

RELATÓRIO FINAL | IBICT – CGTI

Estudo para implantação de Repositório Arquivístico Digital Confiável no âmbito do Tribunal de Justiça do Distrito Federal e Territórios: Relatório Final

Autores

Tiago Emmanuel Nunes Braga, Larissa de Araújo Alves, Alexandre Faria de Oliveira

Colaboradores

Jessica Andrade Costa, Tatiana Canelhas Pignataro, Maxwell Borges Bezerra, Henrique Oliveira Costa, Danilo Li, Marcos Sigismundo da Silva



Brasília
Outubro | 2020

INSTITUTO BRASILEIRO DE INFORMAÇÃO EM
CIÊNCIA E TECNOLOGIA (IBICT)

Diretoria
Cecília Leite Oliveira

Coordenação-Geral de Pesquisa e Desenvolvimento
de Novos Produtos (CGNP)
Anderson Itaborahy

Coordenação -Geral de Pesquisa e Manutenção
de Produtos Consolidados (CGPC)
Bianca Amaro

Coordenação-Geral de Tecnologias de Informação
e Informática (CGTI)
Tiago Emmanuel Nunes Braga

Coordenação de Ensino e Pesquisa, Ciência
e Tecnologia da Informação (COEPE)
Gustavo Saldanha

Divisão de Editoração Científica (DIECI)
Ramón Martins Sodoma da Fonseca



Autores

Tiago Emmanuel Nunes Braga
Larissa de Araújo Alves
Alexandre Faria de Oliveira

Colaboradores

Jessica Andrade Costa
Tatiana Canelhas Pignataro
Maxwell Borges Bezerra
Henrique Oliveira Costa
Danilo Li
Marcos Sigismundo da Silva

Revisão Gramatical

Rafael Teixeira de Souza

Capa

Nuielle Cristine de Medeiros da Silva

Coordenação do projeto

Tiago Emmanuel Nunes Braga
Alexandre Faria de Oliveira

Contato

tiagobraga@ibict.br | (61) 3217-6257

B813 Braga, Tiago Emmanuel Nunes

Estudo para implantação de Repositório Arquivístico Digital Confiável no âmbito do Tribunal de Justiça do Distrito Federal e Territórios: Relatório final / Tiago Emmanuel Nunes Braga, Larissa de Araújo Alves, Alexandre Faria de Oliveira – Brasília, DF: IBICT, 2020.

100 p.

ISBN: 978-65-89167-00-6

1. Repositório arquivístico digital confiável – RDC-Arq. 2. Projeto de pesquisa – resultados. 3. Instituto Brasileiro de Informação em Ciência e Tecnologia. 4. Tribunal de Justiça do Distrito Federal e Territórios. I. Alves, Larissa de Araújo. II. Oliveira, Alexandre Faria de. III. Título.

CDU 930.251:004

SUMÁRIO

1. INTRODUÇÃO	5
2. RESULTADOS	6
2.1 Resultados Meta 1	7
2.2 Resultados Meta 2	7
2.3 Resultados Meta 3	8
2.4 Resultados Meta 4	8
2.4.1 Manuais técnicos	8
2.4.2 Disseminação de conhecimento	9
2.4.3 Avanços no modelo de barramento	12
3. DOCUMENTAÇÃO DO PROJETO	13
4. CONSIDERAÇÕES FINAIS	16
REFERÊNCIAS	17
ANEXO A – MANUAIS	18
ANEXO B – CÓDIGO PILOTO DE ENTRADA HIPÁTIA	79
ANEXO C – CÓDIGO CLASSE GERAL HIPÁTIA	97

1. INTRODUÇÃO

O Instituto Brasileiro de Informação em Ciência e Tecnologia (IBICT), em consonância com sua missão institucional, dedica-se em desempenhar ações vinculadas ao desenvolvimento, aprimoramento e transferência de tecnologias de informação. Dentre os aspectos temáticos que orientam o trabalho do IBICT, a preservação digital, com enfoque na informação científica e tecnológica brasileira, é uma pauta priorizada no Instituto desde de 2002. A pesquisa do IBICT nessa temática culminou com a criação da Rede Cariniana, “uma rede de serviços de preservação digital de documentos eletrônicos brasileiros, com o objetivo de garantir seu acesso contínuo a longo prazo” (IBICT, 2016, on-line), que possuía como foco a preservação de conteúdos científicos.

Em 2018, com o início do projeto *Estudo para Implantação de Repositório Arquivístico Digital Confiável* (RDC-Arq),¹ as ações do IBICT, associadas à preservação digital expandiram seu escopo e passaram, também, a tratar questões relacionadas aos documentos arquivísticos digitais, com foco em documentos jurídicos.

De acordo com as [Diretrizes para a implementação de repositórios arquivísticos digitais confiáveis: RDC-Arq](#) (2017, p. 5),

O arquivamento e a preservação digital constituem uma questão complexa que envolve muitas variáveis, compromissos de longa duração e a necessidade de expressivos investimentos em infraestrutura tecnológica, pesquisa e recursos humanos. Diante disso, a formação de consórcios, em determinados casos, pode ser a solução mais viável.

A partir dessa recomendação, e considerando a expertise que o IBICT possui em propor metodologias e tecnologias adequadas para resolução de problemas relacionados às etapas do ciclo informacional (aqui entendidas como geração, seleção, representação, armazenamento, recuperação, disseminação e preservação), o Instituto firmou parceria com Tribunal de Justiça do Distrito Federal e Territórios (TJDFT) por meio de um Termo de Execução Descentralizado, o qual foi denominado *Estudo para Implantação de Repositório Arquivístico Digital Confiável* (RDC-Arq).

O projeto de pesquisa fruto do acordo entre IBICT e TJDFT, teve como principal objetivo a apresentação

de alternativa de soluções para implantação de um RDC-Arq, integrado com solução de gerenciamento documental, tornando possível a elaboração e disseminação de modelo tecnológico de mapeamento, preservação e disseminação de informações associadas (IBICT, 2018)

Esse projeto possui metas que englobaram desde o levantamento e avaliação inicial das soluções tecnológicas já existentes, com potencial para aplicação às necessidades da pesquisa, até a etapa de transferência dos conhecimentos e resultados produzidos.

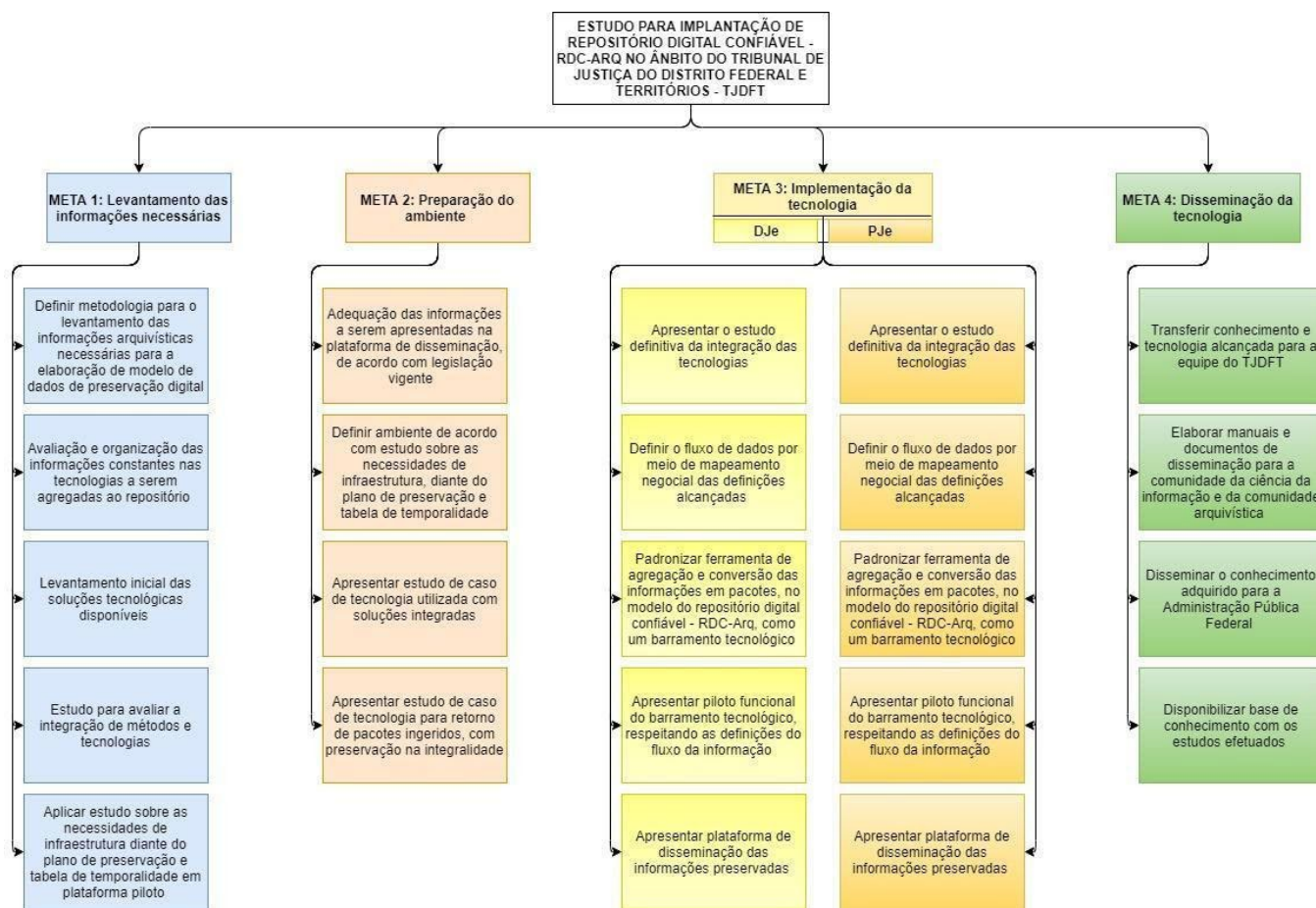
A execução das atividades e o desenvolvimento do projeto de pesquisa em questão gerou conhecimentos relativos às quatro principais metas definidas na proposta de projeto básico construída em parceria entre o IBICT e o TJDFT. São elas:

- Meta 1: Levantamento de informações necessárias;
- Meta 2: Preparação de ambiente
- Meta 3: Implementação da tecnologia;
- Meta 4: Disseminação da tecnologia.

No “Relatório parcial relativo à Meta 3”, de dezembro de 2019, é apresentada a estrutura analítica do projeto (EAP), exposta na **Figura 1**, a seguir, com a finalidade de explicitar o modo como as ações realizadas ao longo da pesquisa apresentam consonância com as metas definidas e foram diretamente orientadas a partir delas.

¹ “Repositório digital confiável é um repositório digital que é capaz de manter autênticos os materiais digitais, de preservá-los e prover acesso a eles pelo tempo necessário” (CONARQ, 2017, p. 9).

Figura 1 – Estrutura analítica do projeto (EAP)



Fonte: Relatório parcial relativo à meta 3 (2019)

Dada a contextualização supramencionada, o escopo do presente relatório é reunir e sistematizar as principais informações vinculadas aos resultados, documentos, produtos e tecnologias oriundas das atividades desenvolvidas no período entre 2018 e 2020 e no contexto do projeto *Estudo para Implantação de Repositório Arquivístico Digital Confiável* (RDC-Arq), com foco nos resultados da Meta 4.

2. RESULTADOS

Dentre os resultados obtidos no contexto do projeto de pesquisa firmado entre IBICT e TJDFT, entende-se o desenvolvimento da infraestrutura RDC-Arq – que envolve o barramento tecnológico para preservação digital, batizado de Hipátia,² software de repositório Archivematica e software de visualização AtoM – como o principal resultado. Esta caracteriza-se como uma solução tecnológica de integração arquivística que possibilita a compatibilidade dos dados entre bases arquivísticas do TJDFT e o software Archivematica e, posteriormente, o AtoM. Além do barramento, outros resultados surgiram por meio dos esforços empreendidos pelas equipes do IBICT e do TJDFT no desempenho das atividades necessárias para alcançar os objetivos do projeto.

Como procedimento de controle adotado pela equipe, ao final da realização das atividades relacionadas à cada meta foi entregue um relatório parcial de resultados do projeto. O presente relatório dá ênfase aos resultados da última meta, a quatro. Todavia, com o intuito de contextualizar a evolução das atividades de divulgação e disseminação do conhecimento, nas seções 2.1 a 2.3 deste documento estão sumarizados os principais aspectos abordados nos relatórios parciais das metas 1, 2 e 3.

² Hipátia foi uma matemática e filósofa ligada à biblioteca de Alexandria, tendo sido assassinada por causa de seus valores ligados aos deuses gregos. Sua história foi esquecida por séculos e, depois, resgatada por historiadores. Por esse motivo, serviu de inspiração para nomear o barramento da pesquisa

Como procedimento de controle adotado pela equipe, ao final da realização das atividades relacionadas à cada meta, foi entregue um relatório parcial de resultados do projeto. O presente relatório dá ênfase aos resultados da última meta, a quatro. Todavia, com o intuito de contextualizar a evolução das atividades de divulgação e disseminação do conhecimento, nas seções 2.1 a 2.3 deste documento estão sumarizados os principais aspectos abordados nos relatórios parciais das metas 1, 2 e 3.

2.1 Resultados Meta 1

Situadas no início do projeto *Estudo para Implantação de Repositório Arquivístico Digital Confiável* (RDC-Arq), as atividades da meta 1 envolviam ações de levantamento e identificação de informações, metodologias e tecnologias relevantes para embasar o prosseguimento das etapas subsequentes. Assim, são atividades de caráter situacional, diagnóstico.

O relatório parcial da meta 1 aponta como um dos resultados a criação da Estrutura Analítica do Projeto (EAP), apresentada na introdução deste documento. Também foram levantadas legislações e normas nacionais, internacionais e arquivísticas, e recomendações de órgãos reguladores, como o Conselho Nacional de Arquivos (CONARQ), importantes para a temática dos repositórios digitais confiáveis.

Além disso, foi apresentada a classificação da pesquisa e uma proposta inicial de metodologia para a construção do barramento arquivístico. Sendo esse, em um primeiro momento, explicado “[...] como link de conexão do produtor [de documentos], uma vez conectado ao banco de dados (produtor), fará sua varredura, respeitando a cadeia de custódia e temporalidade dos documentos arquivísticos digitais e criará Pacotes de Submissão de Informação (SIP) e os enviará ao Archivematica, que os submeterá aos micro serviços e então gerará Pacotes de Arquivamento de Informação (AIP)”.

Aspectos vinculados à identificação de características dos sistemas do Diário de Justiça Eletrônico (DJe) e do Processo Judicial Eletrônico (PJe), a exemplo do volume de armazenamento, também englobam os resultados dessa meta. Por fim, o relatório parcial da meta 1 avaliou algumas tecnologias, a exemplo dos softwares PREMIS, TIKA, JHOVE, DAITSS e Archivematica e classificou o último como o software mais adequado para a construção de um repositório digital confiável inicial, por se tratar de uma tecnologia bem aceita e utilizada pela comunidade arquivística em geral.

2.2 Resultados Meta 2

Para cumprimento da meta de preparação do ambiente, foram realizadas ações baseadas em quatro pontos principais, que são descritos nos próximos parágrafos.

A adequação das informações a serem apresentadas na plataforma de disseminação, de acordo com legislação vigente, ocorreu por meio de mapeamentos dos sistemas do DJe e PJe, além da comparação dos diagnósticos com as legislações levantadas durante a primeira meta.

No tópico de definição de ambiente de acordo com estudo sobre as necessidades de infraestrutura, diante do plano de preservação e tabela de temporalidade, com o levantamento sobre a infraestrutura dos sistemas foram identificados o plano de preservação e a tabela de temporalidade. Essas informações foram relacionadas de modo a criar uma definição do ambiente necessário.

No que se refere à apresentação do estudo de caso de tecnologia utilizada com soluções integradas, a equipe efetuou um estudo com informações técnicas sobre as tecnologias utilizadas, definindo o escopo delas e apontando uma proposta de estrutura tecnológica que fosse mais adequada ao atendimento do projeto.

Por fim, a apresentação estudo de caso de tecnologia para retorno de pacotes ingeridos, com preservação de integridade, evidenciou como os pacotes podem ser ingeridos e preservados integralmente, garantindo os requisitos da cadeia de preservação intersistemas por meio de descrição e execução de teste de ingestão e preservação no Archivematica.

2.3 Resultados Meta 3

Os resultados vinculados à terceira meta do projeto apresentam características voltadas sobretudo à aplicação. As atividades das primeiras metas foram importantes para fornecer subsídios às etapas de 1) desenvolvimento completo do barramento capaz de realizar a integração DJe /Archivematica; e 2) teste de integração do PJe / Archivematica.

Para a Meta 3 estavam previstas ações de integração de tecnologias. Tal combinação ocorreu por meio da preparação de um ambiente no qual foi configurado e instalado o barramento que se caracteriza como o foco da integração das tecnologias, visto que conecta as bases de dados do TJDFT ao Archivematica e AtoM, fazendo uso da linguagem de programação Phyton como base para a construção do mecanismo automatizado.

Assim, vale destacar que os resultados da Meta 3 englobam avanços tecnológicos do projeto. Dentre eles destaca-se a implementação de várias tecnologias integradas, culminando no piloto de barramento arquivístico junto ao modelo de RDC-Arq.

2.4 Resultados Meta 4

Embasada por todos os resultados e esforços empreendidos nas ações vinculadas às metas anteriores, a meta 4 compreendeu a etapa de consolidação do projeto. Nesta seção, a ênfase está na compilação do que foi realizado com o intuito de promover a troca de experiências e a disseminação dos conhecimentos e tecnologias desenvolvidas pelo projeto.

2.4.1 Manuais técnicos

Dentre as atividades previstas para a meta 4 estava a elaboração de manuais técnicos que fossem capazes de orientar a realização de procedimentos de implementação das tecnologias envolvidas no barramento arquivístico. No total, foram criados seis manuais, cujos títulos e breves descrições estão listados a seguir:

- [Formatos aceitos pelo PJe](#)

Descreve os formatos de arquivo aceitos pelo software Archivematica.

- [Manual do levantamento dos Metadados utilizados no crosswalk do barramento arquivístico](#)

Com base na Resolução n. 91 do CNJ, de 29 de setembro de 2009, no e-ARQ Brasil, Resolução n. 24 do CONARQ, apresenta os principais metadados a serem armazenados nas tabelas banco de dados do DJe ou que são pesquisados por meio do sistema do Modelo Nacional de Interoperabilidade (MNI) 2 para o PJe, compilados seguindo as principais normas arquivísticas brasileiras.

- [Archivematica em Red Hat: Instalação com o SGBD, Elasticsearch e MCP Client\(s\) segregados](#)

O manual descreve o procedimento de instalação do Archivematica 1.11 em Red Hat Enterprise Linux 7 customizado conforme as especificações do Tribunal de Justiça do Distrito Federal e Territórios (TJDFT).

- [Integração de AtoM em Docker com Archivematica em VM](#)

Enumera e esclarece os passos necessários para efetuar a integração de um AtoM versão 2.5 instalado em Docker com o software Archivematica na versão 1.10 e que esteja instalado em VM

- [RHEL: atualização do Ghostscript Archivematica](#)

De modo a evitar o bug no software Archivematica, causado pela instalação do Ghostscript na versão 9.25, este manual orienta quais os procedimentos realizar para a atualizar o Ghostscript e o configurar a fim de que o Archivematica passe a usar a versão atualizada.

- [Portable Document Format/Archive: PDF/A](#)

Analisa os requisitos, as características, os níveis de conformidade e os tipos existentes do formato de documento PDF/A.

A íntegra do conteúdo dos manuais supracitados está acessível por meio do Diretório de documentos localizado no Redmine do Projeto (<http://redmine.ibict.br/projects/tjdft/documents>) e seguem anexos a este relatório. Vale ressaltar que esses manuais também são resultados que cumprem com a atividade da meta 4 que previa a transferência de conhecimento e tecnologia alcançada para equipe do TJDFT.

2.4.2 Disseminação de conhecimento

Importante para a manutenção das atividades científicas e para a evolução do conhecimento, a comunicação científica caracteriza-se como um sistema que engloba processos relacionados à produção, circulação, disseminação, avaliação e uso dos resultados de pesquisas, de modo que eles possam ser validados e divulgados tanto para a comunidade acadêmica quanto para a sociedade como um todo.

A **Figura 2**, a seguir, demonstra alguns aspectos envolvidos no ciclo da comunicação científica, posicionando as ações de divulgação e disseminação de informações como etapas intermediárias que fomentam ações posteriores até se chegar à produção de novos conhecimentos.

Figura 2 - Comunicação científica: contextualização



Fonte: elaborada pelos autores

No âmbito do projeto *Estudo para Implantação de Repositório Arquivístico Digital Confiável (RDC-Arq)*, a principal estratégia adotada a fim de promover a disseminação e a troca dos conhecimentos gerados para a Administração Pública Federal e a sociedade civil foi a participação dos pesquisadores envolvidos no projeto de eventos gratuitos *on-line*. Os detalhes de tais eventos estão dispostos a seguir:

- *I Simpósio Preservação Digital no TJMG: desafios e perspectivas para a gestão de documentos arquivísticos eletrônicos*

Organizado na modalidade presencial pelo Tribunal de Justiça do Estado de Minas Gerais (TJMG), este evento ocorreu no dia 10 de março de 2020 e contou com a participação do Coordenador do projeto, Dr. Tiago Emmanuel Nunes Braga, e Tatiana Canelhas Pignataro, pesquisadora do IBICT, que realizou uma apresentação intitulada *Desenvolvimento de tecnologias no TJDFT para Preservação Digital*. Os slides usados pela pesquisadora durante a apresentação estão na aba de documentos do projeto no Redmine.

Figura 3 – Folder de divulgação do I Simpósio Preservação Digital no TJMG



Fonte: <http://ejef.tjmg.jus.br/i-simposio-de-preservacao-digital-no-tjmg-desafios-e-perspectivas-para-a-gestao-de-documentos-arquivisticos-eletronicos/>

Figura 4 – Slides da apresentação da pesquisadora Tatiana Canelhas no I Simpósio



Fonte: Redmine do projeto

- *Webinar Preservação digital: o case do TJDFt na implementação do RDC-Arq*

Transmitido por meio do canal do TJDFt no YouTube, o Webinar em questão ocorreu no dia três de setembro de 2020, teve duração aproximada de 2h30min e contou com a participação dos seguintes palestrantes: Desembargadora Ana Maria Duarte Amarante Brito (Primeira Vice-Presidente do TJDFt); Dr. Cecília Leite Oliveira (Diretora do IBICT); Dr. Tiago Emmanuel Nunes Braga (Coordenador-Geral da CGTI/IBICT); Tatiana Canelhas Pignataro (IBICT); Daniel Flores (TJDFt); Daniel Monteiro (TJDFt); Francisco Antônio Alves de Oliveira (TJDFt); e Luiz Fernando Sirotheau Serique Junior (TJDFt).

O evento foi divulgado nas redes sociais do IBICT e, também, nos sites oficiais, tanto do instituto quanto do TJDFt. As matérias de divulgação estão disponíveis em:

- <https://www.tjdft.jus.br/institucional/escola-de-administracao-judiciaria/programacao/webinar-preservacao-digital-o-case-do-tjdft-na-implementacao-do-rdc-arq-publico-externo>
- <https://ibict.br/sala-de-imprensa/noticias/item/2395-tjdft-na-implementacao-do-rdc-arq-conheca-os-resultados-do-projeto-em-parceria-com-o-ibict>

Após o término do evento, o webinar ficou gravado e está acessível gratuitamente em: <https://www.YouTube.com/watch?v=kWKDlwQbk1k>

Figura 5 – Webinar Preservação digital: vídeo



Fonte: <https://www.youtube.com/watch?v=kWKDIwQbk1k>

- *Seminário Telepresencial da Semana da Memória da Justiça do Trabalho*

Evento organizado pelo Tribunal Superior do Trabalho (TST) e transmitido on-line pelo YouTube do Tribunal, englobou a temática de implantação de RDC-Arq na palestra proferida por Tatiana Canelhas, que teve como título: *A importância da Implantação de um Rdc-Arq (Archivemática e Atom)*.

Figura 6 – Seminário Telepresencial

A graphic for a telepresence seminar. It features a stylized geometric logo on the left. The text on the right reads: "SEMINÁRIO TELEPRESENCIAL da Semana da Memória da Justiça do Trabalho" and "TRABALHO E JUSTIÇA DO TRABALHO: MEMÓRIA E HISTÓRIA". Below this, it states the date and time: "22 DE SETEMBRO (TERÇA-FEIRA), DAS 8H30 ÀS 19H (HORÁRIO DE BRASÍLIA)" and "TRANSMISSÃO AO VIVO NO CANAL DO TST NO YOUTUBE". At the bottom, it lists the 15h lecture: "15h – Palestra 'A Importância da Implantação de um Rdc-Arq (Archivemática e Atom)'" and provides details about the speaker, Dr. Daniel Flores, and the president of the panel, Minister Douglas Alencar Rodrigues.

SEMINÁRIO TELEPRESENCIAL
da Semana da Memória da Justiça do Trabalho

TRABALHO E JUSTIÇA DO TRABALHO:
MEMÓRIA E HISTÓRIA

22 DE SETEMBRO (TERÇA-FEIRA),
DAS 8H30 ÀS 19H (HORÁRIO DE BRASÍLIA)
TRANSMISSÃO AO VIVO NO CANAL DO TST NO YOUTUBE

15h – Palestra “A Importância da Implantação de um Rdc-Arq (Archivemática e Atom)”
Palestrante: professor Dr. Daniel Flores (UFF), com colaboração da pesquisadora Tatiana Canelhas Pignataro
Lattes: <http://lattes.cnpq.br/9640543272532398>
Presidente da mesa: ministro Douglas Alencar Rodrigues, membro da Comissão de Documentação do TST

Fonte: <https://www.tst.jus.br/-/semin%C3%A1rio-e-exposi%C3%A7%C3%A3o-on-line-marcam-semana-da-mem%C3%B3ria-da-justi%C3%A7a-do-trabalho>

Os conteúdos apresentados anteriormente comprovam os esforços e ações empreendidos no âmbito do projeto a fim de cumprir com o previsto para a meta 4. Tanto os manuais quanto os eventos mencionados corroboram a transmissão e disseminação dos conhecimentos produzidos ao longo do projeto de pesquisa entre IBICT e TJDFT.

2.4.3 Avanços no modelo de barramento

Embora o foco da meta 4 estivesse na disseminação dos conteúdos gerados durante a pesquisa, vários avanços no modelo de barramento proposto foram alcançados durante a realização da mesma. Tais avanços permitiram tornar mais genérico o modelo de construção do Hipátia e, também, compreender situações identificadas durante a pesquisa.

Dentre os avanços, pode-se citar a segregação de máquinas para otimização da performance do processo de preservação. Ela é descrita no documento anexo *Archivematica em Red Hat: Instalação com o SGBD, Elasticsearch e MCP Client(s)* segregados e representou a implementação de um modelo de segregação ainda não disseminado em nível internacional. Nesse sentido, o projeto representa um avanço a ser utilizado pela comunidade internacional e coloca em voga a vanguarda obtida pelo projeto.

Outro avanço obtido durante a execução dessa meta foi a implementação dos sistemas em plataformas distintas. O AtoM utilizado para integração do modelo foi disponibilizado segundo a estrutura de *container*, que é um conceito contemporâneo voltado para a otimização dos recursos informacionais. Esta arquitetura foi integrada com o Archivematica, disponibilizado em *Virtual Machine (VM)*, que é uma arquitetura mais tradicional. A integração das duas infraestruturas é um avanço considerável e serve para vislumbrar a possibilidade de adaptação do modelo de preservação a uma gama de infraestruturas ainda não exploradas.

Destaca-se também a refatoração do código do Hipátia, que após a prova de modelo, realizada ao final da meta 2, passou por uma reescrita completa, tendo sido finalizada durante a Meta 4.

A estrutura existente até a meta 3 é apresentada a seguir:

Figura 7 – estrutura do código até a meta 3

Name	Last commit
 <u>Classes</u>	Correções no código e instruções pro Crontab.
 .gitignore	Iniciado o processo de automatização do PJE
 ConsultaBanco.py	Adicionado a conversão de código para nomenclatura dos campos n...
 dje.xml	troca da api_key e Folder_uuid para o archivematica 1.10
 piloto.py	Adicionado a captura do arquivo P7S;Atualizado(não finalizado) nom...
 pje.xml	Inserido envio para o archivematica do PJe
 scriptBD.sql	melhorias

Fonte: elaborada pelo projeto

Com os avanços obtidos após o processo de otimização dos códigos, bem como a implementação das funcionalidades descritas anteriormente, a infraestrutura do sistema foi remodelada, conforme apresentado a seguir

Figura 8 – Estrutura do código remodelada

Name	Last commit
 arquivos	Inserido a criação do bagit
 logs	reestruturando o código
 sistemas	Adicionado comentário em todas as classes e funções e adiciona...
 .gitignore	Iniciado o processo de arquivamento em lote
 ClasseGeral.py	Adicionado comentário em todas as classes e funções e adiciona...
 EnvioArquivematica.py	Adicionado a funcionalidade de limitador de tentativas após um p...
 Imports.py	reestruturando o código
 ProximoArquivo.py	Inserido envio em lote PJe
 VerificaStatus.py	corrigido o arquivo verificadorDeErros.JSON, onde caso não exist...
 bagit.py	melhorias bagit

Fonte: elaborada pelo projeto

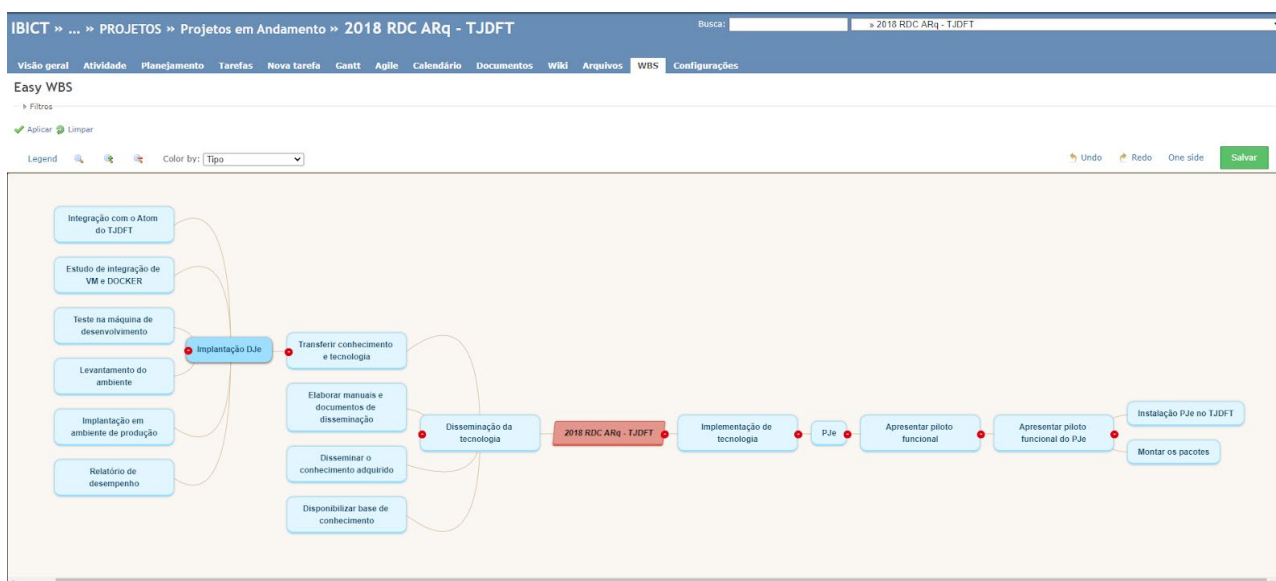
De certa forma, essa refatoração do código também está associada à etapa de disseminação. Ela permite que outras equipes tenham mais facilidade em trabalhar com o Hipátia e adaptá-lo às necessidades das instituições que se propuserem a adotá-lo. Além disso, a implementação de novas funcionalidades será facilitada pela inclusão da organização modular. Os códigos de entrada do Hipátia antigo e atual estão disponíveis nos anexos Piloto.py e ClasseGeral.py, respectivamente.

3. DOCUMENTAÇÃO DO PROJETO

No âmbito da pesquisa científica, o registro das descobertas e dos conhecimentos produzidos a partir da aplicação de metodologias é uma atividade fundamental. Por meio da documentação produzida, um projeto constrói a sua memória informacional e possibilita que as informações possam ser difundidas e disseminadas entre os membros da equipe e outras partes interessadas nos resultados obtidos por meio das atividades de pesquisa. Desse modo, o registro de informações favorece as práticas de comunicação e divulgação científica, as quais são fundamentais para a manutenção e aprimoramento da ciência.

Ao longo do período de andamento do projeto *Estudo para Implantação de Repositório Arquivístico Digital Confiável (RDC-Arq)*, todos os documentos e registros produzidos foram gerenciados e armazenados no diretório de arquivos do software Redmine, um gerenciador de projetos que, em síntese, é um facilitador dos trabalhos de gestão e acompanhamento das atividades de pesquisa. As principais funcionalidades que o software apresenta na execução dos projetos são: a centralização das ações, o controle de entregas, a gestão dos prazos, a apresentação dos resultados, a organização das demandas, a gestão dos pesquisadores e a padronização das entregas.

