem ser associadas à resistência, capacitância e indutância de circuitos elétricos, o que permite uma abordagem mais simplificada dos parâmetros da simulação.

O trabalho foi desenvolvido na linguagem de programação Python, a partir da distribuição Jupyter Notebook, que permite a programação em blocos, o que facilita em caso de alterações, e a escrita de equações.

4) RESULTADOS

Conforme supracitado, inicialmente foi proposta a solução do sistema de equações para um esquema de vasos puramente resistivos, desprezando a complacência dos vasos e a inércia do fluido em seu interior. Um exemplo simplificado da rede proposta é apresentado na Figura 1. Os números em vermelho representam a identificação dos vasos (*vessels*), e os números em preto a identificação das bifurcações (*junctions*). As variáveis do sistema são determinadas a partir da Equação 1, que é a equação geral que governa o modelo^[1].

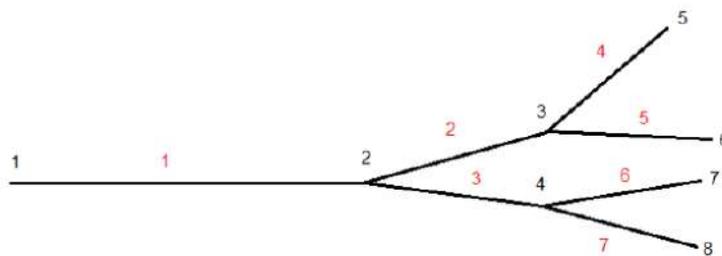


Figura 1 - Rede de vasos puramente resistivos. Fonte: O autor.

$$P_{input} - P_{output} = RQ_{output}$$

Equação 1 - Equação geral do sistema resistivo.

Para esse sistema, há 15 variáveis de Pressão e Vazão, enquanto há somente 10 equações geradas. Dessa forma, é necessário conhecer 5 variáveis para solucionar o sistema algebricamente. Conquanto, foram escolhidas a Vazão no vaso 1 (Q_1), com valor

10, e Pressão nas *junctions* 5, 6, 7 e 8, atribuindo-se o valor 0. Também é necessário determinar a resistência dos vasos como parâmetro constante do sistema, escolhendo-se o valor 10 para todos os vasos.

Contudo, foi desenvolvido um código em Python para a solução do problema. Os resultados são apresentados na Tabela 1. O código foi desenvolvido com o objetivo de ser automático quanto aos parâmetros de entrada. Dessa forma, caso seja requerido alterar alguma condição de entrada ou colocar outras variáveis no lugar das apresentadas acima, basta apenas informar os novos parâmetros de entrada, diferenciando entre vazão e pressão, e o valor conhecido.

$P_1 = 175$	$P_2 = 75$
$P_3 = 25$	$P_4 = 25$
$Q_2 = 5$	$Q_3 = 5$
$Q_4 = 2,5$	$Q_5 = 2,5$
$Q_6 = 2,5$	$Q_7 = 2,5$

Tabela 1 - Solução do sistema de equações do modelo resistivo

Em seguida, foi desenvolvido o modelo considerando a resistência e a complacência dos vasos, além da inércia do fluido em seu interior. O modelo desenvolvido possui apenas uma bifurcação central (3), que faz a divisão do fluxo do vaso pai para dois vasos filhos. Esse modelo pode ser visto na Figura 2. Os parâmetros são determinados a partir das equações Equação 2 e Equação 3. Optou-se por representar a resistência e a inércia antes e depois da complacência, de forma a assemelhar com o comportamento real dos vasos sanguíneos.

