

## **Agradecimentos**

a fazer

## **Resumo**

a fazer

## **Lista de Abreviaturas e Siglas**

- API: Application Programming Interface
- CI: Ciência da Informação
- CIDOC: International Committee for Documentation
- CMS: Collection Management System
- CMS: Content Management System
- CRM: Conceptual Reference Model
- ICOM: International Council of Museums
- JSON: JavaScript Object Notation
- LOD: Linked Open Data
- LOUD: Linked Open Usable Data
- Spectrum: Standard ProCedures for Collections Recording Used in Museums
- XML: Extensible Markup Language

## **1. Introdução**

A Sociedade da Informação impõe novos desafios para as organizações e um desafio maior ainda para as que têm como responsabilidade a organização, preservação e disseminação do conhecimento e patrimônio, como arquivos, bibliotecas, e museus. Esses desafios ficaram mais evidentes quando fomos assolados pela pandemia da covid-19 em 2020, que fez com que museus do mundo inteiro fechassem suas portas para preservar a vida dos funcionários e visitantes, e tornou o remoto praticamente a única realidade de trabalho, estudo,

entretenimento e interação social. Embora os museus já estivessem fazendo uso de Tecnologias da Informação e Comunicação (TICs), vimos uma aceleração deste movimento nestes dois anos de isolamento, com os museus buscando novas formas de interação com o público, como afirma Ana Magalhães: “a pandemia acelerou um processo que já vinha sendo explorado lentamente pelos museus, que é a sua mediação com o público através da dimensão digital” (DERVICHE, 2021).

Esta pesquisa tem como ponto de partida, o desenvolvimento do aplicativo Elucidário.art, projeto de design digital de um site para divulgar a coleção reunida ao longo da vida de Ema Klabin, realizado na pós-graduação em Comunicação e Design Digital na Escola Superior de Propaganda e Marketing (ESPM), a partir de minha experiência como coordenador de comunicação na Casa Museu Ema Klabin. O resultado obtido foi um protótipo de alta fidelidade e uma dissertação<sup>[1]</sup> com foco na usabilidade do usuário final: o visitante do museu.

O site passou a ser desenvolvido e, em 2019, foi lançado oficialmente com o nome Explore a Coleção! (EXPLORE) sob o endereço <https://emaklabin.org.br/explore>. Os dados foram migrados diretamente da base de dados museológica da casa museu, passando por poucos procedimentos de padronização da informação, pois o objetivo da ferramenta era a disponibilização para consulta online da base de dados tal como ela estava.

Após o lançamento do site foi diagnosticada a necessidade de atualizar as bases de dados para que as informações estivessem em padrões internacionais de descrição e representação de itens museológicos, como Linked-art, e que os processos museológicos passassem a aderir metodologias como Standard Procedures for Collections Recording Used in Museums (Spectrum). Portanto, o problema desta pesquisa é a obsolescência do sistema de gestão de documentação museológica da Casa Museu Ema Klabin, que necessita de uma solução mais robusta que permita a gestão de sua documentação e divulgação da coleção utilizando princípios de Linked Open Data, padrões internacionais de representação e recuperação da informação museológica, e que essa solução possa ser utilizada também por museus de pequeno e médio porte.

O objetivo geral desta pesquisa é o desenvolvimento do aplicativo Elucidário.art para gestão de documentação museológica com foco em museus mistos, que atenda a

especificações internacionais de interoperabilidade da informação e possibilite a representação, visualização e recuperação destas informações, e para atingirmos o objetivo geral, definimos os seguintes objetivos específicos:

- Definir informação e documentação museológica para o escopo do trabalho.
- Identificar os modelos conceituais e padrões internacionais de representação da informação museológica que serão utilizados no aplicativo.
- Desenvolver efetivamente o aplicativo Elucidário.art, com base nos modelos conceituais e padrões internacionais de representação da informação museológica identificados.
- E, por fim, testar o uso do aplicativo com um recorte específico da coleção da Casa Museu Ema Klabin de 50 itens da BD Museológica e informações correlatas, como agentes, eventos, lugares, conceitos, etc. A escolha destes itens se deu por terem passado por processo recente de digitalização. De um total de xxx itens digitalizados, este recorte de 50 itens foi escolhido por ser um número que permite a realização de testes e experimentos com o aplicativo, além de representarem diversos tipos de itens, como pinturas, esculturas, mobiliário, etc, e diversos níveis de documentação diferente como empréstimos, exposições, restaurações, etc.

No capítulo 2 percorremos por todos os processos metodológicos utilizados na pesquisa, sejam eles de natureza teórica ou prática, como a revisão bibliográfica, metodologia para desenvolvimento do plugin, e o formato e execução dos testes de validação.

O capítulo 3 apresenta uma contextualização da informação debatendo autores como Aldo de Albuquerque Barreto, Le Codiac, Rafael Capurro, Michael K. Buckland, e outros. Também trazemos a informação para o contexto museológico buscando definir o que é informação museológica. No capítulo 4 aprofundamos o debate da informação museológica ao descrever alguns modelos metadados de representação utilizados para representar estas informações, como Object ID (1997), Esquema de metadados para descrição de obras de arte em museus brasileiros: uma proposta (SILVA, 2020), Spectrum (2017) e Linked Art (2021).

Já no capítulo 5, falamos sobre a documentação percorrendo os procedimentos do Spectrum para gestão de coleções, e no capítulo 6, apresentamos a Casa Museu Ema Klabin, sua coleção e documentação museológica.

No capítulo 7, apresentamos o Elucidário.art ao descrever os três pacotes principais que compõem o plugin: @elucidario/pkg-mdorim, @elucidario/pkg-design-system e @elucidario/pkg-core, em que apresentamos seus escopos, definições e testes utilizados para validação das funcionalidades desenvolvidas. Também apresentamos como se dará o uso do plugin na Casa Museu Ema Klabin.

Por fim, no capítulo 8, apresentamos as considerações finais e as perspectivas de continuidade desta pesquisa.

## **2. Procedimentos metodológicos**

Como o objetivo da pesquisa é o desenvolvimento do aplicativo Elucidário.art e a realização de seus testes, utilizamos diferentes métodos para obtenção dos resultados:

- **Revisão bibliográfica**

Utilizamos livros, artigos, teses, dissertações, legislações, manuais, normas e padrões, nos temas de documentação museológica; interoperabilidade da informação; representação, visualização e recuperação da informação e desenvolvimento de aplicativos, para delimitar o tema da pesquisa e embasar o desenvolvimento do aplicativo Elucidário.art. Buscamos em sites de organizações internacionais como ICOM, Collections Trusts, CHIN, entre outras e nacionais como IBRAM, SISEM, FCC, entre outras. Também buscamos em anais de conferências como CIDOC e ENANCIB.

Realizamos uma análise detalhada de padrões de metadados para objetos de arte como o Esquema de Metadados para Descrição de Obras de Arte em Museus Brasileiros (SILVA, 2020), Object ID (CONSELHO INTERNACIONAL DE MUSEUS, 1999) e Linked Art [(LINKED ART, 2021), para identificar os elementos de metadados que serão utilizados no aplicativo Elucidário.art. Utilizamos também como base os procedimentos do Spectrum (COLLECTIONS TRUST, 2022) para definir os fluxos de trabalho do aplicativo.

- **Desenvolvimento do aplicativo Elucidário.art**

O desenvolvimento de um software é uma tarefa extremamente complexa. É necessário dar conta de diferentes níveis de abstração, desde a concepção do produto até a sua implementação. A grande maioria das metodologias de desenvolvimento são pensadas para equipes com diferentes papéis e responsabilidades dentro do sistema, como desenvolvedores, analistas de negócio, gerentes de projeto, etc. Nas metodologias ágeis, em que na verdade não existem processos ou metodologias Ágeis e sim equipes Ágeis, o que é descrito como Ágil seria o ambiente para uma equipe aprender a ser Ágil (WELLS, 2009). O que torna essas metodologias inviáveis para um desenvolvedor que trabalha sozinho.

Por exemplo, a metodologia Ágil Scrum (1990), que estipula uma reunião no início de cada “Sprint” (ciclo de desenvolvimento que pode durar de 1 a 4 semanas) para definição da História de Usuário (Caso de Uso) que será trabalhada durante o ciclo, e também prevê reuniões diárias para acompanhamento do projeto (SCHWABER; SUTHERLAND, 2020).

Uma outra metodologia Ágil bastante utilizada é a Extreme Programming (XP) (1996) (WELLS, 1999), que tem como objetivo satisfazer o cliente e entregar software de valor a ele o mais rápido possível. Para isso, a XP utiliza práticas como: programação em pares, desenvolvimento orientado a testes, integração contínua, refatoração, entrega contínua, e diversas outras regras e condutas de desenvolvimento (EXTREMEPROGRAMMINGROADPMAP, 2006). O problema da XP para um desenvolvedor que trabalha sozinho é que ela prevê que o desenvolvimento seja feito em pares, o que não é possível, mas as outras etapas e práticas podem ser utilizadas. Na verdade, na Wiki da XP possui inclusive uma página para demonstrar casos de uso da XP para desenvolvedores que trabalham sozinhos (WIKI EXTREME PROGRAMMING, 2014), as práticas que podem ser utilizadas são:

- *Engineering Task* ou Tarefa de Engenharia, que é uma tarefa que deve ser realizada para que o sistema funcione corretamente;
- *User Stories* ou Caso de Uso, que são descrições de uma parte do sistema que o usuário interage para resolver determinado problema, cada *User Stories* é dividido em diferentes *Engineering Tasks*;

- *Iteration Plan* ou Plano de Iteração, que é um plano de curto prazo que descreve o que será feito durante a iteração, lista todas as *User Stories* que serão implementadas;
- *Release Plan* ou Plano de Entrega, que é um plano de médio prazo que descreve quando as funcionalidades serão entregues e lista os *Iteration Plans* que serão implementados;

Também podemos utilizar as práticas recomendadas do XP:

- *Refactor Mercilessly* ou Refatore sem piedade, que é a prática de refatorar o código sempre que possível, para que ele fique mais simples e fácil de entender, esta prática tem como intuito evitar a criação de código duplicado, ou *Once and Only Once*, outra prática do XP;
- *Relentless Testing* ou Teste implacavelmente, que é a prática de escrever testes automatizados para todas as funcionalidades do sistema, para que seja possível realizar testes de regressão, ou seja, testar se as funcionalidades que já funcionavam continuam funcionando após a implementação de novas funcionalidades, uma forma de garantir isto é: “the feature doesn’t actually exist until there are tests that prove it works” (WIKI EXTREME PROGRAMMING, 2009);
- *Continuous Integration* ou Integração Contínua, que é a prática de integrar o código ao repositório de código principal o mais rápido possível, para que seja possível realizar testes de integração e testes de regressão automatizados;

Nesta pesquisa estabelecemos as seguintes etapas de desenvolvimento:

- *Plano de entrega*: um conjunto de funcionalidades que serão implementadas em um determinado período de tempo desenhadas baseadas nas *User Stories*;
- *Plano de iteração*: um conjunto de tarefas que serão realizadas em um determinado período de tempo desenhadas baseadas nas *User Stories*;
- *Caso de uso*: utilizaremos os procedimentos Spectrum para delimitar os Casos de Usos, como entrada de objetos, saída e entrada para empréstimos, controle de localização e entre outros, que serão implementados no aplicativo;

- *Tarefa de engenharia*: tarefas que serão realizadas para implementar as *User Stories*, como por exemplo, a criação de um banco de dados, a criação de uma interface de usuário, a criação das páginas administrativas, a criação do modelo de dados, etc;
- **Testes do aplicativo Elucidário.art**

Os testes do aplicativo podem ser divididos em diferentes tipos:

- **Testes unitários**: testes que verificam se uma unidade de código funciona corretamente, por exemplo, se uma função retorna o valor esperado;
- **Testes de integração e regressão**: testes que verificam se as diferentes unidades de código funcionam corretamente quando integradas e se continuam funcionando após a implementação de novas funcionalidades;
- **Testes de sistema**: testes que verificam se o sistema funciona corretamente, por exemplo, se o sistema consegue realizar as tarefas que foram definidas nas *User Stories*;
- **Testes de aceitação**: testes que verificam se o sistema funciona corretamente para o usuário, por exemplo, se o sistema consegue realizar as tarefas que foram definidas nas *User Stories* para o usuário;
- **Testes de usabilidade**: testes que verificam se o sistema é fácil de usar, por exemplo, se o usuário consegue realizar as tarefas que foram definidas nas *User Stories* de forma fácil;
- Testes unitários e Testes de integração

Como o aplicativo é desenvolvido em diferentes linguagens de programação, PHP, TypeScript, Json, e utiliza frameworks, bibliotecas e ferramentas como WordPress, React, React-hook-form, Radix-UI, AJV Schema, entre outras, estes testes serão realizados utilizando as ferramentas para teste de cada linguagem, framework, biblioteca e ferramenta. Por exemplo, os testes unitários do PHP serão realizados utilizando PHPUnit, os testes

unitários do TypeScript serão realizados utilizando Jest, os testes unitários do Json serão realizados utilizando AJV Schema, e assim por diante.

- Testes de sistema

Os testes de sistema serão realizados utilizando o Cypress, que é uma ferramenta de teste de sistema que permite realizar testes de aceitação, testes de regressão e testes de usabilidade. O Cypress permite realizar testes de sistema em diferentes navegadores, como Chrome, Firefox, Edge, Safari, Electron, entre outros, e também permite realizar testes de sistema em diferentes dispositivos, como computadores, tablets e celulares.

### **3. A Informação**

Esta pesquisa se insere no domínio da Ciência da Informação por ter como objeto a informação museológica que necessita ser descrita, representada e recuperada no âmbito dos Sistemas de Informação da Casa Museu Ema Klabin. Para uma melhor compreensão desta informação museológica é necessário definirmos o termo informação.

Le-Codiac afirma que “a informação comporta um elemento de sentido. É um significado transmitido a um ser consciente por meio de uma mensagem inscrita em um suporte espacial-temporal: impresso, sinal elétrico, onda sonora, etc”. Ela tem por objetivo o conhecimento, ou “apreensão de sentidos ou seres em sua significação” (LE COADIC, 1994).

A informação, como afirma Barreto (BARRETO, 2007) “sintoniza o mundo, pois referencia o homem ao seu passado histórico, às suas cognições prévias e ao seu espaço de convivência, colocando-o em um ponto do presente”. O autor também afirma que os fluxos de informação se movem em dois níveis, sendo o primeiro: os fluxos internos de informação, que “se movimentam entre os elementos de um sistema de armazenamento e recuperação da informação, e se orientam para sua organização e controle” (BARRETO, 2007); e os fluxos extremos: a informação gerada por um autor entra no sistema para ser tratada e assimilada como conhecimento, utilizando processos de entrada, seleção, classificação, armazenamento, recuperação e uso. No fluxo extremo temos um processo de cognição “uma interiorização da informação para um subjetivismo privado” (BARRETO,



2007), e ao mesmo tempo, uma “desapropriação cognitiva, quando o pensamento, do gerador, se arranja em informação, em uma linguagem com inscrições próprias.” (BARRETO, 2007).

