



CAPÍTULO 2

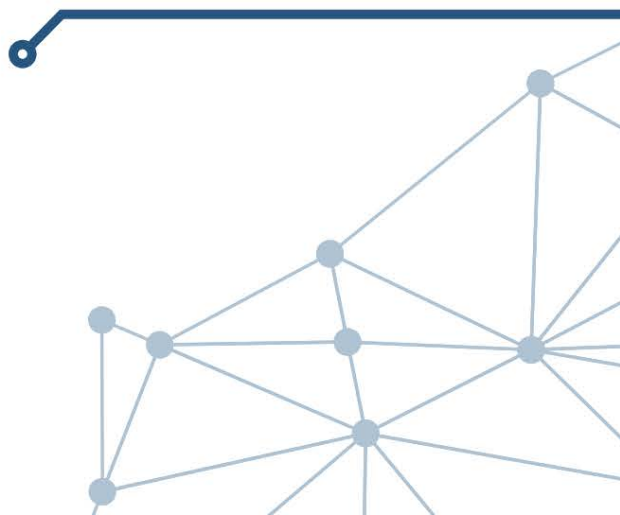
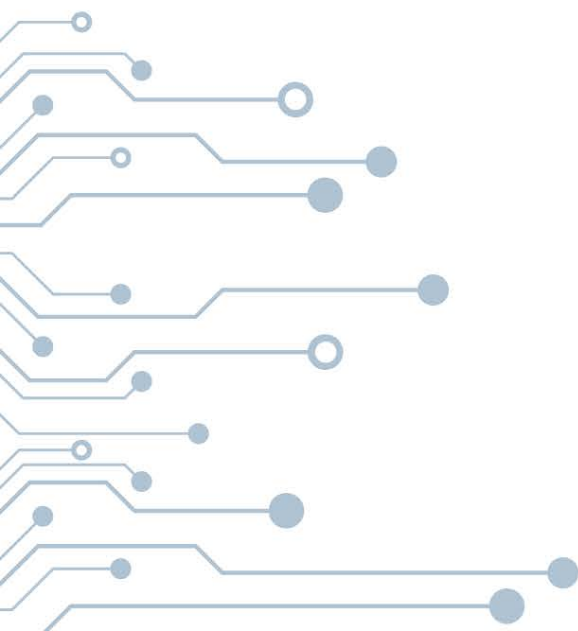
REPOSITÓRIO DE CÓDIGOS ABERTOS

MILTON SHINTAKU

LUCAS RODRIGUES COSTA

REBECA DOS SANTOS MOURA

BERNARDO DIONÍZIO VECHI



■ 2.1 INTRODUÇÃO

Historicamente, com o surgimento dos computadores, também surgiram os programas e seus códigos, que materializam os algoritmos criados para processar dados. Inicialmente, a programação era realizada diretamente em código de máquina, o que ficou conhecido como linguagem *assembly*. Como se sabe, os computadores são máquinas programáveis, capazes de executar algoritmos geralmente codificados em alguma linguagem.

Devido à sua estrutura física, os programas no início eram simples, grande parte destinada à realização de cálculos, com poucas instruções. Com o passar do tempo, os computadores se tornaram progressivamente mais potentes e rápidos, com memórias cada vez maiores. Foi nesse contexto que, em 1965, surgiu a chamada Lei de Moore, formulada por Gordon Earl Moore, químico estadunidense e um dos fundadores da Intel, empresa de pesquisa e produção de processadores. Moore estabeleceu que o poder de processamento dos computadores deveria dobrar a cada 18 meses. Desde então, essa lei tem se mantido em grande parte, a ponto de os smartphones atuais serem imensamente mais poderosos que os primeiros computadores, os quais ocupavam andares inteiros dos edifícios.

Com relação aos códigos, a evolução tecnológica trouxe as linguagens de programação compiladas, eliminando a necessidade de programar diretamente em linguagem de máquina. Foi assim que surgiram linguagens como o FORTRAN (*FORmula TRANslation*), voltada para cálculos lógico-matemáticos, e o COBOL (*COMmon Business Oriented Language*), direcionada para negócios e conhecida por ser próxima ao inglês estruturado. Com o advento da Internet e da Web, com suas páginas hiperligadas, novas linguagens surgiram, como o Java e o PHP (inicialmente *Personal Home Page* e, posteriormente, *PHP Hypertext*



Processador). Estas são algumas das linguagens compiladas desenvolvidas para atender às crescentes necessidades computacionais.

