

Understanding Sustainability and FAIRness of Research Software

Daniela Feitosa*
Christina von Flach*

dfeitosa@ufba.br, flach@ufba.br
Federal University of Bahia, Institute of Computing
Salvador, Bahia, Brazil

Joenio Costa*
joenio@gmail.com

Université Gustave Eiffel, Laboratoire Interdisciplinaire
Sciences Innovations Sociétés (LISIS)
France

RESUMO

The increasing use of research software has urged the scientific community to discuss its sustainability, FAIRness, and ability to support the reproduction of studies by independent researchers. In this paper, we present the results of an exploratory study conducted with a research group in Applied Physics, whose researchers historically developed most of their research software. We analyzed the sustainability and FAIRness of one research software developed by the group and reported the results. Our study allowed us to broaden our understanding of evaluating sustainability and FAIRness, and related challenges, to support future evaluations of research software.

KEYWORDS

Research software, Sustainability, FAIRness, Assessment model

1 INTRODUÇÃO

Software de Pesquisa refere-se ao software desenvolvido em universidades e laboratórios de pesquisa como parte ou para apoiar suas pesquisas. O uso crescente de software de pesquisa tem instigado a comunidade científica a discutir sua sustentabilidade, *FAIRness*¹ e capacidade de apoiar a reprodução de estudos de pesquisadores independentes.

Sustentabilidade de software tem a ver com a capacidade do software de sobreviver e continuar a ser suportado ao longo do tempo, implicando na longevidade e capacidade de manutenção do software e de suas comunidades [5]. O software de pesquisa sustentável promove a reprodutibilidade no longo prazo.

Reprodutibilidade na pesquisa busca garantir que qualquer pessoa com *acesso* aos dados e ao software possa reproduzir os resultados, tanto para verificá-los quanto para desenvolver nova pesquisa com base nos mesmos [7]. Neste contexto, software de pesquisa também se preocupa com abertura (*openness*) e os princípios FAIR (Findable, Accessible, Interoperable, Reusable) usados para Dados Abertos [6]. Considerando que o software de pesquisa é um artefato digital da pesquisa, ele deve ser facilmente localizável, acessível, interoperável e reutilizável [2].

A sustentabilidade e *FAIRness* de software de pesquisa são essenciais para a confiabilidade e a reprodutibilidade pesquisa acadêmica. É relevante entender e descrever a prática atual em universidades, laboratórios de pesquisa e na indústria sobre o desenvolvimento de software de pesquisa e avaliar sua sustentabilidade e *FAIRness* para pesquisas de longo prazo [1, 5].

Apesar de sua importância, esses aspectos são frequentemente negligenciados em projetos de pesquisa, muitas vezes devido à falta de conscientização e conhecimento dos pesquisadores sobre princípios FAIR e boas práticas para o desenvolvimento de software de pesquisa sustentável. Tais informações raramente estão disponíveis, mesmo para projetos de pesquisa financiados por agências e fundações de pesquisa [1].

O objetivo geral de nossa pesquisa é realizar uma investigação abrangente com grupos de pesquisa da Universidade Federal da Bahia (UFBA) para caracterizar a prática atual de desenvolvimento de software de pesquisa. A *avaliação da sustentabilidade e aderência aos princípios FAIR (FAIRness)* do software de pesquisa desenvolvido por grupos de pesquisa é um dos objetivos específicos da pesquisa. Nas interações com diferentes grupos de pesquisa, quatro atividades serão realizadas:

- (1) Apresentação de palestra com uma visão geral sobre Ciência Aberta, software de pesquisa, sustentabilidade e *FAIRness*;
- (2) Entrevista com o líder para coletar informações sobre seu conhecimento de Ciência Aberta, software de pesquisa e práticas de engenharia de software, motivação e desafios para desenvolver software de pesquisa e possíveis esforços para torná-lo sustentável;
- (3) Avaliação, com base em critérios e diretrizes de bem definidos, da sustentabilidade e *FAIRness* de pelo menos um software de pesquisa desenvolvido pelo grupo de pesquisa; e
- (4) Apresentação de relatório com resultados da avaliação ao investigador responsável para receber o seu feedback.

