Repositórios de Dados: Objetivos, Funcionalidades e Alternativas

João Rocha da Silva*
joaorosilva@gmail.com
INESC TEC / Faculdade de Engenharia da Universidade do Porto
Porto, Portugal

RESUMO

Os repositórios de dados de investigação são cada vez mais uma peça essencial para o processo científico. Não só fomentam a reprodutibilidade das conclusões publicadas nos artigos científicos como assumem um papel crucial na atribuição de crédito aos criadores de dados, pois expõem ao público um trabalho de recolha, processamento e anotação que tanto tem de dispensioso como por vezes de invisível. A escolha de um software para suportar um repositório de dados deve ser guiada pelas necessidades das principais partes interessadas. Assim sendo, neste artigo discutem-se as principais funcionalidades desejáveis num repositório de dados, tanto do ponto de vista técnico (software e infraestrutura) como do ponto de vista político, nomeadamente no que diz respeito às garantias e compromissos a assumir pelas instituições que os alojam no sentido de permitir a sua certificação de acordo com a estratégia da European Open Science Cloud.

PALAVRAS-CHAVE

repositorios de dados, ciência aberta, e-Science, Estratégia para os Dados Abertos, Fundação para a Ciência e Tecnologia

1 INTRODUÇÃO

Os princípios FAIR para a gestão de dados de investigação especificam que os dados devem ser Findable (encontráveis), Accessible (acessíveis), Interoperable (interoeráveis) and Reusable (reutilizáveis) [14].

Recentemente, um grupo de peritos da Comissão Europeia debruçou-se sobre o problema dos dados FAIR, tendo concluído que os componentes essenciais de um ecossistema FAIR são: políticas, planos de gestão de dados (DMP), identificadores, normas e repositórios [5].

Neste artigo serão abordados alguns destes tópicos.

2 O PAPEL DO REPOSITÓRIO NA APLICAÇÃO DOS PRINCÍPIOS FAIR

Os princípios FAIR são linhas orientadoras para a melhoria dos processos de gestão de dados, abarcando portanto aspetos bastante mais genéricos do que a tecnologia que suporta um repositório de dados. Contudo, dado que este artigo é bastante focado na tecnologia, utilizaremos os princípios FAIR como guiões orientadores para elencar algumas das funcionalidades relevantes de um repositório de dados.

O primeiro dos princípios FAIR diz que os dados devem ser *Findable*, ou seja, devem ser fáceis de descobrir, tanto por humanos como por máquinas. Para tal, devem incluir (F1) um identificador global único e persistente, (F2) possuir metadados ricos, que por sua vez (F3) devem incluir claramente o identificador dos recursos que descrevem. O quarto e último aspeto, (F4), especifica que os tanto dados como metadados devem estar registados em recursos pesquisáveis.

De forma a facilitar a satisfação destes requisitos, uma plataforma de repositório deverá começar por proporcionar integração programática com um fornecedor de identificadores persistentes como por exemplo DOI[1] ou handle[2]. A vantagem para os gestores de repositório é a possibilidade de atribuir identificadores automaticamente aos conjunto de dados disponibilizados. Um identificador persistente funciona, em palavras simples, como um atalho para um programa num computador. Não é uma cópia do conjunto de dados, mas sim apenas um apontador para o local onde esse conjunto de dados está publicado (tipicamente, um repositório). Sendo assim, deve poder ser *de-referenciado*, ou seja, deve ser possível obter o conjunto de dados a partir do seu identificador. Para um ser humano, tal operação pode consistir apenas num simples clique numa ligação apresentada no seu navegador Web; para uma máquina, contudo, pode ser um pedido de rede à plataforma que gere os identificadores. Em ambos os casos, a entidade solicita o recurso por detrás daquele identificador; no caso do humano, a resposta será em HTML, código que o seu navegador pode interpretar de forma a construir uma página web legível para humanos. Para uma máquina, o formato devolvido poderá ser, por exemplo XML ou RDF (formatos de representação de informaçºao que as máquinas podem interpretar).

No caso do repositório de dados do INESC TEC[3], a FCT negociou um contrato para a aquisição de pacotes de identificadores em colaboração com a DataCite[4], que permitiu ao repositório emitir DOI para os conjuntos de dados despositados.

Para satisfazer os requisitos F2 e F3 é necessário que os registos de metadados associados a cada conjunto de dados obedeçam a um esquema normalizado, como é o caso do DataCite Schema[5]. Uma grande vantagem do uso de um esquema de metadados em conjunção com a atribuição de um DOI é que cada identificador ficará associado a uma ficha de metadados no momento da *cunhagem* do DOI. Tipicamente, as plataformas de emissão de identificadores não permitem sequer a sua cunhagem sem o preenchimento dessa ficha de metadados. Esta ficha ficará então guardada na plataforma emissora do DOI, e mesmo que o conjunto de dados deixe de estar

*Investigador no INESC TEC e Professor Auxiliar Convidado na Faculdade de Engenharia da Universidade do Porto

[1] Digital Object Identifier
[2] http://www.handle.net/
[3] Ligação: https://rdm.inesctec.pt
[4] Ligação: https://datacite.org/
[5] Ligação: https://schema.datacite.org/meta/kernel-4.3/

disponível no repositório, a ficha continuará disponivel para consulta na plataforma fornecedora do identificador (satisfazendo o requisito A2).

O segundo princípio FAIR impõe que os dados sejam *Accessible*, pois tanto dados como metadados devem ser acessíveis através através de identificadores, através de protocolos de comunicações normalizados (A1). Este ponto requer que o protocolo seja aberto, livre e universalmente implementável (A1.1), e que inclua procedimentos de autenticação e autorização, quando necessário (A1.2). O segundo requisito para assegurar a acessibilidade (A2) é que os metadados devem permanecer acessíveis, mesmo quando os dados deixam de o estar.