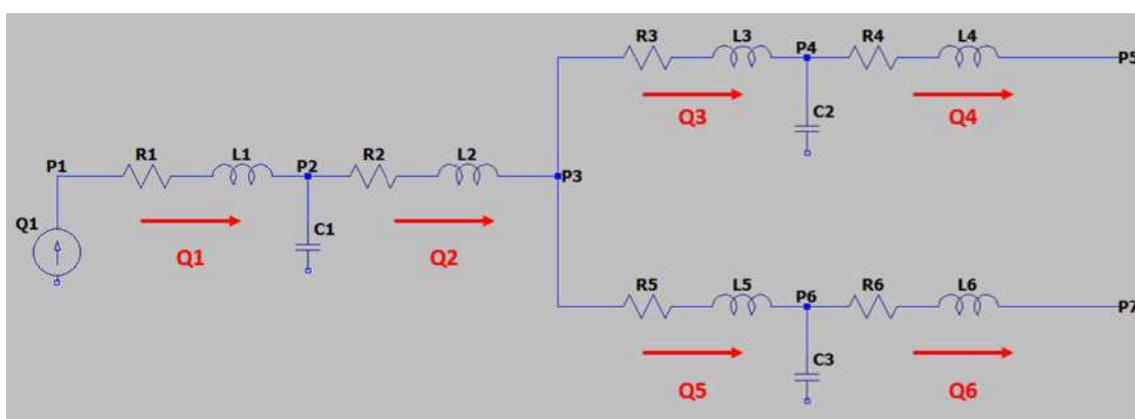


Figura 2 - Modelo resistivo-complacente-inercial. Fonte: O autor.

$$P_{in} - P_{out} = RQ_o + L \frac{dQ_o}{dt}$$

Equação 2 – Equação geral da queda de pressão ao longo do vaso

$$C \frac{dP_{in}}{dt} = Q_{in} - Q_{out}$$

Equação 3 - Equação geral da queda da vazão ao longo do vaso

No entanto, para solucionar o sistema de forma mais simples, os termos diferenciais foram substituídos por aproximações algébricas, representadas de forma generalizada na Equação 4. Neste caso, é necessário conhecer a inicialização de todos os parâmetros do sistema. Optou-se por inicializar todas as variáveis com 0. Todavia, 3 parâmetros devem ser conhecidos, para que as 13 incógnitas sejam determinadas algebricamente a partir do sistema de 13 equações. Desta forma, optou-se pela vazão Q_1 com valor de 10 e as pressões P_5 e P_7 , com valor 0. Esses parâmetros foram mantidos constantes durante a solução do sistema. Também foram definidos os valores das outras propriedades constantes do sistema, sendo todas as resistências iguais a 1, as indutâncias iguais a 0,1 e as capacitâncias iguais a 0,001.

$$\frac{dX}{dt} = \frac{X^{n+1} - X^n}{\Delta t}$$

Equação 4 - Substituição dos termos diferenciais por aproximações algébricas

Contudo, um novo código em Python foi desenvolvido. Na Figura 3 alguns parâmetros são exibidos, como a pressão P_3 (*junction* central), e as vazões Q_3 e Q_5 .