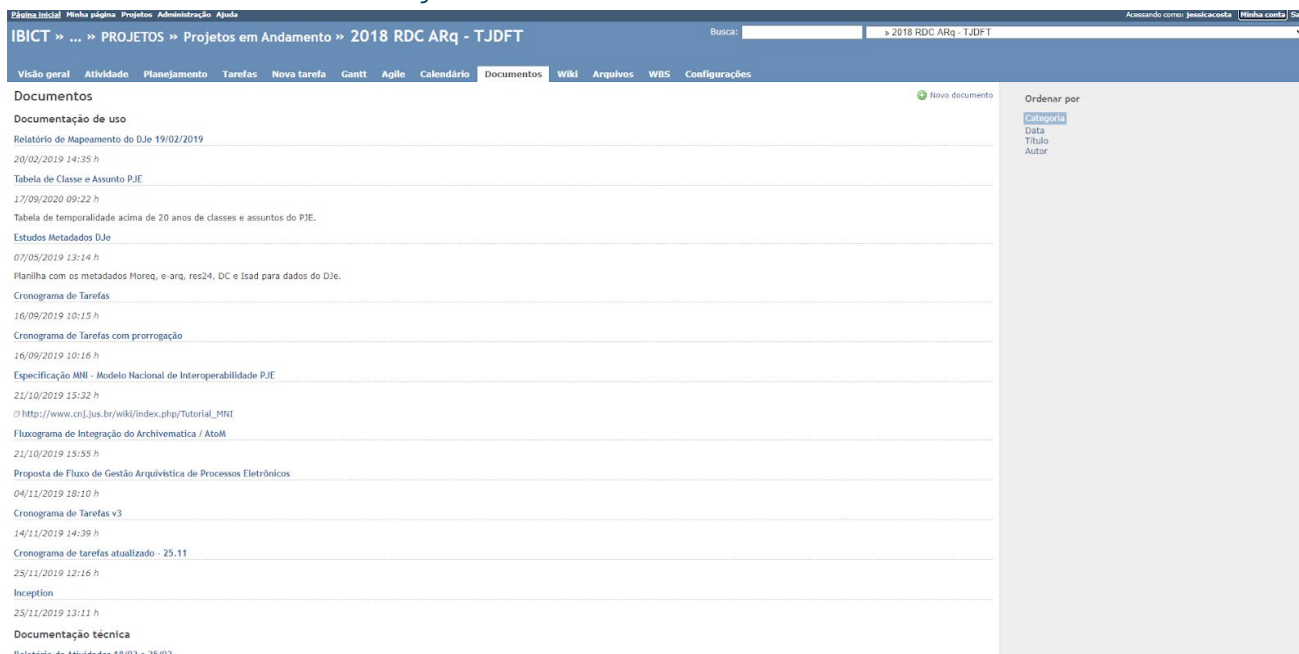
Figura 9 - Work Breakdown Structure



Fonte: redmine.ibict.br/projects/tjdft/easy_wbs

O Redmine possui a ferramenta WBS (Work Breakdown Structure), que mostra de forma interativa e hierárquica a estrutura analítica do projeto, conforme a **Figura 9**. Tal funcionalidade proporcionou o acompanhamento do projeto de forma sistemática, permitindo a rastreabilidade de cada atividade até as metas e objetivos, auxiliando o gerenciamento do todo.

Figura 10 - Aba Documentos software RedMine



Fonte: redmine.ibict.br/projects/tjdft/documents

A documentação produzida pelos pesquisadores foi catalogada e disponibilizada para consulta, permitindo o fácil acesso de toda a equipe envolvida no projeto, além da simplificação no controle de entregas, gestão de prazos e entregas. A **Figura 10** exibe a funcionalidade que permite o gerenciamento da documentação, atuando como a biblioteca do projeto.

Entendido por Cunha e Cavalcanti (2008, p. 320) como “**1.1** Documento que registra o resultado de uma pesquisa ou estudo realizado por um indivíduo ou por um grupo. **1.2** Documento que registra as atividades de uma instituição durante certo período de tempo”, o relatório é o principal tipo de registro produzido pelo projeto

em questão. No que se refere aos subtipos, a pesquisa gerou relatórios de atividades, de produtos e de entregas, os quais encontram-se disponíveis no redmine do projeto.

De modo a oferecer uma ampla visualização da produção documental elaborada pelos bolsistas envolvidos no projeto de implementação de um RDC-Arq, apresentamos, a seguir, o **Quadro 1**, que sistematiza e reúne os documentos a partir de sua tipologia.

Quadro 1 - Documentação do Projeto

RELATÓRIOS DE ATIVIDADES		
Autor (Bolsista)	Data	
Ana Clara Fonseca Vassallo	maio 2019	
Ana Clara Fonseca Vassallo	outubro 2019	
Henrique Oliveira Costa	outubro 2019	
Luísa Kruchak Barros	outubro 2019	
Maxwell Borges Bezerra	setembro 2019	
Tatiana Canelhas Pignataro	outubro 2019	
Tatiana Canelhas Pignataro	janeiro 2020	
RELATÓRIOS DE PRODUTO		
Título	Data	Autor (Bolsista)
Relatório de produto I Estudo de software do projeto RDC-Arq	2019	Ana Clara Vassallo
Relatório de produto II - Metodologia para o levantamento das soluções tecnológicas, tecnologias e funcionalidades	agosto 2019	Ana Clara Vassallo
Relatório de produto III - Visibilidade na integração de tecnologias	agosto 2019	Ana Clara Vassallo
Relatório de produto VI - Estudo de Software para o Projeto RDC-Arq: Metadados	outubro 2019	Ana Clara Vassallo
MANUAIS		
Título		Data
Formatos aceitos pelo Pje		maio 2019
Portable Document Format/Archive: PDF/A		maio 2019
Manual do levantamento dos Metadados utilizados no crosswalk do barramento arquivístico		novembro 2019
Archivematica em Red Hat: Instalação com o SGBD, Elasticsearch e MCP Client(s) segregados		junho 2020
Integração de AtoM em Docker com Archivematica em VM		junho 2020
RHEL: atualização do Ghostscript Archivematica		junho 2020
RELATÓRIOS DE ENTREGA		
Título	Data	
Relatório de Meta I	abril 2019	
Relatório de Meta II	maio 2019	
Relatório de Meta III, v. 1 e v. 2	dezembro 2019	
Relatório Final - Meta IV	outubro 2020	
Tatiana Canelhas Pignataro	outubro 2019	
Tatiana Canelhas Pignataro	janeiro 2020	

Fonte: elaborado pelos autores

4. CONSIDERAÇÕES FINAIS

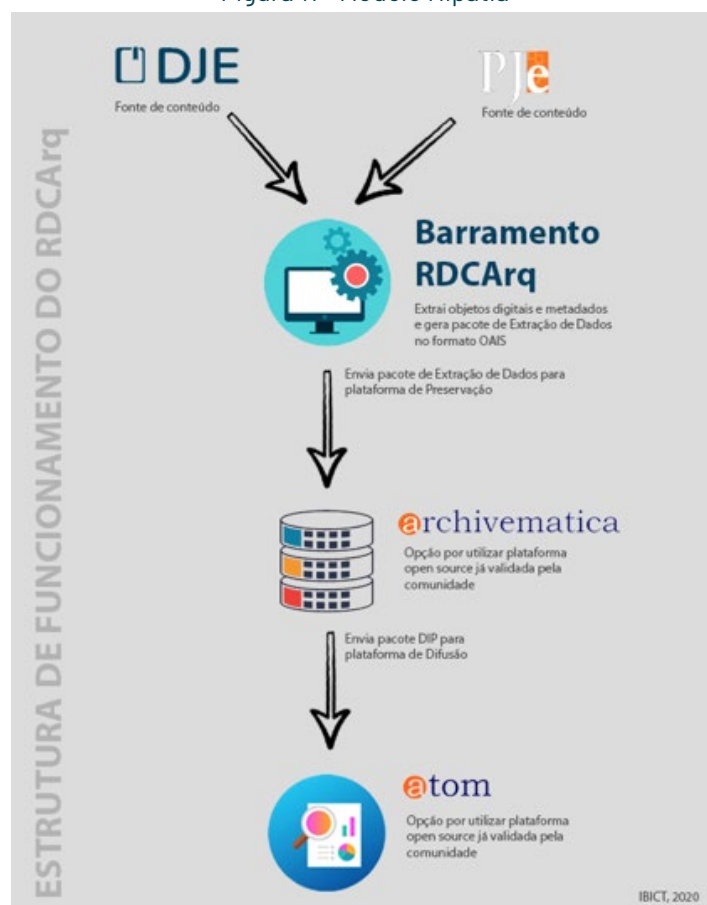
O presente relatório teve como objetivo a sumarização dos resultados obtidos pelo projeto Estudo para Implantação de Repositório Arquivístico Digital Confiável (RDC-Arq), a partir da execução de atividades orientadas conforme a EAP e a apresentação específica dos resultados vinculados à Meta 4: disseminação da tecnologia e dos resultados da pesquisa.

As conquistas alcançadas pelo projeto viabilizado por meio da parceria IBICT-TJDFT demonstram a aplicação e o uso da pesquisa científica como caminho exitoso para o fomento da inovação, da resolução de demandas informacionais e, de modo mais pontual, de questões sobre preservação digital em arquivos confiáveis. Isto posto, é possível afirmar que a aplicação da pesquisa no âmbito do projeto em questão contribuiu de forma vanguardista com a produção de conhecimento científico e tecnológico, e com a garantia de preservação da memória institucional do TJDFT. Atendeu, também, a uma demanda específica da sociedade, que é a garantia de acesso em longo prazo à sua memória.

No plano base do projeto de pesquisa sobre o RDC-Arq, um dos resultados esperados era o “mapeamento dos fluxos necessários para a implantação de tecnologia de preservação da Administração Pública Federal”. Após a finalização do projeto entre IBICT-TJDFT, os resultados obtidos têm despertado o interesse de outros órgãos da esfera pública, como é o caso o Arquivo Nacional, que buscou o IBICT recentemente para adaptar a metodologia de construção de barramento proposto no âmbito do TJDFT para o contexto e as especificações de sistemas daquela instituição. Outras instituições como o STJ, STF, TJMG e TRT4 também demonstraram interesse nos resultados de pesquisa ora obtidos.

O modelo proposto para o Hipátia segue diretrizes modernas e permite sua adaptação para diferentes contextos, conforme mostra a figura a seguir:

Figura 11 - Modelo Hipátia



Fonte: Braga (2020)

A construção do Hipátia de forma genérica foi um dos requisitos estabelecidos no escopo da pesquisa, possibilitando adaptar sua estrutura a outras demandas existentes na administração pública. Desse modo, vale destacar que as soluções tecnológicas e os procedimentos metodológicos adotados pelo Instituto Brasileiro de Informação em Ciência e Tecnologia se mostraram exitosos e prontos para atender demandas e necessidades de forma abrangente, uma vez que podem ser replicados e adaptados a necessidades de outros Tribunais e instituições da Administração Pública brasileira.

Por fim, destaca-se que a opção do TJDF por pautar a solução de um questão institucional por meio do apoio à pesquisa científica contribuiu para o beneficiamento da sociedade de modo mais geral, uma vez que ela poderá se beneficiar dos avanços ora obtidos. A parceria entre instituições públicas por meio do incentivo à pesquisa é um modelo ainda pouco explorado e que possui grande apelo na construção de soluções demandadas pela sociedade.

REFERÊNCIAS

BRAGA, T. **A pesquisa científica e a criação de arquivos digitais confiáveis**. Brasília: [s. n.], 3 set. 2020. Disponível em: <https://www.YouTube.com/watch?v=kWKDlwQbk1k>. Acesso em: 3 out. 2020.

CONSELHO NACIONAL DE ARQUIVOS (CONARQ). Câmara Técnica de Documentos Eletrônicos. **Diretrizes para a implementação de repositórios arquivísticos digitais confiáveis**: RDC-Arq. Rio de Janeiro: Conarq, 2017. Disponível em: http://www.conarq.gov.br/images/publicacoes_textos/diretrizes_rdc_arq.pdf. Acesso em: 31 ago. 2020.

CUNHA, M. B.; CAVALCANTI, C. R. O. **Dicionário de Biblioteconomia e Arquivologia**. Brasília: Briquet de Lemos, 2008. 451 p.

INSTITUTO BRASILEIRO DE INFORMAÇÃO EM CIÊNCIA E TECNOLOGIA (IBICT). **Estudo para Implantação de Repositório Arquivístico Digital Confiável**: RDC-Arq: Proposta de projeto básico. Brasília: IBICT, jul. 2018. Disponível em: <https://drive.google.com/file/d/1AbRlxjgvEjcXsNjIhkgOj3tnDsJmahdJ/view?usp=sharing> e Redmine. Acesso em: 28 ago. 2020.

INSTITUTO BRASILEIRO DE INFORMAÇÃO EM CIÊNCIA E TECNOLOGIA (IBICT). Relatório parcial relativo à **Meta 3**: projeto Estudo para Implantação de Repositório Arquivístico Digital Confiável. Brasília: CGTI, IBICT, dez. 2019. Disponível em: Redmine.

ANEXO A – MANUAIS

FORMATOS ACEITOS PELO PJE

Tatiana Canelhas – 27 maio 2019

PLANO DE PRESERVAÇÃO

Ações de preservação como identificação, caracterização, detalhe do evento, extração de pacote, normalização, transcrição, validação e verificação são gerenciadas na guia “Plano de Preservação” do Archivematica. Nesse software, os formatos, regras e comandos padrões são instalados e acessados através dessa guia.

- **Identificação:** processo de analisar informações fornecidas sobre um arquivo para derivar seu formato (usado na normalização). No Archivematica 1.10, existem três ferramentas de identificação de arquivos:
 - *File Extension* é um script simples que identifica o formato dos arquivos pela sua extensão.
 - O *FIDO*, desenvolvido e mantido pela Open Preservation Foundation, identifica os formatos dos arquivos por sua assinatura e os conectam a um PRONOM ID.
 - *Siegfried*, desenvolvido e mantido por *Richard Lehane*, também identifica os formatos dos arquivos por sua assinatura e os conectam a um ID PRONOM. Esse programa é a ferramenta padrão para identificação de arquivos no Archivematica.¹
- **Caracterização:** processo de produção de metadados técnicos para um objeto. A caracterização do Archivematica tem como objetivo documentar as propriedades significativas do objeto e extrair os metadados técnicos contidos no objeto.
 - A ferramenta de caracterização padrão é o FITS
- **Detalhes do Evento:** estes asseguram que as informações sobre o software que está executando um comando sejam gravadas no arquivo METS com a propriedade “event detail”.
 - Várias ferramentas diferentes são usadas para gravar esses dados no arquivo METS, dependendo do evento. Por exemplo, se o FFmpeg estiver sendo usado para caracterizar um arquivo, o FFmpeg poderá gravar os detalhes do evento no METS. Em outras situações, o comando “echo” de linha de comando é usado para executar esta função.
- **Extração:** usado para extrair o conteúdo de um pacote, como um arquivo ZIP ou uma imagem de disco (ISO). O Archivematica vem com três ferramentas de extração:
 - *7zip*: usado na maioria dos formatos compatíveis com 7zip.
 - *unrar-free*: usado para formatos RAR.
 - *Sleuthkit*: usado para muitos formatos de imagem de disco (ISO).
- **Normalização:** processo de transformar um arquivo de um determinado formato em outro para acesso ou preservação. Várias ferramentas diferentes são usadas para essa tarefa no Archivematica, dependendo do formato do arquivo.
 - Para arquivos de imagem: usa-se o *ImageMagick*, *convert* e o *Inkscape*.
 - Para arquivos audiovisuais: usa-se o *ffmpeg*.
 - Para transformar arquivos em PDF: usa-se *Ghostscript* e *ps2pdf*.
 - Existe ainda umscript específico do Archivematica que transcodifica um *maildir* para o formato mbox, usado para e-mails.
- **Transcrição:** A transcrição é executada nos arquivos de imagem para analisar se eles contêm ou não texto e extrai-lo, salvando o conteúdo em um arquivo de texto (txt).
 - Por padrão, o Archivematica usa a ferramenta de transcrição *Tesseract*, o qual é uma ferramenta de OCR de código aberto.
- **Validação:** A validação de formato garante que os arquivos sejam válidos (bem formados) e compatíveis com todas as especificações de formato relevantes, e pode ser feita com relação a uma política personalizada aplicada ao formato.
 - O Archivematica inclui duas ferramentas de validação: *JHOVE* e *MediaConch*.

¹ Somente um comando pode ser ativado por vez - o Archivematica desativará automaticamente um comando se um novo estiver ativado.

- Verificação: executada automaticamente após o de normalização². O Archivematica executará dois comandos:
 - para verificar se existe um arquivo e (ou seja, o diretório não está vazio);
 - para verificar se o arquivo existe e tem mais de 0 bytes de tamanho.

As políticas de formato padrão do Archivematica são mantidas pela empresa Artefactual, a qual é responsável pelo software, e mudarão à medida que evoluírem os padrões, práticas e ferramentas da comunidade. A atualização dessas políticas só poderá ser feita se fizer o *upgrade* do software sempre que uma nova versão é lançada.

REGISTRO DE POLÍTICA DE FORMATO - FPR

O Archivematica gerencia políticas de formato local e externamente por meio de um Registro de Política de Formato (*Format Policy Registry - FPR*). Essa política indica as ações, ferramentas e configurações a serem aplicadas a um arquivo de um formato específico.

O FPR contém *scripts* configuráveis pelo usuário para todas as ações de preservação, dependendo do formato. Ele também contém uma lista de formatos reconhecidos pelo Archivematica e o relacionamento entre esses formatos e os *scripts*. É possível seguir as regras do FPR de acordo com a instalação padrão do Archivematica, ou atualizá-lo localmente, para adicionar novos formatos ou personalizar o comportamento do Archivematica.

As políticas de formato mudam à medida que os padrões, práticas e ferramentas da comunidade evoluem³. É importante que as instituições estabeleçam políticas e práticas locais que incluam o monitoramento do ambiente de preservação digital para ajudar a informar as regras de normalização do formato ao longo do tempo, à medida que padrões e ferramentas evoluem.⁴

FORMATOS

Um formato é uma maneira padronizada pela qual as informações são codificadas para armazenamento em uma mídia de armazenamento digital. O Archivematica reconhece centenas de formatos de arquivo extraído informações de formato do PRONOM, um registro técnico de formatos de arquivo mantido pelos Arquivos Nacionais do Reino Unido. Na versão 1.10.1 de 2019, estão cadastrados 1.094 formatos.

Figura 1 - Cada entrada é um registro representando uma ou mais versões de formatos relacionados, o qual cada versão também é um registro. Por exemplo, o registro de formato para Graphics Interchange Format (GIF) é composto por registros de versão de formato para várias versões do formato GIF, como GIF 1987a, 1989a e GIF genérico.

Format Graphics Interchange Format							
Specific versions							
Description	Version	Pronom ID	Access format?	Preservation format?	Enabled	Actions	
1987a	1987a	fmt/3	No	No	True	View Replace Disable	
1989a	1989a	fmt/4	No	No	True	View Replace Disable	
Generic gif			No	No	True	View Replace Disable	

Fonte 1 - archivematica.org

² A verificação é executada na saída da normalização, não no arquivo original.

³ Essa atualização no sistema só é possível ao fazer o upgrade de versão do Archivematica.

⁴ É recomendável documentar as políticas e práticas de acordo com o padrão TRAC para auditar Repositórios Digitais Confiáveis (ISO 16363: 2012).

Grupo de Formatos

Refere-se a um agrupamento conveniente de formatos de arquivos relacionados que compartilham propriedades comuns. Cada formato pode pertencer a um grupo de formatos. A seguinte lista de grupos vem pré-cadastrada no FPR do Archivematica:

- Audio
- Binary (Data)
- Binary (Executable)
- CAD
- Data Visualization
- Database
- Dataset
- Desktop Publishing
- Disk Image
- Email
- Flash
- Font
- GIS
- Image (Raster)
- Image (Vector)
- Package
- Portable Document Format
- Presentation
- Spreadsheet
- Statistics
- Text (Markup)
- Text (Plain)
- Text (Source Code)
- Text (Structured)
- Unknown
- Video
- Word Processing

Formatos do PJE

O programa Processo de Justiça Eletrônico (PJE) permite o upload de documentos anexados ao documento principal. Os formatos que esse sistema aceita estão na tabela a seguir:

Tipo de arquivo	Extensão	Tamanho máximo
Documento	Pdf	10 Mb
Imagem	png, jpeg	3 Mb
Vídeo	Mpeg	5 Mb
	ogg, mp4, quicktime	10 Mb
	Wmv	50 Mb
Áudio	mpeg, mp3	5 Mb
	ogg, mp4, vorbis	10 Mb

Tabela 1 - Tabela 1 - Sítio do TJDFt - Pergunta 20⁵

⁵ <https://www.tjdft.jus.br/informacoes/perguntas-mais-frequentes/processo-judicial-eletronico-pje> e https://www.pje.jus.br/wiki/index.php/Manual_do_Advogado#Anexos

IDENTIFICAÇÃO

Para a identificação do formato dos arquivos transferidos para o Archivematica, o software usa como padrão a ferramenta *Siegfried*. O Archivematica também pode ser configurado para pular esse microserviço se necessário (configuração usada quando se já enviam os arquivos normalizados na transferência, por exemplo).

Figura 2 - Ao retornar uma identificação, a ferramenta deve imprimir uma única linha contendo apenas o identificador e sair 0 (exit code = 0). Qualquer mensagem informativa, de diagnóstico e de erro pode ser impressa no stderr, onde estará visível para os usuários do Archivematica que monitoram os resultados da ferramenta (ao clicar na engrenagem do microserviço). Em caso de falha, a ferramenta deve sair diferente de zero.

The screenshot shows the Archivematica web interface. At the top is a navigation bar with links: Transfer, Backlog, Appraisal, Ingest, Archival storage, Preservation planning, Access, Administration, and a test dropdown. Below this is a breadcrumb trail: Transfer e57a2c1e-90d1-4220-a660-cbcbed6a928 / Identify file format / Tasks. The main content area displays details for a task with ID d535cbf4-3dc6-4625-82cb-e3e2aef8233e. It lists metadata for a file: File UUID (1d1af737-6945-4b5d-9c72-e14227352af0), File name (DJ2_2019.pdf), Client, and Exit code (0, highlighted in a green box). It also shows timing information: Start time (12/9/2019, 11:47:38 AM), End time (12/9/2019, 11:47:40 AM), Created time (12/9/2019, 11:47:37 AM), and Duration (2 seconds(s)). A section titled 'Module identifyFileFormat_v0.0' contains a command string: "True" "/var/archivematica/sharedDirectory/watchedDirectories/workFlowDecisions/selectFormatIDToolTransfer/abc-e57a2c1e-90d1-4220-a660-cbcbed6a928/objects/DJ2_2019.pdf" "1d1af737-6945-4b5d-9c72-e14227352af0" --disable-reidentify. Below this, the 'Standard streams' section shows the 'Standard output (stdout)' with a log of the identification process using Siegfried 1.7.10, including UUIDs, file paths, and the final result: 'objects/DJ2_2019.pdf identified as a Acrobat PDF'.

Fonte 2 - archivematica.org

O comando de identificação é executado uma vez para cada arquivo em uma transferência.

NORMALIZAÇÃO

A principal estratégia de preservação do Archivematica é normalizar arquivos após a ingestão. Normalização é o processo de migrar um arquivo de um determinado formato em outro, para uma finalidade declarada (como preservação ou acesso).

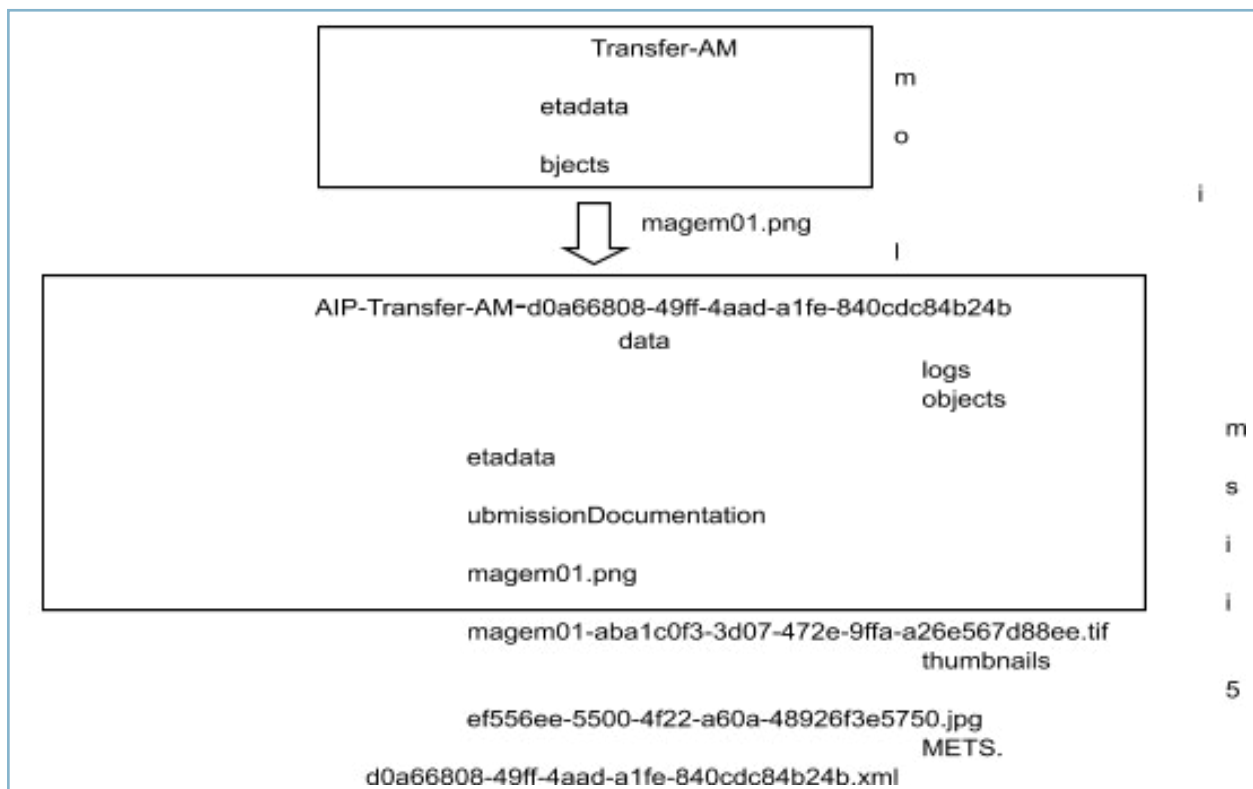
As cópias de preservação são adicionadas ao AIP (*Archival Information Package*)⁶ e as cópias de acesso são usadas para gerar um DIP (*Dissemination Information Package*)⁷ para ser enviada para o sistema de acesso (por exemplo, o programa *Access to Memory, AtoM*). Os arquivos originais são sempre mantidos para permitir diferentes ações de preservação no futuro, como normalização para diferentes formatos de arquivo ou emulação⁸.

6 Pacote de arquivamento de informação.

7 Pacote de disseminação de informação

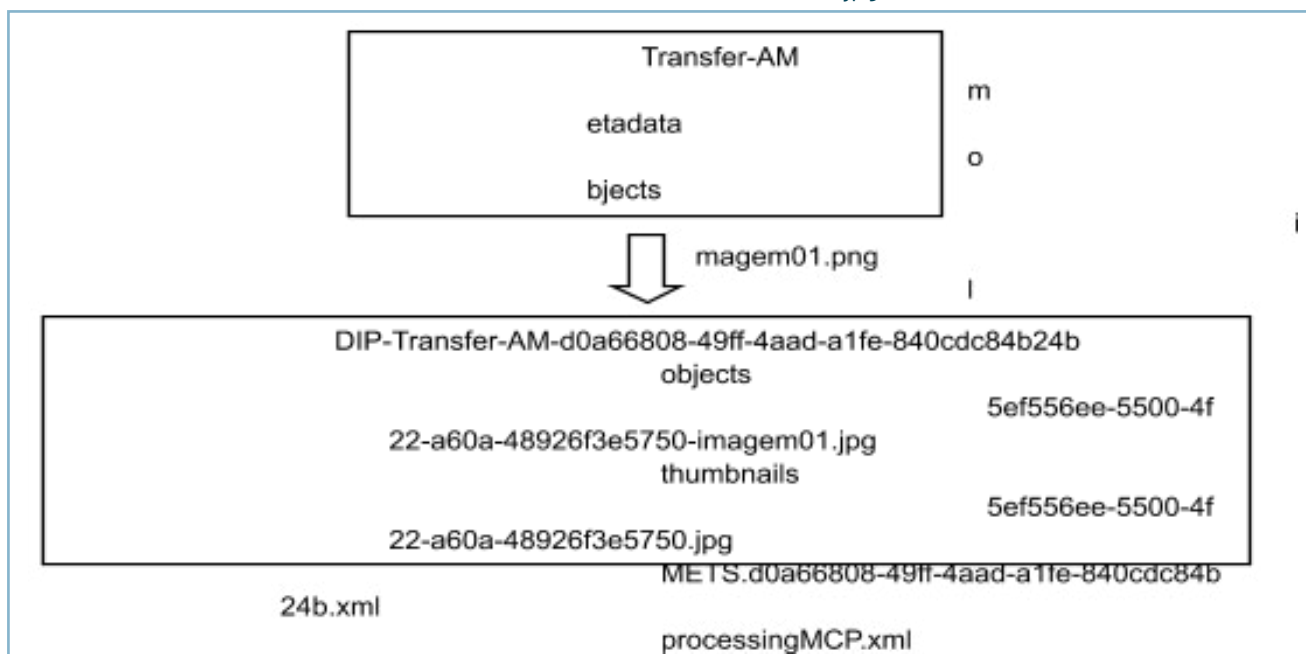
8 Obs.: O Archivematica mantém o formato original de todos os arquivos ingeridos para suportar estratégias de preservação de migração e emulação.

Figura 3 - Transferência da “imagem 01.png” para o Archivemática. Após normalizar, ela se mantém no pacote AIP, e cria derivada de preservação no formato TIFF.



Fonte 3 - Fonte própria.

Figura 4 - Transferência da “imagem 01.png” para o Archivemática. Após normalizar, ela se mantém no pacote DIP, e cria derivada de acesso no formato jpg.



Fonte 4 - Fonte própria.

Note que, conforme as figuras acima, ao normalizar uma imagem em formato PNG, o archivemática migrou para TIF sua derivada de preservação e para JPG sua derivada de acesso, e usou o software *ImageMagick* para fazer essa conversão. Essa escolha de formatos e o uso do software de normalização está de acordo com as políticas de formato do Archivemática, exemplificada na tabela abaixo.

Tabela 2 - As Políticas de Formato indicam qual ferramenta executar ao normalizar para um determinado objetivo (acesso, preservação) quando uma Ferramenta de Identificação de Arquivos específica identifica um Formato de Arquivo específico.

Media type	File formats	Preservation format(s)	Access format(s)	Normalization too
Audio	AC3, AIFF, MP3, WAV, WMA	WAVE (LPCM)	MP3	FFmpeg
Email	PST	MBOX	MBOX	readpst
Email	Maildir	Original format	MBOX	md2mb.py
Office Open XML	DOCX, PPTX, XLSX	Original format	Original format	Tool search in progress
Plain text	TXT	Original format	Original format	None
Portable Document Format	PDF	PDF/A	Original format	Ghostscript
Presentation files	PPT	Original format	PDF	Tool search in progress
Raster images	BMP, GIF, JPG, JP2, PCT, PNG, PSD, TIFF, TGA	Uncompressed TIFF	JPEG	ImageMagick
Raw camera files/ Digital Negative format	3FR, ARW, CR2, CRW, DCR, DNG, ERF, KDC, MRW, NEF, ORF, PEF, RAF, RAW, X3F	Original format	JPEG	ImageMagick/ UFRaw
Spreadsheets	XLS	Original format	Original format	None
Vector images	AI, EPS, SVG	SVG	PDF	Inkscape
Video	AVI, FLV, MOV, MPEG-1, MPEG-2, MPEG-4, SWF, WMV	FFV1/LPCM in MKV	MP4	FFmpeg
Word processing files	DOC, WPD, RTF	Original format	Original format	Tool search in progress

Fonte 5 - https://wiki.archivematica.org/Format_policies#Normalization

FORMATOS DO PJE NO ARCHIVEMATICA

Normalização dos Formatos do PJe

No ANEXO I, encontram-se os formatos normalizados pelo Archivematica para preservação, de acordo com os formatos da tabela acima, e no Anexo II, temos os formatos normalizados para acesso. O quadro de desse microserviço para os formatos aceitos pelo PJe, ficaria então, conforme tabela abaixo:

Media Type	Formato	Preservação	Acesso
Portable Document Format	Portable Document Format - PDF	PDF/A-1b	Formato Original
Áudio	MPEG Audio	É tratado como vídeo	É tratado como vídeo
	MPEG 1/2 Audio Layer 3 - MP3	WAV	MP3
	OGG	Formato Original	Formato Original
	MP4 Apple Lossless Audio Codec	É tratado como vídeo	É tratado como vídeo
	Vorbis-OGG	Formato Original	Formato Original
Imagem Rasterizada	Portable Network Graphics - PNG	TIFF Descompactado	JPEG (qualidade: 60)
	JPEG File Interchange Format	TIFF Descompactado	JPEG (qualidade: 60)

Media Type	Formato	Preservação	Acesso
Video	MPEG Video -codec libx264 -px: yuv420p	MKV	Formato Original
	OGG	Formato Original	Formato Original
	MPEG-4 Media File	MKV	Formato Original
	Quicktime - MOV	Formato Original	Formato Original
	Windows Media Video - WMV	MKV	MP4

Tamanhos Após Normalização

Abaixo, segue uma relação de tamanhos ao transferir para o Archivematica arquivos com os formatos aceitos pelo PJe.