Para satisfazer o requisito A1, um repositório deverá suportar o processo de des-referenciação de identificadores persistentes, quer por máquinas quer por seres humanos. Quando se fala de protocolos de comunicações normalizados, falamos quase sempre do protocolo HTTP. Este é talvez o protocolo mais usado na web para transferência de informação, e satisfaz o requisito A1. Ele inclui dois mecanismos relevantes para a satisfação do requisito A1.1, chamados *Content Negotiation*[6], Authorization [7] e Access Authentication[8].

De uma forma muito breve, este mecanismo permite a um cliente (pode ser um navegador web ou um programa de computador) solicitar ao servidor (máquina que disponibiliza o registo de metadados de um determinado conjunto de dados) que envie a informação determinado formato, ao escrever esse pedido no cabeçalho do pedido HTTP (que pode ser visto como a secção "Destinatário" de um envelope). Ao especificar que pretende HTML, um navegador irá obter o código necessário para apresentar os dados sobre o conjunto de dados a um ser humano; se uma máquina solicitar XML ou RDF, irá obter um documento que é muito mais difícil de ler por parte de um ser humano, mas que a máquina interpreta corretamente. A beleza deste sistema de negociação é que permite obter essas diferentes representações a partir de um mesmo identificador, mudando apenas o cabeçalho da mensagem.

Apesar da autenticação básica oferecida pelo protocolo HTTP ser suficiente para satisfazer requisito A1.1, existem protocolos alternativos que podem ser adoptados caso hajam outros requisitos relevantes para a instituição que implementa um repositório de dados. Listam-se alguns protocolos abertos que podem ser relevantes neste cenário:

- **Autenticação federada**: O protocolo Shibboleth[9] é bastante usado nas instituições de ensino superior portuguesas para controlar o acesso a multiplos recursos através do portal de autenticação federada oferecido pela FCCN. Um repositório institucional deverá registar-se junto do Identity Provider relevante para permitir aos utilizadores autenticar-se com as mesmas credenciais que usam para aceder aos restantes recursos da sua instituição.
- **Autenticação delegada via outros providers**: O protocolo OAuth 2.0[10] permite aos utilizadores autenticar-se com as mesmas credenciais que utilizam em outros serviços como o ORCID[11], que pode ser útil para assegurar a autenticidade de auto-depósitos ou modificações feitas aos metadados pelos próprios criadores dos mesmos. Desta forma, podemos garantir que se uma determinada operação é realizada por alguém, se esse alguém se tiver autenticado com sucesso junto do ORCID, por exemplo. Este mecanismo foi implementado na plataforma de gestão de dados Dendro, para agilizar o registo de novos utilizadores e a sua autenticação. Desta forma, o identificador ORCID de cada utilizador ficará associado ao seu perfil, e indiretamente às operações efetuadas por esse utilizador no repositório.

O terceiro princípio FAIR diz que a gestão de dados deve ser *Interoperable*. Isto deve-se ao facto dos dados precisarem frequentemente de ser integrados com outros dados, e como tal têm que ser facilmente integráveis em fluxos de trabalho de processamento. Só assim podem ser facilmente armazenados e analisados de forma interoperável. Para assegurar a interoperabilidade, tanto dados como metadados devem (I1) usar uma linguagem formal, acessível e largamente aplicada para representação de conhecimento, (I2) devem usar vocabulários que, por sua vez, seguem os princípios FAIR, e (I3) devem incluir referências qualificadas para outros dados e metadados.

Talvez o princípio mais difícil de assegurar, a interoperabilidade implica a representação da informação constante no repositório em formatos que permitam a interpretação por sistemas externos. Isto quer dizer que os dados devem ser representados não só em formatos amigáveis como largamente suportados pelas bibliotecas de manipulação de dados mais usadas e disponíveis em código aberto. Para os dados, por exemplo, devem ser adotados os formatos livres de dependências e largamente suportados, como por exemplo XML, CSV ou TSV, em detrimento de formatos binários ou proprietários. Para os metadados, devem ser representados com recurso a standards bem definidos na comunidade, sejam eles formalizados como esquemas XML, ou ontologias no caso do repositório disponibilizar os registos de metadados como Linked Data.

Alguns exemplos destes esquemas incluem o amplamente utilizado Dublin Core ou o schema.org, um esquema de metadados para a web que reuniu a colaboração das principais tecnológicas (Google, Microsoft, Yahoo e Yandex)[12], que possam tanto ser formalizados como esquema XML como sob a forma de ontologias. Sejam qual forem os vocabulários seleccionados, devem eles próprios seguir os princípios FAIR, estando portando livremente acessíveis e serem devidamente documentados.

Por último, a interoperabilidade também diz respeito à inclusão de referencias para recursos relacionados. Por último, deverão ser incluídas nos registos de metadados de cada conjunto de dados referências para materiais relacionados. Exemplos de materiais relacionados incluem por exemplo um documento de dissertação ou tese, artigos resultantes da produção dos conjuntos em questão. Os esquemas ou ontologias utilizadas para a descrição devem portanto incluir descritores que permitam essas referências, como é o

[6]Ligação: https://www.w3.org/Protocols/rfc2616/rfc2616-sec12.html
[7]Ligação: https://www.w3.org/Protocols/HTTP/1.0/draft-ietf-http-spec.html#Authorization
[8]Ligação: https://www.w3.org/Protocols/HTTP/1.0/draft-ietf-http-spec.html#BasicAA
[9]Ligação: https://www.shibboleth.net/index/
[10]Ligação: https://oauth.net/2/
[11]Ligação: https://members.orcid.org/api/oauth2
[12]Ligação: https://schema.org/

exemplo do descritor `references` do Dublin Core[13] ou o descritor `citation` do schema.org[14].

