



Software Livre:

Pré-requisito para a Ciência Aberta

Paulo Meirelles
Centro de Matemática, Computação e Cognição
Universidade Federal do ABC, São Paulo



Paulo Meirelles, Nelson Lago, Vanessa Sabino, Fabio Kon e Christina von Flach
Creative Commons Atribuição 4.0 Internacional (CC BY 4.0)



ARTIGO

CIÊNCIA ABERTA - COLABORAÇÃO SEM BARREIRAS PARA O AVANÇO DO CONHECIMENTO

POR

Claudia Rauer Medeiros
crauer@unicamp.br

Ciência Aberta – introdução e implementação computacional

O que é Ciência Aberta? Vamos separar a "especificação" da "implementação". Embora não haja uma definição fixa para o termo "Ciência Aberta", ele costuma ser usado para denotar o conjunto de políticas, iniciativas e ações

para disseminar e compartilhar conhecimento, geralmente por meio digital, para que todos os resultados associados à pesquisa científica se tornem acessíveis a todos. O conceito principal desta especificação é colaboração sem barreiras – geográficas, temporais, culturais, sócio-econômicas ou políticas.

A implementação exige pesquisa e desenvolvimento em Computação.



ARTIGO

SOFTWARE LIVRE: PRÉ-REQUISITO PARA A CIÊNCIA ABERTA

POR

Christina von Flach e Fábio Kon
flach@ufba.br, kon@ime.usp.br

A ciência sempre se pautou pela disseminação do conhecimento, com destaque para a reprodutibilidade de experimentos. É claro que há limites práticos – e mesmo éticos – para o que pode ser compartilhado e reproduzido, mas expandir esses limites é um objetivo de interesse para cientistas de todas as áreas. Esse é o foco do movimento pela ciência aberta, que tem tido bastante repercussão nos últimos anos.

A Ciência Aberta requer que as ferramentas e os instrumentos necessários para a prática científica estejam dis-

poníveis para todos, de modo que os experimentos possam ser reproduzidos e os resultados verificados por terceiros. No Século 21, o software consolidou-se como onipresente no conjunto de ferramentas usado por pesquisadores das Ciências Exatas e Biomédicas e tem uso crescente nas Ciências Sociais e Humanas. Software é usado para limpeza, processamento e visualização de dados, bem como para criar modelos e realizar previsões. Algoritmos especializados são codificados na forma de bibliotecas, scripts e metadados. Sem compartilhar todos esses artefatos, é muito difícil e custoso reproduzir a pesquisa científica e validar sua correção.



CHRISTINA VON FLACH é Professora Associada do Instituto de Computação da Universidade Federal da Bahia. Atua na área de Engenharia de Software tendo publicado mais de 100 artigos científicos em congressos e periódicos. Recentemente tem trabalhado com temas relacionados à Educação em Engenharia de Software, Sustentabilidade e Aspectos Sócio-técnicos de Ecossistemas de Software, e Ciência Aberta.



FABIO KON é Professor Titular de Ciência da Computação na Universidade de São Paulo. Atua na área de Sistemas Distribuídos, Engenharia de Software e Ciência de Dados tendo publicado cerca de 200 artigos científicos em congressos e periódicos. Recentemente tem trabalhado na área de Cidades Inteligentes e gostaria de ter mais tempo para tocar seu vibrafone.

Software Livre

[FSF 2022]

*"Software Livre significa software que **respeita a liberdade dos usuários e a comunidade**"*

...

*"**acesso ao código-fonte** é condição necessária para o software livre"*

...

*"usuários têm a liberdade de **executar, copiar, distribuir, estudar, alterar e melhorar o software.**"*

Ciência Aberta

[FOSTER 2022]

*"Ciência Aberta é a prática científica que **permite a colaboração e contribuição de terceiros**"*

...

*"com **disponibilização de dados, anotações, processos de pesquisa**"*

...

*"sob condições que permitem o **reuso, redistribuição e reprodução da pesquisa e seus dados e métodos subjacentes.**"*

O que é Software Livre?

O que é software livre?

- Preço?
- Licença?
- Engenharia de software?
- Movimento social?
- Hobby?
- Luta contra a Microsoft?
- Realidade de mercado?

O que é software livre?

- A *Free Software Foundation* define como:
 - Liberdade para **executar** o programa
 - Liberdade para **estudar e modificar** o programa
 - Liberdade para **redistribuir** o programa
 - Liberdade para **melhorar e redistribuir as melhorias** ao programa

O que aconteceu?

- 1976 - Bill Gates e sua “carta aberta aos hobbistas”
 - Software para microcomputadores tem grande potencial comercial
 - Não é possível financiar desenvolvimento fora do modelo de prateleira

O que aconteceu?