Neste artigo, apresentamos os resultados de um estudo piloto para avaliação de um software de pesquisa desenvolvido por um grupo de Física Aplicada, recomendado pelo líder de grupo de pesquisa durante sua entrevista. A Seção 2 apresenta trabalhos relacionados, a Seção 3 apresenta os resultados da avaliação de sustentabilidade e *FAIRness* do software de pesquisa recomendado, e a Seção 4 apresenta as conclusões do artigo.

2 ABORDAGENS RELACIONADAS

2.1 Avaliação da Sustentabilidade

O Software Sustainability Institute (SSI) fornece um serviço de avaliação online que pode ser usado para avaliar o software de pesquisa. A avaliação refere-se a verificar se o projeto de software apresenta qualidades esperadas de um software sustentável (suporte, documentação, planos para o futuro, disponibilidade do software, estrutura do código-fonte, uso de padrões abertos, teste, portabilidade, comunidade, política de contribuidores, identidade e licenças)²,

*Os autores contribuíram com a pesquisa de forma equilibrada.

¹Não traduzimos o termo "FAIRness" para não perder o alinhamento natural com o acrônimo FAIR.

²O questionário SSI Online Sustainability Evaluation pode ser acessado em <https://www.software.ac.uk/resources/online-sustainability-evaluation>

por meio de perguntas simples como *Seu site e documentação fornecem uma visão geral clara e de alto nível do seu software?*, *Seu software está disponível como um pacote que pode ser implantado sem compilar?*, *Você publica seu histórico de lançamentos, por exemplo, dados de lançamento, números de versão, principais recursos de cada lançamento, etc. em seu site ou em sua documentação?*, *Você tem um conjunto de testes automatizados para seu software?* ou *Você tem uma citação recomendada para seu software?*.

Em nossa pesquisa, utilizamos alguns critérios do SSI, combinamos outros em apenas um critério e desconsideramos outros, propondo um modelo com 16 práticas, organizadas em cinco grupos (ver Seção 3) para um estudo piloto sobre a sustentabilidade do software de pesquisa desenvolvido na universidade.

2.2 Aderência aos princípios FAIR (FAIRness)

Lamprecht e colegas apresentam um resumo dos princípios FAIR (Findable, Accessible, Interoperable, Reusable) adaptados para software de pesquisa [2] e sobre como eles se relacionam com os Princípios Orientadores FAIR para Dados [6]. Cada princípio FAIR para dados foi reformulado para ajustá-lo ao software aberto, estendido para cobrir um escopo mais amplo, reinterpretado para corresponder a um contexto diferente, descartado porque não se aplica a software ou reescrito para software [3]. O conjunto de princípios FAIR para software de pesquisa pode servir como base para a proposta de novos modelos e métricas que mostrem a usuários e desenvolvedores quão “FAIR” é o seu software [2, 3].

Utilizamos integralmente o material mais recente para avaliação de FAIRness de software de pesquisa [2, 3].

2.3 Guias e Recomendações

O SSI fornece guias para conscientizar pesquisadores, especialmente abordando questões sobre licenças de dados e software, para lidar com citação, uso, créditos e contribuição para o software de outros desenvolvedores³ e guias para desenvolvedores com dicas e orientações sobre ferramentas e técnicas para tornar o software aberto, citável, robusto, legível, verificável, reutilizável e fácil de contribuir⁴. *The Carpentries*⁵ é uma organização que oferece treinamentos sobre conceitos e habilidades computacionais para conduzir pesquisas eficientes, abertas e reprodutíveis, com recursos relacionados a software de pesquisa.