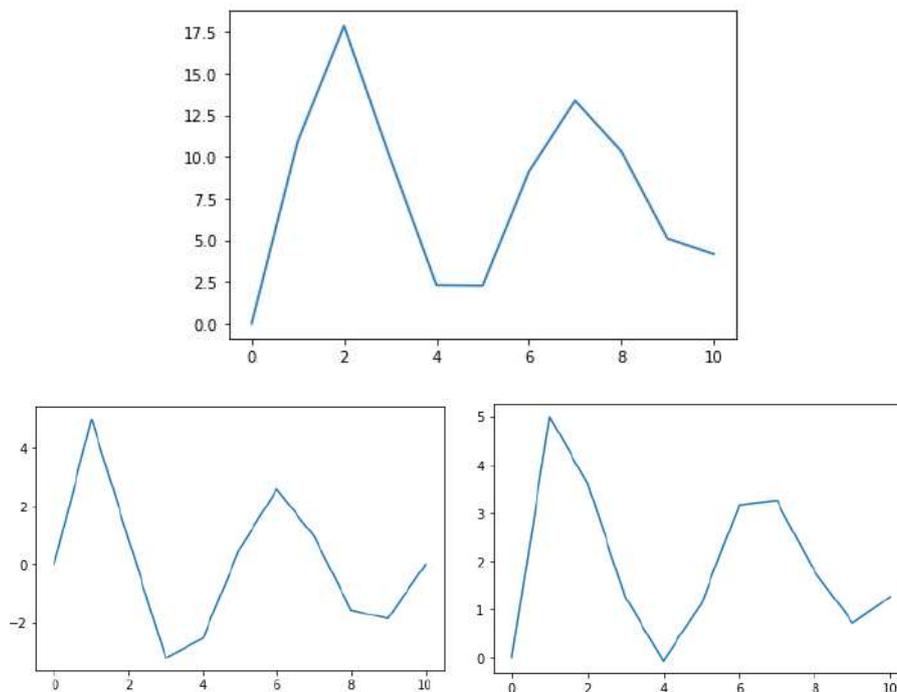


Figura 3 - Representações de P_3 (superior), Q_3 (inferior esquerda) e Q_5 (inferior direita). Fonte: O autor.

5) CONCLUSÕES

Conforme os resultados apresentados acima, pode-se concluir que a metodologia implementada é capaz de realizar a simulação dos parâmetros de escoamentos sanguíneos. Conforme proposto, os códigos foram desenvolvidos de forma a serem adaptados a qualquer rede de vasos, bastando apenas que as condições iniciais sejam definidas. Portanto, não se torna necessário realizar nenhum tipo de mudança significativa nos códigos para que o modelo contenha mais vasos e bifurcações, por exemplo.

Entretanto, pode-se concluir que os resultados apresentados estão dentro do esperado, e que o trabalho desenvolvido agregou diversos conhecimentos ao bolsista acerca da simulação de escoamentos sanguíneos.

6) REFERÊNCIAS

- [1] P.J. Blanco and R. A. Feijóo. *Introdução à Modelagem e Simulação Computacional do Sistema Cardiovascular Humano*. Monografia, Laboratório Nacional de Computação Científica, 2011.
- [2] F. C. Hoppensteadt and C. S. Peskin. *Modeling and Simulation in Medicine and the Life Sciences*. Texts in Applied Mathematics, Springer, Second Edition, 2002.

Modelagem de vigas de Euler Bernoulli e Timoshenko

Bolsista: Tarsiane Ribeiro da Costa

Orientador: Jaime Ediberto Munoz Rivera

Bolsista de Iniciação Científica – CNPq

Período do desenvolvimento: 01/04/22 à 31/07/2022

I. Objetivo

1) Desenvolver as técnicas de modelagem para problemas de equilíbrio e de evolução dos modelos de Euler-Bernoulli e de Timoshenko.

2) Desenvolver métodos numéricos (diferenças e elementos finitos) para os correspondentes modelos estacionários e de evolução. Finalmente, estudar propriedades qualitativas dos modelos de evolução:

- Conservação da energia;
- Dedução dos modelos de equilíbrio e oscilatórios;
- Aplicações para a modelagem de equilíbrios de vigas na engenharia civil;
- Modelamento de mecanismos dissipativos: Estudo e modelamento dos diversos mecanismos dissipativos ;
- Decaimento para zero das vibrações tanto do problema de Euler-Bernoulli como de Timoshenko.

II. Introdução

Muitas estruturas da Engenharia Civil, tais como pontes longas, são modeladas através de uma equação de vigas. Devido aos problemas operacionais e considerações de segurança e performance, os engenheiros utilizam vigas interconectadas umas com as outras para construir tais estruturas. Estas interconexões produzem os chamados problemas de transmissão.



O primeiro problema consiste em modelar as diferentes formas de interconexão entre as vigas. Consideramos num primeiro momento o modelo estacionário, tanto para

os modelos de vigas de Timoshenko como o modelo de Euler Bernoulli. Posteriormente consideramos os correspondentes problemas de evolução no caso conservativo.

O segundo problema consiste em introduzir algum tipo de mecanismo externo que possibilita a estabilização da estrutura. Estes mecanismos podem ser através de controles colocados entre as interconexões da viga, ou também devido às leis constitutivas de cada componente, por exemplo uma componente pode ser viscosa, outra pode ser termoelástica, outra pode ser um material com memória. Mostraremos que a variação de temperatura numa viga produz efeitos mecânicos (dilatação e contração) que são capazes de estabilizar uniformemente a estrutura.