Tipo	Tamanho Original (KB)	Derivada Preservação (KB)	Formato Acesso	Derivada Acesso (KB)	AIP.7z (KB)	DIP.rar (KB)
Audio	1063	4673	MP3	638	4493	800
	1064	1064	OGG	1064	1094	1190
	697	697	OGG	697	720	830
Imagem	87	598	JPEG	42	622	230
	984	3051	JPEG	49	2312	250
Vídeo	446	446	MOV	446	457	550
	3042	36555	MP4	3042	33214	3250
	1282	4684	MPEG	1282	5580	1140
	1695	1695	OGG	1695	1733	1910
	1567	24796	MP4	1032	18746	1180

ANEXO I

Formatos atuais de preservação no FPR (atualizado em agosto de 2015)⁹, de acordo com os formatos aceitos pelo PJE:

FPR group	File Format	Version	PRONOM ID	Description
Audio	WAV		fmt/6	Waveform Audio
		0	fmt/1	Broadcast WAVE 0
				Generic WAV
		1	fmt/2	Broadcast WAVE 1
		2	fmt/527	Broadcast WAVE 2
		2	x-fmt/397	Exchangeable Image File Format (Audio) 2.0
		2.1	x-fmt/389	Exchangeable Image File Format (Audio) 2.1
		2.2	x-fmt/396	Exchangeable Image File Format (Audio) 2.2
			fmt/141	Waveform Audio (PCMWAVEFORMAT)
			fmt/142	Waveform Audio (WAVEFORMATEX)

⁹ https://wiki.archivematica.org/Preservation_formats

FPR group	File Format	Version	PRONOM ID	Description
Image (Raster)	Portable Network Graphics			Generic PNG
		1.1	fmt/12	PNG 1.1
		1	fmt/11	PNG 1.0
		1.2	fmt/13	PNG 1.2
	Tagged Image File Format	1	fmt/436	Digital Negative Format (DNG) 1.0
		1.1	fmt/152	Digital Negative Format (DNG) 1.1
		1.2	fmt/437	Digital Negative Format (DNG) 1.2
		1.3	fmt/438	Digital Negative Format (DNG) 1.3
Portable Document Format	PDF	1.5	fmt/19	Acrobat PDF 1.5

Os formatos de acesso são definidos para os quais exista uma ampla disponibilidade de aplicativos capazes de abrir o formato do arquivo em questão.

O cálculo da volumetria encontra-se no Anexo III deste documento.

ANEXO II

Formatos atuais de acesso no FPR (atualizado em agosto de 2015)¹⁰, de acordo com os formatos aceitos pelo PJE:

FPR group	File Format	Version	PRONOM ID	Description
Image (Raster)	JPEG	\N		Generic JPEG
		1.01	fmt/43	JPEG 1.01
	Portable Network Graphics			Generic PNG
Portable Document Format	PDF			Generic PDF
		1	fmt/14	Acrobat PDF 1.0
		1.1	fmt/15	Acrobat PDF 1.1
		1.2	fmt/16	Acrobat PDF 1.2
		1.3	fmt/17	Acrobat PDF 1.3
		1.4	fmt/18	Acrobat PDF 1.4
		1.6	fmt/20	Acrobat PDF 1.6
		1.7	fmt/276	Acrobat PDF 1.7
		1a	fmt/95	Acrobat PDF/A 1a
		1b	fmt/354	Acrobat PDF/A 1b
		2a	fmt/476	Acrobat PDF/A 2a
		2b	fmt/477	Acrobat PDF/A 2b
		2u	fmt/478	Acrobat PDF/A 2u
		3a	fmt/479	Acrobat PDF/A 3a

¹⁰ https://wiki.archivematica.org/Access_formats

FPR group	File Format	Version	PRONOM ID	Description
Portable Document Format	PDF	3b	fmt/480	Acrobat PDF/A 3b
		3u	fmt/481	Acrobat PDF/A 3u
		E-1	fmt/493	Acrobat PDF for Engineering PDF/E-1
			fmt/157	Acrobat PDF/X Exchange 1a:2001
			fmt/144	Acrobat PDF/X Exchange 1:1999
			fmt/145	Acrobat PDF/X Exchange 1:2001
			fmt/146	Acrobat PDF/X Exchange 1a:2003
			fmt/147	Acrobat PDF/X Exchange 2:2003
			fmt/158	Acrobat PDF/X Exchange 3:2002
			fmt/488	Acrobat PDF/X Exchange PDF/X-4
			fmt/489	Acrobat PDF/X Exchange PDF/X-4p
			fmt/490	Acrobat PDF/X Exchange PDF/X-5g
			fmt/492	Acrobat PDF/X Exchange PDF/X-5n
			fmt/491	Acrobat PDF/X Exchange PDF/X-5pg
			fmt/148	Acrobat PDF/X Exchange 3:2003

PORTABLE DOCUMENT FORMAT/ARCHIVE – PDF/A

Tatiana Canelhas – maio 2019

Alguns requisitos devem ser pensados quando estamos tratando documentos digitais para a preservação e o acesso, para não comprometer sua autenticidade, seu acesso e seu uso ao longo do tempo. Um deles é definição de formatos de arquivo, com independência de sistemas operacionais e hardware (BRASIL, 2016). Nesse sentido, o PDF/A foi adotado por algumas instituições como o formato ideal para arquivamento a longo prazo.

O PDF/A (“A” para “Arquivo”) é uma variante do PDF que foi projetada para preservação de documentos a longo prazo. Aprovado pela ISO – International Organization for Standardization – em

1996 com base no PDF 1.4, o PDF/A é um subconjunto de PDF que elimina 2005 (ISO 19005- 1:2005),

certos riscos para a futura reprodutibilidade do conteúdo.

“A especificação do PDF/A como norma ISO torna o formato um padrão aberto, tornando-se de amplo uso e facilitando a criação de aplicativos pelos desenvolvedores. O formato PDF/A atende à produção dos documentos textuais e imagéticos paginados, permitindo manter sua forma fixa e conteúdo estável”. (BRASIL, 2016)

Diferentemente do PDF normal, o PDF/A exige que todos os recursos necessários para renderizar com precisão o(s) documento(s) que esteja(m) contido(s) no arquivo PDF/A, incluindo fontes, perfis de cores, imagens e assim por diante.

O PDF/A proíbe o conteúdo dinâmico para garantir que o usuário veja exatamente o mesmo conteúdo hoje e nos próximos anos – o objetivo dessa especificação foi criar um formato que possa ser renderizado de maneira confiável em qualquer sistema quando aberto com um visualizador

¹No Brasil a ABNT (Associação Brasileira de Normas Técnicas) realizou a tradução e a adoção do formato, com a norma ABNT NBR ISO 19005-1 em 2009 denominada Gerenciamento de documentos – Formato eletrônico de arquivo de documento para preservação de longo prazo – Parte 1.

Nota: O PDF em si não atendeu a esse critério porque os documentos em PDF podem conter elementos que não são renderizados de maneira confiável, pois sua aparência pode mudar com base no visualizador, no sistema operacional do host ou no estado do próprio PDF.

Alguns exemplos incluem documentos PDF que não incorporam suas fontes e, em vez disso, esperam que os sistemas de visualização os tenham, documentos que usem cores dependentes de dispositivo, documentos cifrados e documentos que contenham conteúdo dinâmico, como JavaScript e 3D. O PDF/A foi desenvolvido para resolver esses problemas.

O formato PDF/A possui a vantagem de ser um padrão aberto. Para o CONARQ (BRASIL, 2011, p. 35) “a adoção de formatos digitais abertos configura-se, adicionalmente, como medida de preservação recomendável e necessária.”

Há muitas razões para usar o PDF/A para fins de arquivamento, mas as duas principais são suas vantagens sobre outros formatos eletrônicos, particularmente sobre o formato TIFF, e sua aceitação no setor.

Um formato amplamente utilizado para arquivamento digital é o TIFF, de imagem rasterizada que promete a mesma aparência visual garantida do documento que o PDF/A. No entanto, por se tratar de um formato de imagem, falta a capacidade de incluir conteúdo vetorial, como formas, gradientes e fontes vetoriais disponíveis em PDF. O conteúdo vetorial não apenas descreve com mais precisão o documento original, como também ocupa menos espaço em disco, o que pode ser significativo ao arquivar muitos documentos. Esse recurso é especialmente relevante para documentos nato-digitais, que agora são muito mais comuns do que quando o TIFF foi adotado pela primeira vez. Outra vantagem do PDF/A sobre o TIFF é que texto Unicode pode ser incluído para garantir que o texto seja extraível e pesquisável, tornando um grande arquivo digital potencialmente muito mais útil. Por fim, o formato PDF/A pode acomodar assinaturas digitais incorporadas, oferecendo aos usuários uma maneira de verificar se o PDF foi alterado bem como identificar com segurança seu autor ou emitente, o que pode ser importante para a admissibilidade legal de um documento

O PDF/A alcançou um alto nível de aceitação da comunidade. Quando o formato foi publicado em 2005, um grupo de empresas na Europa formou o *PDF/A Competence Center* para elevar o perfil do formato e promover seus benefícios para a indústria e o governo. Desde então, muitas instituições, especialmente na Europa, determinaram o PDF/A como o formato de arquivo necessário para arquivamento.

As características do PDF/A têm incentivado o seu crescente uso pelas organizações, governos e pessoas. No Brasil, o PDF/A é um dos formatos de arquivo adotados pelo governo federal, previsto nos Padrões de Interoperabilidade de Governo Eletrônico (e-PING). (BRASIL, 2016).

Várias agências dos Estados Unidos também aceitam PDF/A como um formato, como National Archives and Records Administration (NARA) e PACER. Em segundo lugar, o fato de um arquivo PDF/A ser um arquivo PDF significa que os visualizadores gratuitos estão amplamente disponíveis em praticamente todos os dispositivos de computação.

Características de um PDF/A

O PDF/A é um subconjunto de PDF, obtido excluindo algumas características para arquivamento de longo prazo. Diferentemente do PDF normal, o PDF/A exige que todo o necessário para renderizar com precisão o documento esteja contido no arquivo, incluindo fontes, perfis de cores, imagens e assim por diante.

Esse formato, voltado para o arquivamento do objeto digital, proíbe o conteúdo dinâmico para garantir que o usuário veja exatamente o mesmo conteúdo a longo prazo. Abaixo seguem outras características do PDF/A:

2 A primeira versão da especificação TIFF foi publicada pela Aldus Corporation em 1986, após dois principais lançamentos anteriores (em meados da década de 80). No começo, o TIFF era apenas um formato de imagem binário, porque era tudo o que os scanners de mesa podiam suportar. Com o tempo, o TIFF passou a acomodar imagens em escala de cinza e imagens coloridas (TIFF, s.d.).

- Pode incluir links, mesmo que virem obsoletos ao longo do tempo.
- Proíbe objetos de áudio e vídeo.
- Pode ser um pouco maior que o arquivo PDF original do qual foi criado (desde que não use resoluções de imagem ou métodos de compactação diferentes).
 - As fontes são incorporadas em um arquivo PDF/A (que também costuma ser o caso em arquivos PDF “normais”) e mais informações são armazenadas nos metadados.
 - Alguns perfis de cores podem, em certos casos, levar o PDF a um tamanho de arquivo muito maior, mas isso é raro e depende muito do caso específico.
- Pode incluir arquivos CAD⁴ como objetos incorporados.
- A compactação de arquivos ZIP é permitida e as imagens podem ser compactadas usando a compactação JPEG.
 - A compactação LZW, no entanto, é proibida.
- Proíbe o uso de criptografia no arquivo.
 - Para fins de controle de acesso, é necessário que esse controle seja feito nos locais em que os arquivos PDF/A estão armazenados.
- Os marcadores (bookmarks) são permitidos em PDF/A.
- Acessibilidade e PDF/A andam de mãos dadas. Os arquivos PDF/A também podem estar em conformidade com PDF/UA, o padrão ISO para a tecnologia PDF acessível;
- A maioria das formas de anotações em PDF é permitida em PDF/A.
 - Existem certas restrições, por exemplo, qualquer anotação deve estar visível.
- Os arquivos PDF/A podem conter assinatura eletrônica.

O uso do PDF/A por profissionais da informação ajuda a garantir o acesso futuro aos arquivos, independentemente de plataforma, sistema operacional ou mídia ao longo do tempo. Ações como a adoção desse formato visam amenizar o impacto que as mudanças tecnológicas causam na acessibilidade aos documentos eletrônicos, e conhecer os tipos de PDF/A ajuda a melhorar o valor dos documentos para fins específicos de visualização, compartilhamento, impressão ou arquivamento.

Níveis de Conformidade Diferentes de PDF/A

Os níveis de conformidade controlam os requisitos de “acessibilidade” de um arquivo que afetam a capacidade de máquinas e pessoas de entender o conteúdo. É o conjunto identificado de restrições e requisitos aos quais os arquivos e leitores devem cumprir.

Os documentos criados ou convertidos para o PDF/A podem estar em nível de conformidade “a”, “b” ou “u”. O nível de conformidade “a” é atribuído ao documento que apresenta todos os elementos que a norma considera. O nível de conformidade “b” está relacionado a aparência visual do documento eletrônico e, sendo assim, também a aspectos que garantem a confiabilidade e a legitimidade do documento. O nível de conformidade “u” abrange os requisitos de aparência visual de documentos eletrônicos, com a condição de que todo o texto no documento tenha equivalentes Unicode.

Nível b (Básico)

PDF/A-1b, PDF/A-2b, PDF/A-3b

A conformidade no nível “b” exige apenas que os documentos estejam em conformidade com as diretrizes para uma visualização confiável e, portanto, é o nível mais fácil de alcançar. A partir da especificação ISO 19005-3:2012(en):

“Conformidade nível B: [...] abrange os requisitos desta parte da ISO 19005 em relação à aparência visual de documentos eletrônicos, mas nem suas propriedades estruturais ou semânticas nem o requisito de que todo o texto tenha equivalentes Unicode.” (ISO 19005-3:2012(en), n.d.)

³ <https://www.pacer.gov/announcements/general/pdfa.html>

⁴ Os arquivos CAD (Computer-Aided Design) podem conter desenhos em 2D ou 3D. Eles são gerados por programas de software CAD, que podem ser usados para criar modelos ou planos de arquitetura (FileInfo, s.d.).

Nível a (Acessível)

PDF/A-1a, PDF/A-2ab, PDF/A-3a

A conformidade “acessível” é um superconjunto da conformidade de nível “b”. Ele adiciona requisitos para informações destinadas a preservar a estrutura lógica de um documento, o conteúdo semântico e a ordem natural de leitura.

Em outras palavras, a conformidade em nível “a” não apenas garante que os documentos terão a mesma aparência no futuro, mas também ajuda as máquinas e as pessoas a entenderem e redirecionarem melhor seu conteúdo. Um PDF/A válido de nível superior terá texto que pode ser pesquisado e copiado com confiabilidade, e conteúdo mais acessível a tecnologias como leitores de tela para cegos.

Uma lista de requisitos de nível “a” é a seguinte:

- O conteúdo deve ser marcado (tagged) com uma árvore de estrutura hierárquica, o que significa que elementos como ordem de leitura, figuras e tabelas são explicitamente identificados por meio de metadados.
- O idioma natural do documento deve ser identificado.
- Imagens e símbolos devem ter texto descritivo alternativo.
- O arquivo deve incluir mapeamentos de caracteres para Unicode para pesquisa e cópia confiáveis.

Nota: nenhum desses requisitos alterará a aparência visual de um documento.

Nível u (Unicode)

PDF/A-2u, PDF/A-3u

Como no nível “a”, a conformidade no nível “u” requer mapeamento de caracteres para Unicode. No entanto, descarta os requisitos de nível geral, incluindo estrutura lógica incorporada (ou seja, tags e uma árvore de estrutura), conforme especificado na seção 6.7 da ISO 19005-2 – PDF 1.7 (Anexo I). Portanto, um PDF/A que atenda à conformidade no nível “u” terá texto que pode ser pesquisado e copiado com confiabilidade, mas a ordem de leitura não será garantida. A partir da especificação ISO 19005-3:2012(en):

“Nível de conformidade que abrange os requisitos desta parte da ISO 19005 em relação à aparência visual de documentos eletrônicos, com o requisito de que todo o texto no documento tenha equivalentes Unicode.” (ISO 19005-3:2012(en), n.d.)

PDF/A-1, PDF/A-2 and PDF/A-3

O PDF/A vem em algumas variantes, criadas pela combinação de diferentes padrões ou níveis de conformidade, onde cada um define a variedade de recursos disponíveis e tecnologias de compactação de imagem que ajudam na preservação do conteúdo de um arquivo.

Existem diversas versões de PDF/A:

Tabela 1 – Tipos de PDF/A

PDF/A-1	PDF/A-2	PDF/A-3
PDF/A-1a	PDF/A-2a	PDF/A-3a
PDF/A-1b	PDF/A-2u	PDF/A-3u
	PDF/A-2b	PDF/A-3b

Fonte – Autoria própria

Temos nos Anexo II e Anexo III, quadros comparativos do PDF tipos: PDF/A-1, PDF/A-2 e PDF/A-3.

PDF/A-1

A especificação PDF/A-1 (também conhecida por PDF/A original) é a mais restritiva dos padrões PDF/A. Como foi lançado antes do PDF ser definido em um padrão ISO, ela se baseia no PDF 1.4 e as especificações são abertas, porém proprietárias, publicadas pela *Adobe Systems Inc.* em 2001.⁵

Os arquivos PDF/A-1 não permitem JPEG2000, anexos ou camadas. Apesar de o formato PDF 1.4 suportar transparência, o padrão PDF/A-1 não contempla essa funcionalidade.

- Recursos ausentes: JPEG2000, transparência, camadas e anexos
- Baseado no PDF 1.4
- Níveis de conformidade suportados:
 - "a" (Acessível) – PDF/A-1a
 - "b" (Básico) – PDF/A-1b

PDF/A-2 (ISO 19005-2:2011)

O padrão PDF/A-2 se tornou uma ISO em 2011 (ISO 19005-2), e é considerado o melhor formato padronizado para arquivar os documentos digitais. Baseado no PDF 1.7 (ISO 32000-1: 2008), o PDF/A-2 apresenta vários recursos indisponíveis no PDF 1.4, além de transparência. As adições⁶

incluem camadas, compactação de imagem aprimorada (JPEG2000 e JBIG2) e anexos – desde que esses anexos estejam no formato PDF/A.

O PDF/A-2 não torna obsoletos os arquivos PDF/A-1. Em vez disso, o padrão é compatível com versões anteriores: por exemplo, um arquivo PDF/A-1b válido deve passar na verificação do software configurado para validar o PDF/A-2b ou PDF/A-3b.

Por fim, o nível de conformidade "u" (Unicode) foi introduzido no PDF/A-2.

DIFERENÇAS ENTRE PDF/A-1 E PDF/A-2

Ao contrário do PDF/A-1, o PDF/A-2 é inteiramente baseado nas diretrizes da ISO, conforme publicado na norma ISO 32000-1:2008⁷.

Nota: Se o PDF/A-1 atender aos requisitos de arquivamento, não será necessário alterar para PDF/A-2.

MELHORIAS DO PDF/A-2 EM COMPARAÇÃO COM PDF/A-1:

- Armazena imagens em compactação sem perdas, graças ao formato de arquivo JPEG2000.
 - Como resultado, os documentos PDF com imagens são menores e, portanto, muito mais fáceis de carregar ou baixar.
- Graças ao Unicode, é garantido que se possa pesquisar o texto.
 - Unicode permite indexar o texto. Além disso, pode-se converter o texto de forma confiável de um documento PDF para, por exemplo, outro processador de texto.

⁵ Um arquivo JP2 é uma imagem de bitmap compactada que usa a codificação JPEG 2000 (JP2). Ele suporta qualquer profundidade de bits de cor e metadados de imagem e pode ser compactado com ou sem perdas. Os arquivos JP2 geralmente são usados para armazenar fotos digitais e imagens médicas, como tomografias e ressonâncias magnéticas. (FileInfo, s.d.)

⁶ Joint Bi-level Image Group - Imagem bitonal salva no formato Joint Bi-level Image Group 2 (JBIG2): armazene a imagem usando preto e branco com 1 bit por pixel; pode ser com ou sem perdas; permite maior compactação do que o algoritmo CCITT Fax Group 4; frequentemente usado para imagens e faxes digitalizados; substitui a especificação JBIG anterior. (FileInfo, s.d.)

*Imagens bitonais são imagens que usam apenas duas cores, preto e branco, para criar a imagem. As imagens bitonais são comumente encontradas na geração de imagens de documentos, porque as informações de texto são prontamente armazenadas no formato bitonal.

⁷ <https://www.iso.org/standard/51502.html>

- Várias camadas de PDF.
 - Várias camadas permitem que o autor mostre uma e a mesma página em um idioma diferente com apenas um toque de um botão. Aplicativos para desenhos do AutoCAD são possíveis.
- Suporte de várias fontes.
 - Como o PDF/A-2 suporta várias fontes, é fácil e seguro converter arquivos em PDF/A-2. Portanto, não precisa converter os documentos em uma fonte que é suportada primeiro.
- Transparência.
 - Devido à transparência interna, o texto marcado pode ser facilmente lido.
 - Agora também são possíveis sombras.
- São suportados até 381km × 381km como tamanhos de páginas.
 - Em alguns casos, o formato PDF/A é usado para arquivar mapas ou plantas. No PDF/A-1, isso foi limitado a tamanhos de 5,08m × 5,08m.
- Devido à redução da escala das imagens, muitos detalhes foram perdidos.
- Com o PDF/A-2, as imagens podem ser armazenadas com tamanhos de até 381km × 381km.
- Funcionalidade aprimorada de metadados.
- Conversão de padrões antigos como PDF/X-4⁸ e PDF/E-1⁹ para PDF/A-2.
- Compatibilidade com outros padrões ISO como PDF/X-4 e PDF-E-1.
- Suporte a assinaturas digitais padrão PAdES. Essa diferença é especialmente importante no contexto da administração pública brasileira porque o Instituto de Tecnologia da Informação (ITI), órgão normativo da Infraestrutura Pública de Chaves Brasileira (ICP-Brasil), prevê o uso de tal padrão para assinaturas digitais em arquivos PDF.¹⁰

PDF/A-3 (ISO 19005-3:2012)

O PDF/A-3¹¹ foi lançado em 2010 e é praticamente idêntico ao PDF/A-2. Uma diferença é que o PDF/A-3 permite qualquer tipo de arquivo como anexo, ao invés de apenas outros arquivos PDF.¹²

No entanto, um visualizador de PDF/A não precisa fazer nada extra com os arquivos anexados, além de garantir sua extração adequada. Portanto, o padrão não pode garantir se poder-se-á ler ou usar esses arquivos no futuro, solicitando aos arquivistas que expressem preocupações de que o PDF/A-3 possa contornar as restrições de arquivamento nos formatos permitidos.

Uma resposta à preocupação acima foi observar que um fluxo de trabalho cuidadosamente projetado, construído com considerações de arquivamento em mente, poderia explicar e alavancar os recursos do PDF/A-3. De fato, esse formato foi amplamente inspirado pelo vontade de ter um componente legível por máquina disponível, como dados binários proprietários ou XML, usados em situações em que formatos incorporados poderiam ser prescritos com cuidado.¹³ Ou seja, o PDF/A-3 permite reduções de custos no processamento, fornecendo as versões legíveis por humanos (PDF) e legíveis por máquina (XML) das mesmas informações em um único arquivo.

O PDF/A-3 facilita a coleta de conteúdo em vários formatos para fins de comunicação e referência. Ele possibilita o uso do PDF como veículo de entrega para um ou mais arquivos, de qualquer tipo, tornando-se um

8 Portable Document Format - Exchange PDF/X-4: Especificado na ISO 15930-7:2010 (uma revisão pequena da ISO 15930-7:2008). Para uma troca entre gerenciamento de cores e CMYK, based on PDF 1.6. (LOC, s.d.)

9 Portable Document Format para Engenharia PDF/E-1: ISO 24517-1. Document Management - Engineering document format using PDF. Part 1: Use of PDF 1.6 (PDF/E-1). (LOC, s.d.)

10 DOC-ICP 15.03 - Requisitos das Políticas de Assinatura Digital na ICP-BRASIL (ICP-Brasil, 2018).

11 Document management - Electronic document file format for long-term preservation - Part 3: Use of ISO 32000-1 with support for embedded files (PDF/A-3).

12 Os arquivos que atendem a esses requisitos são denominados arquivos "associados"; uma associação explícita deve ser feita entre cada arquivo incorporado e o PDF ou objeto ou estrutura que contém (por exemplo, imagem, página ou seção lógica) no PDF. Os tipos MIME devem ser fornecidos para arquivos associados. A especificação PDF/A-3 requer o uso de application/octet-stream se um tipo MIME mais específico não for conhecido.

13 Um exemplo é o padrão de faturamento eletrônico híbrido ZUGFeRD, publicado dois anos após a introdução do PDF/A-3, endossado pelo governo alemão e favorecido por muitas organizações e empresas da União Europeia. Aplicado sistematicamente, o PDF/A-3 ajuda a revolucionar o processamento da fatura, reduzindo os tempos de pagamento.

“arquivo zip”. Também, o PDF/A-3 foi projetado pelo comitê ISO para permitir que a avançada tecnologia de assinatura digital forneça prova de violação, estabeleça autoria, aplique certificações, entre outros.

- Recursos novos e permitidos: Anexos (qualquer tipo de arquivo)
- Baseado no PDF 1.7 (ISO 32000-1: 2008)
- Níveis de conformidade suportados:
 - “a” (Acessível) – PDF/A-3a
 - “b” (Básico) – PDF/A-3b
 - “u” (Unicode) – PDF/A-3u

PDF/A-4 (ISO 19005-4: 2019) – Ainda não incorporado pelo Archivematica

O PDF/A-4, a ser definido na ISO 19005-4¹⁴ e provisoriamente também chamado PDF/A-NEXT, será baseado no PDF 2.0. Embora o PDF/A-2 e o PDF/A-3 compreendam três níveis de conformidade diferentes, que tendem a confundir os usuários, o PDF/A-4 simplifica as coisas e elimina as designações de nível de conformidade para “documentos marcados” versus “não marcados” (um documento PDF/A-4 pode ou não conter tags). Diferentemente das partes anteriores do padrão, nenhum nível de conformidade dedicado é necessário para documentos PDF/A-4 marcados, eliminando assim a distinção no nível de conformidade a/b do PDF/A-1/2/3. Da mesma forma, um documento PDF/A-4 pode ou não conter anexos de arquivo. Os arquivos anexados devem estar em conformidade com PDF/A-1, PDF/A-2 ou PDF/A-4. Para cada anexo de arquivo, seu relacionamento com alguma parte do documento principal deve ser especificado com a chave AFRelationship.

Ao abandonar os níveis de conformidade a/b /u anteriores, o PDF/A-4 apresenta dois novos níveis de conformidade:

- O PDF/A-4f permite anexos de arquivos que não sejam PDF/A (de maneira semelhante, o PDF/A-3 estende o PDF/A-2).
- O PDF/A-4e é direcionado à comunidade de engenharia.
 - Ele é apresentado como sucessor da norma existente PDF/E-1 ISO 24517-1, baseada no PDF 1.6.
 - O plano anterior para definir um novo formato PDF/E-2 como ISO 24517-2 foi cancelado. Em vez disso, o PDF/A-4e adiciona anotações RichMedia para conteúdo 3D no formato U3D ou PRC ao formato PDF/A-4 básico.

Nota: Até o momento da escrita deste documento, essa versão ainda não havia sido publicada.

Archivematica

Como os usuários do Archivematica sabem, uma instalação padrão desse software inclui uma Política de Formatação de Registro (FPR – *Format Policy Registry*) que contém regras, comandos e ferramentas para uma ampla variedade de ações de preservação executadas automaticamente durante a ingestão.¹⁵

Um tipo de regra é a Normalização: existem centenas de regras para normalizar durante a ingestão. Se o usuário optar por usar essa regra, essas serão invocadas automaticamente em qualquer arquivo ingerido para o qual exista uma regra de normalização. Aqui, essa regra pode ser usada para normalizar arquivos para preservação (formação de AIPs) e/ou para acesso (formação de DIPs).

Existem alguns motivos para o qual deve-se normalizar o objeto digital após a ingestão:

- Primeiro, restringir seu acervo a um número menor de formatos para preservação a longo prazo, em formatos hoje considerados sustentáveis e “favoráveis à preservação”.

¹⁴ <https://www.iso.org/standard/71832.html>

¹⁵ https://wiki.archivematica.org/Format_policy_registry_requirements

¹⁶ Converter formatos de arquivo em um conjunto selecionado de formatos de preservação

- o Isso significa conter em uma poucos formatos seu repositório, em vez de várias dezenas ou até centenas, dependendo da diversidade de seus produtores de conteúdo.
- Segundo, permite identificar e resolver problemas com formatos durante a ingestão, ao invés de descobri-los muito tempo depois, dificultando essas práticas.
 - Por exemplo, um arquivo de imagem pode não normalizar corretamente porque possui um problema de espaço de cor; então, melhor corrigir esse problema agora, com ferramentas e conhecimentos atuais, do que tentar corrigi-lo em algum momento no futuro.
- Terceiro, existe uma quantidade de trabalho antecipado, permitindo um nível mais alto de confiança onde grande parte do trabalho pesado sobre preservação digital foi feito no momento em que o conteúdo foi colocado em armazenamento de longo prazo (que o AIP é concluído e não precisará ser acessado por muito tempo).

O padrão de normalização do PDF para preservação, conforme FPR, é o PDF/A-1b. Isso porque o suporte à ferramenta para PDF/A-1b era melhor do que para o PDF/A-2b quando começaram a desenvolver o Archivematica e a implementar planos de preservação. O PDF/A-1b era considerado pela *Artefactual* como menos rigoroso e complexo do que outras versões do padrão; portanto, foi considerado como padrão de preservação. Quando o suporte à ferramenta PDF/A-2b melhorou, a empresa parou de codificar os comandos de normalização, substituindo-os pelo FPR, o que significava que os usuários estariam livres para experimentar outros formatos, comandos e ferramentas de destino. Pode-se alterar as configurações padrão no Archivematica FPR no caso do PDF, ou até de outros formatos, já que os padrões fornecidos pelo Archivematica são frequentemente considerados recomendações de fato pela *Artefactual Systems*.

ANEXO I

6 Requisitos Técnicos (Technical Requirements)

6.7 Estrutura Lógica (Logical structure)

6.7.1 Geral (General)

6.7.2 Marcado em PDF (Tagged PDF)

6.7.2.1 Geral (General)

6.7.2.2 Dicionário de informações sobre marcas (Mark information dictionary)

6.7.3 Artefatos (Artefacts)

6.7.3.1 Especificação de artefatos (Specification of artefacts)

6.7.3.2 Limites de palavras (Word boundaries)

6.7.3.3 Estrutura hierárquica (Structure hierarchy)

6.7.3.4 Tipos de estrutura (Structure types)

6.7.4 Especificação da linguagem natural (Natural language specification)

6.7.5 Descrições alternativas (Alternate descriptions)

6.7.6 Anotações não textuais (Non-textual annotations)

6.7.7 Texto de substituição (Replacement text)

6.7.8 Ampliações de abreviações e acrônimos (Expansions of abbreviations and acronyms)

6.8 Arquivos incorporados (Embedded files)

ANEXO II

	PDF/A-1	PDF/A-2	PDF/A-3
Parte	Parte 1	Parte 2	Parte 3
Norma ISO	ISO 19005-1:2005	ISO 19005-2:2011	ISO 19005-3:2012
Ano de liberação	2005	2011	2012
Versão do PDF	Baseado na versão 1.4	Baseado na versão 1.7	Baseado na versão 1.7

	PDF/A-1	PDF/A-2	PDF/A-3
Níveis de conformidade	PDF/A-1a	PDF/A-2a	PDF/A-2a
	PDF/A-1b	PDF/A-2b	PDF/A-3b
	x	PDF/A-2u	PDF/A-3u
Metadados e fontes	Precisam estar embutidas ¹⁷	Precisam estar embutidas	Precisam estar embutidas
		Fontes OpenType pode ser embutidas.	Fontes OpenType pode ser embutidas.
Conteúdo de áudio e vídeo	x	x	x
Encriptação	x	x	x
Links externos	x	x	x
Javascript e arquivos executáveis	x	x	x
Embutir diversos arquivos (como planilhas, documentos de texto, XML, CAD, CSV,...). ¹⁸	x	x	◇
Compressão LZW e JPEG2000	x	◇	◇
Transparência e camadas	x	◇	◇
Disponibilidade de uso de assinatura digital do padrão PAdES	x	◇	◇
Permitido uso de compressão JPEG2000	x	◇	◇
Embutir arquivos PDF/A	x	◇	◇
Formulários	Permitido (com restrições)	◇	◇

Fonte: Retirado da pesquisa.