3 AVALIAÇÃO DO SOFTWARE MOSYN

Um estudo exploratório foi conduzido com um grupo de pesquisa da área de Física Aplicada, escolhido por conveniência, com o objetivo de identificar problemas gerais do software de pesquisa desenvolvido pelo grupo, fazer uma caracterização inicial da sustentabilidade e FAIRness de um software de pesquisa e recomendar boas práticas para torná-lo mais sustentável e aberto. A avaliação do software de pesquisa foi feita manualmente, por meio da inspeção de informações disponíveis no repositório público do software.

Realizamos uma entrevista com um pesquisador sênior e líder do grupo de pesquisa da área, com mais de 30 anos de experiência. O grupo conta com pesquisadores com formação acadêmica

diversificada, com *expertise* em suas respectivas áreas de atuação, e não em engenharia de software e suas práticas. No final da entrevista, solicitamos que o pesquisador indicasse projetos de software de pesquisa para avaliação de sustentabilidade e FAIRness. Para a primeira avaliação, escolhemos o software MoSyn⁶, um aplicativo para análise de grafos variáveis no tempo. Os resultados da avaliação compõem um relatório técnico, com sugestão de práticas e melhorias, a ser encaminhado para o pesquisador.

3.1 Avaliação da Sustentabilidade de MoSyn

O software de pesquisa MoSyn é um aplicativo baseado em MATLAB, projetado para a análise de grafos variantes no tempo (TVGs) e suas medidas associadas. A ferramenta fornece uma estrutura modular com várias classes e funções para lidar com diferentes aspectos da análise, como configuração, recursos gráficos e gerenciamento de projetos. Na avaliação de sustentabilidade, consideramos 16 práticas (P1–P16), e as organizamos em cinco grupos:

- Básicas. (P1) Hospedagem do software, (P2) Controle de versão, (P3) Licenças de software e (P4) Registro do software.
- Organização. (P5) Estrutura de arquivos, (P6) Padronização e (P7) Documentação.
- Qualidade. (P8) Teste automatizado e (P9) Revisão de código.
- Gerência. (P10) Issue Tracker e (P11) Automatização de tarefas, (P12) CI/CD e (P13) Lançamento de versões.
- Reconhecimento. (P14) Comunidade, (P15) Divulgação e (P16) Citação de software.

A Tabela 1 apresenta um resumo da avaliação da sustentabilidade do software MoSyn, indicando se o mesmo atende, não atende ou atende parcialmente (P) às práticas, e comentários sobre como foram usadas. Na sequência, apresentamos trechos do relatório de avaliação do software MoSyn.

3.1.1 Práticas Básicas. O MoSyn esteve hospedado em um repositório público desde o início de seu desenvolvimento (P1). Apesar de sua hospedagem no GitHub e uso das facilidades da plataforma, um backup periódico tem sido realizado em um dispositivo de armazenamento do grupo de pesquisa, mantido nas instalações do grupo. No GitHub, a identidade do software é clara e única: o nome público do software é MoSyn. Apesar de ter um arquivo README.md, não há uma descrição do projeto que facilite sua indexação por mecanismos de busca.

O MoSyn usa o suporte para controle de versão do GitHub (P2). Inicialmente, seu repositório não apresentava a licença de software (P3). Após a entrevista com o pesquisador sênior do grupo, houve um *commit* no repositório para a inclusão do arquivo *LICENSE*, descrevendo a licença escolhida, porém sem deixar claro se as permissões se aplicam a qualquer arquivo fonte do repositório. A licença atribuída ao MoSyn é a MIT⁷, uma licença usada por projetos de software livre e por software proprietário. Quanto ao registro do software (P4), o repositório não menciona se há um DOI associado a ele. Não encontramos uma referência para o MoSyn no Zenodo.

3.1.2 Organização do Projeto. O repositório do MoSyn tem uma estrutura de arquivos (P5) bem definida e usa nomes auto-explicativos para pastas e arquivos que facilitam a compreensão do propósito e

³<https://www.software.ac.uk/resources/guides/guides-researchers>

⁴<https://www.software.ac.uk/resources/guides/guides-developers>

⁵<https://carpentries.org/>

⁶<https://github.com/mpnetto/MoSyn>

⁷<https://opensource.org/licenses/mit/>

Tabela 1: Sustentabilidade do software MoSyn.