III. Material e Métodos ou Metodologia

Usaremos o Princípio de Bernoulli e métodos variacionais para chegar ao modelo que define a posição de equilíbrio de uma viga. Este princípio estabelece que: **A posição de equilíbrios dos corpos elásticos é aquela que minimiza sua energia potencial.** Seguiremos as teorias de Euler Bernoulli e Timoshenko. Para isto, desenvolvemos técnicas minimizantes. Consideramos também o modelo vibratório. Para chegar ao modelo de evolução usaremos as leis de Newton que definem o movimento de corpos acelerados.

IV. Resultado e Discussão

Iremos deduzir o modelo estacionário e dinâmico da equação da viga. Provaremos que os modelos correspondentes estão bem colocados no sentido de que possuem existência e unicidade. Uma vez concluída a primeira fase sobre a dedução físicas dos modelos estacionários de Euler Bernoulli e de Timoshenko, começaremos a estudar métodos numéricos para aproximação da solução dos correspondentes problemas.

Desenvolveremos os métodos de diferenças finitas e de elementos finitos, para ambos os problemas. Para desenvolver o método de diferenças finitas faremos um estudo prévio sobre o teorema de Taylor. Para desenvolver o método de elementos finitos, faremos uma introdução às séries de Fourier assim como aos espaços vetoriais e uma breve introdução aos espaços de Sobolev.

V. Conclusões

1. Entre as conclusões do problema que estamos estudando podemos citar:
 - As diferenças centrais usadas nas aproximações produzem um erro estimado pelo quadrado das aproximações.

- O método das diferenças finitas é eficiente para encontrar soluções aproximadas de segunda ordem.
- As matrizes dos coeficientes no caso unidimensional são matrizes de 3 diagonais simples de resolver.

VI. Referências Bibliográficas

[1]. M. Renardy, R.C. Rogers; An Introduction to partial differential equations, Sringer-Verlag (1993) Vol. 1.

[2]. .R. V. Churchill; Fourier Series and boundary value problems, McGraw-Hill, New York (1988) Vol. I.

[3]. J.E. Munoz Rivera; Introdução às Equações Diferenciais Parciais, LNCC Vol. 1, pages 1-230, (2004). ISBN 978-85-9996108-7

[4]. J.E. Munoz Rivera; Teoria das distribuições e EDP, LNCC Vol. 1, pages 1- 244, (2004). ISBN 85-99961-06-3

Ministério da Ciência, Tecnologia e Inovações
Laboratório Nacional de Computação Científica - LNCC
Instituto Militar de Engenharia
Curso de Formação e Graduação

Relatório de Atividades
Submetido ao Programa Institucional de Bolsas de
Iniciação Científica (PIBIC-LNCC) Edital 2021/2022

Título:

Plataforma móvel Robótica-UV para desinfecção de ambientes: uma solução no combate à COVID-19 e outros agentes patogênicos

Aluno:

Victor Freixinho Nunes

Orientadores:

Paulo Fernando Ferreira Rosa (IME), PhD

Jauvane Cavalcante de Oliveira (LNCC), PhD

Rio de Janeiro
2022

Sumário

1	Objetivos	2
2	Introdução	2
3	Material e Metodologia	3
4	Resultados e Discussão	4
5	Conclusões	4
6	Referências Bibliográficas	5

1 Objetivos

O principal objetivo do trabalho a ser realizado consiste em aprimorar a plataforma móvel Robótica-UV para desinfecção de ambientes, já desenvolvida no IME, adicionando a capacidade de locomoção autônoma da plataforma, de modo que ela possa navegar por um ambiente não estruturado, sem intervenções humanas. Para adicionar essa nova capacidade, serão estudados e testados diferentes algoritmos de navegação, baseados principalmente no SLAM (*Simultaneous Localization and Mapping*), para posteriormente implementar o algoritmo mais eficiente na plataforma.