O quarto e último dos princípios defende que os dados devem ser *Reusable*, pois o objetivo final dos princípios FAIR é fomentar a sua reutilização. Para serem Reutilizáveis, tanto dados como metadados devem (R1) ser descritos com uma grande variedade de atributos corretos e relevantes. Este princípio subdivide-se em três: (R1.1) devem ser publicados com uma licença de utilização clara e acessível, (R1.2) devem estar associados a informação de proveniência detalhada, sendo que esses dados e metadados devem satisfazer as normas relevantes de cada domínio (R1.3).

As licenças a escolher (recomendação R1.1) ficam ao critério da instituição de investigação e do criador dos dados. Contudo, para a disponibilização de dados de investigação em regime de *Open Data* é desejável a adoção de licenças que permitam a reutilização dos dados e metadados com o mínimo de restrições. Por forma a proporcuinar a outros a liberdade de reutilizar os dados para produzir trabalhos derivados e ao mesmo tempo assegurar a atribuição de crédito aos autores, as licenças Creative Commons (nomeadamente a CC BY-4.0)[15] é uma boa candidata na maioria dos casos de conjuntos de dados produzidos por projetos financiados por entidades públicas. Grande parte dos investigadores podem, no entanto, sentir dificuldades no momento da escolha da licença mais adequada para os conjuntos de dados que desejam publicar. É nesta altura que uma plataforma de repositório pode complementar o trabalho dos curadores, ao fornecer uma lista das licenças mais comuns, e ao mesmo tempo proporcionar tanto descrições curtas, claras e sucintas de cada licença, e ao mesmo tempo oferecer uma ligação rápida para a licença completa em linguagem jurídico-legal.

A recomendação R1.2 refere a necessidade de manter um historial que permita estabelecer a proveniência de um conjunto de dados. Enquanto que é relativamente simples manter um historial de modificações dentro de uma plataforma de repositório, o desafio é maior quando se considera que um conjunto de dados pode ser derivado de outro, tendo este último o seu próprio historial de modificações. Mais uma vez existe uma clara vantagem em usar Linked Data como formato de exposição de metadados (incluindo a proveniência) neste cenário. Para tal, é necessário que os repositórios de dados ofereçam um histórico de modificações em Linked Data que obedeça a ontologias como a PROV-O[7] (mais normativa pois é recomendação W3C) ou a PAV [2] (uma alternativa mais simples, também apelidada de *lightweight ontology*). Desta forma, quando uma terceira máquina des-referenciar o conjunto de dados derivado será também interrogado o repositório onde se encontra o conjunto de dados original, e assim sucessivamente se ele próprio também for um conjunto de dados derivado. Desta forma, torna-se transparente e automática a recuperação de todos os dados relativos ao historial de modificações e versões de um conjunto de dados, sem necessidade de indexar todos os recursos e sem que haja lugar à intervenção humana.

É bastante claro que domínios de investigação distintos poderão gerar conjuntos de dados muito diferentes também. Como tal, o processo de descrição dos mesmos deve também ser adaptado. Satisfazer a recomendação R1.3, que preconiza que os dados e metadados devem obedecer às normas vigentes em cada domínio requer suporte por parte do software de repositório, que deve ser flexível ao ponto de permitir a parametrização de múltiplos esquemas de metadados para distintos domínios. Deverá também permitir a escolha e combinação de múltiplos descritores (genéricos e específicos do domínio do conjunto de dados em depósito) aquando do preenchimento do registo de metadados, sejam esses descritores genéricos ou específicos de domínio. É preciso também não esquecer o trabalho fundamental dos curadores no apoio institucional aos seus investigadores, pois a escolha dos descritores mais adequados a cada conjunto de dados

3 CERTIFICAÇÃO

Para além dos princípios FAIR, que se apresentam como linhas gerais para aquilo que um processo de gestão de dados deve proporcionar, surgiram recentemente os princípios TRUST [8]. Estes focam-se no problema da confiança nas infraestruturas, organizações e qualidade dos dados de investigação. Estes princípios oferecem, de acordo com os autores, uma *framework* de discussão para ajudar a melhorar a confiança de todas as partes interessadas no processo de gestão de dados. Estas partes interessadas vão desde os próprios investigadores, passando pelas instituições académicas ou de ciência que os acolhem, as entidades financiadoras, entre outros [13].

De acordo com o Guião Estratégico de Implementação (GEI) da EOSC (European Open Science Cloud), a certificação de repositórios de dados é uma parte essencial na visão a longo prazo da Comissão Europeia. Os processos de certificação são exaustivos e requerem não só excelência técnica como também um compromisso a longo prazo das instituições que os suportam.

3.1 Data Seal of Approval

3.2 nestorSEAL

3.3 ISO 16363

3.4 Core Trust Seal

De um ponto de vista mais económico e de soberania dos dados, a certificação de repositórios assumirá um papel cada vez mais importante no funcionamento de uma instituição académica ou de investigação. Existem diversas razões, de entre as quais se destaca o crescente número de entidades financiadoras e publicações de renome que exigem a disponibilização dos dados-base e do código-fonte que sustenta as publicações. Neste cenário, a qualidade dos repositórios onde tais recursos são disponibilizados terá necessariamente que ser elevada. Desta forma, a certificação tornar-se-à assim uma pedra fundamental no suporte aos projetos de investigação, pois as instituições terão que assegurar aos seus investigadores a satisfação dos presentes e futuros requisitos de financiadores aquando da escrita dos planos de gestão de dados a anexar às propostas de financiamento. Essas instituições terão então duas opções para garantir os requisitos : ou embarcam num processo de certificação dos seus próprios repositórios institucionais, ou subcontratam o processo de depósito e curadoria de dados a um repositório certificado.