ANEXO III

	PDF/A-1	PDF/A-2	PDF/A-3
Inclusão de dados de pesquisa em documentos acadêmicos	x	x	Utiliza-se o perfil Data em formatos abertos como CSV ou XML
			Os dados podem ser relacionados a um elemento do documento (gráfico, tabela etc.) ou ao documento como um todo
Inclusão de arquivos de aplicações CAD	x	x	Utiliza-se o perfil Source para permitir a edição do documento original
Captura de documentos disponíveis na WEB com licença Creative Commons que estão no formato PDF	x	x	A conversão para PDF/A embute o documento PDF original como arquivo associado utilizando o perfil Source, bem como recursos que garantem a preservação, tais como, fontes tipográficas e esquema de cores

¹⁷ Embedding font.

¹⁸ Os chamados arquivos associados.

	PDF/A-1	PDF/A-2	PDF/A-3
Utilização do PDF/A combinado com XML em um fluxo de autoria	x	x	O PDF/A-3 apresenta a informação formatada, para conveniência de leitura, e embute a informação estruturada (XML), para facilidade de criação de novas versões.

Fonte: BRASIL (2016)

Bibliografia

ABNT - Associação Brasileira de Normas Técnicas. ABNT NBR ISO 19005-1 - **Gerenciamento de documentos – Formato eletrônico de arquivo de documento para preservação de longo prazo** – Parte 1. Rio de Janeiro: ABNT, 2009.

ADOBE ACROBAT. **Configurações de conversão do Adobe PDF**. Acesso em 8 de jan de 2020, disponível em <<https://helpx.adobe.com/br/acrobat/using/pdf-conversion-settings.html>>.

BRASIL. Conselho Nacional de Arquivos. **Dicionário brasileiro de terminologia arquivística**. Rio de Janeiro, 2005. ISBN: 85-7009-075-7. 232 páginas. Publicações Técnicas. Número 51. Acesso em 9 jan 2020, disponível em <http://www.arquivonacional.gov.br/images/pdf/Dicion_Term_Arquiv.pdf>.

BRASIL. Conselho Nacional de Arquivos. Câmara Técnica de Documentos Eletrônicos. **Carta para a Preservação do Patrimônio Arquivístico Digital**: preservar para garantir o acesso. Rio de Janeiro, 2005. Acesso em 09 jan 2020, disponível em <http://www.conarq.gov.br/images/publicacoes_textos/Carta_preservacao.pdf>.

BRASIL. Conselho Nacional de Arquivos. Câmara Técnica de Documentos Eletrônicos. e-ARQ Brasil: **Modelo de Requisitos para Sistemas Informatizados de Gestão**. Versão 1.1. Rio de Janeiro, 2011. Acesso em 9 jan 2020, disponível em <<http://www.siga.arquivonacional.gov.br/images/publicacoes/e-arq.pdf>>.

BRASIL. Conselho Nacional de Arquivos. Câmara Técnica de Documentos Eletrônicos. **Orientação Técnica nº 4 – Recomendações de uso do PDF/A para Documentos Arquivísticos**. Rio de Janeiro. 2016. Acesso em 8 jan 2020, disponível em <http://conarq.arquivonacional.gov.br/images/ctde/Orientacoes/Orientacao_tecnica_4.pdf>.

BRASIL. Instituto Nacional de Tecnologia da Informação – ITI (Ed.). (25 de mai de 2018). **Infraestrutura de Chaves Públicas Brasileira**. Acesso em 7 jan 2020, disponível em <https://www.iti.gov.br/images/repositorio/legislacao/documentos-principais/15.3/DOC-ICP-15.03_-_Versao_7.4_REQ_DAS_POL_DE_ASSIN_DIG_NA_ICP-BRASIL.pdf>.

FANNING, B. A. (Abril de 2008). **Preserving the Data Explosion**: Using PDF. Acesso em 26 de nov de 2019, disponível em <<https://www.dpconline.org/docs/technology-watch-reports/86-preserving-the-data-explosion-using-pdf/file>>.

FileInfo (Ed.). (s.d.). **The File Information Database**. Acesso em 7 de jan de 2020, disponível em <<https://fileinfo.com/>>.

ISO, International Organization for Standardization. **ISO 19005-3:2012(en) Document management – Electronic document file format for long-term preservation – Part 3: Use of ISO 32000-1 with support for embedded files (PDF/A-3)**. Acesso em 8 jan 2020, disponível em <<https://www.iso.org/obp/ui/en/#iso:std:iso:19005:-3:ed-1:v1:en:biblref:7>>.

ISO, International Organization for Standardization. **ISO 19005-2:2011(en) Document management – Electronic document file format for long-term preservation – Part 2: Use of ISO 32000-1 (PDF/A-2)**. Acesso em 8 jan 2020, disponível em <<https://www.iso.org/obp/ui/en/#iso:std:iso:19005:-2:ed-1:v1:en:biblref:7>>.

JUSBRASIL (Ed.). (2014). **PJe-JT: dicas para encaminhar petições em formato PDF-A**. Acesso em 8 de jan de 2020, disponível em <<https://oab-ms.jusbrasil.com.br/noticias/112218818/pje-jt-dicas-para-encaminhar-peticoes-em-formato-pdf-a>>.

LOC, Library of Congress (Ed.). (s.d.). **Sustainability of Digital Formats: Planning for Library of Congress Collections**. Acesso em 7 jan 2020, disponível em <<https://www.loc.gov/preservation/digital/formats/>>.

MARTINS, Alexandre Pedron. **O Formato PDF/A como Meio de Preservação Digital no Processo de Digitalização das Teses e Dissertações da Biblioteca Central da UFSC**. Universidade Federal de Santa Catarina. Centro de Ciências da Educação. Curso de Graduação em Biblioteconomia. Florianópolis. 2012. Orientador: Dr. Marcio Matias. 93f. Acesso em 8 jan 2020, disponível em <<https://core.ac.uk/download/pdf/30379512.pdf>>.

MERCHANT, Dave. Acrobat Library. **Working with PDF/A in Acrobat 9. Vídeo tutorial**. Acesso em 8 jan 2020, disponível em <www.adobe.com/go/learn_acr_pdfa_br>.

MITCHAM, Jenny. **Some Thoughts on PDF/A-3. Digital Archiving at the University of York**. 2013. Acesso em 8 jan 2020, disponível em <<http://digital-archiving.blogspot.com/search/label/PDF%2Fa>>.

PACER. **PUBLIC ACCESS TO COURT ELECTRONIC RECORDS – CM/ECF TO TRANSITION TO PDF/A**. Acesso em 8 jan 2020, disponível em <<https://www.pacer.gov/announcements/general/pdfa.html>>.

PDFEN (Ed.). (s.d.). **Differences Between PDF/A-2 and PDF/A-1**. Acesso em 8 de jan de 2020, disponível em <<https://www.pdfen.com/what-is-pdfa/what-are-the-differences-between-pdf-a-2-and-pdf-a-1>>.

PDFEN (Ed.). (s.d.). **WHAT IS PDF/A-2 and What Type of PDF/A-2 Documents are There??** Acesso em 8 jan 2020, disponível em <<https://www.pdfen.com/what-is-pdfa/what-is-pdf-a-2>>.

PDFLIB. **PDFlib GmbH – The PDF/A Family of Archiving Standards**. Acesso em 8 jan 2020, disponível em <<https://www.pdflib.com/pdf-knowledge-base/pdfa/the-pdfa-standards/>>.

PDFTRON (Ed.). (16 de jan de 2019). **What are the Different Versions of PDF/A**. Acesso em 8 de jan de 2020, disponível em <<https://www.pdftron.com/blog/pdfa-format/what-are-the-different-types-of-pdfa/>>.

PDFTRON (Ed.). (s.d.). **All About PDF/A**. Acesso em 8 de jan de 2020, disponível em PDFTRON: <<https://www.pdftron.com/blog/pdfa-format/all-about-pdfa/>>.

RONDINELLI, Rosely Curi. **Gerenciamento arquivístico de documentos eletrônicos**. São Paulo: FGV, 2007.

SANTOS, Vanderlei Batista dos. **Gestão de documentos eletrônicos: uma visão arquivística**. Brasília: ABARQ, 2005.

TIFF. (s.d.). Acesso em 7 de jan de 2020, disponível em <<https://www.tiff.net/>>.

VITAL, L. P. (2011). **O PDF/A na Gestão de Documentos Arquivísticos**. Florianópolis, Santa Catarina: BRAPCI - Base de Dados em Ciência da Informação. Volume 21. Número 43. Acesso em 8 jan 2020, disponível em <<http://www.brapci.inf.br/index.php/res/download/48460>>

MANUAL DE LEVANTAMENTO DO METADADOS UTILIZADOS NO CROSSWALK DO BARRAMENTO ARQUIVÍSTICO

Tatiana Canelhas – novembro 2019

Uma das principais funções de um repositório digital confiável é o gerenciamento da informação, ou seja, é a gestão das informações descritivas (metadados) dos documentos recebidos no repositório. Metadados são dados sobre dados, isto é, são informações que descrevem, dentre outros atributos, a estrutura, a forma e o tempo do documento. A identificação desses elementos é fundamental para saber dados sobre a sua origem, data de criação, local onde foi produzido, formato do suporte e assunto, por exemplo. De acordo com o RDC-Arq,

O principal objetivo desses metadados é apoiar o acesso e a recuperação dos documentos, e vão além das informações descritivas mais usuais (autor, título, data), envolvendo outras informações

descritivas úteis aos usuários, tais como tamanho do arquivo disponível para download ou informação sobre a aplicação necessária para ler o arquivo. (CONARQ & CTDE, 2015, p. 16).

Conforme Rondinelli, o termo metadado na tecnologia da informação designa as informações necessárias para dar significados aos dados armazenados em um sistema de computador. Assim, instrumentos que representam a informação, como inventário, catálogos e índices passaram a ser vistos como metadados, estando ou não em versão computadorizada. Ainda, para a autora, os metadados:

[...] se constituem em componentes do documento eletrônico arquivístico e em instrumentos para sua análise diplomática. É através do domínio desse tipo de análise que será possível estabelecer métodos que garantem a fidedignidade e a autenticidade do documento. (RONDINELLI, 2002, p. 62)

Existem vários tipos de metadados. No mundo do objeto digital, os metadados são geralmente divididos entre as categorias a seguir (UC Santa Cruz, s.d.):

- Metadados administrativos, incluindo:
 - Metadados de direitos (direitos de propriedade intelectual e informações de uso);
 - Metadados técnicos (detalhes técnicos sobre o objeto e suas instâncias como formato, tamanho, como abrir, acessar e usar);
 - Metadados de preservação (registro da série de ações tomadas em um objeto para garantir sua longevidade e viabilidade).
- Metadados descritivos – descrevem o objeto com o objetivo de pesquisa e identificação (autor, título, palavras-chave, resumo, identificador). Ex.: Dublin Core, Nobrade, ISAD(G).
- Metadados estruturais – descrevem como as partes de um único objeto se encaixam e como um objeto se relaciona a outros objetos.

Durante o gerenciamento das informações descritivas por um repositório arquivístico, ou seja, gestão dos metadados dos documentos que serão admitidos no repositório, devem-se considerar os metadados mínimos que permitam a busca e localização dos documentos, conhecidos pela comunidade-alvo de usuários (ex.: número de processo) e a captura ou criação dos metadados mínimos pelo repositório durante o processo de admissão, e associação desses metadados ao AIP correspondente. (CONARQ & CTDE, 2015, p. 16). Estamos falando aqui dos metadados de gestão que serão recuperados dos sistemas DJe e PJe, junto aos metadados descritivos Dublin Core e ISAD(G)¹⁹, previamente mapeados no mesmo sistema.

Esse manual terá as seguintes referências:

1. **Resolução n. 91 do CNJ**, de 29 de setembro de 2009, que institui o Modelo de Requisitos para Sistemas Informatizados de Gestão de Processos e Documentos do Poder Judiciário (MoReq-Jus) e disciplina a obrigatoriedade da sua utilização no desenvolvimento e manutenção de sistemas informatizados para as atividades judiciais e administrativas no âmbito do Poder Judiciário.
2. **e-ARQ Brasil**, Modelo de Requisitos para Sistemas Informatizados de Gestão Arquivística de Documentos, elaborado pela Câmara Técnica de Documentos Eletrônicos do Conselho Nacional de Arquivos (CONARQ), aprovado pela Resolução n. 25, de 27 de abril de 2001, que em sua parte II elenca os metadados a serem associados aos documentos, a fim de apoiar a gestão, a preservação e a presunção de autenticidade dos documentos arquivísticos. A especificação dos metadados considera as seguintes entidades: documento, evento de gestão, classe, agente, componente digital e evento de preservação.
3. **Resolução n. 24 do CONARQ**, que estabelece diretrizes para a transferência e recolhimento de documentos arquivísticos digitais para instituições arquivísticas públicas.

Para garantir a confiabilidade, a autenticidade e o acesso documental, o MoReq-Jus estabelece condições a serem cumpridas durante a gestão documental pelos Sistemas Informatizados de Gestão de Processos e Documentos (GestãoDoc) – sendo que para documentos digitais, este deve registrar os documentos e as

¹⁹ ISAD(G) – General International Standard Archival Description, ou Norma Geral Internacional de Descrição Arquivística, desenvolvida pelo Conselho Internacional de Arquivos (CIA).

operações de produção, de tramitação, de guarda, de armazenamento, de preservação, de arquivamento e de recebimento (Conselho Nacional de Justiça, 2009).

O Diário da Justiça Eletrônico (DJe) é a ferramenta utilizada pelo TJDF-T para emitir publicações oficiais. Nele existem duas plataformas, onde uma delas é o do Sistema, onde os usuários com acesso enviam textos a serem publicados. O Processo Judicial Eletrônico (PJe) é a ferramenta utilizada pelo TJDF-T para tramitação de processos judiciais, em que o objetivo principal é permitir a prática de atos processuais, assim como o seu acompanhamento, independentemente de o processo tramitar em esferas diversas. A Lei nº 11.419, de 19 de dezembro de 2006 dispõe sobre a informatização do processo judicial. Estamos falando aqui de sistemas que não são caracterizados por serem SIGADs ou GestãoDoc, e por isso, não podemos encontrar em seus metadados tudo o que é normativamente visto como obrigatório.

Dito isto, foram-se levantados os principais metadados que estão armazenados nas tabelas do banco de dados do DJe ou que são pesquisados através do sistema do Modelo Nacional de Interoperabilidade (MNI)²⁰ para o PJe, compilados pelas principais normas arquivísticas brasileiras.

Caso: DJe – Diário da Justiça Eletrônico do Tribunal de Justiça do Distrito Federal e Territórios (TJDFT)

Metadados de Gestão (MoReq-Jus, e-Arq e Resolução n. 24):

Número identificador atribuído pelo sistema

Descrição: Número gerado pelo sistema na hora de sua captura. Geralmente é o campo ID de uma tabela.

Campo CSV: numeroSistema

Tabela: tjdf_dje.DiarioEletronico.ID

Exemplo: 5620

Número identificador atribuído pelo usuário

Descrição: Identificar de forma unívoca e persistente o documento. Podemos usar no caso do DJe, por exemplo, o ano concatenado com o número do Diário da Justiça.

Campo CSV: numeroidentificador

Tabela: tjdf_dje.DiarioEletronico.Ano || tjdf_dje.DiarioEletronico.Numero

Exemplo: 201961

Responsável pela publicação

Descrição: Buscar no sistema o login do responsável pela publicação do documento.

Campo CSV: responsavelPubLogin

Tabela: tjdf_dje.DiarioEletronico.Login

Exemplo: t312345

Redator (campo retirado, pois há como informar os redatores de cada redação do DJe)

Descrição: Relacionar o responsável pela redação do documento. Pode ser mais de uma pessoa.

Campo CSV: responsavelRedator

Tabela: tjdf_dje.DiarioEletronico.Login

Exemplo: Pessoa 1, Pessoa 2, Pessoa 3...

Data de Criação

Descrição: Data em que o documento foi criado.

Campo CSV: dataCriacao

Tabela: tjdf_dje.DiarioEletronico.DataHoraDeModificacao

Exemplo: 2019-04-20

²⁰ O MNI representa o padrão para troca de informações no âmbito do Poder Judiciário. A implementação dos serviços está sob responsabilidade dos Tribunais, órgãos da Justiça e Instituições Privadas interessadas em aderir ao modelo.

Data de Publicação

Descrição: Data de publicação do documento.

Campo CSV: dataPublicacao

Tabela: tjdf_dje.DiarioEletronico.DataDePublicacao

Exemplo: 2019-04-29

- Ano
 - anodiario
- Número
 - numerodiario

Código de Classificação

Descrição: Código da classe em que o documento está classificado, de acordo com o Plano de Classificação do Órgão/Entidade.

Campo CSV: codigoClassificacao

Onde buscar: https://www.tjdft.jus.br/publicacoes/publicacoes-oficiais/portarias-seg/2017/MINUTA_DE_PROPOSTA_DO_PCTT_TJDFT___Versao_1_47___para_publicacao.pdf

Exemplo: 06.04.01

Observação: Essa classificação foi uma informação fornecida pela equipe do arquivo do TJDF. É preciso que a classificação documental seja feita para cada tipo que será ingerido pelo repositório arquivístico.

Assunto classificação

Descrição Nome completo da Classe em que o documento está classificado, encontrado na Tabela de Temporalidade do Órgão/Entidade

Campo CSV assuntoClassificacao

Onde buscar https://www.tjdft.jus.br/publicacoes/publicacoes-oficiais/portarias-seg/2017/MINUTA_DE_PROPOSTA_DO_PCTT_TJDFT___Versao_1_47___para_publicacao.pdf

Exemplo GESTÃO DE DOCUMENTOS E INFORMAÇÕES - PRODUÇÃO EDITORIAL - Edição, co-edição e folder - Periódicos

Plano Classificação e Tabela de Temporalidade Utilizada

Descrição: Descrever qual a versão do plano de classificação e tabela de temporalidade foi utilizado para extrair os metadados deste documento.

Campo CSV: versaoPCTT

Onde buscar: https://www.tjdft.jus.br/publicacoes/publicacoes-oficiais/portarias-seg/2017/MINUTA_DE_PROPOSTA_DO_PCTT_TJDFT___Versao_1_47___para_publicacao.pdf

Exemplo: Plano de classificação e tabela de temporalidade de documentos - PCTT-AD TJDF (Minuta de Proposta do PCTT-TJDF - Versão 1-47)

Prazo de guarda

Descrição: Indicação do prazo estabelecido em tabela de temporalidade para o cumprimento da destinação.

Campo CSV: prazoGuardaClassificacao

Onde buscar: https://www.tjdft.jus.br/publicacoes/publicacoes-oficiais/portarias-seg/2017/MINUTA_DE_PROPOSTA_DO_PCTT_TJDFT___Versao_1_47___para_publicacao.pdf

Exemplo: Um (01) ano de guarda em arquivo corrente e depois é recolhido para o arquivo permanente.

Destinação prevista

Descrição: Indicação da próxima ação de destinação (transferência, eliminação ou recolhimento) prevista para o documento, em cumprimento à tabela de temporalidade.

Campo CSV: destinacaoPrevista

Onde buscar: https://www.tjdft.jus.br/publicacoes/publicacoes-oficiais/portarias-seg/2017/MINUTA_DE_PROPOSTA_DO_PCTT_TJDFT___Versao_1_47___para_publicacao.pdf

Exemplo: Repositório Arquivístico Digital - Tribunal de Justiça do Distrito Federal e dos Territórios

Grau de Sigilo (grauSigilo)

Descrição: Registrar informações referentes à classificação de grau de sigilo, tais como, se é de acesso integral, reservada, secreta ou ultra-secreta.

Campo CSV: grauSigilo

Onde buscar: Será um texto fixo, de acordo com o tipo documental.

Exemplo: Acesso integral

Restrição de acesso (restrAcesso)

Descrição: Olhar página: <http://www.acessoainformacao.gov.br/assuntos/aprenda-pedido-de-acesso/excecoes>:

- **Informações Pessoais:** informações relacionadas à pessoa natural identificada ou identificável relativas à intimidade, vida privada, honra e imagem. Saiba mais na seção V da Lei de Acesso à Informação.
- **Informações classificadas com base na Lei nº 12.527:** A Lei de Acesso à Informação prevê hipóteses para classificação de informações que são imprescindíveis à segurança da sociedade ou do Estado. Saiba mais.
- **Informações cujo sigilo é garantido por outras legislações:** por exemplo, sigilo bancário ou comercial, segredo de justiça, hipóteses de segredo industrial, entre outras.

Campo CSV: restricaoAcesso

Onde buscar: Será um texto fixo, de acordo com o tipo documental.

Exemplo: Nenhuma restrição

Identificação do tipo de meio do documento

Descrição: Documento/volume/processo/dossiê: digital, não digital (convencionais/analógicos) ou híbrido, para controlar as relações entre os meios e o monitoramento de preservação.

Campo CSV: meioDocumental

Onde buscar: Será um texto fixo.

Exemplo: Documento digital

Gênero Documental

Descrição: É a configuração da informação no documento de acordo com o sistema de signos utilizado na comunicação do documento. Exemplos: Audiovisual; textual; cartográfico; iconográfico; multimídia.

Campo CSV: generoDocumental

Onde buscar: Será um texto fixo.

Exemplo: Textual

Espécie Documental

Descrição: Divisão de gênero documental que reúne tipos de documentos por seu formato. São exemplos de espécies documentais ata, carta, decreto, memorando, ofício, planta, relatório.

Campo CSV: especieDocumental

Onde buscar: Será um texto fixo.

Exemplo: Diário da Justiça

Tipo Documental

Descrição: É a configuração da espécie documental de acordo com a atividade que a gerou. Serve para complementar a descrição do documento ou a identificação do título. Exemplos: relatório de pesquisa; carta precatória; ofício-circular; prontuário médico; prontuário de funcionário.

Campo CSV: tipoDocumental

Onde buscar: Será um texto fixo.

Exemplo: Diário da justiça para publicação diária de atos judiciais e administrativos e informações relevantes à esfera do TJDF.

Localização

Descrição: Deve ser utilizado, obrigatoriamente, quando o documento não se encontra no sistema de gestão arquivística de documentos, e é mantido em outra área de armazenamento, seja virtual ou física.

Utilizado apenas para os documentos não digitais, para a parte não digital dos documentos híbridos ou para os documentos digitais off-line.

No caso de documentos digitais on-line, este controle é feito com relação aos componentes digitais.

Campo CSV: localização

Onde buscar: Verificar se existe em algum outro lugar esse metadado.

Exemplo: Documento encontrado na plataforma online do Diário da Justiça do TJDF, no endereço <https://pesquisadje.tjdft.jus.br/>

Procedência

Descrição: Relatar a unidade administrativa a qual o documento pertencia antes de ser arquivado no repositório. Pode ser, por exemplo, a unidade administrativa onde o servidor responsável pela publicação está lotado.

Campo CSV: procedencia

Exemplo: NUGAD - Núcleo de Gestão de Sistemas Administrativos

Assinatura digital (campo retirado, pois não obtivemos essa informação pelo arquivo P7S)

Descrição: Nome do agente assinador do documento.

Campo CSV: assinaturaDigital

Onde buscar: Utilizar de codificação para extrair esse dado do documento P7S.

Exemplo: Fulano de Tal (pessoa que assinou o documento)

Checksum (campo retirado, pois o checksums vai em um arquivo a parte)

Descrição: Utilizado para verificação da fixidade de um documento, ou seja, se este não foi corrompido entre o momento em que foi extraído do DJe ao momento que será ingerido no repositório arquivístico digital confiável.

Campo CSV: checksum

Onde buscar: Utilizar de codificação para salvar esse dado.

Exemplo: 12d83cd44473e3aaba.....

Algoritmo checksum

Descrição: Registro do algoritmo utilizado para o criptografia do cálculo do checksum. Exemplo: SHA, MD5

Campo CSV: algoritmoChecksum

Onde buscar: Utilizar de codificação para salvar esse dado.

Exemplo: sha-256 ou MD5

Metadados Descritivos Dublin Core:

CAMPO	CAMPOS DC CSV	EXEMPLO DE PREENCHIMENTO	ONDE BUSCAR
Título	dc.title	Edição no 61/2019	61 - tjdf_dje.DiarioEletronico.Numero 2019 - tjdf_dje.DiarioEletronico.Ano
Criador	dc.creator	Tribunal de Justiça do Distrito Federal e dos Territórios	A pessoa ou organização responsável pelo conteúdo
Assunto	dc.subject	-	Os assuntos cobertos pelo conteúdo
Descrição	dc.description	-	Uma descrição resumida sobre o conteúdo
Editor	dc.publisher	Tribunal de Justiça do Distrito Federal e dos Territórios	Uma entidade (pessoa, organização ou serviço) responsável por deixar o conteúdo disponível
Contribuidor / Colaborador	dc.contributor	Tribunal de Justiça do Distrito Federal e dos Territórios	Pessoa, organização ou serviço que contribuiu com o conteúdo

CAMPO	CAMPOS DC CSV	EXEMPLO DE PREENCHIMENTO	ONDE BUSCAR
Data	dc.date	2019-03-29	tjdf_dje.DiarioEletronico. DataDePublicacao Um ponto ou período de tempo associado com o tempo de vida do conteúdo. Normalmente uma data de quando o conteúdo se tornou disponível. Formato: AAAA-MM-DD ou AAAA-MM ou AAAA
Tipo do Recurso	dc.type	texto	Uma categoria para o tipo de conteúdo. Exemplos: Collection, Dataset, Event, Image, InteractiveResource, MovingImage, PhysicalObject, Service, Software, Sound, StillImage, Text.
Formato	dc.format	application/pdf	Como o conteúdo ou recurso é apresentado (exemplos encontrados em: http://www.iana.org/assignments/media-types/).
Identificador do Recurso	dc.identifier	201961	61 - tjdf_dje.DiarioEletronico.Numero 2019 - tjdf_dje.DiarioEletronico.Ano
			Um identificador único para o seu conteúdo. Pode ser um texto ou um número gerado por um sistema de identificação formal - ou apenas uma URL
			Onde o conteúdo foi inicialmente publicado / originado. Um conteúdo que tem alguma relação intelectual com o conteúdo descrito. Se for conteúdo original, veiculado pela primeira vez e sem nenhuma relação intelectual com outros conteúdos, deve ser a URL para esse conteúdo.
Fonte	dc.source	https://pesquisadje-api.tjdft.jus.br/v1/...	Usar no caso do DJe, o site do Diário da Justiça.
Idioma	dc.language	pt_BR	Em que linguagem o conteúdo foi escrito. Você precisa especificar o código correto da língua.
Relação	dc.relation	-	Como o conteúdo está relacionado com outros recursos
Abrangência / Cobertura	dc.coverage	Distrito Federal	Onde o conteúdo está localizado fisicamente. Normalmente inclui localização espacial (nome do lugar ou coordenadas geográficas), período temporal (data) ou jurisdição.
Gerenciamento de Direitos Autorais	dc.rights	Documento pode ser adquirido neste site, ou no site do Diário da Justiça do TJDF	Informação dos direitos autorais do conteúdo. Pode conter a declaração de Copyright ou outros tipos de licenças

Metadados Descritivos ISAD(G)²¹:

CAMPO	CAMPOS DC CSV	EXEMPLO DE PREENCHIMENTO	OBSERVAÇÕES
Código(s) de Referência:	identifier	201961	Campo igual ao dc.identifier. Regras ISAD 3.1.1
Título	title	Edição no 61/2019	Campo igual ao dc.title. Regras ISAD 3.1.2
Data(s)	eventDates	2019-03-29	Campo igual ao dc.date. Regras ISAD 3.1.3
Nível de Descrição	levelOfDescription	Item	Regras ISAD 3.1.4
Dimensão e Suporte	extentAndMedium	application/pdf (ISAD 3.1.5)	Campo igual ao dc.format. Regras ISAD 3.1.5
Nome do(s) Produtor(es)	eventActors	Tribunal de Justiça do Distrito Federal e dos Territórios	Regras ISAD 3.2.1
História Administrativa / Biografia	eventActorHistories		Regras ISAD 3.2.2
História Arquivística	archivalHistory		Regras ISAD 3.2.3
Procedência	acquisition	Tribunal de Justiça do Distrito Federal e dos Territórios	Regras ISAD 3.2.4
Âmbito e Conteúdo	scopeAndContent		Campo igual ao dc.description. Regras ISAD 3.3.1
Avaliação, Eliminação e Temporalidade	appraisal	06.04.01 - Gestão de Documentos e Informações - Produção Editorial - Edição, co-edição e folder - Periódicos	Regras ISAD 3.3.2
Incorporações	accruals		Regras ISAD 3.3.3
Sistemas de Arranjo	arrangement		Regras ISAD 3.3.4
Condições de Acesso	accessConditions	Documento de caráter ostensivo	Campo igual ao dc.rights. Regras ISAD 3.4.1
Condições de Reprodução	reproductionConditions	Documento pode ser adquirido neste site, ou no site do Diário da Justiça do TJDF	Regras ISAD 3.4.2
Idioma	language	pt_BR	Campo igual ao dc.language. Regras ISAD 3.4.3
Características Físicas e Requisitos Técnicos	physicalCharacteristics	Documento digital	Regras ISAD 3.4.4
Instrumentos de Pesquisa	findingAids		Regras ISAD 3.4.5
Existência e Localização dos Originais	locationOfOriginals	Site do Diário da Justiça	Campo igual ao dc.source. Regras ISAD 3.5.1
Existência e Localização de Cópias	locationOfCopies		Regras ISAD 3.5.2

21 http://conarq.arquivonacional.gov.br/images/publicacoes_textos/isad_g_2001.pdf

CAMPO	CAMPOS DC CSV	EXEMPLO DE PREENCHIMENTO	OBSERVAÇÕES
Unidades de Descrição Relacionadas	relatedUnitsOfDescription		Regras ISAD 3.5.3
Nota sobre Publicação	publicationNote		Regras ISAD 3.5.4
Notas	generalNote		Regras ISAD 3.6.1
Nota do Arquivista	archivistNote		Regras ISAD 3.7.1
Regras ou Convenções	rules	Documento foi descrito com base nas regras da ISAD(G) do Conselho Internacional de Arquivos.	Regras ISAD 3.7.2
Datas de criação, revisão e eliminação	revisionHistory		Registrar responsável(is) e a data(s) em que foi elaborada ou revista. Regras ISAD 3.7.3

Outros Metadados ISAD(G)

Esses metadados foram extraídos na planilha oficial da Artefactual, para importação de metadados ISAD(G) para a plataforma de difusão AtoM.

CAMPO	CAMPOS DC CSV	EXEMPLO DE PREENCHIMENTO	OBSERVAÇÕES
Locais	[placeAccessPoints]	Distrito Federal	Campo igual ao dc.coverage.
Assuntos	[subjectAccessPoints]		Campo igual ao dc.subject.
Pessoas / Entidades	[nameAccessPoints]		Lista de pessoas ou entidades relacionadas ao documento.
Gênero	[genreAccessPoints]	Textual	
Repositório	[repository]	TJDFT	Campo igual ao dc.relation (isLocatedAt).
Sistema de Escrita	[script]	latim	Registrar a escrita do documento
Notas sobre o Idioma	[languageNote]		Especificar qualquer tipo de alfabeto, escrita, sistema de símbolos ou abreviaturas utilizados.
URI do Obj	[digitalObjectURI]		URI enviada do archivematica para o AtoM, preenchida pelo sistema.
Checksum	[digitalObjectChecksum]		
Identificador da Descrição	[descriptionIdentifier]		Identificador único dado pelo usuário da descrição.
Identificador da Instituição	[institutionIdentifier]		Identificador da instituição responsável pela descrição.