P	Descrição	S/N/P	Comentário
P1	O projeto está hospedado em um repositório público	Sim	O software está disponibilizado em um repositório público no GitHub
P2	O software implementa controle de versão	Sim	O controle de versão é implementado por utilizar o GitHub como plataforma de hospedagem
P3	Uma licença de software foi adotada	P	Um arquivo declarando a licença MIT. Porém não está claro se as permissões se aplicam a qualquer arquivo fonte no repositório
P4	O software está publicado formalmente e apresentam um DOI	Não	O repositório não menciona um DOI associado a ele mas pode ser encontrado no GitHub pelo nome
P5	A estrutura de arquivos comunica a finalidade dos elementos do projeto	Sim	A estrutura de pastas está organizada de forma descritiva e permite inferir o conteúdo
P6	Adota formatos de dados e interfaces comuns	Sim	Apesar de não haver documentação explícita, o software utiliza o formato de entrada e saída que facilita a integração com o MATLAB
P7	A documentação apresenta uma visão geral sobre o software	P	O projeto utiliza o <i>GitHub pages</i> mas as informações estão incompletas e algumas URLs direcionam para um destino não válido.
P8	O software implementa testes	Não	Não há testes automatizados para o software
P9	O código é revisado antes de ser incorporado ao código	P	Todos os <i>pull requests</i> listados no repositórios foram aprovados pelo próprio autor, sugerindo que não houve revisão de código por outra pessoa
P10	Disponibiliza e usa <i>issue tracker</i>	Sim	O projeto aproveita a funcionalidade de rastreamento de <i>bugs</i> e tarefas disponível no GitHub
P11	As tarefas repetitivas são automatizadas	Não	Não encontramos tarefas automatizadas no projeto
P12	Há integração e implantação contínua	Não	Não há integração contínua. Por ser um plugin instalado manualmente no MATLAB, a implantação contínua não é viável
P13	O software faz lançamento de versões	Não	O projeto não utiliza a funcionalidade de lançamento de versões no repositório do GitHub
P14	Há evidência de uma comunidade (presente ou futuro)	Não	Há apenas um desenvolvedor como autor
P15	O software é divulgado em eventos científicos	Não	Não encontramos divulgação em eventos científicos
P16	O software é citado em publicações científicas	Não	Não encontramos citação do software em publicações.

utilidade do software de pesquisa. O MoSyn também se preocupa com padronização (P6) visto que depende do software MATLAB⁸ – uma plataforma paga para programação e computação numérica usada por engenheiros e cientistas para analisar dados, desenvolver algoritmos e criar modelos. O MoSyn usa formatos de entrada e saída que facilitam a integração com o MATLAB.

No MoSyn, a preocupação com a documentação do software (P7) é incipiente e direcionada apenas para seus usuários. Inicialmente, o repositório não tinha qualquer documentação. Quatro meses após a entrevista com o pesquisador, um arquivo *README.md* foi adicionado ao repositório com descrição de funcionalidades do MoSyn, informações sobre utilização e contribuição e licença do software (P7). Os autores não estão listados em um arquivo, mas é possível identificar pessoas que contribuíram com o software, inspecionando as mudanças registradas pelo sistema de controle de versão (*commits*). Há um website em construção, usando *GitHub Pages*, para divulgação de informações sobre o projeto de pesquisa e o software.

3.1.3 Qualidade. O *MATLAB Test*⁹ é um conjunto de ferramentas para o desenvolvimento, gerenciamento, análise e teste de aplicações MATLAB. Entretanto, o software de pesquisa MoSyn ainda não implementa testes de software automatizados (P8). O uso das ferramentas de teste do *MATLAB Test* requer assinatura paga.

