
**UNIVERSIDADE FEDERAL DA PARAÍBA**
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

# UMA ABORDAGEM BASEADA EM DLT PARA GARANTIA DA FIXIDEZ DE REPOSITÓRIOS DIGITAIS CONFIÁVEIS

FILIPY GALIZA SOARES

JOÃO PESSOA - PB
2021

**UNIVERSIDADE FEDERAL DA PARAÍBA**
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

# UMA ABORDAGEM BASEADA EM DLT PARA GARANTIA DA FIXIDEZ DE REPOSITÓRIOS DIGITAIS CONFIÁVEIS

**FILIPY GALIZA SOARES**

Dissertação de Mestrado apresentada como requisito parcial para obtenção do título de Mestre em Informática pelo Programa de Pós-Graduação em Informática da Universidade Federal da Paraíba – UFPB.

Orientador: **Dr. Rostand Edson Oliveira Costa**

JOÃO PESSOA - PB
2021

**Catalogação na publicação**
**Seção de Catalogação e Classificação**

S676a Soares, Filipy Galiza.
Uma abordagem baseada em DLT para garantia da fixidez de repositórios digitais confiáveis / Filipy Galiza Soares. – João Pessoa, 2021.
106 f. : il.

Orientação: Rostand Edson Oliveira Costa.
Dissertação (Mestrado) – UFPB/CI.

1. Preservação digital. 2. Fixidez. 3. Árvore – Merkle. 4. DLT. 5. RDC. I. Costa, Rostand Edson Oliveira. II. Título.

UFPB/BC CDU 004.6-049.34(043)

Elaborado por WALQUELINE DA SILVA ARAUJO – CRB-15/514

*Dedico este trabalho à minha saudosa avó, Leocesar Leiga Lira (in memoriam), e à minha mãe, Maria Livalda de Galiza, fontes de educação e exemplos de resiliência e perseverança.*

# AGRADECIMENTOS

Agradeço à minha avó, Leocesar Leiga Lira (*in memoriam*), pelo seu exemplo de vida e de amor à essa, emanando uma alegria inabalável, uma perseverança invejável e humildade inigualável. E apesar de sua limitada e sofrida experiência de vida, demonstrava seus humildes desejos em ampliar suas experiências e conhecimentos e sempre me apoiou nos estudos e se dispôs a escutar e orientar.

Aos familiares que estimularam e me apoiaram em minhas jornadas, especialmente à minha mãe, Maria Livalda de Galiza, que sempre se dispôs a estar do meu lado para ajudar no que fosse preciso, mesmo quando não possível.

Aos professores do PPGI que conduziram meus aprendizados no mestrado, principalmente o professor Rostand Edson Oliveira Costa por sua compreensão e orientação e por me incluir nas atividades do GT-RAP como potencial orientando, o que permitiu o desenvolvimento deste trabalho. Agradeço também ao professor Guido Lemos de Souza Filho, que me apresentou ao trabalho do GT-RAP e, como sempre solícito, prontamente me indicou ao professor Rostand.

À colega e parceira de trabalho no Campus V da UEPB, Viviane Ribeiro Coutinho Freitas Oliveira, por sua contínua disponibilidade e aos professores do curso de arquivologia do mesmo campus, Josemar Henrique de Melo e Sânderson Lopes Dorneles, pelo apoio, estímulo e orientação em suas respectivas áreas de trabalho e à própria Instituição, pela oportunidade de cessão de horas de trabalho ao curso do mestrado.

Ao colega da ETE Aderico Alves de Vasconcelos, Humberto Nunes Filho, por suas recomendações ao curso do PPGI da UFPB.

À minha namorada, Lisandra Caroline de Araújo Lima Teixeira, amigos, parentes e demais que apoiam direta ou indiretamente o desenvolvimento deste trabalho, em especial a comunidade Software Livre, que provê ferramentas essenciais para isso.

# RESUMO

As novas facilidades proporcionadas pelos documentos em formato digital também trazem dificuldades, traduzidas nos desafios de garantir a integridade e autenticidade desses documentos continuamente, sabendo que esse formato é bastante sensível à falhas em alguma das camadas de um sistema computacional. Organizações e normas internacionais traçam recomendações sobre fixidez para guiar os Repositórios Digitais Confiáveis (RDC) em suas missões de tentar garantir a integridade e autenticidade dos documentos ali armazenados pelo tempo que for definido (ou indefinidamente). O uso de funções criptográficas de *hash* possibilita a verificação de integridade de dados e sua posterior autenticidade. Porém, para se confiar na integridade atestada por uma função de *hash* também é necessário confiança no valor *hash* de referência, que eventualmente está vulnerável às mesmas ameaças dos dados os quais representa. Partindo dessa preocupação, este trabalho propõe uma abordagem em que seja feito um encadeamento em Árvores de Merkle dos *hashes* de inúmeros objetos digitais de um RDC e os valores raízes resultantes sejam registrados em *blockchain*. O uso de Árvores de Merkle possibilita a geração de um único valor que referencia a integridade de um grande número de objetos e o registro em *blockchain* permite garantir um valor confiável de referência ao longo do tempo para prova de carimbo de tempo, prova de existência e verificação e auditoria da integridade dos acervos em RDC, possibilitando garantir a fixidez neles. Este trabalho também documenta uma proposta de arquitetura para uma plataforma de registro e auditoria da fixidez de objetos digitais, em que prevê o registro de informações relacionadas a fixidez de objetos digitais com o uso de algoritmos de *hash* e redes *blockchain* redundantes e que ainda permita o registro de alterações na fixidez desses objetos, oriundas de mudanças neles consideradas legítimas e previstas pelo Open Archival Information System (OAIS), referência para o tratamento de objetos digitais em RDC. A partir dessa arquitetura, foi desenvolvida uma versão dessa plataforma como prova de conceito com os principais recursos necessários à demonstração da aplicação da proposta, possibilitando a realização de experimentos em um ambiente simulado de RDC, os quais também são documentados aqui e que traduzem em termos práticos algumas das preocupações documentadas neste trabalho e a aplicação da plataforma.

**Palavras-chave:** preservação digital; fixidez; Árvore de Merkle; DLT; RDC.

# ABSTRACT

The increased ease provides by records in digital format also brings drawbacks, translated on the challenges of ensure the integrity and authenticity of these records continuously, knowing this format is very susceptible to flaws in any layers of one computational system. International organizations and standards draws recomendations about fixity to guide the Trustworthy Digital Repositories (TDRs) in yours missions of ensure the integrity and authenticity of the records kept there for the time it was defined (or indefinitely). The use of cryptography hash functions enables verify the data integrity and yours authenticity then. However, to trust in the integrity verified by a hash function also is needed trust in the hash value of reference, that eventually are susceptible to the same threats of whose data refers. Starting from this worry, this work proposes an approach in which must be made a chaining in Merkle Trees of the hashes from TDR's inumerous digital objects and the resulting root values may be registered in blockchain. The use of Merkle Trees enables the generation of a unique value that links the integrity of a large number of records and the register in blockchain ensure a reliable witness value over time for use in proof-of-timestamp, proof-of-existence and integrity check and audit of TDRs' digital heritage, enabling to ensure their fixity. This work also documents an architecture proposal to a platform for recording and auditing the fixity of digital objects, in which it provides the registration of information related to the digital objects fixity using redundant hash algorithms and blockchain networks and which still allows the record of changes in the fixity of these objects, arising from changes considered legitim on them and foreseen by the Open Archival Information System (OAIS), a reference to the treatment of digital objects in TDRs. Based on this architecture, a version of this platform was developed as a proof of concept with the main resources necessary to demonstrate the application of the proposal, enabling the performance of experiments in a simulated TDR environment, that are also documented here and which translate into practical terms some of the concerns documented in this work and the application of the platform.

**Keywords:** digital preservation; fixity; Merkle Tree; DLT; TDR.

# LISTA DE FIGURAS

Figura 2.1 — Uma taxonomia dos principais conceitos arquivísticos.....22
Figura 2.2 — Ilustração do modelo OAIS.....27
Figura 2.3 — Diagrama de Árvore de Merkle.....35
Figura 2.4 — Esquema de encadeamento de blocos na Bitcoin.....37
Figura 4.1 — Arquitetura da solução proposta.....51
Figura 4.2 — Diagrama sequencial de funcionamento dos registros na proposta.....52
Figura 4.3 — Estrutura das páginas do Livro de Registros AFA.....54
Figura 4.4 — Fluxograma do processo de auditoria na plataforma.....60
Figura 5.1 — Interface inicial da plataforma Archive Fixity Anchor.....67
Figura 5.2 — Interface para a criação de novos registros.....70
Figura 5.3 — Interface para editar ou auditar registros em uma página.....72
Figura 6.1 — Interface de ingestão de um SIP no Archivematica.....80
Figura 6.2 — Informações sobre pacotes na interface do Archivematica Storage Service.....82
Figura 6.3 — Informações sobre pacotes do cenário de experimentos 1 e seus respectivos estados iniciais de fixidez no Archivematica Storage Service.....83
Figura 6.4 — Informações sobre pacotes do cenário de experimentos 1 e seus respectivos estados iniciais de fixidez na AFA.....84
Figura 6.5 — Informação do Archivematica sobre falha de fixidez no pacote alvo do experimento do cenário 1.....85
Figura 6.6 — Detalhes do Archivematica sobre a falha de fixidez no pacote alvo do experimento do cenário 1.....85
Figura 6.7 — Informação da AFA sobre falha da fixidez no pacote alvo do experimento do cenário 1.....86
Figura 6.8 — Informação do Archivematica sobre o falso sucesso de fixidez no pacote alvo do experimento do cenário 2.....88
Figura 6.9 — Detalhes do Archivematica sobre o falso sucesso de fixidez no pacote alvo do experimento do cenário 2.....88
Figura 6.10 — Informação da AFA sobre falha da fixidez no pacote alvo do experimento do cenário 2.....88
Figura 6.11 — Informação da AFA sobre falha da fixidez na página que registra a fixidez do pacote alvo do experimento do cenário 2.....90
Figura 6.12 — Informação da AFA sobre falha da fixidez no pacote alvo do experimento do cenário 4.....92
Figura 6.13 — Informação da AFA sobre sucesso da fixidez da versão atualizada do pacote alvo do experimento do cenário 4.....93
Figura 6.14 — Informação da AFA sobre sucesso da fixidez no pacote alvo do experimento do cenário 4 após seu registro ser atualizado.....93
Figura A.1 — Comparativo de possíveis resultados da corrupção de um documento.....103

# LISTA DE TABELAS

Tabela A.1 — Valores de *checksum* para um objeto digital.....105
Tabela A.2 — Valores de *hash* para um objeto digital.....105

# LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| ABNT | Associação Brasileira de Normas Técnicas |
| ACTDR | Audit and Certification of Trustworthy Digital Repositories |
| AIP | Archival Information Package |
| API | Application Programming Interface |
| CCSDS | Consultative Committee for Space Data Systems |
| CONARQ | Conselho Nacional de Arquivos |
| CRC | Cyclic Redundancy Check/Code |
| DCC | Digital Curation Centre |
| DIP | Dissemination Information Package |
| DLT | Distributed Ledger Technology |
| DNN | Deep Neural Network |
| DPE | DigitalPreservationEurope |
| DRAMBORA | Digital Repository Audit Method Based on Risk Assessment |
| IBICT | Instituto Brasileiro de Informação em Ciência e Tecnologia |
| ICA | International Council on Archives |
| ICP | Infraestrutura de Chaves Públicas |
| ICV | Integrity Check Value |
| InterPARES | International Research on Permanent Authentic Records in Electronic Systems |
| IPFS | InterPlanetary File System |
| JSON | JavaScript Object Notation |
| LOCKSS | Lots of Copies Keep Stuff Safe |
| MAC | Message Authentication Codes |
| MDC | Modification Detection Code |
| MIC | Message Integrity Code |
| NARA | National Archives and Records Administration |
| NESTOR | Network of Expertise in long-term STORage |
| OAIS | Open Archival Information System |
| OCLC | Online Computer Library Center |
| P2P | Peer-to-Peer |
| PDI | Preservation Description Information |

| | |
|---|---|
| RAID | Redundant Array of Independent Disks |
| RDC | Repositório Digital Confiável |
| RLG | Research Libraries Group |
| RODA | Repositório de Objectos Digitais Autênticos |
| SAAI | Sistema Aberto de Arquivamento de Informação |
| SHA | Secure Hash Algorithm |
| SIP | Submission Information Package |
| TDR | Trusted Digital Repositories / Trustworthy Digital Repositories |
| TRAC | Trustworthy Repositories Audit & Certification |
| TSA | Time-Stamp Authority |
| UUID | Universally Unique Identifier |
| WVM | Widely Visible Media |

# SUMÁRIO

**1 INTRODUÇÃO.....14**
1.1 Objetivos.....16
1.2 Metodologia.....17
1.3 Organização do documento.....18
1.4 Terminologia.....19
**2 FUNDAMENTAÇÃO TEÓRICA.....20**
2.1 Os documentos digitais e suas fragilidades.....20
2.1.1 A integridade dos objetos digitais.....21
2.2 Preservação digital e repositórios digitais.....23
2.2.1 A norma ISO 14721 – Open Archival Information System (OAIS).....25
2.2.2 Confiabilidade em repositórios digitais.....27
2.2.3 Auditoria e certificação de repositórios digitais.....29
2.3 Verificação de integridade de dados.....31
2.4 Árvores de Merkle.....35
2.5 Distributed Ledger Technologies (DLT) e blockchains.....36
**3 TRABALHOS RELACIONADOS.....40**
3.1 Um survey de plataformas para garantia de propriedades dos documentos digitais.....40
3.2 Framework para garantia da fixidez de mementos.....41
3.3 Âncora em blockchain das informações de fixidez da camada intelectual dos documentos digitais.....42
3.4 Registro em blockchain da relação orgânica de documentos digitais.....43
3.5 Blockchain permissionária para validação de documentos assinados digitalmente.....44
3.6 Serviços intermediadores para blockchains.....45
**4 SOLUÇÃO PROPOSTA.....47**
4.1 Premissas da solução.....48
4.2 Arquitetura da solução.....50
4.2.1 Módulo de registro local e auditoria (AFA-LOG).....51
4.2.2 Página de registros (Livro de Registros AFA).....53
4.2.3 Módulo de registro e consulta em DLT (AFA-DLT).....56
4.3 Aplicação da solução.....57
4.3.1 Auditoria de acervos.....59

4.4 Considerações finais....61
**5 PROVA DE CONCEITO....64**
5.1 Implementação....64
5.2 Configuração e execução....66
5.2.1 Criando novos registros....69
5.2.2 Editando registros....72
5.2.3 Auditando registros....73
5.3 Considerações finais....74
**6 AVALIAÇÃO EXPERIMENTAL....76**
6.1 Preparação do ambiente de experimentação....76
6.2 Execução dos experimentos....79
6.2.1 Cenário 1: Adulteração simples de um pacote de informação....83
6.2.2 Cenário 2: Adulteração de um pacote de informação e de suas respectivas informações de fixidez....86
6.2.3 Cenário 3: Adulteração de um pacote de informação e de suas respectivas informações de fixidez registradas na AFA....89
6.2.4 Cenário 4: Edição legítima de um pacote de informação....91
6.3 Considerações finais....93
**7 CONSIDERAÇÕES FINAIS....95**
**REFERÊNCIAS....97**
**APÊNDICE A — EXECUÇÃO, DETECÇÃO E COMPARATIVO DE CORRUPÇÕES EM UM DOCUMENTO DIGITAL....103**
Metodologia de corrupção....104
Detecção de corrupção....104
**APÊNDICE B — ITENS NORMATIVOS CANDIDATOS A SEREM ATENDIDOS PELA IMPLEMENTAÇÃO DA PROPOSTA....107**

# 1 INTRODUÇÃO

Uma missão dos Repositórios Digitais Confiáveis (RDC) é tentar preservar a integridade e autenticidade dos documentos neles armazenados pelo tempo que for definido (ou indefinidamente) e fornecer aos usuários um documento íntegro, autêntico e compreensível, independente do momento de acesso (BARROS; FERRER; MAIA, 2018; DE GIUSTI; LUJÁN VILLARREAL, 2018).

O Repositório Digital Confiável é um componente determinante nas ações de preservação digital, que entre outros requisitos, deve se comprometer em fornecer mecanismos de auditoria e controle de integridade dos documentos custodiados (CONSELHO NACIONAL DE ARQUIVOS, 2015).

No âmbito da preservação digital, processos e informações relacionadas à integridade dos objetos digitais preservados são comumente relacionados como fixidez (*fixity*). Informações de fixidez (*fixity information*) são quaisquer informações necessárias ao processo de verificação de fixidez (*fixity check*), que por sua vez deve demonstrar a garantia da integridade de um dado objeto, que esse permanece inalterado, ou seja, sem modificações ou corrupções.

As recomendações de referência para auditoria e certificação de repositórios digitais tem o controle de fixidez como requisito fundamental no conjunto de ações para a preservação digital e construção da confiabilidade dos repositórios, pois se espera que esse controle possa fornecer evidências de que os materiais custodiados estejam íntegros (CONSULTATIVE COMITTEE FOR SPACE DATA SYSTEMS, 2011; CORETRUSTSEAL, 2020; NESTOR WORKING GROUP ON TRUSTED REPOSITORIES CERTIFICATION, 2006; PHILLIPS et al., 2013).

Sistemas de *software* tidos como referência para construção de repositórios digitais, como o Archivematica[1] e o RODA[2] (RODRIGUES, 2015), executam procedimentos padrão de verificação da fixidez dos objetos sob seus domínios: comparando *hashes* gerados instantaneamente a partir dos objetos com *hashes* de referência previamente registrados e armazenados referentes a esses objetos.

Esses processos de verificação normalmente utilizados podem ser insuficientes para garantir a fixidez no repositório frente as ameaças que as informações digitais estão expostas,

1 https://www.archivematica.org/pt-br/
2 https://www.keep.pt/produtos/roda-repositorio-para-preservacao-de-informacao-digital/

principalmente no longo prazo, devido a degradação ocasional[3] (WRIGHT; MILLER; ADDIS, 2009) ou adulteração intencional dos dados (NATIONAL RESEARCH COUNCIL, 2005).

Portanto, a vulnerabilidade dos valores de referência (informações de fixidez) para o controle de fixidez põe em risco a confiabilidade do acervo preservado, pois a desconfiança dessas informações pode fazer com que um acervo adulterado seja considerado legítimo, assim como a confiança em um acervo legítimo pode ser perdida.

Considerando isso, a norma ISO 16363:2012 exige, por exemplo, que se demonstre quais mecanismos são utilizados para a verificação de fixidez e como as informações de fixidez estão separadas dos objetos digitais a serem preservados, a fim de tentar evitar que essas informações sejam facilmente afetadas no caso de corrupções ou adulterações dos objetos digitais. (CONSULTATIVE COMITTEE FOR SPACE DATA SYSTEMS, 2011)

No entanto, ainda que as informações de fixidez estejam separadas dos objetos digitais, essas podem continuar a correr o mesmo risco de sofrerem corrupções ou adulterações pelos mesmos problemas inerentes ao meio digital e essas desconfianças podem provocar suspeitas sobre a autenticidade do acervo preservado, pois, **como pode ser garantida a fixidez das informações de fixidez?**

É fato que quanto maiores e mais robustos forem os mecanismos de redundância implementados, maior pode ser o nível de segurança e credibilidade da fixidez do repositório. Mas, além do armazenamento com essas tecnologias poderem estar igualmente vulneráveis, também se faz necessário acreditar na ação das técnicas de tratamento para possíveis conflitos de cópias relacionadas aos sistemas de redundância. Wright, Miller e Addis (2009, p. 106, tradução nossa) já advertem sobre tais cenários de vulnerabilidade:

> A implicação é clara. Qualquer organização que use sistemas de armazenamento em massa com um requisito de integridade de dados a longo prazo deve empregar um programa contínuo e proativo de verificação e reparo da integridade de dados em um nível de sistema ponto-a-ponto. Não deve ser assumido que qualquer componente do sistema (redes, armazenamento, memória, processamento) seja, de alguma forma, “seguro”, isto é, imune a falhas e problemas de corrupção de dados.

Então, mesmo considerando o uso de sistemas de redundância, ressalta-se a importância da confiabilidade dos valores de referência para o controle de fixidez, uma vez

3 Eventuais deteriorações que os dados podem sofrer devido comportamentos não previstos dos fenômenos físicos relacionados ao *hardware* envolvido no armazenamento desses dados.

que esses valores também permitem obter o consenso sobre o uso de cópias na reposição de objetos danificados.

É para esse contexto que este trabalho vem propor uma abordagem de uso combinado do conceito de Árvores de Merkle e da tecnologia de livro-razão distribuído (Distributed Ledger Technology – DLT) para que sejam gerados e armazenados imutavelmente valores de referencia para o controle de fixidez dos acervos em RDC, além dos recursos de redundância utilizados pelos repositórios.

A proposta também apresenta um esquema que possibilita o registro das mudanças legítimas que esses objetos possam sofrer ao longo de suas vidas devido as ações de preservação, ainda que os registros originais permaneçam imutáveis.

DLT, como as *blockchains*, são sistemas de cadeia de blocos que registram e validam transações. À medida que as transações são realizadas, elas são difundidas e registradas em blocos de transações por nós pertencentes a uma rede específica. Para que esses blocos de transações sejam considerados válidos, eles devem ser fechados e registrados em blocos subsequentes, que são submetidos ao mesmo processo. Essa rede de nós que registra transações e forma uma cadeia de blocos é denominada de *blockchain* e esse esquema de encadeamento torna imutáveis os registros das transações realizadas.

O uso do conceito de Árvores de Merkle e de funções criptográficas unidirecionais de *hash* permite construir uma única assinatura para um conjunto de objetos digitais e possibilita a verificação da integridade de todo o conjunto a partir dessa assinatura.

O registro dessa assinatura em DLT permite que seja garantida sua imutabilidade e a torne um valor de referência confiável para o processo de verificação e auditoria da fixidez dos acervos em repositórios digitais, fornecendo também uma prova de existência e de carimbo de tempo aos objetos vinculados a essa assinatura.

## 1.1 Objetivos

Este trabalho objetiva propor uma abordagem de uso combinado do conceito de Árvores de Merkle e DLTs para criar valores âncoras imutáveis referentes a objetos digitais preserváveis e suas mudanças e fornecer aos Repositórios Digitais Confiáveis valores confiáveis de referência para realização de auditoria da integridade de seus acervos digitais, visando garantir sua fixidez.

Para alcançar o objetivo final do trabalho, foram traçados os seguintes objetivos específicos a serem satisfeitos:

1. Identificar normativas disponíveis para preservação digital, auditoria e certificação de repositórios digitais e suas recomendações sobre a propriedade de fixidez.
2. Levantar a literatura sobre propostas e técnicas relacionadas a garantia da fixidez de acervos digitais.
3. Elaborar uma abordagem combinada de Árvores de Merkle e *blockchain* que gere e armazene imutavelmente assinaturas referentes a conjuntos de registros de objetos digitais que sirvam de referência para auditoria da fixidez desses objetos.
4. Modelar uma arquitetura que permita o registro e auditoria da fixidez de objetos digitais e suas mudanças vinculados a um RDC.
5. Implementar um protótipo da arquitetura como prova de conceito da abordagem proposta.
6. Validar a aplicabilidade da proposta por meio de experimentos com a implementação da arquitetura elaborada para a solução.

## 1.2 Metodologia

Esta pesquisa foi realizada a partir de uma revisão da literatura, baseada em artigos científicos, livros, normas, legislações, teses, dissertações e outras publicações, quase que em sua totalidade de forma virtual e a busca dos trabalhos se deu primariamente através do portal de buscas Google Acadêmico[4].

As buscas e escolhas de trabalhos para o desenvolvimento da pesquisa tiveram como critério a existência de alguma relação do título ou resumo desses trabalhos com a maioria dos seguintes assuntos (considerando também os ternos relacionados em língua inglesa): preservação digital; integridade de dados; *hash* e árvores de Merkle; dlt e *blockchain*.

Para a solução proposta neste trabalho foi elaborada uma arquitetura para uma plataforma denominada Archive Fixity Anchor, que foi implementada como prova de conceito e que possibilitou pôr a concretização da ideia desenvolvida neste trabalho sob experimentação com o objetivo de validar a aplicabilidade da proposta.

A plataforma implementada foi posta em parceria com um *software* para preservação digital em um ambiente que visou simular um repositório digital e viabilizou a realização de

4 https://scholar.google.com/

experimentos com o comportamento do *software* para preservação e da plataforma proposta em situações de registro de objeto, registro de mudanças em objeto e auditoria dos registros de objetos em cenários com funcionamento regular e de adulterações simuladas.

## 1.3 Organização do documento

Este trabalho está dividido em sete capítulos:

- **Capítulo 1:** neste primeiro capítulo é apresentado uma contextualização acerca dos problemas identificados, da motivação, das tecnologias envolvidas e da solução proposta neste trabalho, além dos objetivos, da metodologia e do esclarecimento sobre a terminologia utilizada no trabalho.
- **Capítulo 2:** no segundo capítulo é apresentada a fundamentação teórica, contendo descrições e discussões sobre os conceitos pertinentes ao embasamento da proposta do trabalho.
- **Capítulo 3:** no Capítulo 3 são apresentados alguns trabalhos relacionados com a temática discutida.
- **Capítulo 4:** no Capítulo 4 é descrito a abordagem geral da solução proposta e algumas premissas necessárias a compreensão de uma plataforma de *software* sugerida, a qual é apresentada na sequencia para apoiar a abordagem proposta.
- **Capítulo 5:** no Capítulo 5 é documentada uma prova de conceito criada a partir da implementação em *software* da plataforma documentada no Capítulo 4.
- **Capítulo 6:** no Capítulo 6 são documentados experimentos que foram realizadas a fim de avaliar o comportamento da proposta apresentada partindo da prova de conceito criada no Capítulo 5.
- **Capítulo 7:** no Capítulo 7 são apresentadas as considerações finais sobre o desenvolvimento deste trabalho e sua potencial demanda para continuação em trabalhos futuros.

## 1.4 Terminologia

Devido a interseção de áreas em que este trabalho se encontra, considerando a adaptação de termos entre diferentes idiomas e a pluralidade de termos algumas vezes encontrada, se faz desejável um esclarecimento prévio sobre a terminologia utilizada aqui para o tratamento de assuntos da pesquisa.

No contexto computacional, o termo “arquivo” já identifica um conjunto de bits que forma uma unidade abstrata de armazenamento de informações de forma persistente (TANENBAUM; BOS, 2009), mas na perspectiva arquivística, ele pode ser referenciado como “arquivo digital”, para diferenciá-lo de outros significados de arquivo (que pode se referir a um conjunto de documentos ou uma instituição ou serviço de custódia de documentos) (CONSELHO NACIONAL DE ARQUIVOS, 2016). Ainda assim, dependendo da abordagem, o “arquivo digital” pode acabar por se referir a uma versão digital de outros significados de “arquivo” (BARBEDO et al., 2007).

Também encontra-se nos referenciais o uso dos termos “objetos digitais” e “documentos digitais” para se referir ao mesmo objeto de informação. No entanto, percebe-se que o uso de “documentos digitais” está comumente relacionado ao documento arquivístico[5] digital e se refere aos objetos a serem tratados na perspectiva do uso (como usuário). Por outro lado, o termo “objeto digital” é utilizado para referência aos objetos na perspectiva do tratamento, ainda que esse não seja um documento arquivístico digital.

“Objeto digital” ou “*digital object*”[6] é o termo fortemente utilizado nos referenciais para tratar de preservação digital e se referir aos objetos a serem tratados, inclusive nas normas internacionais de referência e por isso, esse será o principal termo adotado neste trabalho para se referir aos objetos em tratamento, inclusive quando referente a arquivo no sentido computacional.

O termo “arquivo”, quando não ressalvado ou acompanhado de contexto computacional, se referirá a “instituição ou serviço que tem por finalidade a custódia, o processamento técnico, a conservação e o acesso a documento arquivístico” (CONSELHO NACIONAL DE ARQUIVOS, 2016, p. 9).

---

5 “Documento produzido (elaborado ou recebido), no curso de uma atividade prática, como instrumento ou resultado de tal atividade, e retido para ação ou referência.” (CONSELHO NACIONAL DE ARQUIVOS, 2016, p. 20)

6 “Discrete unit of information in digital form. A Digital Object can be an Intellectual Entity, Representation, File, Bitstream, or Filestream.” (PREMIS WORKING GROUP, 2015, p. 269)

# 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os principais conceitos que contextualizam e fundamentam a problemática e a solução proposta. São inicialmente apresentados os conceitos e preocupações em torno dos documentos digitais e suas propriedades, as ações de preservação digital, os repositórios digitais e recomendações e normatizações relacionadas. Na sequência são apresentadas técnicas de referência para verificação de integridade de dados e a tecnologia de *blockchain*.

## 2.1 Os documentos digitais e suas fragilidades

Os documentos digitais proporcionam benefícios já conhecidos, como no armazenamento e transporte, flexibilizam a produção e manipulação das informações, reduzem custos e elevam a eficiência em processos que dependem da utilização de documentos. A versatilidade dos documentos em formato digital atrai o interesse de pessoas e organizações para o uso exclusivo desse formato, ao tempo que os tornam fortemente dependentes desse formato de documentos e do aparato que possibilita seu uso. (CONSELHO NACIONAL DE ARQUIVOS, 2005; MONTEIRO, 2001; SAYÃO, 2010)

Para o Conselho Nacional de Arquivos (2005), as novas facilidades proporcionadas pelos documentos em formato digital também trazem dificuldades, traduzidas nos desafios de garantir a integridade e acessibilidade desses documentos continuamente, sabendo que esse formato é bastante sensível à falhas de *hardware* e *software* e à obsolescência tecnológica.

Com base nas reflexões de Duranti e Macneil (1996) e Ferreira (2006) e em consonância com as definições do Conselho Nacional de Arquivos (2016), pode-se inferir uma divisão de aspecto técnico que, diferente de documentos físicos tradicionais, no meio digital, os documentos são resultados de uma combinação de três diferentes camadas divisíveis, estando os documentos digitais passíveis de problemas em qualquer uma dessas camadas (NATIONAL RESEARCH COUNCIL, 2005; ROSENTHAL, 2010; ROSENTHAL et al., 2005):

- **Camada intelectual** – Onde está contida e organizada a informação em si.
- **Camada lógica** – Onde a camada intelectual está traduzida em estruturas formatadas e padronizadas para processamento computacional.