Capurro e Hjørland (CAPURRO; HJORLAND, 2007) discorrem sobre a natureza interdisciplinar da informação ao afirmar que quase toda disciplina científica utiliza o conceito de informação dentro de seu próprio contexto, os autores analisaram o livro *The study of information: interdisciplinary messages* de Machlup e Mansfield para coletar visões chave “sobre a controvérsia da interdisciplinaridade em ciência da computação, inteligência artificial, biblioteconomia e CI, linguística, psicologia e física, bem como nas ciências sociais” (CAPURRO; HJORLAND, 2007), essa controvérsia se dá pelo fato de que para Machlup “a informação é um fenômeno humano. Envolve indivíduos transmitindo e recebendo mensagens no contexto de suas ações possíveis” (CAPURRO; HJORLAND, 2007). Os autores seguem a análise apresentando novas teorias relacionadas a naturalização da informação: *Information: New questions to a multidisciplinary concept* de Kornwachs e Jacoby (1996) e *Can information be naturalized?* de Zoglauer, este último respondendo negativamente à pergunta do título em relação à informação semântica e pragmática, ou “qualquer unidade semiótica mente-dependente, bem como de informação funcional cujo interpretador pode ser uma máquina de Turing e/ou qualquer tipo de organismo vivo processando informação neural ou genética.” (CAPURRO; HJORLAND, 2007). Capurro e Hjørland apresentam o conceito de informação no contexto da engenharia de transmissão de sinais desenvolvida por Shannon e Weaver (1972) em que seu sentido não pode ser confundido com significado:

A palavra informação, nesta teoria, é usada em um sentido especial que não deve ser confundido com seu uso comum. Em particular, a informação não deve ser confundida com significado. Na verdade, duas mensagens, uma das quais é intensamente carregada de significado e outra que é pura falta de sentido, podem ser exatamente equivalentes, segundo o ponto de vista desta teoria, com relação à informação. É isto, sem dúvida, que Shannon quer dizer quando afirma ‘os aspectos semânticos da comunicação são irrelevantes para os aspectos da engenharia’. Mas isto não significa que os aspectos da engenharia são necessariamente irrelevantes para os aspectos semânticos. (SHANNON e WEAVER, 1972, p.8 apud CAPURRO E HJORLAND, 2007. p.162)

Ainda sobre a interdisciplinaridade da informação, Capurro e Hjørland (2007) utilizam a frase de Norbert Wiener “Informação é informação, não matéria ou energia. Nenhum materialismo que não admita isto pode sobreviver na atualidade” (WIENER, 1961, p.132 apud CAPURRO e HJORLAND, 2007 p.162) para justificar a controvérsia filosófica do século XX com origem na cibernética sobre o conceito de informação.

Saracevic (1996) afirma que entre pioneiros da CI “havia engenheiros, bibliotecários, químicos, linguistas, filósofos, psicólogos, matemáticos, cientistas da computação, homens de negócios e outros vindos de diferentes profissões ou ciências” (SARACEVIC, jan./jul. 1996), mas cita quatro campos que desenvolveram uma relação com a informação mais contundente: biblioteconomia, ciência da computação, ciência cognitiva e comunicação. Em relação a biblioteconomia, Saracevic (1996) esclarece que o ponto central de conexão entre as áreas seria justamente o “papel social e sua preocupação comum com os problemas da efetiva utilização dos registros gráficos” (SARACEVIC, jan./jul. 1996) e expõe cinco diferenças entre as áreas que garantem sua separação como áreas independentes de estudo:

1 - seleção dos problemas propostos e a forma de sua definição; 2 - questões teóricas apresentadas e os modelos explicativos introduzidos; 3 - natureza e grau de experimentação e desenvolvimento empírico, assim como o conhecimento prático/competências derivadas; 4 - instrumentos e enfoques usados; e 5 - a natureza e a força das relações interdisciplinares estabelecidas e sua dependência para o avanço e evolução dos enfoques interdisciplinares. (SARACEVIC, jan./jul. 1996).

Em relação à Ciência da Computação, Saracevic esclarece que a base comum com a CI “reside na aplicação dos computadores e da computação na recuperação da informação, assim como nos produtos, serviços e redes associados” (SARACEVIC, jan./jul. 1996). Já em relação à Ciência Cognitiva, que é a ciência que estuda os processos cognitivos, o conhecimento e a inteligência, uma relação forte com a CI fica a cargo da Inteligência Artificial (IA) que pode ser definida:

como um ramo da engenharia de software IA é um conjunto de técnicas de programação que fazem o computador executar alguns truques... Certamente algumas pessoas estão começando a ganhar rios de dinheiro com estes truques. como uma teoria da ciência da computação, ... IA é uma concepção ímpar do que seja programação... como um ramo da filosofia, IA é um tipo de epistemologia experimental: o que é o

conhecimento? como pode o conhecimento ser representado no computador - ou na mente? como uma ciência da mente, IA uma ideia controversa e instigante: que a mente ... É basicamente um mecanismo processador de informações ... Em seu nível mais profundo, IA liga-se a um dos grandes mistérios não resolvidos da ciência: como pode a mente surgir de uma não-mente? Como pode o cérebro, um objeto feito de matéria comum... (WALDROP, 1987 apud SARACEVIC, 1996, p.52).

Tanto os dois primeiros pontos, ou IA fraca que pode ser fonte de inovações em sistemas de informação, quanto os dois últimos, ou IA forte, que é o modelo teórico da cognição, tem interesse direto para a CI (SARACEVIC, 1996, p.50). Por último, a relação com a Comunicação reside na informação como fenômeno e a comunicação como processo, além de outras dimensões como: “interesse compartilhado na comunicação humana (...); confluências de certas correntes de pesquisa; algumas permutas entre professores; e o potencial de cooperação da prática profissional e dos interesses comerciais/empíricos.” (SARACEVIC, jan./jul. 1996).

A obra “Epistemologia e Ciência da Informação” de Rafael Capurro é um texto seminal para a Ciência da Informação onde o autor defende a tese de que a ciência da informação nasce em meados do século XX com um paradigma físico, questionado por um enfoque cognitivo idealista e individualista, sendo este por sua vez substituído por um paradigma pragmático e social (CAPURRO, 2003)

Capurro parte do conceito de paradigma utilizado por Thomas Kuhn em sua análise da estrutura das revoluções científicas com múltiplas conotações (KUHN 1962/1970, MASTERMANN, 1970, apud CAPURRO, 2003). A noção de paradigma parte da pertinência de combinar enfoques históricos e epistemológicos para o estudo do processo evolutivo de uma disciplina (VEGA-ALMEIDA; FERNÁNDEZ-MOLINA; LINARES, 2009). Segundo Capurro, paradigma é:

Um modelo que nos permite ver uma coisa em analogia a outra. Como toda analogia, chega o momento em que seus limites são evidentes, produzindo-se então uma crise ou, como no caso de teorias científicas, uma “revolução científica”, na qual se passa da situação de “ciência normal” a um período “revolucionário” e em seguida a um novo paradigma. (CAPURRO, 2003)

Para Capurro (2003) o paradigma físico consiste em essência na informação como algo físico, um objeto transmitido de um emissor para o receptor. Este paradigma é embasado pelas teorias de Shannon, que trata a informação como mensagem, ou como signos que devem permanecer iguais entre o emissor e o receptor sem fontes de ruído (CAPURRO, 2003). Sua origem data no pós-guerra e marca o início do desenvolvimento da CI, ainda não como uma disciplina institucionalizada, algo que só viria a acontecer no ano de 1962 na conferência do Georgia Institute of Technology em que o termo “documentação” foi rejeitado e no seu lugar definiu pela primeira vez a Ciência da Informação (VEGA-ALMEIDA; FERNÁNDEZ-MOLINA; LINARES, 2009) como “uma ciência que investiga as propriedades e comportamento da informação” (BORKO, 1969, apud. VEGA-ALMEIDA, FERNÁNDEZ-MOLINA, LINARES, 2009) com caráter duplo: “um componente de ciência pura que investiga seu objeto sem considerar suas aplicações”, e um ‘componente de ciência aplicada que desenvolve produtos e serviços” (TAYLOR, 1966, apud. VEGA-ALMEIDA, FERNÁNDEZ-MOLINA, LINARES, 2009).

Embora os paradigmas possam ser delimitados em faixas temporais, sendo o paradigma físico definido entre os anos de 1945 e meados dos anos 1970, Michael Buckland em 1991 propõe a informação-como-coisa (*information-as-thing*), ou seja, objetos tangíveis, como documentos, livros, itens de museus, ou qualquer outro objeto que possa ter valor informativo (CAPURRO, 2003).

Buckland identifica três usos principais da palavra informação: informação-como-processo (*information-as-process*), informação-como-conhecimento (*information-as-knowledge*) e informação-como-coisa (*information-as-thing*) (BUCKLAND, 1991). A informação como conhecimento tem a intangibilidade como característica principal, ela não pode ser tocada ou medida, mas precisa ser descrita e representada em um suporte físico para ser transmitida, seja por sinal, texto ou comunicação, e qualquer que seja a expressão, descrição ou representação utilizada para sua transmissão, será *information-as-thing* (BUCKLAND, 1991). Buckland esclarece que embora outros autores tenham objeções ao uso do termo informação para denotar uma coisa em seu sentido estrito, como afirma Wiener: “informação é informação, nem matéria nem energia” (MACHLUP, 1983, apud BUCKLAND, 1993, p.1), a linguagem evoluiu e com ela a expansão da TI e a prática de se

referir a comunicações, base de dados, livros e outros itens como informação se tornou mais comum com o passar dos anos (BUCKLAND, 1991).

Já o paradigma cognitivo tem um enfoque mais individualista e busca entender os processos cognitivos de geração de conhecimento no usuário

Sua perspectiva permanece cognitiva no sentido de que se trata de ver de que forma os processos informativos transformam ou não o usuário, entendido em primeiro lugar como sujeito cognoscente possuidor de “modelos mentais” do “mundo exterior” que são transformados durante o processo informacional. (CAPURRO, 2003)

Capurro esclarece que o paradigma cognitivo surge na necessidade do usuários buscar informação, ou seja, o “estado cognitivo anômalo” (BELKIN, 1980; BELKIN, ODDY, BROOKS, 1982 apud CAPURRO, 2003), no qual o conhecimento que o usuário tem não é suficiente para resolver determinado problema. A informação é vista como um conceito diretamente envolvido com a compreensão e o processamento cognitivo (VEGA-ALMEIDA; FERNÁNDEZ-MOLINA; LINARES, 2009)

O paradigma social é colocado por Frohmann como uma crítica ao paradigma cognitivo, que é visto como reducionista, pois é considerado não só como idealista, mas também como associal” (CAPURRO, 2003). Capurro (2003) traz a tona o paradigma social-epistemológico desenvolvido por Hjørland e Albrechtsen, chamado de ““domain analysis” no qual o estudo de campos cognitivos está em relação direta com comunidades discursivas, ou seja, com distintos grupos sociais e de trabalho que constituem uma sociedade moderna” (CAPURRO, 2003). O objeto de estudo da CI são as relações entre os discursos, áreas de conhecimento e documentos com distintas comunidades de usuários (HJORLAND 2003, apud CAPURRO, 2003), o que significa uma integração da perspectiva individualista do paradigma cognitivo dentro do contexto social.

Buckland (1991) questiona: “Why do centers of research assemble many sorts of collections of objects if they do not expect students and researchers to learn something from them?”. Esta pergunta é central nas áreas de documentação museológica e organização do conhecimento, em que o foco é a representação, organização e recuperação da informação

para que o usuário possa compreender, transmitir e retroalimentar um sistema informacional.

Objetos que não são necessariamente documentos textuais, podem, mesmo assim, ser fontes de informação—*information-as-thing* (BUCKLAND, 1991, p. 4). Esta parece ser a primeira e mais óbvia concepção quando falamos da informação museológica. O objeto de museu, ou *musealia*, é o objeto que foi musealizado, que teve seu estatuto modificado pela musealização (CURY, 2017, p.5) “a musealidade é atribuída e pode ocorrer por critérios determinados por especialista e/ou grupos culturais através da participação nos processos de musealização” (CURY, Marília Xavier. *Metamuseologia*, 2017), isso significa que foi atribuído, ou percebido valor histórico, sociológico, artístico, antropológico ou científico no objeto.

Um “objeto de museu” é uma coisa musealizada, sendo “coisa” definida como qualquer tipo de realidade em geral. A expressão “objeto de museu” quase poderia passar por pleonismo, na medida em que o museu é não apenas um local destinado a abrigar objetos, mas também um local cuja função principal é a de transformar as coisas em objetos. (DESVALLÉES, André. MAIRESSE, François. *Conceitos-chave de Museologia*, 2013)

A representação é parte fundamental para a organização da informação. Buckland (1991) afirma que o conhecimento pode ser representado, assim como um evento pode ser filmado, e que, qualquer forma de representação estará necessariamente em forma tangível, como um sinal, dados, texto, filme, etc, ou seja, representação é necessariamente *information-as-thing*. Além do que um objeto museológico representa, como sua cultura, seu contexto histórico, o ponto de vista de um artista, entre outras muitas camadas subjetivas, também precisamos da representação de suas características físicas, como dimensões, técnicas, forma de acondicionamento, laudos e entre outras. A representação de um objeto museológico pode ser feita por meio de descrição, fotografias, vídeos, digitalizações, modelos tridimensionais, todos essencialmente *dados* que precisam ser organizados e armazenados em um sistema para facilitar a retroalimentação desta informação.

O primeiro valor do Código de Ética para Museus do ICOM (2009) define o museu como o responsável pelo “patrimônio natural e cultural, material e imaterial”. A definição anterior de museus, em vigor de 2007 a 2022, dizia que:

O museu é uma instituição permanente, sem fins lucrativos, ao serviço da sociedade e do seu desenvolvimento, aberta ao público, que adquire, conserva, investiga, comunica e expõe o património material e imaterial da humanidade e do seu meio envolvente com fins de educação, estudo e deleite (ICOM, 2022).

Já a atual definição de museus, adotada em 24 de agosto de 2022 durante a Conferência Geral do ICOM em Praga, coloca que:

Um museu é uma instituição permanente, sem fins lucrativos e ao serviço da sociedade que pesquisa, coleciona, conserva, interpreta e expõe o património material e imaterial. Abertos ao público, acessíveis e inclusivos, os museus fomentam a diversidade e a sustentabilidade. Com a participação das comunidades, os museus funcionam e comunicam de forma ética e profissional, proporcionando experiências diversas para educação, fruição, reflexão e partilha de conhecimentos (ICOM, 2022).

Nas duas definições podemos perceber que a informação museológica, mesmo que não citada diretamente, se faz presente em verbos como conserva, investiga, comunica, expõe, interpreta, coleciona, pesquisa, partilha, reflexão e fruição e também em substantivos como património material e imaterial, educação, sociedade e comunidade. A informação museológica é o que permite que o museu cumpra sua função social, e é por meio da informação museológica que o museu se comunica com a sociedade.

#### **4. Metadados de representação e descrição da informação museológica**

Neste capítulo analisaremos os padrões de metadados desenvolvidos por Silva (2020) em sua tese de doutorado: Esquema de metadados para descrição de obras de arte em museus brasileiros: uma proposta, Object ID (1997) desenvolvido pelo Getty Information Institute e o Linked Art (2021), padrão de metadados desenvolvido pelo Grupo de Trabalho do CIDOC-ICOM Linked Art, formado por Robert Sanderson, Athanasios Velios e Emmanuelle Delmas-Glass.

O Object ID é um padrão de documentação para descrever coleções de itens arqueológicos, culturais e artísticos (OBJECT ID, 1997). Foi criado com o objetivo de identificar o património cultural para evitar roubos e contrabandos e, por isso, é promovido por diversas agências governamentais como FBI, *Scotland Yard*, *Interpol*, e organizações como *World*

*Customs Organization* (WCO) e *The United Nations Educational, Scientific and Cultural Organization* (UNESCO) (CONSELHO INTERNACIONAL DE MUSEUS, 1999).

O Object ID define 9 (nove) categorias de informações e 4 (quatro) passos para a documentação. As categorias são:

1. **Type of object** - qual o tipo do objeto? (pintura, gravura, escultura, etc);
2. **Materials and techniques** - de que materiais o objeto é feito? Como ele foi feito?
3. **Measurement** - qual o tamanho e/ou o peso do objeto?
4. **Inscriptions and markings** - existem marcas de identificação, números ou inscrições no objeto?
5. **Distinguishing features** - alguma característica física especial que ajude a identificar? (danos, reparos, defeitos de fabricação, etc);
6. **Title** - o objeto tem um título pelo qual é conhecido?
7. **Subject** - o que está sendo representado no objeto?
8. **Date or period** - quando o objeto foi feito?
9. **Maker** - quem produziu o objeto?

