

XII CONGRESSO DE INICIAÇÃO CIENTÍFICA DO IFSP ITAPETININGA

Itapetininga, 19, 20 e 21 de maio de 2026

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

Campus Itapetininga

REPRODUTIBILIDADE NOS SISTEMAS OPERACIONAIS: REPRESENTAÇÃO E INSTRUMENTAÇÃO ATRAVÉS DE UM PROTÓTIPO

Juno Cremonini Takano – PIVICT/IFSP¹

Prof. Dr. Tardelli Ronan Coelho Stekel - IFSP²

Introdução

O conceito de reprodutibilidade tem ganhado relevância no contexto dos sistemas computacionais por enfatizar a capacidade de identificar, com maior precisão, quais processos e informações levam a um determinado estado, como este estado pode ser definido em termos compreensíveis por seres humanos e como esta representação pode ser usada para sua análise e replicação (Ivie, 2018). De acordo com Syed (2022), essa abordagem traz não somente vantagens práticas ligadas à replicação e confiabilidade, mas também benefícios para a segurança dos sistemas, em particular quanto à sua integridade e autenticidade. Esse trabalho apresenta o desenvolvimento de um protótipo, com finalidade de analisar o sistema onde está sendo executado e capturar parte de seu estado atual em uma representação textual que pode ser usada para replicar este recorte em outro ambiente. Focado em sistemas operacionais baseados em Linux, são exploradas as formas como podemos identificar, segmentar, priorizar e validar as informações que definem o estado de um sistema operacional e, em especial, como representá-lo de maneira que esta representação seja útil e compreensível tanto para seres humanos como outros sistemas computacionais. O trabalho aborda, como uma das sustentações na definição do problema, a afirmação de Thompson (1984) de que só podemos confiar no código que escrevemos pessoalmente. Se interpretada literalmente, esta não é uma proposta realista na escala que o *software* atingiu 42 anos depois. O que temos hoje são diferentes formas de tentar atestar a integridade e a origem de um *software*, da fonte ao artefato binário (Syed, 2022). Isto porque, apesar de sua base aparentemente determinística e matemática, confinada aos limites físicos e arquiteturais dos circuitos que moldam todas as suas possibilidades, os sistemas computacionais têm, como uma força e também uma fraqueza, um altíssimo poder de abstração que os torna capazes de, ao mesmo tempo, representar com alta precisão ou completo engano uma determinada observação do mundo natural (Ivie, 2018). Embora capazes de representar estruturas lógicas complexas em larga escala, os sistemas de *software* enfrentam o mesmo dilema que qualquer outra ciência exata: instrumentos e métodos precisos não necessariamente levam a resultados precisos (Harris, 2009). Com este fim, dois princípios elementares e interligados da ciência moderna aparecem tanto na experimentação científica como especificamente nas ciências computacionais e no desenvolvimento de software: a falseabilidade e a reprodutibilidade (Storer, 2017). Embora este trabalho esteja focado na reprodutibilidade dos sistemas operacionais e suas configurações, a falseabilidade é um princípio fortemente conectado à reprodutibilidade e pode até ser pensado como uma das suas finalidades (McIntosh, 2019). Se não posso

¹Estudante do curso de Tecnologia em Análise e Desenvolvimento de Sistemas, IFSP – Jacareí/SP. juno.takano@ifsp.edu.br.

²Doutorado em Geofísica Espacial (INPE), IFSP – Jacareí/SP. stekel@ifsp.edu.br.