- **Camada física** – Onde estão codificadas em fenômenos físicos as unidades de informação referentes as estruturas da camada lógica.

A camada intelectual pode sofrer com erros operacionais ou adulterações indevidas, intencionais ou não, de *softwares* ou ações humanas. A alteração do conteúdo de um objeto por um usuário ou pela intervenção automatizada de um *software* mal configurado podem ser exemplos de possíveis ameaças a essa camada do documento.

Além dos problemas aos quais a camada intelectual está suscetível, a camada lógica pode também sofrer com a obsolescência tecnológica e funcionamento inadequado de *hardwares* e *softwares*. Um *bug* em determinada versão de um *software* (aplicação ou *driver*) pode causar uma anomalia no armazenamento ou tratamento do objeto, assim como essa anomalia pode ser oriunda de um problema de *hardware*.

A camada física também é vulnerável a erros operacionais, assim como intervenções maliciosas e problemas físicos nos suportes ou sistemas, por problemas de fabricação, instalação, configuração, degradação, erros ocasionais, destruição ou vulnerabilidades inerentes dos componentes. Como exemplo, se pode citar os desastres naturais ou a falha de um disco de armazenamento devido seu desgaste do tempo de vida.

Baumann (2005), Slayman (2011) e Li et al. (2019) debatem sobre a vulnerabilidade de sistemas computacionais a erros transientes ou não e destaca-se, até mesmo, a interferência de raios cósmicos no funcionamento dos sistemas (O'GORMAN et al., 1996; ZIEGLER et al., 1998).

Tais vulnerabilidades são amostras das preocupações que alimentam o debate em torno das ações de preservação da informação em um meio de volatilidade que, como pondera Skinner e Schultz (2010), paradoxalmente pode proporcionar grande risco e grande segurança aos acervos digitais.

### 2.1.1 A integridade dos objetos digitais

Quando relacionada a informação, a integridade é o atributo de que essa não sofreu modificações indesejáveis ou não autorizadas. É um atributo indispensável no tratamento de dados e desejável a ambas as ciências da computação e da informação.

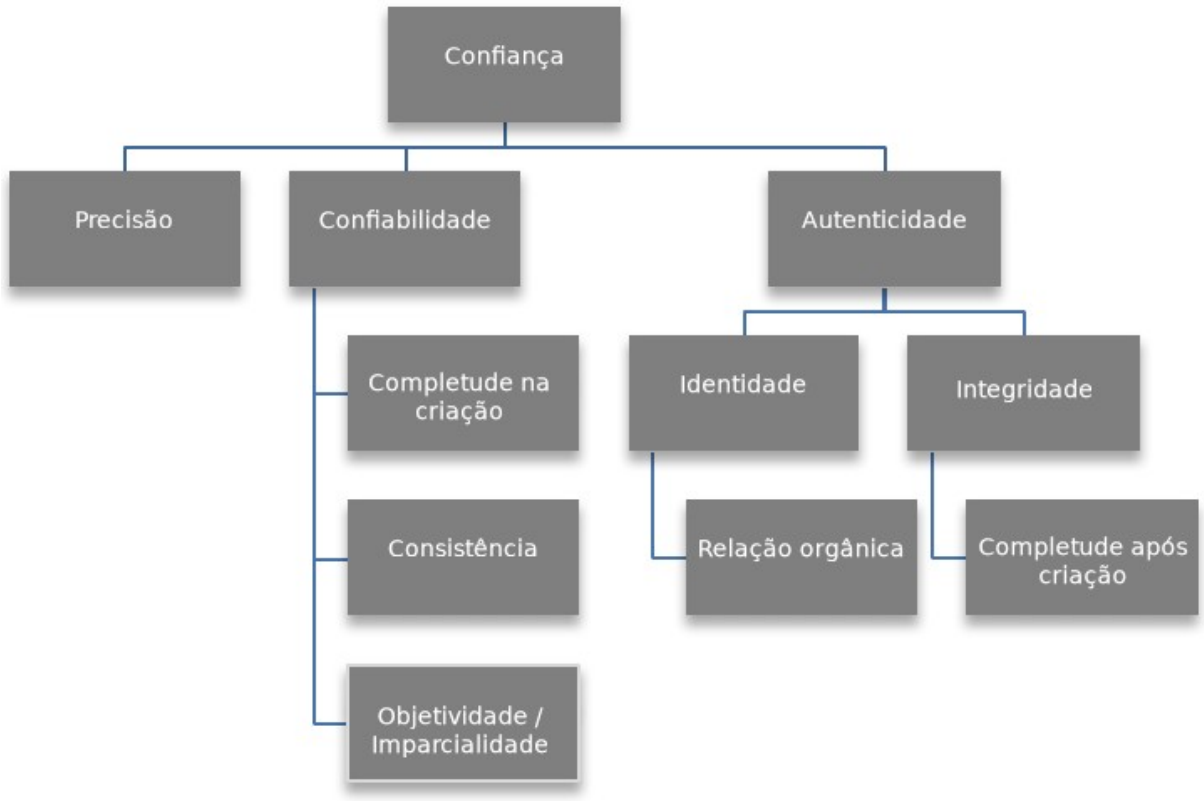
No contexto computacional, a integridade é uma propriedade veterana nas discussões sobre segurança, sendo, segundo Stallings (2015), parte de uma tríade de conceitos

fundamentais a segurança de dados, serviços e sistemas de informação e computação (em companhia da confiabilidade e disponibilidade).

Para a ciência arquivística, a integridade é um dos componentes de uma cadeia essencial para a construção da confiança de um documento, ao tempo em que a confiança é um conceito que acompanha continuamente as discussões nessa ciência e é um elemento indispensável para que esses documentos sejam considerados válidos para fins de evidência.

A Figura 2.1 ilustra uma taxonomia dos principais conceitos arquivísticos, onde se demonstra a relevância da integridade na cadeia de confiança de um documento arquivístico e sua direta relação com as propriedades de completude[7] e autenticidade[8] dos registros.

Figura 2.1 — Uma taxonomia dos principais conceitos arquivísticos




Fonte: (LEMIEUX, 2017, p. 3, tradução nossa)

7 "Atributo de um documento arquivístico que se refere à presença de todos os elementos intrínsecos e extrínsecos exigidos pela organização produtora e pelo sistema jurídico-administrativo a que pertence, de maneira a ser capaz de gerar consequências." (CONSELHO NACIONAL DE ARQUIVOS, 2016, p. 15)

8 "Credibilidade de um documento enquanto documento, isto é, a qualidade de um documento ser o que diz ser e que está livre de adulteração ou qualquer outro tipo de corrupção." (CONSELHO NACIONAL DE ARQUIVOS, 2016, p. 10)

Não obstante a importância da propriedade de integridade em documentos físicos, no meio digital, essa se torna ainda mais crucial, pois o dano de um mínimo bit pode levar ao comprometimento definitivo da informação, conforme corrobora Juels e Kaliski (2007, p. 7, tradução nossa): “Mesmo um único bit ausente ou invertido pode representar uma corrupção semanticamente significativa.” Tal proposição pode ser verificada por um experimento de corrupção de um único bit em um documento, onde sua execução e seus resultados estão documentados no Apêndice A.

De acordo com Lemieux (2017, p. 4, tradução nossa), “se a integridade de um registro for comprometida, é impossível estabelecer a autenticidade dele com qualquer grau de certeza” e, portanto, “para permanecerem autênticos, os registros devem permanecer livres de adulteração, corrupção ou alteração ao longo do tempo”.

Essas preocupações com a integridade dos documentos digitais são traduzidas nas recomendações estabelecidas por normas nacionais e internacionais que orientam o gerenciamento de objetos e repositórios digitais, a exemplo das normas internacionais 14721[9], 15489[10], 16363[11] e 23081[12] da ISO (referências para normas nacionais).

## 2.2 Preservação digital e repositórios digitais

Cópia de segurança (ou *backup*) é uma cópia simples, feita em um ou mais dispositivos, que visa proporcionar redundância e disponibilidade para restauração da informação quando essa sofrer danos parciais ou totais, podendo ser resumida como uma cópia simples de dados selecionados para posterior recuperação (CARVALHO, 2009; CONSELHO NACIONAL DE ARQUIVOS, 2016).

Conforme reforça Vignati (2009) e Skinner e Schultz (2010), a preservação digital não deve ser meramente confundida com *backup*, pois essa última possibilita a recuperação de dados a curto prazo com baixo investimento; enquanto a preservação digital é mais onerosa, pois se trata de um “conjunto de ações gerenciais e técnicas exigidas para superar as mudanças tecnológicas e a fragilidade dos suportes, garantindo o acesso e a interpretação de

9 Space data and information transfer systems — Open archival information system (OAIS) — Reference model
10 Information and documentation — Records management
11 Space data and information transfer systems — Audit and certification of trustworthy digital repositories
12 Information and documentation — Records management processes — Metadata for records

documentos digitais pelo tempo que for necessário" (CONSELHO NACIONAL DE ARQUIVOS, 2016, p. 34).

Em todo caso, as cópias de segurança são imprescindíveis à preservação digital, sendo a realização de *backups* uma das ações de preservação previstas nas recomendações para o gerenciamento dos repositórios digitais, em especial, os *backups off-site*[13] *(CONSULTATIVE COMITTEE FOR SPACE DATA SYSTEMS, 2011; NESTOR WORKING GROUP ON TRUSTED REPOSITORIES CERTIFICATION, 2006; PHILLIPS et al., 2013)*.

Preservação digital também é, mas não é apenas a preservação intacta dos objetos no âmbito de bits. Estão documentadas diversas ações técnicas que auxiliam a preservação digital (FERREIRA, 2006), algumas visam preservar todas as camadas dos objetos, enquanto outras preveem a mudança de estrutura das camadas físicas e lógicas, a fim de tentar superar as mudanças tecnológicas que a preservação da camada intelectual de um objeto possa exigir ao longo do tempo (CONSULTATIVE COMITTEE FOR SPACE DATA SYSTEMS, 2012). Essas ações são apoiadas por plataformas denominadas Repositórios Digitais.

Os Repositórios Digitais são apontados como soluções informatizadas que recebem, preservam e provêm acesso aos documentos digitais, proporcionando um ambiente para armazenamento e gerenciamento desses documentos, ainda conceituado pelo Conselho Nacional de Arquivos (2015, p. 9) como "um complexo que apoia o gerenciamento dos materiais digitais, pelo tempo que for necessário, e é formado por elementos de *hardware*, *software* e metadados, bem como por uma infraestrutura organizacional e procedimentos normativos e técnicos".

O termo "repositório digital" pode ser encontrado nas diversas áreas da ciência da informação, pois esse instrumento é o alicerce para a sustentação digital de repositórios arquivísticos, temáticos e institucionais, além de outros estilos e terminologias que são debatidas no trabalho de Corujo (2014), mas são dispensáveis no contexto deste trabalho.

Os repositórios arquivísticos lidam com documentos arquivísticos em qualquer uma de suas fases (corrente, intermediária e permanente[14]) (CONSELHO NACIONAL DE ARQUIVOS, 2015). Já os repositórios temáticos, são entendidos como aqueles destinados a preservar a produção intelectual de determinada área do conhecimento em particular, ocasionalmente denominados repositórios disciplinares. Nos repositórios institucionais estão todas as produções científicas de uma determinada instituição, também podendo ser

13 "Replicação de cópias de segurança para um espaço geograficamente separado do espaço dos sistemas em produção." (DELL TECHNOLOGIES BRAZIL, 2020, tradução nossa)

14 "Sucessivas fases por que passam os documentos […], da sua produção à guarda permanente ou eliminação." (ARQUIVO NACIONAL (BRASIL), 2005, p. 47)

considerada como a reunião de todos os repositórios temáticos hospedados em uma instituição. (INSTITUTO BRASILEIRO DE INFORMAÇÃO EM CIÊNCIA E TECNOLOGIA, 2018; SAYÃO et al., 2009)

### 2.2.1 A norma ISO 14721 – Open Archival Information System (OAIS)

Em 2002, o Consultative Committee for Space Data Systems (CCSDS) publicou o modelo de referência Open Archival Information System (OAIS), que recomenda práticas para o tratamento dos objetos digitais nos repositórios digitais. Esse documento foi formalizado como a norma ISO 14721 em 2003 e revisado em 2012, a partir de um novo documento publicado pelo CCSDS, também em 2012 (CONSULTATIVE COMITTEE FOR SPACE DATA SYSTEMS, 2012).

A norma internacional de 2003 foi traduzida e adaptada para o Brasil em 2007 pela Associação Brasileira de Normas Técnicas (ABNT) sob a norma NBR 15472, intitulada Sistema Aberto de Arquivamento de Informação (SAAI) (CONSELHO NACIONAL DE ARQUIVOS, 2015).

Para a ABNT (2007, p. vi) um SAAI é um arquivo, compreendido como “uma organização de pessoas e sistemas, que aceitou a responsabilidade de preservar informação e torná-la disponível a uma comunidade-alvo”.

O modelo OAIS (e analogamente o SAAI) é puramente conceitual e não especifica tecnologias que devem ser utilizadas para sua implementação. Sua publicação e adoção como norma é considerada como um marco para a história da preservação digital (OWENS, 2007), visto que a conformidade com esse modelo é tido como requisito fundamental para a criação de Repositórios Digitais Confiáveis pelos autores e normas que abordam essa temática, assim como para o Conselho Nacional de Arquivos (2015).

No modelo OAIS a informação a ser interpretada pelos usuários é denominada de Information Object e conceitualmente é composta por duas partes distintas: o Data Object, parte dos dados a serem preservados, e o Representation Information, parte responsável pela interpretação do Data Object.

Em uma abordagem prática, um Data Object pode ser um documento digital, como uma imagem, que apenas se torna compreensível pelos usuários a partir de uma interpretação computacional, realizada por um processamento sistemático de acordo com especificações para esse fim. As informações necessárias ao correto processamento do Data Object estão

contidas na Representation Information e o resultado dessa interpretação (a informação desejável pelos usuários) é denominada Information Object.

Em um outro nível de abstração, o Information Object passa a ser considerado Content Information e será acompanhado pelo Preservation Description Information (PDI) na geração de um recipiente conceitual intitulado Information Package. Na PDI estão contidas informações auxiliares para o processo de preservação da Content Information, tais como: proveniência, contexto, referência, **fixidez** e direitos de acesso. O Information Package ainda ganha uma informação descritiva (Descriptive Information) para que o pacote seja indexado e possa ser encontrado no sistema de preservação.

Essa estrutura conceitual de pacote é a forma como o modelo prevê o gerenciamento dos objetos digitais para tratamento e intercomunicação da informação entre as entidades do sistema e entre as diferentes fases de fluxo dos objetos nele.

O modelo estabelece três tipos de pacotes de informação (Information Packages) nos quais o objeto digital preservável será incorporado para atravessar as três principais fases do fluxo de informação no sistema: admissão, preservação e acesso.

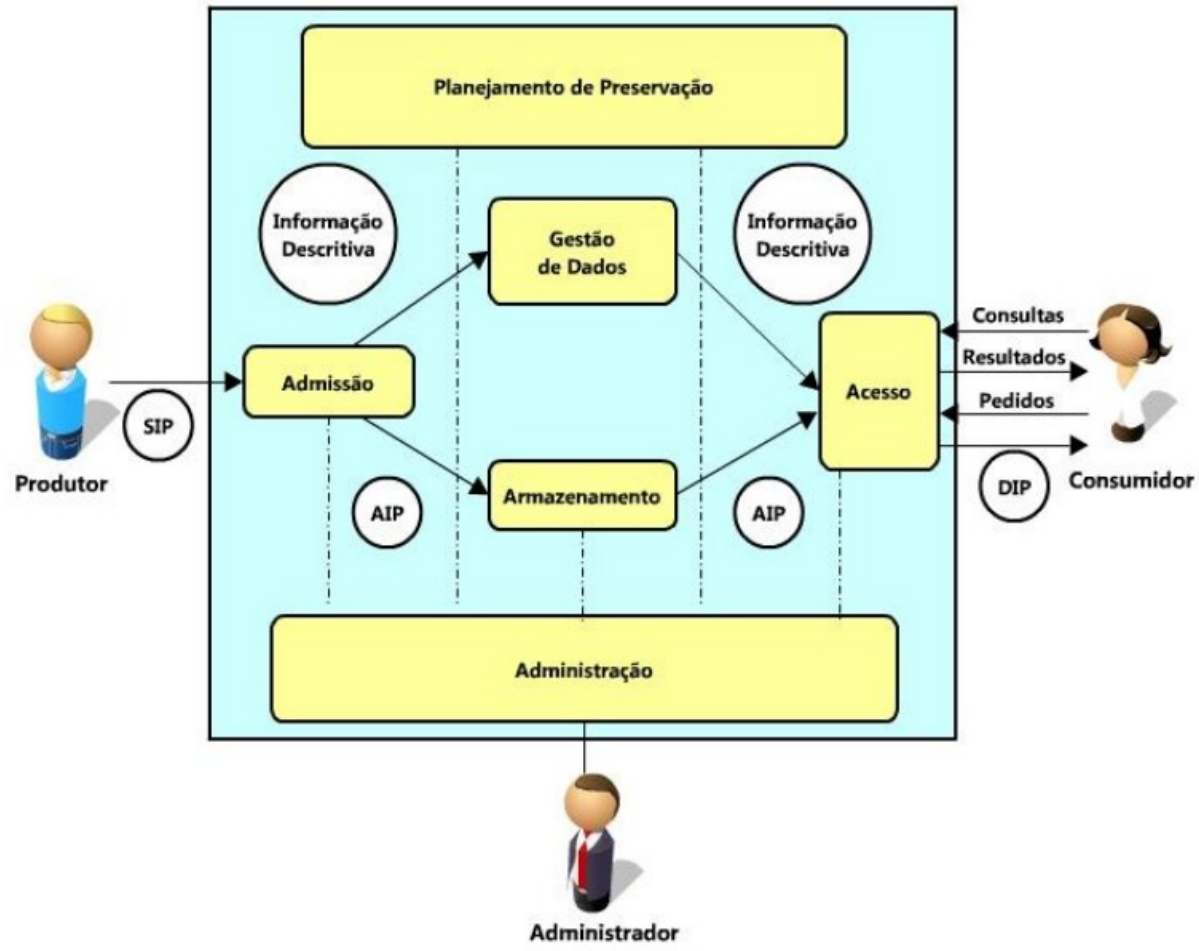
A Figura 2.2 ilustra as entidades envolvidas no sistema, o fluxo da informação e os tipos de pacotes em que a informação está encapsulada em cada fase do processo:

- **Admissão –** Na fase de admissão (frequentemente chamada de "ingestão"), o objeto deve estar encapsulado em um Submission Information Package (SIP) e é inserido no sistema pelo produtor[15]. Nessa etapa o objeto recebe seu tratamento inicial para se tornar apto a preservação.
- **Preservação –** Na fase de preservação, o objeto está encapsulado em um Archival Information Package (AIP) e aqui recebe tratamentos planejados para que a informação se mantenha preservada e recuperável ao longo do tempo.
- **Acesso –** Na fase de acesso, o AIP é transformado em um Dissemination Information Package (DIP) e recebe o tratamento necessário para que a informação desejada seja entregue ao seu consumidor[16].

15 "[,…] papel desempenhado por aquelas pessoas ou sistemas-cliente que fornecem a informação a ser preservada" (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2007, p. 9).

16 "[…] papel desempenhado por aquelas pessoas ou sistemas-cliente que interagem [...] para encontrar e adquirir informação preservada de interesse" (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2007, p. 9).

Figura 2.2 — Ilustração do modelo OAIS




Fonte: (CONSELHO NACIONAL DE ARQUIVOS, 2015).

## 2.2.2 Confiabilidade em repositórios digitais

Desde a década de 1990 emergem debates acerca da manutenção a longo prazo dos documentos digitais, destacando-se, dentre esses estudos, o relatório da “Task Force on Archiving of Digital Information”, publicado em 1996, que de acordo com o Conselho Nacional de Arquivos (2015), foi um precursor de uma série de importantes documentos que trazem conceitos e ideias da implementação de confiabilidade, preservação, acesso e autenticidade em repositórios digitais.

Além de expressar as preocupações em torno da preservação digital, a Força Tarefa, composta pelo The Research Libraries Group e The Commission on Preservation and Access, aborda a confiança como fator preponderante para a preservação da informação, considerando

que as instituições de preservação precisam demonstrar confiabilidade de seus acervos e de suas ações de preservação para garantir ao usuário que aquela informação consumida é o que ele espera que realmente seja. Para isso, a Força Tarefa alega a necessidade de certificação dos repositórios digitais: “Um processo de certificação para arquivos digitais é necessário para criar um clima geral de confiança sobre as perspectivas de preservação da informação digital” (TASK FORCE ON ARCHIVING OF DIGITAL INFORMATION, 1996, p. 40, tradução nossa).

Com base na publicação da Força Tarefa de 1996, o Research Libraries Group (RLG) e o Online Computer Library Center (OCLC), compondo o RLG-OCLC Working Group on Digital Archive Attributes, publicaram o relatório “Trusted Digital Repositories: Attributes and Responsibilities” (documento comumente referenciado como “TDR”) em 2002, que define o conceito de um repositório digital confiável e estabelece um *framework* de propriedades e exigências que um repositório digital deve atender para ser considerado confiável.

No relatório de 2002, o Grupo de Trabalho definiu um repositório digital confiável como “aquele cuja missão é fornecer acesso confiável e de longo prazo a recursos digitais gerenciados para sua comunidade designada, agora e no futuro” (RLG-OCLC WORKING GROUP ON DIGITAL ARCHIVE ATTRIBUTES, 2002, p. 5, tradução nossa). Esse relatório elenca elementos de gestão organizacional e de tecnologia necessários para agregar confiança a um repositório digital.

Owens (2007, p. 280, tradução nossa) já destaca que preservação digital “é tanto quanto ou mais sobre características organizacionais do que questões tecnológicas”, ao tempo em que reforça que o aspecto “tecnologia” da preservação digital é fortemente pesquisado.

Do trabalho de Owens (2007) e do Relatório do Grupo de Trabalho RLG-OCLC (2002) pode se interpretar uma certa divisão do aspecto tecnologia do repositório digital em forma de três camadas, semelhante a divisão da estrutura de um documento digital apresentada na Seção 2.1: física, lógica e intelectual.

- **Camada física –** Aqui, a camada trata dos bits brutos, da mídia de armazenamento e ações de manutenção desse nível;
- **Camada lógica –** Nessa camada são considerados os formatos de arquivos e estruturas de dados, a organização dos bits;
- **Camada intelectual –** Na camada intelectual estaria a informação de como usar ou acessar os dados, principalmente por meio dos metadados.

De acordo com o TDR (RLG-OCLC WORKING GROUP ON DIGITAL ARCHIVE ATTRIBUTES, 2002), as instituições podem delegar a responsabilidade de gerenciamento da camada física do repositório digital a terceiros. Porém, aqui reside a polêmica de como as instituições confiarão no trabalho de terceiros, uma vez que elas já buscam conquistar a confiabilidade de seus usuários a partir de suas ações.

Para o estabelecimento de um repositório digital confiável ao menos três níveis de confiança são aplicáveis, segundo o Grupo de Trabalho RLG-OCLC (2002):

- Confiança do público-alvo na instituição.
- Confiança da instituição nos prestadores de serviços.
- Confiança dos usuários nos documentos providos pela instituição.

Bibliotecas e outras instituições culturais gozam da confiança que adquiriram de seus públicos para e por armazenar o patrimônio arquivístico (não-digital) ao longo dos tempos. Por isso, essas instituições possuem uma confiança preestabelecida de seus públicos para preservar o patrimônio, agora digital. (RLG-OCLC WORKING GROUP ON DIGITAL ARCHIVE ATTRIBUTES, 2002; THOMAZ, 2007).

### 2.2.3 Auditoria e certificação de repositórios digitais

Considerando os estudos de Santos e Flores (2015), Thomaz (2007) e do Grupo de Trabalho RLG-OCLC (2002), a confiança necessária a um repositório é uma propriedade adquirida ao longo do tempo, mas considerando a imediata necessidade de preservar os acervos digitais atuais, um processo de certificação pode fornecer uma base de confiança para um repositório digital. Por isso, a auditoria e a certificação surgem para auxiliarem os repositórios a comprovarem a confiabilidade que ganharão ao longo do tempo de seus públicos.

Para alcançar uma certificação de confiabilidade, deve-se submeter um repositório digital a um processo de auditoria, pois Barros, Ferrer e Maia (2018) consideram que a existência de estratégias, políticas e normas de preservação não são suficientes para garantir a confiabilidade dos repositórios digitais. Então, “auditar os repositórios digitais significa torná-los confiáveis e, na medida do possível, mais seguros para garantir que as informações ali dispostas estejam preservadas ao longo do tempo” (BARROS; FERRER; MAIA, 2018, p. 301). Essa ideia é corroborada por Térmens e Leija (2017, p. 448, tradução nossa) ao afirmar

que “uma das metodologias mais difundidas para a obtenção de repositórios confiáveis são os sistemas de auditoria, realizados por pessoal especializado, que permitem determinar se um repositório é seguro e, portanto, se podemos confiar nele”.

Após o processo de auditoria, a entidade auditora analisa os dados obtidos para avaliar o nível de conformidade do repositório para, com isso, conceder ou não sua certificação (SANTOS; FLORES, 2015).

Em 2007, o RLG e a National Archives and Records Administration (NARA) compondo a RLG-NARA Task Force on Digital Repository Certification publicaram o “Trustworthy Repositories Audit & Certification: Criteria and Checklist” (TRAC), objetivando guiar instituições e provedores de serviço a auditarem e certificarem seus repositórios digitais a partir de uma lista de critérios. Esses critérios contemplam todos os aspectos esperados de um repositório digital e demandam o envolvimento de diversos setores da instituição. (RLG-NARA TASK FORCE ON DIGITAL REPOSITORY CERTIFICATION, 2007).

Esse documento da Força Tarefa RLG-NARA (2007) incorporou elementos de publicações de outros esforços internacionais para referência de auditoria e certificação de repositórios digitais, a saber, o “Catalogue of Criteria for Trusted Digital Repositories” do Network of Expertise in long-term STORage Working Group on Trusted Repositories Certification (NESTOR, 2006) e o “Digital Repository Audit Method Based on Risk Assessment” (DRAMBORA) do Digital Curation Centre (DCC) e DigitalPreservationEurope (DPE) (2007).

A partir do documento TRAC de 2007, o CCSDS elaborou o “Audit and Certification of Trustworthy Digital Repositories” (ACTDR), publicado em 2011. Essa publicação do CCSDS foi consolidada como a norma ISO 16363 em 2012 (CONSULTATIVE COMITTEE FOR SPACE DATA SYSTEMS, 2011).

A norma internacional ISO 16363:2012 reúne em um documento os resultados de diversos trabalhos, reconhecidos no âmbito de direcionar auditorias para certificação de repositórios digitais. Nesse documento, são elencados critérios considerados necessários que um repositório digital contemple para que possa demonstrar confiabilidade, assim como, de que forma esse repositório pode comprovar sua conformidade com cada um desses critérios, sendo a norma sintetizada nas palavras de Gonçalez (2017, p. 217) como “[...] uma ferramenta que viabiliza a auditoria, a avaliação e a certificação dos repositórios digitais”.

Santos e Flores (2015) consideram que a confiabilidade de um repositório digital pode ser adquirida desde que esse se baseie no modelo OAIS e esteja em conformidade com os

requisitos apresentados por algum dos documentos de referência para auditoria, nomeadamente TRAC, ACTDR, NESTOR e DRAMBORA, reforçando a possibilidade de utilização de ferramentas alternativas à norma internacional.

Ao longo do tempo surgiram iniciativas isoladas que foram ganhando reconhecimento e convergiram com outras interessadas no mesmo propósito: discutir a problemática da confiança dos repositórios e como esses poderiam atestar essa confiabilidade.

Alguns trabalhos derivados dessas iniciativas acabaram por compor a norma internacional ISO 16363, enquanto outros seguiram como formas alternativas (e menos onerosas) para auditar repositórios digitais com relevância reconhecida por estudos do tema, a exemplo do CoreTrustSeal (CORETRUSTSEAL, 2018) e o National Digital Stewardship Alliance Levels of Digital Preservation (NATIONAL DIGITAL STEWARDSHIP ALLIANCE; DIGITAL LIBRARY FEDERATION, 2018).

## 2.3 Verificação de integridade de dados

O uso combinado de funções matemáticas possibilita a elaboração de algoritmos computacionais que permitem a geração de assinaturas com extensão predeterminada, a partir de seguimentos de dados de tamanhos variáveis. Essas assinaturas são frequentemente mencionadas como “resumo de mensagem” (“*message digest”* ou apenas “*digest*”), “soma de verificação” (*checksum*) e *hash*.

Com o uso desses algoritmos, é esperado que sempre seja gerado uma mesma assinatura a partir dos mesmos dados e com isso, possibilite realizar um comparativo entre as assinaturas geradas a partir dos dados — que acredita-se serem os mesmos — em diferentes momentos no tempo, para então, ser possível atestar, com um determinado grau de confiança, que tais dados são realmente os mesmos; anunciando que tais dados estão íntegros.

Esse nível de confiança é determinado pelos critérios e prioridades da aplicação, como a disponibilidade de recursos computacionais, o tempo desejável/disponível para processamento e o montante de dados.

Na literatura, normalmente as corrupções dos dados ocorridas de forma acidental ou não maliciosa durante sua transmissão ou armazenamento são tidas como erros (aleatórios) e em algumas aplicações essas são as únicas corrupções que se deseja detectar.

As técnicas de verificação de integridade dedicadas a esse tipo de verificação costumam ser chamadas de técnicas de detecção de erros e se diferem em complexidade e

custo computacional de técnicas mais avançadas, aptas a detectar corrupções de qualquer natureza, inclusive aquelas oriundas de erros.

