

TOP ENGENHARIA DE SOFTWARE I (Replicabilidade, Reprodutibilidade e Reusabilidade)		 PUC <small>RIO</small>
CENTRO UNIVERSITÁRIO CTC DEPARTAMENTO DE INFORMÁTICA	Ano e semestre 2024.2	
INF2005	Profa. Juliana Alves Pereira	
Dia: 6ª feira	CARGA HORÁRIA TOTAL: 45h	CRÉDITOS: 3
Horário: 13 - 15 + 1 SHF	Pré-requisitos: Não há	

OBJETIVOS	<p>Esta disciplina tem como objetivo principal capacitar os alunos a entender e aplicar os conceitos de replicabilidade, reprodutibilidade e reusabilidade em estudos na área de engenharia de software. Especificamente, os objetivos são:</p> <ul style="list-style-type: none"> • Replicabilidade: Capacitar os alunos a desenvolver estudos que possam ser replicados com precisão por outros pesquisadores, garantindo que os resultados obtidos sejam consistentes e confiáveis. • Reprodutibilidade: Capacitar os alunos a desenvolver estudos cujos resultados possam ser reproduzidos por diferentes grupos de pesquisa utilizando as mesmas metodologias. • Reusabilidade: Capacitar os alunos a criar componentes e artefatos que possam ser reutilizados em diferentes contextos e estudos, promovendo a eficiência e a inovação na pesquisa em engenharia de software.
EMENTA	<p>Tópicos abordados: Introdução a engenharia de software. Conceitos fundamentais sobre o uso de soluções de inteligência artificial para engenharia de software. Introdução a boas práticas de projeto associadas ao desenvolvimento dessas soluções. Boas práticas de replicabilidade, reprodutibilidade e reusabilidade. Introdução sobre o processo de produção, avaliação, e submissão de artefatos. Introdução sobre o processo de revisão em pares. Condução da replicação de estudos. Introdução a submissão de artigos e artefatos para periódicos e eventos científicos. Boas práticas de apresentação em conferência. Revisão em pares dos trabalhos produzidos.</p> <p>Importante: o programa sofrerá ajustes conforme as expectativas e interesses dos alunos matriculados na disciplina.</p>
PROGRAMA	<p>Fase 1: Introdução da área Esta primeira fase é voltada para alunos novos que não tenham conhecimento prévio na área. Portanto, esta fase inicial será dividida em três partes: (i)</p>

	<p>Fundamentos de Engenharia de Software; (ii) Fundamentos de IA para Engenharia de Software; e (ii) Introdução dos três “R”s de Open Science: Replicabilidade, Reprodutibilidade e Reusabilidade. Por fim, iremos discutir como esses princípios científicos básicos são essenciais para replicação de experimentos em engenharia de software com a apresentação detalhada de um conjunto de artigos recentes na área que seguem tais princípios.</p> <p>Fase 2: Introdução a publicação científica O objetivo nessa fase é preparar os alunos para a produção de trabalhos acadêmicos e científicos de qualidade. Nessa fase, os alunos são guiados através dos elementos fundamentais da criação e disponibilização dos artefatos do trabalho, com ênfase na clareza, precisão e conformidade com as normas acadêmicas. O objetivo principal é capacitar os alunos a comunicar de maneira eficaz suas pesquisas e ideias, seguindo os padrões reconhecidos pela comunidade científica.</p> <ul style="list-style-type: none"> • Fornecer métodos e ferramentas para documentar de forma detalhada os processos experimentais, dados e configurações de ambiente. • Promover a utilização de práticas de documentação rigorosas e a transparência no compartilhamento de dados e código-fonte, garantindo que os experimentos possam ser reavaliados e validados por terceiros. • Ensinar técnicas para modularizar e documentar o código, bem como para criar e compartilhar bibliotecas de software, datasets e algoritmos que possam ser facilmente integrados em outros projetos. <p>Além disso, busca-se entender todo o ciclo de vida do processo de publicação, desde a escrita até a submissão e revisão, finalizando com a apresentação em conferências científicas.</p> <p>Fase 3: Formação de grupos de trabalho e seleção do estudo replicável Nesta fase, os alunos serão alocados em grupos de acordo com seu tema de interesse. Cada aluno deve selecionar um artigo potencial para replicação. Essa fase será composta por uma série de apresentações e acompanhamento das tarefas para replicação desses artigos.</p> <p>Fase 4: Apresentações, revisões em pares e submissão do trabalho Esta fase corresponde a apresentação final do trabalho com forte enfoque nos resultados e contribuições obtidas. Estas apresentações devem ter o formato equivalente a apresentações feitas em conferências. Antes das apresentações, serão realizadas revisões alternadas em pares dos trabalhos pelos colegas e professora para prover um feedback completo aos grupos durante a apresentação e antes da submissão.</p>
AVALIAÇÃO	Atividades de acompanhamento + Revisão em pares + Trabalho final