Os quatro passos do Object ID para a documentação são:

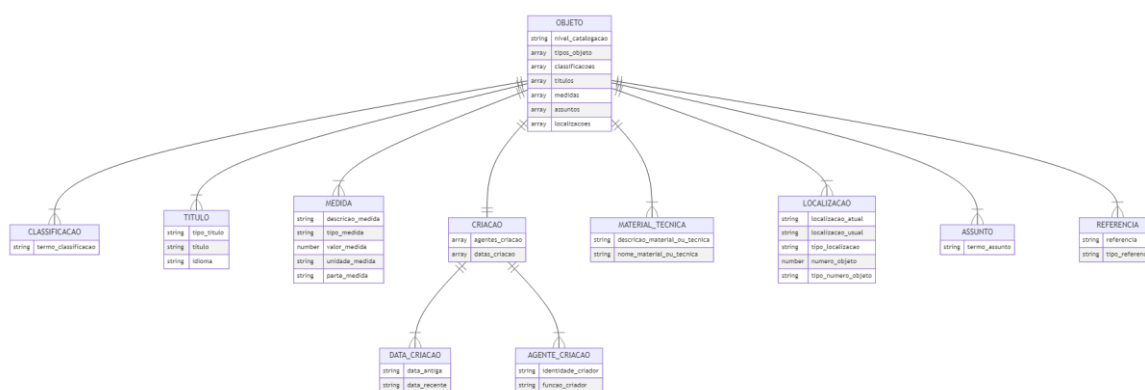
1. Tirar fotografias do objeto;
2. Identificar as 9 categorias de informações acima;
3. Escrever uma descrição, incluindo informações adicionais; e
4. Armazenar a documentação em local seguro.

Na reunião do Grupo de Trabalho - Documentando Museu Iberoamericano (DOMINO) do CIDOC-ICOM realizada no dia 22 de maio de 2022 (DOMINO, 2022), como parte das atividades da Conferência CIDOC Tallinn 2021, foi defendido pelos integrantes que o Object ID seja o padrão mínimo adotado por um museu para sua documentação, devido a sua simplicidade e reconhecimento por autoridades mundiais. O DOMINO tem como objetivo disseminar o trabalho do CIDOC para as comunidades dos idiomas português e espanhol, e servir como uma ponte linguística para troca de conhecimento especializado (INTERNATIONAL COMMITTEE FOR DOCUMENTATION, 2018).



O “Esquema de metadados para descrição de obras de arte em museus brasileiros: uma proposta” (SILVA, 2020), utiliza como referência os conjunto de metadados do *Categories for the Description of Works of Art* (CDWA) mantido pelo *Getty Vocabulary Program* (GVP), nas Categorias de Informação do Comitê Internacional de Documentação (CIDOC-ICOM) e nos grupos de informações da norma Spectrum. A partir daí, Silva apresenta 9 grupos de informações e 25 unidades de informação para descrição e representação de obras de arte, considerados mínimos pela autora, para os museus brasileiros. Na Figura 1 vemos os 9 grupos de informações na linha central e no OBJETO localizado acima de todos—e por onde todos os outros grupos de informações convergem—, os dois elementos de baixo são as subunidades de informação da unidade de informação CRIAÇÃO, os grupos de informações listados no centro são: CLASSIFICACAO, TITULO, MEDIDA, CRIACAO, MATERIAL\_TECNICA, LOCALIZACAO, ASSUNTO E REFERENCIA. Cada grupo de informação possui um conjunto de unidades de informações, à esquerda de cada grupo está listado o tipo de informação da unidade: string para texto, number para números, e array para listas. As conexões com dois traços cruzados ( || ) são referentes a presença obrigatória de ao menos um registro, as com um traço e uma chave ( |{ ) representam uma ou mais entradas.

**Figura 1: Esquema de metadados para descrição de obras de arte em museus brasileiros**



**Fonte:** Elaborado pelo autor, com base em SILVA, Camila Aparecida. **Esquema de metadados para descrição de obras de arte em museus brasileiros: uma proposta.**

2020. Programa de Pós-Graduação em Ciência da Informação - Escola de Comunicações e Artes / Universidade de São Paulo, São Paulo, 2020.

O Spectrum também possui um conjunto de metadados denominados de Grupos e unidades de informação. Os grupos são divididos em três grupos principais, depois divididos em mais subgrupos e por fim nas unidades de informação. Os grupos principais são: Grupos de informação de objeto, Grupos de informação procedimental e Grupos de informação de gestão de registros.

#### 1. Grupos de informação de objeto (*Object information groups*)

Consistem em informações sobre o objeto, e são divididas em 18 subgrupos de informações diferentes, totalizando 246 metadados:

**Quadro 1: Grupos de informação de objeto (*Object information groups*)**

Nome	Metadados
<b>Informação de identificação do objeto</b> ( <i>Object Identification Information</i> )	Object number, Other number, Other number type, Number of objects, Object name, Object name currency, Object name level, Object name note, Object name system, Object name type, Object name/title language, Title, Object name/title language, Title translation, Title type, Brief description, Comments, Distinguishing features, Responsible department/section
<b>Informação de descrição do objeto</b> ( <i>Object Description Information</i> )	Age, Age qualifier, Age unit, Color, Content - activity, Content - concept, Content - date, Content - description, Content - event name, Content - event name type, Content - language, Content - note, Content - object, Content - object type, Content - organization (Org), Content - other, Content - other type, Content - people (Peo), Content - person (Per), Content - place (Pla), Content - position, Content - script, Copy number, Dimension, Dimension measured part, Dimension value, Dimension value date, Dimension measurement unit, Dimension value qualifier, Edition number, Form, Inscription content, Inscríber (Org, Peo, Per), Inscription date, Inscription description, Inscription interpretation, Inscription language, Inscription method, Inscription position, Inscription script, Inscription translation, Inscription transliteration, Inscription type, Material, Material component, Material component note, Material name, Material source, Object status, Phase, Physical description, Sex, Style, Technical attribute, Technical attribute measurement, Technical attribute measurement unit, Object component name, Object component information
<b>Informação de localização do objeto</b> ( <i>Object Location Information</i> )	Current location, Location date, Current location fitness, Current location note, Normal location, Previous location, Previous location begin date, Previous location end date

Nome	Metadados
<b>Informação de exigências do objeto</b> ( <i>Object Requirement Information</i> )	Display recommendations, Environmental recommendations, Handling recommendations, Packing recommendations, Security recommendations, Special requirements, Storage recommendations, Salvage priority code, Salvage priority code date, Legal/license requirements, Legal/license requirements held, Legal/license requirements held begin date, Legal/license requirements held end date, Legal/license requirements held number, Legal/license requirements held renewal date
<b>Informação de produção do objeto</b> ( <i>Object Production Information</i> )	Object production date, Object production note, Object production organization (Org), Object production people (Peo), Object production person (Per), Object production place (Pla), Object production reason, Technique, Technique type
<b>Informação sobre o histórico e associação do objeto</b> ( <i>Object History and Association Information</i> )	Associated activity, Associated activity note, Associated concept, Associated cultural affinity, Associated date, Associated event date, Associated event name, Associated event name type, Associated event organization (Org), Associated event people (Peo), Associated event person (Per), Associated event place (Pla), Associated object, Associated object type, Associated organization (Org), Associated people (Peo), Associated person (Per), Associated place (Pla), Association note, Association type, Object history note, Owner (Org, Per), Ownership access, Ownership category, Ownership dates, Ownership exchange method, Ownership exchange note, Ownership exchange price, Ownership place (Pla), Related object number, Related object association, Related object note, Usage, Usage note
<b>Informação sobre a coleta do objeto</b> ( <i>Object Collection Information</i> )	Field collection event reference number, Field collection number, Field collection date, Field collection event name, Field collection method, Field collection note, Field collection place (Pla), Field collection source (Org, Peo, Per), Field collector (Org, Per), Geological complex name, Habitat, Habitat note, Stratigraphic unit name, Stratigraphic unit type, Stratigraphic unit note
<b>Informação sobre os direitos do objeto</b> ( <i>Rights Information</i> )	Right reference number, Right type, Right holder (Org, Peo, Per), Right holder's contact (Org, Per), Right begin date, Right end date, Right note
<b>Informação sobre os direitos de entrada do objeto</b> ( <i>Rights in Information</i> )	Rights in reference number, Rights in type, Rights in consent status, Rights in consent status date, Rights in begin date, Rights in end date, Rights in note
<b>Informação sobre os direitos de saída do objeto</b> ( <i>Rights out Information</i> )	Rights out reference number, Rights out type, Rights out requester, Rights out consent status, Rights out consent status date, Rights out authorizer, Rights out authorization date, Rights out begin date, Rights out end date, Rights out notes
<b>Informação sobre as condições e avaliação técnica do objeto</b> ( <i>Object condition and technical assessment information</i> )	Completeness, Completeness date, Completeness note, Condition, Condition date, Condition note, Conservation treatment priority, Environmental condition note, Environmental condition note date, Next condition check/assessment date, Technical assessment, Technical assessment date, Hazard, Hazard date, Hazards note

Nome	Metadados
<b>Informação sobre as conservação e tratamento do objeto</b> ( <i>Object conservation and treatment information</i> )	Treatment report, Treatment begin date, Treatment end date, Conservation material, Conservation note
<b>Informação de contribuição do proprietário do objeto</b> ( <i>Object owner contribution information</i> )	Owner's personal experience, Owner's personal response, Owner's reference, Owner's contribution, note
<b>Informação de uso da informação</b> ( <i>Use information</i> )	Credit line, Use status, Use status date, Text reference number, Text, Text audience, Text author, Text date, Text language, Text note, Text reason, Use measurement type, Use measurement value, Use measurement value unit, Use measurement value qualifier, Use measurement date, Use note, Use restriction, Use restriction date, Use restriction note
<b>Informação de valoração do objeto</b> ( <i>Object valuation information</i> )	Object valuation, Object valuation date, Object valuation type, Object valuation note, Object valuation reason, Object valuation renewal date
<b>Informação de contribuição do usuário do objeto</b> ( <i>Object user contribution information</i> )	User's reference, User's role, User's personal experience, User's personal response, User's contribution note
<b>Informação de revisão de coleção de objetos</b> ( <i>Object collection review information</i> )	Collections review criterion, Collections review result, Collections review result date, Collections review action
<b>Informação sobre auditoria</b> ( <i>Object Audit information</i> )	Object audit category, Object audit date, Object audit information unit, Object audit note, Object audit result

**Fonte:** Elaborado pelo autor baseado em *Object information groups* do Spectrum (2017).

## 2. Grupos de informação procedimental (*Procedural information groups*)

Consistem nas informações relacionadas aos procedimentos Spectrum, como ações de conservação, empréstimos, entre outros. São divididos em 27 subgrupos de informações diferentes, totalizando 309 metadados:

### Quadro 2: Grupos de informações procedimentais (*Procedural information groups*)

Nome	Descrição
<b>Informações de entrada de objeto</b> ( <i>Object entry information</i> )	Entry number, Entry date, Entry method, Entry reason, Entry manager, Current owner, (Org, Peo, Per), Depositor (Org, Per), Depositor's requirements, Entry note, Packing note, Return date
<b>Informações de aquisição</b>	Acquisition reference number, Accession date, Acquisition

Nome	Descrição
<b>de objeto</b> ( <i>Acquisition information</i> )	authorizer (Per), Acquisition authorization date, Acquisition date, Acquisition funding, Acquisition funding source (Org, Per), Acquisition funding source provisos, Acquisition method, Acquisition note, Acquisition provisos, Acquisition reason, Acquisition source (Org, Peo, Per), Group purchase price, Object offer price, Object purchaser offer price, Object purchase price, Original object purchase price, Original object purchase price denomination, Transfer of title number
<b>Informações de movimentação de objeto</b> ( <i>Movement information</i> )	Movement reference number, Movement authorizer (Per), Movement authorization date, Movement contact (Per), Movement method, Movement note, Movement reason, Planned removal date, Removal date, Shipper (Org, Per), Shipper's contact (Per), Shipping note, Organization courier (Per)
<b>Informações de saída de objeto</b> ( <i>Object exit information</i> )	Exit reference number, Delivery date, Exit authorizer (Per), Exit authorization date, Exit date, Exit destination (Org, Per), Exit method, Exit note, Exit reason, Expected return date, Expected return method
<b>Informações de entrada de empréstimos do objeto</b> ( <i>Loan in information</i> )	Loan in reference number, Loan in reason, Lender (Org, Peo, Per), Lender's authorizer (Org, Per), Lender's authorization date, Lender's contact, Loan in begin date, Loan in end date, Loan in contact (Org, Per), Loan in conditions, Loan in note, Loan in status, Loan in status date
<b>Informações de saída de empréstimos do objeto</b> ( <i>Loan out information</i> )	Loan out reference number, Loan out authorizer (Per), Loan out authorization date, Borrower (Org, Per), Borrower's contact (Org, Per), Borrower's status, Special conditions of loan, Loan out begin date, Loan out end date, Loan out reason, Loan out note, Loaned object status, Loaned object status date
<b>Informações sobre as verificações técnicas de condições</b> ( <i>Conditional check technical assessment information</i> )	Condition check/assessment reference number, Condition checker/assessor (Per), Condition check/assessment reason, Condition check/assessment date, Condition check/assessment method, Condition check/assessment note
<b>Informações sobre conservação e tratamento</b> ( <i>Conservation and treatment information</i> )	Conservation reference number, Conservation authorizer (Per), Conservation authorization date, Conservation method, Conservator (Org, Per), Recall date
<b>Informação de valoração</b> ( <i>Valuation information</i> )	Valuation reference number, Valuation authorizer (Per), Valuation authorization date, Valuer (Org, Per)
<b>Informação de seguro</b> ( <i>Insurance information</i> )	Insurance reference number, Insurer (Org, Per), Insurance policy number, Insurance amount, Insurance authorizer (Per), Insurance authorization date, Insurance begin, date, Insurance end date, Insurance note, Insurance renewal date
<b>Informação de indenização</b> ( <i>Indemnity information</i> )	Indemnity reference number, Indemnifier (Org), Indemnity authorizer (Per), Indemnity authorization date, Indemnity request date, Indemnity confirmation date, Indemnity note, Indemnity begin date, Indemnity end date, Indemnity renewal date, Object minimum liability sum