- 1984 - Richard Stallman lança o projeto GNU
 - uso de software restrito não é ético
- 1985 - A Free Software Foundation é fundada por Richard Stallman
 - dedica-se a eliminação de restrições sobre a cópia, estudo e modificação de programas de computadores
- bandeiras do movimento do software livre, em essência

O que aconteceu?

- 1991 - É lançamento/surgimento do Linux
 - Linux refere-se aos sistemas operacionais que utilizam o Kernel Linux
 - O núcleo foi desenvolvido por Linus Torvalds, inspirado no sistema Minix, como projeto informal, apenas para estudo

O que aconteceu?

- 1995 – Boom da Internet
 - tem início a comercialização em escala do Linux, por exemplo, o Red Hat Linux.

O que aconteceu?

- 1997 – Eric Raymond apresenta “a Catedral e o Bazar”
 - vantagens técnicas do software livre
 - mecanismos de funcionamento do desenvolvimento descentralizado

O que aconteceu?

- 1998 – Netscape libera o código fonte do navegador Mozilla sob licença livre
- 1998 – Eric Raymond, Linus Torvalds e outros lançam o movimento Open Source
 - software livre por razões técnicas
 - expressão open source ao invés de free software

O que aconteceu?

- 1999 – Sourceforge é lançado
- 2000 – OpenOffice é lançado
- 2001 – IBM investe US\$1bi no Linux

Direitos autorais e software

Direitos autorais e software

- Propriedade física x Propriedade intelectual
 - objetos \neq ideias
- Desenvolvimento do conhecimento depende do compartilhamento de ideias
- Desenvolver novas ideias depende também de dinheiro

Direitos autorais e software

- A solução de compromisso dos direitos de autor procurava **equilibrar o interesse privado e o interesse público**
- Hoje em dia, mudanças na legislação e na tecnologia destruíram esse equilíbrio

Direitos autorais e software

- Quem decide?
 - dentro do nosso contexto social e cultural, só há uma resposta razoável: **o autor**
 - a escolha por parte do autor também é adequada do ponto de vista legal, já que ele é o detentor do direito de autor (ou **copyright**)

Direitos autorais e software

- Copyright

- para que alguém diferente do autor possa fazer praticamente qualquer tipo de uso desse trabalho, é preciso haver **uma permissão explícita por parte do autor**
- tipicamente, essa permissão é formalizada através de uma **licença ou contrato escrito**

Direitos autorais e software

- Copyright

- essa licença ou contrato é um **documento jurídico** e, como tal, precisa ser razoavelmente detalhada e precisa
- o **autor é quem define** os termos dessa licença ou contrato

Direitos autorais e software

- Copyright

- Nas leis americanas, europeias e brasileiras, trata-se de direitos outorgados pelo estado e **não de “direitos naturais”**; tanto que passam para domínio público e têm restrições através do **“fair use”**, diferentemente da propriedade física

Direitos autorais e software

- Software é um caso especial:
 - É um meio de produção, diferentemente de um livro
 - Ainda assim, é abstrato
 - Poucas “invenções” são de fato patenteáveis; além disso, **a complexidade reduz grandemente a relevância de invenções isoladas**

Direitos autorais e software

- Ainda assim, o software é tipicamente protegido por **copyright (por 95 anos)** e por patentes nos EUA (por 17 anos)
- Além disso, o código fonte é secreto
- O resultado é que há um **desequilíbrio entre o interesse público e o privado**

Por que Software Livre?

Software Livre

- Surgimento da comunidade de software livre foi uma reação
 - retorno ao compartilhamento (do código-fonte) e à colaboração (troca de ideias e trabalho em equipe)
 - só é possível em um ambiente que facilite a troca de código-fonte (Internet)

Software Livre

- Opção ética
 - **conhecimento público, disponível a toda sociedade**
- Impacto na sociedade
 - **conhecimento estratégico e fundamental demais para ficar nas mãos de apenas um grupo**
- Metodologia tecnicamente interessante
 - vantagens metodológicas, técnicas e de mercado

Software Livre

- Incentivo à colaboração como forma de **relacionamento humano e de produção**
- É uma ótima plataforma para a educação tecnológica
- Acesso a tecnologia de ponta a baixo custo

Software Livre

- Simplifica o compartilhamento do **software de forma legal**
- Oferece um **melhor equilíbrio** entre o interesse privado e o interesse público em relação ao estado atual das leis de copyright

Software Livre

- Relação da população em geral com o software tende a se tornar cada vez mais próxima
- O compartilhamento do conhecimento é uma forma de inclusão digital

Software Livre

- Software Livre permite o compartilhamento de código, otimizando o uso dos recursos
 - Menos duplicação de esforço
 - Menor custo de desenvolvimento