A paridade (bits de paridade ou dígitos verificadores) pode ser considerada a técnica mais básica de verificação de dados. As assinaturas e a forma como são geradas nessa técnica são demasiadas simples e inexpressivas, o que conduz a baixa precisão e uso em aplicações mais específicas, como alguns níveis de RAID[17]. (KUROSE; ROSS, 2013; SOMASUNDARAM, G. SHRIVASTAVA, 2011; TANENBAUM; BOS, 2009; TANENBAUM; WETHERALL, 2011)

Um pouco mais elaborada, a soma de verificação (*checksum*) essencialmente se baseia em operações de soma para gerar as assinaturas de verificação, resultando em uma certa simplicidade e mantendo semelhança com a técnica de paridade, mas entregando resultados mais confiáveis, a tornando útil para comunicações, como na Internet (BRADEN; BORMAN; PARTRIDGE, 1988; STONE; PARTRIDGE, 2000).

Os algoritmos de verificação cíclica de redundância ou códigos de verificação cíclica (Cyclic Redundancy Check/Code – CRC) são mais complexos do que as somas de verificação, se baseando na aritmética de polinômios para geração das assinaturas (algumas vezes chamadas de “códigos polinomiais”) e por isso, conseguem entregar resultados ainda mais confiáveis na detecção de erros.

Por manterem uma simplicidade, os CRCs também são adotados como mecanismos de detecção de erros em transmissões de dados, como na camada de enlace das redes de computadores. (KUROSE; ROSS, 2013; TANENBAUM; WETHERALL, 2011)

Todavia, CRCs são eventualmente considerados na literatura como parte da categoria de *checksums*, pois Tanebaum e Wetherall (2011, p. 132) observam que “o termo ‘checksum’ normalmente é usado para indicar um grupo de bits de verificação associados a uma mensagem, independentemente de como são calculados”, mas os autores preferem evidenciar as diferenças em suas explicações. Já para Menezes, Oorschot e Vanstone (2001) e Stallings (2015), a classificação em uma mesma categoria se dá devido as semelhanças na forma como as assinaturas são geradas (usando funções lineares). Apesar das semelhanças, na publicação de Kurose e Ross (2013), os autores preferem distinguir as funções de *checksum* das de CRC.

A verificação da integridade de dados por *checksums* ou CRC acontece de uma forma simples e rápida, demandando pouco tempo e poder computacional, o que as fazem ser desejáveis em aplicações que tenham esses atributos como prioridades.

17 Acrônimo para Arranjo Redundante de Discos Independentes (Redundant Array of Independent Disks), inicialmente chamado de Arranjo Redundante de Discos Baratos (Redundant Array of Inexpensive Disks).

Conforme esclarece Menezes, Oorschot e Vanstone (2001) e Stone e Partidge (2000), *checksums* são funções baratas (computacionalmente) e pertencem a um conjunto de técnicas auxiliares para detecção e correção de erros: “As somas de verificação são verificações de erros mais simples, projetadas para equilibrar o custo de computação (normalmente em *software*) com a chance de detectar um erro com êxito.” (STONE; PARTRIDGE, 2000, p. 309, tradução nossa)

Por outro lado, a simplicidade dessas funções compromete o nível de confiança da integridade dos dados, pois segundo Stallings (2015), a utilização de técnicas de verificação de dados baseadas em funções lineares para a geração dos códigos permitem análises estatísticas, que possibilitam potenciais intervenções maliciosas em sistemas que usem unicamente esse tipo de técnica em seus processos:

> Um ponto fraco em potencial do CRC como uma verificação de integridade é que essa é uma função linear. Isso significa que você pode prever quais bits do CRC são mudados se um único bit da mensagem for alterado. Além do mais, é possível determinar qual combinação de bits poderia ser invertida na mensagem de modo que o resultado final seja nenhuma mudança no CRC. Assim, existem diversas combinações de inversões de bit da mensagem de texto claro que deixam o CRC inalterado, de modo que a integridade da mensagem é impedida. (STALLINGS, 2015, p. 464)

Aqueles que visam garantir níveis elevados de confiança na garantia da integridade dos dados devem utilizar algoritmos criptográficos, que foram projetados objetivando, antes de tudo, fornecer essa garantia.

As funções criptográficas nomeadas de “*hash*” são funções significativamente mais complexas que as de *checksum* e, como reforça Menezes, Oorschot e Vanstone (2001, p. 363, tradução nossa): “Em contraste com as somas de verificação, os mecanismos de integridade de dados baseados em funções (criptográficas) de *hash* são projetados especificamente para impedir a modificação intencional indetectável.”

Algoritmos de *checksum*, CRC e *hash* apresentam propostas diferentes, mas não raramente, são tidos como sinônimos em publicações diversas.

Considerando uma função *hash h()* e uma mensagem *m*, a função *hash h(m)* gera um valor *hash w* para essa mensagem, resultado que pouco frequentemente, também pode ser

mencionado como Integrity Check Value (ICV), Modification Detection Code (MDC) ou Message Integrity Code (MIC)[18].

Para que sejam consideradas funções criptográficas de *hash* e que possam entregar um satisfatório nível de confiança da integridade dos dados, Stallings (2015) e Menezes, Oorschot e Vanstone (2001) definem alguns requisitos mínimos que os algoritmos dessa classe devem atender:

1. **Compressão –** A função deve gerar um valor de saída de tamanho fixo e único para dados de tamanhos variáveis e finitos na entrada.
2. **Eficiência –** A função deve ser fácil de computar.
3. **Resistência a pré-imagem (*preimage resistance*) –** O fluxo da função deve ser unidirecional e não pode permitir que os dados *m* sejam descobertos a partir de seus valores *hashes w*.
4. **Resistência a 2ª pré-imagem (*2nd-preimage resistance*) –** A função deve tornar computacionalmente inviável o encontro de diferentes dados *m′* que resultem no mesmo valor *hash w* a partir do conhecimento dos dados *m*.
5. **Resistência a colisão (*collision resistance*) –** A função deve tornar computacionalmente inviável o encontro de diferentes dados *m′* que resultem no mesmo valor *hash w*.

Por tais características, Stallings (2015) lembra que as funções de *hash* são primariamente destinadas a verificação de integridade de dados, uma vez que, com o uso desse método, se espera que sejam detectadas quaisquer alterações nesses dados.

Conquanto, Menezes, Oorschot e Vanstone (2001) recomendam o uso de mecanismos adicionais para garantir a integridade do próprio valor de *hash* e, consequentemente, assegurar a garantia da integridade dos dados.

Atualmente, os algoritmos de *hash* são extensivamente utilizados nas diversas aplicações de segurança em sistemas computacionais, a exemplo de protocolos de segurança na *web* (em conjunto com técnicas de assinaturas digitais).

No Apêndice A, o leitor pode conferir uma aplicação de populares algoritmos de *checksum* (CRC-32) e *hash* (SHA-256) na verificação da integridade de três versões de um

18 Os Message Authentication Codes (MACs) são um outro tipo de hash, que utiliza uma chave para criação dos códigos. Esse tipo oferece uma abordagem divergente ao proposto neste trabalho e, por isso, não serão apresentados mais detalhes sobre ele.

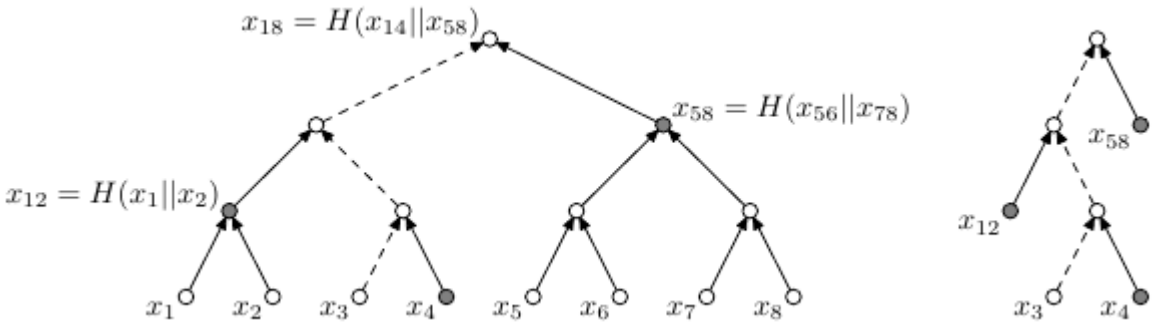
objeto digital (sua versão original e duas versões intencionalmente corrompidas), exemplificando a eficácia e os valores gerados a partir de cada função.

## 2.4 Árvores de Merkle

Um carimbo de tempo (*timestamp*) é uma forma de fornecer uma certa evidência de que tal evento ocorreu em determinado momento. Haber e Stornetta (1991) sugerem o uso de um esquema de encadeamento de registros para prover uma forma de "carimbar temporalmente" documentos digitais. Nesse esquema, um serviço de carimbo de tempo enfileira os documentos a serem registrados (carimbados) em uma ordem temporal e o registro de um documento possui informações do próprio documento e do registro do documento anterior e posterior, incluindo informações de *hash* (TRUU, 2010).

Para uma cadeia de registro de *n* documentos nesse esquema de encadeamento linear, a auditoria de um documento só é possível com o conhecimento dos *n* registros. Para aprimorar esse modelo, Bayer, Haber e Stornetta (1993) propuseram a substituição dele por um tipo de encadeamento em árvore binária, utilizando o conceito de Árvore de Merkle, que reduz a complexidade de *O(n)* para $O(log_2 n)$ em um processo de verificação.

Figura 2.3 — Diagrama de Árvore de Merkle




(a) (b)

Fonte: (TRUU, 2010)

Baseado na função unidirecional de *hash* e destinado inicialmente ao aprimoramento da eficiência no processo de assinatura digital de documentos, Merkle (1990) propôs um conceito de autenticação em árvore, modelo que ficou conhecido como Árvore de Merkle (Merkle Tree). Tal modelo é sucintamente definido por Truu (2010, p. 39, tradução nossa)

como “uma árvore binária onde os nós folha são preenchidos com alguns valores *hash* e cada nó não-folha contém o valor *hash* da concatenação de seus nós filhos”.

Utilizando técnicas de *hash* e Árvore de Merkle é possível construir uma cadeia de registros em um modelo de árvore binária, onde cada par de registro é resumido em um, sucessivamente, até que reste apenas um registro: o valor “raiz” (*hash root*). Assim, é possível obter um único valor que represente a integridade de *n* documentos, conforme é ilustrado na Figura 2.3a.

Uma relevante propriedade dessa estrutura de dados é a possibilidade de se verificar, a partir de uma Árvore de Merkle *A*, a integridade de um único registro *r* sem a necessidade de conhecimento prévio do seu valor *hash*, através da comprovação de que $r \in A$, o que Greve et al. (2018) chama de prova de filiação (*proof of membership*). Para isso, basta conhecer os níveis da árvore *A* e seus respectivos valores pares dos quais o registro pertença; a essas informações são dados nomes como “prova de verificação” (*audit proof*) ou “caminho de verificação” (*audit path*).

Esse processo pode ser acompanhado na Figura 2.3b, onde, para a auditoria de um documento $x_3$ são dados os valores pares correspondentes àqueles que dependem do próprio valor do objeto para serem gerados ($x_4$, $x_{12}$, $x_{58}$).

Para a correta auditoria, é necessário conhecer os valores pares — desde o último nível da árvore (folhas da árvore) até aquele imediatamente antecessor a raiz — e suas posições (ordem de concatenação), que no exemplo citado já está autoexplicado pela numeração dos registros.

No fim, o auditor terá um *hash* raiz gerado a partir da prova de verificação e do objeto de verificação, restando apenas comparar esse *hash* raiz com um *hash* raiz de confiança referente a essa árvore e, se forem idênticos e a função *hash* utilizada ainda seja considerada segura, comprova, com elevado grau de confiança, que $x_3$ foi registrado naquela posição daquela árvore.

## 2.5 Distributed Ledger Technologies (DLT) e blockchains

Tecnologias de registros/livros-razão distribuídos (Distributed Ledger Technologies – DLT) ou compartilhados são baseadas nas arquiteturas ponto-a-ponto (*peer-to-peer* – P2P) e permitem a replicação e sincronização de dados de forma descentralizada, utilizando

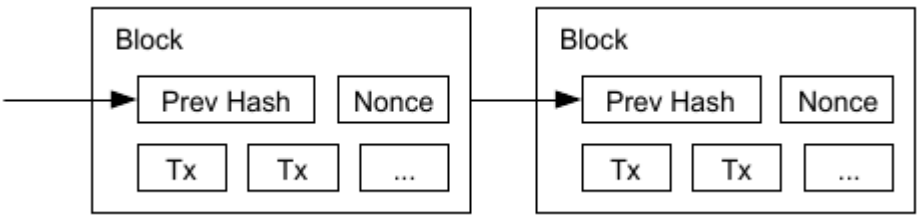
algoritmos de consenso (NATIONAL ARCHIVES AND RECORDS ADMINISTRATION, 2019; THE WORLD BANK, 2017).

Em 2008, Nakamoto (2008) propôs um sistema de moedas puramente virtual, seguro e descentralizado, baseado na arquitetura P2P, denominado Bitcoin. A tecnologia proposta para esse sistema ficou conhecida como *blockchain* e é resultado de uma combinação de técnicas já existentes e amplamente conhecidas da computação distribuída confiável, criptografia e teoria dos jogos (GREVE et al., 2018).

As *blockchains* funcionam como uma rede de nós distribuída, em que cada nó possui e mantém uma réplica de um conjunto de transações que foram processadas por nós dessa rede a partir de requisições de clientes, os quais não precisam necessariamente ser um dos nós de processamento. As transações são estruturadas em uma forma de livro-razão (*ledger*) e sua distribuição e aceitação entre os nós acontece de acordo com técnicas de consenso adotadas na rede e que podem conter algum tipo de recompensa pelo processamento das transações. Os registros das transações são sequenciais e seguem um processamento progressivo, em que se torna impraticável a adulteração de registros já processados, ao tempo que os registros de transações permanecem acessíveis para verificação e auditoria. (BASHIR, 2018; GREVE et al., 2018)

Para o Banco Mundial (2017) e Greve *et al.* (2018), o grande feito oriundo da Bitcoin foi a eliminação da necessidade de uma terceira parte de confiança em transações onde isso se fazia indispensável e tal tecnologia ganha destaque devido as propriedades de descentralização, disponibilidade e integridade, transparência e auditabilidade, imutabilidade e irrefutabilidade, privacidade e anonimidade, desintermediação, cooperação e incentivos.

Figura 2.4 — Esquema de encadeamento de blocos na Bitcoin




Fonte: (NAKAMOTO, 2008)

Um dos principais mecanismos envolvidos na tecnologia de *blockchain* é o princípio de encadeamento de blocos, ilustrado pela Figura 2.4 e que pode ter seu funcionamento resumido nos seguintes passos:

1. Um conjunto de transações formam um bloco.
2. Um novo bloco contém registro do bloco anterior.
3. O registro do bloco anterior no novo bloco valida o primeiro e torna as transações nele imutáveis.
4. A geração de novos blocos com o registro dos blocos anteriores forma uma cadeia de blocos imutáveis.

O consenso nas redes de *blockchain* podem variar de acordo com o mecanismo escolhido para tal, sendo o de prova-de-trabalho (Proof of Work) um dos mais populares e adotado na rede Bitcoin.

Além da prova-de-trabalho, alguns dos mecanismos mais notáveis listados por Bashir (2018) são: Proof of Stake, Proof of Elapsed Time, Proof of Deposit, Proof of Importance, Proof of Activity e Proof of Storage.

A escolha por algum dos mecanismos de consenso depende muito da categoria de rede *blockchain* que se está adotando. As *blockchains* abertas e permissionárias são as principais categorias desse tipo de rede consideradas na literatura (BASHIR, 2018; GREVE et al., 2018; THE WORLD BANK, 2017):

- **Blockchains abertas** – As redes abertas, a exemplo da Bitcoin, são redes públicas que permitem a participação livre e dinâmica de nós, que costumam ser anônimos. Nesse tipo de rede há uma desconfiança entre os nós participantes, o que exige a utilização de mecanismos mais caros de consenso.
- **Blockchains permissionárias** – As redes permissionárias, as vezes chamadas de federadas, podem ter um caráter público ou privado, mas possuem restrições a participação de nós por uma ou mais autoridades.

Dentre as possíveis aplicações de *blockchain* dentro e fora da área financeira documentadas por Crosby (2016), chama atenção a possibilidade de uso em serviços de notarização, aplicação vista como grande oportunidade para auxiliar atividades na área de preservação digital (INTERPARES TRUST PROJECT, 2018; LEMIEUX; SPORNY, 2017; NATIONAL ARCHIVES AND RECORDS ADMINISTRATION, 2019).

Atualmente, as DLT encontram sua maior representação na tecnologia de *blockchain* e apesar de que o uso de uma não implica no uso da outra, esses termos são normalmente usados de forma intercambiável, pois como o Banco Mundial (2017, p. 2) esclarece, “a

terminologia neste campo ainda está em evolução e definições universais ainda não foram formalizadas".

# 3 TRABALHOS RELACIONADOS

Neste capítulo são apresentados alguns trabalhos relacionados que foram considerados relevantes pela proposta apresentada, selecionados pela abordagem semelhante e em destaque nos seguintes aspectos: o uso de *blockchain* para garantia de integridade, a preocupação com a integridade em repositórios digitais ou uma abordagem combinada de ambos.

## 3.1 Um survey de plataformas para garantia de propriedades dos documentos digitais

O trabalho de Vigil et al. (2015) traz um *survey* sobre diversos esforços documentados na perspectiva da garantia dos atributos de integridade, autenticidade, não-repúdio e prova de existência para preservação de longo prazo nos arquivos, avaliando os trabalhos em três principais abordagens: baseados em carimbo de tempo, notarização e replicação.

Para os autores, o maior problema em aberto é a inexistência de sistemas que permitam a garantia dos atributos prometidos no caso de uma abrupta perda de confiança nos mecanismos criptográficos, como a descoberta de uma falha em uma das propriedades de algum algoritmo popular de *hash*.

Com base em quase todos os trabalhos analisados (e considerados para uso), os autores também destacam a preocupação na demasiada necessidade de confiança em um terceiro de confiança, já que os trabalhos abordados se baseiam fortemente no uso de uma autoridade de carimbo de tempo (Time Stamp Authority – TSA), especialmente aqueles que propõem o uso de um notário para atestar algum dos atributos defendidos.

O LOCKSS — Lots of Copies Keep Stuff Safe (MANIATIS et al., 2005) — foi o único trabalho considerado com abordagem de replicação e conforme referendado pelo *survey*, a garantia da integridade nessa proposta muito depende da honestidade e do número de nós pertencentes a rede. Apesar disso, naqueles repositórios onde o sigilo não é um requisito essencial, esse sistema pode ser uma solução para gerar uma redundância de dados geograficamente distribuída e atuar como mecanismo de correção (após a detecção e confirmação de uma corrupção).

Ainda no mesmo *survey* são documentadas as propostas de Haber e Kamat (2006) e Song e JaJa (2009), onde são sugeridas plataformas que consideram a substituição do uso e

confiança dos atestados de TSA pela publicação em mídias de ampla divulgação (Widely Visible Media – WVM), como jornais físicos ou listas de e-mails.

Em 1993, com o uso de funções criptográficas de *hash* e Árvores de Merkle, Bayer, Haber e Stornetta (1993, p. 4, tradução nossa) já documentavam o uso de WVM, a exemplo dos jornais físicos, como meio de oferecer uma confiável prova de integridade e carimbo de tempo, uma vez que, um *hash* de referência podia ser publicado nesse meio, onde eram datados e amplamente divulgados: “Os jornais funcionam como um amplo registro público disponível, cuja preservação de longo prazo em vários lugares torna sua adulteração muito difícil.”

Partindo da premissa de confiança em WVM, Song e JaJa (2009) propõem uma plataforma denominada Audit Control Enviroment (ACE), que utiliza listas de e-mails como solução de WVM para divulgação de valores *hash* — também gerados a partir de Árvores de Merkle — que servem de referência para a verificação e garantia da fixidez dos acervos nos repositórios digitais.

Apesar da combinação elaborada de mecanismos que proporcionam certa robustez a essa proposta, a utilização de listas de e-mail como WVM pode ser um problema para garantir a fixidez dos acervos no longo prazo, além da fragilidade apontada por Vigil et al. (2015) na regeneração dos valores de evidência a partir de um novo algoritmo de *hash*, que apenas acontece em um certo nível da arquitetura.

## 3.2 Framework para garantia da fixidez de mementos

Aturban et al. (2019) enaltecem os esforços feitos por alguns arquivos *web* (repositórios de páginas) no sentido de tentar capturar e arquivar como se apresentam as páginas da *web* em determinado momento. Esse tipo de material é chamado de memento[19]: uma versão arquivada de uma página *web* original.

Os autores destacam a preocupação em conseguir atestar a fixidez dos mementos desses arquivos e para isso, recorrem a recomendações para o estabelecimento de confiança em repositórios digitais já citadas neste trabalho (como o TRAC) para propor um *framework* que permita garantir a fixidez desses objetos.

Archival Fixity Server é o nome dado pelos autores ao serviço, que pela proposta, será o intermediador responsável de capturar o memento, gerar suas informações de fixidez,

19 https://tools.ietf.org/html/rfc7089

disseminar essas informações entre outros arquivos e verificar posteriormente. Os autores informam que a geração e disseminação dessas informações podem seguir uma abordagem atômica ou de bloco.

Como é descrito no trabalho, na abordagem atômica o serviço gera um *manifest* individual com as informações de fixidez de um determinado memento e o dissemina entre os serviços de arquivo. Na abordagem de bloco, inúmeras informações de fixidez são registradas em um mesmo *manifest*, que são encadeados e cada bloco de informações de fixidez (*manifest*) é disseminado entre os serviços de arquivo, em um estilo de *blockchain*.

## 3.3 Âncora em blockchain das informações de fixidez da camada intelectual dos documentos digitais

Os trabalhos de Collomosse *et al.* (2018) e Bui *et al.* (2019) documentam uma audaciosa proposta que parte de uma abordagem do uso de redes neurais profundas (Deep Neural Network – DNN)[20] para registrar em *blockchains* uma evidência da camada intelectual dos documentos digitais arquivados em repositórios.

Como documentado pelos autores, essa evidência é um valor *hash* gerado utilizando uma função *hash* típica (SHA-256, por exemplo) e esse valor é associado com alguns metadados para significar e descrever tal evidência e o documento ao qual se referem. A forma para geração de *hash* e extração da camada intelectual pode variar de acordo com o formato do documento e essas informações também deverão estar associadas ao objeto.

O conjunto de informações relativo a tal documento é registrado em uma *blockchain* para preservar a imutabilidade da prova de evidência desse documento, permitindo que independente das eventuais transformações que ele possa sofrer, sempre poderá ser verificado que a informação (camada intelectual) do documento é a mesma.

A plataforma, denominada ARCHANGEL, é apresentada como uma solução auxiliar para o controle de fixidez aos sistemas de gerenciamento de documentos digitais nos repositórios, não prevendo o gerenciamento e armazenamento desses documentos.

Os autores propõem uma arquitetura baseada em *blockchains* permissionárias com consenso do tipo Proof of Authority ou Proof of Work e consideram que tais redes sejam compostas por outras (diversas) instituições de arquivo.

20 Um ramo do estudo de aprendizado de máquina e redes neurais artificiais, campos de estudo da Inteligência Artificial.

Devido a natureza das *blockchains*, a credibilidade dos registros é proporcional a quantidade de nós participantes, podendo tornar questionável a credibilidade dos registros quando essas redes são formadas por poucos nós, exigindo maior atenção quando se considera o contexto de uma rede permissionária e de nicho (apenas instituições de arquivo).

Conforme o relatório do projeto (UNIVERSITY OF SURREY; NATIONAL ACHIVES (UK); OPEN DATA INSTITUTE, 2019) a proposta já teve uma implementação com algumas instituições parceiras e seu principal enfoque é na geração de *hashes* a partir de redes neurais profundas, enfatizando objetos de imagem e vídeo.

Apesar do foco em preservação digital, o trabalho não discute muito sobre sua consonância com o modelo arquitetural do OAIS e não deixa claro como seria tratável a fixidez dos metadados relacionados, que são habilitados a serem atualizados, mesmo quando não há alterações no objeto digital preservável (CONSULTATIVE COMITTEE FOR SPACE DATA SYSTEMS, 2012).

## 3.4 Registro em blockchain da relação orgânica de documentos digitais

Para Lemieux e Sporny (2017, p. 1437, tradução nossa), DLT, como a *blockchain*, são apontadas como potenciais sistemas de preservação digital nos aspectos arquivísticos, visto que “preocupa-se com a manutenção de registros, que podem ser descritos como documentos”.

No entanto, os autores demonstram preocupação sobre utilizar tais sistemas nas ações de preservação digital, pois na perspectiva deles, apesar desse modelo parecer adequado em alguns aspectos, ele sofre de diversas falhas. O trabalho então, foca em debater e propor um modelo para contornar uma dessas falhas: a inexistência da relação orgânica[21] entre os diversos documentos. Ao lado da integridade, a garantia dessa propriedade é considerada como um dos elementos essenciais para confiança de um documento, como pôde ser demonstrado na Figura 2.1 (página 22).

Segundo os autores, a forma de funcionamento de uma *blockchain* com mecanismos de consenso do tipo de prova-de-trabalho, como a Bitcoin, pode até transparecer que mantêm a relação orgânica, entretanto, “mesmo que a natureza do tempo ordenado dos registros das

21 “Vínculos que os documentos arquivísticos guardam entre si e que expressam as funções e atividades da pessoa ou organização que os produziu.” (CONSELHO NACIONAL DE ARQUIVOS, 2016, p. 36)

transações seja preservada, a ligação para seu contexto processual e a relação com outros registros transacionais relativos ao mesmo procedimento não é" (LEMIEUX; SPORNY, 2017, p. 1439, tradução nossa).

Então, Lemieux e Sporny (2017) propõem um modelo e sintaxe de dados como registro em que a relação orgânica possa ser devidamente tratada em sistemas de preservação baseados em *blockchain* e que possa ser criptograficamente verificada. De acordo com a proposta, esse modelo, inclusive, pode estabelecer a relação orgânica de registros que estejam em diferentes redes (*ledgers*).

O modelo é baseado em uma estrutura de dados utilizando princípios e padrões da *web*, especificadamente o padrão JavaScript Object Notation (JSON)[22], mas não é compatível com os modelos de *blockchains* consolidadas atualmente (Bitcoin, Ethereum etc), sendo necessário para isso, gerar *hashes* dos estados do sistema para registrá-los nessas redes e assim, conseguir provar a integridade dos registros posteriormente.

## 3.5 Blockchain permissionária para validação de documentos assinados digitalmente

Preocupados com a validade ao longo do tempo das assinaturas digitais utilizadas em documentos arquivísticos, Bralić, Kuleš e Stančić (2017) sugerem um modelo de preservação da validade das assinaturas no longo prazo, denominado TrustChain.

A arquitetura pensada é baseada na criação de uma rede *blockchain* permissionária, onde apenas instituições acreditadas e com fins em comum farão parte e endossarão a credibilidade do serviço.

O serviço é proposto como um módulo intermediador entre o arquivo e a rede *blockchain*, pois como os autores deixam claro, "TrustChain é um sistema que complementa outros sistemas de gerenciamento de documentos digitais, arquivos digitais ou sistemas de repositório e não substitui eles" (BRALIĆ; KULEŠ; STANČIĆ, 2017, p. 92, tradução nossa) e por isso, os documentos são mantidos e gerenciados em sistemas paralelos, já projetados e consolidados para esse fim.

O modelo prevê que um arquivo, participante ou não da rede, solicite nela o registro da validade de um documento assinado. O módulo intermediador então, verifica e valida a

22 https://tools.ietf.org/html/rfc8259

assinatura com uma autoridade certificadora[23]. No caso de invalidade, o registro é imediatamente negado, mas no contrário, o registro é assinado pelo nó avaliador daquele instante e é posto em uma fila de registros pendentes, para que os demais nós efetuem a mesma verificação e admitam a validade da assinatura (e do documento), votando pelo registro na *blockchain* associada.

Conforme mencionado na Seção 3.3, a confiança dos registros em *blockchains* permissionárias está fortemente ligada ao número de nós pertencentes a tais redes, mas considerando um cenário em que haja significativa adoção a tal rede, os arquivos poderão solicitar e conseguir um atestado que determinado documento assinado em seu acervo é ou não autêntico, independente da validade da assinatura ou existência dos serviços relacionados ao certificado digital associado. Essa validação acontecerá com a busca e verificação do registro na rede relativa a tal documento que, caso existindo, provará que a assinatura era válida e foi confirmada por um número de instituições relevantes naquele momento.

Esse trabalho foi resultado de pesquisas do International Research on Permanent Authentic Records in Electronic Systems (InterPARES, 2018), destacável aqui pelo reflexo na preocupação dos autores com os conceitos arquivísticos, que pode ser visto com a proposta de construir um modelo de registro que inclui metadados de um documento, com campos pertinentes a descrição arquivística.

## 3.6 Serviços intermediadores para blockchains

Por fim, são apresentados aqui alguns serviços documentados que propõem uma forma de atestar a integridade, prova de existência e carimbo de tempo de objetos digitais utilizando *blockchains*. Ressalta-se que não houve uma busca dedicada a procura desses serviços e que a seleção apresentada aqui é derivada de menções dos mesmos na literatura visitada.

Esses serviços disponibilizam um meio onde um usuário (indivíduo ou outro serviço) pode solicitar a inclusão do *hash* de um objeto digital para ser registrado em uma *blockchain* e com a referência da transação onde esse *hash* foi registrado, eles podem comparar um *hash* do objeto digital gerado instantaneamente com aquele gravado na *blockchain*, podendo assim, atestar que aquele objeto está integro e que existiu e foi registrado naquele tempo.

23 “Organização que emite certificados digitais obedecendo às práticas definidas na Infra-estrutura de Chaves Públicas – ICP.” (CONSELHO NACIONAL DE ARQUIVOS, 2016, p. 10)

O serviço Proof of Existence[24] fornece uma interface *web* intuitiva para que seja usada por indivíduos, assim como uma API (Application Programming Interface) para utilização por sistemas. O serviço proporciona o registro do *hash* diretamente na Bitcoin e para isso, exige do usuário o pagamento de uma taxa (0,00025 BTC) para que isso seja efetivado.