Nesse contexto, surgiram os sistemas informatizados, compostos por um conjunto de programas que buscam automatizar determinadas atividades dentro de um sistema de informação. Entende-se que um sistema de informação é constituído por pessoas, atividades, processos, dados, informações, fluxos informacionais, entre outros elementos, podendo ou não contar com um sistema informatizado. Mesmo antes da informatização, já existiam sistemas de informação bem estruturados, que, com o advento dos computadores, tornaram-se significativamente mais eficientes.

Feita essa distinção, observa-se que os sistemas informatizados se tornaram comuns, com uma oferta crescente de serviços e produtos. Atualmente, a maioria dos sistemas de informação conta com um sistema informatizado que o atende total ou parcialmente, desenvolvido com uma ampla variedade de tecnologias, entre elas os chamados softwares livres, que marcaram uma grande mudança no modelo de negócios das ferramentas informatizadas.

Se, inicialmente, os sistemas informatizados eram “caseiros”, ou seja, desenvolvidos internamente por cada empresa – um modelo que ainda persiste, mas em menor escala –, esse cenário criou oportunidades para o surgimento de empresas cujo produto principal era o software. Observou-se que muitas empresas compartilhavam as mesmas necessidades em termos de sistemas de informação, como gestão de recursos humanos, contabilidade e controle de documentos. Assim, surgiram empresas dedicadas ao desenvolvimento e à comercialização de sistemas informatizados.

Como reação ao comércio de softwares, emergiram os chamados softwares livres, distribuídos de forma gratuita, permitindo que os usuá-



rios os utilizassem sem restrições. Historicamente, a primeira ocorrência desse tipo de distribuição remonta a 1956, quando a AT&T passou a liberar sua versão do sistema operacional Unix, devido à imposição do governo estadunidense, que proibia a empresa de comercializar softwares. Entretanto, a expansão dos softwares livres ocorreu com a popularização dos computadores pessoais e da web.

A mudança no modelo de negócios imposta pelo surgimento do software livre foi significativa, uma vez que o foco passou da comercialização de um produto para a oferta de serviços. Enquanto os softwares devem ser distribuídos livremente, serviços como consultoria, ajustes e suporte podem ser cobrados. Dessa forma, qualquer pessoa com conhecimento em informática pode se tornar um produtor de software, democratizando o processo de desenvolvimento.

2.2 REPOSITÓRIOS DE CÓDIGO ABERTO

O movimento do software livre começou com a distribuição dos programas em formato executável, ou seja, como ferramentas fechadas apenas para a execução. Esse formato é semelhante aos aplicativos móveis atuais, nos quais o usuário pode baixar e utilizar o software sem restrições, mas não tem acesso ao código-fonte para análise ou modificação. Posteriormente, o movimento passou a permitir o acesso ao código-fonte, com algumas ou nenhuma restrição.

A quebra de barreiras para o uso livre de software, enquanto movimento mundial, nasceu com o Projeto GNU. Seu criador, Richard Stallman, relata que o projeto tem raízes no início da década de 1970, quando



fazia parte de uma comunidade de compartilhamento de software, no contexto das pesquisas em inteligência artificial. Stallman (1998) define software livre como a liberdade de executar um programa sem restrições, alterá-lo conforme as suas necessidades e redistribuí-lo, inclusive com suas modificações.

Seguindo essa linha, Silveira (2018) destaca que, embora o software livre tenha autoria, vinculada aos seus desenvolvedores, ele não possui um dono. Para isso, utiliza licenças não proprietárias, como as criadas pelo Projeto GNU, que se assemelham aos direitos autorais de trabalhos científicos de Acesso Aberto. Stallman (Gay, 2002, p. 20, tradução nossa) define essa licença como: “O Copyleft utiliza a lei dos direitos autorais, mas a inverte para servir ao oposto de seu propósito habitual. Em vez de privatizar o software, torna-o um meio de mantê-lo livre”.

Para armazenar e facilitar o compartilhamento dos códigos-fonte dos softwares livres, foram criados os chamados Repositórios de Código-Fonte, iniciativas acessíveis via web. Com esses repositórios, os membros da comunidade podem depositar livremente seus códigos-fonte, contribuir com os códigos já existentes ou baixá-los para uso. Esses repositórios oferecem uma ampla gama de serviços para facilitar o uso e o desenvolvimento colaborativo.