Software Livre

- Oportunidade para melhoria da qualidade
 - Vários olhos enxergam mais
 - Orgulho pessoal incentiva desenvolvedor a ser mais cuidadoso
 - Vários usuários envolvidos promovem melhorias e relatórios de erros

Software Livre

- Não há restrição de fornecedor: mais interessante para o usuário
- Sempre é possível evoluir o código, mesmo que o fornecedor original abandone o mercado

Licenças de Software Livre

Origem das licenças de software livre

- Software não-livre é a norma
- Há leis rígidas sobre copyright
- Patentes de software (inválidas no Brasil, porém) que afetam o desenvolvimento
- Pessoas diferentes têm visões diferentes

Licenças de Software Livre

- Os desenvolvedores de software livre **evitam redigir “do zero”** as licenças detalhando essas condições
- dá trabalho escrever uma licença consistente do ponto de vista jurídico

Licenças de Software Livre

- não é necessário “reinventar a roda”
- se o objetivo é o compartilhamento do código,
**códigos diferentes com condições de uso e
distribuição diferentes atrapalham**

Licenças de Software Livre

- Por isso, **existem licenças comumente usadas** por programadores com visões e interesses similares dependendo do contexto e do projeto
 - Ainda assim, **existem mais de 70 licenças** “reconhecidas” pela OSI (Open Source Initiative)

Licenças de Software Livre

- Apesar desse grande número de licenças diferentes, pode-se agrupá-las em 3 categorias principais:
 - Recíprocas totais
 - Recíprocas parciais
 - Permissivas

Licenças de Software Livre

- Recíprocas totais
 - GPL, AGPL e similares
 - O software é livre, deve permanecer livre e trabalhos derivados devem ser também livres
 - Exigem que trabalhos relacionados ao trabalho original de alguma maneira sejam licenciados sob a mesma licença do trabalho original

Licenças de Software Livre

- Recíprocas parciais
 - LGPL, EPL, MPL e outras
 - O software é livre e deve permanecer livre, mas trabalhos derivados não precisam ser livres
 - Permitem que projetos independentes que apenas utilizem as funcionalidades deste código sejam licenciados de forma independente

Licenças de Software Livre

- Permissivas

- Apache, MIT/X11, BSD e outras
- O software é livre, mas pode ser relicenciado sem permissão adicional do autor
 - Interessantes quando o objetivo é disseminar o software e a tecnologia correspondente, independente da licença final
 - a implementação TCP/IP do BSD, que foi a primeira utilizada no linux mas também foi a utilizada no Windows

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)[torvalds / linux](#) Public[Watch](#) 8.1k[Fork](#) 45.1k[Star](#) 140k[Code](#) [Pull requests](#) 312 [Actions](#) [Projects](#) [Security](#) [Insights](#)[master](#) [1 branch](#) [763 tags](#)[Go to file](#)[Add file](#)[Code](#)[Merge tag 'for-6.1-rc1-tag' of git://git.kernel.org/pub/scm/linux/ker...](#) aae703b 1 hour ago 1,135,998 commits

Documentation	Merge tag 'random-6.1-rc1-for-linux' of git://git.kernel.org/pub/scm/...	2 days ago
LICENSES	LICENSES/LGPL-2.1: Add LGPL-2.1-or-later as valid identifiers	10 months ago
arch	Merge tag 'random-6.1-rc1-for-linux' of git://git.kernel.org/pub/scm/...	2 days ago
block	Merge tag 'random-6.1-rc1-for-linux' of git://git.kernel.org/pub/scm/...	2 days ago
certs	certs: make system keyring depend on built-in x509 parser	25 days ago
crypto	treewide: use get_random_bytes() when possible	7 days ago
drivers	Merge tag 'v6.1-p2' of git://git.kernel.org/pub/scm/linux/kernel/git/...	yesterday
fs	Merge tag 'for-6.1-rc1-tag' of git://git.kernel.org/pub/scm/linux/ker...	1 hour ago
include	Merge tag 'cgroup-for-6.1-rc1-fixes' of git://git.kernel.org/pub/scm/...	17 hours ago
init	Merge tag 'mm-nonmm-stable-2022-10-11' of git://git.kernel.org/pub/sc...	6 days ago
io_uring	io_uring/rw: ensure kiocb_end_write() is always called	6 days ago
ipc	Merge tag 'mm-nonmm-stable-2022-10-11' of git://git.kernel.org/pub/sc...	6 days ago
kernel	Merge tag 'cgroup-for-6.1-rc1-fixes' of git://git.kernel.org/pub/scm/...	17 hours ago
lib	Merge tag 'random-6.1-rc1-for-linux' of git://git.kernel.org/pub/scm/...	2 days ago
mm	Merge tag 'random-6.1-rc1-for-linux' of git://git.kernel.org/pub/scm/...	2 days ago
net	Merge tag 'random-6.1-rc1-for-linux' of git://git.kernel.org/pub/scm/...	2 days ago
rust	Kbuild: add Rust support	21 days ago
samples	Merge tag 'vfio-v6.1-rc1' of https://github.com/awilliam/linux-vfio	6 days ago
scripts	Merge tag 'kbuild-fixes-v6.1' of git://git.kernel.org/pub/scm/linux/k...	2 days ago
security	Merge tag 'mm-stable-2022-10-08' of git://git.kernel.org/pub/scm/linu...	8 days ago
sound	Merge tag 'sound-fix-6.1-rc1' of git://git.kernel.org/pub/scm/linux/k...	4 days ago