Considerando que o MoSyn está no GitHub, é possível realizar a revisão de código (P9) colaborativamente, usando a infraestrutura oferecida pela plataforma. Porém, até o dia da avaliação do

software de pesquisa, todos os pedidos de mudança no código (*pull requests*) listados no repositório foram aprovados pelo próprio autor, sugerindo que não houve revisão de código por pares.

3.1.4 Gerência. O software MoSyn está hospedado no GitHub e conta com o seu rastreador de tarefas e *bugs* nativo (P10). O *GitHub Docs*¹⁰ apresenta uma breve introdução sobre como reportar um problema usando o rastreador de tarefas. Entretanto, o rastreador do GitHub ainda não foi utilizado no projeto MoSyn. Não foram encontradas tarefas automatizadas ou indício de automatização de tarefas (P11). Por fim, não há suporte para integração e implantação contínuas (P12). Por ser um *plugin* instalado manualmente pelo usuário no MATLAB, tais práticas não são viáveis.

O projeto não segue a prática de lançamento de versões (P13) no repositório do GitHub. As funcionalidades de 'Releases' e 'Tags' que facilitariam a referência e a descrição das funcionalidades e *bugs* incluídos em versões específicas não são usadas. Mesmo sem lançamentos oficiais de versões, para citar o software de pesquisa em um artigo, os autores ainda podem fazer referência a um *commit* específico no repositório do MoSyn. No histórico do projeto, vimos que um *commit* com a mensagem "mosyn 2.0" foi incorporado após a entrevista. Tal mensagem pode indicar a intenção de realizar uma alteração significativa no projeto e associar um número de versão.

3.1.5 Reconhecimento. O MoSyn não possui uma comunidade (P14) aparente no GitHub. Apenas o dono do repositório submeteu *commits* e *pull requests* no projeto. Não há outros usuários que marcaram

⁸<https://www.mathworks.com/products/matlab.html>

⁹<https://www.mathworks.com/products/matlab-test.html>

¹⁰<https://docs.github.com/pt>

o projeto como favorito ou que fizeram uma cópia independente (*fork*). Não encontramos artigos ou outras formas de divulgação do software de pesquisa em eventos científicos (P15). O software MoSyn / MATLAB é mencionado e foi usado em uma dissertação de mestrado na área de Saúde para extrair índices de redes funcionais cerebrais (RFC) para análise estatística [4]. Entretanto, não encontramos no texto da dissertação um identificador ou referência para o software ou para a versão usada. Também não encontramos citação do software em publicações e na dissertação mencionada (P16), nem informações sobre como o software deveria ser citado.

3.2 Avaliação de FAIRness do software MoSyn

A Tabela 2 apresenta uma avaliação preliminar de FAIRness do MoSyn. Cada linha da tabela apresenta um princípio, sua descrição, indicação se o software atende (S), atende parcialmente (P) ou não atende (N) ao princípio, e uma justificativa, se procedente. A seguir, discutimos como software de pesquisa incorporou os princípios FAIR.

3.2.1 (F): O software e seus metadados associados são facilmente encontrados tanto por humanos quanto por máquinas. O software MoSyn está hospedado com uma identidade clara e única no GitHub, não globalmente e o repositório não garante a persistência do identificador se o software for movido, por exemplo (F1). Os componentes do software, como classes e bibliotecas, apresentam identificadores distintos (F1.1). Cada versão do software pode ser unicamente identificada por um *hash de commit* e, a partir dele, é possível recuperar os metadados da versão específica (F1.2). O software apresenta metadados, mas não descreve as dependências, informações detalhadas sobre utilização e configuração e como deve ser a entrada e saída de arquivos (F2). Ao analisar os metadados a partir de um commit, é possível verificar qual versão específica do software o metadado está se referindo (F3). Apesar de incluir um arquivo README com informações do projeto, não há uma descrição na configuração do projeto que facilite a indexação nos mecanismos de busca (F4).