Chapman (2009), ao tratar do desenvolvimento de sites, descreve os repositórios de código-fonte como locais onde são armazenadas cópias do código e seu histórico de alterações (versionamento), permitindo que o usuário selecione qual versão utilizar. Isso facilita o desenvolvimento colaborativo, especialmente em equipes descentralizadas. Mesmo que várias pessoas estejam trabalhando no mesmo código, os repositórios oferecem funcionalidades voltadas à gestão desses casos. No entanto, segundo o autor, o maior benefício dos repositórios talvez resida na manutenção de diversas cópias de segurança, o que garante maior segurança e versatilidade no desenvolvimento de software.



Nesse contexto, pode-se afirmar que a existência dos softwares livres está intimamente ligada à dos repositórios, pois estes são as ferramentas pelas quais o software se estabelece como verdadeiramente livre. Os repositórios de código-fonte tornam-se, portanto, canais essenciais não apenas para o compartilhamento, mas também, e principalmente, para a colaboração, viabilizando a atuação da comunidade na melhoria e evolução contínua de um software livre.

2.3 HISTÓRICO DOS REPOSITÓRIOS DE CÓDIGO ABERTO

Como mencionado anteriormente, a primeira referência ao software livre pode ser identificada em 1965, com a distribuição da versão do sistema operacional Unix pela AT&T. Contudo, essa distribuição não ocorreu por meio de repositórios, mas via portais de empresas ou instituições, que ofereciam uma área para descrição do software e a possibilidade de download. Ainda hoje, várias organizações utilizam esse modelo, disponibilizando seus softwares para download diretamente em seus portais, mesmo que sejam softwares livres.

No entanto, muitos desenvolvedores de software livre não dispõem de estrutura para manter um portal próprio, visto que não recebem retorno financeiro pelo uso das ferramentas. Dessa forma, a criação de um local seguro para depositar seus códigos, permitindo que outros possam baixá-los, ajustá-los, adicionar novas funcionalidades e colaborar no desenvolvimento, tornou-se uma necessidade crucial para a comunidade de desenvolvedores e usuários. Nesse contexto, os repositórios de código-fonte se tornaram fundamentais para a estrutura das comunidades de software livre.



O início dos repositórios de código-fonte pode ser atribuído ao projeto Git, proposto por Linus Torvalds (criador do Linux), como uma plataforma para controle de versão e desenvolvimento colaborativo de software livre, licenciado pelo GNU. Inicialmente, o termo "Git" possuía uma conotação menos formal em inglês, mas, dentro da comunidade de desenvolvedores, passou a ser reconhecido como um ambiente essencial para o compartilhamento de códigos.

Spinellis (2012) relata que o projeto Git começou a ser utilizado em 2005 como um gestor de revisões para o desenvolvimento do núcleo do sistema operacional Linux, e sua funcionalidade, portabilidade e eficiência o tornaram pioneiro entre os repositórios de código-fonte. Esse modelo foi tão eficaz que acabou sendo replicado em outras iniciativas.

Loeliger e McCulough (2012) informam que o primeiro commit no Git ocorreu em 7 de abril de 2005, marcando o início das atividades da plataforma. Em 20 de abril do mesmo ano, houve a primeira utilização efetiva do Git, com a aplicação de códigos no Linux, provando sua eficácia no desenvolvimento colaborativo. Desde então, o Git tornou-se um modelo consolidado para repositórios de código-fonte.

Em 2008, surgiu o *GitHub*, baseado no modelo Git e desenvolvido pela empresa de mesmo nome, adquirida em 2018 pela Microsoft. O *GitHub* rapidamente se tornou popular entre desenvolvedores, com mais de 46 mil projetos já depositados após apenas um ano de operação. Lima, Rossi e Musolesi (2014) relatam que, em 2013, o *GitHub* já contava com mais de 3,5 milhões de usuários e mais de 10 milhões de projetos armazenados. Originalmente, o *GitHub* denomina cada conjunto de códigos-fonte como “repositório”, mas, por questões conceituais, nesta obra será considerado o termo “projeto”.

A atuação em nuvem foi possivelmente um dos fatores que impulsionou o sucesso do *GitHub*, mesmo sendo, atualmente, uma organi-



zação privada. A maioria dos projetos atualmente depositados são de software livre, muitos dos quais pertencem a empresas, e não apenas a desenvolvedores independentes. Outro fator de sucesso do *GitHub* foi sua capacidade de atender livremente à comunidade, permitindo que qualquer pessoa pudesse depositar seus códigos.