Para diluir o custo da taxa ou até mesmo evitar cobrá-la dos usuários, serviços como OriginStamp (GIPP; MEUSCHKE; GERNANDT, 2015)[25], Chainpoint (VAUGHAN; BUKOWSKI; WILKINSON, 2016)[26], Stampery (CRESPO; GARCÍA, 2017)[27] e OpenTimestamps[28] apresentam um modelo em que ao invés de registrar o *hash* do documento do usuário diretamente nas *blockchains*, esse *hash* é encadeado com diversos outros *hashes* (a partir da solicitação de outros usuários, por exemplo) em uma Árvore de Merkle e eles registram apenas o *hash* raiz dessa árvore nas *blockchains*.

Como já demonstrado na Seção 2.4, o *hash* raiz de uma Árvore de Merkle pode verificar criptograficamente um único *hash* folha, mas para isso é necessário conhecer a prova de verificação (*audit proof*).

Então, assim como foi necessário uma etapa intermediária para o registro (com o intuito de reduzir custos no registro), é necessário uma etapa de mesmo nível para a verificação.

Para esse processo os serviços entregam uma prova (*audit proof*) onde, a partir do objeto original e associado com os valores da prova, os usuários podem reconstruir a árvore em que tal objeto fez parte e podem comparar o seu valor *hash* raiz resultante com o valor referente já registrado na *blockchain*, alcançando assim um resultado de registro semelhante, mas com um menor custo financeiro, em troca de um mínimo custo computacional adicional.

24 https://proofofexistence.com/
25 https://originstamp.com/
26 https://chainpoint.org/
27 https://stampery.com/
28 https://opentimestamps.org/

# 4 SOLUÇÃO PROPOSTA

A solução aqui proposta é uma aplicação da combinação de funções *hashes*, Árvores de Merkle e DLT, como as *blockchains*, para criar e tornar imutáveis valores *hashes* referentes a conjuntos de objetos digitais custodiados por RDC. Este processo visa ter esses valores como âncoras de referência na verificação e auditoria da fixidez dos acervos nesses repositórios.

Com o uso dessa abordagem os RDC podem auditar a fixidez dos objetos e das próprias informações de fixidez, desde que essas informações estejam associadas a seus próprios objetos através de identificadores únicos, que no contexto proposto já é previsto com o uso do conceito de Pacotes de Arquivamento de Informações (Archival Information Packages – AIP), os quais devem ser gerenciados por um sistema apropriado e dedicado a isso, de acordo com o normatizado pelo OAIS.

Também é possível apoiar os RDC na detecção confiável de objetos danificados e na reposição desses a partir de sistemas de *backup* e redundância, fornecendo a garantia de que o objeto a ser substituído está realmente comprometido e que o objeto candidato a substituto é, de fato, uma cópia autêntica dele.

Aqui, as *blockchains* podem ser consideradas como ferramentas muito mais robustas de mídias de ampla divulgação (WVM), permitindo abertamente (no caso de *blockchains* abertas) publicações e consultas de informações nessas redes a qualquer interessado, enquanto mantêm a segurança e credibilidade dessas informações, conforme corrobora Lemieux (2017, p. 10, tradução nossa): “A proteção da integridade dos registros, ou pelo menos a indicação de que a integridade foi afetada, é um dos pontos fortes das soluções baseadas em *blockchains* para a preservação digital.”

Os processos e os custos relativos ao registro em *blockchains* variam de acordo com a forma de uso e a rede escolhida, sendo possível o uso de serviços intermediários para facilitar o registro e reduzir os custos (conforme propostas apresentadas na Seção 3.6), atenuando assim, uma das preocupações apresentadas por Lemieux (2017) e Smith (2017) para esse tipo de aplicação.

Nesta proposta, é previsto que as informações de fixidez sejam geradas por mais de um algoritmo *hash* e que, para cada valor *hash* raiz de uma árvore referente a um algoritmo, se tenha uma rede *blockchain* vinculada.

Recomenda-se que a geração das informações de fixidez seja feita utilizando dois ou mais algoritmos de *hash* com funções de paradigmas diferentes (SHA-3 e RipeMD, por exemplo). Essa estratégia pode permitir contornar a preocupação de segurança documentada por Vigil *et al.* (2015) e abordada na Seção 3.1 (página 40), permitindo manter a confiabilidade da fixidez do acervo mesmo que um dos algoritmos *hash* utilizados se torne imediatamente inseguro e abra brechas para a contestação de fixidez do acervo (como saber se tal brecha de segurança já não estava em uso por um atacante no repositório?). Para demonstrar a manutenção da fixidez nesse caso, bastará uma auditoria de todo o acervo utilizando as informações de fixidez de um algoritmo *hash* distinto e, se for o caso, já criar e registrar novas informações de fixidez utilizando um novo algoritmo *hash*.

A mesma recomendação de redundância e diversificação é feita para a escolha das redes *blockchains*, que deve priorizar redes abertas e bem consolidadas, sempre buscando se precaver de eventuais incredibilidades que essas redes possam ter no longo prazo (LEMIEUX, 2017; SMITH, 2017).

## 4.1 Premissas da solução

Conforme descrito nas recomendações de referência para certificação de repositórios digitais candidatos a confiáveis, esses repositórios precisam gerenciar os objetos digitais de acordo com o modelo OAIS, o que implica assumir que os objetos preserváveis e suas informações relacionadas estarão empacotadas em um AIP e, portanto, **garantindo a fixidez desse pacote consequentemente se garante a fixidez do seu conteúdo**, que pode conter mais do que apenas um objeto digital e seus metadados (CONSULTATIVE COMITTEE FOR SPACE DATA SYSTEMS, 2012).

Partindo dessa premissa, foi pensada uma plataforma que permita aos administradores de um arquivo, o registro e posterior auditoria da fixidez dos AIP dos RDC. Essa plataforma também é passível de ser utilizada com objetos além dos AIP e que não estejam em conformidade com o OAIS, mas deve-se observar que sua construção foi guiada pelas diretrizes estabelecidas por esse padrão e, por isso, pode ser necessário eventuais adequações de acordo com o caso.

Para facilitar a referência a tal plataforma, essa será nomeada como Archive Fixity Anchor (**AFA**) e, assim como a maioria das propostas apresentadas no Capítulo 3, esta também visa complementar um serviço de gerenciamento dos objetos digitais já existente,

dando suporte ao complexo trabalho de gerenciar os diversos aspectos e relações dos objetos digitais preconizados no OAIS.

Ao estar em conformidade com o modelo OAIS, o sistema de gerenciamento dos objetos digitais seguirá procedimentos normatizados (como a formação de um AIP e sua identificação com um identificador único, por exemplo), o que permitirá a AFA se comportar da maneira esperada no processamento dos registros.

Pelo OAIS, o sistema para um RDC deve se preocupar com a fixidez do objeto em todas as suas fases (ingestão, preservação e acesso). Todavia, a fase de ingestão, que acontece por meio de SIP, é transitória e a garantia da fixidez nessa fase deve ser tratada particularmente, pois a confiança deve ser negociada com o produtor, não havendo uma preocupação real de preservar as informações dessa fase por longos períodos. Por esse motivo, a preocupação deste trabalho é voltada especialmente à fase de preservação.

O acesso dos consumidores aos objetos é viabilizado pelos DIP, que podem ser gerados como cópias dos objetos nos AIP ou em uma forma convertida de um ou mais AIP. Dependendo da política do repositório, um DIP pode ser gerado por demanda ou pode ser gerado previamente e mantido preservado para reduzir o processamento de entrega na requisição dos consumidores (no caso de um objeto constantemente requisitado). Contudo, a proposta aqui apresentada não visa nesse momento contemplar o tratamento da preservação desses DIP, ainda que seja possível, é necessário uma melhor análise para que se possa afirmar sua compatibilidade.

Apesar da plataforma ser aplicável a outros cenários, a arquitetura elaborada foi montada com foco em repositórios digitais que já sejam ou tenham pretensões de demonstrar confiabilidade e que, por isso, gerenciam seus objetos digitais segundo as recomendações do OAIS.

A preocupação em restringir a aplicação da proposta apenas ao gerenciamento da fixidez dos AIP nesse momento, se dá devido a posição de importância desse pacote dentro de um RDC e sua padronização permite elaborar tratamentos específicos para suas particularidades. Um exemplo disso, é o tratamento das mudanças sofridas legitimamente pelo pacote e que muda sua estrutura em nível de bits, enquanto mantém sua mesma estrutura intelectual. Essa situação é prevista no OAIS e acontece quando informações são incrementadas ao PDI do AIP, por exemplo, e seu Content Information não é alterado, ou quando esses não são alterados o suficiente a ponto de serem considerados um novo pacote em nível intelectual (e ganharem um novo identificador). Deve-se também considerar a possibilidade de exclusão desses objetos, uma vez que é possível que precisem serem

preservados por muito tempo, mas não será necessariamente por um tempo indefinido, o que pode variar de acordo com o objeto e as políticas definidas para tal.

Apesar do termo “pacote”, o modelo OAIS não define que a estrutura do AIP esteja acondicionada em um único recipiente (“um arquivo” no âmbito computacional), mas uma vez que é previsto o uso de recipientes para o gerenciamento desses pacotes por sistemas de referência para repositórios digitais (Archivematica[29] e RODA[30]), se propõe que inicialmente a AFA preveja o tratamento de objetos digitais apenas daqueles que estejam encapsulados em um recipiente, prevendo unidade e uniformidade nos objetos monitorados para o correto tratamento desses e execução de suas funções.

Nessa proposta, o tratamento de documentos assinados digitalmente segue uma abordagem recomendada pelo Projeto InterPARES (2018) de realizar o tratamento desses documentos na fase de ingestão no repositório, verificando as assinaturas e adicionando essa validação aos metadados associados ao objeto nesse momento e, portanto, a **garantia da fixidez do AIP também pode garantir sua autenticidade vinculada a uma assinatura digital**:

> Ao inserir um documento arquivístico com uma assinatura digital em um arquivo digital, a validade da assinatura digital pode ser verificada e as informações podem ser registradas nos metadados. Após a verificação, o documento arquivístico é armazenado em um Pacote de Arquivamento de Informações (AIP) com os metadados associados. (INTERPARES TRUST PROJECT, 2018, p. 22, tradução nossa)

## 4.2 Arquitetura da solução

De acordo com os recursos propostos e prezando pela simplicidade da plataforma, foi pensada uma arquitetura em que tais recursos fossem agrupados em dois módulos: um módulo de registro local e auditoria, chamado aqui de AFA-LOG e um módulo de registro e consulta em DLT, chamado de AFA-DLT. Os recursos e funcionalidades foram agrupados nesses módulos de acordo com suas relações, conforme apresentado a seguir:

- **AFA-LOG –** Esse é o módulo responsável por criar (para registro) e recriar (para verificação) as Árvores de Merkle a partir dos registros e provê uma interface ao

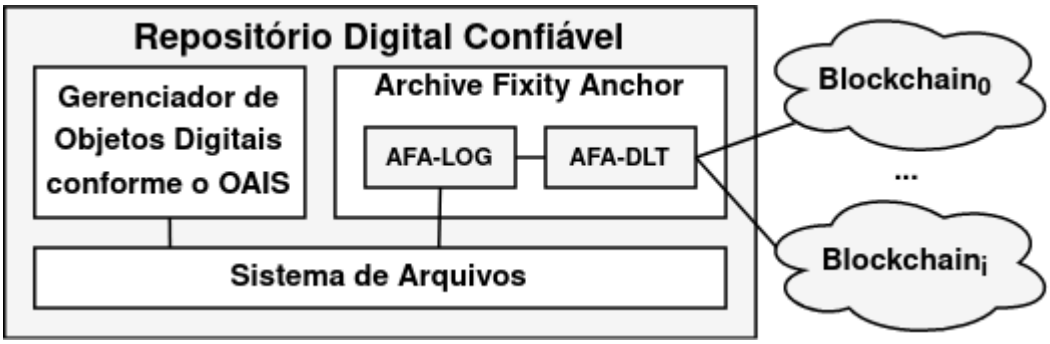
29 https://www.archivematica.org/pt-br/docs/archivematica-1.12/user-manual/archival-storage/archival-storage/
30 https://earkaip.dilcis.eu

administrador do repositório para que esse efetue o registro de inclusão ou modificação dos AIP na plataforma e execute a auditoria dos registros.

- **AFA-DLT –** Esse é o módulo dedicado a estabelecer a comunicação com as DLT ou serviços intermediários com a mesma finalidade, para solicitar o registro nas redes ou recuperar as informações delas.

A Figura 4.1 ilustra a arquitetura da solução proposta e sua relação com os componentes do sistema.

Figura 4.1 — Arquitetura da solução proposta




Fonte: Próprio autor.

### 4.2.1 Módulo de registro local e auditoria (AFA-LOG)

O módulo de registro local e auditoria, já denominado por AFA-LOG, é o elemento da plataforma proposta responsável por fornecer uma interface de acesso e administração para a inclusão e modificação dos registros, geração das informações de fixidez e realização das devidas verificações de fixidez desses registros.

É previsto o uso de um documento, gerenciado pelo AFA-LOG, que consolidará as informações referentes aos pacotes de informação do repositório cadastrados na AFA. Nesse documento devem ser registradas as informações necessárias ao processo de auditoria de fixidez (algoritmo *hash*, valor *hash* e prova de verificação) de um conjunto de pacotes e as informações de registro em *blockchain* vinculadas a esse conjunto.

Se propõe a criação de quantos documentos forem necessários para o registro dos pacotes no repositório e que a quantidade de registros em cada um desses documentos seja

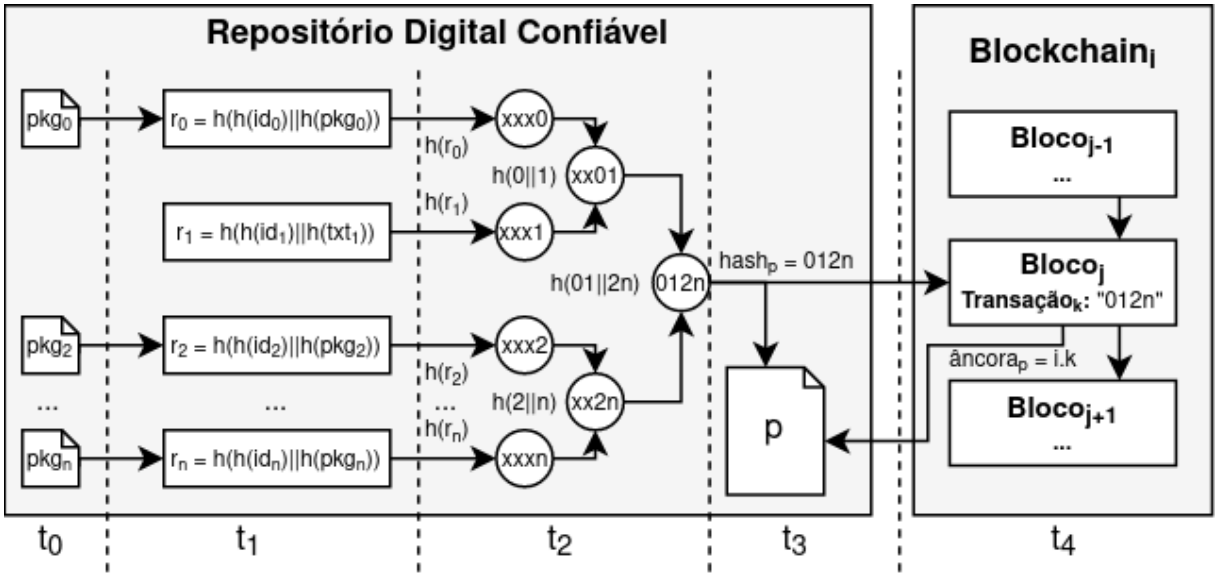
delimitada por critérios a serem definidos pelo administrador do sistema, como um intervalo de tempo ou uma quantidade de registros.

A ideia, então, é ter um conjunto de *p* documentos que documentem informações necessárias a verificação da fixidez de *n* pacotes. Para facilitar a referência a essas entidades, o conjunto de documentos será nomeado por Livro de Registros AFA ou simplesmente “livro” e cada documento individual desse livro será chamado de “página”.

Ao satisfazer o critério de delimitação de registros em uma página *p*, considera-se que essa página é fechada e que novos registros acontecerão na página *p+1*. O fechamento da página significa a criação das Árvores de Merkle a partir dos registros, a vinculação das informações de fixidez relacionadas a eles e o registro em *blockchains* dos valores *hashes* de referências gerados pelas árvores.

O registro em *blockchain* acontece pela invocação do AFA-DLT e a entrega a esse do valor *hash* e da informação de qual algoritmo utilizado para gerá-lo. Essa solicitação acontecerá para cada valor $hash_p$ gerado por um algoritmo *hash* diferente referente a mesma página *p*. As informações relacionadas a cada valor *hash* registrado em *blockchain* devem ser retornadas pelo AFA-DLT para o devido registro na própria página *p*.

Figura 4.2 — Diagrama sequencial de funcionamento dos registros na proposta




Fonte: Próprio autor.

O processo de registro executado pela plataforma é ilustrado por um diagrama de sequência na Figura 4.2. Nele, um conjunto de *n* pacotes é criado no repositório em um

instante $t_0$ e no instante $t_1$ seguinte esses pacotes são cadastrados na plataforma AFA através do AFA-LOG, que com o uso de uma função *hash h()* criará um valor *hash* para esses pacotes a partir de seus conteúdos e identificadores, com exceção do registro $r_1$, que visa registrar um evento sem o rastreio dos bits de um pacote (*hash* do conteúdo). Os detalhes sobre essas estratégias de registros são melhor apresentadas na Seção 4.3.

No instante $t_2$ do mesmo diagrama de sequência ocorre a criação de uma Árvore de Merkle com *h()* a partir dos registros na Plataforma e que resulta na geração de um valor raiz $hash_p$, que será registrado localmente junto a prova de verificação desses registros na página $p$ no instante $t_3$. O processo de registro segue no instante $t_4$ com a invocação do AFA-DLT para registro do valor $hash_p$ na $blockchain_i$ e o registro em $p$ pelo AFA-LOG das informações de registro na *blockchain* devolvidas pela rede através do AFA-DLT.

Apesar de não considerado no diagrama, é possível o uso de serviços intermediadores para efetivar o registro em *blockchain* visando poupar custos financeiros — inerentes as transações realizadas nessas redes — nesse caso, os serviços deverão retornar informações adicionais que serão repassadas pelo AFA-DLT e que também deverão ser registradas em $p$, para possibilitar a validação futura do $hash_p$.

No momento de atestar a fixidez dos pacotes do repositório com a AFA, a Plataforma verificará se o valor $hash_p$ local (registrado em $p$) é idêntico ao seu valor âncora, difundido na rede $blockchain_i$ pela $transação_k$. Caso essa igualdade se confirme, a Plataforma repetirá paralelamente (sem registrar informações) os passos contidos nos instantes $t_1$ e $t_2$ do diagrama da Figura 4.2, mas ao invés de registrar o valor raiz recriado ($hash_{p'}$), esse será comparado ao valor arquivado em $p$ (e já atestado pela consulta a rede *blockchain*) e, portanto, se $hash_{p'} = hash_p \leftrightarrow hash_p \in transação_k \wedge transação_k \in blockchain_i$, pode-se assumir que os $n$ pacotes associados permanecem intactos e os $n$ registros estão consistentes.

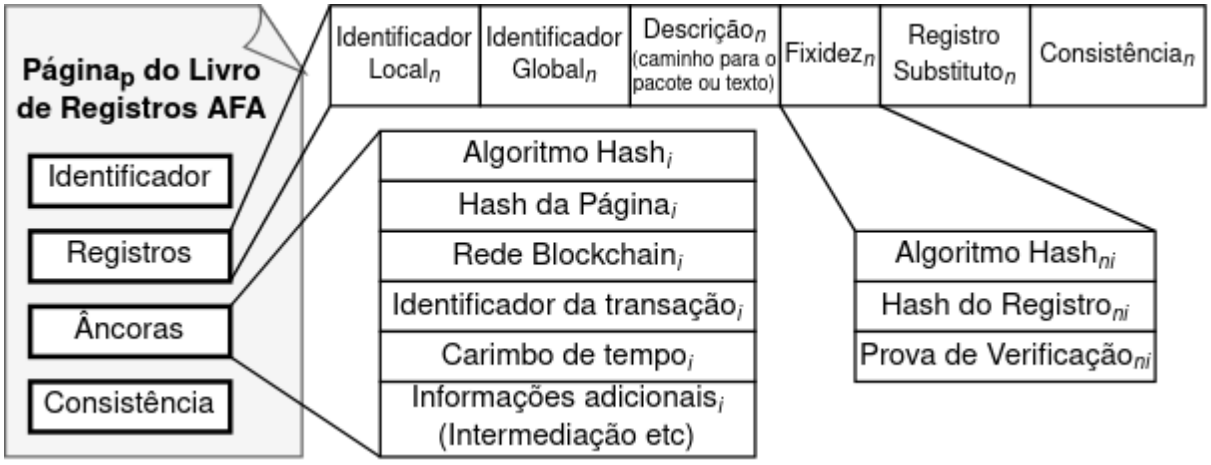
### 4.2.2 Página de registros (Livro de Registros AFA)

As páginas do Livro de Registros AFA são fundamentais para o funcionamento do AFA-LOG e, consequentemente, da plataforma, pois são nesses documentos que a plataforma realiza os registros necessários a proposta e deles são extraídos as informações pertinentes as devidas verificações, sendo a consistência de seu conteúdo atestado por valores âncoras registrados em DLT.

Em respeito às diretrizes da preservação digital, as páginas do livro devem ser documentos constituídos de texto puro e as informações neles registradas devem estar representadas e organizadas em estruturas formais, humanamente e computacionalmente fáceis de interpretar (utilizando padrões como XML[31] ou JSON, por exemplo). A estruturação dos dados dessa forma permite maior legibilidade, facilita o desenvolvimento de aplicações para seu gerenciamento e possibilita que as informações possam ser facilmente migradas ao longo do tempo para outras formas de representação.

A organização dessas páginas e a independência das informações nelas registradas permitem que os registros desses documentos sejam ocasionalmente verificados de forma manual, mesmo sem a execução da plataforma, bastando para isso conhecer a estrutura das informações dessas páginas.

Figura 4.3 — Estrutura das páginas do Livro de Registros AFA




Fonte: Próprio autor.

A Figura 4.3 ilustra os campos de informações e como esses campos estão estruturados nas páginas do Livro de Registros AFA e a seguir são descritos os detalhes pertinentes ao entendimento de cada um desses campos de informação e como eles se relacionam:

1. **Identificador –** Um número inteiro (iniciando em zero) sequencial e único na plataforma para identificar a página. É utilizado para referenciar a página e registros substitutos dentro da plataforma.
2. **Registros –** Informações relativas aos pacotes de informação. Divide-se em outros 6 campos.

31 https://www.w3.org/TR/xml

2.1. **Identificador Local –** Um número inteiro (iniciando em zero) sequencial e único na página que identifica o registro na plataforma junto ao número da página.

2.2. **Identificador Global –** Uma identificação única atribuída pelo sistema de repositório, gerado a partir de especificações disponíveis para esse fim, como o Universally Unique Identifier (UUID)[32]. Identifica o pacote no sistema de repositório e é uma das informações necessárias a formação do *hash* referente ao pacote em questão.

2.3. **Descrição –** Uma referência ao pacote, pela descrição do caminho no sistema para obtê-lo (de acordo com especificações para isso[33]) ou um texto que justifique um evento referente a um registro anterior correspondente ao mesmo Identificador Global em questão. Quando o texto for utilizado, o *hash* desse texto será usado para a criação do *hash* correspondente ao respectivo registro, do contrário, o *hash* do próprio pacote será utilizado para isso.

2.4. **Fixidez –** Informações de fixidez referentes ao registro. Para cada algoritmo *hash* utilizado, existirão outros 3 campos relacionados.

2.4.1. **Algoritmo Hash –** Algoritmo *hash* utilizado na geração de um valor *hash* para o registro.

2.4.2. **Hash do Registro –** Valor *hash* relacionado ao registro, gerado a partir do *hash* da concatenação do valor hash do Identificador Global e do valor *hash* da Descrição do registro.

2.4.3. **Prova de Verificação –** Informações relacionadas ao registro necessárias para a reconstrução da Árvore de Merkle a partir do valor *hash* do registro.

2.5. **Registro Substituto** – Informações de página e registro que apontam para uma versão atualizada do registro em questão.

2.6. **Consistência –** Uma marcação de que as informações do registro estão consistentes (íntegras) ou não. É utilizado pelo processo de auditoria para indicar a situação do registro após os devidos processos de verificação.

3. **Âncoras –** Informações relativas ao registro das informações de fixidez do conjunto de registros da página na *blockchain*. Para cada rede utilizada, poderão existir outros 6 campos relacionados.

32 https://tools.ietf.org/html/rfc4122
33 https://tools.ietf.org/html/rfc3986

3.1. **Algoritmo Hash –** Algoritmo *hash* utilizado na geração de um valor *hash* para a página.

3.2. **Hash da Página –** Valor *hash* raiz da Árvore de Merkle gerada a partir dos registros da página (*hashes* dos registros). Esse valor será registrado em DLT para se ter um valor âncora de confiança para uso nos eventuais processos de auditoria.

3.3. **Rede Blockchain –** Rede *blockchain* utilizada para registro do *hash* da página pelo AFA-DLT.

3.4. **Identificador da transação –** Identificação da transação na rede *blockchain* que permite recuperar os dados registrados nela (valor âncora) posteriormente.

3.5. **Carimbo de tempo –** Um valor de padrão definido pela rede *blockchain* que marca o instante no tempo em que a transação contendo o valor âncora foi registrado na rede, a exemplo do Epoch Unix[34] utilizado na *blockchain* Bitcoin.

3.6. **Informações adicionais –** Informações adicionais necessárias para auditoria do valor *hash* raiz. Importante para cenários com a utilização de serviços intermediários para registro em *blockchain*, que demandem informações complementares a recuperação do valor âncora registrado.

4. **Consistência –** Uma marcação de que as informações da página estão consistentes (íntegras) ou não. É utilizado pelo processo de auditoria para indicar a situação da página após os devidos processos de verificação.

### 4.2.3 Módulo de registro e consulta em DLT (AFA-DLT)

O módulo de registro e consulta em DLT, outrora denominado AFA-DLT, é o responsável por se comunicar com as redes ou serviços de *blockchain* e intermediará as requisições do AFA-LOG para registros e consultas a essas redes.

Esse é um módulo dinâmico e que deve permitir certa flexibilidade, pois a sua responsabilidade de comunicação com as DLT inclui possíveis adaptações para adição ou remoção de serviços e redes, além de adequações para modificações nos processos de interação com essas entidades.

34 https://pt.wikipedia.org/wiki/Era_Unix

Além dos processos de interação com os serviços e redes, o AFA-DLT deve envolver e viabilizar a fácil administração de elementos necessários à interação com as DLT, como o gerenciamento de carteiras e seus recursos, necessários para efetivar transações nas principais redes.

O AFA-DLT deve fornecer ao AFA-LOG e ao administrador as opções dos serviços disponíveis aptos a processar os pedidos de registros realizados nos eventos de fechamento das páginas do livro.

De acordo com as configurações definidas, no fechamento de uma página o AFA-LOG invocará o AFA-DLT para que se registre o valor âncora referente a página fechada e para isso passará como parâmetros a rede a ser utilizada e a informação a ser registrada ($hash_p$). O AFA-DLT, então, passará a executar os procedimentos necessários para cumprir a missão de registro e deverá notificar o AFA-LOG do resultado de seus esforços, fornecendo informações adicionais para tratamento das ocorrências no caso de falha.

Devido a latência inerente nas redes *blockchains* existente entre a efetivação da transação e sua confirmação de aceitação pela rede, é necessário que o AFA-DLT seja eventualmente chamado para consultar a situação das transações por ele realizadas e que, por ventura, ainda não tenham sido confirmadas.

É fundamental que a plataforma obtenha a confirmação de que a transação foi aceita, pois é a partir dela que o AFA-LOG obterá o carimbo de tempo para registro na página, consolidando o fechamento da mesma ou o AFA-DLT adotará medidas para tratar a situação no caso dessa confirmação não ser efetivada.

A realização de consultas às DLT deve acontecer durante os eventos de auditoria. Nesse processo, o AFA-DLT se comunica às respectivas redes a partir dos parâmetros de rede e identificador da transação (ou adquirindo esses parâmetros a partir de informações adicionais dos serviços intermediadores) para resgatar os dados da transação difundida na rede e poder atestar ao AFA-LOG, a partir do valor âncora confiável na rede, de que as informações de fixidez contidas na página em processo de verificação são consistentes ou não.

## 4.3 Aplicação da solução

Conforme apresentado como premissas, o OAIS prevê que os AIPs sofram alterações a nível de bits enquanto mantêm sua integridade intelectual ou sejam excluídos do repositório.

Essa abordagem diverge da natureza imutável dos registros em *blockchains*, onde a improbabilidade de adulteração acaba por fornecer a confiança e garantia necessária ao registro das informações de fixidez dos pacotes nos RDC.

Para contornar essa dificuldade na atualização de um registro ao mesmo tempo em que se tenta manter a mesma confiança desses registros, opta-se pela abordagem já citada por Lemieux (2017), de criar um novo registro que atualize seu antecessor. Para isso, um esquema foi montado para manter todos os registros e continuar permitindo que esses possam ser auditados. Esse último caso é possível pela persistência das informações de fixidez na plataforma e pela validação do registro atual (se ambos possuem o mesmo identificador e se a versão atual está íntegra).