Nome	Descrição
<b>Informação de perdas e danos</b> ( <i>Damage/loss information</i> )	Damage/loss reference number, Damage/loss date, Damage/loss reporter (Per), Damage/loss method, Damage/loss note
<b>Informação de descarte</b> ( <i>Disposal information</i> )	Disposal reference number, Disposal reason, Disposal authorizer (Per), Disposal authorization date, Disposal method, Deaccession date, Disposal date, Disposal new object number, Disposal note, Disposal price, Disposal provisos, Disposal recipient (Org, Peo, Per), Group disposal price
<b>Informação de uso das coleções</b> ( <i>Use of collections information</i> )	Use reference number, Use request note, Use type, Use authorizer (Per), Use authorization date, Use title, Use begin date, Use end date, User (Org, Per), User's contact (Per), Use organizer (Org, Per), Use venue (Pla), Use provisos, Use result
<b>Informação de auditoria</b> ( <i>Audit information</i> )	Audit reference number, Audit method, Audit authorizer (Per), Audit authorization date, Audit type, Auditor (Org, Per)
<b>Informação de revisão de coleção</b> ( <i>Collection review information</i> )	Collections review reference number, Collections review authorizer, Collections review authorization date, Collections review title, Collections review reason, Collections review type, Collections review method, Collections review manager, Collections review note, Collections review status, Collections review status date, Collections review begin date, Collections review end date
<b>Unidades procedimentais comuns</b> ( <i>Common procedural units</i> )	Authorizer (Per), Authorization date, Cost, Cost note, Procedure begin date, Procedure end date, Procedure manager, Procedure note, Procedure title, Reason, Request, Request date, Requester (Org, Per), Requested action date, Scheduled begin date, Scheduled end date, Status, Status date, Status review date
<b>Informação de processo</b> ( <i>Process information</i> )	Process reference number, Process date, Process method, Process note, Process person/organization (Per, Org)
<b>Informação de reprodução</b> ( <i>Reproduction information</i> )	Reproduction number, Current reproduction location, Reproduction creator (Org, Per), Reproduction date, Reproduction description, Reproduction format, Reproduction status, Reproduction type, Reproduction reason, Reproduction requester (Org, Per)
<b>Informações de endereço</b> ( <i>Address information</i> )	Address - text, Address - place, Address - postcode, Address - e-mail, Address - fax number, Address - telephone number, Address type
<b>Informações de data</b> ( <i>Date information</i> )	Date - association, Date - earliest/single, Date - earliest/single certainty, Date - earliest/single qualifier, Date - latest, Date - latest certainty, Date - latest qualifier, Date - period, Date text
<b>Informações de localização</b> ( <i>Location information</i> )	Location reference name/number, Location type, Location address, Location access note, Location condition note, Location condition note date, Location security note, Environment measurement type, Environment measurement value, Environment measurement value unit, Environment measurement value qualifier, Environment measurement date
<b>Informações sobre organização</b> ( <i>Organization</i> )	Organization's reference number, Organization's address, Organization's additions to name, Organization's contact name,

Nome	Descrição
<i>information)</i>	Organization's foundation date, Organization's foundation place, Organization's dissolution date, Organization's function, Organization's group, Organization's history, Organization's MDA code, Organization's main body, Organization's sub-body, Organization's association
<b>Informações sobre pessoas</b> ( <i>People information</i> )	People's culture, People's association, People's group, People's linguistic group
<b>Informações sobre pessoa</b> ( <i>Person information</i> )	Person's reference number, Person's forenames, Person's surname, Person's title, Person's additions to name, Person's initials, Person's salutation, Person's address, Person's birth date, Person's place of birth, Person's death date, Person's place of death, Person's biographical note, Person's gender, Person's association, Person's group, Person's name notes, Person's nationality, Person's occupation, Person's school/style
<b>Informações sobre local</b> ( <i>Place information</i> )	Place reference number, Place reference number type, Place name, Place name type, Place association, Place context, Place context date, Place context level, Place coordinates, Place coordinates qualifier, Place coordinates type, Place environmental details, Place feature, Place feature date, Place feature type, Place note, Place owner, Place position, Place status, Place type system
<b>Informações sobre referências</b> ( <i>Reference information</i> )	Place reference number, Place reference number type, Place name, Place name type, Place association, Place context, Place context date, Place context level, Place coordinates, Place coordinates qualifier, Place coordinates type, Place, environmental details, Place feature, Place feature date, Place feature type, Place note, Place owner, Place position, Place status, Place type system

**Fonte:** Elaborado pelo autor, com base em *Object procedural groups* do Spectrum (2017).

### 3. Grupos de informação de gestão de registros (*Record management information groups*)

São unidades de informação para realizar anotações no sistema de documentação, como notas de administração, entre outros. São divididos em 3 subgrupos de informações diferentes, totalizando 11 metadados:

#### **Quadro 3: Grupos de Informações de Gerenciamento de Registros (*Record Management information groups*)**

Nome	Descrição
<b>Histórico de alterações</b> ( <i>Amendment history</i> )	Unit of information added, Amendment history authorizer, Unit of information scheme, Information source, Information source date, Recorder, Recording date, Recording progress
<b>Informações sobre o registro</b> ( <i>Record information</i> )	Record type

Nome	Descrição
<b>Uso e provisionamento da informação</b> ( <i>Use and provision of information</i> )	Confidentiality note, Information usage

**Fonte:** Elaborado pelo autor, com base em *Record Management information groups* do Spectrum (2017).

Já o Linked Art é uma comunidade formada por representantes de instituições ao redor do mundo como The Canadian Heritage Information Network (CHIN), J. Paul Getty Trust, The Frick Collection, Europeana, Louvre, Rijksmuseum, The Victoria and Albert Museum, entre outras, com o objetivo de criar um modelo compartilhado baseado em *Linked Open Data* para descrever Arte (LINKED ART, 2021a, 2021b). O projeto é coordenado por um quadro editorial em que Robert Sanderson (Yale University) e Emmanuelle Delmas-Glass (Yale Center for British Art) compartilham o posto de co-presidentes.

A comunidade Linked Art parte do conceito da usabilidade para a audiência correta, em que o maior público interessado em “dados” seriam os desenvolvedores que poderiam criar interfaces para o público final. Desdobrando esta ideia em cinco princípios de design, temos:

1. **Abstração para a audiência correta:** desenvolvedores não precisam do mesmo acesso aos dados como os ontologistas;
2. **Poucas barreiras de entrada:** deve ser fácil começar a trabalhar com dados e construir algo: *“If it takes a long time to understand the model, ontology, sparql query syntax and so forth, then developers will look for easier targets”* (SANDERSON, 2018);
3. **Compreensível pela introspecção:** os dados devem ser compreensíveis pela leitura humana, usar JSON-LD é utilizar a linguagem que o desenvolvedor já compreende;
4. **Documentação com exemplos funcionais:** você nunca poderá intuir todas as regras dos dados, documentação serve para mapear os padrões que o desenvolvedor pode encontrar; e
5. **Poucas exceções em vez de vários padrões consistentes:** cada exceção em uma API é outra regra que o desenvolvedor precisa apreender: *“every exception is jarring, and requires additional code to manage. While not everything is*



*homogenous, a set of patterns that manage exceptions well is better than many custom fields.” (SANDERSON, 2018).*

O Linked Art está sob desenvolvimento ativo e a versão atual é a 0.8.0 e é considerada instável, passível de mudanças. A versão 1.0.0 estava prevista para o final de 2021, mas foi atrasada por conta da pandemia de COVID-19 (LINKED ART, 2021).

O modelo Linked Art tem como foco principal a descrição de recursos do patrimônio cultural artístico e atividades de museus. Para isto, o modelo utiliza um perfil do CIDOC *Conceptual Reference Model* (CIDOC-CRM) que combina a facilidade de uso do JSON-LD com a excelência do CRM, mas reduzindo sua complexidade para um conjunto de classes e propriedades que são mais comuns em 90% dos casos de usos de 90% das organizações (LINKED ART, 2021)

*JavaScript Object Notation* (JSON) é um formato aberto de arquivo para intercâmbio de informações (W3C JSON-LD WORKING GROUP, 2014). Consiste em arquivos de fácil leitura por humanos e máquinas com a extensão .json. Mesmo que tenha se originado na sintaxe de objetos JavaScript, por isso seu nome, pode ser utilizado por diversos ambientes e linguagens de programação diferentes devido sua sintaxe simples (MDN WEB DOCS, 2023).

Um arquivo JSON pode ser expressado da seguinte forma:

```
{  
  "title": "One and three chairs",  
  "author": "Joseph Kosuth"  
}
```

Mas o que este JSON representa para alguém que não conhece o contexto artístico? O que é “title” e “author” para uma máquina, e quais tipos de valores eles podem receber?

*JSON for Linked Data* (JSON-LD) busca solucionar este problema adicionando uma camada de contexto ao JSON, ao fazer um link com a definição do vocabulário utilizado (SPORNY, 2012). Além disso é possível adicionar um identificador ao objeto, desta forma, o mesmo arquivo, porém agora com a definição do vocabulário e utilizando o contexto do Linked Art, pode ser expresso da seguinte forma:

```

{
  "@context": "https://linked.art/ns/v1/linked-art.json",
  "id": "https://www.moma.org/collection/works/81435",
  "_label": "One and three chairs",
  "type": "HumanMadeObject",
  "identified_by": [
    {
      "type": "Identifier",
      "_label": "One and three chairs",
      "classified_as": [
        {
          "id": "http://vocab.getty.edu/aat/300417201",
          "type": "Type",
          "_label": "Title"
        }
      ],
      "content": "One and three chairs"
    }
  ],
  "produced_by": {
    "type": "Production",
    "carried_out_by": [
      {
        "type": "Person",
        "id": "https://www.moma.org/artists/3228",
        "_label": "Joseph Kosuth"
      }
    ]
  }
}

```

Parece mais complexo que o exemplo anterior, e de fato é, mas agora temos um arquivo JSON que pode ser lido por humanos e máquinas, e que pode ser utilizado para conectar dados, trazendo muito mais contexto para a informação, em que: “@context” é a URI do vocabulário utilizado, no caso Linked Art; “id” é o identificador do objeto, a URI para o registro no MoMA; “\_label” é um rótulo para leitura pelo desenvolvedor; “type” é o tipo de entidade; “identified\_by” é a propriedade para identificar a obra, recebe um array de objetos que podem ser tanto “Name” quanto “Identifier”, no caso é um “Identifier” com o valor “One and three chairs” classificado como “Title” e referenciado à definição de “title” no vocabulário AAT do Getty. Por fim, “produced\_by” é a propriedade que recebe um objeto “Production” que tem como propriedade “carried\_out\_by” que recebe um array de objetos “Person” com a propriedade “id” e a URI para o registro do artista no MoMA e “\_label” com o nome do artista.

O JSON-LD fornece uma maneira de os dados JSON serem interoperáveis na escala da Web (JSON-LD WORKING GROUP, 2020). E, tem como principal intenção “ser uma maneira de usar Linked Data em ambientes de programação baseados na Web, para construir serviços Web interoperáveis e para armazenar Linked Data em mecanismos de armazenamento baseados em JSON” (JSON-LD WORKING GROUP, 2020), ou seja, é o formato ideal para serviços REST.

REST é uma arquitetura para distribuição de sistemas hipermídia (FIELDING, 2000), que consiste em adicionar camadas de restrições à aplicações:

There are two common perspectives on the process of architectural design, whether it be for buildings or for software. The first is that a designer starts with nothing—a blank slate, whiteboard, or drawing board—and builds-up an architecture from familiar components until it satisfies the needs of the intended system. The second is that a designer starts with the system needs as a whole, without constraints, and then incrementally identifies and applies constraints to elements of the system in order to differentiate the design space and allow the forces that influence system behavior to flow naturally, in harmony with the system. Where the first emphasizes creativity and unbounded vision, the second emphasizes restraint and understanding of the system context. REST has been developed using the latter process. (FIELDING, 2000)

As camadas de restrições são (FIELDING, 2000):

1. Separação Cliente-servidor: ao separar a lógica dos dados da interface do usuário, melhoramos a portabilidade da interface de usuário em várias plataformas (computador, celular, tablet) e melhoramos a escalabilidade ao simplificar os componentes do servidor;
2. Stateless: a comunicação do cliente com o servidor tem que ser sem estado por natureza. Isso significa que cada requisição feita do cliente para o servidor deverá conter toda a informação necessária para a requisição ser compreendida;
3. Cache: como a comunicação é sem estado, isso pode acarretar em ineficiência da rede, dessa forma adicionamos o controle do cache para reutilização de informação requisitada anteriormente;
4. Interface uniforme: a característica principal que distingue a arquitetura REST de outra arquitetura de rede é a interface uniforme entre componentes. As

implementações são dissociadas dos serviços que fornecem, o que incentiva a evolução independente; e

5. Sistema em camadas: ao compor a arquitetura em camadas hierárquicas, restringimos os componentes a não irem além de seus escopos.

A API do Linked Art é dividida em duas partes: *Shared Constructs*, ou Estruturas Compartilhadas (EC), e *Entity Endpoints*, Endpoints de Entidades (EE). Como as propriedades definidas no Linked Art se repetem pelas EC e EE, separamos as propriedades em dois quadros distintos para facilitar a visualização. No primeiro, apresentamos as propriedades que são obrigatórias nas EE e, com exceção de “@context” e “id”, também são obrigatórias em todas as EC. No segundo, apresentamos as propriedades que podem ou não serem utilizadas em mais de uma EC ou EE. Após os quadros, apresentamos as EC e EE. Propriedades exclusivas de cada EC ou EE serão apresentadas em seus respectivos quadros.

#### Quadro 4: Propriedades comuns e obrigatórias do Linked Art.

Nome	Tipo de dado	Exigência	Descrição
@context	string	obrigatório	DEVE ser a URI do contexto Linked Art: “https://linked.art/ns/v1/linked-art.json”
id	string	obrigatório	DEVE ser uma URI que define o <i>Shared Construct</i> ou <i>Entity Endpoint</i>
type	string	obrigatório	O tipo da classe
_label	string	obrigatório	Um rótulo legível por humanos, para desenvolvedores
classified_as	array	opcional	Uma ou mais classificações do recurso e DEVE seguir os requisitos para <i>Type</i>
identified_by	array	recomendado	Um ou mais identificadores do recurso e DEVE seguir os requisitos para <i>Name</i> ou <i>Identifier</i>
referred_to_by	array	opcional	Uma ou mais referências a um <i>TextualWork</i> que referencia o recurso, ou uma declaração incorporada sobre o recurso, e DEVE seguir os requisitos para <i>Reference</i> ou <i>Statement</i>
equivalent	array	opcional	Um array de objetos em que cada um é uma referência a uma identidade e descrição externa do recurso

**Fonte:** Elaborado pelo autor. Propriedades comuns do Linked Art (2021).

#### Quadro 5: Propriedades do Linked Art.

Nome	Tipo de dado	Descrição
representation	array	Uma ou mais referências a um <i>VisualWork</i> que retrata o recurso
member_of	array	Uma ou mais referências a um <i>Set</i> que este recurso é um membro
subject_of	array	Uma ou mais referências a um <i>Textual</i>
attributed_by	array	Array de <i>Assignments</i>
broader	array	Um array de referências a outro <i>Concept</i> que é mais amplo do que este
created_by	object	-
timespan	object	Um objeto JSON que registra quando a <i>Creation</i> ocorreu, e deve seguir o formato de <i>Timespan</i>
influenced_by	array	Um array de objetos JSON, em que cada um deles é uma referência a uma entidade que a influenciou
took_place_at	array	Uma ou mais referências a um <i>Place</i> que o evento ocorreu
caused_by	array	Uma ou mais referências a um <i>Event</i> que causou este evento
carried_out_by	array	Uma ou mais referências a uma <i>Person</i> ou <i>Group</i> que realizou esta atividade
technique	array	Uma ou mais referências a um <i>Concept</i> que define a técnica usada na atividade
dimension	array	Um array de objetos JSON, em que cada um deles é um <i>Dimension</i>
part_of	array	Um array de objetos JSON, em que cada um deles é uma referência a outra entidade que este recurso é parte
format	string	O tipo de mídia do recurso, por exemplo “image/jpeg”
conforms_to	string	Um array de objetos JSON, em que cada um deles é uma referência externa a que este recurso está em conformidade, o tipo da referência deve ser <i>InformationObject</i>
digitally_carries	array	Um array de objetos JSON, em que cada um deles é uma referência a um <i>TextualWork</i> que este recurso digitalmente carrega
digitally_shows	array	Um array de objetos JSON, em que cada um deles é uma referência a um <i>VisualWork</i> que este recurso digitalmente mostra
access_point	array	Um array de objetos JSON, em que cada um deles é uma referência a uma URI em que esta entidade pode ser acessada
used_specific_object	array	Um array de objetos JSON, em que cada um deles é uma referência a outra entidade que foi usada nesta atividade

**Fonte:** Elaborado pelo autor. Propriedades do Linked Art (2021).

As EC são estruturas de dados definidas no Linked Art que são utilizadas por mais de um endpoint. São objetos JSON que por sua vez utilizam as propriedades definidas nos quadros anteriores, além de apresentar algumas novas propriedades exclusivas. São elas (LINKED ART, 2021):

**Quadro 6: Propriedades de *Dimension*.**

Nome	Exigência	Descrição
id	opcional	-
type	obrigatório	DEVE ser <i>Dimension</i>
_label	recomendado	-
value	obrigatório	O valor numérico da dimensão
unit	obrigatório	A unidade para a dimensão, que DEVE seguir os requisitos de <i>MeasurementUnit</i> , que é uma extensão de <i>Type</i>
classified_as	recomendado	-
identified_by	recomendado	Somente <i>Name</i>
upper_value_limit	opcional	Um número que representa o valor máximo da dimensão
lower_value_limit	opcional	Um número que representa o valor mínimo da dimensão
referred_to_by	opcional	-
assigned_by	opcional	Um array de objetos <i>Measurement</i> descrito a seguir

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *Dimensions* do Linked Art (2021).