Na esteira do *GitHub*, em 2011 foi criado o *GitLab*, desenvolvido no próprio *GitHub* como um software livre para construção de repositórios de código-fonte, licenciado sob a MIT License (Hethey, 2013). O *GitLab* permitiu que organizações criassem seus próprios repositórios em suas infraestruturas computacionais. Um fator importante para o seu sucesso foi a migração de usuários quando a Microsoft adquiriu o *GitHub*.

Diversas outras iniciativas surgiram ao longo da história dos repositórios de código-fonte, embora poucas tenham tido o impacto do *Git*, *GitHub* e *GitLab*. Atualmente, com o movimento de Ciência Aberta, os repositórios de código-fonte abertos têm ganhado destaque no desenvolvimento de softwares relacionados a pesquisas acadêmicas. Dessa forma, os softwares livres tornam-se protagonistas no cenário científico, trazendo consigo o protagonismo dos repositórios de código-fonte.

2.4 TECNOLOGIAS PARA CONSTRUÇÃO DE REPOSITÓRIOS DE CÓDIGOS ABERTOS

A gestão de código-fonte é uma prática essencial no desenvolvimento de software contemporâneo. Com o aumento da colaboração em projetos, a necessidade de repositórios de código robustos e eficientes



tornou-se indispensável. Repositórios de código aberto não apenas facilitam o trabalho colaborativo entre desenvolvedores, mas também promovem transparência, reutilização de código e inovação. Nesse cenário, surgiram diversas tecnologias para apoiar a criação e a gestão desses repositórios. Entre as principais plataformas utilizadas para esse fim, destacam-se o *GitLab*, *Apache Allura*, *Phabricator* e *Gitea*.

GitLab¹ é uma plataforma completa para DevOps, oferecendo um ciclo de vida de desenvolvimento de software que abrange desde a codificação até a implementação. Ela fornece uma ampla gama de funcionalidades, como repositórios Git, integração contínua/entrega contínua (CI/CD), gestão de problemas, revisões de código e muito mais. O *GitLab* é amplamente reconhecido por sua interface intuitiva e robusta, além de permitir uma integração profunda com diversas ferramentas de desenvolvimento, tornando-o uma escolha popular entre equipes de todos os tamanhos. Além disso, por ser um projeto de código aberto, o *GitLab* promove a transparência e a colaboração dentro da comunidade de desenvolvimento, permitindo que qualquer pessoa possa visualizar, modificar e distribuir o código.

Apache Allura² é uma plataforma de software de código aberto que oferece ferramentas para o gerenciamento de projetos de software, incluindo rastreamento de problemas, wikis e múltiplos repositórios de código-fonte. Altamente customizável e extensível, o *Allura* permite que os usuários adaptem a plataforma às suas necessidades específicas. Originalmente desenvolvido para o *SourceForge*, o *Allura* continua sendo uma solução poderosa para aqueles que buscam uma plataforma flexível e robusta para a hospedagem de projetos de código aberto.

¹ Disponível em: <https://docs.gitlab.com/ee/devsecops.html/>. Acesso em: 13 jun. 2024.

² Disponível em: <https://allura.apache.org/>. Acesso em: 13 jun. 2024.



Phabricator³ é uma suíte de ferramentas de desenvolvimento de software que inclui funcionalidades para revisão de código, rastreamento de tarefas e bugs, além de hospedar repositórios Git, Mercurial e Subversion. Originalmente desenvolvido pelo Facebook, o Phabricator é amplamente reconhecido por sua flexibilidade e pela variedade de ferramentas integradas que oferece. É ideal para grandes equipes que necessitam de uma solução unificada para gerenciar projetos de software complexos. A natureza de código aberto do Phabricator promove a transparência e a inovação, permitindo que qualquer desenvolvedor acesse, modifique e redistribua o código, fortalecendo a comunidade de desenvolvimento.

Gitea⁴ é uma plataforma leve e autossuficiente para hospedagem de repositórios Git. Projetado para ser simples de instalar e configurar, o Gitea é ideal para pequenas e médias equipes que necessitam de uma solução eficiente e fácil de usar. Ele oferece funcionalidades essenciais, como gestão de repositórios, revisão de código, rastreamento de issues e integração contínua/entrega contínua (CI/CD) básico. A simplicidade e eficiência do Gitea, aliadas à sua natureza de código aberto, promovem a colaboração e a inovação dentro da comunidade de desenvolvimento, tornando-o uma escolha popular entre desenvolvedores que valorizam transparência e flexibilidade.