Considerando as possíveis mudanças que um AIP possa sofrer, a plataforma prevê as seguintes situações e tratamentos para o registro desses pacotes:

- **Novo pacote –** O pacote deve ser registrado seguindo as instruções padrões da plataforma e será tratado como registro original.
- **Pacote inalterado na camada intelectual e lógica –** Para o caso do administrador desejar vincular uma informação ao registro sem alteração do pacote. Para isso, um novo registro deve ser feito utilizando o mesmo identificador original e nesse registro o administrador pode inserir as informações desejadas e esse texto passa a ser parte do registro vinculado ao pacote.
- **Pacote inalterado na camada intelectual e alterado na camada lógica –** O registro original deve ser marcado como substituído, um novo registro deve ser feito utilizando o mesmo identificador original e as informações referenciadoras desse novo registro devem ser indicadas no registro original, para possibilitar a auditoria desse último.
- **Exclusão de pacote –** O registro original deve ser marcado como substituído, um novo registro deve ser feito utilizando o mesmo identificador original e as informações referenciadoras desse novo registro devem ser indicadas no registro original, para possibilitar a auditoria desse último. Nesse caso, o novo registro será um tipo de texto, onde o administrador deve documentar sua ação e esse texto passa a ser (no lugar do pacote) parte do registro vinculado a ele.
- **Pacote alterado na camada intelectual e lógica (novo pacote substituto) e exclusão de pacote substituído –** O registro original deve ser marcado como substituído, um novo registro deve ser feito utilizando o mesmo identificador original e as informações referenciadoras desse novo registro devem ser indicadas no registro original, para possibilitar a auditoria desse último. Nesse caso, o novo registro será um tipo de texto,

onde o administrador deve documentar sua ação e esse texto passa a ser (no lugar do pacote) parte do registro vinculado a ele.

- **Pacote alterado na camada intelectual e lógica (novo pacote substituto) e manutenção do pacote substituído para referência –** Um novo registro deve ser feito utilizando o mesmo identificador original e esse registro será um tipo de texto, onde o administrador deve documentar sua ação, apontando inclusive, informações referenciadoras do registro desse novo pacote.

O esquema de vincular o Identificador Global do pacote com seu *hash*, permite garantir que aquele registro trata, de fato, do pacote em questão e que não houve alteração em sua referência por uma eventual fraude em que se queira fazer um pacote se passar por outro.

Em nível de página, cada uma carrega o valor puro do *hash* raiz ($hash_p$) vinculado para que sempre e facilmente possa se verificar a fixidez da página antes de qualquer verificação da fixidez dos registros nela (no caso de auditorias parciais).

### 4.3.1 Auditoria de acervos

A essência da funcionalidade da plataforma está na possibilidade de verificar pacotes de informação e atestar a fixidez desses através do auxílio de um valor âncora confiável. Esse valor âncora é alcançado a partir do uso de DLT para registrar imutavelmente um valor *hash* gerado a partir da criação de uma Árvore de Merkle que rastreia a fixidez de *n* pacotes de informação e seus eventos correlacionados em um RDC.

Um RDC que adote a AFA para controle da fixidez de seus objetos custodiados e que tenha seu acervo devidamente registrado na plataforma, pode atestar a fixidez dos objetos ou de todo o acervo do repositório através dos processos de auditoria.

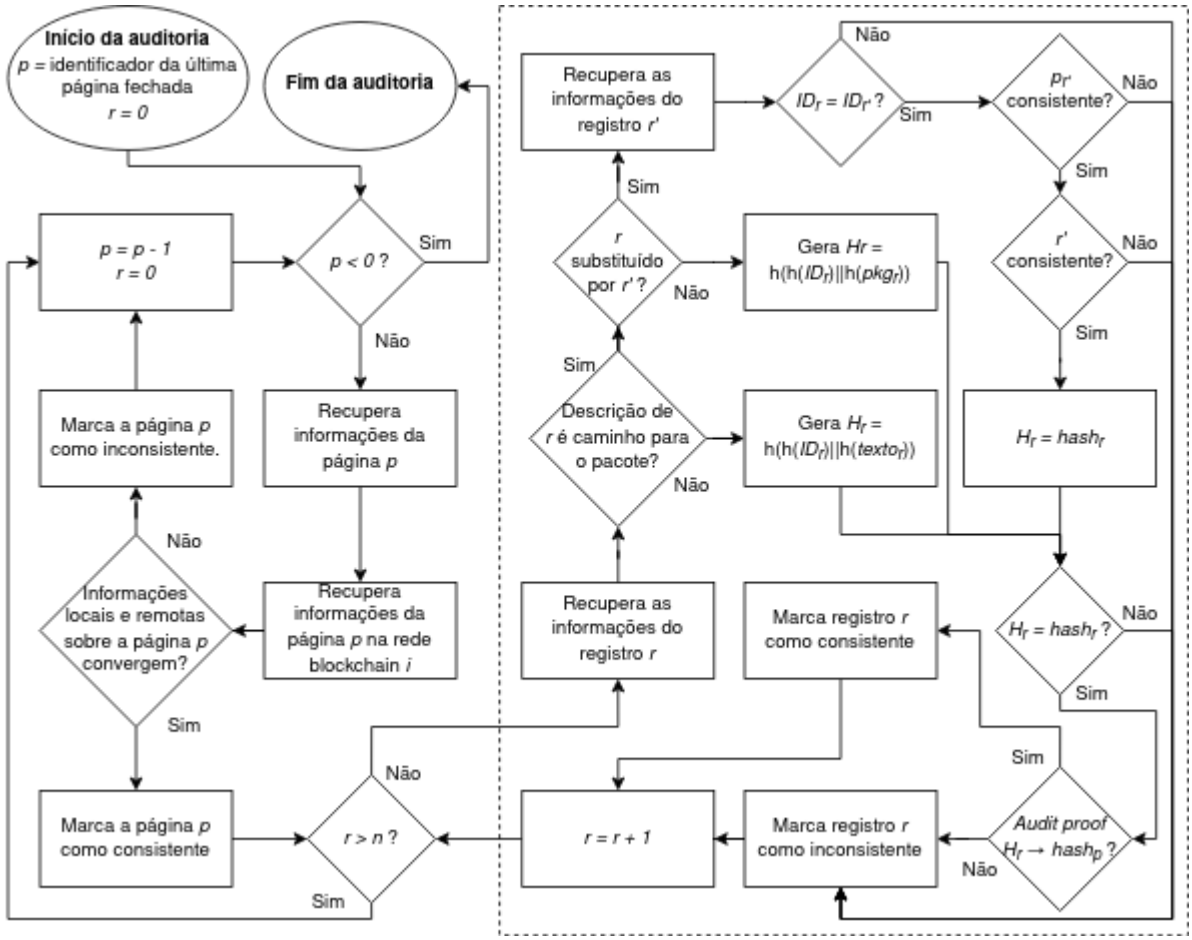
É previsto que a realização das auditorias na plataforma possam ser do tipo completa (de todo o acervo registrado) ou parcial (de páginas individuais). A Figura 4.4 traz um fluxograma de como deve se dar uma auditoria completa na plataforma AFA. Esse processo de auditoria deve ser realizado para cada algoritmo *hash* utilizado no sistema.

Para o entendimento do processo da auditoria ilustrado na Figura 4.4 de *n* registros *r* em uma página *p* utilizando um algoritmo *hash* de função *h()*, deve-se considerar ainda:

- *i* como a *blockchain* utilizada para registro das informações de fixidez com algoritmo *hash* de função *h()*;

- *pkg* como o pacote de informação armazenado em um local no sistema;
- *Audit proof* como o procedimento da verificação de consistência da prova de verificação para *r*, gerada a partir de uma Árvore de Merkle.

Figura 4.4 — Fluxograma do processo de auditoria na plataforma




Fonte: Próprio autor.

Nas auditorias completas, o processo acontece regressivamente, iniciando pela última página fechada do livro (*p*) e caminhando em direção a primeira (*p=0*). Isso se dá devido o caráter progressivo dos registros nas *blockchains* e nesse caso a verificação dos registros mais recentes permite validar os registros antigos que tenham sidos substituídos por uma atualização.

Para essa situação, o processo de auditoria de um registro *r*, que tenha sido marcado como substituído, passará pela verificação da consistência do seu registro substituto *r'*, gerado em um momento $t_{r'} > t_r$ e por isso, caso o registro *r'* esteja marcado como consistente, implica

dizer que esse é legítimo e, portanto, como sua versão mais recente, *r'* acaba por validar a consistência de *r*.

O processo de auditoria parcial pode interessar àqueles momentos onde há uma demanda de confirmar a fixidez de um ou mais registros pontuais naquele momento e a realização de uma auditoria em todo o acervo seria dispensável e mesmo oneroso (em tempo e recursos computacionais), dependendo da extensão do acervo.

No caso de uma auditoria parcial, durante a verificação da página em questão, o processo de verificação de um registro *r* é semelhante ao processo de quando na auditoria completa e difere apenas quando esse estiver marcado como substituído, pois nesse caso, será chamada recursivamente a mesma verificação individual para todas as versões de *r*, até a sua mais recente.

## 4.4 Considerações finais

Acredita-se que a abordagem aqui proposta possa viabilizar aos RDC de qualquer porte um sistema simplificado e potencialmente eficiente e seguro para garantir a fixidez de seus acervos no longo prazo. Além disso, espera-se que a simplicidade e independência prevista para o funcionamento da plataforma possa beneficiar aqueles repositórios que lidem com poucos recursos para sua manutenção, uma vez que a arquitetura da plataforma não deve exigir recursos computacionais adicionais e o registro em *blockchain* utilizando intermediadores, torna seu custo ainda mais irrelevante.

Considerando que a proposta é o registro de uma única evidência de fixidez referente a *n* pacotes no repositório, a preservação das páginas do Livro de Registro AFA se torna imprescindível, pois é necessário o conhecimento das estruturas dos *n* registros para que possam ser realizadas as auditorias de fixidez a partir dos registros nas DLT, uma vez que será necessário o conhecimento da ordem das informações e ordem de registro dos pacotes para correta construção das Árvores de Merkle e consulta em *blockchain*.

Porém, a simplicidade do sistema também facilita sua manutenção, uma vez que todas as páginas estarão dispostas em arquivos de texto puro, que demandam pouco espaço de armazenamento nos sistemas e são facilmente gerenciáveis, possibilitando a criação de cópias de segurança dos registros pela simples compactação das *p* páginas. A partir disso, é possível distribuir essas cópias visando sua redundância e sempre conseguindo garantir que tais cópias eventualmente restauradas estão de fato íntegras, a partir da auditoria dos *hashes* das páginas.

Essa simplicidade arquitetural da plataforma é sua principal característica, em que se prevê a exigência de baixo investimento (implementação simples, poucos dados para controle, pouco uso de poder computacional e possibilidade de utilizar serviços intermediadores para registro em *blockchain*) e praticamente nenhuma mudança nos sistemas de repositório (execução paralela com os sistemas de repositório e uso de *blockchains* públicas)

A seguir são listadas características que se destacam na proposta da plataforma e que, combinadas, permitem diferenciar essa proposta de outras que compartilham a mesma preocupação de garantir a fixidez dos acervos nos repositórios digitais, a exemplo de propostas trazidas em alguns dos trabalhos documentados no Capítulo 3:

- **Arquitetura simplificada –** Uso de apenas dois módulos com funções básicas.
- **Informações de controle independentes –** Uso de estruturas de dados simplificadas e passíveis de gerenciamento fora da plataforma.
- **Flexibilidade às mudanças dos AIP –** Permite as mudanças passíveis nos AIP sem perder sua funcionalidade.
- **Garantia da fixidez das informações locais –** Ainda que as informações referentes a plataforma sejam facilmente manipuláveis, qualquer interferência que vise forçar a autenticidade de uma informação será tão detectada quão forte for a segurança dos algoritmos de *hash*.
- **Redundância de algoritmos *hash*** – Uso de múltiplos algoritmos *hash* nas funções da plataforma visando minimizar potenciais riscos de segurança no caso de um dos algoritmos em uso se tornar vulnerável.
- **Forte publicidade e segurança dos valores âncoras** – Uso de *blockchains* públicas e consolidadas para registro dos valores âncoras.
- **Redundância de redes *blockchain*** – Uso de múltiplas redes públicas de *blockchain* para registro dos valores âncoras visando minimizar potenciais riscos de segurança no caso de uma das redes em uso se tornar vulnerável.
- **Pequenos valores âncoras para grandes dados** – Uso do conceito de Árvores de Merkle para viabilizar um valor pequeno que rastreie a fixidez de inúmeros pacotes de informação e, com isso, permitir poucos registros nas redes ainda que sejam relacionados a muitos pacotes.
- **Compartilhamento de transações em *blockchain* para redução de custos** – Uso previsto de serviços intermediadores de registro em *blockchain*, que possibilitam o registro de dados em *blockchains* públicas com custo reduzido ou inexistente através do compartilhamento de transações com outros dados.

Todavia, ainda que seja previsto, nesse momento ainda não foram definidas estratégias para tratar as mudanças eventualmente necessárias de algoritmos de *hash* e redes *blockchains* em uso. Assim como também não está sendo considerado a utilização dessa plataforma para auditoria em sistemas de *backup* e redundância, mas pela simplicidade da arquitetura, se espera que a adequação da plataforma a esse uso seja facilmente alcançada.

# 5 PROVA DE CONCEITO

Para demonstração da aplicabilidade da proposta, uma prova de conceito foi elaborada implementando os principais mecanismos para apoiar os conceitos já apresentados: permitindo registrar manualmente os pacotes de informação de um acervo com seus identificadores associados, assim como os eventos relacionados às mudanças nesses pacotes e viabilizando a auditoria da fixidez desses pacotes a partir da consulta de valores confiáveis de referência ancorados em DLT.

## 5.1 Implementação

Para a implementação dos módulos da arquitetura proposta da AFA foram desenvolvidos dois *scripts* na linguagem de programação Python (*backend*) e um terceiro *script* que se divide em trechos da mesma linguagem e se relaciona com outros escritos em HTML (*frontend*) para proporcionar uma interface *web* que permite a interação do administrador do arquivo com a plataforma.

Os dois primeiros *scripts* (*afa_dlt.py* e *afa_log.py*) implementam as funcionalidades dos módulos AFA-DLT e AFA-LOG propostos pela arquitetura da plataforma, enquanto o terceiro *script* (*afa_index.py*) surge para viabilizar uma interação com o AFA-LOG de forma amigável ao administrador do arquivo. Os *scripts* desenvolvidos somam pouco mais de 600 linhas de códigos com comentários e estão disponíveis em um repositório público de códigos[35].

Pela existência de experiência prévia a linguagem de programação Python foi a escolhida para a implementação da solução, uma vez que ainda há nessa linguagem relevante acessibilidade e popularidade, dispondo de bibliotecas (nativas ou não) que auxiliaram a implementação de funções essenciais a proposta, a exemplo do gerenciamento de Árvores de Merkle e transações em *blockchain*.

Para essa versão implementada da proposta ainda não há redundância e nem a possibilidade da escolha dos algoritmos de *hash* e das redes *blockchains*, devendo a rede escolhida para utilização ser definida estaticamente a partir de ajustes para tal no AFA-DLT.

35 https://github.com/filipy65/ufpb-ppgi

Nesse contexto, foi definida a versão de testes da rede Bitcoin (testnet), escolhida pela sua credibilidade, ausência de custos[36] e simplicidade de uso.

O armazenamento de dados na rede Bitcoin se dá pelo uso de uma instrução especial nas transações, chamada de OP_RETURN[37], que de acordo com as análises de Sward et al. (2018), é o modo mais eficiente e apropriado para armazenamento de pequenas quantidades de dados nessa rede, onde é possível o armazenamento de até 80 bytes de dados.

A execução das transações varia de acordo com a implementação da solução, mas essencialmente é necessário um endereço de carteira válido na rede e que haja fundos disponíveis nessa carteira para pagamento da taxa de processamento da transação[38].

O limite de 80 bytes para armazenamento de dados nessa rede se mostra mais do que o suficiente para armazenar um valor *hash* SHA-256 (32 bytes.), definido como o único algoritmo *hash* disponível nas operações da plataforma nesse momento e que gera uma cadeia de 64 caracteres hexadecimais. A popularidade desse algoritmo reflete seu custo-benefício, que alia robustez e eficiência, sendo um algoritmo amplamente difundido e bem suportado e, por isso, escolhido como algoritmo *hash* primário nessa solução.

Já para o armazenamento local dos dados de controle da plataforma, o JSON foi o formato escolhido para a estruturação desses dados, escolha também guiada por sua reconhecida acessibilidade e popularidade.

Dentre esses dados de controle estão os documentos de páginas (*p.json*, onde *p* se refere ao número da página), organizados de acordo com o padrão sugerido na arquitetura da plataforma e um documento de estado do sistema (*status.json*) que armazena algumas informações úteis à eficiência de execução da plataforma, como qual é a página aberta para novos registros e quais páginas apresentaram inconsistência nos dados com base no último processo de auditoria[39].

Já considerando o uso da rede *blockchain* de testes da Bitcoin (Bitcoin testnet), ainda são listados a seguir os elementos de *software* cujo uso apoiou a condução da implementação dessa solução ao estágio atual de desenvolvimento e que, portanto, são considerados requisitos para a execução e funcionamento da plataforma:

---

36 Apesar do custo de transação ser inerente ao protocolo, a versão testnet da Bitcoin funciona de forma semelhante a sua vertente oficial, mas sem valor real. É destinada a interessados em testar o protocolo sem custos reais e para isso existem portais (*faucets*) que disponibilizam recursos gratuitamente para uso exclusivo nessa rede.

37 Uma das instruções especiais disponíveis na linguagem de *script* da Bitcoin.

38 https://en.bitcoin.it/wiki/Miner_fees

39 Nesse primeiro momento esses documentos estão configurados para serem armazenados em um único diretório (*afa_pages*), no entanto, se espera que em futuras versões, já para ambientes de produção, os documentos de páginas sejam distribuídas em vários subdiretórios sob critérios ainda não definidos.

- Interpretador Python (>= 3.6) e bibliotecas:
  - cgi – Permite a interação do administrador do arquivo com a plataforma através de métodos HTTP.
  - datetime – Permite manipular formatos de data e hora.
  - hashlib – Viabiliza as operações com *hash*.
  - json – Possibilita conversão de e para documentos no formato de dados JSON.
  - time – Gera intervalos de tempo ao consultar informações sobre transações (e evita que as requisições sejam identificadas como maliciosas — *spam*).
  - urllib – Gerencia requisições de URL por HTTP.
  - bitcoinlib[40] (>= 0.5) – Gerencia carteiras e transações na rede Bitcoin.
  - merkletools[41] – Viabiliza a geração de Árvores de Merkle, assim como a verificação a partir das provas de verificação.

De acordo com esses requisitos a solução pode ser considerada multiplataforma (a depender da disponibilidade do interpretador Python) e suas poucas diretrizes a serem ajustadas devem ser definidas manualmente com a mudança de valores iniciais de variáveis nos *scripts* referentes aos módulos da plataforma, a saber: o nome da carteira e rede Bitcoin a ser usada no módulo AFA-DLT e diretório raiz para armazenamento das páginas no AFA-LOG.

## 5.2 Configuração e execução

Para a execução da solução o administrador do servidor do arquivo deve posicionar os *scripts* da plataforma (conteúdo do *afa_sources*) no devido espaço para a execução de *scripts* Python e deve renomear o *script* da interface de gerenciamento (*afa_index.py*) ou adaptar as configurações de seu servidor de acordo com suas preferências de acesso ao serviço.

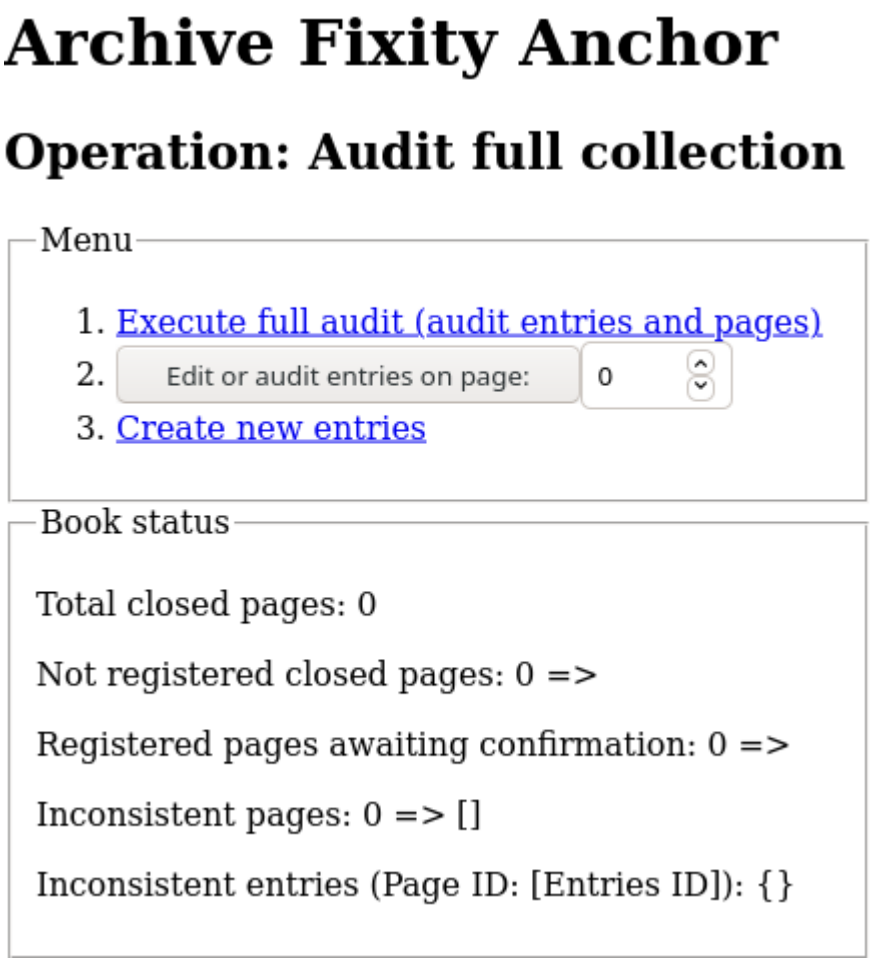
Como já adiantado, o estágio atual da solução apenas prevê o registro e consulta na rede *blockchain* Bitcoin e, por padrão, o módulo de registro e consulta em DLT é previamente configurado para criar uma carteira Bitcoin na rede de testes, que ao ser invocado para seu primeiro registro, irá informar o endereço da carteira que deverá receber fundos para efetivar os registros pelo módulo.

40 http://github.com/1200wd/bitcoinlib
41 https://github.com/Tierion/pymerkletools

O módulo de registro local e auditoria é configurado para gerenciar as páginas do Livro de Registro AFA em um diretório irmão por padrão (*afa_pages*), mas que deve ser ajustado de acordo com as políticas da infraestrutura local e de parâmetros do servidor[39] (e permitir acesso de leitura e escrita para o *afa_log.py*).

Da mesma forma, devem ser previstos os ajustes para o devido acesso do módulo aos pacotes que deverão ser cadastrados no sistema e monitorados por ele. Um diretório irmão (*afa_data*) acompanha os *scripts* para servir de exemplo, podendo o administrador do servidor utilizá-lo como um atalho para o diretório real dos pacotes (facilitando o processo de registro, ainda manual). Pois, ainda deve se considerar que nessa versão, apenas o armazenamento local é suportado, ou seja, a plataforma deve estar sendo executada sob o mesmo sistema que os pacotes de informação a serem preservados e esses devem estar acessíveis para leitura pelo AFA-LOG.

Figura 5.1 — Interface inicial da plataforma Archive Fixity Anchor




Fonte: Próprio autor.

A Figura 5.1 representa a interface inicial da AFA a qual o usuário da plataforma[42] deve se deparar em seu primeiro acesso. Nessa interface está contido um menu de opções e uma área de informações sobre o estado do livro de registros. Ambos os itens estão presentes em todas as interfaces da plataforma e são os únicos a aparecer no caso de uma execução inicial (sem ações definidas) ou quando há alguma ação definida com parâmetros ou resultados inválidos (nesse caso, uma mensagem adicional informativa sobre o problema é apresentada).

A área “Menu” apresenta as três únicas ações disponíveis na plataforma:

1. Executar uma auditoria completa do acervo registrado (todas as páginas do livro).
2. Editar registros (apontar registros substitutos) ou auditar páginas individuais (auditoria parcial).
3. Criar novos registros (registrar novos pacotes ou eventos relacionados).

Na área de informações do livro (book status)[43] é apresentada uma estatística sobre os registros na plataforma:

- **Total closed pages** – Contabiliza todas as páginas fechadas na plataforma.
- **Not registered closed pages** – Contabiliza e aponta todas as páginas fechadas na plataforma que ainda não tiveram seu *hash* registrado em DLT.
- **Registered pages awaiting confirmation** – Contabiliza e aponta todas as páginas fechadas na plataforma que ainda não tiveram o registro de seu *hash* em DLT confirmado (aguardando reforço do encadeamento do bloco onde se encontra a transação).
- **Inconsistent pages** – Contabiliza e aponta as páginas auditadas que apresentaram inconsistência em sua fixidez.
- **Inconsistent entries** – Aponta os registros auditados (e suas respectivas páginas) que apresentaram inconsistência em sua fixidez.

Pode ser percebido pelas informações do livro listadas que nessa implementação o fechamento da página não implica automaticamente no registro dessa em DLT. O fechamento

42 Ressalta-se que a partir desse momento o administrador do arquivo será referenciado apenas como “usuário”, uma vez que essa passa a ser sua posição de relação com a plataforma.

43 Para não necessitar calcular e gerar as informações sempre que necessário, as informações do livro são sempre salvas no documento *status.json*, no diretório *afa_pages*, que serve para controle e é criado por padrão na primeira execução da plataforma.

da página, como proposto na arquitetura da solução (Seção 4.2.1, página 51) nessa versão da implementação é dividida em três fases:

1. Fechamento da página para novos registros – A página não permite novos registros e ganha seu *hash* raiz da Árvore de Merkle, criada a partir dos registros na página.
2. Registro em DLT da página fechada – A página fechada fica aguardando o usuário solicitar o registro em DLT e continuará disponível para esse registro em caso de algum erro na requisição anterior de registro.
3. Confirmação do registro em DLT – A transação referente ao registro da página ainda não obteve a quantidade de confirmações necessárias para ser considerada imutável. A plataforma se encarrega de atualizar o estado de confirmação dessas transações, mas de toda forma, as páginas listadas nessa fase também ficam passíveis de sofrerem um novo pedido de registro pelo usuário, uma vez que a transação relacionada pode ter sofrida algum impedimento a sua aceitação, difusão ou confirmação.

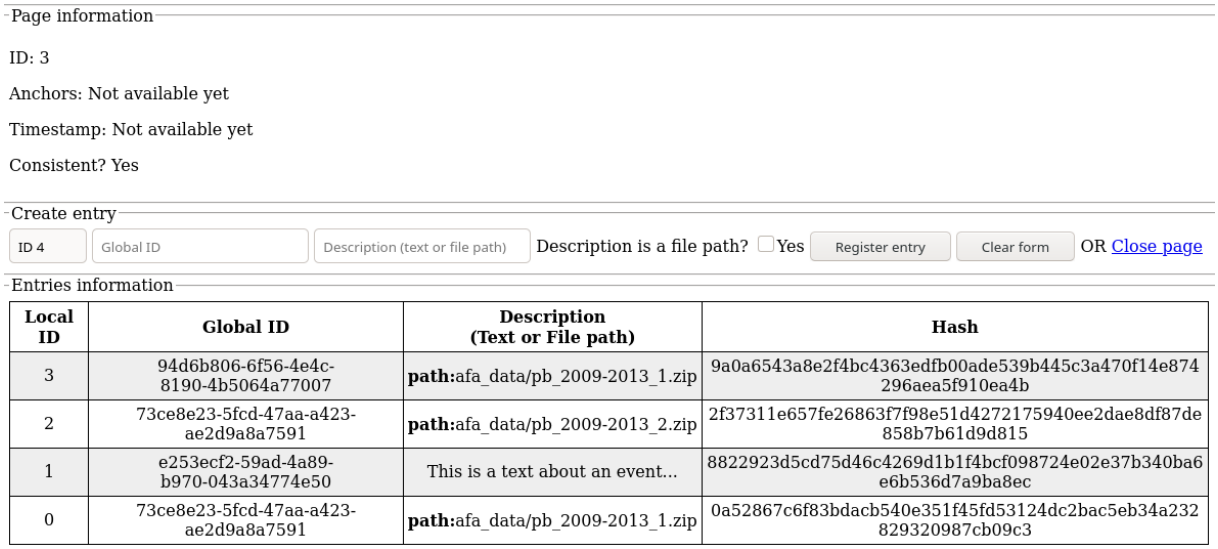
Todas as ações referentes ao fechamento e registro das páginas são facilmente acessíveis a partir de links que ficam disponíveis ao usuário na área de informações sobre o estado do livro e na área de criação de registros (para o fechamento inicial da página), como será apresentado na próxima subseção.

### 5.2.1 Criando novos registros

Ao optar por criar novos registros, o usuário é encaminhado a uma interface dedicada a essa ação, onde poucos dados são necessários para o registro, demandando apenas o Identificador Global referente ao pacote (gerado pelo *software* de repositório) e um texto, que descreve um evento relacionado ao pacote ou descreve o caminho para acesso ao pacote no sistema.

A Figura 5.2 apresenta áreas adicionais que a interface para a criação de novos registros da plataforma oferece (além da área de menu e de informações sobre o estado do livro, apresentados anteriormente e omitidos na Figura). Nessas áreas estão contidas as informações sobre a página atual e os registros já realizados e os campos para preenchimento necessários a um novo registro.

Figura 5.2 — Interface para a criação de novos registros



Page information

ID: 3

Anchors: Not available yet

Timestamp: Not available yet

Consistent? Yes

Create entry

ID 4 | Global ID | Description (text or file path) | Description is a file path? Yes | Register entry | Clear form | OR Close page

Entries information

| Local ID | Global ID | Description (Text or File path) | Hash |
|---|---|---|---|
| 3 | 94d6b806-6f56-4e4c-8190-4b5064a77007 | **path:**afa_data/pb_2009-2013_1.zip | 9a0a6543a8e2f4bc4363edfb00ade539b445c3a470f14e874296aea5f910ea4b |
| 2 | 73ce8e23-5fcd-47aa-a423-ae2d9a8a7591 | **path:**afa_data/pb_2009-2013_2.zip | 2f37311e657fe26863f7f98e51d4272175940ee2dae8df87de858b7b61d9d815 |
| 1 | e253ecf2-59ad-4a89-b970-043a34774e50 | This is a text about an event... | 8822923d5cd75d46c4269d1b1f4bcf098724e02e37b340ba6e6b536d7a9ba8ec |
| 0 | 73ce8e23-5fcd-47aa-a423-ae2d9a8a7591 | **path:**afa_data/pb_2009-2013_1.zip | 0a52867c6f83bdacb540e351f45fd53124dc2bac5eb34a232829320987cb09c3 |

Fonte: Próprio autor.