2 Introdução

A doença COVID-19 causada pelo vírus SARS-CoV-2 [1] causou incontáveis mortes no mundo inteiro [2]. Na conjuntura atual, o surgimento das vacinas provoca na população um relaxamento do cumprimento das medidas sanitárias, e isso é um problema, pois a transmissão do vírus ocorre através do ar, por meio de gotículas que podem ser expelidas pelos humanos [3]. Essas gotículas podem cair em diferentes superfícies e serem transportadas para outros lugares por mãos humanas contaminadas. Uma forma de desinfecção já largamente utilizada por ambientes hospitalares é a adoção da radiação UVC [4], através do método UVGI (*Ultraviolet germicidal irradiation*).

Com base neste cenário, foi feito desenvolvimento do robô, inicialmente controlado remotamente por um homem, para desinfecção de ambientes internos, com foco inicial em salas de aula.



Figura 1: Plataforma móvel Robótica-UV desenvolvida.

3 Material e Metodologia

O robô possui embarcado na placa mãe o microcontrolador - que processa os dados sensoriais; a unidade IMU (*Inertial Measurement Unit*) — utilizada para garantir a precisão de movimento através de acelerômetros, giroscópios e magnetômetros; conectores para os sensores ultrassônicos — úteis para evitar a colisão física do robô com superfícies do ambiente; conectores para interfaceamento com a câmera USB.

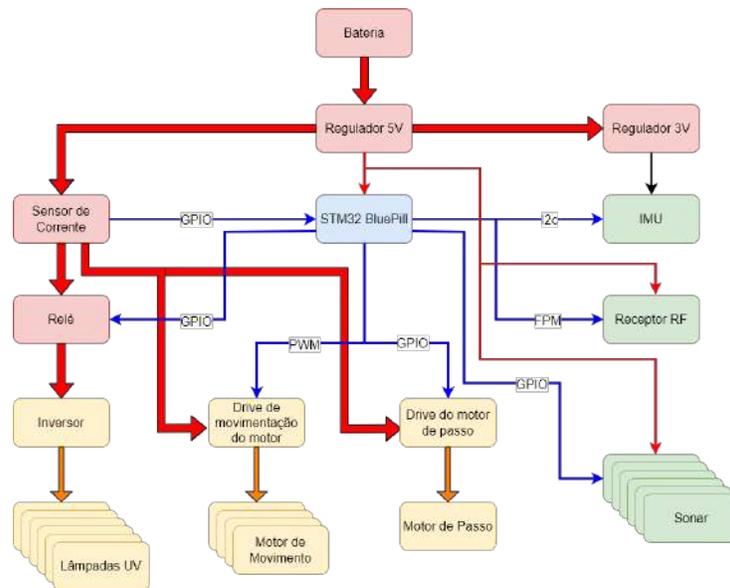


Figura 2: Arquitetura Geral da Eletrônica Embarcada do robô.

Utilizando os dados dos sensores e da câmera, é possível então desenvolver um algoritmo de Inteligência Artificial que interprete esses dados, de modo que o robô seja capaz de se localizar, mapear o local e então navegar pelo ambiente não estruturado de maneira eficiente, ou seja, evitando colisões, desinfectando áreas de modo preciso e calculando rotas de modo a minimizar o consumo de bateria.

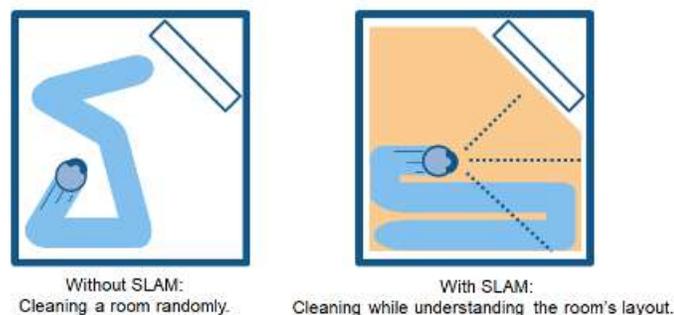


Figura 3: Exemplo de benefícios do SLAM para robôs de limpeza [5].