[13]Ligação: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/terms/references/
[14]https://schema.org/citation
[15]Ligação: https://creativecommons.org/licenses/by/4.0/

4 AUTOMATIZAÇÃO DA GESTÃO DE DADOS

Qualquer processo de gestão de dados de investigação deve começar com o desenho de um Plano de Gestão de Dados (ou DMP, do inglês Data Management Plan). Um DMP é um documento tipicamente escrito e aprovado antes do início de um projeto de investigação. Em alguns casos, é já requisito obrigatório em chamadas de propostas de projetos de investigação suportados por fundos públicos [10]) e que especifica todos os aspetos relacionados com a gestão dos dados produzidos ao longo do projeto [].

Entre outros aspetos, que dados serão produzidos no contexto de um projeto, que metadados serão anexados aos dados publicados, onde eles irão ficar disponíveis após o fim do projeto, entre outros pontos. No Reino Unido, o Digital Curation Centre disponibiliza tanto um guião [6] para a escrita deste tipo de documentos como uma ferramenta online, o DMP Online [16], que assiste os investigadores e curadores na elaboração dos seus DMP, assistidos por uma base de dados de templates de DMP aceites por diferentes entidades financiadoras.

Com a grande quantidade de dados de investigação gerada diariamente surge a necessidade de acompanhar e monitorizar a implementação de DMP, mas de forma automática. Três exemplos simples de operações frequentes, repetitivas e facilmente automatizáveis são:

- Simples verificação da disponibilidade dos dados depositados
- Controlo de qualidade dos metadados associados aos conjuntos de dados acabados
- Verificação das licenças associadas aos conjuntos de dados

Para lidar com estas operações e outras de forma mais automática surgiram os chamados DMP Accionáveis por Máquina (maDMP, do inglês *Machine-Actionable DMP*). Desta forma é possível assim libertar os curadores de dados para outras operações menos automatizáveis, como o suporte direto aos investigadores ou atividades de formação.

Como preconizado pelos autores das 10 regras para maDMP [9], deve ser possível aos sistemas informáticos levar a cabo ações em nome das partes interessadas no processo de gestão de dados de investigação. Operações dadas como exemplo incluem a recolha automática de informação administrativa relevante para anotação dos recursos produzidos. Exemplos dessa informação incluem, por exemplo, as recuperação de referências aos financiadores dos projetos, os currículos dos autores dos conjuntos de dados assim como as informações corretas das instituições de acolhimento dos investigadores.

Esta necessidade de suporte a automação e integração tem implicações vastas do ponto de vista do desenho e funcionalidades de um repositório de dados, na medida em que a solução deve disponibilizar uma API (*Application Programming Interface*) completa e bem documentada, de forma a permitir a sistemas externos executar estas operações sem a necessidade de intervenção humana.

[16]Ligação:https://dmponline.dcc.ac.uk

4.1 O papel da interoperabilidade na descoberta de conjuntos de dados

Ao mesmo tempo, a integração com diretórios de dados deve ser transparente e assente em protocolos de interoperabilidade standard. Estes agregadores são portais que indexam o conteúdo dos repositórios para facilitar a pesquisa e descoberta de recursos neles contidos. Exemplos destes agregadores são, por exemplo, o portal re3data da DataCite [17] ou o Dataset Search da Google[18].

Para facilitar a indexação dos conteúdos dos repositórios, estes implementam suporte ao protocolo OAI-PMH (Open Access Initiative Protocol for Metadata Harvesting). Este protocolo expõe os metadados de todos os registos no repositório de forma paginada, para que seja lida sequencialmente e indexada; contudo, não possibilita a pesquisa e recuperação de registos específicos por termos contidos nos registos de metadados, por exemplo. Para conseguir tal funcionalidade é necessário proceder à inserção de todos os documentos num índice de pesquisa como é o caso do Lucene ou Solr. Uma consequência óbvia é a necessidade de manutenção de cópias dos registos e atualizações periódicas de todo o índice—uma operação lenta e dispendiosa em termos de recursos informáticos, que coloca todo o custo de manutenção do serviço do lado dos clientes.

Uma situação oposta ocorre com a adoção de Dados Ligados (*Linked Data*) na representação de metadados para conjuntos de dados de investigação. Esta representação de dados vem eliminar a necessidade de indexação periódica de conteúdos e melhorar muito a precisão das pesquisas, caso os dados sejam estruturados de acordo com uma ou mais ontologias. Uma ontologia é definida, na Ciência da Informação, como uma *especificação de uma conceptualização*[3], que permite às máquinas, por exemplo, interpretar o significado dos tipos de entidades representadas num sistema e as relações que se podem estabelecer entre elas.

A representação via Linked Data permite uma maior descentralização da carga de pesquisas sobre os servidores onde os repositórios estão alojados. Reduz também a necessidade de indexação porque é possível correr pesquisas nos próprios servidores sem ter que primeiro indexar todo o seu conteúdo e permite aos clientes do serviço recuperar conjuntos de dados com muito mais precisão do que nos casos em que se usa um índice de pesquisa simples baseado em palavras-chave, pois é possível especificar critérios muito detalhados sob a forma interrogações SPARQL.

A disponibilização de Linked Data por parte dos repositórios tem, contudo, um custo para quem hospeda esses repositórios. As consultas SPARQL são exigentes do ponto de vista computacional e, se um número elevado de utilizadores (máquinas ou humanos) aceder ao website ao mesmo tempo, é fácil afetar os tempos de resposta da infraestrutura. Isto é especialmente relevante quando se considera a funcionalidade de pesquisa federada, onde múltiplos servidores podem ser contactados para conseguir obter os resultados pretendidos por uma única pesquisa iniciada em qualquer dos nós da federação.

