UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA
Autorizada pelo Decreto Federal nº 77.496 de 27/04/76
Recredenciamento pelo Decreto nº17.228 de 25/11/2016
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
COORDENAÇÃO DE INICIAÇÃO CIENTÍFICA

XXIX SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UEFS
SEMANA NACIONAL DE CIÊNCIA E TECNOLOGIA - 2025

Desenvolvimento de infraestrutura para uso de LLMs em supercomputadores do Laboratório Nacional de Computação Científica

Letícia Teixeira Ribeiro dos Santos[1]; Jose Amancio Macedo Santos[2]
1.Letícia Teixeira Ribeiro dos Santos, PIBIC-Af/CNPq, ENGENHARIA DE COMPUTAÇÃO, e-mail: letsribeiro2@gmail.com
2.Jose Amancio Macedo Santos, DEPARTAMENTO DE TECNOLOGIA, e-mail: zeamancio@uefs.br

PALAVRAS-CHAVE: modelos de linguagem de grande porte; LLM; engenharia de software

INTRODUÇÃO

Historicamente, a Engenharia de Software (ES) concentrou-se em aspectos técnicos, priorizando a construção de sistemas funcionais e confiáveis (Seaman et al., 2025; Restrepo-Tamayo et al., 2024). Com o aumento da complexidade dos projetos, tornou-se evidente que o sucesso também depende de fatores humanos e sociais, como habilidades interpessoais, cultura e motivação, o que reforça a importância da perspectiva sociotécnica na área (Hoda, 2021; Rabelo et al., 2022; Cerqueira et al., 2024; Santana et al., 2024).

Estudos sobre fatores humanos tipicamente demandam análises qualitativas de dado. A pesquisa qualitativa nesse campo enfrenta desafios importantes, como lidar com grandes volumes de dados textuais, a subjetividade da análise e a necessidade de assegurar validade e confiabilidade. Nesse contexto, os Large Language Models (LLMs) despontam como uma tecnologia promissora para apoiar a organização, a codificação e a síntese de informações, ainda que permaneçam preocupações quanto à reprodutibilidade e à confiabilidade dos resultados (Bano et al., 2024).

Apesar da crescente adoção de ferramentas baseadas em LLMs, como Gemini e GPT, seu uso em pesquisa científica enfrenta limitações, incluindo instabilidade de APIs, falta de acesso a dados e arquiteturas, custos elevados e riscos de privacidade (NIST, 2023). Para superar essas restrições, pesquisadores têm recorrido a supercomputadores, que oferecem maior controle experimental, escalabilidade e segurança. No Brasil, destaca-se o supercomputador Santos Dumont, do Laboratório Nacional de Computação Científica (LNCC), que se trata do maior computador da América Latina para pesquisa científica, localizado fisicamente em Petrópolis, no Rio de Janeiro (LNCC, 2021). O foco deste trabalho é a configuração de um setup para execução de LLMs no supercomputador Santos Dumont do LNCC.

MATERIAL E MÉTODOS OU METODOLOGIA (ou equivalente)

Nesta pesquisa foi analisado um extenso conjunto de dados textuais sobre empatia em Engenharia de Software. Estes dados foram classificados segundo códigos associados a diferentes dimensões de empatia em um estudo proposto por Cerqueira et al., (2024). A análise começou pela API do Gemini, mas as restrições do plano gratuito levaram à busca por uma alternativa mais flexível e controlada. Optou-se, então, pela execução no supercomputador Santos Dumont do LNCC, acessado remotamente via VPN (*Virtual Private Network*) e SSH (*Secure Shell*), a partir do estudo do manual e da preparação de scripts específicos.

Após revisão da literatura, adotou-se o modelo LLaMA 3.1 70B, reconhecido por seu alto desempenho em classificação de texto (Kostina et al., 2025). Para as RQs, foram definidos diferentes esquemas de *prompting*: *one-shot* e *few-shot*. O ambiente foi configurado por meio do SLURM, com alocação de recursos de CPU, GPU e memória. Utilizou-se a ferramenta Ollama para gerenciamento do modelo, e foram desenvolvidos scripts em Python capazes de ler os arquivos das RQs, submeter os prompts à LLM e consolidar os resultados em novas planilhas.

A execução foi organizada por meio de shell scripts de submissão de jobs, nos quais se definiram parâmetros de fila e recursos computacionais. Os resultados obtidos foram comparados às classificações originais, permitindo avaliar a consistência da abordagem. Por fim, a experiência positiva levou à elaboração de um manual detalhado para auxiliar outros pesquisadores do grupo a reproduzir o processo ou adaptá-lo em etapas futuras da investigação.