4 Resultados e Discussão

Objetiva-se estudar e testar diferentes algoritmos e técnicas de navegação autônoma, baseados principalmente em SLAM, dado a necessidade de locomoção com desvio de obstáculos para o robô e a construção de um mapa, para otimizar o tempo de passeio e economizar o consumo de energia. Deste modo, os resultados poderão ser aplicados na plataforma móvel Robótica-UV, de modo a otimizar sua usabilidade.

5 Conclusões

Conforme o apresentado, o estudo e o aperfeiçoamento de algoritmos e técnicas de navegação autônoma com base em SLAM permite ao robô se mover de forma eficiente e independente por um ambiente não estruturado, sendo possível aplicar os resultados encontrados na plataforma móvel Robótica-UV, de modo a viabilizar a locomoção autônoma desta. Isso será significativo para o aprendizado de Engenharia de Computação, seja no estudo teórico de linguagens de programação, ou na aplicação prática de algoritmos de *machine learning*.

6 Referências Bibliográficas

- [1] R. M. Lana, F. C. Coelho, M. F. d. C. Gomes, O. G. Cruz, L. S. Bastos, D. A. M. Villela, and C. T. Codeço, “The novel coronavirus (sars-cov-2) emergency and the role of timely and effective national health surveillance,” *Cad. Saúde Pública [online]*, vol. 36, 00 2020. [Online]. Available: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-311X2020000300301&nrm=iso
- [2] Our World in Data, “Coronavirus pandemic (covid-19) – the data,” <https://ourworldindata.org/coronavirus-data>, 2021, accessed: 2021-11-11.
- [3] D. Fisher and D. Heymann, “Q&a: The novel coronavirus outbreak causing covid-19,” *BMC Medicine*, vol. 18, p. 57, 02 2020.
- [4] A. Johnson, “An evaluation of the ultra-vtm (ultra-violet) decontamination system as an adjunct to cleaning in a district general hospital,” *Journal of Hospital Infection*, vol. 94, 07 2016.
- [5] “What is slam? 3 things you need to know,” <https://www.mathworks.com/discovery/slam.html>, accessed: 2022-06-30.

Relatório de Atividades

Título: Gerenciamento de grandes volumes de dados nos sistemas Web do SINAPAD

Bolsista: Lucas Fernandes Calheiros

Orientador: Antônio Tadeu Azevedo Gomes

Tipo da bolsa: PIBIT/LNCC

Período: Set/2021 a Jul/2022

Áreas prioritárias (Portaria MCTI 1.122/2020):

Tecnologias Habilitadoras (Inteligência Artificial)

-

Sub-área do conhecimento (Tabela CNPq):

1.03.03.00-6 – Metodologia e Técnicas da Computação (Engenharia de Software)

1 Objetivos

O objetivo principal deste projeto é: **investigar alternativas arquiteturais para os sistemas Web do SINAPAD, considerando abordagens modernas de gerenciamento de grandes volumes de dados, e prototipar soluções baseadas na(s) alternativa(s) mais promissora(s).**

2 Introdução

Como parte de seus objetivos, o sistema SINAPAD oferece 3 sistemas Web:

- O sistema de gerenciamento de projetos, de utilização de recursos e de produções científicas associados ao supercomputador Santos Dumont (<http://sdumont.lncc.br> e <https://intranet.sdumont.lncc.br>);
- O portal de modelagem molecular Dockthor (<http://dockthor.lncc.br>); e
- O portal de bioinformática BioInfo (<http://bioinfo.lncc.br>).

Os 3 sistemas empregam basicamente as mesmas tecnologias de desenvolvimento Web (Bootstrap, AngularJS, JQuery) e de bancos de dados (MySQL). Mais importante, esses sistemas compartilham uma estrutura arquitetural semelhante, com aplicação de *design patterns* consagrados na literatura de desenvolvimento Web, como *Model-View-Controller* [1].