CAMPO	CAMPOS DC CSV	EXEMPLO DE PREENCHIMENTO	OBSERVAÇÕES
Status da Descrição	[descriptionStatus]	Draft	Registrar o estado da descrição, indicando se está em elaboração (rascunho), revisada ou final (concluída).
Nível de detalhamento da Descrição	[levelOfDetail]	Minimal	Indicar o nível de detalhe da descrição (mínima, parcial ou completa) de acordo com as regras ou diretrizes nacionais e internacionais.
Idioma(s) da Descrição	[languageOfDescription]	pt_BR	Indicar o idioma(s) utilizado para criar a descrição do material arquivístico
Escrita(s) da descrição	[scriptOfDescription]	latim	Indicar a escrita utilizada para criar a descrição do material arquivístico.
Fontes	[sources]	https://pesquisadje-api.tjdft.jus.br/v1/...	Site do Diário da Justiça
Status da publicação	[publicationStatus]	Draft	Preencher entre Rascunho ou Publicado
Nome do Objeto físico	[physicalObjectName]		Ex.: Caixa Nro 10
Endereçamento do objeto físico	[physicalObjectLocation]		Ex.: Arquivo Judiciário, Módulo 05, Estante 03, Prateleira 01. Ex: AJ M5 E3 P1
Tipo do objeto físico	[physicalObjectType]		Exemplos: caixa, ficha, pasta, prateleira, armário, etc.
Identificador alternativo	[alternativeIdentifiers]		Introduza um código de referência anterior, identificador alternativo, ou qualquer outra sequência alfanumérica associada ao registro.
Etiqueta do Id. alternativo	[alternativeIdentifierLabels]		Digite um nome para o campo identificador alternativo que indique sua finalidade e uso. Ex. código de barras, AIP

CAMPO	CAMPOS DC CSV	EXEMPLO DE PREENCHIMENTO	OBSERVAÇÕES
Tipo de data	[eventTypes]	Creation	Indique se são data(s) em que os documentos de arquivo foram acumulados pelo produtor no exercício das suas atividades, ou seja, integrados num sistema de arquivo, ou, se são data(s) em que os documentos foram produzidos, ou seja, que inclui a data de cópias, edições, versões, anexos ou originais de peças produzidas antes da sua acumulação ou integração no sistema de arquivo. Escolher entre Produção ou Acumulação.
Data Inicial	[eventStartDates]		Indique a data de início
Data Final	[eventEndDates]		Indique a data final, se houver.
Descrição do evento	[eventDescriptions]		
Data Tópica	[eventPlaces]		
Cultura	[culture]	pt_BR	Dado automático, preenchido pelo sistema.

Caso: PJe – Processo Judicial Eletrônico do Tribunal de Justiça do Distrito Federal e Territórios (TJDFT)

Metadados de Gestão (PJe) (MoReq-Jus, e-Arq e Resolução n. 24)

Dados do Processo

Número do Processo

Descrição: Número do processo que se deseja responder o expediente. O número do processo judicial deve estar de acordo com as regras definidas pela Resolução do CNJ nº 65, de 16 de dezembro de 2008.

Campo CSV: numeroProcesso

Exemplo: 07196889620188070003

Órgão Julgador - Código

Descrição: Identificador do órgão julgador do processo.

Campo CSV: codigoOrgaoJulgador

Exemplo: 144

Órgão Julgador - Nome

Descrição: Descrição do órgão julgador do processo.

Campo CSV: nomeOrgaoJulgador

Exemplo: 2ª Vara Cível de Ceilândia

Classe Processual - Código

Descrição: Número identificador da classe processual, segundo o padrão estabelecido no Sistema de Gestão de Tabelas Processuais Unificadas. As classes estão disponíveis, de acordo com a configuração de classes para cada jurisdição.

Campo CSV: codigoClasseProcessual

Exemplo: 49

Classe Processual - Nome

Descrição: Descrição da classe processual, segundo o padrão estabelecido no Sistema de Gestão de Tabelas Processuais Unificadas²². As classes estão disponíveis, de acordo com a configuração de classes para cada jurisdição.

Campo CSV: nomeClasseProcessual

Exemplo: Usucapião

Classe Processual - Temporalidade

- Esse campo, a princípio, será preenchido conforme estabelecido na descrição abaixo, de acordo com a tabela de temporalidade do CNJ, apenas para processos com temporalidade permanente. Os que não tiverem essa temporalidade, o campo irá preenchido em branco.
- Na próxima versão do barramento, o TJDFt irá modificar o código para acrescentar a tabela que trata da temporalidade de acordo com a classe documental. Para isso, é preciso que essa parte do código seja documentada, e assim, a equipe de TI do TJDFt consiga achar com facilidade a parte do código que será modificada.

Descrição: Determinar se a temporalidade do assunto é de cunho Permanente ou não.

Se for permanente, preencher o campo com a palavra "Permanente". Caso contrário, não preencher o campo.

Códigos de Classe Permanente: 49,63,64,65,66,88,90,95,110,118,119,121,216,457,1296,1297,1178,11791,1396,1399,1690,1705,1303,1304,1307

Campo CSV: temporalidadeClasseProcessual

Exemplo: Permanente

Localidade - Código (Jurisdição)

Descrição: Códigos das seções/subseções (jurisdição) da instalação do PJe.

Campo CSV: codigoLocalidade

Exemplo: 4

Localidade - Nome (Jurisdição)

Descrição: Descrição das seções/subseções (jurisdição) da instalação do PJe.

Campo CSV: nomeLocalidade

Exemplo: Ceilândia - Fórum Des. José Manoel Coelho

Nível de Sigilo - Código

Descrição: Código do nível de sigilo a ser aplicado ao processo. Dever-se-á utilizar os seguintes níveis:

- 0: públicos, acessíveis a todos os servidores do Judiciário e dos demais órgãos públicos de colaboração na administração da Justiça, assim como aos advogados e a qualquer cidadão
- 1: segredo de justiça, acessíveis aos servidores do Judiciário, aos servidores dos órgãos públicos de colaboração na administração da Justiça e às partes do processo.
- 2: sigilo mínimo, acessível aos servidores do Judiciário e aos demais órgãos públicos de colaboração na administração da Justiça
- 3: sigilo médio, acessível aos servidores do órgão em que tramita o processo, à(s) parte(s) que provocou(ram) o incidente e àqueles que forem expressamente incluídos
- 4: sigilo intenso, acessível a classes de servidores qualificados (magistrado, diretor de secretaria/escrivão, oficial de gabinete/assessor) do órgão em que tramita o processo, às partes que provocaram o incidente e àqueles que forem expressamente incluídos
- 5: sigilo absoluto, acessível apenas ao magistrado do órgão em que tramita, aos servidores e demais usuários por ele indicado e às partes que provocaram o incidente.

Campo CSV: codigoNivelSigilo

Exemplo: 0

²² https://www.cnj.jus.br/sgt/consulta_publica_classes.php

Nível de Sigilo - Nome

Descrição: Nível de sigilo a ser aplicado ao processo. Dever-se-á utilizar os seguintes níveis:

- 0: públicos, acessíveis a todos os servidores do Judiciário e dos demais órgãos públicos de colaboração na administração da Justiça, assim como aos advogados e a qualquer cidadão
- 1: segredo de justiça, acessíveis aos servidores do Judiciário, aos servidores dos órgãos públicos de colaboração na administração da Justiça e às partes do processo.
- 2: sigilo mínimo, acessível aos servidores do Judiciário e aos demais órgãos públicos de colaboração na administração da Justiça
- 3: sigilo médio, acessível aos servidores do órgão em que tramita o processo, à(s) parte(s) que provocou(ram) o incidente e àqueles que forem expressamente incluídos
- 4: sigilo intenso, acessível a classes de servidores qualificados (magistrado, diretor de secretaria/escrivão, oficial de gabinete/assessor) do órgão em que tramita o processo, às partes que provocaram o incidente e àqueles que forem expressamente incluídos
- 5: sigilo absoluto, acessível apenas ao magistrado do órgão em que tramita, aos servidores e demais usuários por ele indicado e às partes que provocaram o incidente.

Campo CSV: nomeNivelSigilo

Exemplo: Públicos, acessíveis a todos os servidores do Judiciário e dos demais órgãos públicos de colaboração na administração da Justiça, assim como aos advogados e a qualquer cidadão.

Data de Ajuizamento

Descrição: Sequência de ano+dia+mês/hora+minuto+segundo. Este atributo corresponde a data de autuação do processo e sempre virá preenchido nos dados básicos do processo após uma consulta.

Campo CSV: dataAjuizamento

Exemplo: 20181207221457

Outro Parâmetro

Descrição: Atributo genérico, podendo ser usado em diversas situações, tais como:

Valor="Andamento" nome="mni:situacaoProcesso" - Indica a situação do processo.

Campo CSV: outroParametro

Exemplo: mni:pje:pedidoLiminarOuAntecipacaoTutela

Competência - Código

Descrição: Identificador da competência a que ele se destina. O parâmetro competência será obrigatório quando, na Jurisdição de peticionamento, houver potencial conflito de competência, ou seja, a relação Classe e Assunto possuem mais de um Órgão Julgador para o processo ser distribuído.

Campo CSV: codigoCompetencia

Exemplo: 7

Competência - Nome

Descrição: Descrição da competência a que ele se destina. O parâmetro competência será obrigatório quando, na Jurisdição de peticionamento, houver potencial conflito de competência, ou seja, a relação Classe e Assunto possuem mais de um Órgão Julgador para o processo ser distribuído.

Campo CSV: nomeCompetencia

Exemplo: Cível

Processo Vinculado

Descrição: Elemento destinado a permitir a indicação da existência de um ou mais processos judiciais vinculados.

Campo CSV: processoVinculado

Exemplo: 07140995020198070016

Modalidade Vinculação Processo

Descrição: Tipo de elemento destinado a permitir a identificação entre processos judiciais. As vinculações podem ser:

- 'AP' (apenso);
- 'AR' (ação rescisória);

- 'CD' (competência delegada);
- 'CT' (continência);
- 'CX' (conexão);
- 'DP' (dependência);
- 'OG' (originário);
- 'OR' (outro tipo de associação que não os anteriores);
- 'PU' (precedente de uniformização);
- 'RG' (repercussão geral);
- 'RR' (recurso repetitivo).

Campo CSV: modalidadeVinculacaoProcesso

Exemplo: Conexão

Prioridade

Descrição: Elemento destinado a permitir a identificação da existência de prioridades processuais não óbvias, ou seja, aquelas que não são resultado direto da identificação da classe processual (ex.: habeas corpus ou mandado de segurança). Na versão 2.0, será texto livre, mas é recomendável utilizar os seguintes textos identificadores: "IDOSO(A)" "RÉU PRESO" "PERECIMENTO" "MENOR" "DOENÇA TERMINAL" "DEFICIENTE FÍSICO"

Campo CSV: prioridade

Exemplo: Idoso(a)

Valor Causa

Descrição: Valor da relação jurídica.

Campo CSV: valorCausa

Exemplo: 200000.0

Assunto Processual - Principal

Descrição: Indica se o assunto será o principal do processo.

Se true, escrever: Assunto Principal.

Se False: deixar o campo vazio.

Campo CSV: assuntoPrincipal

Exemplo: Assunto Principal

Assunto Processual - Código

Descrição: Código do assunto processual, segundo SGT.

Campo CSV: codigoAssuntoProcessual

Exemplo: 899

Assunto Processual - Nome

Descrição: Descrição do assunto processual, segundo SGT.

Campo CSV: nomeAssuntoProcessual

Exemplo: Direito Civil

Assunto Processual - Temporalidade

- Esse campo, a princípio, será preenchido conforme estabelecido na descrição abaixo, de acordo com a tabela de temporalidade do CNJ, apenas para processos com temporalidade permanente. Os que não tiverem essa temporalidade, o campo irá preenchido em branco.
- Na próxima versão do barramento, o TJDFR irá modificar o código para acrescentar a tabela que trata da temporalidade de acordo com o assunto documental. Para isso, é preciso que essa parte do código seja documentada, e assim, a equipe de TI do TJDFR consiga achar com facilidade a parte do código que será modificada.

Descrição: Determinar se a temporalidade do assunto é de cunho Permanente ou não.

Se for permanente, preencher o campo com a palavra "Permanente". Caso contrário, não preencher o campo.

Códigos de Assunto Permanente: 3613,3653,7661,5804,7667,7671,4656,4660,4670,4680,10499,10449,10450,10451,10452,10453,10454,10456,10457,10458,10459,10460,10500,10461,10469,10470,10480,10481,11412,11413,6206,6207,6213,6216,6218,6219,9565,7732,7735,7834,7835,7925,7926,9972,9973,9974,9975,9990,10012,1

0013,10014,10108,10109,10131,10121,10122,10123,10125,10126,1017,10128,10129,10130,10132,10133,10134,10135,10931,10932,10414,10416

Campo CSV: temporalidadeAssuntoProcessual

Exemplo: Permanente

Mimetype

- Texto fixo

Descrição: Tipo do arquivo que será enviado, seguindo o seguinte domínio: application/pdf ou html/text.

Campo CSV: mimetype

Exemplo: application/pdf

Meio Documental

- Texto fixo

Descrição: Para: documento, volume, processo, dossiê: usar Digital, Não Digital (que são os processos convencionais ou analógicos) ou Híbrido, para controlar as relações entre os meios e o monitoramento de preservação.

Campo CSV: meioDocumental

Exemplo: Digital

Dados referentes à classificação e temporalidade do processo não podem ser recuperadas, pois a Tabela de Temporalidade do órgão não está implementada no seu ambiente de gestão. Assim como metadados internos que seriam retornados do banco de dados do TJDF, como os responsáveis pelo cadastro ou publicação dos processos e as datas de criação e publicação. Já o gênero²³, espécie²⁴ e tipo documental²⁵, metadados que classificam o documento arquivisticamente, não serão aqui tratados, pois a variedade destas é extensa no PJe.

CAMPO	CAMPOS DC CSV	EXEMPLO DE PREENCHIMENTO	ONDE BUSCAR
Título	dc.title	Processo no 07193054-20.2019.8.07.0003	Nro do Processo Não precisa estar com máscara no numero do processo!
Criador	dc.creator	Tribunal de Justiça do Distrito Federal e dos Territórios	Texto fixo
Assunto	dc.subject	-	
Descrição	dc.description	-	
Editor	dc.publisher	Tribunal de Justiça do Distrito Federal e dos Territórios	Texto fixo
Contribuidor / Colaborador	dc.contributor		
Data	dc.date	2019-03-29	Campo vem dos metadados de gestão: Data de Ajuizamento <dataAjuizamento>
			Formato: AAAA-MM-DD ou AAAA-MM ou AAAA

²³ É a configuração da informação no documento de acordo com o sistema de signos utilizado na comunicação do documento. Exemplos: Audiovisual; textual; cartográfico; iconográfico; multimídia.

²⁴ Divisão de gênero documental que reúne tipos de documentos por seu formato. São exemplos de espécies documentais ata, carta, decreto, memorando, ofício, planta, relatório.

²⁵ É a configuração da espécie documental de acordo com a atividade que a gerou. Serve para complementar a descrição do documento ou a identificação do título. Exemplos: relatório de pesquisa; carta precatória; ofício-circular; prontuário médico; prontuário de funcionário.

CAMPO	CAMPOS DC CSV	EXEMPLO DE PREENCHIMENTO	ONDE BUSCAR
Tipo do Recurso	dc.type	Texto	Uma categoria para o tipo de conteúdo Exemplos: Collection, Dataset, Event, Image, InteractiveResource, MovingImage, PhysicalObject, Service, Software, Sound, StillImage, Text. Campo fixo
Formato	dc.format	application/pdf	Mimetype Campo fixo
Identificador do Recurso	dc.identifier	07193054-20.2019.8.07.0003	Nro do Processo Não precisa estar com máscara no número do processo! Um identificador único para o seu conteúdo. Pode ser um texto ou um número gerado por um sistema de identificação formal - ou apenas uma URL
Fonte	dc.source	Site de pesquisa do PJe: https://pje.tjdft.jus.br/consultapublica/ConsultaPublica/listView.seam	Onde o conteúdo foi inicialmente publicado / originado. Um conteúdo que tem alguma relação intelectual com o conteúdo descrito. Se for conteúdo original, veiculado pela primeira vez e sem nenhuma relação intelectual com outros conteúdos, deve ser a URL para esse conteúdo. Campo fixo
Idioma	dc.language	pt_BR	Em que linguagem o conteúdo foi escrito. Você precisa especificar o código correto da língua. Campo fixo
Relação	dc.relation		
Abrangência / Cobertura	dc.coverage	Distrito Federal	Onde o conteúdo está localizado fisicamente. Normalmente inclui localização espacial (nome do lugar ou coordenadas geográficas), período temporal (data) ou jurisdição. Campo fixo

CAMPO	CAMPOS DC CSV	EXEMPLO DE PREENCHIMENTO	ONDE BUSCAR
Gerenciamento de Direitos Autorais	dc.rights	Sigilo absoluto, acessível apenas ao magistrado do órgão em que tramita, aos servidores e demais usuários por ele indicado e às partes que provocaram o incidente.	Informação dos direitos autorais do conteúdo. Pode conter a declaração de Copyright ou outros tipos de licença. Campo vem dos metadados de gestão: Sigilo do Processo: <nívelSigilo - nomeSigilo>

Metadados Descritivos ISAD(G)²⁶

CAMPO	CAMPOS DC CSV	EXEMPLO DE PREENCHIMENTO	ONDE BUSCAR
Código(s) de Referência:	isad.identifier		Campo igual ao dc.identifier. Regras ISAD 3.1.1
Título	isad.title		Campo igual ao dc.title. Regras ISAD 3.1.2
Data(s)	isad.eventDates		Campo igual ao dc.date. Regras ISAD 3.1.3
Nível de Descrição	isad.levelOfDescription		Regras ISAD 3.1.4
Dimensão e Suporte	isad.extentAndMedium		Campo igual ao dc.format. Regras ISAD 3.1.5
Nome do(s) Produtor(es)	isad.eventActors	Tribunal de Justiça do Distrito Federal e dos Territórios	Regras ISAD 3.2.1 Campo fixo
História Administrativa / Biografia	isad.eventActorHistories		Regras ISAD 3.2.2
História Arquivística	isad.archivalHistory		Regras ISAD 3.2.3
Procedência	isad.acquisition	Tribunal de Justiça do Distrito Federal e dos Territórios	Regras ISAD 3.2.4 Campo fixo
Âmbito e Conteúdo	isad.scopeAndContent		Campo igual ao dc.description. Regras ISAD 3.3.1
Avaliação, Eliminação e Temporalidade	isad.appraisal	-Documento classificado com a classe de temporalidade permanente. -Documento classificado com assunto de temporalidade permanente.	Regras ISAD 3.3.2 Preencher o campo de acordo com o dado de temporalidadeNacionalAssunto e temporalidadeNacionalClasse que estão entre os metadados de gestão do processo.
Incorporações	isad.accruals		Regras ISAD 3.3.3
Sistemas de Arranjo	isad.arrangement		Regras ISAD 3.3.4
Condições de Acesso	isad.accessConditions		Campo igual ao dc.rights. Regras ISAD 3.4.1

²⁶ http://conarq.arquivonacional.gov.br/images/publicacoes_textos/isad_g_2001.pdf

CAMPO	CAMPOS DC CSV	EXEMPLO DE PREENCHIMENTO	ONDE BUSCAR
Condições de Reprodução	isad. reproductionConditions	Documento pode ser adquirido no site de pesquisa do PJe: https://pje.tjdft.jus.br/consultapublica/ConsultaPublica/listView.seam	Regras ISAD 3.4.2 Campo fixo
Idioma	isad.language	pt_BR	Campo igual ao dc.language. Regras ISAD 3.4.3 Campo fixo
Características Físicas e Requisitos Técnicos	isad. physicalCharacteristics	application/pdf	Regras ISAD 3.4.4 Campo fixo
Instrumentos de Pesquisa	isad.findingAids		Regras ISAD 3.4.5
Existência e Localização dos Originais	isad.locationOfOriginals		Campo igual ao dc.source. Regras ISAD 3.5.1
Existência e Localização de Cópias	isad.locationOfCopies		Regras ISAD 3.5.2
Unidades de Descrição Relacionadas	isad. relatedUnitsOfDescription		Regras ISAD 3.5.3
Nota sobre Publicação	isad.publicationNote		Regras ISAD 3.5.4
Notas	isad.generalNote		Regras ISAD 3.6.1
Nota do Arquivista	isad.archivistNote		Regras ISAD 3.7.1
Regras ou Convenções	isad.rules	Documento foi descrito baseado nas regras ISAD(G), do Conselho Internacional de Arquivos.	Regras ISAD 3.7.2 Campo fixo
Datas de criação, revisão e eliminação	isad.revisionHistory		Regras ISAD 3.7.3

Outros Metadados ISAD(G)

Esses metadados foram extraídos na planilha oficial da Artefactual, para importação de metadados ISAD(G) à plataforma de difusão AtoM.

CAMPO	CAMPOS DC CSV	EXEMPLO DE PREENCHIMENTO	ONDE BUSCAR
Locais	[isad.placeAccessPoints]		Campo igual ao dc.coverage.
Assuntos	[isad.subjectAccessPoints]		Campo igual ao dc.subject.
Pessoas / Entidades	[isad.nameAccessPoints]		Lista de pessoas ou entidades relacionadas ao documento.
Gênero	[isad.genreAccessPoints]		
Repositório	[isad.repository]	Tribunal de Justiça do Distrito Federal e dos Territórios	Campo igual ao dc.relation (isLocatedAt).

CAMPO	CAMPOS DC CSV	EXEMPLO DE PREENCHIMENTO	ONDE BUSCAR
Sistema de Escrita	[isad.script]	latim	Registrar a escrita do documento Campo fixo
Notas sobre o Idioma	[isad.languageNote]		Especificar qualquer tipo de alfabeto, escrita, sistema de símbolos ou abreviaturas utilizadas.
URI do Obj	[isad.digitalObjectURI]		URI enviada do archivematica para o AtoM, preenchida pelo sistema.
Checksum	[isad.digitalObjectChecksum]		Calculo matemático do documento.
Identificador da Descrição	[descriptionIdentifier]	Identificador da empresa responsável pela descrição	Identificador único dado pelo usuário da descrição.
Identificador da Instituição	[isad.institutionIdentifier]	Tribunal de Justiça do Distrito Federal e dos Territórios	Identificação da instituição responsável pela descrição.
Status da Descrição	[isad.descriptionStatus]	Draft	Registrar o estado da descrição, indicando se está em elaboração (rascunho), revisada ou final (concluída). Campo fixo
Nível de detalhamento da Descrição	[isad.levelOfDetail]	Minimal	Indicar o nível de detalhe da descrição (mínima, parcial ou completa) de acordo com as regras ou diretrizes nacionais e internacionais.
Idioma(s) da Descrição	[isad.languageOfDescription]	pt_BR	Indicar o idioma(s) utilizado para criar a descrição do material arquivístico Campo fixo
Escrita(s) da descrição	[isad.scriptOfDescription]	latim	Indicar a escrita utilizada para criar a descrição do material arquivístico. Campo fixo
Fontes	[isad.sources]	Site de pesquisa do PJe: https://pje.tjdft.jus.br/consultapublica/ConsultaPublica/listView.seam	Site do PJe Campo fixo
Status da publicação	[isad.publicationStatus]	Draft	Preencher entre Rascunho ou Publicado Campo fixo
Nome do Objeto físico	[isad.physicalObjectName]		Ex.: Caixa Nro 10
Endereçamento do objeto físico	[isad.physicalObjectLocation]		Ex.: Arquivo Judiciário, Módulo 05, Estante 03, Prateleira 01
Tipo do objeto físico	[isad.physicalObjectType]		Exemplos: caixa, ficha, pasta, prateleira, armário, etc.

CAMPO	CAMPOS DC CSV	EXEMPLO DE PREENCHIMENTO	ONDE BUSCAR
Identificador alternativo	[isad.alternativeIdentifiers]		Introduza um código de referência anterior, identificador alternativo, ou qualquer outra sequência alfanumérica associada ao registro.
Etiqueta do Id. alternativo	[isad.alternativeIdentifierLabels]		Digite um nome para o campo identificador alternativo que indique sua finalidade e uso. Ex. código de barras, AIP
Tipo de data	[isad.eventTypes]		Indique se são data(s) em que os documentos de arquivo foram acumulados pelo produtor no exercício das suas atividades, ou seja, integrados num sistema de arquivo, ou, se são data(s) em que os documentos foram produzidos, ou seja, que inclui a data de cópias, edições, versões, anexos ou originais de peças produzidas antes da sua acumulação ou integração no sistema de arquivo. Escolher entre Produção ou Acumulação.
Data Inicial	[isad.eventStartDates]		Indique a data de início.
Data Final	[isad.eventEndDates]		Indique a data final, se houver.
Descrição do evento	[isad.eventDescriptions]		
Data Tópica	[isad.eventPlaces]		
Cultura	[isad.culture]		Dado automático, preenchido pelo sistema.

Submission Documentation

Documentos

Identificador Documento

Descrição: Identificador do Documento

Campo CSV: idDocumento

Exemplo: 29119057

Identificador Documento Vinculado

Descrição: Identificador do documento principal caso este documento seja um anexo.

Campo CSV: idDocumentoVinculado

Exemplo: 29119055

Tipo de Documento - Código

Descrição: Código do tipo de documento disponível no serviço "ConsultaPJe" - Ex.: 58 - Petição Inicial, 118A - Informações. O código variará dependendo da instalação do PJe.

Campo CSV: codigoTipoDocumento

Exemplo: 58

Tipo de Documento - Nome

- A ser desenvolvido pela SDES/TJDFT

Descrição: Descrição do tipo de documento disponível no serviço "ConsultaPJe" - Ex.: 58 - Petição Inicial, 118A - Informações.

Campo CSV: nomeTipoDocumento

Exemplo: Petição Inicial, 118A - Informações.

Descrição

Descrição: Descrição do documento.

Campo CSV: descricao

Exemplo: Petição Inicial

Data/Hora

Descrição: Sequência de ano+dia+mês/hora+minuto+segundo. Atributo destinado a permitir a informação do momento em que o documento foi recebido pelo Poder Judiciário. É opcional no caso de documento enviado para protocolo.

Campo CSV: dataHora

Exemplo: 20190725181625

Mimetype

Descrição: Tipo do arquivo que será enviado, seguindo o seguinte domínio: application/pdf ou html/text.

Campo CSV: mimetype

Exemplo: html/text

Nível de Sigilo - Código

Descrição: Código do nível de sigilo a ser aplicado ao processo. Dever-se-á utilizar os seguintes níveis:

- 0: públicos, acessíveis a todos os servidores do Judiciário e dos demais órgãos públicos de colaboração na administração da Justiça, assim como aos advogados e a qualquer cidadão
- 1: segredo de justiça, acessíveis aos servidores do Judiciário, aos servidores dos órgãos públicos de colaboração na administração da Justiça e às partes do processo.
- 2: sigilo mínimo, acessível aos servidores do Judiciário e aos demais órgãos públicos de colaboração na administração da Justiça
- 3: sigilo médio, acessível aos servidores do órgão em que tramita o processo, à(s) parte(s) que provocou(ram) o incidente e àqueles que forem expressamente incluídos
- 4: sigilo intenso, acessível a classes de servidores qualificados (magistrado, diretor de secretaria/escrivão, oficial de gabinete/assessor) do órgão em que tramita o processo, às partes que provocaram o incidente e àqueles que forem expressamente incluídos
- 5: sigilo absoluto, acessível apenas ao magistrado do órgão em que tramita, aos servidores e demais usuários por ele indicado e às partes que provocaram o incidente.

Campo CSV: codigoNivelSigilo

Exemplo: 0

Nível de Sigilo - Nome

Descrição: Nível de sigilo a ser aplicado ao processo. Dever-se-á utilizar os seguintes níveis:

- 0: públicos, acessíveis a todos os servidores do Judiciário e dos demais órgãos públicos de colaboração na administração da Justiça, assim como aos advogados e a qualquer cidadão
- 1: segredo de justiça, acessíveis aos servidores do Judiciário, aos servidores dos órgãos públicos de colaboração na administração da Justiça e às partes do processo.
- 2: sigilo mínimo, acessível aos servidores do Judiciário e aos demais órgãos públicos de colaboração na administração da Justiça
- 3: sigilo médio, acessível aos servidores do órgão em que tramita o processo, à(s) parte(s) que provocou(ram) o incidente e àqueles que forem expressamente incluídos
- 4: sigilo intenso, acessível a classes de servidores qualificados (magistrado, diretor de secretaria/escrivão, oficial de gabinete/assessor) do órgão em que tramita o processo, às partes que provocaram o incidente e àqueles que forem expressamente incluídos
- 5: sigilo absoluto, acessível apenas ao magistrado do órgão em que tramita, aos servidores e demais usuários por ele indicado e às partes que provocaram o incidente.

Campo CSV: nomeNivelSigilo

Exemplo: Públicos, acessíveis a todos os servidores do Judiciário e dos demais órgãos públicos de colaboração na administração da Justiça, assim como aos advogados e a qualquer cidadão.

Identificador do Movimento

Descrição: Identificador do movimento que gerou o documento

Campo CSV: idMovimento

Exemplo: 28196207

Hash

Descrição: Resumo matemático do documento enviado pelo algoritmo SHA-256.

Campo CSV: hash

Exemplo: b0c020615d91c745a8cb54025cece674

Conteúdo XMLNS

Descrição: Arquivo convertido em Base64

Campo CSV: conteudo

Exemplo: xop:<http://www.w3.org/2004/08/xop/include>

Conteúdo HREF:CID

Descrição: Arquivo convertido em Base64

Campo CSV: conteudo

Exemplo: b9025e40-4fba-4bd8-bbf4-69fded72a9d0-5914@http%3A%2F%2Fwww.cnj.jus.br%2Fintercomunicacao-2.2.3

Assinatura

Descrição: Contém os dados em Base64 do arquivo de assinatura do documento. Para arquivos em PDF ou HTML, que não sejam P7S, é necessário informar os dados da assinatura.

Campo CSV: assinatura

Exemplo: assinaturadeteste

Data Assinatura

Descrição: Data e Hora da assinatura, obedecendo a sequência de ano+dia+mês+minuto+segundo.

Campo CSV: dataAssinatura

Exemplo: 20190725181340

Cadeia Certificado Assinatura

Descrição: String do certificado. Caso seja do tipo PEM é necessário incluir inclusive as strings "BEGIN CERTIFICATE" e "END CERTIFICATE".

Campo CSV: cadeiaCertificadoAssinatura

Exemplo: "-----BEGIN CERTIFICATE----- certchaindeteste= -----END CERTIFICATE-----"

Algoritmo Hash Assinatura

Descrição: Algoritmo utilizado para gerar a assinatura: MD5withRSA, SHA1withRSA, SHA256withRSA ou ASN1MD5withRSA. Atualmente, o mais utilizado é o SHA256withRSA.

Campo CSV: algoritmoHashAssinatura

Exemplo: MD5withRSA

Codificação Certificado Assinatura

Descrição: Codificação utilizada no certificado: "PEM" ou "PkiPath".

Campo CSV: codificacaoCertificadoAssinatura

Exemplo: PkiPath

Signatario Login

Descrição: Informação referente a signatário que fez uso de login e senha para assinar o documento.

Campo CSV: signatarioLoginAssinatura

Partes

Polo

Descrição: Sigla para identificação do tipo de polo. Para polo ativo, usar:

- "AT" = Ativo;
- "PA" = Passivo;
- "TC" = Terceiro;
- "FL" = Fiscal da lei diverso;
- "TJ" = Testemunha do juízo;
- "AD" = Assistente simples desinteressado (amicus curiae);
- "VI" = Vítima.

Campo CSV: polo

Exemplo: Ativo

Assistência Judiciária

Descrição: Atributo que define se o processo terá assistência judiciária às pessoas consideradas juridicamente pobres.

Se "true" = "O processo terá assistência judiciária às pessoas consideradas juridicamente pobres"

Se "false" = texto vazio

Campo CSV: assistenciaJudiciaria

Exemplo: O processo terá assistência judiciária às pessoas consideradas juridicamente pobres.

Nome Parte

Descrição: Nome da parte que compõe o polo do processo.

Campo CSV: nomeParte

Exemplo: DEUSUITA ALVES PEREIRA

Sexo Parte

Descrição: Correspondente a pessoa parte, onde:

- "M" masculino
- "F" feminino
- "D" desconhecido

Campo CSV: sexoParte

Exemplo: Feminino

Nome Genitor

Descrição: Nome da parte que compõe o polo do processo.

Campo CSV: nomeGenitor

Exemplo: MANOEL MESSIAS PEREIRA DA SILVA

Nome Genitora

Descrição: Nome da parte que compõe o polo do processo.

Campo CSV: nomeGenitora

Exemplo: DEUSELINA ALVES PEREIRA

Número Documento Principal

Descrição: Número do documento principal da pessoa parte, devendo ser utilizado o RIC ou o CPF para pessoas físicas, nessa ordem, ou o CNPJ para pessoas jurídicas.

O atributo é opcional em razão da possibilidade de haver pessoas sem documentos ou cujos dados não estão disponíveis.

Campo CSV: numeroDocumentoPrincipal

Exemplo: 82375046153

Tipo Pessoa

Descrição: Indicar qual o tipo de pessoa que será integrante do processo. Enviar um dos seguintes valores,

sem acentos: “física”; “jurídica”; “autoridade”; “orgão representacao”.

Campo CSV: tipoPessoa

Exemplo: Fisica

Nacionalidade

Descrição: País de nascimento da pessoa parte. Exemplo: “BR”-Brasil. Dever-se-á utilizar, preferencialmente, o código ISO-3166-1-alpha-2 (http://www.iso.org/iso/english_country_names_and_code_elements) pertinente ao país respectivo.

Campo CSV: nacionalidade

Exemplo: BR

COLOCAR OUTROS DADOS DA PARTE

Código do Documento

Descrição: Número do documento identificador vinculado ao tipo de pessoa, podendo ser o número do CPF, CNPJ, RG, TIT, etc. São uma sequência de caracteres que tornam único o documento descrito sem os separadores característicos.

Campo CSV: codigoDocumento

Exemplo: 12194262038

Emissor Documento

Descrição: Entidade emissora do documento. Exemplo: SSP, Cartórios, Justiça Eleitoral, SRFB.