About

Linux kernel source tree

[Readme](#)[View license](#)[140k stars](#)[8.1k watching](#)[45.1k forks](#)

Releases

[763 tags](#)

Packages

No packages published

Contributors

 5,000+[+ 13,582 contributors](#)

Languages



[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)[torvalds](#) / [linux](#) Public[Watch](#) 8.1k[Fork](#) 45.1k[Star](#) 140k[Code](#) [Pull requests](#) 312 [Actions](#) [Projects](#) [Security](#) [Insights](#)master [linux / COPYING](#)[Go to file](#)[gregkh](#) COPYING: state that all contributions really are covered by this file ...Latest commit 74835c7 on Feb 6, 2020 [History](#)[2 contributors](#)

20 lines (11 sloc) | 496 Bytes

[Raw](#)[Blame](#)

```
1 The Linux Kernel is provided under:
2
3     SPDX-License-Identifier: GPL-2.0 WITH Linux-syscall-note
4
5 Being under the terms of the GNU General Public License version 2 only,
6 according with:
7
8     LICENSES/preferred/GPL-2.0
9
10 With an explicit syscall exception, as stated at:
11
12     LICENSES/exceptions/Linux-syscall-note
13
14 In addition, other licenses may also apply. Please see:
15
16     Documentation/process/license-rules.rst
17
18 for more details.
19
20 All contributions to the Linux Kernel are subject to this COPYING file.
```



© 2022 GitHub, Inc.

[Terms](#)[Privacy](#)[Security](#)[Status](#)[Docs](#)[Contact GitHub](#)[Pricing](#)[API](#)[Training](#)[Blog](#)[About](#)

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)[torvalds / linux](#) Public[Watch](#) 8.1k[Fork](#) 45.1k[Star](#) 140k[Code](#) [Pull requests](#) 312 [Actions](#) [Projects](#) [Security](#) [Insights](#)master [linux / LICENSES / preferred / LGPL-2.0](#)[Go to file](#)**Thomas Gleixner** LICENSES: Add the LGPL 2.0 license ...

Latest commit 4ed5176 on Dec 28, 2017

[History](#)

0 contributors

487 lines (403 sloc) | 25.6 KB

[Raw](#)[Blame](#)

```
1 Valid-License-Identifier: LGPL-2.0
2 Valid-License-Identifier: LGPL-2.0+
3 SPDX-URL: https://spdx.org/licenses/LGPL-2.0.html
4 Usage-Guide:
5   To use this license in source code, put one of the following SPDX
6   tag/value pairs into a comment according to the placement
7   guidelines in the licensing rules documentation.
8   For 'GNU Library General Public License (LGPL) version 2.0 only' use:
9     SPDX-License-Identifier: LGPL-2.0
10  For 'GNU Library General Public License (LGPL) version 2.0 or any later
11  version' use:
12    SPDX-License-Identifier: LGPL-2.0+
13  License-Text:
14
15  GNU LIBRARY GENERAL PUBLIC LICENSE
16  Version 2, June 1991
17
18  Copyright (C) 1991 Free Software Foundation, Inc.
19  51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA
20
21  Everyone is permitted to copy and distribute verbatim copies of this
22  license document, but changing it is not allowed.
23
24  [This is the first released version of the library GPL. It is numbered 2
25  because it goes with version 2 of the ordinary GPL.]
```

(Creative Commons)