Cada uma dessas tecnologias oferece um conjunto distinto de funcionalidades e vantagens, permitindo que desenvolvedores e equipes escolham a ferramenta mais adequada para suas necessidades específicas. Compreendendo as características únicas de cada plataforma, é possível construir repositórios de código aberto que sejam eficientes, colaborativos e robustos, promovendo a inovação e a qualidade no desenvolvimento de software.

³ Disponível em: <https://secure.phabricator.com/book/phabricator/article/introduction/>. Acesso em: 13 jun. 2024.

⁴ Disponível em: <https://docs.gitea.com/>. Acesso em: 13 jun. 2024.



2.5 COMUNIDADES DE DESENVOLVIMENTO DE SOFTWARES LIVRE

As comunidades de desenvolvimento de software livre são a espinha dorsal do movimento de código aberto, promovendo a colaboração, a troca de conhecimento e o apoio mútuo entre desenvolvedores. Essas comunidades não apenas facilitam o desenvolvimento de software, mas também ajudam a resolver problemas, compartilhar boas práticas e criar um senso de pertencimento entre seus membros. Ferramentas de fórum, como o *Discourse* e o *Stack Overflow*, desempenham um papel fundamental nesse ecossistema, oferecendo plataformas para discussão e suporte técnico.

Discourse⁵ é uma plataforma de fórum de código aberto que se destaca por sua interface moderna e funcionalidades avançadas de discussão. Projetado para ser uma ferramenta de comunicação flexível e poderosa, o *Discourse* suporta discussões em tempo real, moderação comunitária e integração com outras ferramentas de desenvolvimento. Sua natureza de código aberto permite que as comunidades adaptem e personalizem a plataforma de acordo com suas necessidades específicas, promovendo um ambiente colaborativo e inclusivo.

Stack Overflow⁶ é uma plataforma de perguntas e respostas focada em programação e desenvolvimento de software, destacando-se por sua estrutura organizada e pela qualidade das respostas fornecidas pela comunidade. Projetado para ser uma ferramenta eficiente e poderosa de suporte técnico, o *Stack Overflow* permite que os usuários façam

⁵ Disponível em: <https://www.discourse.org/>. Acesso em: 13 jun. 2024.

⁶ Disponível em: <https://stackoverflow.com/>. Acesso em: 13 jun. 2024.



perguntas específicas e recebam respostas detalhadas de desenvolvedores experientes. Sua natureza colaborativa incentiva a participação e a contribuição de especialistas, enquanto a gamificação, através de pontos e badges, motiva a qualidade e a precisão das respostas. O Stack Overflow se tornou uma referência indispensável para desenvolvedores que buscam soluções rápidas e confiáveis para problemas de programação, promovendo um ambiente de aprendizado e troca de conhecimento contínuo.

Fóruns como o Stack Overflow têm sido ferramentas essenciais para comunidades de desenvolvedores, proporcionando um espaço onde os membros podem compartilhar problemas e soluções, discutir novas ideias e colaborar em projetos. Essas plataformas permitem a criação de tópicos organizados, facilitando a busca por informações e a participação em discussões significativas. Em resumo, as comunidades de desenvolvimento de software livre são fundamentais para o avanço do software de código aberto, promovendo a colaboração e o desenvolvimento contínuo de soluções inovadoras.

■ 2.6 INICIATIVAS INTERNACIONAIS

As iniciativas internacionais no campo dos repositórios de código aberto têm desempenhado um papel crucial na promoção da colaboração global entre desenvolvedores. Plataformas de repositório como serviço (Repository as a Service - RaaS) oferecem uma infraestrutura robusta e escalável para armazenamento e gestão de código-fonte, permitindo que desenvolvedores de todo o mundo contribuam para projetos de forma eficiente.



O RaaS é uma abordagem moderna para a gestão de código-fonte que proporciona uma infraestrutura baseada em nuvem para armazenamento, gerenciamento e colaboração de código. Este modelo revolucionou a maneira como os desenvolvedores trabalham, oferecendo acesso fácil e seguro a repositórios de código, além de uma série de ferramentas integradas para facilitar o desenvolvimento de software.