Campo CSV: emissorDocumento

Exemplo: Justiça Eleitoral

Tipo Documento

Descrição: Elemento destinado a permitir a identificação dos diversos tipos de documentos de identificação existentes. Trata-se de enumeração dos seguintes tipos:

- CI: carteira de identidade
- CNH: carteira nacional de habilitação
- TE: título de eleitor
- CN: certidão de nascimento
- CC: certidão de casamento
- PAS: passaporte
- CT: carteira de trabalho
- RIC: registro individual do cidadão
- CMF: cadastro no Ministério da Fazenda Brasileiro
- PIS_PASEP: número no programa de integração social
- CEI: cadastro específico do INSS
- NIT: número de identificação do trabalho
- CP: cadastro em conselhos profissionais
- IF: identidade funcional
- OAB: número de cadastro na Ordem dos Advogados do Brasil
- RJC: número de inscrição empresarial;
- RGE: registro de identificação do estrangeiro

Campo CSV: tipoDocumento

Exemplo: Título de Eleitor

Nome Apresentado no Documento

Descrição: Nome da parte detentora do documento.

Campo CSV: nomeDocumento

Exemplo: DEUSUITA ALVES PEREIRA

Outro Nome

Descrição: Elemento destinado a permitir a inclusão de outros nomes da pessoa individualizada. Devem ser

incluídos sob esse elemento os nomes que NÃO constam em documentos específicos. Em caso de nomes divergentes do principal existentes em documentos específicos, esses nomes devem ser indicados no atributo pertinente do documento.

Campo CSV: outroNome

CEP Endereço Parte

Descrição: CEP da parte do processo. Atributo indicador do código de endereçamento postal do endereço no diretório nacional de endereços da ECT. O valor deverá ser uma sequência de 8 dígitos, sem qualquer separador. O atributo é opcional para permitir a apresentação de endereços desprovidos de CEP e de endereços internacionais.

Campo CSV: cepEnderecoParte

Exemplo: 70094900

Logradouro Endereço Parte

Descrição: Logradouro vinculado ao CEP, tais como as ruas, avenidas, praças, jardins, etc.

Campo CSV: logradouroEnderecoParte

Exemplo: Praça do Buriti, Lote 2, Sede do MPDFT

Número Endereço Parte

Descrição: Número vinculado ao CEP, podendo ser número da casa, apartamento, etc.

Campo CSV: numeroEnderecoParte

Exemplo: 12

Complemento Endereço Parte

Descrição: Dados adicionais do endereço.

Campo CSV: complementoEnderecoParte

Exemplo: Eixo Monumental

Bairro Endereço Parte

Descrição: Parte que compõe a cidade, devendo ser vinculada ao CEP.

Campo CSV: bairroEnderecoParte

Exemplo: Asa Norte

Cidade Endereço Parte

Descrição: Vinculada ao CEP, ou seja, nome da zona geográfica onde se agrupam as residências, comércios e indústrias.

Campo CSV: cidadeEnderecoParte

Exemplo: Brasília

Estado Endereço Parte

Descrição: Sigla do Estado brasileiro, composto por duas letras.

Ex: DF.

Campo CSV: estadoEnderecoParte

Exemplo: DF

País Endereço Parte

Descrição: Sigla referente ao país do Estado que compõe o endereço, composto por duas letras. Ex: BR. Dever-se-á utilizar, preferencialmente, o código ISO-3166-1-alpha-2²⁷ pertinente ao país respectivo.

Campo CSV: paisEnderecoParte

Exemplo: Brasil

Representante Processual - Intimação

Descrição: Indicativo verdadeiro (true) ou falso (false) relativo à escolha de o advogado, escritório ou órgão de representação ser o(s) preferencial(is) para a realização de intimações.

²⁷ http://www.iso.org/iso/english_country_names_and_code_elements

Se for false: "O advogado, escritório ou órgão de representação não são preferenciais para a realização de intimações."

Se for true: "O advogado, escritório ou órgão de representação são preferenciais para a realização de intimações."

Campo CSV: representanteIntimacao

Exemplo: O advogado, escritório ou órgão de representação não são preferenciais para a realização de intimações.

Representante Processual - Nome

Descrição: Nome do advogado ou do escritório de advocacia ou órgão de representação processual (MP, advocacia pública ou defensoria pública. A identificação relativa a se tratar de advogado ou de escritório de advocacia será feita pelo numeroDocumentoPrincipal, caso se trate de CPF, deverá ser tratado como advogado, caso se trate de CNPJ, como escritório de advocacia ou órgão de representação de interesses (MP, advocacia pública e defensoria pública).

Campo CSV: representanteNome

Exemplo: IGOR DE SOUSA SILVA TAVARES

Representante Processual - Inscrição

Descrição: Inscrição do advogado ou escritório de advocacia no cadastro da Ordem dos Advogados do Brasil. O campo deve ser preenchido no formato CCDDDDDDDC, sendo os primeiros 'C' dois caracteres indicativos da unidade federativa brasileira em que há a inscrição, os sete dígitos 'D' seguintes o número da inscrição, devendo incluir zeros ('0') à esquerda caso o número tenha menos de 7 dígitos e o último caracter a letra identificadora do tipo de inscrição.

Campo CSV: representanteInscricao

Exemplo: DF5846400A

Representante Processual - Documento

Descrição: O número RIC ou o CPF, para advogados, ou o CNPJ, para escritórios de advocacia e órgãos de representação processual (MP, advocacia pública e defensoria pública).

Campo CSV: representanteNroDocPrincipal

Exemplo: 04905188180

Representante Processual - Tipo Representante

Descrição: Atributo que permite indicar o tipo de representante processual.

- "A" advogado;
- "E" Escritório de advocacia;
- "M" Ministério público;
- "D" Defensoria pública;
- "P" outros órgãos de representação processual pública (advocacia pública em geral).

Campo CSV: tipoRepresentante

Exemplo: Advogado

Assistência Judiciária

Descrição: Atributo que define se o processo terá assistência judiciária às pessoas consideradas juridicamente pobres.

Se "true" = "O processo terá assistência judiciária às pessoas consideradas juridicamente pobres"

Se "false" = texto vazio

Campo CSV: assistenciaJudiciaria

Exemplo: O processo terá assistência judiciária às pessoas consideradas juridicamente pobres.

Movimentação Processual

Data/Hora

Descrição: Sequência de ano+dia+mês/hora+minuto+segundo. Este atributo corresponde a data da movimentação do documento.

Campo CSV: datahora
Exemplo: 20190319181411

Nível de Sigilo - Código

Descrição: Código do nível de sigilo a ser aplicado ao processo. Dever-se-á utilizar os seguintes níveis:

- 0: públicos, acessíveis a todos os servidores do Judiciário e dos demais órgãos públicos de colaboração na administração da Justiça, assim como aos advogados e a qualquer cidadão
- 1: segredo de justiça, acessíveis aos servidores do Judiciário, aos servidores dos órgãos públicos de colaboração na administração da Justiça e às partes do processo.
- 2: sigilo mínimo, acessível aos servidores do Judiciário e aos demais órgãos públicos de colaboração na administração da Justiça
- 3: sigilo médio, acessível aos servidores do órgão em que tramita o processo, à(s) parte(s) que provocou(ram) o incidente e àqueles que forem expressamente incluídos
- 4: sigilo intenso, acessível a classes de servidores qualificados (magistrado, diretor de secretaria/escrivão, oficial de gabinete/assessor) do órgão em que tramita o processo, às partes que provocaram o incidente e àqueles que forem expressamente incluídos
- 5: sigilo absoluto, acessível apenas ao magistrado do órgão em que tramita, aos servidores e demais usuários por ele indicado e às partes que provocaram o incidente.

Campo CSV: codigoNivelSigilo

Exemplo: 0

Nível de Sigilo - Nome

Descrição: Nível de sigilo a ser aplicado ao processo. Dever-se-á utilizar os seguintes níveis:

- 0: públicos, acessíveis a todos os servidores do Judiciário e dos demais órgãos públicos de colaboração na administração da Justiça, assim como aos advogados e a qualquer cidadão
- 1: segredo de justiça, acessíveis aos servidores do Judiciário, aos servidores dos órgãos públicos de colaboração na administração da Justiça e às partes do processo.
- 2: sigilo mínimo, acessível aos servidores do Judiciário e aos demais órgãos públicos de colaboração na administração da Justiça
- 3: sigilo médio, acessível aos servidores do órgão em que tramita o processo, à(s) parte(s) que provocou(ram) o incidente e àqueles que forem expressamente incluídos
- 4: sigilo intenso, acessível a classes de servidores qualificados (magistrado, diretor de secretaria/escrivão, oficial de gabinete/assessor) do órgão em que tramita o processo, às partes que provocaram o incidente e àqueles que forem expressamente incluídos
- 5: sigilo absoluto, acessível apenas ao magistrado do órgão em que tramita, aos servidores e demais usuários por ele indicado e às partes que provocaram o incidente.

Campo CSV: nomeNivelSigilo

Exemplo: Públicos, acessíveis a todos os servidores do Judiciário e dos demais órgãos públicos de colaboração na administração da Justiça, assim como aos advogados e a qualquer cidadão.

Identificador Movimento

Descrição: Identificador do movimento do documento

Campo CSV: identificadorMovimento

Exemplo: 28196207

Movimento Nacional - Código

Descrição: Código da movimentação processual, segundo o padrão estabelecido no Sistema de Gestão de Tabelas Processuais Unificadas²⁸

Campo CSV: movimentoCodigoNacional

Exemplo: 581

²⁸ https://www.cnj.jus.br/sgt/consulta_publica_classes.php

Movimento Nacional - Nome

Descrição: Descrição da movimentação processual, segundo o padrão estabelecido no Sistema de Gestão de Tabelas Processuais Unificadas.²⁹

Campo CSV: movimentoNomeNacional

Exemplo: Juntada - Documento

Complemento Movimento

Descrição: Texto complementar à descrição da movimentação processual.

Campo CSV: complemento

Exemplo: Juntada de certidão

Documento Vinculado

Descrição: Identificador do documento vinculado

Campo CSV: idDocumentoVinculado

Exemplo: 30507293

Referências

CONARQ, C. (2006). Resolução nº 24, de 03 de agosto de 2006. **Estabelece diretrizes para a transferência e recolhimento de documentos arquivísticos digitais para instituições arquivísticas públicas**. Acesso em 05 de abr de 2019, disponível em <http://conarq.arquivonacional.gov.br/resolucoes-do-conarq/266-resolucao-n-24,-de-3-de-agosto-de-2006.html>

CONARQ, C. N., & CTDE, C. T. (2015). **Diretrizes para a Implementação de Repositórios Arquivísticos Digitais Confiáveis – RDC-Arq**. Rio de Janeiro. Acesso em 05 de abr de 2019, disponível em http://www.conarq.gov.br/images/publicacoes_textos/diretrizes_rdc_arq.pdf

CONARQ, Conselho Nacional de Arquivos. (2000). ISAD(G): Norma geral internacional de descrição arquivística. (Segunda Edição). Rio de Janeiro, RJ. Acesso em 10 de mai de 2019, disponível em http://conarq.arquivo-nacional.gov.br/images/publicacoes_textos/isad_g_2001.pdf

CONARQ, Conselho Nacional de Arquivos; CTDE, Câmara Técnica de Documentos Eletrônicos. (2011). Modelo de requisitos para sistemas informatizados de gestão arquivística de documentos: e-ARQ. Rio de Janeiro. Acesso em 05 de abr de 2019, disponível em <http://www.siga.arquivonacional.gov.br/images/publicacoes/e-arq.pdf>

Conselho Nacional de Justiça. (agosto de 2009). Modelo de Requisitos para Sistemas Informatizados de Gestão de Processos e Documentos do Judiciário Brasileiro – MoReq-Jus. (v.1.0). Brasília, DF, Brasil. Acesso em 12 de abr de 2019, disponível em http://www.cnj.jus.br/images/stories/docs_cnj/resolucao/manualmoreq.pdf

Dublin Core Metadata Initiative. (14 de jun de 2012). Dublin Core Metadata Element Set, Version 1.1: Reference Description. Estados Unidos. Acesso em 10 de mai de 2019, disponível em <http://www.dublincore.org/specifications/dublin-core/dces/>

InterPARES. (2013). Diretrizes do Produtor – A Elaboração e a Manutenção de Materiais Digitais: Diretrizes Para Indivíduos e Diretrizes do Preservador – A Preservação de Documentos Arquivísticos digitais: Diretrizes para Organizações. (A. N. Deputados, Trad.) Vancouver, British Columbia, Canadá: Câmara dos Deputados. Acesso em 07 de 04 de 2019, disponível em http://conarq.gov.br/images/publicacoes_textos/Diretrizes_produtores_preservadores.pdf

RONDINELLI, R. C. (2002). **Gerenciamento Arquivístico de Documentos Eletrônicos** (1a ed. ed.). Rio de Janeiro, RJ, Brasil: Editora FGV. Acesso em 08 de 04 de 2019

UC Santa Cruz. (s.d.). **University Library**. Acesso em 08 de 04 de 2019, disponível em UC Santa Cruz University Library: <https://guides.library.ucsc.edu/c.php?g=618773>

²⁹ https://www.cnj.jus.br/sgt/consulta_publica_classes.php

Este manual descreve o procedimento de instalação do Archivematica 1.11 em Red Hat Enterprise Linux 7 customizado conforme especificação do TJDF, tendo as seguintes características:

- Banco de dados segregado em máquina dedicada
- Elasticsearch segregado em máquina dedicada
- MCP Client(s) segregado(s) em máquinas dedicadas

NOTA: a segregação do(s) MCP Client(s) depende de um compartilhamento de pasta por NFS. Existem relatos (<https://github.com/archivematica/Issues/issues/988>) que essa integração causa problemas intermitentes em cenários de alta carga do sistema.

Instalação do SGBD

Na máquina que hospedará o banco de dados, atualize o sistema:

```
sudo yum -y update
```

Dê o comando `subscription-manager attach` se o `update` aparecer erros de *mirror*.

Instale, habilite e ative o MariaDB:

```
sudo -u root yum install -y mariadb-server
sudo -u root systemctl enable mariadb sudo
-u root systemctl start mariadb
```

Embora o MariaDB tenha sido originado a partir do MySQL, recomenda-se utilizá-lo por ser o banco recomendado pelo fabricante para instalações em Red Hat Enterprise Linux.

Crie o usuário e o banco do Archivematica, substituindo a `SENHA_DO_BANCO`

```
sudo -H -u root mysql -hlocalhost -uroot -e "DROP DATABASE IF EXISTS MCP; CREAT
sudo -H -u root mysql -hlocalhost -uroot -e "CREATE USER 'archivematica'@'%' ID
sudo -H -u root mysql -hlocalhost -uroot -e "GRANT ALL ON MCP.* TO 'archivemati
```

Libere a porta 3306 para conexões externas:

```
firewall-cmd --permanent --add-port=3306/tcp
firewall-cmd --reload
```

Idealmente, os components do Archivematica deveriam se comunicar com a máquina do banco de dados por uma rede segregada, sem conexão com a internet. Entretanto, se esse não for o caso, ao invés de liberar a porta do banco de dados para qualquer conexão externa, libere restringindo pelos IPs de origem:

```
firewall-cmd --permanent --new-zone=archivematica firewall-cmd
--permanent --zone=archivematica --add-port=3306/tcp

firewall-cmd --permanent --zone=archivematica
--add-source=IP_DO_ARCHIVEMATICA/ firewall-cmd --permanent
--zone=archivematica --add-source=IP_DO_MCP_CLIENT/32 firewall-cmd --reload
```

Repita o comando de liberação por IP para cada MCP Client

Use `firewall-cmd --permanent --zone=archivematica --add-source=SUBNET/MASK` para liberar a subrede inteira onde ficarão o Archivematica e os MCP Clients

Instalação do Elasticsearch

Na máquina que hospedará o elasticsearch, atualize o sistema:

```
sudo yum -y update
```

Dê o comando `subscription-manager attach` se o `update` aparecer erros de *mirror*. Se aparecer erros de *Protected Multilib Versions*, seguir os passos da resposta 03 do link

(<https://stackoverflow.com/questions/34501760/centos-yum-install-fails-protected-multilib-versions-problems-found-libsel>).

Adicione a chave de assinatura de pacotes do *elasticsearch*:

```
sudo -u root rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch
```

Adicione o repositório elasticsearch (copie o comando todo de uma só vez):

```
sudo -u root bash -c 'cat << EOF > /etc/yum.repos.d/elasticsearch.repo
[elasticsearch-6.x]

name=Elasticsearch repository for 6.x packages

baseurl=https://artifacts.elastic.co/packages/6.x/yum

gpgcheck=1

gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch

enabled=1

autorefresh=1

type=rpm-md

EOF'
```

Instale o *elasticsearch*:

```
sudo -u root yum install -y java-1.8.0-openjdk-headless elasticsearch
```

Edite o arquivo de configuração do elasticsearch:

```
nano /etc/elasticsearch/elasticsearch.yml
```

Na seção *Network*, descomente a linha *network.host* e substitua o valor por 0.0.0.0 :

```
#Set the bind address to a specific IP (IPv4 or IPv6):
#
network.host: 0.0.0.0
...
```

Habilite e inicie o *elasticsearch*:

```
sudo -u root systemctl enable
elasticsearch sudo -u root systemctl start
elasticsearch
```

Libere a porta 9200 para conexões externas:

```
sudo firewall-cmd --permanent --add-port=9200/tcp
sudo firewall-cmd --reload
```

Idealmente, os componentes do Archivematica deveriam se comunicar com a máquina do Elasticsearch por uma rede segregada, sem conexão com a internet. Entretanto, se esse não for o caso, ao invés de liberar para qualquer conexão externa, libere restringindo pelos IPs de origem:

```
sudo firewall-cmd--permanent --new-zone=archivematica
sudo firewall-cmd--permanent --zone=archivematica --add-port=9200/tcp
--permanent --zone=archivematica
sudo firewall-cmd--add-source=IP_DO_ARCHIVEMA
--permanent --zone=archivematica
sudo firewall-cmd--add-source=IP_DO_MCP_CLIEN
sudo firewall-cmd --reload
```

Repita o comando de liberação por IP para cada MCP Client

Use `firewall-cmd --permanent --zone=archivematica --add-source=SUBNET/MASK` para liberar a subrede inteira onde ficarão o Archivematica e os MCP Clients

Instalação do Archivematica

Na máquina que hospedará o Archivematica, siga os passos abaixo.

Comece atualizando seu sistema:

```
sudo yum -y update
```

SELinux

Verifique se o seu ambiente usa SELinux:

<!-- <https://www.thegeekdiary.com/how-to-check-whether-selinux-is-enabled-or-disabled/> (<https://www.thegeekdiary.com/how-to-check-whether-selinux-is-enabled-or-disabled/>) -->

```
getenforce
```

```
sestatus
```

Caso o seu sistema utilize SELinux, será necessário pelo menos fazer as configurações descritas abaixo. Caso contrário, ignore o restante desta seção.

Permita que o Nginx use as portas 81 e 8001:

```
sudo semanage port -m -t http_port_t -p tcp 81
```

```
sudo semanage port -a -t http_port_t -p tcp
8001
```

```
<!-- https://www.tecmint.com/fix-semanage-command-not-found-in-centos-rhel/
(https://www.tecmint.com/fix-semanage-command-not-found-in-centos-rhel/) --> <!-- yum provides/
usr/sbin/semanage -->
```

Se o comando `semanage` não estiver instalado, execute `yum install policycoreutils-python`

Permita que o Nginx se conecte ao MySQL server e ao Unicorn:

```
sudo setsebool -P
httpd_can_network_connect_db=1 sudo setsebool
-P httpd_can_network_connect=1
```

Permita que o Nginx mude os limites do sistema:

```
sudo setsebool -P httpd_setrlimit 1
```

Dependências

Habilite o repositório de pacotes extras do Linux Enterprise:

```
sudo yum install
https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.no sudo yum
install -y epel-release
```

Adicione o repositório *archivematica* (copie o comando todo de uma só vez):

```
sudo -u root bash -c 'cat << EOF > /etc/yum.repos.d/archivematica.repo
[archivematica]
name=archivematica
baseurl=https://packages.archivematica.org/1.11.x/centos
gpgcheck=1
gpgkey=https://packages.archivematica.org/1.11.x/key.asc
enabled=1
EOF'
```

Adicione o repositório *archivematica-extras* (copie o comando todo de uma só vez):

```
sudo -u root bash -c 'cat << EOF > /etc/yum.repos.d/archivematica-extras.repo
[archivematica-extras]
name=archivematica-extras
baseurl=https://packages.archivematica.org/1.11.x/centos-extra
gpgcheck=1
gpgkey=https://packages.archivematica.org/1.11.x/key.asc
enabled=1
EOF'
```

Instale, habilite e inicie o Gearmand:

```
sudo -u root yum install -y java-1.8.0-openjdk-headless gearmand
sudo -u root systemctl enable gearmand sudo -u root systemctl
start gearmand
```

Instalação do Archivematica Storage Service

Instale os pacotes:

```
sudo -u root yum install -y python-pip archivematica-storage-service
```

Nesse passo, pode acontecer o erro *Package does not match intended download*. Nesse caso, execute `sudo yum clean all` e tente novamente.

Depois, popule a base de dados SQLite e pegue alguns arquivos estáticos usados pelo Django (copie o comando todo):

```
sudo -u archivematica bash -c "\
set -a -e -x

source /etc/sysconfig/archivematica-storage-service cd
/usr/lib/archivematica/storage-service
/usr/share/archivematica/virtualenvs/archivematica-storage-service/bin/python m
```

O banco de dados SQLite utilizado pelo Storage Service tem muito pouco volume, e por isso não será segregado

Em seguida habilite e inicie os serviços:

```
sudo -u root systemctl enable archivematica-storage-service
sudo -u root systemctl start archivematica-storage-service
sudo -u root systemctl enable nginx sudo -u root systemctl
start nginx

sudo -u root systemctl enable rngd

sudo -u root systemctl start rngd
```

O Storage Service estará disponível em `http://HOSTNAME:8001` (`http://HOSTNAME:8001`)

Instalação do Archivematica Dashboard e MCP Server

Instale os pacotes:

```
sudo -u root yum install -y archivematica-common archivematica-mcp-server archi
```

Altere as configurações de conexão com o banco de dados do Dashboard:

```
nano /etc/sysconfig/archivematica-dashboard
```

Modifique os seguintes trechos:

```
...
ARCHIVEMATICA_DASHBOARD_CLIENT_PASSWORD=<senha-banco-de-dados>
ARCHIVEMATICA_DASHBOARD_CLIENT_HOST=<hostname-maquina-banco-de-dados>
...
ARCHIVEMATICA_DASHBOARD_DASHBOARD_ELASTICSEARCH_SERVER=<hostname-maquina-elasti
...
```

Note que a última configuração não é escrita como

`ARCHIVEMATICA_DASHBOARD_ELASTICSEARCH` (pode estar assim no arquivo existente), mas sim `ARCHIVEMATICA_DASHBOARD_DASHBOARD_ELASTICSEARCH_SERVER`. O exemplo dado no arquivo está incorreto conforme esse relato de bug (<https://github.com/archivematica/Issues/issues/1118>).

Altere as configurações de conexão com o banco de dados do MCP Server:

```
nano /etc/sysconfig/archivematica-mcp-server
```

Modifique os seguintes trechos:

```
...
ARCHIVEMATICA_MCPSERVER_CLIENT_HOST=<hostname-maquina-banco-de-dados>
...
ARCHIVEMATICA_MCPSERVER_CLIENT_PASSWORD=<senha-banco-de-dados>
...
```

Utilizando o usuário *archivematica*, execute as migrações sobre o banco de dados (copie o comando todo). Antes de executar esse comando, certifique-se de que máquina do SGBD está ligada:

```
sudo -u archivematica bash -c " \
set -a -e -x
source /etc/sysconfig/archivematica-dashboard
cd /usr/share/archivematica/dashboard
/usr/share/archivematica/virtualenvs/archivematica-dashboard/bin/python manage.
";
```

Esse passo pode levar alguns minutos

Inicie e ative os serviços:

```
sudo -u root systemctl enable archivematica-mcp-server
sudo -u root systemctl start archivematica-mcp-server
sudo -u root systemctl enable archivematica-dashboard
sudo -u root systemctl start archivematica-dashboard
```

Reinicie o Nginx para carregar o arquivo de configuração do dashboard:

```
sudo -u root systemctl restart nginx
```

O dashboard estará disponível em `http://HOSTNAME:81` (`http://HOSTNAME:81`)

Compartilhamento do Archivematica shared directory via NFS

Instale o pacote `nfs-utils` :

```
sudo -u root yum install -y nfs-utils
```

Edite o arquivo `/etc/exports` :

```
sudo nano /etc/exports
```

Adicione a linha abaixo (substitua o `<MCP_CLIENT_HOST>`)


```
/var/archivematica/sharedDirectory <MCP_CLIENT_HOST>(rw, sync, no_subtree_check)
```

Use `/var/archivematica/sharedDirectory CLIENT1(rw, sync, no_subtree_check) CLIENT2(-rw, sync, no_subtree_check)` e assim sucessivamente se houver mais de um MCP Client

Use `/var/archivematica/sharedDirectory SUBNET/MASK(rw, sync, no_subtree_check)` para liberar acesso à subrede inteira onde estarão os MCP Clients, por exemplo, `/var/archivematica/sharedDirectory 10.1.1.0/24(rw, sync, no_subtree_check)`, para liberar qualquer IP do 10.1.1.0 ao 10.1.1.255.

Recarregue o arquivo de exports, habilite o servidor de NFS e inicie-o:

```
sudo exportfs -r
sudo systemctl enable nfs
sudo systemctl start nfs
```

Em seguida, conceda acesso no firewall ao MCP Server e ao NFS:

```
sudo firewall-cmd --permanent --add-port=4730/tcp
sudo firewall-cmd --permanent --add-service=nfs
sudo firewall-cmd --permanent
--add-service=mountd sudo firewall-cmd --reload
```

Idealmente, o(s) MCP Client(s) deveria(m) se comunicar com a máquina do Archivematica por uma rede segregada, sem conexão com a internet. Entretanto, se esse não for o caso, ao invés de liberar o acesso para qualquer conexão externa, libere restringindo pelo IP de origem:

Repita o comando de liberação por IP para cada MCP Client

Use `firewall-cmd --permanent --zone=mcpclients --add-source=SUBNET/MASK` para liberar a subrede inteira onde ficarão os MCP Clients, por exemplo, `firewall-cmd --permanent --zone=mcpclients --add-source=10.0.0.0/8`, para liberar todos os IPs de 10.0.0.0 ao 10.255.255.255.

Por fim, tome nota do *id* do usuário e grupo archivematica :

```
sudo id -u archivematica
sudo id -g archivematica
```

Instalação do(s) MCP Client(s)

Repita os passos dessa seção para cada MCP Client a ser instalado.
Comece atualizando o sistema:

```
sudo yum -y update
```

Habilite o repositório de pacotes extras do Linux Enterprise:

```
sudo yum install
https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.no sudo yum
install -y epel-release
```

Adicione o repositório *archivematica* (copie o comando todo de uma só vez):

```
sudo -u root bash -c 'cat << EOF > /etc/yum.repos.d/archivematica.repo
```

```
[archivematica]
name=archivematica
baseurl=https://packages.archivematica.org/1.11.x/centos
gpgcheck=1
gpgkey=https://packages.archivematica.org/1.11.x/key.asc
enabled=1
EOF'
```

Adicione o repositório archivematica-extras (copie o comando todo de uma só vez):

```
sudo -u root bash -c 'cat << EOF > /etc/yum.repos.d/archivematica-extras.repo
[archivematica-extras]
name=archivematica-extras
baseurl=https://packages.archivematica.org/1.11.x/centos-extras
gpgcheck=1
gpgkey=https://packages.archivematica.org/1.11.x/key.asc
enabled=1
EOF'
```

Instale o pacote *archivematica-mcp-client*:

```
sudo -u root yum install -y archivematica-dashboard archivematica-mcp-client
```

Embora pareça estranho, é necessário instalar o pacote do dashboard na máquina do MCP Client por causa de um bug do Archivematica (<https://github.com/archivematica/issues/issues/564>)

O comando anterior cria o usuário e grupo archivematica . Confira se o id do usuário e do grupo ficaram iguais aos do servidor do Archivematica (isso é importante para a integração via NFS):

```
sudo id -u archivematica
sudo id -g archivematica
```

O MCP Client espera alguns programas em certos caminhos, por isso execute:

```
sudo ln -sf /usr/bin/7za /usr/bin/7z
```

Ajuste a configuração do ClamAV:

```
sudo -u root sed -i 's/^#TCPSToken/TCPSToken/g' /etc/clamd.d/scan.conf
sudo -u root sed -i 's/^Example//g' /etc/clamd.d/scan.conf
```

Altere as configurações de conexão com o banco de dados:

```
nano /etc/sysconfig/archivematica-mcp-client
```

Modifique os seguintes trechos:

```
...
ARCHIVEMATICA_MCPCLIENT_CLIENT_HOST=<hostname-maquina-banco-de-dados>

ARCHIVEMATICA_MCPCLIENT_CLIENT_DATABASE=MCP
ARCHIVEMATICA_MCPCLIENT_CLIENT_USER=archivematica
ARCHIVEMATICA_MCPCLIENT_CLIENT_PASSWORD=<senha-banco-de-dados>
> ...
```

```
ARCHIVEMATICA_MCPCLIENT_MCPCLIENT_ELASTICSEARCHSERVER=<hostname-do-elasticsearc
ARCHIVEMATICA_MCPCLIENT_MCPCLIENT_MCPARCHIVEMATICA SERVER=<hostname-do-archivema
...
```

Altere também o arquivo `/etc/archivematica/clientConfig.conf`:

```
nano /etc/archivematica/clientConfig.conf

[MCPCClient]
MCPArchivematicaServer = <hostname-do-archivematica>:4730
...
elasticsearchServer = <hostname-maquina-elasticsearch>:9200
...
[client]

user = archivematica
password = <senha-banco-de-dados>
host = <hostname-maquina-banco-de-dados>
database = MCP
port = 3306
engine = django.db.backends.mysql
```

Instale o pacote `nfs-utils` :

```
sudo -u root yum install -y nfs-utils
```

Edite o arquivo `/etc/fstab` :

```
sudo nano /etc/fstab
```

Adicione a seguinte linha (substitua o `<HOSTNAME_DO_ARCHIVEMATICA>`):

```
<HOSTNAME_DO_ARCHIVEMATICA>:/var/archivematica/sharedDirectory /var/archivemati
```

Monte o compartilhamento:

```
sudo mount -a
```

Teste o acesso:

```
sudo -u archivematica touch /var/archivematica/sharedDirectory/foo
```

No servidor do Archivematica, verifique se o arquivo aparece no servidor com usuário e grupo `archivematica`, depois apague-o:

```
sudo ls -l /var/archivematica/sharedDirectory
sudo rm
/var/archivematica/sharedDirectory/foo
```

De volta ao servidor do MCP Client, habilite e inicie os serviços:

```
sudo -u root systemctl enable archivematica-mcp-client
sudo -u root systemctl start archivematica-mcp-client
sudo -u root systemctl enable fits-nailgun sudo -u
root systemctl start fits-nailgun
```

```
sudo -u root systemctl enable clamd@scan
sudo -u root systemctl start clamd@scan
```

Caso queira executar mais de um MCP Client por máquina, faça quantas cópias desejar do arquivo de definição de serviço do MCP Client:

```
sudo cp /usr/lib/systemd/system/archivematica-mcp-client.service /usr/lib/syste
```

Convém executar `ls -l /usr/lib/systemd/system/archivematica-mcp-client*.service` e conferir se as permissões e *ownership* das cópias ficaram iguais ao do original

A configuração ideal de quantidade de instâncias por máquina depende do tipo de documentos que serão submetidos, de modo que o dimensionamento ideal só poderá ser determinado empiricamente.

Em seguida, habilite e inicie os serviços:

```
sudo -u root systemctl enable
archivematica-mcp-client-N sudo -u root systemctl start
archivematica-mcp-client-N
```

Passos finais

Caso a máquina do Archivematica esteja com firewall habilitado (bastante provável no Red Hat), será necessário abrir as portas 81 e 8001 para conexões externas:

```
sudo firewall-cmd --permanent --add-port=81/tcp
sudo firewall-cmd --permanent --add-port=8001/tcp
sudo firewall-cmd --reload
```

Pós-instalação

Após concluir a instalação, siga os passos descritos em Post-install configuration (<https://www.archivematica.org/en/docs/archivematica-1.11/admin-manual/installation-setup/installation/install-centos/#post-install-configuration>) normalmente.

Fontes

<https://www.archivematica.org/en/docs/archivematica-1.11/admin-manual/installation-setup/installation/install-centos/#install-pkg-centos>

<https://www.archivematica.org/en/docs/archivematica-1.11/admin-manual/installation-setup/installation/install-centos/#install-pkg-cento>

Integração de AtoM em Docker com Archivematica em VM

Para integrar um AtoM versão 2.5 instalado em Docker com um Archivematica versão 1.10 instalado em VM, siga os passos abaixo.

Archivematica server

Menu

Gere um par de chaves de SSH para o Archivematica autenticar-se no servidor de rsync do AtoM. Quando solicitado por um caminho para salvar o arquivo, apenas pressione ENTER. Faça o mesmo quando for solicitada uma senha (*passphrase*).

```
sudo -u archivematica ssh-keygen
```

Copie o conteúdo de `/var/lib/archivematica/.ssh/id_rsa.pub` para um bloco de notas.