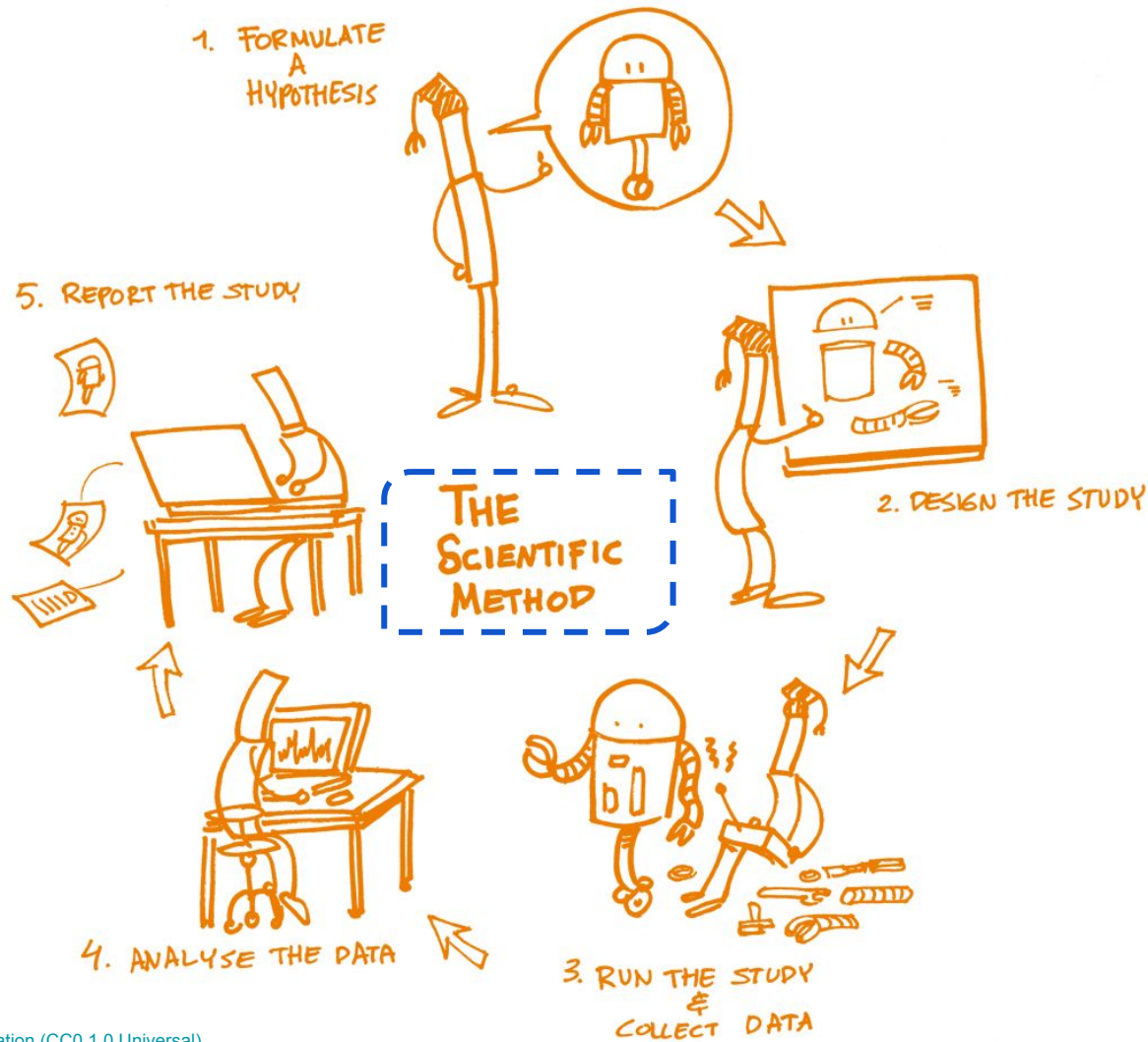


O que é Ciência Aberta?