RESULTADOS E/OU DISCUSSÃO (ou Análise e discussão dos resultados)

Os resultados obtidos incluíram o desenvolvimento de scripts em Bash e Python. O script em Bash foi empregado para estabelecer a comunicação com os nós de computação do Santos Dumont e realizar a submissão dos jobs, enquanto o script em Python foi utilizado para a execução da API do Ollama.

```
#!/bin/bash
#SBATCH --job-name=ollama_classification
#SBATCH --nodes=4
#SBATCH --ntasks-per-node=2
#SBATCH --cpus-per-task=24
#SBATCH --mem=256G
#SBATCH -p sequana_gpu
#SBATCH --gpus-per-node=4
#SBATCH --output=saida_ollama.out
#SBATCH --error=erro_ollama.err
#SBATCH --time=02:00:00
```

Figura 1: Detalhes dos recursos alocados para execução do modelo Llama. Incluem solicitação de memória RAM, GPUs e CPUs.

```
singularity exec --nv \
                 --bind $OLLAMA_MODELS:/root/.ollama \
                 ollama.sif \
                 ollama serve &
```

Figura 2: Utilização do Ollama para executar o modelo Llama

```
response = ollama.generate(model='llama3.3:70b', prompt=prompt_RQ1_One, options={"num_ctx": 8192})

print(f"Resposta: {response['response']}")
print("********************")
print()
analysis_responses[id] = response['response']
```

Figura 3: Trecho do código em Python utilizado para comunicação via API e armazenamento das respostas da LLM em uma estrutura de dados

Durante as primeiras execuções, alguns problemas foram identificados. Constatou-se que o tamanho do contexto da LLM, que é a quantidade máxima de *tokens* que o modelo consegue considerar simultaneamente no processamento de entradas e saídas, estava configurado em seu valor padrão de 4096 *tokens*, insuficiente para acomodar o volume de dados exigido. Essa limitação resultava em respostas com códigos de empatia inexistentes no *prompt*, pois o modelo "perdia a memória" durante o processamento. Após dobrar o tamanho do contexto através da alteração do parâmetro responsável, os resultados passaram a apresentar maior consistência e adequação com o que foi solicitado no *prompt*.

A execução final dos jobs resultou na geração de arquivos no formato .CSV, contendo as classificações atribuídas a cada *snippet* de texto. Esses arquivos foram, em seguida, consolidados e organizados em uma planilha final, de modo a facilitar a análise dos dados.

Como produto adicional, foi elaborado um manual detalhado com instruções passo a passo para a configuração do ambiente de execução de modelos de IA no supercomputador Santos Dumont. Esse documento descreve desde a preparação do setup até a submissão de tarefas, para que outros integrantes do grupo de pesquisa possam reproduzir o procedimento com autonomia. O manual pode ser encontrado no link: Manual de Utilização Santos Dumont .

CONSIDERAÇÕES FINAIS

A execução dos experimentos no supercomputador Santos Dumont demonstrou a viabilidade do uso de modelos de linguagem de grande porte, integrando scripts em Bash e Python para automatizar a submissão de jobs. Apesar das dificuldades iniciais relacionadas à configuração do tamanho do contexto da LLM, os ajustes realizados permitiram obter resultados mais alinhados com os objetivos propostos.

Além disso, a elaboração de um manual de uso do ambiente Santos Dumont para IA consolidou o conhecimento adquirido, garantindo que outros membros do grupo de

pesquisa possam replicar o processo de forma independente. Dessa forma, o trabalho não apenas alcançou êxito na análise automática dos trechos de texto, mas também deixou como contribuição prática um guia útil para futuras aplicações em projetos semelhantes.

REFERÊNCIAS

KOSTINA, Arina; DIKIAKOS, Marios D.; STEFANIDIS, Dimosthenis; PALLIS, George. 2025 [online]. *Large Language Models For Text Classification: Case Study And Comprehensive Review*. https://arxiv.org/html/2501.08457

LNCC. 2025 [online]. Supercomputador Santos Dumont. Disponível em: https://www.gov.br/lncc/pt-br/supercomputador-santos-dumont

Bano, M., Hoda, R., Zowghi, D., e Treude, C. 2024. Large language models for qualitative research in software engineering: exploring opportunities and challenges. Automated Software Engineering, 31(1):8.

Rabelo, D., Lopes, A., Mendes, W., de Souza, C., Gama, K., Monteiro, D., e Pinto, G. 2022. The role of non-technical skills in the software development market. In Proceedings of the XXXVI Brazilian Symposium on Software Engineering, p.31–40.

Cerqueira, L., Freire, S., Neves, D. F., Bastos, J. P. S., Santana, B., Spínola, R., Mendonça, M., e Santos, J. A. M. 2024. Empathy and its effects on software practitioners' well-being and mental health. IEEE Software.

Seaman, C., Hoda, R., e Feldt, R. 2025. Qualitative research methods in software engineering: Past, present, and future. IEEE Transactions on Software Engineering.

Restrepo-Tamayo, L. M., Gasca-Hurtado, G. P., e Valencia-Calvo, J. (2024). Characterizing social and human factors in software development team productivity: A system dynamics approach. IEEE Access.

Santana, B., Freire, S., Santos, J. A., e Mendonça, M. (2024). Psychological safety in the software work environment. IEEE Software.

National Institute of Standards and Technology (NIST). [s.d.] [online]. Appendix B: How AI Risks Differ from Traditional Software Risks. https://airc.nist.gov/airmf-resources/airmf/appendices/app-b-how-ai-risks-differ-from-traditional-software-risks/