XII CONGRESSO DE INICIAÇÃO CIENTÍFICA DO IFSP ITAPETININGA

Itapetininga, 19, 20 e 21 de maio de 2026

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

Câmpus Itapetininga

reproduzir o processo que supõe-se levar a uma determinada evidência, como poderia falseá-la? E se não há como falseá-la, como poderia verificar que realmente produz o resultado esperado de forma consistente? Em termos mais concretos, podemos tomar o objeto deste trabalho: se não posso, de forma consistente, reproduzir um determinado estado de um sistema computacional, como posso testar sua integridade? E se não posso testar sua integridade, como posso afirmar que este estado é realmente aquele que descrevo e espero? Em outras palavras, se não há teste possível que demonstraria a falha, como posso atestar o sucesso? Este é um problema que se potencializa diante de outra variável: a escala, tanto vertical quanto horizontal, deste sistema (Ramesh, 2025). Podemos ter, verticalmente, um sistema que utiliza grandes volumes de dados, que possui uma vasta e complexa gama de configurações possíveis; mas ainda podemos ter, horizontalmente, múltiplas instâncias possíveis e independentes deste sistema, que sequer, necessariamente, utilizam os mesmos sistemas operacionais, entre outras infinitas possibilidades que podem diferir entre elas. Este problema de escalabilidade foi o motor para o surgimento de diferentes soluções para a operacionalização de sistemas computacionais (Ramesh, 2025), duas delas abordadas aqui sendo as de containerização e a declarabilidade como ferramentas que, isoladamente ou em conjunto, podem tornar estes sistemas mais escaláveis, gerenciáveis, observáveis e seguros. Ao mesmo tempo, isto impõe também um aumento na sua complexidade quando pensamos no sistema final como um todo, incluídos os sistemas que servem como a infraestrutura dessa arquitetura onde todo o sistema, incluído aquele que operacionaliza o sistema final, se torna objeto de um terceiro sistema que orquestra e supervisiona este processo (Ramesh, 2025). O protótipo aqui apresentado propõe possibilitar tal reprodutibilidade através da modelagem, captura e replicação destas diferentes variáveis que definem os possíveis estados de um sistema computacional a qualquer tempo, objetivando compreender como capturá-las na prática e representá-las de forma inteligível, e como replicar este estado em um ambiente distinto de maneira a poder atestar a integridade desta reprodução. O conceito de containerização é abordado e também utilizado na automatização de testes na implementação. Contudo, ele está voltado à criação de instâncias efêmeras de recursos computacionais que existem isoladamente no sistema operacional e compartilham seus recursos, o que não é o princípio nem o fim do ciclo de vida do software que assim foi operacionalizado (Ramesh, 2025). O protótipo proposto neste trabalho pode ser usado tanto dentro quanto fora de um ambiente containerizado, inspecionando as variáveis declaradas como relevantes e capturando seu estado em uma nova representação que pode ser comparada com o estado esperado. É, portanto, uma ferramenta útil para a análise destes estados *-in loco*, seja antes, durante ou após o processo de estabelecer a reprodutibilidade como parte do processo. Ele pode auxiliar na implementação de um processo reproduzível e atestar de maneira independente a integridade desta reprodução.

Objetivo

Busca-se responder a perguntas como “o estado E1 que busquei reproduzir com a tecnologia T corresponde ao estado E2 que obtive com ela?”, ou “o estado E1 na máquina M1 e o estado E2 na máquina M2 são idênticos?” ou ainda, “o estado matriz E0; sua reprodução E1 que obtive com a tecnologia T1; e sua reprodução E2 que obtive com a tecnologia T2; todos correspondem entre si?”. Para tanto, propõe-se identificar possíveis aplicações e sistemas operacionais que sejam passíveis de reprodução; integrar suas

XII CONGRESSO DE INICIAÇÃO CIENTÍFICA DO IFSP ITAPETININGA

Itapetininga, 19, 20 e 21 de maio de 2026

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

Câmpus Itapetininga

diferentes formas de persistir o estado que criam; descrever este estado em uma estrutura de dados legível; e instrumentalizar este processo em um programa capaz de replicar tal estado a partir desta representação. A hipótese aqui levantada é portanto que capturar o estado de um sistema, em parte ou no todo, como uma representação humanamente inteligível, possibilita e potencializa a reprodutibilidade deste sistema.

Metodologia

O relatório final aborda escolhas metodológicas e referencial teórico que atravessam questões práticas e conceituais, como as diferentes abordagens possíveis do Desenvolvimento Dirigido a Testes (TDD) e a evolução do conceito de confiança no contexto da computação e como ele se relaciona à necessidade de podermos atestar reprodutibilidade de um determinado programa. O protótipo demonstra como representações legíveis e técnicas criptográficas de *hashing* podem tornar a replicação e a análise da integridade destes sistemas possível e mais fácil de ser atestada. Foram utilizadas práticas de desenvolvimento ágil como o já mencionado TDD, técnicas de testes de propriedade (*property testing*), testes de mutação (*mutation testing*) e análises estáticas de cobertura de testes e importação de auditorias externas. As técnicas de testes usando valores aleatórios em um amplo espaço de possibilidades (testes de propriedade) demonstraram que o protótipo é capaz de interpretar e replicar uma ampla variedade de estados possíveis dentro dos tipos de estado suportados. Tais testes são independentes dos sistemas utilizados no desenvolvimento, e podem ser replicados em qualquer ambiente que utilize uma versão suportada do sistema operacional abordado.