Ciência

Método Científico

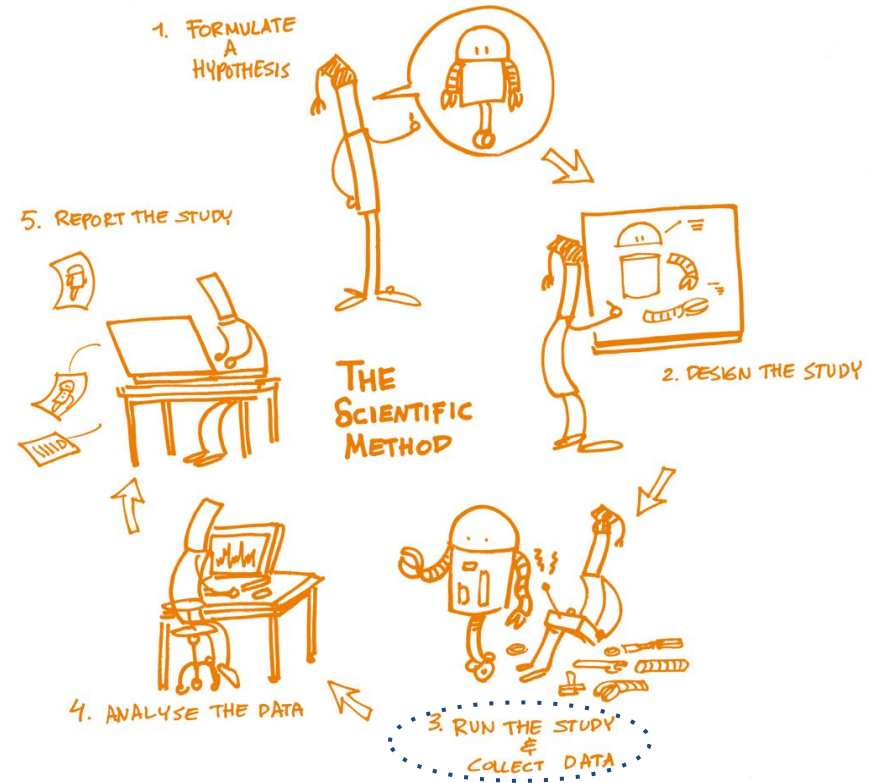


Ciência 1.0

- coleta de dados individual
- escrever artigo
- submeter artigo para revisão
- "gatekeepers" e revisão por pares
- publicar ou rejeitar



***informação disponível
para o público (ou não?)***



Ciência 1.0

- coleta de dados individual
- escrever artigo
- submeter artigo para revisão
- "gatekeepers" e revisão por pares
- publicar ou rejeitar

*informação disponível
para o público (ou não?)*

Ciência 2.0

- pré-registrar estudos
- compartilhar ideias, dados, métodos e protocolos, via blogs, plataformas, repositórios
- submeter *preprints*
- publicar em blogs, wikis, e *journals*



***informação e dados
disponíveis para o público***

Ciência Aberta

Recomendação da UNESCO:

*Open Science is the movement to
make scientific research and data accessible to all.*

It includes practices such as publishing open scientific research, campaigning for open access and generally making it easier to publish and communicate scientific knowledge.

Colocar referência para documento.

Ciência Aberta

Recomendação da UNESCO (continuação)

*"Additionally, it includes other ways to make Science more **transparent** and **accessible** during the research process.*

*This includes [...] aspects of **open source** and crowdfunded research projects."*

Ciência Aberta

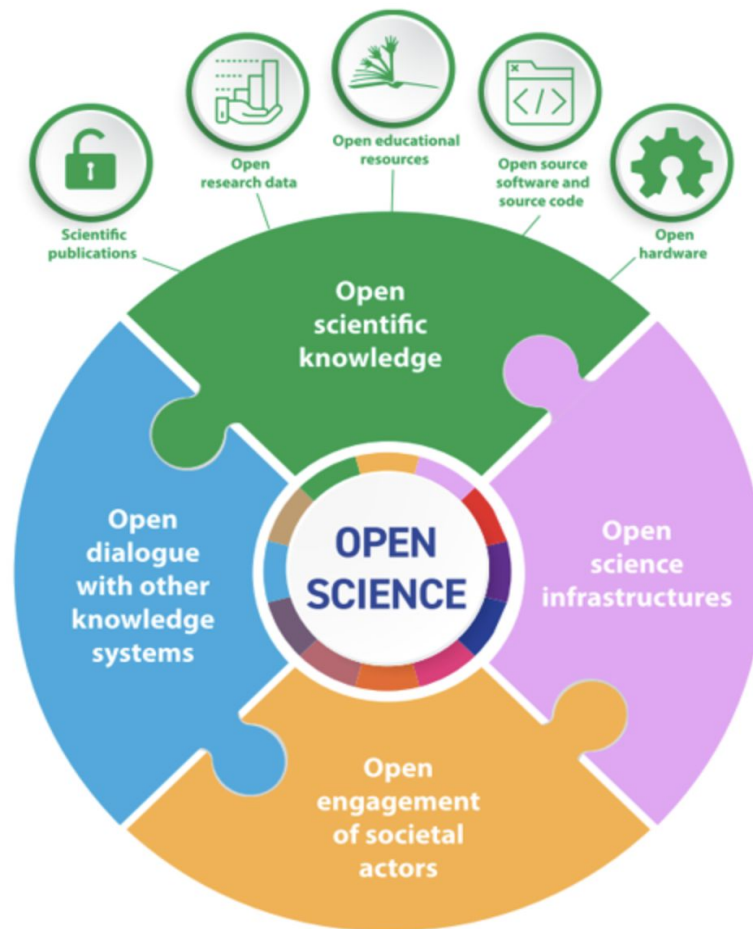
Definição da Comissão Europeia:

*Open Science represents a new approach to the scientific process based on cooperative work and new ways of diffusing knowledge by using **digital technologies** and new collaborative tools.*

Ciência Aberta

Open*

- *open access (publications)*
- *open data*
- *open educational resources*
- *open source software*
- *open hardware*



Pesquisa Aberta e Reprodutível

Reprodutibilidade é essencial

Um estudo é reproduzível se houver um conjunto específico de funções/análises computacionais (geralmente especificadas em termos de código) que reproduzam exatamente todos os resultados de um artigo publicado a partir de dados brutos.