**Quadro 7: Propriedades de *Measurement*.**

Nome	Exigência	Descrição
id	opcional	-
type	obrigatório	DEVE ser <i>AttributeAssignment</i>
_label	recomendado	-
classified_as	recomendado	-
carried_out_by	recomendado	-
timespan	opcional	-
referred_to_by	opcional	-

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *Measurements* do Linked Art (2021).

- *Dimension* - dimensões de um recurso físico ou digital, com uma unidade e um tipo de dimensão;

**Quadro 8: Propriedades de *Identifier*.**

Nome	Exigência	Descrição
id	opcional	-
type	obrigatório	DEVE ser <i>Identifier</i>
_label	recomendado	-
content	obrigatório	O conteúdo <i>string</i> do identificador
classified_as	recomendado	-
referred_to_by	opcional	-

Nome	Exigência	Descrição
assigned_by	opcional	Um <i>array</i> de objetos JSON, cada um dos quais DEVEM seguir os requisitos para <i>IdentifierAssignment</i>

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *Identifier* do Linked Art (2021).

#### **Quadro 9: Propriedades de *Identifiers Assignments*.**

Nome	Exigência	Descrição
id	opcional	-
type	obrigatório	DEVE ser <i>AttributeAssignment</i>
_label	recomendado	-
classified_as	recomendado	-
carried_out_by	recomendado	-
timespan	opcional	-
referred_to_by	opcional	-
used_specific_object	opcional	Um <i>array</i> de objetos JSON, cada um dos quais é uma referência para outro <i>Endpoint</i> que foi instrumental na atribuição do identificador

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *Identifiers Assignments* do Linked Art (2021).

- *Identifier* - identificadores de um recurso, como um número de catálogo ou um número de inventário;

#### **Quadro 10: Propriedades de *MonetaryAmount*.**

Nome	Exigência	Descrição
id	opcional	-
type	obrigatório	DEVE ser <i>MonetaryAmount</i>
_label	recomendado	-
value	obrigatório	O valor numérico
currency	obrigatório	O código da moeda, e DEVE ser um objeto <i>Currency</i>
classified_as	recomendado	-
identified_by	recomendado	Somente <i>Name</i>
upper_value_limit	opcional	Um número que representa o valor máximo da quantia monetária
lower_value_limit	opcional	Um número que representa o valor mínimo da quantia monetária
referred_to_by	opcional	-

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *MonetaryAmount* do Linked Art (2021).

- *Monetary Amount* - similares a *Dimensions*, mas para valores monetários, no caso a propriedade *Type* é definida como *MonetaryAmount*, os objetos *MonetaryAmounts* são usados somente nas Atividades de Proveniência;

#### Quadro 11: Propriedades de *Name*.

Nome	Exigência	Descrição
id	opcional	-
type	obrigatório	DEVE ser <i>Name</i>
_label	recomendado	-
content	obrigatório	O conteúdo <i>string</i> do nome
classified_as	recomendado	-
language	recomendado	Um array de objetos <i>Language</i>
part	opcional	Um array de objetos <i>Name</i> que são partes deste nome.
referred_to_by	opcional	-

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *Name* do Linked Art (2021).

- *Name* - Nomes são rótulos linguísticos para uma entidade. Eles podem ser nomes de pessoas, títulos de obras, nomes de lugares, etc.;

#### Quadro 12: Propriedades de *Statement*.

Nome	Exigência	Descrição
id	opcional	-
type	obrigatório	DEVE ser <i>LinguisticObject</i>
_label	recomendado	-
classified_as	recomendado	-
language	recomendado	Um array de objetos <i>_Language</i>
referred_to_by	opcional	-

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *Statement* do Linked Art (2021).

- *Statement* - são expressões do conteúdo ou nota sobre a entidade que esta sendo descrita.

#### Quadro 13: Propriedades de *TimeSpan*.

Nome	Exigência	Descrição
id	opcional	-
type	obrigatório	DEVE ser <i>TimeSpan</i>
_label	recomendado	-



Nome	Exigência	Descrição
value	obrigatório	O valor numérico
classified_as	recomendado	-
identified_by	recomendado	Somente <i>Name</i>
begin_of_the_begin	recomendado	Uma data e hora ISO 8601 representando a data mais antiga que este intervalo de tempo pode ter começado
end_of_the_end	recomendado	Uma data e hora ISO 8601 representando a data mais recente que este intervalo de tempo pode ter terminado
end_of_the_begin	opcional	Uma data e hora ISO 8601 representando a data mais recente que este intervalo de tempo pode ter começado
begin_of_the_end	opcional	Uma data e hora ISO 8601 representando a data mais antiga que este intervalo de tempo pode ter terminado
duration	opcional	Um objeto <i>Dimension</i> que representa a duração do intervalo de tempo

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *TimeSpan* do Linked Art (2021).

- *TimeSpan* - são intervalos de tempos utilizados para descrever a duração de um evento ou atividade;

#### Quadro 14: Propriedades de *Type/Currency/Language/Material/MeasurementUnit*.

Nome	Exigência	Descrição
id	opcional	-
type	obrigatório	DEVE ser <i>Type</i> , <i>Currency</i> , <i>Language</i> , <i>Material</i> ou <i>MeasurementUnit</i>
_label	recomendado	-
classified_as	recomendado	-

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *Type/Currency/Language/Material/MeasurementUnit* do Linked Art (2021).

- *Type/Concept* - conceitos, como tipos, são partes principais do modelo e estão presentes em praticamente todos os recursos descritos (LINKED ART, ). Eles dão acesso a vocabulários controlados externos como o AAT do Getty e outros:

#### Quadro 15: Propriedades de *AttributeAssignment*.

Nome	Exigência	Descrição
id	opcional	-
type	obrigatório	DEVE ser <i>AttributeAssignment</i>
_label	recomendado	-
content	obrigatório	O conteúdo <i>string</i> do nome

Nome	Exigência	Descrição
classified_as	recomendado	-
referred_to_by	opcional	-
carried_out_by	opcional	-
timespan	opcional	-
assigned	obrigatório	Um array de objetos json que sejam referências a outra entidade relacionada a esta
assigned_property	opcional	Uma URI para a relação entre a entidade principal e a entidade referenciada em <i>assigned</i>

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *AttributeAssignment* do Linked Art (2021).

- *AttributeAssignment* - é utilizado para relações entre entidades não declaradas, como relacionamentos interpessoais que são muito complexas para serem descritos semanticamente, ou apenas um conjunto arbitrário de recomendações para outras entidades (LINKED ART, 2021).

#### Quadro 16: Propriedades de *Reference*.

Nome	Exigência	Descrição
id	obrigatório	DEVE ser a URI da entidade referenciada
type	obrigatório	DEVE ser do mesmo tipo da entidade referenciada
_label	recomendado	-

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *Reference* do Linked Art (2021).

- *Entity Reference* - como uma API hipermídia, o Linked Art utiliza referências para outras entidades, que podem ser tanto internas quanto externas, como o AAT do Getty, e também são utilizadas para descrever relações entre entidades, como a relação entre uma obra e seu autor, por exemplo. É uma das estruturas mais comuns no modelo, e sempre que falamos sobre uma referência estamos falando de um objeto JSON com essas propriedades.

Os EE são as entidades que podem ser descritas pelo Linked Art, e são divididas em 11 tipos. Cada tipo de entidade possui um conjunto de propriedades obrigatórias e opcionais, e descrevem os recursos do patrimônio cultural artístico (LINKED ART, 2021). Como as propriedades comuns do Linked Art, citadas no Quadro 13 se repetem em todas as EE, não serão apresentadas novamente nos quadros a seguir, e as outras propriedades serão apenas referenciadas.

1. *Concept* - tipos, materiais, técnicas, idiomas, entre outros que sejam registros completos, ao contrário de referências externas;

Os *Concepts* possuem as seguintes propriedades:

**Quadro 17: Propriedades de *Concept*.**

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo do <i>Concept</i> , deve ser <i>Type</i> , <i>Material</i> , <i>Language</i> , <i>Currency</i> ou <i>MeasurementUnit</i>
representation	array	opcional	-
member_of	array	opcional	-
subject_of	array	opcional	-
attributed_by	array	opcional	-
broader	array	opcional	-
created_by	object	opcional	Um objeto que descreve a criação do <i>Concept</i> , que deve seguir o formato do objeto <i>Creation</i> descrito a seguir.

**Fonte:** Elaborado pelo autor. Metadados utilizados na entidade *Concept* do Linked Art (2021).

**Quadro 18: Propriedades de *Creation*.**

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo da classe que deve ser <i>Creation</i>
timespan	array	opcional	-
influenced_by	array	opcional	-

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *Concept Creation* do Linked Art (2021).

2. *DigitalObject* - imagens, vídeos, áudios, documentos, webpages, ou outros recursos digitais;

Os *DigitalObjects* possuem as seguintes propriedades:

**Quadro 19: Propriedades de *Digital*.**

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>DigitalObject</i> .
member_of	array	opcional	-
attributed_by	array	opcional	-
dimension	array	opcional	-
part_of	array	opcional	-
format	array	opcional	-

Nome	Tipo	Exigência	Descrição
conforms_to	array	opcional	-
digitally_carries	array	opcional	-
digitally_shows	array	opcional	-
digitally_available_via	array	opcional	Um array de objetos JSON, em que cada um deles deve seguir o formato do <i>DigitalService</i> descrito a seguir
access_point	array	opcional	-
created_by	object	opcional	Um objeto JSON representando a criação do <i>Concept</i> , que deve seguir o formato do objeto <i>Creation</i> descrito a seguir
used_for	array	opcional	Um array de objetos JSON, em que cada um deles representa uma <i>PublicationActivity</i> , como descrita a seguir

**Fonte:** Elaborado pelo autor. Metadados utilizados na entidade *Digital* do Linked Art (2021).

#### **Quadro 20: Propriedades de *DigitalService*.**

Nome	Tipo	Exigência	Descrição
id	string	obrigatório	-
type	string	obrigatório	O tipo deve ser <i>DigitalService</i> .
access_point	array	opcional	-
conforms_to	array	opcional	-

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *DigitalService* do Linked Art (2021).

#### **Quadro 21: Propriedades de *Creation, Publication Activity*.**

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>Creation</i> ou <i>Activity</i> .
timespan	array	opcional	-
took_place_at	array	opcional	-
caused_by	array	opcional	-
influenced_by	array	opcional	-
carried_out_by	array	opcional	-
used_specific_object	array	opcional	-
technique	array	opcional	-
part	array	opcional	Um array de objetos JSON, em que cada um deles é uma outra instância do mesmo tipo que descreve aspectos diferentes da criação ou publicação

**Fonte:** Elaborado pelo autor. Metadados utilizados nos objetos *Creation, Publication Activity* do Linked Art (2021).

3. *Event* - eventos e atividades não específicas que estão relacionadas, mas não são parte de outra entidade (LINKED ART, 2021);

Os *Events* possuem as seguintes propriedades:

**Quadro 22: Propriedades de *Event*.**

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>Period</i> , <i>Event</i> ou <i>Activity</i> .
representation	array	opcional	-
member_of	array	opcional	-
subject_of	array	opcional	-
attributed_by	array	opcional	-
part_of	array	opcional	-
timespan	array	opcional	-
took_place_at	array	opcional	-
caused_by	array	opcional	-
influenced_by	array	opcional	-
carried_out_by	array	opcional	-
used_specific_object	array	opcional	-

**Fonte:** Elaborado pelo autor. Metadados utilizados na entidade *Event* do Linked Art (2021).

4. *Groups* - grupos de pessoas, organizações, ou outras entidades similares (LINKED ART, 2021);

Os *Groups* possuem as seguintes propriedades:

**Quadro 23: Propriedades de *Group*.**

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>Period</i> , <i>Event</i> ou <i>Activity</i> .
representation	array	opcional	-
member_of	array	opcional	-
subject_of	array	opcional	-
attributed_by	array	opcional	-
carried_out	array	opcional	-
contact_point	array	opcional	-
residence	array	opcional	-
formed_by	array	opcional	-
dissolved_by	array	opcional	-

**Fonte:** Elaborado pelo autor. Metadados utilizados na entidade *Group* do Linked Art (2021).

5. *People* - pessoas (LINKED ART, 2021);

*People* possuem as seguintes propriedades:

**Quadro 24: Propriedades de *People*.**

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>People</i> .
representation	array	opcional	-
member_of	array	opcional	-
subject_of	array	opcional	-
attributed_by	array	opcional	-
carried_out	array	opcional	-
contact_point	array	opcional	-
residence	array	opcional	-
born	array	opcional	-
died	array	opcional	-

**Fonte:** Elaborado pelo autor. Metadados utilizados na entidade *People* do Linked Art (2021).

6. *PhysicalObject* - objetos físicos, incluindo obras de arte, artefatos, edifícios, partes de objetos, entre outros (LINKED ART, 2021);

*PhysicalObjects* possuem as seguintes propriedades:

**Quadro 25: Propriedades de *Object*'.**

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>Period</i> , <i>Event</i> ou <i>Activity</i> .
representation	array	opcional	-
member_of	array	opcional	-
subject_of	array	opcional	-
attributed_by	array	opcional	-
part_of	array	opcional	-
dimension	array	opcional	-
made_of	array	opcional	-
current_owner	array	opcional	-
current_custodian	array	opcional	-
current_permanent_custodian	array	opcional	-
current_location	array	opcional	-

Nome	Tipo	Exigência	Descrição
current_permanent_location	array	opcional	-
carries	array	opcional	-
shows	array	opcional	-
produced_by	array	opcional	-
destroyed_by	array	opcional	-
removed_by	array	opcional	-

**Fonte:** Elaborado pelo autor. Metadados utilizados na entidade *Object* do Linked Art (2021).

## 7. *Place* - locais (LINKED ART, 2021);

*Places* possuem as seguintes propriedades:

### Quadro 26: Propriedades de *Place*.

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>Place</i> .
representation	array	opcional	-
member_of	array	opcional	-
subject_of	array	opcional	-
attributed_by	array	opcional	-
part_of	array	opcional	-
approximated_by	array	opcional	-
defined_by	array	opcional	-

**Fonte:** Elaborado pelo autor. Metadados utilizados na entidade *Place* do Linked Art (2021).

## 8. *ProvenanceActivity* - atividades de proveniência (LINKED ART, 2021);

*ProvenanceActivities* possuem as seguintes propriedades:

### Quadro 27: Propriedades de *Provenance*.

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>Event</i> ou <i>Activity</i> .
representation	array	opcional	-
member_of	array	opcional	-
subject_of	array	opcional	-
attributed_by	array	opcional	-
part_of	array	opcional	-
timespan	array	opcional	-
took_place_at	array	opcional	-

Nome	Tipo	Exigência	Descrição
caused_by	array	opcional	-
influenced_by	array	opcional	-
carried_out_by	array	opcional	-
used_specific_object	array	opcional	-
part	array	obrigatório	Deve ser um dos seguintes objetos descritos a seguir <i>Acquisition, Payment, TransferOfCustody, Encounter, RightAcquisition, Move, Promise, Right</i>

**Fonte:** Elaborado pelo autor. Metadados utilizados na entidade *Provenance* do Linked Art (2021).

Os objetos descritos a seguir são opções de entrada para o campo *part* descrito no objeto *ProvenanceActivity* acima. Todos eles possuem as propriedades comuns do Linked Art, além de incluírem *timespan, took\_place\_at, influenced\_by, carried\_out\_by, used\_specific\_object*. Cada um dos objetos também possui propriedades exclusivas, descritas em seu respectivo quadro:

#### Quadro 28: Propriedades de *Acquisition*.

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>Acquisition</i> .
transfer_tite_of	array	obrigatório	Um array de referências a entidade <i>_PhysicalObject_</i> .
transfer_tite_from	array	opcional	Um array de referências a uma entidade <i>Person</i> ou <i>Group</i> .
transfer_tite_to	array	opcional	Um array de referências a uma entidade <i>Person</i> ou <i>Group</i> .