<p>BIBLIOGRAFIA PRINCIPAL</p>	<p>Abou Khalil, Z., & Zacchiroli, S. (2022, September). Software Artifact Mining in Software Engineering Conferences: A Meta-Analysis. In ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM 2022).</p> <p>Acher, M., Martin, H., Pereira, J. A., Lesoil, L., Blouin, A., Jézéquel, J. M., & Barais, O. (2022). Feature Subset Selection for Learning Huge Configuration Spaces: The case of Linux Kernel Size. In Systems and Software Product Line Conference (SPLC 2022).</p> <p>Amann, S., Beyer, S., Kevic, K., & Gall, H. (2013). Software mining studies: Goals, approaches, artifacts, and replicability. In Software Engineering (pp. 121-158). Springer, Cham.</p> <p>Brooks, A., Roper, M., Wood, M., Daly, J., & Miller, J. (2008). Replication's role in software engineering. In Guide to advanced empirical software engineering (pp. 365-379). Springer, London.</p> <p>Herbold, S., Trautsch, F., Harms, P., Herbold, V., & Grabowski, J. (2019). Experiences with replicable experiments and replication kits for software engineering research. In Advances in Computers (Vol. 113, pp. 315-344). Elsevier.</p> <p>Lewowski, T., & Madeyski, L. (2022). How far are we from reproducible research on code smell detection? A systematic literature review. Information and Software Technology, 144, 106783.</p> <p>Liu, C., Gao, C., Xia, X., Lo, D., Grundy, J., & Yang, X. (2021). On the reproducibility and replicability of deep learning in software engineering. ACM Transactions on Software Engineering and Methodology (TOSEM), 31(1), 1-46.</p> <p>Mahmood, Z., Bowes, D., Hall, T., Lane, P. C., & Petrić, J. (2018). Reproducibility and replicability of software defect prediction studies. Information and Software Technology, 99, 148-163.</p> <p>Martin, H., Acher, M., Lesoil, L., Jezequel, J. M., Khelladi, D. E., & Pereira, J. A. (2021). Transfer learning across variants and versions: The case of linux kernel size. IEEE Transactions on Software Engineering.</p> <p>Martin, H., Acher, M., Pereira, J. A., & Jézéquel, J. M. (2021). A comparison of performance specialization learning for configurable systems. In Proceedings of the 25th ACM International Systems and Software Product Line Conference-Volume A (pp. 46-57).</p> <p>Pereira, J. A., Acher, M., Martin, H., Jézéquel, J. M., Botterweck, G., & Ventresque, A. (2021). Learning software configuration spaces: A systematic literature review. Journal of Systems and Software, 182, 111044.</p> <p>Pereira, J. A., Acher, M., Martin, H., & Jézéquel, J. M. (2020). Sampling effect on performance prediction of configurable systems: A case study. In Proceedings of the ACM/SPEC International Conference on Performance Engineering (pp. 277-288).</p> <p>Pineau, Joelle, et al. Improving reproducibility in machine learning research. Journal of Machine Learning Research 22 (2021).</p>
<p>BIBLIOGRAFIA COMPLEMENTAR</p>	