Pesquisa Aberta e Reprodutível

Replicabilidade é mais "forte"

Um estudo só é replicável se alguém realizar exatamente o mesmo experimento (pelo menos) duas vezes, coletar dados da mesma maneira nas duas vezes, realizar a mesma análise de dados e chegar às mesmas conclusões.

Pesquisa Aberta e Reprodutível

Se **reprodutibilidade** é uma característica relevante, **preservar e dar acesso**

a **código e workflows**

torna-se vital para a **credibilidade da pesquisa**.

Software na Ciência Aberta

Bibliotecas, scripts, ferramentas, ...

- Limpeza de dados
- Processamento e visualização de dados,
- Criação de modelos
- Análise e predição, etc.

Better Software, Better Research

*“Devemos **reconhecer o software como instrumento experimental de primeira-classe na Ciência.**”*

Carole GOBLE. Better software, better research.
IEEE Internet Computing, IEEE, v. 18, n. 5, p. 4–8, 2014.

"Existência de muitos projetos com poucos usuários, com ciclos de vida curtos que se encerram junto ao financiamento inicial, comunidades de usuários desconectadas e paralelas, incompatibilidades entre os projetos de maneira persistente e imutável, e tentativas constantes e aparentemente não coordenadas de "reiniciar" tudo (re-boots)."

HOWISON, J. et al. Understanding the scientific software ecosystem and its impact: Current and future measures. Research Evaluation, 2015.

"Poucos estudos são replicáveis: faltam informações incluindo dados e ferramentas, e apenas 20% dos estudos possuem ferramentas disponíveis.

AMANN, S. et al. Software mining studies: Goals, approaches, artifacts, and replicability. In: Software Engineering. [S.l.]: Springer, 2015. p. 121–158.

"... o atual modelo científico produz um 'tsunami' de ferramentas não-usáveis ... apenas metade delas são 'fáceis de instalar', e próximo de 1/3 não são mais instaláveis a partir da URL fornecida no artigo original ..."

HERSHBERG, Elliot. How Software in the Life Sciences Actually Works (And Doesn't Work). New Science, 2022: <https://newscience.org/how-software-in-the-life-sciences-actually-works-and-doesnt-work>.

Boas práticas no desenvolvimento de Software Livre

- uso de repositórios públicos
- sistemas de controle de versão
- colaboração por pares
- revisão de código
- testes automatizados
- formatos e interfaces padrão
- documentação relevante
- ...

**Boas práticas
para a
Ciência Aberta?**

Software Livre e Ciência Aberta

Liberdade 1

A liberdade para executar o software para qualquer finalidade:

- permite que pesquisadores utilizem o software existente sem ter que comprá-lo ou construí-lo do zero para realizar seus próprios estudos.

Liberdade 2

A liberdade para estudar e modificar o software na forma de código-fonte:

- oferece suporte à **reprodutibilidade e replicabilidade**, divulgando software de pesquisa, artefatos relacionados e o conhecimento embutido neles.
- aumenta a **transparência** (fluxos de trabalho visíveis), **auditabilidade** e **confiabilidade** (os resultados podem ser verificados por terceiros e qualquer pessoa pode detectar e corrigir um erro ou um recurso malicioso).

Liberdade 3

A liberdade para redistribuir cópias possibilita o compartilhamento de pacotes de **reprodução e replicação** com:

- a. **dados** brutos originais
- b. **código** necessário para realização dos experimentos, análises e interpretações em diferentes ambientes.

Liberdade 4

A liberdade para distribuir versões modificadas do software permite que pesquisadores

- desenvolvam seu próprio trabalho, reutilizando e expandindo o fluxo de trabalho, código-fonte ou ferramenta de terceiros
- compartilhem o novo conhecimento para o benefício de toda a comunidade científica.

Práticas

Sistemas de Controle de versão

- Manutenção de várias versões do mesmo software, possivelmente, referenciadas por experimentos de pesquisa e artigos científicos.

Repositórios Abertos

- Compartilhamento de vários tipos de ativos de pesquisa (algoritmos, dados, código, relatórios e fluxos de trabalho), apoiando a reprodutibilidade, reduzindo a redundância e promovendo a colaboração científica aberta.