Uma vez que a página ainda se encontra em construção, além de seu identificador (ID), a mesma ainda não dispõe de dados relevantes para a exibição, o que só ocorrerá após seu fechamento e confirmação de registro em DLT[44].

Na área para criação de novos registros (create entry) estão dispostos os seguintes campos:

- **Local ID** – Um número inteiro (iniciando em zero) sequencial e único na página que identifica o registro na plataforma junto ao número da página. É criado automaticamente e não é permitido sua alteração pelo usuário.
- **Global ID** – Um campo de texto para a inserção de uma identificação única atribuída pelo sistema de repositório ao pacote em questão. Identifica o pacote no sistema de repositório e é uma das informações necessárias a formação do *hash* referente ao pacote em questão.
- **Description (text or file path)** – Um campo de texto para inserir uma referência ao pacote, pela descrição do caminho no sistema para obtê-lo (*file path*) ou um texto que justifique um evento referente a um registro anterior correspondente ao mesmo Identificador Global em questão. Quando o texto for utilizado, o *hash* desse texto será

44 Sempre que não houver uma página aberta (no caso de uma primeira execução), ao optar por criar novos registros um documento *p.json* é criado no diretório *afa_pages*, onde p é o número identificador da página (que nesse caso é 0).

usado para a criação do *hash* correspondente ao respectivo registro, do contrário, o *hash* do próprio pacote será utilizado para isso.

- **Description is a file path?** – Se marcado (Yes), instrui a plataforma a utilizar o valor inserido no campo anterior como um caminho no sistema para acessar o pacote e considerar esse parâmetro para tratar esse registro adequadamente (gerando *hash* para o pacote ou para o texto).
- **Register entry / Clear form** – Botões de ação para efetivar o registro utilizando os dados informados ou para limpar os campos de informações.
- **Close page** – *Link* que conduz a plataforma ao fechamento da página em questão de acordo com os procedimentos já apresentados, levando a página para estágios de fechamento apresentados ainda nesta seção.

Na área de informações dos registros (entries information) são listados em ordem decrescente os registros já realizados na página atual, trazendo além do Local ID, Global ID e Description, o seu valor *hash*, gerado a partir do *hash* da concatenação do valor *hash* do Identificador Global (Global ID) e do valor *hash* da Descrição do registro (Description).

Ao selecionar a opção por fechar a página (*link* “Close page”), o usuário é redirecionado a mesma interface de criar novos registros, mas agora será exibida uma mensagem relatando o sucesso nesse primeiro estágio de fechamento da página e o usuário poderá perceber que tal página já se mostra contabilizada no campo “not registered closed pages”.

O próximo passo para dar prosseguimento ao processo de registro de uma página é selecionar seu ID (como um *link*), exposto no campo “not registered closed pages”, que requisitará o registro dela em uma *blockchain*. Após isso, caso não haja fundos suficientes a plataforma alertará o usuário sobre essa situação e informará a rede e um endereço da carteira nessa rede no qual deve ser depositado um valor para suprir o custo da transação[45]. Caso contrário, o usuário é notificado sobre a efetivação de registro da página e o ID dessa é migrado do campo “not registered closed pages” para “registered pages awaiting confirmation”, onde permanecerá até que a plataforma receba a sua confirmação pela rede[46] e

45 A taxa para registrar transações na rede varia ao longo do tempo, mas nessa implementação abordada (utilizando a rede de testes) uma quantia mínima de 100μBTC foi definida estaticamente como montante mínimo necessário para efetivar as transações, visando oferecer uma certa margem para avaliar a ferramenta, pois nesse período de testes essa taxa girou em torno de 6,5μBTC.

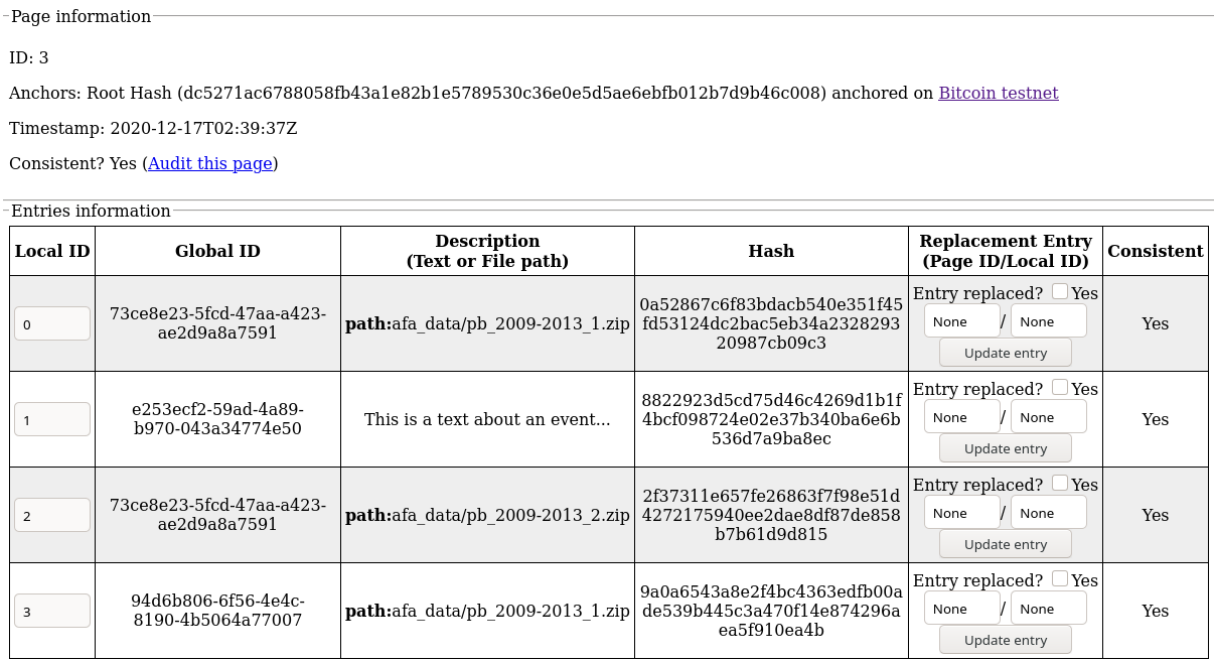
46 A confirmação nessa rede se dá quando o bloco que alocou a transação é sucedido por mais de 6 blocos subsequentes sem que ocorra divergências de blocos.

notifique o usuário de que o registro da referida página foi confirmado, encerrando o ciclo de fechamento da página e a tornando passível de auditoria.

### 5.2.2 Editando registros

A partir da opção no menu de editar ou auditar registros em uma página (edit or audit entries), o usuário obterá as informações e registros de uma página, que já deve estar fechada e registrada, o que lhe permitirá auditar os registros dessa página, assim como adicionar dados de registros substitutos àqueles registros de pacotes que tenham sofrido alguma alteração e que, por isso, precisam agora referenciar sua versão atualizada ou a descrição de um evento relacionado a si. Essa interface para edição ou auditoria de registros em uma página está representada na Figura 5.3.

Figura 5.3 — Interface para editar ou auditar registros em uma página




Fonte: Próprio autor.

Diferente da criação de registros, nessa fase já é possível consultar informações sobre o registro da página e sua fixidez através dos seguintes campos na área de informações da página (Page information):

- **ID** – Um número inteiro (iniciando em zero) sequencial e único na plataforma para identificar a página. É utilizado para referenciar a página e registros substitutos dentro da plataforma.
- **Anchors** – Valor *hash* raiz da Árvore de Merkle gerada a partir dos registros da página (*hashes* dos registros) e um *link* para consultar esse valor ancorado na *blockchain* (através do nome da rede e o identificador da transação nela)[47].
- **Timestamp** – Um carimbo de tempo que marca o momento em que a transação com o valor âncora foi aceita pela rede (a partir da mineração do bloco que a continha).
- **Consistent?** – Marcação da fixidez da página a partir da realização da última auditoria. Antes de passar por uma primeira auditoria todas as páginas são consideradas consistentes por padrão. Nesse campo ainda é dado uma opção ao usuário para requisitar a auditoria da página (auditoria parcial) através do *link* “Audit this page”.

Na área de informação dos registros (Entries information), além dos campos já previamente apresentados na interface de criação de novos registros (Local ID, Global ID, Description e Hash), estão contidos os seguintes campos:

- **Replacement Entry (Page ID/Local ID)** – Permite a marcação de que o respectivo registro deve ser considerado substituído por um registro atualizado, o qual é referenciado por um determinado identificador (Local ID) em uma determinada página (Page ID). No caso de atualização do registro, os respectivos campos devem ser preenchidos e em seguida deve ser acionado o botão “Update entry” para gravar as informações na página.
- **Consistent** – Marcação da fixidez do registro a partir da realização da última auditoria. Antes de passar por uma primeira auditoria todos os registros são considerados consistentes por padrão.

### 5.2.3 Auditando registros

Nessa implementação da plataforma é possível optar pelo processo de auditoria total (todas as páginas) ou parcial (páginas individuais). A requisição de uma auditoria total pode

47 Nessa implementação é utilizado o serviço de exploração de blocos da rede de testes da Bitcoin Blockstream (https://blockstream.info/testnet/).

ser feita a qualquer momento a partir do *link* “Execute full audit (audit entries and pages)” disponível na área de menu que acompanha todas as interfaces da plataforma. Enquanto a requisição de uma auditoria parcial deve ser solicitada a partir da interface de edição da página alvo (conforme apresentado na Seção 5.2.2) através do link “Audit this page”.

O processo de auditoria implementado segue o raciocínio exposto em 4.3.1 (página 59), só funcionará se não houver páginas com registro pendente e tem funcionamento similar em ambos os modos de auditoria, no entanto, como esperado, a auditoria total iniciará pela última página fechada e registrada e só encerrará ao processar a página de ID 0.

Seja qual for o tipo de auditoria, o seu resultado será exposto pela plataforma em mensagem ao usuário, que poderá consultar a qualquer tempo quais páginas e registros foram identificados como inconsistentes no último processo.

Dentre os resultados dos processos de auditoria na plataforma, há um comportamento que merece destaque devido sua excepcionalidade, que ocorre quando ao menos um registro em uma página se monstra inconsistente, mas o *hash* original da página confere com o valor âncora, demonstrando que houve um distúrbio nos elementos relacionado àquele(s) registro(s) (corrupção de pacote ou das informações de fixidez da plataforma), nesse caso a plataforma terá o seguinte comportamento:

- A página não é marcada como inconsistente e assim não será marcada como tal no campo “Inconsistent pages” na área de informação do livro (Book status).
- O(s) registro(s) é(são) marcado(s) como inconsistente(s), sendo apontado(s) como tal no campo “Inconsistent entries” junto com a página da qual faz(em) parte.
- Na interface “edit or audit entries” ambos, registro(s) e página são mostrados como inconsistentes.

A marcação de inconsistência nos registros e páginas persistem por padrão até que o problema interferente seja sanado e um novo processo de auditoria seja executado contemplando a respectiva página ou registro.

## 5.3 Considerações finais

Nessa versão documentada a plataforma ainda apresenta uma interface rudimentar, mas suficientemente usual, pois poucas são as ações disponíveis ao usuário e os rótulos são

autoexplicativos, mas de toda forma, este capítulo visou documentar todas as opções disponíveis, o fluxo de execução das ações na plataforma e o seu funcionamento.

No estágio atual de implementação a plataforma já entrega os recursos necessários ao registro e auditoria de um acervo de pacotes de informação que atenda as premissas da solução previamente apresentadas, permitindo uma prova de conceito e avaliações experimentais iniciais, mas ainda carece de opções que incrementam robustez a proposta, como as opções de redundância de redes *blockchains* e de algoritmos de *hash*. Dessa forma, apesar de possível, ainda não é recomendável o uso da implementação apresentada neste capítulo para ambientes de produção[48].

Além dos recursos de redundância, se deseja que outras melhorias sejam introduzidas na plataforma, adicionando recursos e funcionalidades não implementados neste primeiro momento, como a automatização de registro dos pacotes, a busca de pacotes fora do sistema de execução da plataforma, o tratamento de pacotes de informação não organizados e contidos em recipientes e a avaliação e possíveis adequações para uso da plataforma com pacotes do tipo DIP.

48 Recomenda-se ao leitor que consulte as notas de versão referente ao momento em que estiver consultando este trabalho para conhecer o estágio atualizado da implementação.

# 6 AVALIAÇÃO EXPERIMENTAL

Para avaliar a problemática apresentada neste trabalho e sua respectiva proposta de solução, alguns cenários de avaliação foram executados na tentativa de observar o comportamento desses problemas acompanhados das tratativas trazidas pela solução.

Para a execução desses cenários um ambiente de experimentação foi montado utilizando os elementos mínimos necessários a essa tarefa, resumido no uso da versão implementada da plataforma Archive Fixity Anchor (apresentada no Capítulo 5) e de um *software* de repositório digital, assim como todo o arcabouço tecnológico necessário a correta execução desses elementos.

As subseções seguintes detalham a preparação do ambiente e dos experimentos e documentam os parâmetros utilizados na preparação e execução desses experimentos, assim como seus resultados.

## 6.1 Preparação do ambiente de experimentação

Conforme previamente tratado no curso deste trabalho, a preocupação na garantia de preservar acervos documentais no longo prazo essencialmente passa pela escolha de soluções que atendam aos requisitos de um RDC e consequentemente às premissas da solução proposta (Seção 4.1, página 48).

Para tanto, o Archivematica[49] foi escolhido como o *software* de repositório digital, o qual é gratuito, de código aberto e que conta com uma vasta documentação. Além disso, o Archivematica é considerado por Rodrigues (2015) como uma das opções mais completas e aptas ao gerenciamento de objetos digitais a longo prazo em RDC, sendo o uso desse *software* citado, indicado ou recomendado por instituições de relevância nacional e internacional, a exemplo do Instituto Brasileiro de Informação em Ciência e Tecnologia (IBICT), o Arquivo Nacional, o International Council on Archives (ICA) e a UNESCO (AGUIAR, 2018; LAMPERT, 2016)

Portanto, para montar um ambiente que seguramente apoiasse o Archivematica foi preparado um *hardware* que atendesse seus requisitos mínimos[50] para o ambiente de testes,

49 https://www.archivematica.org/pt-br/
50 https://www.archivematica.org/pt-br/docs/archivematica-1.12/admin-manual/installation-setup/installation/installation/#requirements-small

cuja configuração não foi impactada pelo uso da AFA, uma vez que, como já relatado, se trata de uma solução de baixíssimo custo computacional e que não demanda recursos adicionais além dos já reservados ao *software* de repositório.

Como solução de *hardware* foi montada e preparada uma máquina padrão IBM PC com as seguintes configurações:

- 1x Processador Intel Core i3 530 (2.93GHz)
- 2x Memória DDR3 4GiB
- 1x Disco rígido 500GB

Com essa configuração montada também é possível atender aos requisitos impostos para a instalação do sistema utilizando o esquema de máquina virtual pré-configurada através de uso dos *softwares* VirtualBox[51] e Vagrant[52], sendo essa uma das modalidades de instalação para um ambiente de testes recomendada pela documentação do Archivematica[53] e que nesse contexto se apresentou como a forma mais eficiente de instalação.

Como base para esse método de instalação e uso dos *softwares* necessários a isso optou-se pelo uso do sistema Ubuntu 18.04 64-bit Server Edition, sendo essa a versão mais atual do sistema Ubuntu[54] suportada pela documentação do Archivematica (no momento da execução deste trabalho), a qual oferece a possibilidade de uso de apenas um outro sistema (CentOS[55]). A preferência pelo uso do Ubuntu se deu unicamente pela já existente familiaridade de interação com esse sistema.

Após a preparação do *hardware*, a instalação e configuração dos sistemas e do *software* de repositório seguiram as devidas instruções e configurações padrões, o que resultou em um ambiente de repositório funcional e apto a ingestão de objetos para preservação. Nesse ponto foi, então, colocado a plataforma AFA para ser executada concomitantemente com o *software* de repositório, conforme prevê a arquitetura da solução.

Priorizando um ambiente simplificado e não voltado para uso em produção, a execução da plataforma pode ser feita através da invocação por linha de comando de um módulo de servidor HTTP (com suporte a CGI, necessário a execução dos *scripts*) disponível a partir do próprio interpretador Python[56].

---

51 https://www.virtualbox.org/
52 https://www.vagrantup.com/
53 https://www.archivematica.org/pt-br/docs/archivematica-1.12/getting-started/quick-start/quick-start/#installing-on-vm
54 https://ubuntu.com/
55 https://www.centos.org/
56 https://docs.python.org/3/library/http.server.html

Portanto, a estratégia utilizada para melhor execução da plataforma nesse ambiente foi colocá-la em um caminho padrão de diretórios destinado ao uso exclusivo do Archivematica, referente ao caminho absoluto no sistema */var/archivematica/sharedDirectory/www*, o qual será, daqui em diante, apelidado por <*afa_path*>.

Para seguir a estratégia delineada se fez necessário a criação e adequação das permissões de diretórios para permitir a correta execução da plataforma e a liberação de acesso dessa aos AIP, o que demandou as seguintes ações:

1. Criação dos diretórios *cgi-bin*[57] e *afa_pages* em <*afa_path*> para acomodação dos *scripts* e dos dados de controle da plataforma.
2. Posicionamento dos *scripts* AFA (contidos em *afa_sources*) no diretório <*afa_path*>*/cgi-bin* e a correta adequação de permissões para permitir a execução dos *scripts*.
3. Adequação de permissões do diretório <*afa_path*>*/afa_pages* para permitir a correta leitura e escrita dos dados de controle da AFA.
4. Ajuste de permissões recursivas ao diretório <*afa_path*>*/AIPsStore* para permitir o acesso da AFA a leitura dos AIP, que são criados pelo Archivematica e posicionados por ele em subdiretórios nesse local[58].

Após a preparação do ambiente de acordo com as instruções bastará a execução do comando *"python3 -m http.server --cgi 8888*" a partir do <*afa_path*> para invocar o módulo de servidor HTTP do Python com suporte a CGI escutando na porta 8888[5960]. Isso fará com que a plataforma se torne acessível por um navegador *web* através do endereço *http://*<*server*>*:8888/cgi-bin/afa_index.py*, em que <*server*> se refere ao endereço do servidor em questão.

Por fim, com as diretrizes documentadas aqui foi possível estabelecer um ambiente funcional e adequado aos sistemas necessários a execução da avaliação experimental que será apresentada na sequência, dispondo dos mecanismos mínimos necessários a construção de um repositório digital.

---

57 O módulo de servidor HTTP do Python restringe a execução de *scripts* ao diretório nomeado *cgi-bin*.

58 https://www.archivematica.org/pt-br/docs/archivematica-1.12/user-manual/archival-storage/archival-storage/#stored-aip-structure

59 Acompanha os códigos da AFA um arquivo (computacional) denominado *afa.service*, que pode ser instalado no sistema sugerido para fornecer um serviço (*daemon*) que facilitará a execução automatizada do servidor para a disponibilidade da plataforma.

60 A porta 8888 é sugerida para se manter um endereçamento padrão e evitar conflitos mais comuns, uma vez que o serviço do Archivematica estará escutando na porta 80, que o Archivematica Storage Service estará escutando na porta 8000 e que 8080 é uma porta comumente utilizada como alternativa para serviços *web*.

## 6.2 Execução dos experimentos

Como experimentos foram simulados quatro possíveis cenários considerados passíveis de acontecimento na prática de um repositório digital.

Se avalia que nos três primeiros cenários poderia se levar ao comprometimento da credibilidade do acervo do repositório e nos quais a solução proposta neste trabalho visou atuar a fim de contornar as desconfianças e garantir a crença na autenticidade do repositório.

Um sumário desses cenários é mostrado a seguir e deve-se saber que em todos eles se visou avaliar o comportamento do Archivematica e da plataforma AFA no âmbito da verificação de fixidez:

- **Cenário 1: Adulteração simples de um pacote de informação** – Neste cenário se visou avaliar o comportamento do Archivematica e da AFA frente a simulação de uma possível adulteração de pacotes de informação, que poderia acontecer de forma maliciosa ou ocasional.
- **Cenário 2: Adulteração de um pacote de informação e de suas respectivas informações de fixidez** – Neste cenário se visou avaliar o comportamento do Archivematica e da AFA frente a simulação de uma possível adulteração puramente maliciosa de um pacote de informação, onde um atacante tentaria forjar a autenticidade de um pacote alterado.
- **Cenário 3: Adulteração de um pacote de informação e de suas respectivas informações de fixidez registradas na AFA** – Neste cenário se visou avaliar o comportamento da AFA frente a simulação de uma possível adulteração puramente maliciosa de um pacote de informação e da página da AFA que referencia esse pacote, onde um atacante tentaria forjar a autenticidade de um pacote alterado manipulando os dados da AFA para que essa apoie sua ação.
- **Cenário 4: Edição legítima de um pacote de informação** – Neste cenário se visou avaliar o comportamento do Archivematica e da AFA frente a simulação de uma edição de um pacote de informação, onde o administrador do arquivo ou o sistema do repositório realiza edições legítimas em dados referentes a esse pacote, como a inserção de metadados.

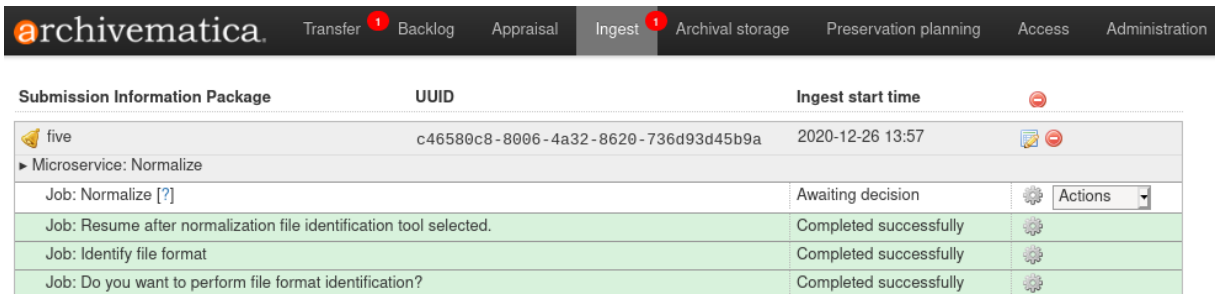
Para a realização dos experimentos deve-se considerar, além da correta execução dos sistemas, a alocação de objetos digitais que são transformados em pacotes e que se tornam protagonistas na simulação dos cenários citados.

Nesse caso foram selecionados 15 documentos de imagem de autoria própria[61], onde cada um deles foi alocado em um diretório numerado (de 1 a 15) e posicionado em um local no sistema no qual o Archivematica considere como fonte para transferência dos objetos digitais, sendo definido por padrão o diretório */home/vagrant* na configuração aqui documentada. Esses documentos foram registrados no Archivematica gradativamente e essas etapas acompanham cada um dos cenários de experimentos relatados nas próximas subseções.

Deve-se entender o registro de objetos no Archivematica como o processo de colocá-los em um AIP pronto para preservação, o que nesse sistema acontece com a transferência dos objetos a serem preservados para o Archivematica, a adequação desses em um SIP e posterior transformação desse SIP em um AIP, que seguirá para armazenamento. Todos esses processos são conduzidos automaticamente pelo Archivematica, que, por padrão, permite o acompanhamento e possíveis intervenções pelo usuário.

Durante esses processos são realizadas ações pertinentes a adequação aos padrões e aos objetivos de preservação. Dentre essas ações se encontram algumas que fornecem opções de personalização de acordo com a demanda do repositório.

Figura 6.1 — Interface de ingestão de um SIP no Archivematica




Fonte: Próprio autor.

A Figura 6.1 demonstra uma interface do Archivematica que mostra o momento em que o processo de transformação de um SIP em AIP demanda a intervenção do usuário para que se defina uma ação para a opção de normatização de formato.

61 Disponíveis em https://github.com/filipy65/ufpb-ppgi/tree/main/afa_data.

Para os cenários de experimentos aqui documentados, além de não adicionar metadados durante esses processos iniciais, essas opções foram definidas com valores recomendados pela documentação do Archivematica ou que pudessem levar a simplificação de seus processamentos.

Logo abaixo são relatadas quais as opções foram oferecidas e quais os respectivos valores definidos durante os processos de registro de objetos no Archivematica para todos os cenários:

- Na aba *Transfer*:
  - Definido o valor “No” para a opção “Do you want to perform file format identification?”.
  - Definido o valor “Skip examine contents” para a opção “Examine contents?”.
  - Definido o valor “Create single SIP and continue processing” para a opção “Create SIP(s)”.
- Na aba *Ingest*:
  - Definido o valor “Do not normalize” para a opção “Normalize”.
  - Definido o valor “No” para a opção “Transcribe SIP contents?”.
  - Definido o valor “Store AIP” para a opção “Store AIP”.
  - Definido o valor “Store AIP in standard Archivematica directory” para a opção “Store AIP location”.

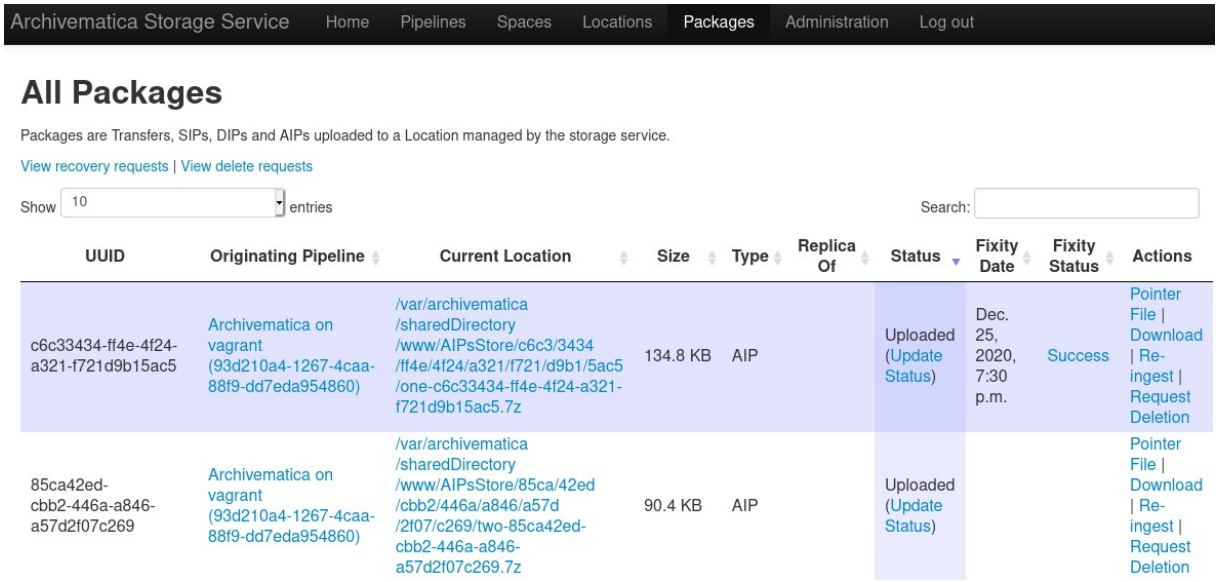
Cada transferência de objetos no Archivematica demanda um nome que comporá o nome do pacote resultante no sistema de arquivos e para facilitar essa referência o nome de cada transferência foi definido de acordo com o número do objeto.

Após os processos de transferência serem finalizados, os pacotes resultantes podem ser consultados pela aba *Archival storage* no Archivematica ou na aba *Packages* do Archivematica Storage Service, onde são apresentadas informações como o identificador único (UUID) e o local atual de armazenamento (caminho no sistema) para se ter acesso ao pacote, informações indispensáveis para o registro do mesmo na AFA, o que já pode ocorrer a partir desse momento. A Figura 6.2 demonstra essa interface do Archivematica Storage Service, que também traz informações sobre a fixidez desses pacotes.

Além dos registros dos objetos digitais, outra ação recorrente nos cenários de experimento é o processo de verificação da fixidez dos pacotes no Archivematica. Por isso se faz necessário destacar, desde já, que o Archivematica dispõe de um mecanismo nativo para

controle de fixidez[62] dos pacotes armazenados. Esse mecanismo acompanha o Archivematica Storage Service e está disponível através de um API *endpoint* que pode ser invocado manualmente através de uma chamada HTTP GET[63] ou por meio de uma aplicação[64], passível de instalação no sistema.

Figura 6.2 — Informações sobre pacotes na interface do Archivematica Storage Service




Fonte: Próprio autor.

Por se tratar de um experimento com poucas amostras, se optou pela invocação manual do mecanismo, que na configuração aqui documentada pode ser realizada a partir da chamada do endereço *http://<server>:8000/api/v2/file/<UUID>/check_fixity/* em um navegador *web*, onde *<UUID>* se refere ao identificador único do pacote. Para facilitar sua referência esse método de verificação será referenciado nos experimentos apenas como *fixity endpoint*.

Mas independente do meio para invocação do mecanismo, as informações sobre a verificação de fixidez são sempre mostradas na aba *Packages* do Archivematica Storage Service (desde que tenha sido realizada uma primeira verificação), onde se apresenta a data e hora da última verificação e o resultado dessa verificação. A Figura 6.2 ilustra essa situação, onde um dos pacotes ainda não passou pela verificação e por isso não apresenta essas informações relacionadas.

62 https://www.archivematica.org/pt-br/docs/storage-service-0.17/fixity
63 https://wiki.archivematica.org/Storage_Service_API#Check_fixity
64 https://github.com/artefactual/fixity

### 6.2.1 Cenário 1: Adulteração simples de um pacote de informação

Este cenário visou avaliar o comportamento do Archivematica e da AFA frente a simulação de uma possível adulteração de pacotes de informação, que poderia acontecer de forma maliciosa ou ocasional.

Para a construção deste cenário foram escolhidos quatro dos objetos digitais disponíveis (1, 2, 3 e 4), os quais foram registrados no Archivematica e seus pacotes resultantes foram respectivamente registrados em uma página 0 na plataforma AFA, cuja página foi devidamente fechada e teve seu registro em DLT confirmado.