Resultados

Foi desenvolvido um protótipo na linguagem de programação Rust para o sistema operacional Debian. O protótipo demonstra como podemos definir um recorte do estado deste sistema através de caminhos de arquivos, saídas de comandos e pelos dados obtidos pela interação com o gerenciador de pacotes do sistema. Este estado é armazenado em conjunto com saídas de funções criptográficas, permitindo a verificação de sua integridade. Todas estas verificações foram implementadas em um fluxo de integração contínua, capaz de executar todos os testes unitários e de integração de maneira automática, independente ao ambiente de desenvolvimento, sempre que o conteúdo do repositório de código é alterado. Estes testes automatizados permitiram ampliar a robustez e confiabilidade do protótipo, que demonstrou ser capaz de capturar e reproduzir o estado do sistema operacional de maneira consistente.

Conclusão

O protótipo aqui implementado, documentado e disponibilizado demonstrou como podemos aplicar técnicas de programação e criptografia para investigar e intervir no problema da replicação de estados em sistemas computacionais, cujas formas de reprodução e representação muitas vezes são desconhecidas ou ausentes. O protótipo trouxe resultados positivos na solução do problema abordado e pôde ser aplicado para reproduzir o próprio ambiente de testes em um dos contêineres onde foi implantado e automatizado. Todo o código fonte necessário para compilar e compreender o protótipo está publicamente disponível e licenciado de maneira a permitir sua livre reprodução e modificação. Estas premissas, que possibilitam também a reprodutibilidade do próprio protótipo, foram

XII CONGRESSO DE INICIAÇÃO CIENTÍFICA DO IFSP ITAPETININGA

Itapetininga, 19, 20 e 21 de maio de 2026

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

Câmpus Itapetininga

norteadoras durante todo o desenvolvimento, e seu resultado final demonstra como os princípios objetivados foram aplicados ao protótipo ao mesmo tempo que são demonstrados por ele.

Referências

IVIE, Peter; THAIN, Douglas. Reproducibility in Scientific Computing. *ACM Computing Surveys*, [S. l.], v. 51, n. 3, p. 1–36, 2018. DOI: 10.1145/3186266. Disponível em: <http://dx.doi.org/10.1145/3186266>.

SYED, Naeem Firdous; SHAH, Syed W.; TRUJILLO-RASUA, Rolando; DOSS, Robin. Traceability in supply chains: A Cyber security analysis. *Computers & Security*, [S. l.], v. 112, p. 102536, 2022. DOI: 10.1016/j.cose.2021.102536. Disponível em: <http://dx.doi.org/10.1016/j.cose.2021.102536>.

THOMPSON, Ken. Reflections on trusting trust. *Communications of the ACM*, [S. l.], v. 27, n. 8, p. 761–763, 1984. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/358198.358210. Disponível em: <https://dl.acm.org/doi/10.1145/358198.358210>. Acesso em: 3 abr. 2026.

HARRIS, Edward F.; SMITH, Richard N. Accounting for measurement error: A critical but often overlooked process. *Archives of Oral Biology*, [S. l.], v. 54, p. S107–S117, 2009. DOI: 10.1016/j.archoralbio.2008.04.010. Disponível em: <http://dx.doi.org/10.1016/j.archoralbio.2008.04.010>.

STORER, Tim. Bridging the Chasm. *ACM Computing Surveys*, [S. l.], v. 50, n. 4, p. 1–32, 2017. DOI: 10.1145/3084225. Disponível em: <http://dx.doi.org/10.1145/3084225>.

MCINTOSH, Leslie D.; VITALE, Cynthia Hudson; JUEHNE, Anthony; MOTHERSHEAD, Sasha; SUMNER, Josh; HAYNES, Leah; SCIENCE, Digital. Making Science Better: Reproducibility, Falsifiability and the Scientific Method. [s.l.], *Digital Science*, 2019. DOI: 10.6084/M9.FIGSHARE.9633158. Disponível em: <https://doi.org/10.6084/m9.figshare.9633158>.

RAMESH, G. et al. A Comprehensive Review on Scaling Machine Learning Workflows Using Cloud Technologies and DevOps. *IEEE Access*, [S. l.], v. 13, p. 148559–148594, 2025. DOI: 10.1109/access.2025.3599281. Disponível em: <http://dx.doi.org/10.1109/ACCESS.2025.3599281>.