Todos os sistemas Web do SINAPAD podem ser hoje considerados *sistemas intensivos de dados*, isto é, sistemas que manipulam um grande volume de dados e que apresentam atributos de qualidade de desempenho como escalabilidade e disponibilidade. Esses sistemas também podem ser considerados como *reativos*, no sentido de demandarem atributos de qualidade como usabilidade e tempo de resposta. Para lidar com essa combinação “volume de dados X reatividade”, a equipe SINAPAD desenvolveu adaptações específicas às tecnologias listadas acima, adaptações estas que foram evoluindo em consonância com a evolução dessas tecnologias. Um efeito negativo dessa evolução acoplada foi a degradação arquitetural dos sistemas Web do SINAPAD, que se tornaram demasiado complexos de se manter.

Com o advento de sistemas específicos de gerenciamento de grandes volumes de dados (como Spark e Cassandra) e de processamento de eventos (como Kafka), a equipe SINAPAD tem a oportunidade de rever a organização arquitetural dos sistemas Web do SINAPAD, de modo a melhor atender os atributos de qualidade supracitados. Além disso, o suporte dessas tecnologias abre a possibilidade de uso de soluções de Inteligência Artificial que apoiem as tomadas de decisão dos usuários desses sistemas no nível de gestão, como por exemplo na atualização das políticas de filas do supercomputador Santos Dumont.

Para alcançar o objetivo do projeto, propomos o seguinte conjunto de atividades e vislumbramos os resultados esperados correspondentes:

1. Estudar arquiteturas baseadas em microsserviços / Conhecimento adquirido;
2. Estudar arquiteturas dirigidas por eventos / Conhecimento adquirido;
3. Analisar tecnologias existentes de gerenciamento de grandes volumes de dados / Conhecimento adquirido;
4. Analisar tecnologias existentes de processamento de eventos / Conhecimento adquirido;
5. Propor alternativas arquiteturais para os sistemas Web do SINAPAD / Relatório técnico;
6. Prototipar continuamente os sistemas Web do SINAPAD com base nas tecnologias estudadas e alternativas arquiteturais propostas / Pacote de software.

3 Metodologia

Dentro do contexto da atividade 5, remodelamos a arquitetura do sistema de gerenciamento do Santos Dumont, tendo em vista que alteramos a arquitetura inicialmente composta apenas pelo docker (ilustrada na Figura 1) para uma nova arquitetura baseada no Kubernetes, onde podemos explorar o conceito de pods contendo mais do que um container em cada um, sendo assim mais eficiente,

mais seguro e mais tolerante a falhas. Como passo seguinte, pela atividade 1, foi realizado o estudo de arquiteturas baseadas em microsserviços, empregando o uso do kubernetes ¹ como estudo de caso e mapeando assim os componentes da nova arquitetura em diferentes pods, com múltiplos containers (como ilustrado na Figura 2).

Após a implementação do kubernetes, voltamos à atividade 6, prototipando a parte do sistema de gerenciamento do SDumont responsável pela apresentação de gráficos de monitoramento e de estatísticas das filas e de projetos do SDumont. Para tanto, estamos utilizando o Grafana,², uma ferramenta para consultar e visualizar métricas graficamente, porém, recentemente começamos a estudar e prototipar novas ferramentas para visualização de gráfico, mais especificamente, Notebook Jupyter,³ que engloba também a atividade 3 do plano de trabalho, já que com esta ferramenta, seremos capazes de no futuro controlar melhor o gerenciamento dos dados do SDumont, podendo assim criar painéis mais interativos e proporcionar mais liberdade no uso de dados (como na Figura 3).