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *Acquisition* do Linked Art (2021).

#### Quadro 29: Propriedades de *Payment*.

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>Payment</i> .
paid_amount	array	opcional	Um objeto <i>MonetaryAmount</i> que descreve a quantia do pagamento.
paid_from	array	opcional	Um array de referências a uma entidade <i>Person</i> ou <i>Group</i> .
paid_to	array	opcional	Um array de referências a uma entidade <i>Person</i> ou <i>Group</i> .

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *Payment* do Linked Art (2021).

#### Quadro 30: Propriedades de *TransferOfCustody*.

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>TransferOfCustody</i> .



Nome	Tipo	Exigência	Descrição
transfer_custody_of	array	obrigatório	Um array de referências a entidade <i>_PhysicalObject_</i> .
transfer_custody_from	array	opcional	Um array de referências a uma entidade <i>Person</i> ou <i>Group</i> .
transfer_custody_to	array	opcional	Um array de referências a uma entidade <i>Person</i> ou <i>Group</i> .

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *TransferOfCustody* do Linked Art (2021).

### Quadro 31: Propriedades de *Encounter*.

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>Encounter</i> .
encountered	array	obrigatório	Um array de referências a entidade <i>_PhysicalObject_</i> .

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *Encounter* do Linked Art (2021).

### Quadro 32: Propriedades de *RightAcquisition*.

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>RightAcquisition</i> .
establishes	array	obrigatório	Um array de objetos <i>Right</i> descrito abaixo.
invalidate	array	opcional	Um array de objetos <i>Right</i> descrito abaixo.

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *RightAcquisition* do Linked Art (2021).

### Quadro 33: Propriedades de *Move*.

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>Move</i> .
moved	array	obrigatório	Um array de referências a entidade <i>_PhysicalObject_</i> .
moved_from	array	opcional	Uma referência a uma entidade <i>Place</i> .
moved_to	array	opcional	Uma referência a uma entidade <i>Place</i> .

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *Move* do Linked Art (2021).

### Quadro 34: Propriedades de *Activity*.

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>Activity</i> .
classified_as	array	obrigatório	Um array de objetos JSON, como a propriedade comum <i>classified_as</i> , porém um dos itens deve ter o valor <a href="http://vocab.getty.edu/aat/300435599">http://vocab.getty.edu/aat/300435599</a> para distinguir esta entidade como uma promessa.

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *Activity* do Linked Art (2021).

**Quadro 35: Propriedades de *Right*.**

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>Right</i>
identified_by	array	recomendado	-
classified_as	array	recomendado	-
referred_to_by	array	recomendado	-
dimension	array	opcional	-
possessed_by	array	opcional	Um array de referência a uma <i>Person</i> ou um <i>Group</i>
amplifies_to	array	opcional	Um array de referência a um <i>PhysicalObject</i>
part	array	opcional	Um array de objetos <i>Right</i>

**Fonte:** Elaborado pelo autor. Metadados utilizados no objeto *Right* do Linked Art (2021).

9. *Sets* - conjuntos de entidades (LINKED ART, 2021);

*Sets* possuem as seguintes propriedades:

**Quadro 36: Propriedades de *Set*.**

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>Period</i> , <i>Event</i> ou <i>Activity</i> .
representation	array	opcional	-
member_of	array	opcional	-
subject_of	array	opcional	-
attributed_by	array	opcional	-
dimension	array	opcional	-
created_by	array	opcional	-

**Fonte:** Elaborado pelo autor. Metadados utilizados na entidade *Set* do Linked Art (2021).

10. *TextualWork* - obras textuais que merecem descrição como entidades únicas, como conteúdo de livro ou artigos, entre outros (LINKED ART, 2021):

*TextualWorks* possuem as seguintes propriedades:

**Quadro 37: Propriedades de *Textual*.**

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>Event</i> ou <i>Activity</i> .
representation	array	opcional	-
member_of	array	opcional	-
subject_of	array	opcional	-

Nome	Tipo	Exigência	Descrição
attributed_by	array	opcional	-
language	array	opcional	-
dimension	array	opcional	-
part_of	array	opcional	-
content	array	opcional	-
format	array	opcional	-
digitally_carried_by	array	opcional	-
carried_by	array	opcional	-
about	array	opcional	-
refers_to	array	opcional	-
created_by	array	opcional	-
used_for	array	opcional	-

**Fonte:** Elaborado pelo autor. Metadados utilizados na entidade *Textual* do Linked Art (2021).

11. *VisualWork* - conteúdo imagético que merece descrição como entidades únicas, como a imagem exibida em uma pintura ou desenho, entre outros (LINKED ART, 2021).

*VisualWorks* possuem as seguintes propriedades:

### **Quadro 38: Propriedades de *Visual*.**

Nome	Tipo	Exigência	Descrição
type	string	obrigatório	O tipo deve ser <i>Event</i> ou <i>Activity</i> .
member_of	array	opcional	-
subject_of	array	opcional	-
attributed_by	array	opcional	-
dimension	array	opcional	-
part_of	array	opcional	-
digitally_shown_by	array	opcional	-
shown_by	array	opcional	-
about	array	opcional	-
represents	array	opcional	-
represents_entity_of_type	array	opcional	-
created_by	array	opcional	-
used_for	array	opcional	-

**Fonte:** Elaborado pelo autor. Metadados utilizados na entidade *Visual* do Linked Art (2021).

Nesta seção apresentamos quatro modelos de dados para representação e descrição da informação museológica, dois deles mais simples, o Object ID e o Esquema de metadados para descrição de obras de arte em museus brasileiros, que são ótimas opções para museus que não possuem um sistema de informação e pretende iniciar a documentação de seus itens. E os outros dois modelos, Spectrum e Linked Art, que são mais complexos e exigem um maior esforço por parte do museu para sua devida implementação, porém são os modelos mais completos e que abrangem diversas facetas da informação museológica.

O quadro a seguir compila, em resumo, as principais características dos modelos apresentados.

**Quadro 39: Comparação entre os modelos de dados Object ID (1997), Esquema de metadados para descrição de obras de arte em museus brasileiros (EMDOAMB) (2020), Spectrum (2022) e Linked Art (2021).**

Pontos de comparação	Object ID	EMDOAMB	Spectrum	Linked art
Descrição	Elaborada a partir de um esforço do Getty Information Institute que coletou resposta de 84 instituições museológicas ao redor do mundo para estabelecer um padrão mínimo de dados de representação de obras de arte	Criado pela Camila Aparecida da Silva, pesquisadora brasileira da USP (2020), com base nos modelos CDWA, Categorias de Informação do CIDOC-ICOM e nos grupos de informações do Spectrum. Seu foco principal é ser uma proposta de modelo de dados para representação de obras de arte em museus brasileiros	Procedimentos e unidades de informação para documentação museológica	Modelo de dados para aplicações para representação de informações sobre obras de artes em museus com foco na interoperabilidade entre sistemas
Origem	Lançado em 1997, é promovida por instituições como FBI, Scotland	Publicado em 2020 como resultado de uma pesquisa de doutorado da	Reino unido	Multi-institucional

Pontos de comparação	Object ID	EMDOAMB	Spectrum	Linked art
	Yard, Interpol e organizações como Unesco e WCO.	USP		
Versão	não se aplica	não se aplica	5.1	0.8.0
Mantenedores	Embora tenha sido criado pelo Getty Information Institute, o modelo foi mantido de 1999 a 2004 pelo <i>Council for the Prevention of Art Theft</i> (CoPAT) no Reino Unido, em 2005 o ICOM assina um acordo com J. Paul Getty Trust para uso não exclusivo do modelo, mantendo a licença deste então	Camila Aparecida da Silva	Collections Trust	Grupo de Trabalho Linked Art - CIDOC-ICOM
Principais utilizadores	J. Paul Getty Trust, European Commission, National Museum of World Cultures Foundation (NMVW - Países Baixos)	não se aplica, pois é um modelo de dados proposto para uso em museus brasileiros	Todos os museus acreditados à Collections Trust no Reino Unido	Getty, MoMA, Rijksmuseum
Principais utilizadores no Brasil	não encontrado	não encontrado	Pinacoteca de São Paulo, Museu do Café, Museu da Imigração	não encontrado
Formato	Ficha ou tabela simples	Ficha ou tabela simples	Não definido, estrutura “flat”. Apresenta metadados especializados para cada tipo de informação descrita, pode ser definido em fichas ou tabelas	JSON-LD. Apresenta metadados mais genéricos e abrangentes, porém que permitem a contextualização mais precisa com o uso de vocabulários controlados

Pontos de comparação	Object ID	EMDOAMB	Spectrum	Linked art
Dimensão	O Object ID define 9 categorias de informações e 4 passos para documentação	O EMDOAMB define 9 grupos de informações e 25 unidades de informação para descrição e representação de obras de arte brasileiras	O Spectrum define 21 procedimentos para gestão museológica e 546 metadados para representação dos objetos e entidades relacionadas, que são divididos em três grupos: object information, procedural information e record management information	O Linked Art apresenta 11 entidades principais para representação da informação museológica e mais 27 classes complementares. O modelo também apresenta 91 metadados diferentes que são utilizados para descrever as classes e entidades

**Fonte:** Elaborado pelo autor.

Para a elaboração do modelo utilizado no Elucidário.art, optamos por utilizar o Linked Art como base principal, por sua característica de ser um modelo de dados para a Web Semântica e focado na interoperabilidade dos dados. Seu formato em JSON permite a inserção de diferentes tipos de informações estruturadas e contextualizadas com o uso de vocabulários controlados externos, e a sua arquitetura REST permite a criação de uma API hipermídia que pode ser utilizada por diferentes aplicações. Embora o Spectrum, dos quatro, seja o modelo mais preciso e abrangente, ele não possui uma estrutura de dados definida, e exigiria um maior esforço na manutenção deste conjunto de dados. Os outros dois modelos, Object ID e Esquema de metadados para descrição de obras de arte em museus brasileiros, não abordam facetas da informação museológicas importantes para o Elucidário.art, como a proveniência, conservação, restauros, procedimentos realizados, direitos, entre outras.

A melhor das alternativas seria uma união entre o Linked Art e o Spectrum, em que utilizaríamos os procedimentos do Spectrum para definir os fluxos de trabalho do Elucidário.art, e seus grupos e unidades de informações seriam mapeados para a estrutura

do Linked Art. Esse mapeamento é possível em dois níveis, o primeiro diretamente entre as entidades do Linked Art e suas propriedades e os Grupos de informação do Spectrum, por exemplo, o grupo de informações de identificação do Spectrum poderia ser mapeado para os objetos *Identifier* ou *Name* do Linked Art, dependendo de qual tipo de identificação estamos descrevendo. O segundo nível é a contextualização por meio do uso de propriedades de referências do Linked Art que conectam com termos em vocabulários externos, como o AAT do Getty, que representam as unidades de informação do Spectrum, por exemplo o metadado “*Object Number*” do grupo “*Object identification information*” do Spectrum poderia ser descrito no Linked Art utilizando a propriedade *identified\_by*, com um objeto *Identifier* classificado como “identification number” (aat:300404626, <http://vocab.getty.edu/aat/300404626>), enquanto o metadado *Title* poderia ser mapeado para um objeto *Name* classificado como qualquer um dos termos descritos na hierarquia do AAT para “titles (general, names)”: *abbreviated titles*, *alternative titles*, *full title*, *original title*, entre outros, além disso podemos também utilizar o metadado *Title language* do Spectrum com a propriedade *language* e um objeto do tipo *Language* no Linked Art.

```
{
  "identified_by": [
    {
      "type": "Identifier",
      "_label": "M-0785",
      "classified_as": [
        {
          "id": "http://vocab.getty.edu/aat/300404626",
          "_label": "identification number"
        }
      ],
      "content": "M-0785"
    },
    {
      "type": "Name",
      "_label": "No campo",
      "classified_as": [
        {
          "id": "http://vocab.getty.edu/aat/300417204",
          "_label": "original titles"
        }
      ],
      "content": "No campo",
      "language": [
        {
```

```

        "id": "http://vocab.getty.edu/aat/300440719",
        "type": "Language",
        "_label": "Brazilian portuguese"
    }
],
},
{
    "type": "Name",
    "_label": "À La campagne",
    "classified_as": [
        {
            "id": "http://vocab.getty.edu/aat/300417204",
            "_label": "original titles"
        }
    ],
    "content": "À La campagne",
    "language": [
        {
            "id": "http://vocab.getty.edu/aat/300388306",
            "type": "Language",
            "_label": "French"
        }
    ]
}
]
}
}

```

## 5. Documentação museológica

O documento é suporte que evidencia algo a alguém e que, ao passar por um processo técnico específico, manifesta seu potencial informativo. Ele é o meio que nos traz a informação e, assim, permite que o indivíduo produza conhecimentos diversos. (PADILHA, 2014)

Além do objeto em si, o museu tem por obrigação manter a documentação dos itens em sua salvaguarda: “esta documentação deve permitir a identificação e a descrição completa de cada item, dos elementos a ele associados, de sua procedência, de seu estado de conservação, dos tratamentos a que já foram submetidos e de sua localização”. (CÓDIGO DE ÉTICA PARA MUSEUS, 2009).

O Código de Ética para Museus (2009) define que a documentação museológica deve abranger a descrição, os elementos associados, procedência, informações sobre o estado de conservação, tratamentos submetidos e histórico de localização. A Declaração de Princípios



de Documentação em Museus, publicação presente na Coleção Gestão e Documentação de Acervos: Textos de Referência, do Conselho Internacional de Museus (ICOM), esclarece que “o museu deverá implementar um sistema de documentação que inclua informações sobre os objetos e ofereça suporte a procedimentos práticos de gestão de acervo, tais como incorporação, gestão de empréstimos, localização de objetos e controle de sua movimentação” (CIDOC, 2014, p. 19).

Na ausência de uma autoridade nacional para definir o escopo de trabalho dos museus, o Código de Ética para Museus do ICOM (2009) deve ser utilizado como guia. No Brasil possuímos a lei 11.904 de 14 de janeiro de 2009, sancionada no segundo mandato do presidente Luís Inácio Lula da Silva instituindo o Estatuto de Museus e que dá outras providências. No artigo 39 desta mesma lei é definida uma das obrigações dos museus brasileiros: “manter documentação sistematicamente atualizada sobre os bens culturais que integram seus acervos, na forma de registros e inventários”. E aprofunda no parágrafo 1º: “o registro e o inventário dos bens culturais dos museus devem estruturar-se de forma a assegurar a compatibilização com o inventário nacional dos bens culturais”.

Já no decreto nº 8.124 de 17 de outubro de 2013 assinado pela então presidenta Dilma Rousseff, que regulamenta os dispositivos da lei nº 11.904, decreta no artigo 23, item IV, letra c, que o Plano Museológico deve conter um programa de acervos e que este programa abranja “o processamento técnico e o gerenciamento dos diferentes tipos de acervos da instituição, incluídos os de origem arquivística e bibliográfica”.