AtoM host server

No servidor de host do Docker do AtoM, pare o AtoM (caso esteja rodando):

```
docker-compose down
```

Altere o arquivo `docker-compose.dev.yml` do AtoM:

```
nano docker-compose.dev.yml
```

Adicione dois volumes chamados `transfer` e `rsync_ssh_host_keys`:

```
volumes:
...
transfer:
rsync_ssh_host_keys:
```

Conceda acesso ao volume `transfer` aos serviços `atom` e `atom_worker`:

```
services:
atom:
...
volumes:
...
- transfer:/transfer

atom_worker:
...
volumes:
...
- transfer:/transfer
```

Adicione um novo serviço denominado `rsync_server`, substituindo a chave pública do Archivematica gerada anteriormente (conteúdo do arquivo `id_rsa.pub`):

```
rsync_server:
image: canelhas/rsync:2.3.1

volumes:
transfer:/data
rsync_ssh_host_keys:/ssh_host_keys environment:
SSH_AUTH_KEY_1: "CHAVE PUBLICA DO ARCHIVEMATICA" ports:
"2222:22"

command: server
```

A imagem `canelhas/rsync` é um *fork* da `eeacms/rsync`

(<https://hub.docker.com/r/eeacms/rsync/>), imagem com mais de 1 milhão de downloads feita pela *European Environment Agency*. No fork foi feita apenas uma alteração, para que a chave de host de SSH do container seja persistente, o que é fundamental para que a integração seja resistente à recriação do stack do AtoM

OBS: cole a chave com o prefixo "ssh-rsa", por exemplo `SSH_AUTH_KEY_1: "ssh-rsa AAAA...Lz1"`

O Archivematica precisará se conectar à porta 2222, libere esse acesso no firewall caso necessário

```
sudo firewall-cmd --add-port=8001/tcp --permanent
sudo firewall-cmd --reload
```

Inicie o AtoM:

```
docker-compose up -
```

Archivematica server

De volta ao servidor do Archivematica, exercite a comunicação com o servidor de rsync do AtoM executando os comandos abaixo, substituindo o hostname do AtoM:

```
echo "Teste" > /tmp/teste
sudo -H -u archivematica rsync -e 'ssh -p 2222' /tmp/teste root@HOSTNAME_DO_ATO
```

OBS: o hostname a ser utilizado acima é o público, e não o IP interno de um dos containeres do AtoM

Será apresentada a chave pública do servidor de rsync do AtoM (gerada automaticamente durante a primeira execução do serviço `rsync_server` do AtoM). Responda yes para confiar na chave.

Essa operação serve não só para testar a comunicação mas, principalmente, para adicionar a chave pública do servidor de rsync do AtoM ao arquivo de known hosts do Archivematica localizado em

```
/var/lib/archivematica/.ssh/known_hosts .
```

AtoM host server

De volta ao servidor de host do Docker do AtoM, confira se o arquivo teste está presente na pasta `/transfer` dos containeres `docker_atom_1` e `docker_atom_worker_1`:

```
docker exec docker_atom_1 cat /transfer/teste
docker exec docker_atom_worker_1 cat /transfer/teste
```

Configuração do AtoM

Acesse o dashboard do AtoM em um navegador e siga os passos abaixo:

1. Autentique-se como um usuário administrador
2. Clique no ícone de engrenagens no menu superior, em seguida clique em **Plugins** (ou *Componentes*)
3. Habilite os plugins **arRestApiPlugin** e **qtSwordPlugin**, em seguida clique em **Salvar**
4. Clique novamente no ícone de engrenagens, em seguida clique em **Configurações**
5. Selecione a seção **Global** no menu lateral
6. Na opção **Diretório de depósito SWORD**, digite `/transfer`, em seguida clique em **Salvar**
7. Clique novamente no ícone de engrenagens, em seguida clique em **Usuários**
8. Clique em **Adicionar Novo** para criar um usuário administrador para ser usado pelo Archivematica
 - a. Em **Username**, digite `archivematica`
 - b. Em **E-mail**, digite `archivematica@sua_entidade.com`
 - c. Digite uma senha forte de sua preferência
 - d. Em **Grupos de usuários**, selecione o grupo **administrador**
 - e. Em **REST API chave de acesso**, selecione **(Re)gerar chave API**
 - f. Clique em **Criar**
9. Copie a chave de API do usuário gerada (campo **REST API chave**) e guarde em um bloco de notas

Configuração do Archivematica

Acesse o dashboard do Archivematica em um navegador e siga os passos abaixo:

1. Autentique-se como um usuário administrador
2. Clique em **Administration**
3. No menu lateral, na seção **DIP upload**, clique em **AtoM/Binder** e preencha:
 - a. Em **Upload URL**, digite `http://HOSTNAME_DO_ATOM:63001` (`http://HOSTNAME_DO_ATOM:63001`) (porta do serviço *nginx* do stack do AtoM)
 - b. Em **Login email**, digite o e-mail preenchido anteriormente no AtoM para o usuário do Archivematica
 - c. Em **Login password**, digite a senha preenchida anteriormente no AtoM para o usuário do Archivematica
 - d. Em **AtoM/Binder version**, deixe selecionada a opção **2.x**
 - e. Em **Rsync target**, preencha `root@HOSTNAME_DO_ATOM:/data`
 - f. Em **Rsync command**, preencha `ssh -p 2222`
 - g. Em **Debug mode**, selecione o que preferir
 - h. Em **REST API key**, cole a chave de API do usuário Archivematica
 - i. Clique em **Save**
4. Para testar a comunicação com o AtoM, clique no botão **Levels of Description** (localizado no canto superior direito), em seguida no botão **Fetch from AtoM** (também no canto superior direito)

Caso os níveis de descrição sejam trazidos do AtoM, significa que a configuração está correta.

Cada vez que um nível for adicionado no AtoM, é preciso clicar novamente no botão **Fetch from AtoM** para que o Archivematica tome conhecimento do novo nível.

Teste de integração

Para testar a integração, faça um Transfer com o campo *Access system ID* preenchido com um slug de uma descrição previamente criada no AtoM. Na fase de Ingest, no microsserviço de *Normalização*, selecione a opção de **Normalizar para acesso ou Normalizar para preservação e acesso**. Ao final do processamento, verifique se o arquivo aparece no AtoM.

Teste de “statelessness”

Para conferir que o setup feito é realmente *stateless*, armazenando todo o estado externamente nos volumes – sendo portanto resistente à recriação do stack do AtoM – você pode fazer:

```
docker-compose down
docker-compose up -d
```

Após a recriação do stack do AtoM, a integração continua funcionando sem necessitar de nenhuma tarefa adicional.

META Gestão de Documentos Digitais

RHEL – Atualização do Ghostscript Archivematica

Quando instalamos o MCPClient nas versões atuais do Archivematica, é instalado o Ghostscript na versão 9.25, que esta com um *bug* (https://bugs.ghostscript.com/show_bug.cgi?id=700124), que toma muito tempo para ser executado ao trabalhar com arquivos com muitas páginas, no formato PDF.

Por isso, segue um passo a passo de como fazer a atualização do Ghostscript e como configurar para o Archivematica passar a usar a versão atualizada.

1. Confira a versão do comando atual:

```
sudo -u archivematica
gs -v
```

Provavelmente vai aparecer que a versão atual é a 9.25.

2. Agora, instale o GS sem usar o comando yum:

Menu

```
cd /tmp
curl -L -o ghostscript-9.27.tar.gz https://github.com/ArtifexSoftware/ghostpdl-
tar -zxvf ghostscript-9.27.tar.gz
cd ghostscript-9.27
./configure
make
make install
```

3. Confira o local do novo executável (deve vir a versão mais nova):

```
/usr/local/bin/gs -v
```

4. Mude o comando “gs” para a versão mais nova:

```
sudo mv /bin/gs /bin/gs925
sudo ln -sf /usr/local/bin/gs /bin/gs
```

Note que este último comando é importante, pois coloca o Ghostscript no *path* do Archivematica. Confira se agora o comando `gs`, quando executado com o usuário do Archivematica, vai para o novo:

```
sudo -u archivematica gs -v
```

> O argumento `-u archivematica` no comando acima é fundamental para testar o comando da forma como ele será executado pelo Archivematica.

Fontes:

https://bugs.ghostscript.com/show_bug.cgi?id=700124

<https://www.ghostscript.com/doc/current/History9.htm#Version9.26>

<http://git.ghostscript.com/?p=ghostpdl.git;a=commitdi;h=85e6fd4df9bb51d5c36e7811a2bac-4c52898b08c>

<https://askubuntu.com/questions/199489/whats-the-easiest-way-to-upgrade-ghostscript>

ANEXO B – CÓDIGO PILOTO DE ENTRADA HIPÁTIA

Piloto.py

```
import codecs
import xml.etree.ElementTree as ET
import os
import csv
import json
import requests
import base64
import time
import urllib.request
import schedule
import datetime
import hashlib
import shutil
import ConsultaBanco
import numpy as np
from datetime import date

# Variaveis globais
colorErro = '\033[91m'
colorPass = '\033[0m'
pastaRaiz = os.path.abspath(__file__) # Recupera o path do script python
pastaRaiz = (os.path.split(pastaRaiz))[0] # Recupera o path do projeto
IDFuturo=[]
# from memory_profiler import profile
# fp2 = open('memory_log.log', 'w+')
# @profile(stream=fp2)
def infoArchivematica(root):
    infoServer = []

    # Recupera informações do servidor do archivematica
    for servidor in root.iter('fileserver'):
        print('Recuperado informações do Archivematica')
        for mapping_tags in servidor:
            infoServer.append(mapping_tags.text)

    return infoServer

def geraMetadatas(root):
    dados = []
    nomeArquivo = []
    nomePasta = []
    PDF = []
    envioSequencial = 2
    _templ = 0
    _temp = []
    temp={}
    templ={}

    # Pega o root do XML e apresenta todas as suas tags
    for principal in root.iter('mapping'):
        print('Gerando metadatas para criação do CSV...')
        # Apresenta todas as tags dentro da MAPPING
        for mapping_tags in principal:
            # Procura a tag METADATAS que possua tabelas associadas

            if mapping_tags.tag == 'metadatas' and mapping_tags.get('table') is not None:
                assinatura = ConsultaBanco.tabela

                try:
                    nomeTabela = assinatura[mapping_tags.get('table')]
                except KeyError:
```

```

        print('Erro ao extrair os dados. Não possui a tabela' + mapping_tags.get('table'))
        return

    if not nomeTabela:
        print('Sem dados novos!')
        return

    # Verifica quais diários estão assinados
    for i in range(len(nomeTabela)):

        # Verifica se o arquivo foi publicado dentro de 2 dias
        date_time_obj = nomeTabela[i].DataDePublicacao
        if (date.today() - date_time_obj).days < 2:
            print('Metadatos não gerado. Arquivo publicado dentro de 2 dias. Data de publicacao: ' +
                str(
                    nomeTabela[i].DataDePublicacao))
            IDFuturo.append(nomeTabela[i].ID)
            continue

        # Coleta o número de dados descritos no envioSequencial
        if _templ == envioSequencial:
            break

        _templ = 1 + _templ
        metadatos = [] # Inicializa e zera a variável metadatos (segunda linha do CSV)
        identificadorMetadatos = [] # Inicializa e zera a variável identificadorMetadatos (primeira linha
do CSV)

        # Recupera todas as linhas do banco que estão com o status 'assinado'
        for metadatos_tags in mapping_tags:
            # Pega o texto da tag e divide ele em partes onde possui espaço, caso possua o espaço, o
vetor possuirá tamanho maior que 1
            variaveis = (metadatos_tags.text).split()

            # Caso possua duas variáveis no texto, o IF separa cada uma, caso contrário realiza o ELSE
            if len(variaveis) > 1:
                temporaria = '' # Variável temporária
                temporario = []

                for palavra in variaveis:
                    if palavra[0] != '"':
                        # print(consultaPalavra.replace('==CONSULTA==', palavra))

                        temporario += str(ConsultaBanco.valor(nomeTabela[i], palavra))

                    else:
                        temporario += palavra.replace('"', '')

                if metadatos_tags.tag == 'pdf':
                    temporario.append(temporario)
                    temporario = ''

            if metadatos_tags.tag == 'nomeArquivo':
                nomeArquivo.append(temporario)

            elif metadatos_tags.tag == 'nomePasta':
                nomePasta.append(temporario)

            elif metadatos_tags.tag == 'pdf':
                PDF.append(temporario)

            else:
                metadatos.append(temporario)
                identificadorMetadatos.append(metadatos_tags.tag)

        else:
            if metadatos_tags.tag == 'nomeArquivo':
                nomeArquivo.append(metadatos_tags.text)

            elif metadatos_tags.tag == 'nomePasta':
                nomePasta.append(metadatos_tags.text)

            else:
                metadatos.append(ConsultaBanco.valor(nomeTabela[i], metadatos_tags.text))
                identificadorMetadatos.append(metadatos_tags.tag)

```

```

# Une o identificadorMetadatas (primeira linha do CSV) com os metadatos (segunda linha do CSV)

# armazena em uma nova variavel chamada dados
dados.append([identificadorMetadatas] + [metadatos])

# Procura a tag METADATAS que não possua tabelas associadas
elif mapping_tags.tag == 'metadatos' and mapping_tags.get('table') is None:
    temporaria = []
    temporario = []
    for metadatos_tags in mapping_tags:
        temporaria.append(metadatos_tags.tag)
        temporario.append(metadatos_tags.text)

        for i in range(len(dados)):
            dados[i][0] = (dados[i][0] + temporaria)
            dados[i][1] = (dados[i][1] + temporario)

    elif mapping_tags.tag == 'submissionDocumentation':
        for dados1 in range(len(dados)):
            for submission_tags in mapping_tags:
                join = ''
                palavras = (submission_tags.text).split('$')
                for index, item in enumerate(palavras):
                    if item[0] == '@':
                        for index1, item in enumerate(dados[dados1][0]):
                            if (item == palavras[index].replace('@', '')):
                                palavras[index] = dados[dados1][1][index1]

                consulta = ConsultaBanco.consultas2(join.join(palavras))
                for row in consulta[0]:
                    templ.update({submission_tags.tag: dict(zip(consulta[1], row))})

            temp.update({dados1: templ})
            templ={}

variaveis={}
variaveis.update({'submissionDocumentation': temp})
templ = {}
temp = {}
for i in range(len(dados)):
    temp.update({0: dados[i][0]})
    temp.update({1: dados[i][1]})
    templ.update({i: temp})
    temp = {}

variaveis.update({'metadados': templ})

return variaveis, nomeArquivo, PDF

def conx(root, rootJ, TipoDocumento, tabelaCompetencia, processo, classeProcessual, nomeFinalMovimento):
    #XML do PJE
    XML = (ET.parse('pje.xml')).getroot()
    URL = "{http://www.cnj.jus.br/intercomunicacao-2.2.3}"

    dicionario = {
        "filename": "filename",
        "numero": "numeroProcesso",
        "codigoOrgao": "codigoOrgaoJulgador",
        "nomeOrgao": "nomeOrgaoJulgador",
        "classeProcessual": "codigoClasseProcessual",
        "nomeClasseProcessual": "nomeClasseProcessual",
        "codigoLocalidade": "codigoLocalidade",
        "nomeLocalidade": "nomeLocalidade",
        "nivelSigilo": "codigoNivelSigilo",
        "nomeNivelSigilo": "nomeNivelSigilo",
        "dataAjuizamento": "dataAjuizamento",
        "nome": "outroParametro",
        ..
    }

```

```

        "competencia": "codigoCompetencia",
        "nomeCompetencia": "nomeCompetencia",
        "numeroProcesso": "processoVinculado",
        "vinculo": "modalidadeVinculacaoProcesso",
        "prioridade": "prioridade",
        "valorCausa": "valorCausa",
        "codigoNacional": "codigoAssuntoProcessual",
        "nomeAssuntoProcessual": "nomeAssuntoProcessual",
        "temporalidadeAssuntoProcessual": "temporalidadeAssuntoProcessual",
        "mimetype": "mimetype",
        "meioDocumental": "meioDocumental"
    }
    dicionariofinal = {
        "filename": "",
        "numero": "",
        "codigoOrgao": "",
        "nomeOrgao": "",
        "classeProcessual": "",
        "nomeClasseProcessual": "",
        "codigoLocalidade": "",
        "nomeLocalidade": "",
        "nivelSigilo": "",
        "nomeNivelSigilo": "",
        "dataAjuizamento": "",
        "nome": "",
        "competencia": "",
        "nomeCompetencia": "",
        "numeroProcesso": "",
        "vinculo": "",
        "prioridade": "",
        "valorCausa": "",
        "principal": "",
        "codigoNacional": "",
        "nomeAssuntoProcessual": "",
        "temporalidadeAssuntoProcessual": "",
        "mimetype": "application/pdf",
        "meioDocumental": "Digital",
        "issad.appraisal": ""
    }

    grauSigilo = {
        "0": "públicos, acessíveis a todos os servidores do Judiciário e dos demais órgãos públicos de colaboração na administração da Justiça, assim como aos advogados e a qualquer cidadão",
        "1": "segredo de justiça, acessíveis aos servidores do Judiciário, aos servidores dos órgãos públicos de colaboração na administração da Justiça e às partes do processo.",
        "2": "sigilo mínimo, acessível aos servidores do Judiciário e aos demais órgãos públicos de colaboração na administração da Justiça",
        "3": "sigilo médio, acessível aos servidores do órgão em que tramita o processo, à(s) parte(s) que provocou(ram) o incidente e àqueles que forem expressamente incluídos",
        "4": "sigilo intenso, acessível a classes de servidores qualificados (magistrado, diretor de secretaria/escrivão, oficial de gabinete/assessor) do órgão em que tramita o processo, às partes que provocaram o incidente e àqueles que forem expressamente incluídos",
        "5": "sigilo absoluto, acessível apenas ao magistrado do órgão em que tramita, aos servidores e demais usuários por ele indicado e às partes que provocaram o incidente."
    }
    dicionarioDoc = {
        "idDocumento": "idDocumento",
        "idDocumentoVinculado": "idDocumentoVinculado",
        "tipoDocumento": "codigoTipoDocumento",
        "nomeTipoDocumento": "nomeTipoDocumento",
        "dataHora": "dataHora",
        "descricao": "descricao",
        "hash": "hash",
        "nivelSigilo": "codigoNivelSigilo",
        "nomeNivelSigilo": "nomeNivelSigilo",
        "movimento": "movimento",
        "conteudo": "conteudo",
        "mimetype": "mimetype",
        "assinatura": "assinatura",
        "algoritmoHashAssinatura": "algoritmoHashAssinatura",
        "cadeiaCertificadoAssinatura": "cadeiaCertificadoAssinatura",
        "codificacaoCertificadoAssinatura": "codificacaoCertificadoAssinatura",
        "dataAssinatura": "dataAssinatura",
        "signatarioLogin": "signatarioLogin"
    }

```

```

}
dicionarioFinalDoc = {
  "idDocumento": "",
  "idDocumentoVinculado": "",
  "tipoDocumento": "",
  "nomeTipoDocumento": "",
  "dataHora": "",
  "descricao": "",
  "hash": "",
  "nivelSigilo": "",
  "nomeNivelSigilo": "",
  "movimento": "",
  "conteudo": "",
  "mimetype": "",
  "assinatura": "",
  "algoritmoHashAssinatura": "",
  "cadeiaCertificadoAssinatura": "",
  "codificacaoCertificadoAssinatura": "",
  "dataAssinatura": "",
  "signatarioLogin": ""
}
dicionarioPartes = {
  "polo": "polo",
  "nome": "nomeParte",
  "nomeGenitor": "nomeGenitor",
  "nomeGenitora": "nomeGenitora",
  "numeroDocumentoPrincipal": "numeroDocumentoPrincipal",
  "tipoPessoa": "tipoPessoa",
  "nacionalidade": "nacionalidade",
  "codigoDocumento": "codigoDocumento",
  "emissorDocumento": "emissorDocumento",
  "tipoDocumento": "tipoDocumento",
  "nomeNoDoc": "nomeDocumento",
  "outroNome": "outroNome",
  "assistenciaJudiciaria": "assistenciaJudiciaria",
  "intimacao": "representanteIntimacao",
  "nomeAdv": "representanteNome",
  "inscricao": "representanteInscricao",
  "numeroDocumentoPrincipal": "representanteNroDocPrincipal",
  "tipoRepresentante": "tipoRepresentante"
}
dicionarioFinalPartes = {
  "polo": "",
  "nome": "",
  "nomeGenitor": "",
  "nomeGenitora": "",
  "numeroDocumentoPrincipal": "",
  "tipoPessoa": "",
  "nacionalidade": ""
}
dicionarioFinalPartes2 = {
  "codigoDocumento": "",
  "emissorDocumento": "",
  "tipoDocumento": "",
  "nomeNoDoc": "",
  "outroNome": "",
  "assistenciaJudiciaria": "",
  "intimacao": "",
  "nomeAdv": "",
  "inscricao": "",
  "numeroDocumentoPrincipal": "",
  "tipoRepresentante": ""
}
dicionarioMov = {
  "dataHora": "dataHora",
  "nivelSigilo": "codigoNivelSigilo",
  "nomeNivelSigilo": "nomeNivelSigilo",

```



```

        "identificadorMovimento": "identificadorMovimento",
        "codigoNacional": "movimentoCodigoNacional",
        "movimentoNomeNacional": "movimentoNomeNacional",
        "complemento": "complemento",
        "idDocumentoVinculado": "idDocumentoVinculado",
        "movimento": "movimento"
    }
    dicionarioFinalMov = {
        "dataHora": "",
        "nivelSigilo": "",
        "nomeNivelSigilo": "",
        "identificadorMovimento": "",
        "codigoNacional": "",
        "movimentoNomeNacional": "",
        "complemento": "",
        "idDocumentoVinculado": "",
        "movimento": ""
    }
    dicioPolo = {
        "AT": "Ativo",
        "PA": "Passivo",
        "TC": "Terceiro",
        "FL": "Fiscal da lei diverso",
        "TJ": "Testemunha do juízo",
        "AD": "Assistente simples desinteressado (amicus curiae)",
        "VI": "Vitima"
    }
    dicioTipoDocumento = {
        "CI": "carteira de identidade",
        "CNH": "carteira nacional de habilitação",
        "TE": "título de eleitor",
        "CN": "certidão de nascimento",
        "CC": "certidão de casamento",
        "PAS": "passaporte",
        "CT": "carteira de trabalho",
        "RIC": "registro individual do cidadão",
        "CMF": "cadastro no Ministério da Fazenda Brasileiro",
        "PIS_PASEP": "número no programa de integração social",
        "CEI": "cadastro específico do INSS",
        "NIT": "número de identificação do trabalho",
        "CP": "cadastro em conselhos profissionais",
        "IF": "identidade funcional",
        "OAB": "número de cadastro na Ordem dos Advogados do Brasil",
        "RJC": "número de inscrição empresarial",
        "RGE": "registro de identificação do estrangeiro",
    }
    dicioTipoRepresentante = {
        "A": "advogado",
        "E": "Escritório de advocacia",
        "M": "Ministério público",
        "D": "Defensoria pública",
        "P": "outros órgãos de representação processual pública (advocacia pública em geral)"
    }
    primeiroMetadado = {"filename":""}

    resultadoFinal = []
    resultadoFinal2 = []
    listaGuardaPermanenteAssunto = [3613, 3653, 7661, 5804, 7667, 7671, 4656, 4660, 4670, 4680, 10499, 10449,
10450,
                                10451, 10452, 10453, 10454, 10456, 10457, 10458, 10459, 10460, 10500, 10461,
10469,
                                10470, 10480, 10481, 11412, 11413, 6206, 6207, 6213, 6216, 6218, 6219, 9565,
7732,
                                7735, 7834, 7835, 7925, 7926, 9972, 9973, 9974, 9975, 9990, 10012, 10013,
10014,
                                10108, 10109, 10131, 10121, 10122, 10123, 10125, 10126, 10127, 10128, 10129,
10130,
                                10132, 10133, 10134, 10135, 10931, 10932, 10414, 10416]
    listaGuardaPermanenteClasse = [216, 457, 49, 63, 64, 65, 66, 90, 95, 110, 118, 119, 121, 1296, 1297, 1178,
11791,
                                1396, 1399, 1690, 1705, 88, 1303, 1304, 1307, ]

```

```

class Item():
    def __init__(self, idDocumento, regiao):
        self.nomecoluna = idDocumento
        self.conteudo = regiao

class Inserir():
    def __init__(self, nomenclatura, conteudo):
        self.texto = nomenclatura
        self.valor = conteudo

class Mapear():
    def __init__(self):
        self.listaColunas = []
        print("Mapeando...")

    def temSub(self, obj=[]):
        if type(obj) == list:
            for x in obj:
                self.temSub(x)
        else:
            self.listaColunas.append(obj.tag)

class Especifico():
    pass

    def __init__(self, resultadoPartes, resultadoPartesNome, countgrav, gravametadada, gravametadada2, xmlnome,
countPartes, gravacaoDaPartel, gravacaoDaParte2, menorDataHora, counthora,
docMaisRecente, countmetadada):
        print("Extraindo...")
        self.resultadoPartes = resultadoPartes
        self.resultadoPartesNome = resultadoPartesNome
        self.countgrav = countgrav
        self.gravametadada = gravametadada
        self.gravametadada2 = gravametadada2
        self.xmlnome = xmlnome
        self.countPartes = countPartes
        self.gravacaoDaPartel = gravacaoDaPartel
        self.gravacaoDaParte2 = gravacaoDaParte2
        self.menorDataHora = menorDataHora
        self.counthora = counthora
        self.docMaisRecente = docMaisRecente
        self.countmetadada = countmetadada
        self.permanentecount = permanentecount

#complementa os campos do isad com o valor do dublicore
def xml(self):
    for x in XML:
        for y in x:
            for i in y:
                if i.text is None:
                    #dicionariofinal[i.tag] = ""
                    primeiroMetadado[i.tag] = ""
                else:
                    #dicionariofinal[i.tag] = i.text
                    primeiroMetadado[i.tag] = i.text
                    self.xmlnome.append(i.tag)
            # dicionariofinal["dc.title"] = ('Processo n°' + dicionariofinal["numero"])
            # dicionariofinal["dc.date"] = dicionariofinal["dataAjuizamento"][:8]
            # dicionariofinal["dc.identifier"] = dicionariofinal["numero"]
            # dicionariofinal["dc.rights"] = dicionariofinal["nomeNivelSigilo"]
            # dicionariofinal["isad.identifier"] = dicionariofinal["dc.identifier"]
            # dicionariofinal["isad.title"] = dicionariofinal["dc.title"]
            # dicionariofinal["isad.eventDates"] = dicionariofinal["dc.date"]
            # dicionariofinal["isad.accessConditions"] = dicionariofinal["dc.rights"]

            primeiroMetadado["dc.title"] = ('Processo n°' + dicionariofinal["numero"])
            primeiroMetadado["dc.date"] = dicionariofinal["dataAjuizamento"][:8]
            primeiroMetadado["dc.identifier"] = dicionariofinal["numero"]
            primeiroMetadado["dc.rights"] = dicionariofinal["nomeNivelSigilo"]
            primeiroMetadado["isad.identifier"] = primeiroMetadado["dc.identifier"]
            primeiroMetadado["isad.title"] = primeiroMetadado["dc.title"]

```

```

primeiroMetadado["isad.eventDates"] = primeiroMetadado["dc.date"]
primeiroMetadado["isad.accessConditions"] = primeiroMetadado["dc.rights"]

# temp = datetime.datetime.strptime(dicionariofinal["dc.date"], '%Y%m%d')
# dicionariofinal["dc.date"] = temp.strftime('%Y-%m-%d')

temp = datetime.datetime.strptime(primeiroMetadado["dc.date"], '%Y%m%d')
primeiroMetadado["dc.date"] = temp.strftime('%Y-%m-%d')

# print(" HENRIQUE:",primeiroMetadado)
# print(dicionariofinal)
#
# for m in primeiroMetadado:
#     print( m, primeiroMetadado[m])

# função dos metadados do Metadata.CSV
def processo(self, parametro):
    if i.tag.replace(URL, "") == parametro:
        if dicionarioFinalMov["dataHora"] == "":
            for v in i.attrib:
                if v in dicionariofinal:
                    dicionariofinal[v] = i.get(v)
                    conversor.sigilo(v)
                    conversor.localidade(v, i.get(v))
                    conversor.competencia(v, i.get(v))
                    if i.tag.replace(URL, "") == "assunto":
                        if i.get(v) == "true":
                            dicionariofinal[v] = "Assunto Principal"
                        elif i.get(v) == "false":
                            dicionariofinal[v] = ""
                    if i.tag.replace(URL, "") == "dadobasico":
                        buscar.nomeProcesso()
            if i.tag.replace(URL, "") == "orgaoJulgador" and v == "codigoMunicipioIBGE":
                buscar.xml()

#
# verifica se o assunto e classe é de guarda permanente
if dicionariofinal["codigoNacional"] != "":
    print("codigoNacional", dicionariofinal["codigoNacional"])
    print("classeprocessual", dicionariofinal["classeProcessual"])
    for x in listaGuardaPermanenteAssunto:
        if x == int(dicionariofinal["codigoNacional"]):
            dicionariofinal["temporalidadeAssuntoProcessual"] = "Permanente"
            dicionariofinal["isad.appraisal"] = "Documento classificado com assunto de
temporalidade permanente."
            self.permanentecount += 1
    for y in listaGuardaPermanenteClasse:
        if y == int(dicionariofinal["classeProcessual"]):
            dicionariofinal["isad.appraisal"] = "Documento classificado com a classe de
temporalidade permanente."
            self.permanentecount += 1
    if self.permanentecount == 2:
        dicionariofinal["isad.appraisal"] = "Documento classificado com assunto de
temporalidade permanente. Documento classificado com a classe de temporalidade permanente."
        # RETORNAR QUANDO NORMALIZAR
        #print("dicionariofinal[isad.appraisal]=====",dicionariofinal["isad.appraisal"])

# Extrai os valor que estão em formato text para a Movimentação
def textoMov(self, parametro):
    if i.tag.replace(URL, "") == parametro:
        dicionarioFinalMov[parametro] = i.text

# Extrai os valor que estão em formato text para Metadata.csv
def texto(self, parametro):
    if i.tag.replace(URL, "") == parametro:
        dicionariofinal[parametro] = i.text
    if i.tag.replace(URL, "") == "codigoNacional":
        if self.countgrav == 1:
            buscar.gravarPartes()
        self.countgrav = 2

```

```

def documentos(self, parametro):
    if i.tag.replace(URL, "") == parametro:
        if dicionariofinal["codigoNacional"] != "":
            for v in i.attrib:

                dicionarioFinalDoc[v] = i.get(v)
                conversor.sigiloDoc(v)

            # RETORNAR QUANDO NORMALIZAR O WEBSERVICE
            # if v == "tipoDocumento":
            #     if i.get(v) in dicionarioTipoDocumentoProcessual:
            #         dicionarioFinalDoc["nomeTipoDocumento"] =
dicionarioTipoDocumentoProcessual[i.get(v)]

            # captura o documento mais antigo
            if v == "dataHora":
                if self.counthora == 0:
                    self.menorDataHora = int(i.get("dataHora"))
                    self.counthora = 1
                    self.docMaisRecente = i.get("idDocumento")
                    self.docMaisRecenteMimetype = i.get("mimetype").split('/')[1]
                    # dicionariofinal['filename'] = str(
                    #     "objects/" + self.docMaisRecente + "." + self.docMaisRecenteMimetype)
                    primeiroMetadado['filename'] = str(
                        "objects/" + self.docMaisRecente + "." + self.docMaisRecenteMimetype)
                elif self.counthora == 1:
                    if int(i.get("dataHora")) < self.menorDataHora:
                        self.menorDataHora = int(i.get("dataHora"))
                        self.docMaisRecente = i.get("idDocumento")
                        self.docMaisRecenteMimetype = i.get("mimetype").split('/')[1]
                        # dicionariofinal['filename'] = str(
                        #     "objects/" + self.docMaisRecente + "." + self.docMaisRecenteMimetype)

                        primeiroMetadado['filename'] = str(
                            "objects/" + self.docMaisRecente + "." + self.docMaisRecenteMimetype)

            if parametro == "assinatura" and dicionarioFinalDoc["idDocumento"] != "":
                gravacao.gravarCsv("Documentos")
                for x in dicionarioFinalDoc:
                    dicionarioFinalDoc[x] = ""

def movimento(self, parametro):
    if i.tag.replace(URL, "") == parametro:
        for v in i.attrib:
            if parametro == "movimento" and v == "dataHora":
                gravacao.gravarMov()
                dicionarioFinalMov[v] = i.get(v)
                conversor.sigiloMov(v)

def gravMetadatas(self):
    #busca o nome da Classe Processual
    for x in classeProcessual.iter():
        dicionariofinal["nomeClasseProcessual"] = x.text

    if self.countmetadata == 0:
        for x in dicionariofinal:
            self.gravametadata.append(dicionariofinal[x])
        for x in dicionario:
            self.gravametadata2.append(x)
        for x in self.xmlnome:
            self.gravametadata2.append(x)
        nomenclaturaMetadata = []
        for nom in self.gravametadata2:
            if nom in dicionario:
                nomenclaturaMetadata.append(dicionario[nom])
            else:
                nomenclaturaMetadata.append(nom)
        self.gravametadata2 = nomenclaturaMetadata
        self.countmetadata = 2