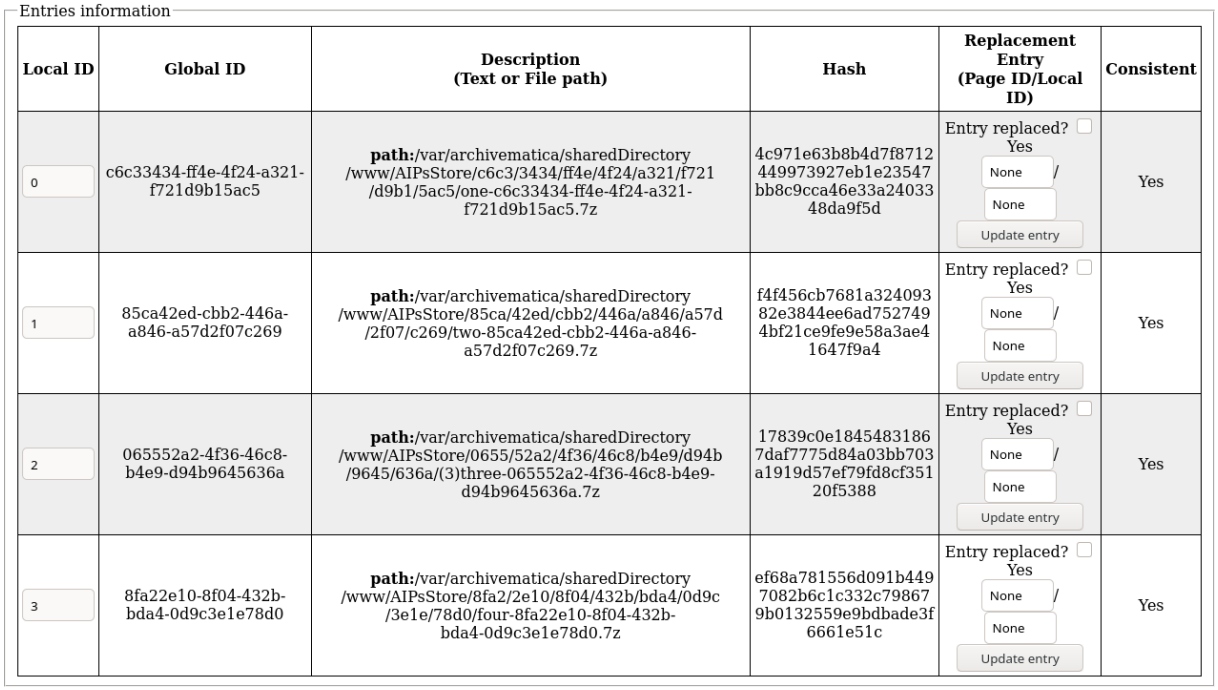
Após o registro, para se obter as primeiras informações de fixidez no Archivematica foi invocado o *fixity endpoint* para cada um dos quatro pacotes registrados, levando ao registro de verificação bem sucedida para todos os pacotes, como demonstra Figura 6.3, sendo esse mesmo resultado positivo replicado na AFA conforme documenta a Figura 6.4.

Figura 6.3 — Informações sobre pacotes do cenário de experimentos 1 e seus respectivos estados iniciais de fixidez no Archivematica Storage Service

| UUID | Originating Pipeline | Current Location | Size | Type | Replica Of | Status | Fixity Date | Fixity Status | Actions |
|---|---|---|---|---|---|---|---|---|---|
| c6c33434-ff4e-4f24-a321-f721d9b15ac5 | Archivematica on vagrant (93d210a4-1267-4caa-88f9-dd7eda954860) | /var/archivematica/sharedDirectory/www/AIPsStore/c6c3/3434/ff4e/4f24/a321/f721/d9b1/5ac5/one-c6c33434-ff4e-4f24-a321-f721d9b15ac5.7z | 134.8 KB | AIP | | Uploaded (Update Status) | Dec. 25, 2020, 8:05 p.m. | Success | Pointer File \| Download \| Re-ingest \| Request Deletion |
| 85ca42ed-cbb2-446a-a846-a57d2f07c269 | Archivematica on vagrant (93d210a4-1267-4caa-88f9-dd7eda954860) | /var/archivematica/sharedDirectory/www/AIPsStore/85ca/42ed/cbb2/446a/a846/a57d/2f07/c269/two-85ca42ed-cbb2-446a-a846-a57d2f07c269.7z | 90.4 KB | AIP | | Uploaded (Update Status) | Dec. 25, 2020, 8:05 p.m. | Success | Pointer File \| Download \| Re-ingest \| Request Deletion |
| 065552a2-4f36-46c8-b4e9-d94b9645636a | Archivematica on vagrant (93d210a4-1267-4caa-88f9-dd7eda954860) | /var/archivematica/sharedDirectory/www/AIPsStore/0655/52a2/4f36/46c8/b4e9/d94b/9645/636a/(3)three-065552a2-4f36-46c8-b4e9-d94b9645636a.7z | 77.0 KB | AIP | | Uploaded (Update Status) | Dec. 25, 2020, 8:05 p.m. | Success | Pointer File \| Download \| Re-ingest \| Request Deletion |
| 8fa22e10-8f04-432b-bda4-0d9c3e1e78d0 | Archivematica on vagrant (93d210a4-1267-4caa-88f9-dd7eda954860) | /var/archivematica/sharedDirectory/www/AIPsStore/8fa2/2e10/8f04/432b/bda4/0d9c/3e1e/78d0/four-8fa22e10-8f04-432b-bda4-0d9c3e1e78d0.7z | 80.2 KB | AIP | | Uploaded (Update Status) | Dec. 25, 2020, 8:05 p.m. | Success | Pointer File \| Download \| Re-ingest \| Request Deletion |

Fonte: Próprio autor.

Figura 6.4 — Informações sobre pacotes do cenário de experimentos 1 e seus respectivos estados iniciais de fixidez na AFA



Entries information

| Local ID | Global ID | Description (Text or File path) | Hash | Replacement Entry (Page ID/Local ID) | Consistent |
|---|---|---|---|---|---|
| 0 | c6c33434-ff4e-4f24-a321-f721d9b15ac5 | **path:**/var/archivematica/sharedDirectory /www/AIPsStore/c6c3/3434/ff4e/4f24/a321/f721 /d9b1/5ac5/one-c6c33434-ff4e-4f24-a321-f721d9b15ac5.7z | 4c971e63b8b4d7f8712449973927eb1e23547bb8c9cca46e33a2403348da9f5d | Entry replaced? Yes None / None Update entry | Yes |
| 1 | 85ca42ed-cbb2-446a-a846-a57d2f07c269 | **path:**/var/archivematica/sharedDirectory /www/AIPsStore/85ca/42ed/cbb2/446a/a846/a57d /2f07/c269/two-85ca42ed-cbb2-446a-a846-a57d2f07c269.7z | f4f456cb7681a32409382e3844ee6ad7527494bf21ce9fe9e58a3ae41647f9a4 | Entry replaced? Yes None / None Update entry | Yes |
| 2 | 065552a2-4f36-46c8-b4e9-d94b9645636a | **path:**/var/archivematica/sharedDirectory /www/AIPsStore/0655/52a2/4f36/46c8/b4e9/d94b /9645/636a/(3)three-065552a2-4f36-46c8-b4e9-d94b9645636a.7z | 17839c0e18454831867daf7775d84a03bb703a1919d57ef79fd8cf35120f5388 | Entry replaced? Yes None / None Update entry | Yes |
| 3 | 8fa22e10-8f04-432b-bda4-0d9c3e1e78d0 | **path:**/var/archivematica/sharedDirectory /www/AIPsStore/8fa2/2e10/8f04/432b/bda4/0d9c /3e1e/78d0/four-8fa22e10-8f04-432b-bda4-0d9c3e1e78d0.7z | ef68a781556d091b4497082b6c1c332c798679b0132559e9bdbade3f6661e51c | Entry replaced? Yes None / None Update entry | Yes |

Fonte: Próprio autor.

Depois das confirmações sobre o estado de fixidez dos pacotes, o pacote de UUID 065552a2-4f36-46c8-b4e9-d94b9645636a teve seu conteúdo intencionalmente modificado, onde o objeto digital alvo de preservação do pacote (nomeado como “pb_2009-2013_3.jpg”) foi substituído por outro objeto de conteúdo arbitrário (nomeado por “pb_2009-2013_13.jpg”), mas apesar da substituição, foi mantido o mesmo nome identificador do objeto original, a fim de tentar passar esse objeto e seu pacote como autênticos e inalterados e simular uma corrupção maliciosa.

Nesse caso, a manipulação do pacote aconteceu externamente ao servidor, com a obtenção do pacote para uma estação de trabalho, manipulação desse utilizando uma aplicação gráfica de compactação e descompactação de arquivos (computacionais)[65] e reenvio desse ao servidor, tendo a cautela de manter as devidas permissões de leitura e escrita do pacote, assim como seu proprietário no sistema de arquivos.

Essa manipulação também possibilita simular uma eventual corrupção que o pacote possa sofrer ocasionalmente e que venha a alterar seu conteúdo, sendo que aqui a alteração do conteúdo foi direcionada e não passou pela alteração das informações de controle do pacote,

65 https://utils.kde.org/projects/ark/

evento que poderia acontecer em cenários de corrupção aleatória, como a documentada no Apêndice A.

Após efetivado a mudança, uma nova checagem de fixidez foi requisitada através da consulta ao *fixity endpoint* e da auditoria de página na AFA para se obter o estado atualizado de fixidez desse pacote, sabendo que esse não corresponde mais ao seu estado original.

Figura 6.5 — Informação do Archivematica sobre falha de fixidez no pacote alvo do experimento do cenário 1

| UUID | Originating Pipeline | Current Location | Size | Type | Replica Of | Status | Fixity Date | Fixity Status |
|---|---|---|---|---|---|---|---|---|
| 065552a2-4f36-46c8-b4e9-d94b9645636a | Archivematica on vagrant (93d210a4-1267-4caa-88f9-dd7eda954860) | /var/archivematica/sharedDirectory/www/AIPsStore/0655/52a2/4f36/46c8/b4e9/d94b/9645/636a/(3)three-065552a2-4f36-46c8-b4e9-d94b9645636a.7z | 77.0 KB | AIP | | Uploaded (Update Status) | Dec. 25, 2020, 8:54 p.m. | Failed |

Fonte: Próprio autor.

Conforme apresentado na Figura 6.5 a ferramenta do Archivematica detectou a inconsistência do pacote e informa sobre o estado de falha da fixidez desse, permitindo obter mais informações sobre o problema ao selecionar o *link* de descrição do estado da fixidez (“Failed”), que levará o usuário a saber que nessa última verificação foi detectado uma alteração no tamanho do pacote, conforme é apresentado na Figura 6.6.

Figura 6.6 — Detalhes do Archivematica sobre a falha de fixidez no pacote alvo do experimento do cenário 1

**Fixity Checks**

Show 10 entries Search:

| Date | Error |
|---|---|
| Dec. 25, 2020, 8:05 p.m. | None |
| Dec. 25, 2020, 8:54 p.m. | Payload-Oxum validation failed. Expected 10 files and 234902 bytes but found 10 files and 423843 bytes |

Showing 1 to 2 of 2 entries Previous Next

Fonte: Próprio autor.

De maneira semelhante, a verificação de fixidez do pacote na AFA (realizada a partir da auditoria da página na qual o pacote está registrado) também retornou uma marca sobre a inconsistência de sua fixidez, conforme traz a Figura 6.7.

Figura 6.7 — Informação da AFA sobre falha da fixidez no pacote alvo do experimento do cenário 1

Entries information

| Local ID | Global ID | Description (Text or File path) | Hash | Replacement Entry (Page ID/Local ID) | Consistent |
|---|---|---|---|---|---|
| 2 | 065552a2-4f36-46c8-b4e9-d94b9645636a | **path:**/var/archivematica/sharedDirectory /www/AIPsStore/0655/52a2/4f36/46c8/b4e9/d94b /9645/636a/(3)three-065552a2-4f36-46c8-b4e9-d94b9645636a.7z | 17839c0e1845483186 7daf7775d84a03bb703 a1919d57ef79fd8cf351 20f5388 | Entry replaced? Yes None / None Update entry | NO |

Fonte: Próprio autor.

Mas diferente do Archivematica, a AFA não fornece detalhes sobre a inconsistência do registro. Isso acontece devido as diferentes formas de implementação do controle da fixidez, que no Archivematica tem como base o uso das especificações do formato Bagit[66] para guiar a formação dos pacotes, o que inclui informações adicionais sobre seu conteúdo.

Entretanto, independente dos detalhes fornecidos, neste cenário, tanto o Archivematica quanto a AFA alertaram corretamente para a falha da fixidez do pacote adulterado, ficando o repositório munido de duas ferramentas que lhe garante a correta detecção de corrupção de um pacote como o simulado nesse cenário.

### 6.2.2 Cenário 2: Adulteração de um pacote de informação e de suas respectivas informações de fixidez

Este cenário visou avaliar o comportamento do Archivematica e da AFA frente a simulação de uma possível adulteração puramente maliciosa de um pacote de informação, onde um atacante tentaria forjar a autenticidade de um pacote alterado.

Para a construção deste cenário foram escolhidos mais quatro dos objetos digitais disponíveis (5, 6, 7 e 8), os quais foram registrados no Archivematica e seus pacotes resultantes foram respectivamente registrados em uma página 1 na plataforma AFA, cuja página foi devidamente fechada e teve seu registro em DLT confirmado.

Após o registro, para se obter as primeiras informações de fixidez no Archivematica foi invocado o *fixity endpoint* para cada um dos quatro pacotes registrados, levando ao registro de verificação bem sucedida para todos os pacotes nos mesmos moldes do que se

66 https://tools.ietf.org/html/rfc8493

demonstrou na Figura 6.3, sendo esse mesmo resultado positivo também replicado na AFA de forma semelhante ao que trouxe a Figura 6.4.

Depois das confirmações sobre o estado de fixidez dos pacotes, o pacote de UUID ed63ce9a-ad27-4d28-8885-c99b1651c3e7 teve seu conteúdo intencionalmente modificado, onde o objeto digital alvo de preservação do pacote (nomeado como “pb_2009-2013_6.jpg”) foi substituído por outro objeto de conteúdo arbitrário (nomeado por “pb_2009-2013_12.jpg”), mas apesar da substituição, foi mantido o mesmo nome identificador do objeto original, a fim de tentar passar esse objeto e seu pacote como autênticos e inalterados. Mas, além disso, nesse cenário o pacote foi reconstruído de acordo com as especificações do Bagit com suas novas informações de fixidez, almejando o reconhecimento como autêntico pelo *fixity endpoint* do Archivematica.

Aqui a manipulação do pacote também aconteceu externamente ao servidor, seguindo as mesmas estratégias descritas no Cenário 1 (6.2.1), mas com o diferencial de utilizar adicionalmente a ferramenta bagit-python[67] para validar a estrutura do pacote com as novas informações de fixidez.

Depois de validado, o pacote foi compactado em um recipiente do tipo 7-zip[68], conforme utilizado pelo Archivematica, e foi reposicionado em seu respectivo diretório no servidor, mantendo as mesmas preocupações em preservar as devidas permissões de leitura e escrita do pacote, assim como seu proprietário.

Após a manipulação do pacote, uma nova checagem de fixidez foi requisitada através da consulta ao *fixity endpoint* e da auditoria de página na AFA para se obter o estado atualizado de fixidez desse pacote, sabendo que esse não corresponde mais ao seu estado original.

Conforme apresenta a Figura 6.8, a ferramenta do Archivematica considerou o pacote em questão como se permanecesse com sua fixidez consistente, ainda que o conteúdo do pacote esteja alterado. Esse falso sucesso na verificação é mantido ao selecionar o *link* de descrição do estado da fixidez e ser conduzido à obtenção de mais informações sobre as verificações executadas, que levará o usuário a crer que a fixidez do pacote continua inabalada, como se apresenta na Figura 6.9.

67 https://github.com/LibraryOfCongress/bagit-python
68 https://www.7-zip.org/7z.html

Figura 6.8 — Informação do Archivematica sobre o falso sucesso de fixidez no pacote alvo do experimento do cenário 2

| UUID | Originating Pipeline | Current Location | Size | Type | Replica Of | Status | Fixity Date | Fixity Status |
|---|---|---|---|---|---|---|---|---|
| ed63ce9a-ad27-4d28-8885-c99b1651c3e7 | Archivematica on vagrant (93d210a4-1267-4caa-88f9-dd7eda954860) | /var/archivematica/sharedDirectory/www/AIPsStore/ed63/ce9a/ad27/4d28/8885/c99b/1651/c3e7/(6)six-ed63ce9a-ad27-4d28-8885-c99b1651c3e7.7z | 93.3 KB | AIP | | Uploaded (Update Status) | Dec. 26, 2020, 12:38 p.m. | Success |

Fonte: Próprio autor.

Figura 6.9 — Detalhes do Archivematica sobre o falso sucesso de fixidez no pacote alvo do experimento do cenário 2


Fonte: Próprio autor.

Por outro lado, a verificação de fixidez do pacote na AFA (realizada a partir da auditoria da página na qual o pacote está registrado) alerta sobre a inconsistência de sua fixidez e seu estado é atualizado e apresentado pela AFA, conforme traz a Figura 6.10.

Figura 6.10 — Informação da AFA sobre falha da fixidez no pacote alvo do experimento do cenário 2

Entries information

| Local ID | Global ID | Description (Text or File path) | Hash | Replacement Entry (Page ID/Local ID) | Consistent |
|---|---|---|---|---|---|
| 1 | ed63ce9a-ad27-4d28-8885-c99b1651c3e7 | **path:**/var/archivematica/sharedDirectory/www/AIPsStore/ed63/ce9a/ad27/4d28/8885/c99b/1651/c3e7/(6)six-ed63ce9a-ad27-4d28-8885-c99b1651c3e7.7z | 19e7d32c405e4dd9d10b7a25dfda723c050dbdc92833da40bb57bed6e5b02a2d | Entry replaced? Yes None / None Update entry | NO |

Fonte: Próprio autor.

Essa divergência de resultados acontece devido as diferenças na implementação do controle da fixidez, como já previamente esclarecido e, nesse caso, mostrou a relevância do uso da AFA para apoiar o controle de fixidez em um repositório, que apenas com a ferramenta de verificação nativa do Archivematica pôde ser facilmente burlado nesse cenário.

### 6.2.3 Cenário 3: Adulteração de um pacote de informação e de suas respectivas informações de fixidez registradas na AFA

Este cenário visou avaliar o comportamento da AFA frente a simulação de uma possível adulteração puramente maliciosa de um pacote de informação e da página da AFA que referencia esse pacote, onde um atacante tentaria forjar a autenticidade de um pacote alterado manipulando os dados da AFA para que essa apoie sua ação.

Para este cenário se optou por utilizar o mesmo conjunto de pacotes já previamente registrados no Cenário 2 (6.2.2) tanto no Archivematica como na AFA, dando continuidade aos processos iniciados nesse cenário anterior e eliminando a necessidade de refazer os mesmos processos a fim de se alcançar o estado no qual se findou o experimento documentado naquele cenário.

Foi considerado então a escolha do mesmo pacote alvo do cenário anterior, onde esse foi devidamente manipulado e se conheceu o comportamento do *fixity endpoint* do Archivematica e da AFA frente a essa manipulação. Mas agora se pretende forçar a situação para que a AFA também apoie a legitimidade desse pacote adulterado.

Conhecendo o funcionamento da plataforma (Capítulo 5), foi possível replicar as ações que ela realiza para estruturar e fechar uma página e dessa forma foi possível manipular adequadamente a página 1 (na qual o pacote alvo tinha sido registrado) para que essa refletisse o registro do pacote adulterado de forma transparente.

Para a manipulação da página bastou estruturar as informações dos pacotes na mesma ordem em que se executaram os registros legítimos, posicionando devidamente os novos valores de prova de verificação e do *hash* raiz.

No entanto, apesar da devida manipulação da página, ao se chamar uma auditoria sobre a mesma para verificar a consistência de sua fixidez, a plataforma alerta para sua inconsistência, como ilustra a Figura 6.11. Essa detecção de inconsistência é sentida pela plataforma devido a divergência entre o valor *hash* da página armazenado localmente e seu valor âncora salvo na rede *blockchain*.

Figura 6.11 — Informação da AFA sobre falha da fixidez na página que registra a fixidez do pacote alvo do experimento do cenário 2


Fonte: Próprio autor.

O valor *hash* local foi manualmente modificado para o mesmo valor já ancorado na rede na tentativa de contornar essa inconsistência entre tais valores, porém, a plataforma detecta que esse valor não condiz com o valor *hash* raiz gerado pela Árvore de Merkle considerando o *hash* do pacote adulterado e continua alertando sobre a inconsistência de fixidez da página.

Dessa forma, foi observado que a AFA resistiu às tentativas de manipulação aqui documentadas e conseguiu informar corretamente ao usuário sobre o estado das informações de fixidez referentes ao AIP em questão.

Entretanto, se considera que uma das formas que um atacante pode ter para tentar contornar esse resultado seria descobrindo uma outra transação na mesma rede *blockchain* que carregue dados que represente o valor âncora necessário para fazer os dados da página parecerem autênticos e com isso ele poderia ter um endereço de transação coerente para substituir na página e finalmente validar os registros adulterados. Porém, essa é uma condição improvável de alcançar, principalmente se for considerado que em uma implementação ideal, a plataforma prevê o uso de redes e algoritmos de *hash* redundantes, o que reduziria ainda mais a probabilidade do atacante coincidir todas essas variáveis a seu favor. Isso tudo sem

considerar a marca temporal das transações, que a plataforma não está considerando em suas análises de fixidez, mas que se vier a ser considerada de alguma forma, se torna mais uma variável a pesar na obstrução do ataque.

Uma outra forma provável de ataque ao sistema e considerada até mais simples de burlar o mecanismo de auditoria seria a de executar o registro regular de uma página com as informações desejadas (derivadas de adulteração) e assim manipular as páginas localmente para que uma possa se passar pela outra. Esse ataque se mostra facilmente viável principalmente quando a página alvo do ataque é relativamente recente, uma vez que a marca temporal pode não ser tão decisiva nesses casos. Porém, nesse último caso pode se pôr em discussão o interesse de ataque a registros recentes e para todos os casos pode se considerar a implementação de recursos que tratem as transações realizadas pela carteira do repositório na rede *blockchain* com fins de registro como referência para as transações registradas localmente, além da possibilidade de se considerar a marca temporal das transações como mais um elemento de controle.

### 6.2.4 Cenário 4: Edição legítima de um pacote de informação

Este cenário visou avaliar o comportamento do Archivematica e da AFA frente a simulação de uma edição de um pacote de informação, onde o administrador do arquivo ou o sistema do repositório realiza edições legítimas em dados referentes a esse pacote, como a inserção de metadados.

Para a construção deste cenário foram escolhidos outros quatro objetos digitais disponíveis (9, 10, 11 e 12), os quais foram registrados no Archivematica e seus pacotes resultantes foram respectivamente registrados em uma página 2 na plataforma AFA, cuja página foi devidamente fechada e teve seu registro em DLT confirmado.

Após o registro, para se obter as primeiras informações de fixidez no Archivematica foi invocado o *fixity endpoint* para cada um dos quatro pacotes registrados, levando ao registro de verificação bem sucedida para todos os pacotes nos mesmos moldes do que se demonstrou na Figura 6.3, sendo esse mesmo resultado positivo também replicado na AFA de forma semelhante ao que trouxe a Figura 6.4.

Depois das confirmações sobre o estado de fixidez dos pacotes, o pacote de UUID 55d4feb9-c2fb-4970-99a8-dd0aa0472e00 foi submetido ao processo de re-ingestão de

metadados, por onde se permite que sejam adicionados valores de metadados ao AIP pela interface do Archivematica, conforme sua documentação[69].

Com o pedido de re-ingestão o pacote é submetido às mesmas ações de ingestão que são oferecidas quando na formação inicial do pacote e nessa hora há a opção de adicionar metadados, a qual foi selecionada e na qual apenas foi adicionado um título para o pacote, sendo esse campo de metadados salvo e dado continuidade a reconstrução do pacote seguindo as mesmas definições para as demais ações já previstas para esse fim.

Essa mudança nos metadados acaba por resultar em um pacote inalterado na camada intelectual e alterado na camada lógica, o que conforme fora previsto no funcionamento da plataforma (Seção 4.3, página 57), exige um novo registro do mesmo pacote a fim de documentar sua mudança nessa camada e que caso não seja feito compromete a confiabilidade das informações da AFA, uma vez que como mostrado na Figura 6.12, o pacote permanecerá marcado como inconsistente, ainda que esse não seja o seu real estado, o que nesse caso é corretamente registrado pelo Archivematica através da chamada ao *fixity endpoint*.

Figura 6.12 — Informação da AFA sobre falha da fixidez no pacote alvo do experimento do cenário 4

Entries information

| Local ID | Global ID | Description (Text or File path) | Hash | Replacement Entry (Page ID/Local ID) | Consistent |
|---|---|---|---|---|---|
| 1 | 55d4feb9-c2fb-4970-99a8-dd0aa0472e00 | **path:**/var/archivematica/sharedDirectory/www/AIPsStore/55d4/feb9/c2fb/4970/99a8/dd0a/a047/2e00/(10)ten-55d4feb9-c2fb-4970-99a8-dd0aa0472e00.7z | bc51c02c1b2783884df7024495e49cdcfb80022ea392c963a66154f2a79a2703 | Entry replaced? Yes None / None Update entry | NO |

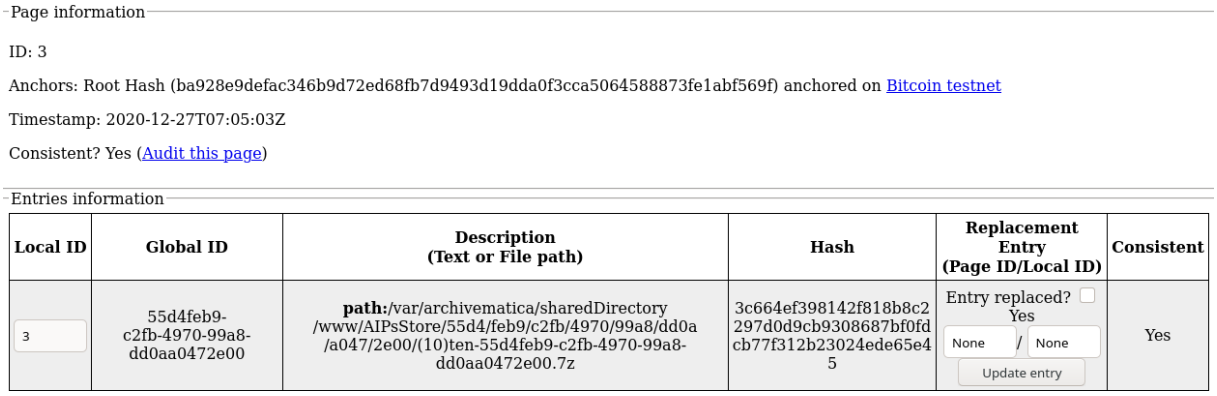
Fonte: Próprio autor.

Para efetivar a atualização do pacote na AFA foi necessário criar um novo registro para o mesmo, o qual ocorreu em uma página 3 que foi devidamente fechada e registrada em DLT, conforme documentado na Figura 6.13.

Após confirmação de sua versão atualizada, o registro inicial do pacote alvo na AFA recebeu uma alteração, onde se definiu (na coluna “Replacement Entry”) que esse registro foi substituído (marcado o valor “Yes” para a opção “Entry replaced?”) e qual o registro que o substituiu (definido o valor “3” para os campos “Page ID” e “Local ID”, conforme relata a Figura 6.13).

69 https://www.archivematica.org/pt-br/docs/archivematica-1.11/user-manual/ingest/ingest/#reingest

Figura 6.13 — Informação da AFA sobre sucesso da fixidez da versão atualizada do pacote alvo do experimento do cenário 4



Page information

ID: 3

Anchors: Root Hash (ba928e9defac346b9d72ed68fb7d9493d19dda0f3cca5064588873fe1abf569f) anchored on Bitcoin testnet

Timestamp: 2020-12-27T07:05:03Z

Consistent? Yes (Audit this page)

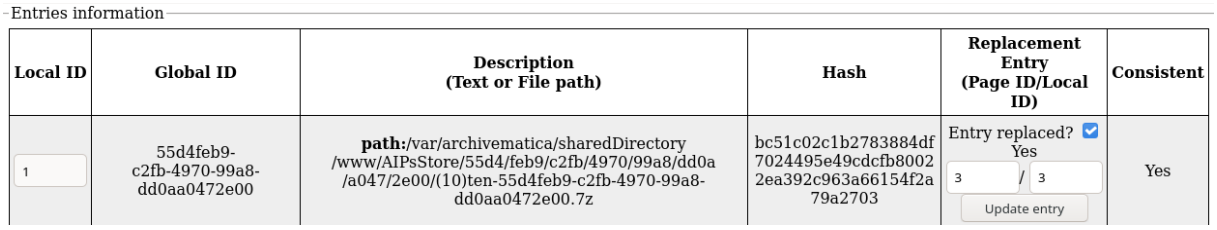
Entries information

| Local ID | Global ID | Description (Text or File path) | Hash | Replacement Entry (Page ID/Local ID) | Consistent |
|---|---|---|---|---|---|
| 3 | 55d4feb9-c2fb-4970-99a8-dd0aa0472e00 | **path:**/var/archivematica/sharedDirectory/www/AIPsStore/55d4/feb9/c2fb/4970/99a8/dd0a/a047/2e00/(10)ten-55d4feb9-c2fb-4970-99a8-dd0aa0472e00.7z | 3c664ef398142f818b8c2297d0d9cb9308687bf0fdcb77f312b23024ede65e45 | Entry replaced? Yes None / None Update entry | Yes |

Fonte: Próprio autor.