3.2.2 (A) O software e seus metadados podem ser obtidos através de protocolos padronizados. O software MoSyn pode ser obtido a partir do repositório do projeto no GitHub (A1). Não há restrições para baixar o código-fonte, como taxas ou custos relacionados à licença (A1.1). É possível configurar o repositório para ser acessado apenas por pessoas autorizadas (A1.2). Os metadados estão descritos em arquivos no repositório, então ele não seriam acessíveis caso o repositório do software não esteja mais disponíveis (A2).

3.2.3 (I) O software interopera com outros software trocando dados e/ou metadados através da interação via interface de programação de aplicativos (APIs), descrito por meio de padrões. O software MoSyn é uma aplicação para MATLAB e, portanto, interopera seguindo seu padrão, mas a forma como a interação ocorre não está explicitamente descrito (I1). O repositório menciona o MATLAB como pré-requisito para executar a aplicação e inclui agradecimentos pela utilização de bibliotecas e recursos externos e recursos, mas não inclui referências qualificadas, como websites ou os nomes das bibliotecas e recursos externos utilizados (I2).

3.2.4 (R) O software é tanto utilizável (pode ser executado) quanto reutilizável (pode ser compreendido, modificado, aprimorado ou incorporado a outros softwares). O software MoSyn não disponibiliza muitas informações descrevendo como reutilizar o software (R1). O software está licenciado sob a licença de código aberto MIT, que permite a reutilização, mas não deixa claro se todas as partes do software seguem a mesma licença (R1.1). A partir dos commits e da lista de contribuidores do projeto no GitHub é possível saber quais pessoas contribuíram com o software, mas não há informações explícitas sobre como o software foi desenvolvido ou quais foram as intenções originais (R1.2). Os nomes e informações sobre as bibliotecas utilizadas não são mencionadas na documentação (R2). A comunidade MATLAB recomenda que sejam seguidas as melhores práticas de desenvolvimento de software em relação a correção, clareza e generalização e o software, em geral, atende a esses padrões (R3).

4 CONCLUSÕES

O estudo piloto mostrou a viabilidade de uma avaliação da sustentabilidade e FAIRness de um software de pesquisa desenvolvido por um grupo de pesquisa da área de Física. Entretanto, a avaliação do software de pesquisa foi feita manualmente, por meio da inspeção de informações disponíveis no repositório público do software. Neste contexto, estamos trabalhando no suporte automatizado para a avaliação e recomendação de melhorias, desde que artefatos digitais e código do software de pesquisa estejam disponíveis.

Esperamos que os resultados preliminares de nosso estudo piloto inspirem os pesquisadores a refletir sobre a sustentabilidade e FAIRness de seu software de pesquisa, práticas de engenharia de software que podem ajudá-los a alcançá-los, desenvolver ou refinar modelos de avaliação e contribuir para sua adoção.

ARTEFATOS DA PESQUISA.

Os artefatos ou links para artefatos de pesquisa estão disponíveis no próprio artigo.

REFERÊNCIAS

- [1] J.C. Carver, N. Weber, K. Ram, S. Gesing, and D.S. Katz. 2022. A survey of the state of the practice for research software in the United States. *PeerJ Comput. Science* (2022). <https://doi.org/10.7717/peerj-cs.963>
- [2] N.P. Chue Hong, D.S. Katz, M. Barker, A. Lamprecht, C. Martinez, F.E. Psomopoulos, J. Harrow, L.J. Castro, M. Gruenpeter, P.A. Martinez, and others. 2021. *FAIR principles for research software (FAIR4RS principles)*. Technical Report. ReSA. <https://doi.org/10.15497/RDA00065>
- [3] Anna-Lena Lamprecht et al. 2020. Towards FAIR Principles for Research Software. *Data Science* 3, 1 (2020), 37–59.
- [4] Thaise Grazielle Lima de Oliveira Toutain. 2019. *Avaliação da estabilidade cerebral e conexões intra e inter-hemisféricas na modulação afetiva da dor*. Master's thesis. Universidade Federal da Bahia, (PPGIOS), Brasil.
- [5] C. Venters, S. Akinli kocak, S. Betz, I. Brooks, R. Capilla, R. Chitchyan, L. Duboc, R. Heldal, A. Moreira, S. Oyedeji, B. Penzenstadler, and J. Porras. 2021. Software Sustainability: Beyond the Tower of Babel. <https://doi.org/10.6084/m9.figshare.14370611.v1>
- [6] Mark D. Wilkinson et al. 2016. The FAIR Guiding Principles for Scientific Data Management and Stewardship. *Scientific Data* 3 (2016), 160018. <https://doi.org/10.1038/sdata.2016.18>
- [7] G. Wilson, J. Bryan, K. Cranston, J. Kitzes, L. Nederbragt, and T.K. Teal. 2017. Good enough practices in scientific computing. *PLOS Computational Biology* 13, 6 (06 2017), 1–20. <https://doi.org/10.1371/journal.pcbi.1005510>