```

```

def gravarPartes(self):
    if self.countPartes == 1:
        for x in dicioTipoDocumento:
            if x in self.resultadoPartes:
                posicaoTipoDocumento = int(self.resultadoPartes.index(x))
                valorDaPosicaoTipoDocumento = self.resultadoPartes[posicaoTipoDocumento]
                self.resultadoPartes[posicaoTipoDocumento] =
dicioTipoDocumento[valorDaPosicaoTipoDocumento]

        if self.resultadoPartes[0] in dicioPolo: # transforma a sigla, na palavra completa
            self.resultadoPartes[0] = dicioPolo[self.resultadoPartes[0]]
        self.gravacaoDaPartel.append(self.resultadoPartesNome)
        self.gravacaoDaParte2.append(self.resultadoPartes)
        self.resultadoPartesNome = []
        self.resultadoPartes = []
    else:
        self.countPartes += 1

def partes(self, parametro):
    if i.tag.replace(URL, "") == parametro:
        if dicionarioFinal["codigoNacional"] == "": # isso q faz buscar só documento de partes
            if i.tag.replace(URL, "") == "polo":
                buscar.gravarPartes()
                for v in i.attrib:
                    if v == "polo":
                        if i.get(v) in dicioPolo:
                            dicionarioFinalPartes["polo"] = i.get(v)
                            self.resultadoPartesNome.append("polo")
                        else:
                            if v in dicionarioFinalPartes:
                                dicionarioFinalPartes[v] = i.get(v)
                                self.resultadoPartesNome.append(v)
                            self.resultadoPartes.append(i.get(v))
            if i.tag.replace(URL, "") == "pessoa":
                for v in i.attrib:
                    if v == "nome":
                        if v in dicionarioFinalPartes:
                            dicionarioFinalPartes["nome"] = i.get(v)
                            self.resultadoPartesNome.append("nomeParte")
                            self.resultadoPartes.append(i.get(v))
                        else:
                            if v in dicionarioFinalPartes:
                                dicionarioFinalPartes[v] = i.get(v)
                                self.resultadoPartesNome.append(v)
                                self.resultadoPartes.append(i.get(v))
            if i.tag.replace(URL, "") == "outroNome":
                dicionarioFinalPartes2["outroNome"] = i.text
                self.resultadoPartesNome.append("outroNome")
                self.resultadoPartes.append(i.text)
            if i.tag.replace(URL, "") == "documento":
                for v in i.attrib:
                    if v == "nome":
                        dicionarioFinalPartes2["nomeNoDoc"] = i.get(v)
                        self.resultadoPartesNome.append("nomeDocumento")
                    else:
                        dicionarioFinalPartes2[v] = i.get(v)
                        self.resultadoPartesNome.append(v)
                        self.resultadoPartes.append(i.get(v))

            if i.tag.replace(URL, "") == "parte":
                if i.get("assistenciaJudiciaria") == 'false':
                    dicionarioFinalPartes2["assistenciaJudiciaria"] = i.get("assistenciaJudiciaria")
                    self.resultadoPartes.append("")
                if i.get("assistenciaJudiciaria") == 'true':
                    dicionarioFinalPartes2["assistenciaJudiciaria"] = i.get("assistenciaJudiciaria")
                    self.resultadoPartes.append(
                        "O processo terá assistência judiciária às pessoas consideradas juridicamente
pobres")

```



```

        self.resultadoPartesNome.append("assistenciaJudiciaria")

    if i.tag.replace(URL, "") == "advogado":
        for v in i.attrib:
            if v == "nome":
                dicionarioFinalPartes2["nomeAdv"] = i.get(v)
                self.resultadoPartesNome.append("representanteNome")
                self.resultadoPartes.append(i.get(v))
            elif v == "intimacao":
                if i.get(v) == "false":
                    self.resultadoPartes.append(
                        "O advogado, escritório ou órgão de representação não são " \
                        "preferenciais para a realização " \
                        "de intimações. ")
                elif i.get(v) == "true":
                    self.resultadoPartes.append("O advogado, escritório ou órgão " \
                                                "de representação são " \
                                                "preferenciais para a realização " \
                                                "de intimações. ")
                    self.resultadoPartesNome.append("representanteIntimacao")
            elif v == "inscricao":
                if v in dicionarioFinalPartes2:
                    dicionarioFinalPartes2["inscricao"] = i.get(v)
                    self.resultadoPartesNome.append("representanteInscricao")
                    self.resultadoPartes.append(i.get(v))
            elif v == "numeroDocumentoPrincipal":
                if v in dicionarioFinalPartes2:
                    dicionarioFinalPartes2["numeroDocumentoPrincipal"] = i.get(v)
                    self.resultadoPartesNome.append("representanteNroDocPrincipal")
                    self.resultadoPartes.append(i.get(v))
            elif v == "tipoRepresentante":
                if i.get(v) in dicioTipoRepresentante:
                    dicionarioFinalPartes2[v] = dicioTipoRepresentante[i.get(v)]
                    self.resultadoPartesNome.append("tipoRepresentante")
                    self.resultadoPartes.append(dicionarioFinalPartes2[v])
            else:
                dicionarioFinalPartes2[v] = i.get(v)
                self.resultadoPartesNome.append(v)
                self.resultadoPartes.append(i.get(v))

def metadados(parametro):
    if i.tag.replace(URL, "") == parametro:
        for v in i.attrib:
            resultadofinal.append(i.get(v))
            if v in dicionario:
                resultadofinal2.append(dicionario[v])
            else:
                resultadofinal2.append(v)

# def movimentacao(parametro):
#     if i.tag.replace(URL, "") == parametro:
#         for v in i.attrib:
#             if v in dicionario:
#                 resultadcmovimentacao2.append(dicionario[v])
#             else:
#                 resultadcmovimentacao2.append(v)

class Conversor():
    def __init__(self, cnt, nomeLocalidade, dicioTipoDocumento, countcriacao, dicioCompetencia,
countcompetencia):
        self.cnt = cnt
        self.nomeLocalidade = nomeLocalidade
        self.dicioTipoDocumento = dicioTipoDocumento
        self.countcriacao = countcriacao
        self.dicioCompetencia = dicioCompetencia

```

```

        self.countcompetencia = countcompetencia
        print("Convertendo nomenclatura...")

def tipoDocumento(self, v, cod):
    if v == "tipoDocumento":
        for x in TipoDocumento:
            for y in x:
                for Return in y:
                    for i in Return:
                        if self.countcriacao == 0:
                            codigo = i.text
                            self.countcriacao += 1
                        elif self.countcriacao == 1:
                            descricao = i.text
                            self.dicioTipoDocumento[codigo] = descricao
                            self.countcriacao = 0
    if cod in self.dicioTipoDocumento:
        dicionarioFinalDoc["nomeTipoDocumento"] = self.dicioTipoDocumento[cod]

def competencia(self, v, cod):
    if v == "competencia":
        for x in tabelaCompetencia:
            for y in x:
                for Return in y:
                    for i in Return:
                        if self.countcompetencia == 0:
                            descricao = i.text
                            self.countcompetencia += 1
                        elif self.countcompetencia == 1:
                            idCompetencia = i.text
                            self.dicioCompetencia[idCompetencia] = descricao
                            self.countcompetencia = 0
    if cod in self.dicioCompetencia:
        dicionariofinal["nomeCompetencia"] = self.dicioCompetencia[cod]

def sigiloMov(self, v):
    if v == "nivelSigilo":
        dicionarioFinalMov[dicionarioMov["nomeNivelSigilo"]] = grauSigilo[i.get(v)]

def sigilo(self, v):
    if v == "nivelSigilo":
        dicionariofinal[dicionario["nomeNivelSigilo"]] = grauSigilo[i.get(v)]

def sigiloDoc(self, v):
    if v == "nivelSigilo":
        dicionarioFinalDoc[dicionarioDoc["nomeNivelSigilo"]] = grauSigilo[i.get(v)]

def localidade(self, v, i):
    if v == "codigoLocalidade":
        for z in rootJ.iter():
            lista.temSub(z)
            if z.tag == "id":
                idi = z.text
                self.cnt = self.cnt + 1
            elif z.tag == "descricao":
                descricao = z.text
                self.cnt = self.cnt + 1
            if self.cnt == 2:
                self.nomeLocalidade[idi] = descricao
                self.cnt = 0
        dicionariofinal[dicionario["nomeLocalidade"]] = self.nomeLocalidade[i]

def nomenclatura(self, parametro):
    if parametro in dicionarioPartes:
        convertido = dicionarioPartes[parametro]
        self.resultadoPartesNome.append(convertido)
    else:
        self.resultadoPartesNome.append(parametro)

```



```

class Gravacao():
    def __init__(self, linha1, linha2, linhaP1, linhaP2, cd, cp, countMov, linhaM1, linhaM2,
        gravacaoDaMovimentacao1, gravacaoDaMovimentacao2, gravacaoDeDocumento1, gravacaoDeDocumento2,
countMovNome):
        print("Gravando...")
        self.linha1 = linha1
        self.linha2 = linha2
        self.linhaP1 = linhaP1
        self.linhaP2 = linhaP2
        self.cd = cd
        self.cp = cp
        self.countMov = countMov
        self.linhaM1 = linhaM1
        self.linhaM2 = linhaM2
        self.gravacaoDaMovimentacao1 = gravacaoDaMovimentacao1
        self.gravacaoDaMovimentacao2 = gravacaoDaMovimentacao2
        self.gravacaoDeDocumento1 = gravacaoDeDocumento1
        self.gravacaoDeDocumento2 = gravacaoDeDocumento2
        self.countMovNome = countMovNome

    def gravarCsv(self, nomeArquivo):
        if self.cd == 0:
            for p in dicionarioDoc:
                self.linha1.append(dicionarioDoc[p])
                self.gravacaoDeDocumento1.append(self.linha1)
                self.cd = 1
        if self.cd == 1:
            for p in dicionarioFinalDoc:
                self.linha2.append(dicionarioFinalDoc[p])
                self.gravacaoDeDocumento2.append(self.linha2)
                self.linha2 = []

    def gravarMov(self):
        if self.countMov == 0:
            for p in dicionarioMov:
                self.linhaM1.append(dicionarioMov[p])
                self.gravacaoDaMovimentacao1.append(self.linhaM1)
                self.countMov = 1
        elif self.countMov == 1:
            dicionarioFinalMov["movimentoNomeNacional"] = nomeFinalMovimento[self.countMovNome]
            self.countMovNome += 1
            for p in dicionarioFinalMov:
                self.linhaM2.append(dicionarioFinalMov[p])
                self.gravacaoDaMovimentacao2.append(self.linhaM2)
                self.linhaM2 = []
            for p in dicionarioFinalMov:
                dicionarioFinalMov[p] = ""

# inicializa o init
lista = Mapear()
buscar = Especifico([], [], 1, [], [], [], 0, [], [], 0, 0, "", 0, 0)
gravacao = Gravacao([], [], [], [], 0, 0, 0, [], [], [], [], [], [], 0)
conversor = Conversor(0, {"": ""}, {"": ""}, 0, {"": ""}, 0)

# Executa a extração e salvamento nos arquivos csv
for i in root.iter():
    lista.temSub(i)
    #print(i.tag.replace(URL, ""), i.attrib)
    #print("")
    buscar.processo("dadosBasicos")
    buscar.processo("assunto")
    buscar.processo("processoVinculado")
    buscar.processo("outroParametro")
    buscar.processo("movimentoNacional")
    buscar.processo("orgaoJulgador")
    buscar.documentos("documento")
    buscar.documentos("documentoVinculado")
    buscar.documentos("assinatura")
    buscar.partes("polo")

```

```

        buscar.partes("parte")
        buscar.partes("pessoa")
        buscar.partes("outroNome")
        buscar.partes("documento")
        buscar.partes("advogado")
        buscar.texto("prioridade")
        buscar.texto("valorCausa")
        buscar.texto("codigoNacional")
        buscar.movimento("movimento")
        buscar.movimento("movimentoNacional")
        buscar.textoMov("complemento")
        buscar.textoMov("idDocumentoVinculado")

gravacao.gravarMov()

buscar.gravMetadatas()

dicionarioFinalMov = dict(
    (dicionarioMov[k], dicionarioFinalMov[k]) for k, _ in dicionarioFinalMov.items() if k in dicionarioMov)
for each in dicionarioFinal:
    primeiroMetadado[each] = dicionarioFinal[each]

variaveis = {}
templ = {}
variaveis.update({'objects': processo.documento})

temp = gravacao.gravacaoDeDocumento1
for i in range(len(gravacao.gravacaoDeDocumento2)):
    temp.append(gravacao.gravacaoDeDocumento2[i])
templ.update({'documentos': temp})

temp = {}
temp.update({0: buscar.gravametadata2})
temp.update({1: buscar.gravametadata})
variaveis.update({'metadados': temp})

temp = gravacao.gravacaoDaMovimentacao1
for i in range(len(gravacao.gravacaoDaMovimentacao2)):
    temp.append(gravacao.gravacaoDaMovimentacao2[i])
templ.update({'movimentos': temp})

temp = [buscar.gravacaoDaParte1[0]]
for i in range(len(buscar.gravacaoDaParte2)):
    temp.append(buscar.gravacaoDaParte2[i])
templ.update({'partes': temp})

variaveis.update({'submissionDocumentation': templ})

return variaveis

def envioArquivematica(path, infoServer, raiz, nomeArquivo, identificador):
    try:
        with open(path + '/metadata/checksum.md5', 'w+') as hashmd:
            for root, dirs, files in os.walk(path + "/objects"):
                for file in files:
                    if file != 'checksum.md5':
                        filename = root + '/' + file
                        hasher = hashlib.md5()
                        with open(filename, 'rb') as openfile:
                            content = openfile.read()
                            hasher.update(content)
                            hashDoArquivo = hasher.hexdigest()

                        hashmd.write(hashDoArquivo + ' ' + file + '\n')

                    openfile.close()

        hashmd.close()

```

```

except Exception:
    print(colorErro + "Erro ao gerar o checksum." + colorPass)

try:
    print('Enviando para Archivematica')
    url = "http://" + infoServer[0] + ":" + infoServer[1] + "/api/v2beta/package/?username=" + infoServer[2] +
"&api_key=" + infoServer[3]
    diretorio = infoServer[4] + ':' + raiz + nomeArquivo.replace("'", "")
    parametros = '{"name":"' + nomeArquivo.replace("'",
        "")) +
        '"', "access_system_id": "tjdft", "auto_approve": true, "type": "' + \
        infoServer[5] + '", "path": "' + str(base64.b64encode(diretorio.encode("utf-8"))), "utf-8") +
        "'}'
    resposta = (requests.post(url, data=parametros)).json()

except Exception as e:
    print(colorErro + "Erro ao iniciar a transferencia para o Archivematica." + colorPass)
    return

if 'resposta' in locals() and resposta['id']:
    time.sleep(3)

    print('\n#####Realizando Transfer!
#####')
    while True:
        urlstatus = "http://" + infoServer[0] + ":" + infoServer[1] + "/api/transfer/status/" + resposta[
            'id'] + "?username=" + infoServer[2] + "&api_key=" + infoServer[3]
        res = (requests.get(urlstatus)).json()
        try:
            print("Transfer Status:" + res['status'] + ' as ' + str(datetime.datetime.now()))

            if (res['status'] == 'COMPLETE'):
                time.sleep(5)
                uuidinj = res['sip_uuid']
                hidetransfer = "http://" + infoServer[0] + ":" + infoServer[1] + "/api/transfer/" + resposta[
                    'id'] + "/delete/?username=" + infoServer[2] + "&api_key=" + infoServer[3]
                (requests.delete(hidetransfer)).json()
                break

            elif (res['status'] == 'FAILED' or res['status'] == 'USER_INPUT'):
                with open('erros.txt', 'a') as gravarID:
                    gravarID.write("Erro ao efetuar o Transfer: " + str(identificador) + " " + str(
                        datetime.datetime.now()) + " " + res['status'] + "\n")
                    gravarID.close()

                with open('verificadorDeID.txt', 'w+') as gravarID:
                    gravarID.write(str(identificador))
                    gravarID.write('\n' + ' '.join(map(str, IDFuturo)))
                    gravarID.close()
                break

            time.sleep(2)

        except Exception:
            print(colorErro + "Erro ao verificar o status do TRANSFER." + colorPass)

    print('#####Realizando Ingest!
#####')

    while True:
        urlstatusinjest = "http://" + infoServer[0] + ":" + infoServer[
            1] + "/api/ingest/status/" + uuidinj + "?username=" + infoServer[2] + "&api_key=" + infoServer[3]
        inj = (requests.get(urlstatusinjest)).json()
        print("Ingest Status::" + inj['status'] + ' as ' + str(datetime.datetime.now()))

        if (inj['status'] == 'COMPLETE'):
            try:
                hideingest = "http://" + infoServer[0] + ":" + infoServer[
                    1] + "/api/ingest/" + uuidinj + "/delete/?username=" + infoServer[2] + "&api_key=" +

```

```

infoServer[
    3]
    (requests.delete(hideingest)).json()
    with open('verificadorDeID.txt', 'w+') as gravarID:
        print("")
        print('Gravando o ID ' + str(identificador) + "\n")
        gravarID.write(str(identificador))
        gravarID.write('\n' + ' '.join(map(str, IDFuturo)))
        gravarID.close()
    break

except Exception:
    print(colorErro + "Erro ao gravar o ultimo ID....." + colorPass)

elif (inj['status'] == 'FAILED' or inj['status'] == 'USER_INPUT'):
    with open('erros.txt', 'a+') as gravarID:
        gravarID.write(
            "Erro ao gravar ingest: " + str(identificador) + " " + str(datetime.datetime.now()) + "
" +
            inj['status'] + "\n")
        gravarID.close()
    with open('verificadorDeID.txt', 'w+') as gravarID:
        gravarID.write(str(identificador))
        gravarID.write('\n' + ' '.join(map(str, IDFuturo)))
        gravarID.close()
    break

time.sleep(2)

else:
    print(resposta['message'])
else:
    print('erro')

# Apaga a pasta documentos ao final do processo
try:
    shutil.rmtree(pastaRaiz + '/documentos/')
except OSError as e:
    print(colorErro + "Erro: %s - %s. \n" % (e.filename, e.strerror) + colorPass)

def job(metadata, infoServer):
    if (ConsultaBanco.sistema == 'dje'):
        nomeArquivo = metadata[1]
        PDF = metadata[2]
        _templ = 0
        _temp = []

        # Escreve o CSV
        for i in range(len(metadata[0]['metadados'])):
            identificador = metadata[0]['metadados'][i][1][1]
            raiz = pastaRaiz + '/documentos/'
            path = raiz + nomeArquivo[i].replace('"', "") # Caminho onde o arquivo será criado

            print('Criando estrutura de pastas para envio...')
            if not os.path.isdir(path):
                os.makedirs(path)

            if not os.path.isdir(path + '/metadata'):
                os.makedirs(path + '/metadata')

            if not os.path.isdir(path + '/metadata/submissionDocumentation'):
                os.makedirs(path + '/metadata/submissionDocumentation')

            if not os.path.isdir(path + '/objects'):
                os.makedirs(path + '/objects')

        # Grava o CSV no diretorio designado
        print('Gravando informações no CSV...')

```

```

with open(path + '/metadata/metadata.csv', 'w', newline='') as csvFile:
    writer = csv.writer(csvFile, delimiter=',')
    writer.writerow(metadata[0]['metadados'][i][0])
    writer.writerow(metadata[0]['metadados'][i][1])
csvFile.close()

for cont in range(len(metadata[0]['submissionDocumentation'][i])):
    with open(path + '/metadata/submissionDocumentation/' + list(metadata[0]
['submissionDocumentation'][i].keys())[cont] + '.csv', 'w', newline='') as csvFile:
        writer = csv.writer(csvFile, delimiter=',')
        writer.writerow(list(list(metadata[0]['submissionDocumentation'][i].values())[cont].keys()))
        writer.writerow(list(list(metadata[0]['submissionDocumentation'][i].values())[cont].values()))

    csvFile.close()

with codecs.open(path + '/metadata/metadata.csv', 'r', encoding='utf8') as transformutf:
    arquivoANSI = transformutf.read()

with codecs.open(path + '/metadata/metadata.csv', 'w', encoding='utf8') as transformutf:
    transformutf.write(arquivoANSI)

transformutf.close()

try:
    print('\nBaixando Diário ' + str(nomeArquivo[i].replace("'", "")) + "...")
    urldje = 'https://pesquisadje-api.tjdft.jus.br/vl/diarios/pdf/' + PDF[i][0] + '/' + PDF[i][1] +
'.pdf'
    urldjeps = 'https://dje.tjdft.jus.br/dje/jsp/dje/DownloadDeDiario.jsp?dj=DJ'+ PDF[i][1]+'_'+PDF[i]
[0]+'-ASSINADO.PDF.P7S&statusDoDiario=ASSINADO'
    urllib.request.urlretrieve(urldje, path + '/objects/' + nomeArquivo[i].replace("'", "") + '.pdf')
    urllib.request.urlretrieve(urldjeps, path + '/objects/' + nomeArquivo[i].replace("'", "") + '.p7s')

except Exception:
    print(colorErro + "Erro ao baixar PDF não será feito o arquivamento." + colorPass)

envioArquivematica(path, infoServer, raiz, nomeArquivo[i], identificador)

elif ConsultaBanco.sistema == 'pje':
    nomeArquivo = metadata['metadados'][1][1]
    raiz = pastaRaiz + '/documentos/'
    path = raiz + nomeArquivo # Caminho onde o arquivo será criado
    identificador = metadata['metadados'][1][1]
    documento = metadata['objects']

    print('Criando estrutura de pastas para envio...')
    if not os.path.isdir(path):
        os.makedirs(path)

    if not os.path.isdir(path + '/metadata'):
        os.makedirs(path + '/metadata')

    if not os.path.isdir(path + '/metadata/submissionDocumentation'):
        os.makedirs(path + '/metadata/submissionDocumentation')

    if not os.path.isdir(path + '/objects'):
        os.makedirs(path + '/objects')

    idDocumentos = []
    for a in range(len(documento)):
        if documento[a]['_value_1'] is not None:
            for k in documento[a]['_value_1']:
                if str(type(k).__name__) == '_Element':
                    if k.tag == '{http://www.cnj.jus.br/intercomunicacao-2.2.3}conteudo':
                        tipoDocumento = documento[a]['_value_1'].attrib
                        formato = (tipoDocumento['mimetype'].split('/')[1])

                        text_filel = open(path + "/objects/" + tipoDocumento['idDocumento'] + "." + formato,
"wb")

                        text_filel.write(base64.b64decode(k.text))
                        text_filel.close()
                        idDocumentos.append(tipoDocumento['idDocumento'])

```



```

formato = (documento[a]['mimetype'].split('/'))[1]
idDocumentos.append(documento[a]['idDocumento'])
text_file = open(path + "/objects/" + documento[a]['idDocumento'] + "." + formato, "wb")
if (formato == 'html'):
    text_file.write(str.encode('<!DOCTYPE html>'))
text_file.write(documento[a]['conteudo'])
text_file.close()

for b in range(len(documento[a]['documentoVinculado'])):
    idDocumentos.append((documento[a]['documentoVinculado'])[b]['idDocumento'])
    formatoDocVinculado = ((documento[a]['documentoVinculado'])[b]['mimetype'].split('/'))[1]
    text_file = open(path + "/objects/" + (documento[a]['documentoVinculado'])[b][
        'idDocumento'] + "." + formatoDocVinculado, "wb")
    if formatoDocVinculado == 'html':
        text_file.write(str.encode('<!DOCTYPE html>'))
    text_file.write(documento[a]['documentoVinculado'][b]['conteudo'])
    text_file.close()

with open(path + '/metadata/metadata.csv', 'w', newline='') as csvFile:
    writer = csv.writer(csvFile, delimiter=',')
    writer.writerow(metadata['metadados'][0])
    writer.writerow(metadata['metadados'][1])
csvFile.close()

for i,x in enumerate(list(metadata['submissionDocumentation'].keys())):
    with open(path + '/metadata/submissionDocumentation/' + x + '.csv', 'w', newline='') as csvFile:
        writer = csv.writer(csvFile, delimiter=',')
        for k in range(len(metadata['submissionDocumentation'][x])):
            writer.writerow(metadata['submissionDocumentation'][x][k])
        csvFile.close()

# with open(path + '/metadata/submissionDocumentation/documentos.csv', 'w', newline='') as csvFile:
#     writer = csv.writer(csvFile, delimiter=',')
#     for i in range(len(metadata['submissionDocumentation'])):
#         writer.writerow(metadata['submissionDocumentation'][i])
#         if metadata['submissionDocumentation'][i][0] not in idDocumentos and i > 0:
#             with open('erros.txt', 'a') as gravarID:
#                 gravarID.write("Erro ao baixar arquivo: " + metadata['submissionDocumentation'][i][
#                     0] + " do processo numero " + metadata['metadados'][1][1] + "\n")
#                 gravarID.close()
#             print(colorErro + "Erro ao baixar: " + metadata['submissionDocumentation'][i][0] +
colorPass)
# csvFile.close()

envioArquivematica(path, infoServer, raiz, nomeArquivo, identificador)

return

def main():
    IDFuturo = []
    xml = ConsultaBanco.root
    if xml:
        retornoConsulta = ConsultaBanco.consultas()
        if retornoConsulta:
            if ConsultaBanco.sistema == 'pje':
                metadados = conx(retornoConsulta[0], retornoConsulta[1], retornoConsulta[2], retornoConsulta[3],
                    retornoConsulta[4], retornoConsulta[5], retornoConsulta[6])

            elif (ConsultaBanco.sistema == 'dje'):
                metadados = geraMetadados(xml)

        if metadados is not None:
            infoServer = infoArquivematica(xml)
            job(metadados, infoServer)

schedule.every(5).seconds.do(main)
while True:
    schedule.run_pending()
    print("Ultima verificação do sistema:" + str(datetime.datetime.now()))
    time.sleep(1)

```

ANEXO C – CÓDIGO CLASSE GERAL HIPÁTIA

ClasseGeral.py

```
from VerificaStatus import verificaStatus
from ProximoArquivo import verificadorDeID
from EnvioArquivematica import envioArchivematica
from bagit import criarBagit

#Classe com funções generalizadas
class Geral:
    """
        Executa cada uma das etapas necessárias para a criação do pacote OAIS.
    """
    metadatas=0
    enviosSequenciais=0

    #Variaveis geradas dentro de funcoes da classe
    #sistema='Pje' ou 'Dje' (De acordo com o fornecido ao inicializar o codigo)
    #pastaRaiz= path onde este script esta localizado
    #root= XML dos metadados lido (O nome do xml deve ser igual a da variavel sistema, ie. Pje.xml ou
Dje.xml)
    #infoServer= array com todos os dados necessarios para acessar o servidor do archivematica
    #cursor = retorno de conexaoComBanco
    #retornoConsultas= retorno de consultas
    #metadatas= Dicionario com todos os metadados gerados do sistema

    #Inicializa a classe
    def __init__(self, sistema, qtdEnvios):
        self.sistema=sistema
        self.qtdEnvios=qtdEnvios
        self.recuperaXML()
        print (colorFinish+'### Busca do XML realizada com sucesso ###'+colorPass)
        self.infoArchivematica()
        print (colorFinish+'### Informações Archivematica buscadas com sucesso ###'+colorPass)
        return

    #Obtem o arquivo XML
    def recuperaXML(self):
        """
            Mapeia o obtêm o arquivo XML.
        """
        pastaRaiz = os.path.abspath(__file__) # Recupera o path do script python
        self.pastaRaiz= (os.path.split(pastaRaiz))[0] # Recupera o path do projeto
        root = (ET.parse(self.pastaRaiz + '/sistemas/'+self.sistema + '/' + self.sistema+'.xml')).getroot()
# Abre o arquivo XML
        self.root = root
        return

    # Obtem as informacoes do archivematica
    def infoArchivematica(self):
        """
            Executa a função de extrair as informações do Archivematica.
        """
        self.infoServer = []

        # Recupera informações do servidor do archivematica
        for servidor in self.root.iter('fileserver'):
            for mapping_tags in servidor:
                self.infoServer.append(mapping_tags.text)
        return self.infoServer
```



```

# Verifica o status dos arquivos enviados na ultima execução bem-sucedida.
### Arquivo verificaStatus.py
def status(self):
    """
        Executa a função de verificação do status da transferência.

    """
    return verificaStatus(self.infoServer, self.pastaRaiz)

# Realiza a obtenção dos dados
### Arquivo {SISTEMA}/ConexaoBanco
def conexaoBanco(self):
    """
        Executa a função de conexão com o banco de dados do DJe.

    """
    exec('from sistemas.'+ self.sistema +'ConexaoBanco'+self.sistema+' import *',globals())
    IdContinua = verificadorDeID(self.pastaRaiz,self.sistema)
    self.IdContinua = IdContinua
    self.cursor=conexaoComBanco(self.root.iter('conections'),IdContinua)

    return

# Cria e trata a variavel contendo as informações que foram obtidas
### Arquivo {SISTEMA}/Consulta{SISTEMA}.py
def geradorConsultas(self):
    """
        Executa a função de consulta dos metadados no banco de dados.

    """
    exec('from sistemas.'+ self.sistema +'Consulta'+self.sistema+' import *',globals())

self.retornoConsultas=consultas(self.root.iter('metadatas'),self.cursor,self.enviosSequenciais,self.IdContinua)

    return

# Função para geração dos metadados importado do arquivo Metadatas + 'Nome do projeto'
def geradorMetadatas(self):
    """
        Executa a função de extração de metadados.

    """
    exec('from sistemas.'+ self.sistema +'Metadatas'+self.sistema+' import *',globals())
    self.metadatas=geraMetadatas(self.root,self.retornoConsultas,self.cursor)

# Função job
def executarJob(self):
    """
        Executa a função job.

    """
    exec('from sistemas.'+ self.sistema +'Job'+self.sistema+' import *',globals())
    pastaArquivos = self.pastaRaiz+'/documentos'

    if (self.sistema == 'Dje'):
        metadados = self.metadatas[0]['metadados']
        subDoc = self.metadatas[0]['submissionDocumentation']
        nomesArquivos = self.metadatas[1]
        PDF = self.metadatas[2]

        for i in range(len(metadados)):
            identificador = metadados[i][1][16]
            pastaArquivo = pastaArquivos + '/' + (nomesArquivos[i]).replace("'", "")
            nomeArquivo = (nomesArquivos[i]).replace("'", "").replace(" ", "o")

            criarDiretorios(pastaArquivo)
            criarCsv(pastaArquivo, metadados[i])
            criarSubmissionDocumentation(pastaArquivo, subDoc, i)
            baixarArquivos(nomeArquivo, PDF, i, pastaArquivo)

```

```

        # criarCheckSum(pastaArquivo)
        criarBagit(pastaArquivo,pastaArquivos,nomeArquivo)
        envioArchivematica(self.infoServer, pastaArquivo, nomeArquivo, identificador,
self.pastaRaiz)

    elif (self.sistema == 'Pje'):
        metadados = self.metadatas
        nomeArquivo = metadados['metadados'][1][3]
        pastaArquivo = pastaArquivos+'/'+nomeArquivo

        identificador = self.IdContinua
        criarDiretorios(pastaArquivo)
        criarCsv(pastaArquivo,metadados)
        criarSubmissionDocumentation(pastaArquivo,metadados['submissionDocumentation'],0)
        baixarArquivos('',metadados,0,pastaArquivo)
        criarBagit(pastaArquivo,pastaArquivos,nomeArquivo)
        envioArchivematica(self.infoServer, pastaArquivo, nomeArquivo, identificador, self.pastaRaiz)

        self.enviosSequenciais=self.enviosSequenciais+1
        if (self.qtdEnvios > self.enviosSequenciais):
            novo.runAll()

    elif (self.sistema == 'An'):
        print ('')

    else:
        print('Sistema não encontrado!')

# Função para rodar todas as funções
def runAll(self):
    """
        Executa todas as funções necessárias para funcionamento do barramento.
    """
    self.status()
    print(colorFinish+'### Realizado atualização dos arquivos texto! ###'+colorPass)

    self.conexaoBanco()
    print(colorFinish+'### Conexão com o banco realizada com sucesso! ###'+colorPass)

    self.geradorConsultas()
    print(colorFinish+'### Dados recuperados com sucesso! ###'+colorPass)

    self.geradorMetadatas()
    print(colorFinish+'### Gerado metadatas para criação do CSV ###'+colorPass)

    self.executarJob()
    #path, infoServer, raiz, nomeArquivo, identificador, pastaRaiz = self.executarJob()
    print(colorFinish+'### Informações tratadas, enviando para o Archivematica ###'+colorPass)
    return

if __name__ == '__main__':
    novo=Geral('Pje',1)
    novo.runAll()

```

SAS - Quadra 05 - Lote 06 -
Bloco H - Sobreloja
Cep: 70070-912 - Brasília / DF

Telefone: +55 61 3217 6213
E-mail: shintaku@ibict.br