Práticas

Colaboração entre Pares

- A colaboração entre pares é frequente e habilitada por meio de acesso compartilhado ao código-fonte e vários canais de comunicação.

Revisão de Código

- A revisão de código promove a melhoria da qualidade do software por meio de compartilhamento, colaboração e revisão por pares e pode ser aplicada a outros ativos de pesquisa.

Práticas

Testes

- Os testes automatizados aumentam a confiabilidade e a facilidade de manutenção, além de promover agilidade no desenvolvimento de novos recursos.

Práticas

Padronização

- O uso de formatos de dados e interfaces padronizados facilita a integração com outros sistemas e desencoraja a dependência de fornecedores específicos.

Práticas

Documentação

- A documentação frequente e contínua é uma prática recomendada para manter os guias do usuário, manuais e outros documentos relevantes atualizados em relação à versão mais recente do software.

Boas práticas de Software Livre **podem ser consideradas** **Boas práticas para a Ciência Aberta**

Suporte contínuo à:

- disponibilidade e reúso de ativos
- colaboração entre pares
- transparência do fluxo de trabalho
- confiabilidade

Software Acadêmico

**(software desenvolvido ou utilizado no contexto
da pesquisa científica)**

Research Software

FAIR 4 Software

O software desenvolvido ou utilizado no contexto de uma pesquisa científica deve ser **FAIR**:

(F)indable: localizável

(A)ccessible: acessível

(I)nteroperable: interoperável

(R)eusable: reutilizável

Software Livre é FAIR

Findable

Software livre pode ser localizado em repositórios com base em **identificadores e descritores**, utilizando diversos critérios como palavras-chave, linguagem de programação, versão do software, entre outros.

Software Livre é FAIR

Acessible

A acessibilidade é encorajada em software livre disponível em **repositórios abertos**, com licenças de compartilhamento explícitas e bem definidas e documentação associada.

Software Livre é FAIR

Interoperable, Reusable

A definição de interfaces de programação e formatos de entrada/saída e o uso de padrões promovem a interoperabilidade e o reuso por vários grupos de pesquisa em todo o mundo.

Software Livre é fundamental para a Ciência Aberta

Esta mensagem deve ser transmitida a cientistas, agências de fomento à pesquisa, organizações científicas e governamentais.

Better Software, Better Research

Livre

*“Devemos reconhecer o software
como instrumento experimental
de primeira-classe na Ciência.”*

Paulo Meirelles

@paulormm

paulo.meirelles@ufabc.edu.br

Christina von Flach

@chrisflach

flach@ufba.br



Perguntas?



O Papel da Computação na Ciência Aberta



ARTIGO

CIÊNCIA ABERTA - COLABORAÇÃO SEM BARREIRAS PARA O AVANÇO DO CONHECIMENTO

POR

Claudia Buzer Medeiros
cmbz@unicamp.br

Ciência Aberta – introdução e implementação computacional

O que é Ciência Aberta? Vamos separar a "especificação" da "implementação". Embora não haja uma definição fixa para o termo "Ciência Aberta", ele costuma ser usado para denotar o conjunto de políticas, iniciativas e ações

para disseminar e compartilhar conhecimento, geralmente por meio digital, para que todos os resultados associados à pesquisa científica se tornem acessíveis a todos. O conceito principal desta especificação é colaboração sem barreiras – geográficas, temporais, culturais, sócio-econômicas ou políticas.

A implementação exige pesquisa e desenvolvimento em Computação.



ARTIGO

SOFTWARE LIVRE: PRÉ-REQUISITO PARA A CIÊNCIA ABERTA

POR

Christina von Flach e Fábio Kon
flach@ufba.br, kon@ime.usp.br

A ciência sempre se pautou pela disseminação do conhecimento, com destaque para a reprodutibilidade de experimentos. É claro que há limites práticos – e mesmo éticos – para o que pode ser compartilhado e reproduzido, mas expandir esses limites é um objetivo de interesse para cientistas de todas as áreas. Esse é o foco do movimento pela ciência aberta, que tem tido bastante repercussão nos últimos anos.

A Ciência Aberta requer que as ferramentas e os instrumentos necessários para a prática científica estejam dis-

poníveis para todos, de modo que os experimentos possam ser reproduzidos e os resultados verificados por terceiros. No Século 21, o software consolidou-se como onipresente no conjunto de ferramentas usado por pesquisadores das Ciências Exatas e Biomédicas e tem uso crescente nas Ciências Sociais e Humanas. Software é usado para limpeza, processamento e visualização de dados, bem como para criar modelos e realizar previsões. Algoritmos especializados são codificados na forma de bibliotecas, scripts e metadados. Sem compartilhar todos esses artefatos, é muito difícil e custoso reproduzir a pesquisa científica e validar sua correção.



Obrigado!