Ainda, como esclarecido por (PADILHA, 2014), a documentação museológica pode ser de dois vieses: documentação do objeto e a documentação das práticas administrativas. “O primeiro trata da compilação dos dados e do tratamento informacional extraídos de cada objeto adquirido pelo museu, enquanto que o segundo considera toda a documentação produzida pela instituição para legitimar suas práticas desenvolvidas” (PADILHA, 2014). Seguimos aqui o primeiro viés, o da documentação do objeto.

O Spectrum é um padrão para gestão de coleções museológicas criado no Reino Unido por Alice Grant e publicada inicialmente em 1994 pela *Museum Documentation Association* (MDA), em 2005 passa a ser editada por Gordon McKenna (3ª Edição) (MATOS, 2014). O Spectrum é utilizado por todos os museus acreditados pela *Collections Trust*

(COLLECTIONS TRUST, 2022) no Reino Unido e diversos outros ao redor do mundo, e define 21 procedimentos para gestão de coleções museológicas e unidades de informações utilizadas nos procedimentos.

Em 2014, com um esforço conjunto da Secretaria de Estado de Cultura, Associação de Amigos do Museu do Café e Pinacoteca do Estado de São Paulo, a versão 4.0 do Spectrum foi traduzida para o português-brasileiro [^1], baseado na versão previamente traduzida para o português pela Universidade do Porto em Portugal. Nesta versão, originalmente editada em 2011 por Alex Dawson e Susie Hillhouse, tem seu nome alterado de “UK Museum Documentation Standard” para “UK Collections Management Standard”, e apresenta uma nova estrutura que passa a publicar os procedimentos e unidades de informação em documentos separados, e os procedimentos passam a ser apresentados em fluxos de trabalho (COLLECTIONS TRUST, [s.d.]).

A versão 5.0 é publicada em 2017 e foi editada por Kevin Gosling e Gordon McKenna. Esta versão passa a apresentar os procedimentos em fluxos de trabalho e em texto utilizado em versões anteriores. O padrão também foi reescrito de uma forma mais clara e concisa, criando uma distinção entre “padrão” e “orientação”:

- **Padrão:** é o que deve ser feito, é obrigatório para a acreditação à *Collections Trust*;
- **Orientação:** é o que pode ser feito, é opcional e são apenas sugestões de boas práticas.

A versão 5.0 também substitui o procedimento “Documentação retrospectiva” por dois novos procedimentos: “Planejamento da documentação” e “Inventário”, para melhorar a distinção no que é válido como documentação retrospectiva (COLLECTIONS TRUST, [s.d.]).

A versão 5.1, publicada em 2022 como parte da campanha “rethink cataloguing” (COLLECTIONS TRUST, [s.d.]), modifica os procedimentos de Catalogação e Uso das coleções para estimular uma abordagem mais inclusiva nestas atividades (COLLECTIONS TRUST, [s.d.]).

Utilizamos a versão 5.1 em nossos estudos por ser a versão mais recente do Spectrum.

O Spectrum define 21 procedimentos para gestão de coleções museológicas (COLLECTIONS TRUST, 2017), são os procedimentos que um museu irá utilizar em todo o ciclo de vida de um objeto musealizado, desde a sua aquisição até a sua desincorporação.

Um procedimento Spectrum é descrito como um fluxo de trabalho, composto por uma definição, notas de escopo, o padrão Spectrum (o que o museu deve alcançar) e o procedimento sugerido. O padrão Spectrum que deve ser alcançado é dividido em duas partes: políticas e requisitos mínimos. As políticas são questões que o museu deve considerar ao definir o seu procedimento, e os requisitos mínimos são as atividades que devem ser realizadas para que o procedimento seja considerado minimamente completo. O procedimento sugerido também é apresentado em duas partes: um diagrama de fluxo de trabalho e uma descrição textual do procedimento (COLLECTIONS TRUST, 2022).

Dos 21 procedimentos, 9 são obrigatórios para acreditação à *Collections Trust* e são os que o museu irá utilizar em todos os objetos musealizados, os outros 12 são opcionais e o museu pode escolher quais utilizar de acordo com a sua necessidade. Os procedimentos podem ser entendidos como uma cadeia de atividades que se interconectam, em que um procedimento pode desencadear o outro.

Em cada procedimento o Spectrum identifica as unidades de informação que são utilizadas, e estas unidades de informação são definidas em um documento separado: o *Spectrum Units of Information*, que falaremos mais adiante no capítulo 4 - Metadados de representação e descrição da informação museológica.

A seguir listamos os 21 procedimentos do Spectrum, e os que são obrigatórios para a acreditação à *Collections Trust* estão marcados com um asterisco (\*). Como as políticas vão além do escopo do aplicativo, e sim do museu —por que você escolhe adquirir obras de arte contemporânea ao invés de obras do renascimento?— não iremos apresentá-las aqui, mas o Spectrum sugere como elaborar uma política para cada procedimento descrito. Os procedimentos serão apresentados em português, e os termos originais em inglês serão apresentados entre parênteses. Quando um procedimento fizer referência a outro, este será escrito em negrito, sem o termo original em inglês. As unidades de informações para cada procedimento podem aparecer ou em quadros, quando estivermos falando das unidades de

informações referentes a cada procedimento, ou em *itálico*, quando estivermos falando de alguma unidade de informação específica durante o texto.

1. **\*Entrada de objetos (*Object Entry*)**

Entrada de objetos no museu por qualquer motivo (COLLECTIONS TRUST, 2017).

1. Preparação para a entrada: Relacionado a **Aquisição e adesão** e **Entrada de empréstimos**

A preparação pode ser simples como avisar alguém que irá receber, ou durar anos como no caso de coleções arqueológicas.

- Se necessário arranjar transporte vá para **Localização e controle de movimentação**;
- Se necessário atualizar informações de seguro e indenização vá para **Seguro e indenização**

2. Crie um registro e um recibo:

Ver Quadro 1 a seguir.

1. Verifique e registre as condições do objeto e quaisquer riscos associados:
  - Vá para **Verificação de condições e avaliação técnica**;
  - Vá para **Reprodução**.

2. Envie ou entregue uma cópia do registro de entrada.

3. Processando objetos recém chegados:

1. Marque o objeto com um marcador temporário com o *Número de referência a entrada de empréstimos*;
2. Registre a primeira localização dos objetos - **Localização e controle de movimentação**;

3. Se o objeto é uma aquisição planejada ou entrada de empréstimos - volte ao procedimento relevante **Aquisição e adesão** ou **Entrada de empréstimos**;
4. Se o objeto chega inesperadamente - considere a oferta e vá para **Aquisição e adesão**;
5. Se o dono deixa para identificação - realize a tarefa no tempo acordado e vá para **Saída de objeto**;
6. Se o objeto chega anonimamente - lide de acordo com a política da instituição;

**Quadro 40: Grupos de informação para entrada de objetos.**

Grupo de informação	Exigência
<i>Informação de entrada de empréstimos (Loan in)</i>	opcional
<i>Informações de identificação do objeto (Object identification information)</i>	obrigatório
<i>Informações de descrição do objeto (Object description information)</i>	obrigatório
<i>Informações de entrada do objeto (Object entry information)</i>	obrigatório

**Fonte:** Elaborado pelo autor. Grupos de informação para o procedimento *Object Entry* do Spectrum 5.

2. **\*Aquisição e adesão (*Acquisition and accessioning*)**

O compromisso formal do corpo governante do museu de cuidar do objeto a longo prazo (COLLECTIONS TRUST, 2017).

1. Avaliação de potenciais aquisições

Faça um caso para a aquisição:

- Siga a política de aquisição do museu.

Avalie a proposta:

- Informe o doador ou vendedor sobre a decisão.

2. Obtenção de títulos e direitos autorais

Registro de evidências de propriedade e direitos autorais:

- Referência ao título formal da pessoa jurídica que está adquirindo o objeto;
  - O nome e endereço do antigo dono;
  - Uma breve descrição do objeto;
  - Assinatura confirmando a transferência de propriedade;
  - O método de aquisição.
3. Recebimento de objetos que ainda não está com você: Planejamento da chegada do objeto:

- Vá para **Entrada de objetos**.

4. Processando novas aquisições

Etiquetagem ou marcação de objetos:

- A política deve especificar o formato de numeração e marcação.

Registro de informações:

- Ver Quadro 2 a seguir.

Se o objeto for um presente, envie agradecimentos e um recibo:

- Inclua informações sobre como o objeto pode ser acessado no futuro.

5. Adesão de objetos à coleção:

Se o objeto adquirido for fazer parte da coleção permanente do museu, faça o registro das seguintes informações:

- *Número de entrada (Entry number);*
- *Número do objeto (Object number);*
- *Data de aquisição (Date acquired);*
- *De quem foi adquirido (Who it was acquired from);*
- *Descrição (Description);*
- *Nome do objeto (Object name);*
- *Título (Title);*

- *Breve descrição (Brief description);*
- *Outras informações (Any other information).*

#### **Quadro 41: Grupos de informação para aquisição de objetos**

Grupo de informação	Exigência
<i>Informação de identificação do objeto (Object identification information)</i>	obrigatório
<i>Informação de aquisição (Acquisition information)</i>	obrigatório
<i>Histórico do objeto e informações de associação (Object history and association information)</i>	obrigatório
<i>Informações de uso do objeto (_Object use information)</i>	obrigatório
<i>Informações de contribuição dos proprietários do objeto (Object owners contribution information)</i>	obrigatório

**Fonte:** Elaborado pelo autor. Grupos de informação para o procedimento *Aquisição e adesão* do Spectrum 5.

### **3. \*Localização e controle de movimentação (*Location and movement control*)**

Registro da localização dos objetos e controle de movimentação, seja internamente ou externamente ao museu (COLLECTIONS TRUST, 2017).

#### **1. Identificando e descrevendo localizações**

Atribua um nome para cada display ou área de armazenamento

Registre as informações da localização de cada objeto;

- *Número de referência da localização (Location reference number);*
- *Tipo de localização (Location type);*
- *Endereço da localização (Location address);*
- *Condições da localização (Location condition);*
- *Nota da condição da localização (Location condition note);*
- *Data da nota da condição da localização (Location condition note date);*
- *Nota de segurança da localização (Location security note);*
- *Nota sobre o acesso a localização (Location access note).*

#### **2. Mantenha um sistema de localização: Faça atualizações e adicione informações quando necessário:**

- *Tipo de medição do ambiente (Environment environment type);*
  - *Medida do ambiente (Environment measurement value);*
  - *Unidade de medida do ambiente (Environment measurement value unit);*
  - *Data da medida do ambiente (Environment measurement date).*
3. Registre a localização dos objetos: Ver Quadro 3 a seguir.
4. Movimentação de objetos Geralmente ocorre em decorrência de outro procedimento Spectrum.
- Obtenha e registre autorização para todas as movimentações,
  - Registre as informações do grupo *Informação de movimentação (Movement information)*;
  - Verifique as condições do objeto antes de movimentá-lo;
  - Vá para **Verificação de condições e avaliação técnica**;
  - Verifique se há riscos ou recomendações no manuseio e movimentação:
    - *Recomendações de exibição (Display recommendations);*
    - *Recomendações de ambiente (Environment recommendations);*
    - *Recomendações de manuseio (Handling recommendations);*
    - *Recomendações de embalagem (Packing recommendations);*
    - *Recomendações de segurança (Security recommendations);*
    - *Recomendações de transporte (Transport recommendations)*
    - ;
    - *Recomendações de transporte (Specials recommendations);*
    - *Recomendações de armazenamento (Storage recommendations);*
  - Verifique se há obstáculos para a movimentação;
  - Providencie qualquer atividade de conservação ou embalagem necessária;



- Vá para **Cuidado e conservação da coleção**.
- Se nenhum transporte está envolvido, vá para *Atualize o registro do objeto*;
- Se o transporte está envolvido, defina e registre o meio mais apropriado;
- Providencia seguro e indenização se necessário:
  - Vá para **Seguro e indenizações**. Providencia devida documentação para acompanhar o objeto:
  - Transporte e confirme a chegada do objeto:
  - Atualize o registro do objeto e de controle de movimentação:
- Ver Quadro 4 a seguir;
- Retorne ao procedimento que iniciou a movimentação.

#### **Quadro 42: Grupos de informação para localização de objetos.**

Grupo de informação	Exigência
Object identification information	requerido
Object location information	requerido

**Fonte:** Elaborado pelo autor. Grupos de informação para o procedimento *Object location and movement control* do Spectrum 5.

#### **Quadro 43: Grupos de informação para movimentação de objetos.**

Grupo de informação	Exigência
Object identification information	requerido
Movement information	requerido
Object location information	requerido

**Fonte:** Elaborado pelo autor. Grupos de informação para o procedimento *Object location and movement control* do Spectrum 5.

#### **4. \*Inventário (*Inventory*)**

Certifica que você tem as informações básicas de cada objeto (COLLECTIONS TRUST, 2017).

1. Todos os procedimentos primários foram completados:

1. Verifica se você atendeu a todos os requisitos dos outros 8 procedimentos primários
  2. Se todos os critérios dos outros procedimentos principais forem atendidos, vá para a próxima etapa, se não volte para o procedimento principal que não foi atendido.
2. Verifique se você possui informações fundamentais do objeto
    1. Verifique as informações:
      1. *Object number*;
      2. *Object name*;
      3. *Number of objects*;
      4. *Brief description*;
      5. *Current location*;
      6. *Current owner*;
      7. *Recording date*.
    2. Crie um plano para obter as informações que faltam.
      - Va para e volte de *Documentation planning*.
  3. Produzindo um inventário
    1. Vá para as localizações que está inventariando e liste todo objeto neste local.
    2. Verifique se o objeto possui uma marcação de numeração ou etiqueta
      1. Caso não possua, atribua um número temporário;
    3. Atualize ou crie um registro para objetos que possuem marcação ou etiqueta;
  4. Verifique por discrepâncias entre o inventário e o registro
    1. Tente identificar os objetos com numeração temporária;
    2. Se for possível identificar o objeto, remova a marcação temporária adicione a marcação correta;

3. Atualize o registro do objeto, incluindo a localização;
4. Mantenha informações essenciais por meio de outros procedimentos.
5. Resolução de problemas pendentes

- Decida como lidar com os objetos que não foram identificados;

5. **\*Catalogação (*Cataloguing*)**

É o processo contínuo de registro e gestão da informação sobre as coleção, sob diferentes perspectivas (COLLECTIONS TRUST, 2017).

1. Criando registro de catalogação:

1. Crie um registro para cada objeto, grupo de objetos, acessíveis via numeração, as informações obrigatórias são:

- *Object number*;
- *Object name*;
- *Number of objects*;
- *Brief description*;
- *Current location*;
- *Current owner*;
- *Recording date*.

2. Adicione outras informações disponíveis

1. Adicione as informações como descrita no Quadro 5 a seguir;
2. Verifique se as informações registradas atendem o seu alvo mínimo de informações.

- Relacionado com *Audit*;
- Se houver problemas ou melhoras a fazer, vá para e retorne de *Documentation planning*.

2. Adicionando a registros de catalogação:

1. Adicione informações sobre projetos de documentação

- Relacionado com *Documentation planning*.
- 2. Adicione informações sobre pesquisa e interpretação
  - Relacionado com *Use of collections*.
- 3. Adicione informações advindas de pesquisa e interpretação
  - Relacionado com *Use of collections*.
- 4. Adicione informações sobre outros procedimentos.
- 3. Mantendo registros de catalogação
  1. Providencias acesso aos registros por meio de indexação:
    - Object name;
    - Subject;
    - Artist;
    - Source
    - Production date;
    - Associated people, places and events;
    - entre outros.
  2. Mantenha o catalogo seguro, incluindo backups, e cópias impressas
- 4. Auditando registros de catalogação
  - Relacionado com *Audit*.