Uma das principais vantagens do RaaS é a sua capacidade de escalar conforme as necessidades dos usuários. Em vez de depender de servidores locais e infraestrutura interna, os desenvolvedores podem utilizar serviços em nuvem que garantem repositórios de alta disponibilidade, assegurando que o código esteja sempre acessível, independentemente da localização ou do dispositivo utilizado. Além disso, a manutenção e a atualização desses serviços são gerenciadas pelos provedores, permitindo que os desenvolvedores se concentrem nas tarefas de desenvolvimento sem se preocupar com a infraestrutura subjacente.

Entre as principais plataformas disponíveis, destacam-se *GitHub*, *GitLab* e *Bitbucket*, que oferecem serviços baseados em nuvem acessíveis a toda a comunidade de desenvolvimento. Essas plataformas RaaS não apenas fornecem armazenamento de código, mas também uma gama de funcionalidades avançadas, como integração contínua/entrega contínua (CI/CD), gestão de problemas, revisão de código e colaboração em tempo real. Por exemplo, o *GitHub* oferece *GitHub Actions* para automação de fluxos de trabalho, enquanto o *GitLab* é conhecido por suas capacidades completas de *DevOps*, permitindo uma integração contínua e fluida de processos de desenvolvimento.

Essas plataformas não só fornecem os meios técnicos para hospedar e gerenciar código, mas também promovem a colaboração global e a inovação. Ao oferecer serviços baseados em nuvem acessíveis, elas permitem que desenvolvedores de todas as partes do mundo colaborem em projetos de software livre, contribuindo para o avanço da tecnologia



e a criação de soluções inovadoras. Desenvolvedores podem trabalhar juntos em projetos de código aberto ou privado, facilitando a revisão de código, a detecção de bugs e a implementação de novas funcionalidades.

Além disso, a segurança é uma preocupação central para os provedores de RaaS. Esses serviços implementam medidas rigorosas de segurança, como autenticação de dois fatores, criptografia de dados e backups regulares, garantindo que o código esteja protegido contra acesso não autorizado e perda de dados.

O RaaS também oferece excelente integração com outras ferramentas e serviços de desenvolvimento. Por exemplo, integrações com plataformas de comunicação como Slack, ferramentas de gestão de projetos como *Jira* e *Trello*, e ambientes de desenvolvimento integrados (IDEs) como *Visual Studio Code*, tornam o fluxo de trabalho dos desenvolvedores mais eficiente e coeso.

Essas iniciativas internacionais têm sido fundamentais para a democratização do desenvolvimento de software, proporcionando ferramentas que qualquer pessoa pode usar para aprender, colaborar e criar.

2.7 INICIATIVAS NACIONAIS (REDE MOARA)

A Rede Moara⁷ representa uma inovação no contexto brasileiro, destacando-se por promover a Ciência Aberta e a colaboração científica. A plataforma oferece um espaço para discussão e troca de conhecimentos,

⁷ Disponível em: <https://redemoara.ibict.br/>. Acesso em: 13 jun. 2024.



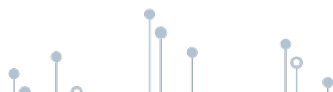
essencial para o progresso científico e tecnológico. Além da cooperação técnica, a Rede Moara aborda práticas de pesquisa aberta, reutilização de dados e a construção de uma comunidade científica integrada.

A ideia da Rede Moara é que ela seja composta por várias instituições e pesquisadores, com o objetivo de criar um ecossistema de Ciência Aberta. Este ecossistema facilita o compartilhamento de códigos de software, permitindo o uso, cópia, modificação e redistribuição desses códigos. A infraestrutura da Rede Moara inclui ferramentas e tecnologias que suportam esses processos, como APIs para comunicação com fóruns, educação a distância (EaD), bibliotecas digitais e wikis. A Rede Moara depende de políticas e diretrizes que possibilitam a implementação de suas iniciativas. Essas políticas garantem que os projetos e processos associados à Ciência Aberta sejam integrados nos laboratórios, institutos de pesquisa e discussões acadêmicas.

Ao incentivar a colaboração entre pesquisadores e a utilização de software livre, a Rede Moara visa democratizar a ciência e acelerar o desenvolvimento científico no Brasil. Este esforço colaborativo busca criar um ambiente de pesquisa mais inclusivo e acessível, alinhado com os princípios da Ciência Aberta.