O futuro dos repositórios de dados deve passar assim por uma solução híbrida, onde servidores e clientes partilham informação para possibilitar também a partilha do esforço de interrogação e descoberta de dados. Soluções como a Linked Data Fragments[12]

[17]Ligação: https://www.re3data.org/
[18]Ligação: https://datasetsearch.research.google.com/

propõem a combinação de interrogações sobre *endpoints* tradicionais (no servidor do repositório que expõe Dados Ligados) com a interrogação a dados locais a cada cliente, reduzindo assim a carga sobre o servidor.

4.2 Repositórios como plataformas de computação

Cada vez mais, o valor de um repositório de dados reside na facilidade de reutilização dos dados nele contidos. Tecnologias recentes como os Research Notebooks e a Containerização permitem tornar os algoritmos mais portáteis portáteis, de forma a poderem ser executados remotamente sobre os dados depositados num repositório. O papel deste torna-se assim muito mais do que o de simples armazenista, indexador e recuperador de pacotes de dados.

É muito comum descobrir conjuntos de dados que, apesar de publicados e devidamente descritos com metadados completos, não são interpretáveis ou reproduzíveis quando um investigador tenta repetir os passos experimentais reportados num artigo de investigação[].

Os Research Notebooks são cadernos de laboratório interactivos que encerram em si os dados-base, código de processamento que transforma os dados base em dados processados, e até mesmo visualizações e gráficos gerados na hora com base nos dados e no código do Notebook.

Os Notebooks vêm também possibilitar suportar uma dos maiores mudanças no paradigma de publicação científica dos últimos tempos. A combinação elegante de texto, trechos de código e visualizações apelativas permitiu a criação dos chamados "Web Journals" como por exemplo o distill.pub. Estes distinguem-se das publicações convencionais, ao incorporar visualizações interativas no próprio texto dos artigos científicos, fomentando a experimentação através da manipulação de parâmetros de entrada dos algoritmos, produzindo um feedback visual imediato aos leitores. Toda a computação tem, no entanto, que ser executada em tempo real, o que implica a montagem de uma réplica do ambiente de processamento por detrás do portal web que suporta o journal.

Com o problema do empacotamento do processamento e dos dados a ser abordado pelos Notebooks, resta o problema de onde realizar a execução desses algoritmos. Nem todos os conjuntos de dados se prestam a ser transportados pela rede em tempo útil para a execução de um determinado algoritmo, pois em certas disciplinas o seu tamanho pode ascender a centenas de gigabytes ou mesmo terabytes. Assim sendo, é necessário levar a computação até aos dados em vez de ter que transmitir os dados até ao local onde se realiza a computação. Este tipo de computação que ocorre junto dos dados é um paradigma é implementado em portais de computação como o D4Science [19] ou o EUDAT B2Stage[20]. Com a esperada descentralização e interoperabilidade que se espera dos repositórios de dados defendida pelo GEI da EOSC, espera-se que cada repositório seja capaz de oferecer capacidade de computação local a pequena escala e junto das fontes de dados. Este cenário é mais próximo do *edge computing*, um termo mais vulgarmente associado às aplicações IoT. Por contraponto ao *cloud computing*, onde todos os recursos se encontram na nuvem e os dados têm que ser cuidadosamente aproximados dos nós de computação (mesmo em termos geográficos) para reduzir tráfego na rede, o *edge computing* defende que cada nó da rede deve ter uma pequena mas importante capacidade de computação para executar operações à medida que os dados são produzidos ou atualizados.

Quando se fala de distribuição de computação e reproducibilidade, surge imediatamente a necessidade de replicar também todo o ambiente de execução no qual essa computação é executada. Sem uma solução de virtualização, isso implica a instalação de um sistema operativo e de todas as dependências relevantes para a execução do código relevante. Este é um processo manual e dispendioso que torna claramente necessário assegurar a portabilidade do ambiente de execução, tornando-o instalável automaticamente em qualquer nó com o mínimo de dependências.

A containerização é uma tecnologia de virtualização que, quando comparada com as máquinas virtuais convencionais, elimina a necessidade de virtualizar um sistema operativo para executar um determinado programa, substituindo essas dependências por um ambiente de execução (ou *runtime*) próprio. O exemplo mais comum de ambiente de virtualização é o Docker[21]. Esta eficiência torna possível a um repositório de dados executar algoritmos directamente sobre os dados nele depositados e retornar apenas os resultados aos clientes que pedem esses processamentos, em vez de exigir a descarga dos dados para o cliente. Talvez a funcionalidade mais interessante para um repositório de dados é a possibilidade de especificar todos os passos de instalação de dependências necessárias à execução dos algoritmos de processamento de dados, reduzindo em larga medida os problemas de replicação do ambiente de execução de determinado algoritmo.

4.3 A *blockchain* e as suas aplicações aos Dados Abertos

Num futuro de repositórios de dados descentralizados e interligados num ambiente descentralizado será necessário conceber soluções para assegurar a autenticidade da informação colocada em cada nó da federação. Em particular, a criação de um *ledger* de transações distribuído, como por exemplo uma *blockchain* envolvendo os repositórios dda federação, reduz a dependência de entidades terceiras—como os emissores de certificados—para assegurar a autenticidade dos dados. Este tema já está a ser alvo de investigação, com o objetivo de descobrir formas de utilização da blockchain para assegurar a validade dos resultados de operações sobre dados abertos [11]. É portanto claro o potencial de aplicação deste tipo de tecnologias aos diversos aspetos da gestão de dados abertos, destacando-se dois pontos:

- **Proveniência**: Através de um registo distribuído de modificações feitas a dados e metadados será possível assegurar a sua autenticidade e não-repúdio relativamente à autoria de quaisquer operações realizadas sobre os dados. Esta prática deve estar presente antes, durante e após a publicação dos conjuntos de dados; quando aplicada antes da publicação, permite rastrear todas as modificações feitas ao longo do tempo, para reduzir situações de *p-hacking*[4], por exemplo. Após a publicação, um registo distribuído de alterações permite aos autores continuar a modificar um conjunto de

[19] Ligação: https://www.d4science.org
[20] Ligação: https://www.eudat.eu/b2stage
[21] Ligação: https://www.docker.com/

dados após a publicação do conjunto de dados considerado "final", de forma completamente transparente. Uma outra vantagem é a resiliência do sistema, pois mesmo que o repositório onde um conjunto de dados está publicado seja desligado, esse historial de modificações continuará registado nos restantes nós da *blockchain*.