Tabela 2: FAIRness no Software MoSyn.

	Localizável Findable	S/N/P	Comentário
F:	O software e seus metadados associados são facilmente encontrados tanto por humanos quanto por máquinas.	<i>P</i>	
F1	O software recebe um identificador globalmente único e persistente.	<i>P</i>	O software possui uma identidade única apenas no GitHub e não é garantida a persistência.
F1.1	Aos componentes do software que representam diferentes níveis de granularidade são atribuídos identificadores distintos.	<i>S</i>	
F1.2	As diferentes versões do software recebem identificadores distintos.	<i>S</i>	
F2	O software é descrito com metadados detalhados.	<i>P</i>	
F3	Os metadados contêm de forma clara e explícita o identificador do software que descrevem.	<i>P</i>	Os metadados não apresentam de forma clara, mas é possível obter a partir do <i>commit</i>
F4	Os metadados seguem os princípios FAIR, são pesquisáveis e indexáveis.	<i>P</i>	Há informações nos arquivos do repositório, mas não há descrição na configuração do projeto para facilitar indexação
	Acessível		
A:	O software e seus metadados podem ser obtidos através de protocolos padronizados.	<i>P</i>	
A1	O software é pode ser obtido por meio de seu identificador utilizando um protocolo de comunicação padronizado.	<i>S</i>	
A1.1	O protocolo é aberto, gratuito e universalmente implementável.	<i>S</i>	
A1.2	O protocolo permite procedimentos de autenticação e autorização, quando necessário.	<i>S</i>	
A2	Os metadados são acessíveis, mesmo quando o software não está mais disponível.	<i>S</i>	Caso o software não esteja mais disponível, os metadados também ficarão inacessíveis
	Interoperável		
I:	O software interopera com outros softwares, trocando dados e/ou metadados através da interação via APIs, descrito por meio de padrões.	<i>P</i>	
I1	O software lê, escreve e troca dados de acordo com padrões da comunidade relacionados ao domínio.	<i>P</i>	Não está explícito como a troca de dados ocorre com o MATLAB
I2	O software inclui referências qualificadas a outros objetos.	<i>P</i>	As referências se resumem a menção sobre utilização, sem citar nomes, websites ou detalhes dos outros objetos
	Reusável		
R:	O software é utilizável (pode ser executado) e reutilizável (pode ser compreendido, modificado, aprimorado ou incorporado a outros softwares).	<i>P</i>	
R1	O software é descrito com uma variedade de atributos precisos e relevantes.	<i>P</i>	Não há muitas informações sobre atributos e como reutilizar o software
R1.1	É atribuída uma licença clara e acessível ao software.	<i>S</i>	<i>MIT License.</i>
R1.2	O software possui informações detalhadas de procedência.	<i>P</i>	Não há informações sobre como o software foi desenvolvido ou quais foram as intenções originais
R2	O software inclui referências qualificadas a outros software.	<i>P</i>	Não há informações relevantes sobre dependências.
R3	O software atende aos padrões relevantes da comunidade do domínio.	<i>S</i>	