Depois do registro atualizado, uma nova auditoria na página 2 do livro da AFA foi requisitada e dessa vez não indicou inconsistência na fixidez do pacote em questão, pois agora a plataforma conhece que o registro na página 2 não reflete mais a realidade das informações de fixidez daquele pacote e por isso irá buscar em seu substituto essas informações atualizadas, o que nesse caso é traduzido em um resultado positivo conforme pôde ser conferido na Figura 6.13, assim como na Figura 6.14, que demonstra a situação do registro inicial assim como seus dados para substituição do registro.

Figura 6.14 — Informação da AFA sobre sucesso da fixidez no pacote alvo do experimento do cenário 4 após seu registro ser atualizado



Entries information

| Local ID | Global ID | Description (Text or File path) | Hash | Replacement Entry (Page ID/Local ID) | Consistent |
|---|---|---|---|---|---|
| 1 | 55d4feb9-c2fb-4970-99a8-dd0aa0472e00 | **path:**/var/archivematica/sharedDirectory/www/AIPsStore/55d4/feb9/c2fb/4970/99a8/dd0a/a047/2e00/(10)ten-55d4feb9-c2fb-4970-99a8-dd0aa0472e00.7z | bc51c02c1b2783884df7024495e49cdcfb80022ea392c963a66154f2a79a2703 | Entry replaced? Yes 3 / 3 Update entry | Yes |

Fonte: Próprio autor.

## 6.3 Considerações finais

A partir dos experimentos aqui documentados pôde se observar que a plataforma AFA se comportou de acordo com o esperado para sua versão implementada e entregou resultados

considerados satisfatórios para o momento: alertando sobre o estado genuíno de fixidez daqueles pacotes em si registrados e de suas próprias informações locais sobre os registros a partir do uso de um valor âncora confiável em DLT e permitindo a atualização dos registros, acompanhando assim a dinamicidade prevista nos pacotes de informação custodiados em um RDC.

Se pensa que uma das formas de burlar o correto funcionamento da plataforma e não consideradas nos experimentos poderia ser a de adulteração de seu código fonte a fim de direcionar seu funcionamento em favor de um ataque. Para esses casos resta a realização de uma auditoria no código fonte da plataforma em execução ou a obtenção, sempre que se considerar pertinente, dos códigos oficiais da plataforma nos casos em que não se trabalha com versões modificadas dessa.

O fato de realização dos experimentos utilizando um número reduzido de amostras se deu pela necessidade de intervenção manual para registro na AFA e a racionalização do tempo desta pesquisa. Apesar de não abordado nesses experimentos, o Archivematica dispõe de mecanismos para automatizar o registro de objetos digitais, mas a indisponibilidade desse tipo de recurso na AFA compromete o registro de grandes acervos (centenas ou milhares de pacotes) e consequentemente seu uso em produção. No entanto, acredita-se no potencial de escalabilidade da plataforma e que o uso com um maior número de registros não deve comprometer seu correto funcionamento e nem os resultados aqui documentados.

# 7 CONSIDERAÇÕES FINAIS

Este trabalho documentou o uso de uma abordagem combinada de Árvores de Merkle e DLT para a criação de valores âncoras confiáveis referentes a conjuntos de objetos digitais preserváveis, para que seja possível, com isso, garantir a fixidez desses objetos no contexto de Repositórios Digitais Confiáveis.

Os Repositórios Digitais Confiáveis precisam demonstrar confiabilidade frente ao seu público e essa demonstração passa pela conformidade com itens normativos definidos por documentos de referência para tal. Aquele Repositório Digital que implementar a proposta aqui apresentada, poderá considerar relatar mais essa prática como mecanismo de demonstração de conformidade com itens normativos que tratem do controle de fixidez dos objetos digitais no repositório, podendo demonstrar ou reforçar sua confiabilidade. No Apêndice B são listados itens previstos nos documentos de referência para auditoria e certificação de RDC nos quais acredita-se que a aplicação da proposta deste trabalho pode ser documentada.

A abordagem apresentada neste trabalho foi modelada em uma plataforma nomeada de Archive Fixity Anchor (AFA), que tem como foco o registro de Archival Information Packages (AIP) gerados por sistemas de *software* de gerenciamento de objetos digitais em conformidade com o modelo OAIS e detalha o fluxo de gerenciamento dos registros e dos processos de auditoria, incluindo situações onde os objetos são alvos das ações de preservação e sofrem mudanças que alteram suas informações de fixidez. A análise de conformidade da proposta para tratamento dos pacotes Dissemination Information Packages (DIP) preserváveis é considerada como uma demanda para trabalhos futuros.

Uma versão da plataforma AFA foi implementada e serviu como prova de conceito da aplicabilidade da proposta e para compor a execução de uma avaliação experimental, que foi realizada em cenários simulados dentro de um ambiente montado de repositório digital.

Como avaliação experimental foram simulados quatro possíveis cenários considerados passíveis de acontecimento na prática de um repositório digital e onde em três deles se poderia levar ao comprometimento da credibilidade do acervo do repositório e nos quais a solução proposta neste trabalho visou atuar a fim de contornar as desconfianças e garantir a crença na autenticidade do repositório.

A partir dos experimentos realizados pôde se observar que a plataforma AFA se comportou como o esperado para sua versão implementada e entregou resultados

considerados satisfatórios para o momento: alertando sobre o estado genuíno de fixidez daqueles pacotes em si registrados e de suas próprias informações locais sobre os registros, a partir do uso de um valor âncora confiável em DLT e permitindo a atualização dos registros, acompanhando a dinamicidade prevista nos pacotes de informação custodiados em um RDC.

Essa versão implementada da plataforma e documentada neste trabalho ainda não é indicada para uso em ambientes de produção, no entanto, se espera que com uma versão completa e devidamente testada e apta para esse uso, essa plataforma possa ser adotada por RDC de diferentes portes devido sua simplicidade para implantação e do que pode entregar diante seu custo-benefício.

A implementação dos recursos de redundância documentados na proposta e não oferecidos nessa primeira versão da implementação da plataforma é considerada, assim como a realização de novos experimentos fazendo o uso desses recursos, como mais uma demanda para desenvolvimento de trabalhos futuros.

Levando em conta as fragilidades do funcionamento da AFA, identificadas e documentadas na seção 6.2.3 (página 89), também é desejável que trabalhos futuros avaliem a possibilidade de considerar as marcas temporais (*timestamp*) das transações em *blockchains* e a ordem de execução dessas pela plataforma com o rastreamento dessas transações como um potencial incremento na confiança dos mecanismos da AFA.

E como mais uma demanda para trabalhos futuros, se elenca a análise de potencial aplicação no âmbito da preservação digital da proposta do InterPlanetary File System – IPFS (BENET, 2014) e a investigação de seu possível uso como potencial alternativa para solução de armazenamento dos dados da AFA.

# REFERÊNCIAS


AGUIAR, F. L. DE. **Dspace e archivematica: concepção e criação de um repositório digital aplicado no domínio da SBPC - sob uma perspectiva interdisciplinar entre arquivística e organização e representação do conhecimento**. [s.l.] Universidade de São Paulo, 2018.

ARQUIVO NACIONAL (BRASIL). **Dicionário Brasileiro de Terminologia Arquivística**. 51. ed. Rio de Janeiro: Arquivo Nacional, 2005.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 15472: Sistemas espaciais de dados e informações - Modelo de referência para um sistema aberto de arquivamento de informação (SAAI)**, 2007.

ATURBAN, M. et al. Archive assisted archival fixity verification framework. **Proceedings of the ACM/IEEE Joint Conference on Digital Libraries**, v. 2019- June, p. 162–171, 2019.

BARBEDO, F. et al. RODA : Repositório de Objectos Digitais Autênticos. **Event London**, p. 38, 2007.

BARROS, D. B. S.; FERRER, I. D.; MAIA, C. M. DE S. Auditoria de repositórios digitais preserváveis. **Revista Ibero-Americana de Ciência da Informação**, v. 11, n. 1, p. 300–313, 2018.

BASHIR, I. **Mastering Blockchain**. Birmingham: Packt Publishing Ltd., 2018.

BAUMANN, R. Soft-Error Mitigation. **IEEE Design & Test of Computers**, v. 22, n. 3, p. 258–266, 2005.

BAYER, D.; HABER, S.; STORNETTA, W. S. Improving the Efficiency and Reliability of Digital Time-Stamping. In: **Sequences II**. New York, NY: Springer New York, 1993. p. 329–334.

BENET, J. IPFS - Content Addressed, Versioned, P2P File System. n. Draft 3, 2014.

BRADEN, R.; BORMAN, D.; PARTRIDGE, C. **Computing the Internet ChecksumRFC 1071**. [s.l: s.n.].

BRALIĆ, V.; KULEŠ, M.; STANČIĆ, H. Model for long-term preservation of digital signature validity: TrustChain. n. November, p. 89–103, 2017.

BUI, T. et al. ARCHANGEL: Tamper-proofing Video Archives using Temporal Content Hashes on the Blockchain. 26 abr. 2019.

CARVALHO, N. **Organizações e Segurança Informática**. Rio Tinto, Portugal: Lugar da Palavra, 2009.

CASTAGNOLI, G.; BRÄUER, S.; HERRMANN, M. Optimization of Cyclic Redundancy-Check Codes with 24 and 32 Parity Bits. **IEEE Transactions on Communications**, v. 41, n. 6, p. 883–892, 1993.

COLLOMOSSE, J. et al. **ARCHANGEL: Trusted Archives of Digital Public Documents**. Proceedings of the ACM Symposium on Document Engineering 2018 - DocEng '18. **Anais**...New York, New York, USA: ACM Press, 2018

CONSELHO NACIONAL DE ARQUIVOS. **Carta para a preservação do patrimônio arquivístico digital**, 2005.

CONSELHO NACIONAL DE ARQUIVOS. **Diretrizes para a Implementação de Repositórios Arquivísticos Digitais Confiáveis - RDC-Arq**Brasil, 2015.

CONSELHO NACIONAL DE ARQUIVOS. **Glossário: Documentos Arquivísticos Digitais**, 2016.

CONSULTATIVE COMITTEE FOR SPACE DATA SYSTEMS. **Audit and Certification of Trustworthy Digital Repositories**. Washington, DC: [s.n.].

CONSULTATIVE COMITTEE FOR SPACE DATA SYSTEMS. **Reference model for an Open Archival Information System (OAIS)**. Washington, DC: [s.n.].

CORETRUSTSEAL. **About – CoreTrustSeal**. Disponível em: <https://www.coretrustseal.org/about/>. Acesso em: 29 out. 2018.

CORETRUSTSEAL. CoreTrustSeal Trustworthy Data Repositories Requirements 2020-2022. 2020.

CORUJO, L. M. N. **Repositórios Digitais e Confiança - Um exemplo de repositório de Preservação Digital : o RODA**. [s.l.] Faculdade de Letras da Universidade de Lisboa, 2014.

CRESPO, A. S. DE P.; GARCÍA, L. I. C. Stampery Blockchain Timestamping Architecture (BTA) - Version 6. p. 1–18, 13 nov. 2017.

CROSBY, M. et al. BlockChain Technology: Beyond Bitcoin. **Applied Innovation Review**, n. 2, p. 6–19, 2016.

DE GIUSTI, M. R.; LUJÁN VILLARREAL, G. Revision of different implementations for digital preservation: towards a methodological proposal for preserving and auditing IR reliability. **RDBCI: Revista Digital de Biblioteconomia e Ciência da Informação**, v. 16, n. 2, p. 273–292, 2018.

DELL TECHNOLOGIES BRAZIL. **Offsite Backup**. Disponível em: <https://www.delltechnologies.com/pt-br/glossary/offsite-backup.htm>. Acesso em: 23 jul. 2020.

DIGITAL CURATION CENTRE; DIGITALPRESERVATIONEUROPE. **Digital Repository Audit Method Based on Risk Assessment**. Glasgow, UK: HATII at the University of Glasgow and Digital Curation Centre (DCC) and DigitalPreservationEurope (DPE), 2007. v. 0

DURANTI, L.; MACNEIL, H. The protection of the integrity of electronic records: An overview of the UBC-MAS research project. **Archivaria**, v. 42, n. 1, p. 46–67, 1996.

FERREIRA, M. **Introdução à preservação digital – Conceitos, estratégias e actuais consensos**. Guimarães, Portugal: Escola de Engenharia da Universidade do Minho, 2006.

GIPP, B.; MEUSCHKE, N.; GERNANDT, A. Decentralized Trusted Timestamping using the Crypto Currency Bitcoin. **iConference 2015 Proceedings**, p. 1–5, 13 fev. 2015.

GONÇALEZ, P. R. V. A. Recomendações para certificação ou medição de confiabilidade de Repositórios Arquivísticos Digitas com ênfase no acesso à informação. **Informação & Informação**, v. 22, n. 1, p. 215, 2017.

GREVE, F. et al. Blockchain e a Revolução do Consenso sob Demanda. **Minicursos do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, p. 52, 2018.

HABER, S.; KAMAT, P. A content integrity service for long-term digital archives. **Archiving 2006 - Final Program and Proceedings**, p. 159–164, 2006.

HABER, S.; SCOTT STORNETTA, W. How to time-stamp a digital document. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 537 LNCS, p. 437–455, 1991.

INSTITUTO BRASILEIRO DE INFORMAÇÃO EM CIÊNCIA E TECNOLOGIA. **Repositórios Digitais**. Disponível em: <http://www.ibict.br/informacao-para-a-pesquisa/repositorios-digitais>. Acesso em: 23 jul. 2020.

INTERPARES TRUST PROJECT. **Model for Preservation of Trustworthiness of the Digitally Signed, Timestamped and/or Sealed Digital Records (TRUSTER Preservation Model)**. [s.l: s.n.].

JUELS, A.; KALISKI, B. S. Pors: Proofs of retrievability for large files. **Proceedings of the ACM Conference on Computer and Communications Security**, p. 584–597, 2007.

KUROSE, J.; ROSS, K. **Redes de Computadores e a Internet: Uma Abordagem Top-Down**. 6. ed. São Paulo: Pearson Education do Brasil, 2013.

LAMPERT, S. R. Os repositórios DSpace e Archivematica para documentos arquivísticos digitais. **Acervo**, v. 29, n. 2, p. 143–154, 2016.

LEMIEUX, V. L. Blockchain and Distributed Ledgers as Trusted Recordkeeping Systems : An Archival Theoretic Evaluation Framework. **Future Technologies Conference (FTC) 2017**, n. June, p. 1–11, 2017.

LEMIEUX, V. L.; SPORNY, M. **Preserving the Archival Bond in Distributed Ledgers**. Proceedings of the 26th International Conference on World Wide Web Companion - WWW ’17 Companion. **Anais**...New York, New York, USA: ACM Press, 2017

LI, X. et al. A realistic evaluation of memory hardware errors and software system susceptibility. **Proceedings of the 2010 USENIX Annual Technical Conference, USENIX ATC 2010**, p. 75–88, 2019.

MANIATIS, P. et al. The LOCKSS peer-to-peer digital preservation system. **ACM Transactions on Computer Systems**, v. 23, n. 1 SPEC. ISS., p. 2–50, 2005.

MENEZES, A. J.; OORSCHOT, P. C. VAN; VANSTONE, S. A. Hash Functions and Data Integrity. In: **Handbook of Applied Cryptography**. 5. ed. [s.l.] CRC Press, 2001. p. 320–383.

MERKLE, R. C. A certified digital signature. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 435 LNCS, p. 218–238, 1990.

MONTEIRO, L. **Do papel ao monitor possibilidades e limitações do meio eletrônico**. Anais do 24º Congresso Anual em Ciência da Comunicação. **Anais**...Campo Grande, MS: 2001

NAKAMOTO, S. Bitcoin: A Peer-to-Peer Electronic Cash System. n. 1, p. 9, 2008.

NATIONAL ARCHIVES AND RECORDS ADMINISTRATION. **Blockchain White Paper**. [s.l: s.n.].

NATIONAL DIGITAL STEWARDSHIP ALLIANCE; DIGITAL LIBRARY FEDERATION. **About the NDSA**. Disponível em: <https://ndsa.org/about/>. Acesso em: 1 dez. 2018.

NATIONAL RESEARCH COUNCIL. **Building an Electronic Records Archive at the National Archives and Records Administration: Recommendations for a Long-Term Strategy**. Washington, D.C.: National Academies Press, 2005.

NESTOR WORKING GROUP ON TRUSTED REPOSITORIES CERTIFICATION. **Catalogue of criteria for trusted digital repositories**. [s.l: s.n.]. v. 1

O'GORMAN, T. J. et al. Field testing for cosmic ray soft errors in semiconductor memories. **IBM Journal of Research and Development**, v. 40, n. 1, p. 41–49, 1996.

OWENS, E. **Digital Preservation and Electronic Journals**. (S. Ricketts, C. Birdie, E. Isaksson, Eds.)Library and Information Services in Astronomy V: Common Challenges, Uncommon Solutions. **Anais**...Cambridge, Massachusetts, USA: 2007

PHILLIPS, M. et al. **The NDSA Levels of digital preservation: An explanation and uses**. [s.l: s.n.].

PREMIS WORKING GROUP. Data Dictionary for Preservation Metadata. n. June, 2015.

RLG-NARA TASK FORCE ON DIGITAL REPOSITORY CERTIFICATION. **Trustworthy Repositories Audit & Certification: Criteria and Checklist**. [s.l: s.n.].

RLG-OCLC WORKING GROUP ON DIGITAL ARCHIVE ATTRIBUTES. **Trusted Digital Repositories: Attributes and Responsibilities**. Mountain View, CA: [s.n.].

RODRIGUES, M. D. M. **Repositório Arquivístico Digital Confiável para o Patrimônio Documental Oriundo do Processo Judicial Eletrônico**. [s.l.] Universidade Federal de Santa Maria, 2015.

ROSENTHAL, D. S. H. et al. Requirements for Digital Preservation Systems. **D-Lib Magazine**, v. 11, n. 11, nov. 2005.

ROSENTHAL, D. S. H. Bit Preservation: A Solved Problem? **International Journal of Digital Curation**, v. 5, n. 1, p. 134–148, 22 jun. 2010.

SANTOS, H. M. DOS; FLORES, D. Repositórios digitais confiáveis para documentos arquivísticos: ponderações sobre a preservação em longo prazo. **Perspectivas em Ciência da Informação**, v. 20, n. 2, p. 198–218, 2015.

SAYÃO, L. F. et al. **Implantação e gestão de repositórios institucionais: políticas, memória, livre acesso e preservação**. Salvador: EDUFBA, 2009.

SAYÃO, L. F. Repositórios digitais confiáveis para a preservação de periódicos eletrônicos cientificos. **Ponto de Acesso**, v. 4, n. 3, p. 68–94, 2010.

SKINNER, K.; SCHULTZ, M. **A Guide to Distributed Digital Preservation**. Atlanta, GA: Educopia Institute, 2010.

SLAYMAN, C. Soft error trends and mitigation techniques in memory devices. **Proceedings - Annual Reliability and Maintainability Symposium**, 2011.

SMITH, T. D. The blockchain litmus test. **Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017**, v. 2018- Janua, p. 2299–2308, 2017.

SOMASUNDARAM, G. SHRIVASTAVA, A. **Armazenamento e Gerenciamento de Informações: Como armazenar, gerenciar e proteger informações digitais**. Porto Algre: Bookman, 2011.

SONG, S.; JAJA, J. Techniques to audit and certify the long-term integrity of digital archives. **International Journal on Digital Libraries**, v. 10, n. 2, p. 123–131, 2009.

STALLINGS, W. **Criptografia e segurança de redes: princípios e práticas**. 6. ed. São Paulo: Pearson Education do Brasil, 2015.

STONE, J.; PARTRIDGE, C. When the CRC and TCP checksum disagree. **Computer Communication Review**, v. 30, n. 4, p. 309–319, 2000.

SWARD, A.; VECNA, I.; STONEDAHL, F. Data Insertion in Bitcoin’s Blockchain. **Ledger**, v. 3, p. 1–23, 2018.

TANENBAUM, A. S.; BOS, H. **Sistemas Operacionais Modernos**. [s.l: s.n.].

TANENBAUM, A. S.; WETHERALL, D. J. **Redes de Computadores**. 5. ed. São Paulo: Pearson Education do Brasil, 2011.

TASK FORCE ON ARCHIVING OF DIGITAL INFORMATION. Preserving digital information: A review. **Archives and Museum Informatics**, v. 10, n. 2, p. 148–153, jun. 1996.

TÉRMENS, M.; LEIJA, D. Auditoría de preservación digital con NDSA Levels. **El Profesional de la Información**, v. 26, n. 3, p. 447, 2017.

THE WORLD BANK. Distributed Ledger Technology (DLT) and Blockchain. **FinTech note**, n. 1, p. 1–60, 2017.

THOMAZ, K. DE P. Repositórios digitais confiáveis e certificação. **Arquivística.net**, v. 3, n. 1, p. 80–89, 2007.

TRUU, A. **Standards for Hash-Linking Based Time-Stamping Schemes**. [s.l.] University of Tartu, 2010.

UNIVERSITY OF SURREY; NATIONAL ACHIVES (UK); OPEN DATA INSTITUTE. **ARCHANGEL: guaranteeing the integrity of digital archives**. [s.l: s.n.].

VAUGHAN, W.; BUKOWSKI, J.; WILKINSON, S. Chainpoint: A scalable protocol for recording data in the blockchain and generating blockchain receipts. 2016.

VIGIL, M. et al. Integrity, authenticity, non-repudiation, and proof of existence for long-term archiving: A survey. **Computers and Security**, v. 50, p. 16–32, 2015.

VIGNATTI, T. Arquivamento Digital a Longo Prazo Baseado em Seleção de Repositórios em Redes Peer-to-Peer. 2009.

WRIGHT, R.; MILLER, A.; ADDIS, M. The Significance of Storage in the “Cost of Risk” of Digital Preservation. **International Journal of Digital Curation**, v. 4, n. 3, p. 104–122, 2009.

ZIEGLER, J. F. et al. Cosmic ray soft error rates of 16-Mb DRAM memory chips. **IEEE Journal of Solid-State Circuits**, v. 33, n. 2, p. 246–251, 1998.

# Apêndice A — Execução, detecção e comparativo de corrupções em um documento digital

A partir da proposição apontada na Seção 2.1.1, de que o significado de um documento pode ser comprometido com a perda da informação de um único bit[70], foi executado um experimento, a fim de verificar as possíveis adulterações que um documento pode sofrer com a mudança de um bit de informação.

Para este experimento, foi selecionada uma imagem (Figura A.1a) de 508.536 bits (~62,1KiB) como documento, em que foi forçado uma corrupção com o uso de uma aplicação de edição de dados brutos[71]. Com essa aplicação foram gerados duas novas versões do documento (Figura A.1b e Figura A.1c) a partir da simples inversão de um único bit do documento original.

Figura A.1 — Comparativo de possíveis resultados da corrupção de um documento

(a) (b) (c)

Fonte: Próprio autor.

Dentre os possíveis resultados a partir da corrupção de um bit no documento original, foram escolhidas duas versões que enaltecem bem o contraste de possibilidades de corrupção. A mudança de um bit de diferentes formas e em diferentes posições do documento resultou

70 A menor unidade de informação digital.
71 https://userbase.kde.org/Okteta

em estados distintos do mesmo, em que um resultado aparenta estar intacto (Figura A.1b) e, no outro, a informação já se apresenta incompreensível ( Figura A.1c).

A inversão dos bits se deu de maneira aleatória a fim de simular uma potencial instabilidade na mídia de armazenamento. Valendo observar que esse experimento apenas se focou em apresentar o resultado desse possível fenômeno e não realiza uma análise de aspectos técnicos relacionados aos variados tipos de memórias e que possam resultar em tal fenômeno. Para isso devem ser consultadas as análises referenciadas na Seção 2.1 (página 20).

## Metodologia de corrupção

- Para gerar o documento corrompido Figura A.1b a partir do documento Figura A.1a se seguiu os seguintes passos:
  i. Utilização da aplicação Okteta[71] para leitura dos dados brutos do documento original.
  ii. Mudança na forma de codificação dos valores de hexadecimal para executável (binário).
  iii. Alteração do valor ‘00010010’ para ‘00000010’ na posição ‘0000:C986’.

- Para gerar o documento corrompido Figura A.1c a partir do documento Figura A.1a se seguiu os seguintes passos:
  i. Utilização da aplicação Okteta[71] para leitura dos dados brutos do documento original.
  ii. Mudança na forma de codificação dos valores de hexadecimal para executável (binário).
  iii. Alteração do valor ‘11011111’ para ‘11111111’ na posição ‘0000:5D71’.

## Detecção de corrupção

Conforme descrito na Seção 2.3 (página 31) algumas técnicas computacionais permitem a identificação de erros (aleatórios ou intencionais) nos dados durante sua

transmissão ou armazenamento. A geração de um código individual em diferentes instantes no tempo e a comparação desses códigos em seguida pode apontar inconsistências em dados aparentemente intactos, como no caso exemplificado pelo documento Figura A.1b.

Já sabendo da diferença de dados (de um bit) entre elas, as três versões (a, b e c) do documento Figura A.1 foram submetidas a dois algoritmos de verificação de integridade de diferentes classes (*checksum* e *hash*) para observar se os códigos gerados refletiriam essas diferenças e o quanto refletem.

Para a função de *checksum* foi escolhido o CRC-32 devido sua referência e popularidade para esse tipo de função (CASTAGNOLI; BRÄUER; HERRMANN, 1993), é fortemente adotado em padrões de redes de computadores.

O SHA-2 (Secure Hash Algorithm 2) é uma família de algoritmos criptográficos de hash ainda considerados referências para as diversas aplicações que demandem esse tipo de função nos dias atuais (STALLINGS, 2015). O SHA-256 foi a versão selecionada dessa família, que entrega um ótimo custo-benefício (ainda é bastante segura e é mais econômica que o SHA-512), sendo muito popular e indicada, por padrão, em protocolos de segurança na internet[72].

Tabela A.1 — Valores de *checksum* para um objeto digital

| Objeto | Checksum (CRC-32) |
|---|---|
| Figura A.1a | 1660101973 |
| Figura A.1b | 3851490416 |
| Figura A.1c | 3656850431 |

Fonte: Próprio autor.

Tabela A.2 — Valores de *hash* para um objeto digital

| Objeto | Hash (SHA-256) |
|---|---|
| Figura A.1a | 202a1fd52a63a31eabff0383164dbd04a5fe31f4c63103e9ec1c91e8ca998c4e |
| Figura A.1b | c9084d704f675f9c10a9fb43b7ceac1d32ff5ce7999f59d42b740bf28db880d3 |
| Figura A.1c | ba40facb3f6cc77218e767049dc4c9c2a3b6ee05e7c2253fab60a53f31653bb8 |

Fonte: Próprio autor.

72 https://tools.ietf.org/html/rfc8446

Com o uso de implementações dos algoritmos de CRC-32[73] e SHA-256[74] se pôde gerar, para cada um desses tipos, códigos individuais para cada versão do documento (Figura A.1).

As Tabelas A.1 e A.2 registram os códigos gerados para cada versão do documento por cada algoritmo e, pela análise dos valores, nota-se que a corrupção de um único bit foi identificada com sucesso por cada algoritmo, percebendo que foi gerado um código significativamente diferente para cada uma das versões do documento. Com isso, e partindo do estabelecimento do documento Figura A.1a como versão original e dos documentos Figura A.1b e Figura A.1c como potenciais versões autênticas do documento Figura A.1a, pode ser atestado que os documentos Figura A.1b e Figura A.1c são diferentes a nível de bits de sua versão original e, portanto, não podem ser considerados objetos íntegros e muito menos autênticos.

Apesar da detecção de corrupção satisfatória por ambos os algoritmos, caso os resultados demonstrassem o contrário, ainda assim não se poderia atestar a integridade e autenticidade dos documentos apenas com os resultados originados por *checksum*, devido suas fragilidades já apontadas na Seção 2.3 (página 31).

73 cksum (https://pubs.opengroup.org/onlinepubs/9699919799/utilities/cksum.html)
74 sha256sum (https://www.gnu.org/software/coreutils/manual/html_node/sha2-utilities.html)

# Apêndice B — Itens normativos candidatos a serem atendidos pela implementação da proposta

A partir dos documentos apresentados na Seção 2.2.3 (página 29), nota-se que a proposta deste trabalho pode ser somada a práticas já executadas ou recomendadas pelos documentos de referência nos repositórios. A seguir, são listados itens desses documentos em que, de alguma forma, o RDC pode apontar o uso de uma implementação desta proposta para demonstrar um nível de conformidade ou incremento no nível de confiabilidade desses itens:

1. **NDSA Levels –** A proposta pode atender a todos os níveis da categoria File Fixity and Data Integrity.
2. **CoreTrustSeal –** A proposta pode atender a todos os níveis da categoria Data integrity and authenticity.
3. **Catalogue of Criteria for Trusted Digital Repositories (NESTOR)**
   - 6 — The digital repository ensures the integrity of the digital objects during all processing stages.
   - 6.2 — Archival Storage: the digital repository ensures the integrity of the digital objects.
   - 6.3 — Access: the digital repository ensures the integrity of the digital objects.
   - 8 — The digital repository has a strategic plan for its technical preservation measures.
   - 10.3 — The digital repository guarantees the storage and readability of the AIPs.
   - 12 — The data management system is capable of providing the necessary digital repository functions.
   - 13.2 — The IT infrastructure implements the security demands of the IT security system.
4. **ISO 16363 (ACTDR)**
   - 3.3.5 — The repository shall define, collect, track, and appropriately provide its information integrity measurements.
   - 4.2.6.3 — The repository shall ensure that the PDI is persistently associated with the relevant Content Information.
   - 4.2.8 — The repository shall verify each AIP for completeness and correctness at the point it is created.

- 4.2.9 — The repository shall provide an independent mechanism for verifying the integrity of the repository collection/content.
- 4.4.1.2 — The repository shall actively monitor the integrity of AIPs.
- 5.1.1.2 — The repository shall have adequate hardware and software support for backup functionality sufficient for preserving the repository content and tracking repository functions.
- 5.1.1.3 — The repository shall have effective mechanisms to detect bit corruption or loss.
- 5.1.1.3.1 — The repository shall record and report to its administration all incidents of data corruption or loss, and steps shall be taken to repair/replace corrupt or lost data.
- 5.1.2 — The repository shall manage the number and location of copies of all digital objects.
- 5.1.2.1 — The repository shall have mechanisms in place to ensure any/multiple copies of digital objects are synchronized.