- **Fiabilidade de resultados**: Regra geral, um valor só é registado na *blockchain* se existir um consenso na rede de computadores que colaboram nessa *blockchain*. Este mecanismo pode ser usado para atestar a reprodutibilidade das operações de transformação realizadas sobre um conjunto de dados. Neste caso em particular, só após um determinado número de nós (dependendo da política em vigor na *blockchain*) levarem a cabo as mesmas operações e confirmarem os resultados é que é criado o *consenso* necessário à escrita para a *blockchain*, passando aquela operação a fazer parte do historial de operações sobre aquele conjunto de dados.

5 VISÃO GERAL DE FUNCIONALIDADES

Existem diferentes alternativas para a montagem de um repositório de dados. Uma análise comparativa produzida pelo projeto Dataverse[22] apresenta uma comparação das principais plataformas de repositórios de dados, tendo em conta três grupos de facetas: Funcionalidades de Software, Controlo/Organização e Conteúdo. Uma outra comparação [1] discute em mais profundidade algumas características técnicas de diversas plataformas. Esta análise foca-se apenas nas alternativas disponíveis em regime de código aberto e instaláveis localmente, de forma a poder retirar conclusões relativamente ao modelo de dados por detrás de cada solução de software.

Tão importante como comparar funcionalidades oferecidas pelas plataformas é estudar quais funcionalidades são mais valorizadas pelos utilizadores destas plataformas. Para responder a esta questão, o Grupo de Interesse das Plataformas de Gestão de Dados de Investigação (RPRD IG) da Research Data Alliance (RDA) compilou uma Matriz de Casos de Uso e Requisitos Funcionais para as Plataformas de Gestão de Dados de Investigação. Esta matriz resultou de um inquérito a 11 grupos que apresentaram os seus casos de uso para repositórios de dados de investigação. Cada um destes grupos classificou a importância de 26 requisitos vulgarmente associados a um repositório de dados. No final, foi atribuída uma pontuação a cada um dos requisitos funcionais, com base na importância que a totalidade dos grupos lhe atribuiu no contexto do seu caso de uso.

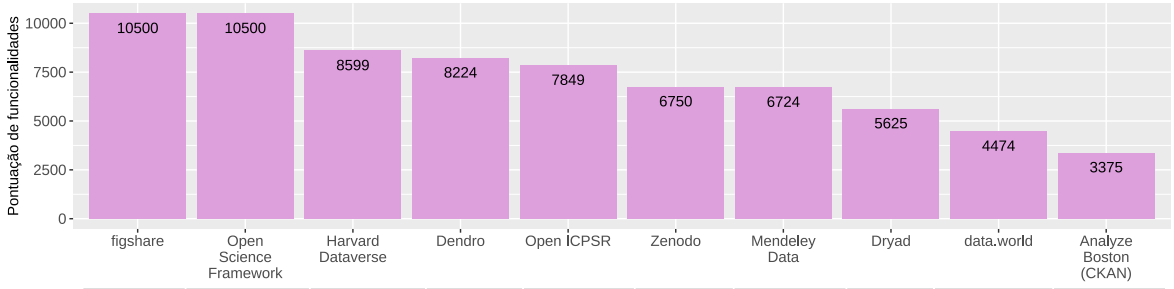
De forma a comparar de forma sucinta algumas das plataformas mais utilizadas, é necessário considerar não só a quantidade e variedade de funcionalidades como também a sua importância para a maioria dos utilizadores. Assim, na Figura **??** apresenta-se não só uma pontuação de funcionalidades, como também algumas das principais características dessas plataformas. Nesta figura, apresenta-se primeiramente um gráfico de barras que ordena as plataformas de acordo com uma "Pontuação de funcionalidades". Esta foi determinada associando manualmente os requisitos funcionais da RDA às características listadas na comparação de repositórios apresentada pela DataVerse, e calculando o produto de cada relação estabelecida pela importância dada ao requisito pela RDA. Desta forma, foi possível atribuir diferentes *pesos* às características desta última comparação [23].

A figura apresenta também uma tabela com algumas características mais relevantes de cada plataforma:

- **Código Aberto**
  Na primeira linha distinguem-se as soluções que estão disponíveis em regime de código aberto daquelas em que não é o caso. O facto do código de uma solução estar disponível em acesso aberto tem inúmeras vantagens, desde que haja uma comunidade disposta a manter esse software. Exemplos são uma maior segurança, pois o facto do código ser aberto facilita a correção rápida de falhas de segurança por parte da comunidade, ao invés de ter que esperar pela correção por parte da equipa de uma empresa. Qualquer pessoa tem também a liberdade de modificar e melhorar o software para satisfazer uma determinada necessidade, sendo que esses melhoramentos poderão potencialmente ser incluídos no tronco comum da solução, caso se verifique que o requisito se verifica em outras comunidades e os desenvolvedores assim o entenderem. Esta flexibilidade pode proporcionar um ritmo e velocidade de desenvolvimento mais elevado do que as soluções proprietárias, que têm equipas de desenvolvimento limitadas; contudo, a própria comunidade de desenvolvimento é responsável pela manutenção da qualidade do código. O modelo de código aberto não invalida a hipótese das entidades que suportam um repositório realizarem alterações pagas ao código, num cenário semelhante à consultoria de implementação do repositório. Nestes casos, o "produto" em si não é o software, mas sim o serviço de instalação e personalização da solução de código aberto numa determinada instituição. De qualquer forma, o código desenvolvido no contexto destas implementações pode também ser incluído no tronco comum, beneficiando toda a comunidade que utiliza esse software. É também normal encontrar nos projetos de software open-source um conjunto de fóruns públicos onde os utilizadores podem propor melhorias ao software e reportar *bugs*.
- **Grátis** Quase todas as soluções em análise são completamente grátis ou oferecem um nível de serviço base grátis. Os sistemas que funcionam com base na cloud incluem normalmente uma capacidade de armazenamento de dados limitada, sendo que os utilizadores podem adquirir um pacote de armazenamento maior se necessitarem. As soluções que podem ser instaladas localmente apenas dependem da capacidade de armazenamento do servidor no qual são instaladas.
- **Número de conjuntos de dados, ficheiros e utilizadores** O número de recursos geridos pelas plataformas e o número de utilizadores que confiam nessas plataformas para a sua gestão de dados é uma importante métrica da sua maturidade.