¹<https://kubernetes.io/>

²<http://grafana.com>

³<https://jupyter.org/>

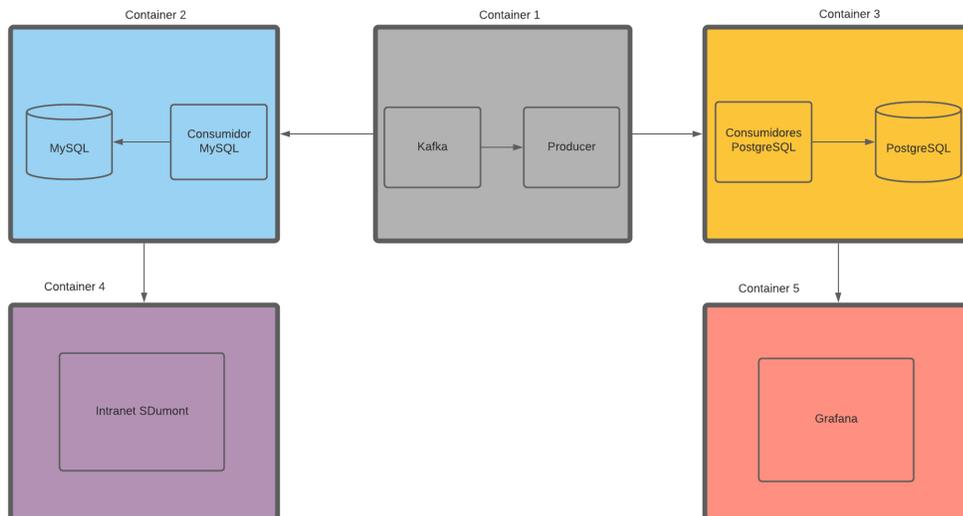


Figura 1: Arquitetura do sistema de gerenciamento do SDumont proposta inicialmente baseada no Docker.

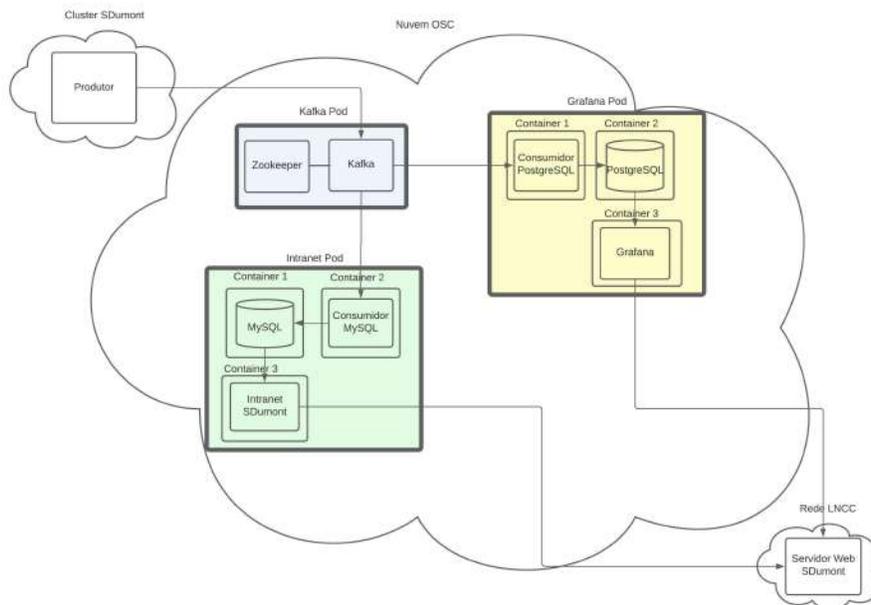


Figura 2: Nova arquitetura proposta, baseada em Kubernetes.



Figura 3: Protótipo de gráfico gerado pelo Jupyter.

4 Resultados

Após os estudos e aplicação das novas tecnologias, obtivemos uma arquitetura aprimorada, com maior segurança, maior tolerância a falhas e também mais eficiente em comparação à arquitetura antes proposta. Mais à frente pretende-se refinar a implementação dos gráficos de monitoramento e de apresentação de estatísticas das filas com a utilização do Notebook Jupyter, que está atualmente sendo estudado e a implementação destes no sistema de produção.

5 Conclusões

Desde o último relatório, tivemos uma grande evolução na arquitetura do nosso sistema, tendo agora uma base sólida no Kubernetes, o que nos dá mais chão para explorar novas alternativas de gerenciamento, que nos proporcionam o tratamento de dados mais rápido e flexível, no momento, principalmente na área de geração de gráficos de monitoramento e de estatísticas. Continuando com a mesma proposta de fragmentação da base de dados e utilização de componentes especializados, interligados via um barramento de mensagens(Kafka), para cada uma das tarefas presentes no sistema de gerenciamento do SDumont.

Referências

- [1] Trygve Reenskaug. Models-views-controllers. Technical report, Xerox PARC, 1979.