■ 2.8 CONSIDERAÇÕES FINAIS

A criação de um repositório de códigos abertos é um passo importante para o avanço da Ciência Aberta no Brasil. Por meio da Rede Moara, busca-se promover a colaboração e o compartilhamento de conhecimentos entre pesquisadores e instituições, visando um ambiente de pesquisa mais inclusivo e eficiente.



Para o futuro, são propostas as seguintes ações para fortalecer e expandir a Rede Moara:

- **Estabelecimento de Parcerias:** buscar colaborações com outras redes de pesquisa, universidades e empresas de tecnologia para ampliar o alcance da Rede Moara;
- **Desenvolvimento de Capacitações:** oferecer treinamentos e workshops sobre práticas de compartilhamento de código, uso de ferramentas de repositório e desenvolvimento de software livre;
- **Incentivo à Participação:** criar programas de incentivo para que mais pesquisadores e instituições participem ativamente da Rede Moara, contribuindo com seus códigos e conhecimentos;
- **Melhoria da Infraestrutura:** continuar a desenvolver e aprimorar a infraestrutura tecnológica da Rede Moara, garantindo que ela suporte eficientemente o aumento de usuários e projetos;
- **Monitoramento e Avaliação:** implementar um sistema de monitoramento e avaliação contínua para medir o impacto da Rede Moara na comunidade científica e identificar áreas de melhoria;
- **Promoção da Cultura de Ciência Aberta:** realizar campanhas de sensibilização e divulgação para promover a importância da Ciência Aberta e do compartilhamento de conhecimento na pesquisa científica.

Essas proposições visam não apenas a consolidação da Rede Moara, mas também o fortalecimento da cultura de colaboração e transparência



na ciência. Com a implementação dessas ações, a Rede Moara poderá se tornar um pilar para o desenvolvimento científico e tecnológico no Brasil, contribuindo para a democratização do conhecimento e a inovação contínua.



REFERÊNCIAS

CHAPMAN, Daniel. **Joomla! 1.5x customization**: make your site adapt to your needs. Birmingham: Packt, 2009.

GAY, Joshua (ed.). **Free software, free society**: selected essays of Richard M. Stallman. Boston: GNU Press, 2002. Disponível em: <https://www.gnu.org/philosophy/fsfs/rms-essays.pdf>. Acesso em: 13 set. 2024.

HETHEY, Jonathan M. **GitLab repository management**. Birmingham: Packt, 2013.

LIMA, Antonio; ROSSI, Luca; MUSOLESI, Mirco. Coding together at scale: GitHub as a collaborative social network. **Proceedings of the International AAAI Conference on Web and Social Media**, Michigan, v. 8, n. 1, p. 295-304, 2014. DOI: 10.1609/icwsm.v8i1.14552. Disponível em: <https://ojs.aaai.org/index.php/ICWSM/article/view/14552>. Acesso em: 13 jun. 2024.

LOELIGER, Jon; MCCULLOUGH, Matthew. **Version control with Git**: powerful tools and techniques for collaborative software development. Cambridge: O'Reilly, 2012.

SILVEIRA, Sérgio Amadeu da. **Software livre**: a luta pela liberdade do conhecimento. São Paulo: Editora Fundação Perseu Abramo, 2004.

SPINELLIS, Diomidis. Git. **IEEE Software**, [S. l.], v. 29, n. 3, p. 100-101, maio/jun. 2012. DOI: 10.1109/MS.2012.61. Disponível em: <https://ieeexplore.ieee.org/document/6188603>. Acesso em: 13 jun. 2024.

STALLMAN, Richard. **O Projeto GNU**. [S. l.]: GNU.org, 24 jul. 2023. Disponível em: <https://www.gnu.org/gnu/thegnuproject.pt-br.html>. Acesso em: 13 jun. 2024.



COMO CITAR ESTE CAPÍTULO:

SHINTAKU, Milton; COSTA, Lucas Rodrigues; MOURA, Rebeca dos Santos; Vechi, Bernardo Dionízio. Repositório de códigos abertos. In: DRUCKER, Debora Pignatari; CIUFFO, Leandro; SAYÃO, Luis Fernando; SHINTAKU, Milton; VIDOTTI, Silvana Aparecida Borsetti Gregorio (org.) **Infraestruturas de suporte à Ciência Aberta**. Brasília, DF: Editora Ibict, 2025. p. 58-76. DOI: 10.22477/9786589167754.cap2.