[22] Ligação: https://dataverse.org/blog/comparative-review-various-data-repositories

[23] Os dados base deste estudo estão disponíveis no repositório GitHub deste relatório, como materiais auxiliares. Ligação: https://github.com/silvae86/repositories-paper-eda



	figshare	Open Science Framework	Harvard Dataverse	Dendro	Open ICPSR	Zenodo	Mendeley Data	Dryad	data.world	Analyze Boston (CKAN)
Código Aberto	Não	Sim	Sim	Sim	Não	Sim	Não	Sim	Não	Sim
Grátis ("Free To Use")	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Não	Sim	Sim
Núm. Conjuntos Dados (6/2017)	N/D	12709	21846	N/D	243	4397	1500	15325	N/D	116
Num. Ficheiros (5/2017)	800000	885531	197844	N/D	N/D	191837	N/D	50000	N/D	426
Num. Utilizadores (5/2017)	N/D	56000	13916	N/D	N/D	N/D	N/D	N/D	N/D	N/D
Certificação	Não	Não	Não	Não	Sim (DSA+WDS)	Não - ISO no futuro	Sim	Não	Não	Não
Infraestrutura	SaaS	SaaS + Local	SaaS + Local	Local	SaaS	SaaS	SaaS	SaaS	SaaS	SaaS + Local

Figura 1: Análise comparativa de diferentes plataformas de *software* para repositórios de dados

- **Certificação** A certificação de repositórios continuará a assumir relevância aquando da adoção de uma solução de repositório. No caso das soluções SaaS, a certificação do repositório pode beneficiar todas as instituições que contratem os serviços dessas plataformas, pois todos os seus investigadores poderão mencionar esse aspeto na escrita dos DMP que anexam às suas propostas de projetos. Ao mesmo tempo, as soluções SaaS têm todo o interesse em procurar a certificação, pois pode oferecer mais valor aos seus clientes e pode ser determinante para a adoção das suas soluções. No caso das alternativas instaladas e mantidas localmente, e apesar de uma solução de software em particular não a

Por fim, é necessário considerar o modelo de hospedagem ou infraestrutura do repositório, que está geralmente ligado ao modelo de negócio das entidades que o suportam. Neste domínio, existem as soluções SaaS[24]Software as a Service, que podem ser vistos como "chave-na-mão". Nestes modelos, toda a infraestrutura e serviços por detrás da página web do repositório (alojamento, manutenção, assistência técnica, curadoria ou atribuição de identificadores) faz parte de um pacote oferecido às instituições interessadas nesse repositório. Em alternativa, apresentam-se as soluções alojadas localmente, que diferem das soluções SaaS na medida em que exigem instalação e manutenção por parte de uma equipa de TI interna à organização. Da mesma forma, o trabalho de curadoria na plataforma também tem que ser levado a cabo inteiramente pelos elementos da organização e geralmente não existe nenhuma obrigação de assistência por parte dos desenvolvedores do software.

6 O FUTURO DOS REPOSITÓRIOS DE DADOS

7 DISCUSSÃO E CONCLUSÕES

Os princípios FAIR para a gestão de dados de investigação, e mais recentemente os princípios TRUST, assumem-se como as linhas orientadoras para o desenho de fluxos de trabalho de gestão de dados, nos quais o repositório se assume como a pedra fundamental. Enquanto que os primeiros se focam nas boas práticas necessárias para a reutilização de conjuntos de dados, os segundos orientam a construção e manutenção de repositórios de dados confiáveis.

Os repositórios têm vindo a assumir um papel cada vez mais importante na divulgação de ciência. Enquanto arquivos de dados de investigação, o seu valor é cada vez mais medido pela sua capacidade de fomentar a reutilização dos recursos neles depositados.

[24]

Desta forma, o simples armazenamento e descrição dos documentos neles depositados não chega. Os repositórios devem permitir a publicação de dados e metadados de forma interoperável, de forma a suportar a descoberta desses recursos, tanto por seres humanos como por máquinas, de forma automática.

Para além da descoberta e interrogação automática dos seus conteúdos por parte de sistemas externos, os repositórios devem oferecer uma interface de interação programática—ou API—completa. Só desta forma poderão suportar a execução de Planos de Gestão de Dados Acionáveis por Máquinas, ou maDMP. Estes modelos vão além dos documentos convencionais, ao servirem de especificação legível, auditável e executável por máquinas das práticas a seguir durante a gestão de dados. Por exemplo, um maDMP pode especificar que determinados conjuntos de dados devem estar disponíveis, devem incluir um DOI nos seus identificadores e os seus metadados devem obedecer a determinada norma: o repositório onde esses dados estiverem depositados tem que ser capaz de responder com essa informação—via API—quando o sistema externo encarregado de assegurar o cumprimento do maDMP o interrogar sobre a presença desses elementos.

A capacidade de versionamento consequente auditoria transparente às alterações feitas a um conjunto de dados é também uma funcionalidade essencial nos repositórios atuais. Ambas são prérequisito para a rastreabilidade desses dados, permitindo não só a sua evolução contínua—mesmo após a publicação como anexo a um artigo científico. Ao mesmo tempo, este registo permite atribuir aos autores dessas modificações o devido crédito.

A atribuição de crédito pelo trabalho de produção e descrição correta de conjuntos de dados é uma das questões mais relevantes para a motivação dos investigadores em todo este processo. As plataformas de repositório desempenham aqui também um papel essencial, pois são elas que devem manter a informação de quem criou ou modificou cada conjunto de dados e seus metadados que permitirá o cálculo de uma métrica de mérito. A inclusão dessa métrica de publicação nos critérios de avaliação institucional dos investigadores torna-se assim tecnicamente possível.

A oferta de capacidades de computação sobre os dados depositados num repositório requer a ligação entre os dados e plataformas de computação como por exemplo os Jupyter Research Notebooks. Desta forma, o repositório guardará não só os dados mas também os processos de análise desses dados que sustenta às conclusões publicadas. Torna-se também mais fácil para terceiros re-executar esses passos, pois reduz-se o número de dependências a instalar para recuperar o seu contexto de execução.

Por último, e em jeito de conclusão, a tecnologia desempenha um papel essencial no suporte à reprodutibilidade dos resultados apresentados nas publicações. Contudo, essa mesma tecnologia não pode nunca substituir o papel dos curadores de dados. O investimento na implementação de um repositório deve assim ser alicerçado num igual investimento na formação dos responsáveis pela gestão dos dados de investigação. São esses peritos que levam a cabo o suporte aos investigadores e que conseguem dessa forma criar um clima de confiança no processo de gestão de dados e por consequência forma salientar a proposta de valor de qualquer solução tecnológica que o sustente.

REFERÊNCIAS

[1] Ricardo Carvalho Amorim, João Aguiar Castro, João Rocha da Silva, and Cristina Ribeiro. 2017. A comparison of research data management platforms: architecture, flexible metadata and interoperability. *Universal Access in the Information Society* 16, 4 (2017), 851–862. https://doi.org/10.1007/s10209-016-0475-y

[2] Paolo Ciccarese, Stian Soiland-Reyes, Khalid Belhajjame, Alasdair JG Gray, Carole Goble, and Tim Clark. 2013. PAV ontology: provenance, authoring and versioning. *Journal of biomedical semantics* 4, 1 (2013), 37.

[3] Thomas R Gruber. 1995. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies* 43, 5-6 (1995), 907–928.

[4] Megan L. Head, Luke Holman, Rob Lanfear, Andrew T. Kahn, and Michael D. Jennions. 2015. The Extent and Consequences of P-Hacking in Science. *PLOS Biology* 13, 3 (03 2015), 1–15. https://doi.org/10.1371/journal.pbio.1002106

[5] S Hodson, S Collins, F Genova, N Harrower, S Jones, L Laaksonen, D Mietchen, R Petrauskaité, and P Wittenburg. 2018. Turning FAIR into reality: Final report and action plan from the European Commission expert group on FAIR data. *European Union: Brussels, Belgium* (2018).

[6] Sarah Jones. 2011. How to Develop a Data Management and Sharing Plan. https://www.dcc.ac.uk/guidance/how-guides/develop-data-plan. Accessed: May 2020.

[7] Timothy Lebo, Satya Sahoo, Deborah McGuinness, Khalid Belhajjame, James Cheney, David Corsar, Daniel Garijo, Stian Soiland-Reyes, Stephan Zednik, and Jun Zhao. 2013. Prov-o: The prov ontology. *W3C recommendation* 30 (2013).

[8] Dawei Lin, Jonathan Crabtree, Ingrid Dillo, Robert R. Downs, Rorie Edmunds, David Giaretta, Marisa De Giusti, Hervé L'Hours, Wim Hugo, Reyna Jenkyns, Varsha Khodiyar, Maryann E. Martone, Mustapha Mokrane, Vivek Navale, Jonathan Petters, Barbara Sierman, Dina V. Sokolova, Martina Stockhause, and John Westbrook. 2020. The TRUST Principles for digital repositories. *Scientific Data* 7, 1 (2020), 144. https://doi.org/10.1038/s41597-020-0486-7

[9] Tomasz Miksa, Stephanie Simms, Daniel Mietchen, and Sarah Jones. 2018. Ten simple rules for machine-actionable data management plans (preprint). https://doi.org/10.5281/zenodo.1172673

[10] National Science Foundation. 2011. Grants.Gov Application Guide A Guide for Preparation and Submission of NSF Applications via Grants.gov. (2011).

[11] Bruno Tavares, Filipe Figueiredo Correia, and André Restivo. 2020. Trusted Data Transformation with Blockchain Technology in Open Data. In *Distributed Computing and Artificial Intelligence, 16th International Conference, Special Sessions*, Enrique Herrera-Viedma, Zita Vale, Peter Nielsen, Angel Martin Del Rey, and Roberto Casado Vara (Eds.). Springer International Publishing, Cham, 213–216.

[12] Ruben Verborgh, Miel [Vander Sande], Olaf Hartig, Joachim [Van Herwegen], Laurens [De Vocht], Ben [De Meester], Gerald Haesendonck, and Pieter Colpaert. 2016. Triple Pattern Fragments: A low-cost knowledge graph interface for the Web. *Journal of Web Semantics* 37-38 (2016), 184 – 206. https://doi.org/10.1016/j.websem.2016.03.003

[13] A Whyte and S. (Eds) Allard. 2014. How to Discover Research Data Management Service Requirements. https://www.dcc.ac.uk/guidance/how-guides/how-discover-requirements. Accessed: May 2020.

[14] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* 3 (2016).