#### **Quadro 44: Grupos de informação para catalogação de objetos.**

Grupo de informação	Exigência
Object identification information	requerido
Object description information	requerido
Object history and association information	requerido
Field collection information	requerido
Object production information	requerido
Object owners contribution information	requerido
Reference information	requerido
Amendment history (information about change to the catalogue)	requerido

**Fonte:** Elaborado pelo autor. Grupos de informação para o procedimento *Cataloguing* do Spectrum 5.

**6. \*Saída de objetos (*Object exit*)**

O procedimento para registrar a saída de objetos do museu, seja por empréstimo, doação, venda, alienação, etc (COLLECTIONS TRUST, 2017).

1. Autorize a saída de acordo com a sua política e o procedimento relacionado;
2. Se o objeto será transportado, vá para *Location and movement control*;
  - Se o objeto será coletado diretamente pelo dono, vá para o próximo passo, se não, vá para *Location and movement control*;
3. Agende a coleta dos objetos;
4. Providencia para o objeto estar na hora e local combinados para a coleta;
5. Atualize o registro do objeto para refletir a saída;
  - Grupo de informações *Movement information*
6. Registre informações sobre a saída do objeto, como descrito no Quadro 6 a seguir;
7. Atualize *Insurance and indemnity* se necessário;
8. Retorne para o procedimento que levou a saída do objeto.

**Quadro 45: Grupos de informação para saída de objetos.**

Grupo de informação	Exigência
Object identification information	requerido
Object entry information	requerido
Object exit information	requerido
Object valuation information	requerido
Object condition information	requerido

**Fonte:** Elaborado pelo autor. Grupos de informação para o procedimento *Object exit* do Spectrum 5.

**7. \*Entrada de empréstimos (*Loans in*)**

Gerenciar objetos emprestados por um período fixo de tempo e para uma finalidade específica (COLLECTIONS TRUST, 2017).

1. Pesquisa para o empréstimo

1. Faça o caso para o empréstimo e obtenha autorização;
2. Envie um pedido de empréstimo para o prestador;

Deve conter:

- Datas de início e término do empréstimo;
  - Orçamento para o empréstimo;
  - Contexto e propósito do empréstimo;
  - Informações sobre o responsável pelo empréstimo;
  - Contato e informações sobre o responsável pelo empréstimo;
  - Breve descrição do objeto;
  - Informações sobre o seguro e indenização;
3. Registre detalhes da solicitação como descrito no Quadro 7 a seguir:
  4. Mantenha um registro atualizado do status do empréstimo;

2. Troca de informações adicionais:

1. Conclua e envie um relatório sobre as instalações do museu para o prestador;
2. Obtenha informações adicionais para cada objeto e atualize o registro;

As informações são:

- O número de registro do objeto pelo prestador (se necessário);
- Descrição do objeto;
- Nome e endereço do prestador;

- Sua avaliação do valor do objeto;
- Sua condição;
- Suas exigências de exibição;
- Suas exigências de ambientação;
- Suas exigências de manuseio;
- Quaisquer riscos potenciais;
- Dimensões
- Fotografias;
- Detalhes de quaisquer direitos autorais ou outras restrições;
- Quaisquer informações históricas ou de contexto;
- Número de exibição (se necessário);

3. Fechando acordo de empréstimo:

1. Prepare um contrato de empréstimo;
2. Registre informações sobre o empréstimo:

Registre as *Loan in information*

4. Preparando para o recebimento do objeto:

1. Agende e prepare a recepção do objeto;
  - *Insurance and indemnity*;
  - *Object entry*;
  - *Location and movement control*;
2. Registre informações sobre a recepção do objeto e envie para o emprestador;
  - *Object entry*
3. Se trabalho de conservação for necessário, combine com o emprestador;
  - *Collections care and conservation*

5. Monitore o empréstimo:
  1. Monitore e reporte as condições do objeto durante o empréstimo
    - *Condition checking and technical assessment*
    - *Location and movement control*
6. Estender o empréstimo:
  1. No caso de extensão do período de empréstimo
    - *Insurance and indemnity;*
    - *Valuation*
7. Retornar o objeto:
  1. Contate o prestador para organizar a devolução do objeto;
  2. Realize uma nova avaliação de condições objeto;
    - *Condition checking and technical assessment*
  3. Retorne o objeto;
    - *Object exit;*
    - *Location and movement control;*
  4. Confirme que todas as condições foram cumpridas e feche o registro do empréstimo;

**Quadro 46: Grupos de informação para entrada de empréstimos de objetos.**

Grupo de informação	Exigência
Loan in information	requerido
Object identification information	requerido

**Fonte:** Elaborado pelo autor. Grupos de informação para o procedimento *Loan in (borrowing objects)* do Spectrum 5.

**8. \*Saída de empréstimos (*Loan out - lending objects*)**

Avaliação de solicitação de empréstimos e gerenciamento dos processos até que o objeto retorne a você (COLLECTIONS TRUST, 2017).



1. Avaliação do pedido

1. Crie um caso para o pedido de empréstimo;
2. Verifique os detalhes do pedido como descrito no Quadro 8 a seguir;
3. Considere o pedido de empréstimo;
4. Comunique a decisão ao prestador;
5. Reserve o objeto para empréstimo;

2. Solicitando informações adicionais:

1. Solicite informações adicionais sobre o solicitante;
2. Considere fatores como:
  - Referências do solicitante;
  - Informações obtidas em uma visita ao local proposto para a exibição;
  - Consulte órgãos setoriais regionais ou nacionais, se relevante;
  - Garanta que o objeto está adequadamente segurado;
  - Cronograma necessário para cumprir o empréstimo;
  - Custos;
  - Requisitos especiais, como couriers;
  - Considerações de segurança;
  - Considerações de ambientação;

No caso de não aceitação escreva uma justificativa e envie ao solicitante;

3. Concordado com o empréstimo:

1. Envie informações adicionais ao solicitante:
  - *Object number*;
  - *Object description*;

- *Valuation;*
- *Condition;*
- *Display requirements;*
- *Environmental requirements;*
- *Handling requirements;*
- *Any potential risks;*
- *Dimensions;*
- *Photographs;*
- *Details of any copyright or other restrictions;*
- *Any historical or contextual information;*
- *Estimated costs;*

2. Registre as condições finais do empréstimo e crie o contrato;

Registre as informações de *Loan out information*:

3. As duas partes assinam o acordo de empréstimo;

4. Preparando para o empréstimo:

1. Verifique e registre as condições do objeto:

- *Condition checking and technical assessment;*

2. Realize quaisquer trabalhos de conservação necessários:

- *Collections care and conservation;*

3. Assegure-se que todas as exigências de segurança foram atendidas;

4. Obtenha garantias de seguro e indenização e atualize os registros:

- *Insurance and indemnity;*

5. Enviando os objetos:

Realize as providências para a saída do objeto

- Vá para *Object exit* e *Location and movement control*;

6. Monitore o empréstimo:
  1. Monitore as condições e localização do objeto;
7. Estender o empréstimo
  1. Considere o pedido de extensão;
    - *Valuation*;
    - *Insurance and indemnity*;
8. Providencie o retorno:
 

Confirme as providências para o retorno do objeto;

  - Vá para *Location and movement control*;
9. Encerrando o empréstimo:
  1. Envie a fatura de quaisquer custos adicionais;
  2. Confirme o retorno do objeto e que todas as condições foram atendidas;
  3. Adiciona informações sobre o empréstimo ao registro do objeto;
    - *Use of collections*;
  4. Encerre o caso do empréstimo.

**Quadro 47: Grupos de informação para saída de empréstimos de objetos.**

Grupo de informação	Exigência
Object identification information	requerido
Loan out information	requerido

**Fonte:** Elaborado pelo autor. Grupos de informação para o procedimento *Loan out (lending objects)* do Spectrum 5.

9. **\*Planejamento da documentação (*Documentation planning*)**

Tornar o sistema de documentação melhor e aprimorar as informações sobre as coleções, em um processo de melhoria contínua (COLLECTIONS TRUST, 2017).

Este processo é mais genérico que os demais, são foco principal é o planejamento de melhorias em qualquer projeto de documentação.

1. Revisão das informações existentes sobre a coleção:
  1. Verifique se as informações registradas atendem o seu alvo mínimo de informações.
    - *Inventory*;
    - *Collections review*;
    - *Audit*;
    - *Rights management*;
2. Escreva o plano de documentação

Deve conter:

  - Objetivos;
  - Ações
  - Recursos;
  - Resultados esperados;
  - Metas;
3. Coloque em prática o plano
  1. Receba aprovação do plano de documentação;
  2. Coloque o plano em ação;
  3. Revise regularmente o progresso baseado nas metas;
  4. Finalize o trabalho e avalie o plano;
4. Melhoria continuada:
  - Repita o procedimento;

## **6. A Casa Museu Ema Klabin**

A Casa Museu Ema Klabin é uma casa museu de colecionador localizada na Rua Portugal, 43 do bairro Jardim Europa em São Paulo, SP.

**Imagem 1: Fachada da Casa Museu Ema Klabin vista do Jardim.**



*Foto de Nelson Kon, 2022.*

Ema Klabin (1907 - 1994) foi uma mecenas e colecionadora de arte, filha de Hessel Klabin, um dos sócios-fundadores da empresa hoje conhecida como Klabin S.A., de papel e celulose. Em 1946, antes de seu falecimento, Hessel doou todos seus bens para as duas filhas (Ema e Eva Klabin), inclusive sua parte da empresa (COSTA, 2007). Paulo de Freitas Costa, curador da Casa Museu Ema Klabin, descreve Ema em seu livro “Sinfonia de Objetos” (COSTA, 2007) como uma “grande apreciadora de música e de arte, leitora voraz e frequentadora assídua de concertos e apresentações de teatro, ópera e balé, bem como de exposições em museus e galerias”, Paulo também menciona a atuação ativa de Ema no setor cultural paulistano:

Participou dos conselhos da Fundação Bienal de São Paulo, do Museu de Arte de São Paulo Assis Chateaubriand—MASP— e do Museu de Arte Moderna de São Paulo, e colaborou na criação do Museu Lasar Segall e da Fundação Magda Tagliaferro, entre outros. Foi sócia da Sociedade de Cultura Artística, da Orquestra Filarmônica de São Paulo e da Sociedade dos Cem Bibliófilos do Brasil, criada por Raymundo Ottoni de Castro Maya. (COSTA, 2007)

## **6.1. A Coleção Ema Klabin**

A Coleção Ema Klabin é formada por mais de 1700 itens e “contém exemplares das mais variadas procedências e técnicas, e seu denominador comum —seu fio condutor— é o período de vida de Ema Klabin e o julgamento estético e histórico que norteou suas escolhas” (COSTA, 2007). Uma das grandes características da coleção é justamente sua heterogeneidade, sendo composta por pinturas, gravuras, esculturas, artes decorativas, arte-sacra, roupas, e outros. Os itens são de diversos períodos e culturas e abrangem um arco histórico de 35 séculos (COSTA, 2021). Ema adquiriu a primeira peça da coleção “Ariadne” de Jean Baptiste Greuze em 1948, e as outras aquisições aconteceram ao longo de sua vida.

Costa (2007) define quatro movimentos da coleção de Ema, sendo o primeiro “A tradição Europeia”, que segue o processo de formação de muitas coleções ao redor do mundo. Marchands aproveitaram o momento que a Europa estava se recuperando dos danos pós-guerra para negociar obras para o Novo Mundo, entre eles destaca-se Pietro Maria Bardi, que desembarca no Brasil em 1946 e influencia a formação de diversas coleções privadas no Brasil como a de Ema Klabin e do Museu de Arte de São Paulo (MASP).

“Ainda dentro da tradição europeia, mas em contraponto à cultura ocidental, está o conjunto de peças orientais reunidas por Ema Klabin” (COSTA, 2007). Este movimento da coleção ainda pode ser subdividido pela intenção pela qual foram colecionados em três conjuntos: “objetos decorativos feitos no oriente para o mercado europeu, os souvenirs de viagem e as peças arqueológicas”;

**Imagem 2: Salão da Casa Museu Ema Klabin.**



*Foto de Nelson Kon, 2022.*

Na Imagem 2, em primeiro plano vemos a coleção de bronzes chineses, com peças do século XIV ac como o Recipiente para aquecimento de vinho ritual (jue) da Dinastia Shang (o segundo da direita para esquerda). No centro do salão, em cima da mesa uma cópia de estudo de uma ânfora, provavelmente realizada em Roma no século XIX. Ao fundo o quadro Triunfo de Baco e Ariadne de Giovanni Battista Gaulli (Il Baciccio) de 1675.

Conforme Costa (2007), o segundo movimento “Raízes brasileiras” apresenta temas totalmente novos, relacionados ao nosso passado imperial e colonial, explorando os elementos artísticos brasileiros. As primeiras peças brasileiras a entrar na coleção de Ema são as imagens do Bom Jesus da Lapa, da Santa Luzia, da Nossa Senhora Divina Pastora e da Nossa Senhora da Assunção, que foram expostas inicialmente na exposição Imagens Religiosas Brasileiras, realizada em comemoração ao IV Centenário da Cidade de São Paulo, junto com mais outras exposições que buscavam entender a formação da nação brasileira a partir de referências paulistas (COSTA, 2007). Costa destaca que a formação da coleção brasileira de Ema se caracteriza por um olhar estrangeiro, embora Ema tenha



nascido no Rio de Janeiro, é filha de imigrantes e teve sua educação realizada na Europa “para Ema, talvez, uma Virgem Maria de nosso barroco fosse tão distante quanto uma Guanyin chinesa do século XVII” (COSTA, 2007);

**Imagem 3: Detalhe da Sala de Jantar da Casa Museu Ema Klabin. Foto de Nelson Kon.**



*Foto de Nelson Kon, 2022.*

Na Imagem 3 ao centro vemos um armário-vitrine do século XX construído com fragmentos de talhas do Mestre Valentim da Fonseca, dentro dela vemos diversas imagens



sacras. Ao lado, a direita, vemos uma placa misular do século XIX também do Mestre Valentim da Fonseca, logo abaixo vemos a Imagem de Santana Mestra do século XVIII. À esquerda do armário vitrine, Vaso com Flores de Maurice de Vlaminck e Natureza morta com limões e xícara de Pierre-Auguste Renoir.

O terceiro movimento “Aproximações com o Modernismo tem seu início no ano de 1961,”quando Ema, recém-instalada em sua nova casa no Jardim Europa, assume um papel mais significativo no cenário cultural da cidade” (COSTA, 2007). Este movimento se caracteriza pela entrada de obras representativas do modernismo brasileiro. Neste período Ema se empenhou na organização de leilões em prol da construção do Hospital Israelita Albert Einstein, “que tiveram papel importante na consolidação do mercado de arte moderna brasileira” (COSTA, 2007). As aquisições realmente significativas deste movimento ocorreram no final da década de 1960, quando Ema se envolveu na criação do Museu Lasar Segall, e adquiriu três telas do artista: Retrato Feminino (1927), Retrato de Lucy (1937) e Natureza-morta com frutas (1938).

**Imagem 4: Detalhe do quarto principal da Casa Museu Ema Klabin.**



*Foto de Nelson Kon, 2022.*

Na Imagem 4 vemos o Retrato Feminino de Emiliano Di Cavalcanti de 1955 à esquerda, Rebanho em Repouso de 1944 de Lasar Segall à direita do espelho, e mais à direita a tela Galos de Cândido Portinari de 1940.

Em 1972, Ema e sua irmã Eva, preocupadas com o destino de suas coleções, convidam Karl Katz, especialista em arte e arqueologia e membro do conselho do Metropolitan Museum de Nova Iorque, para contribuir com um projeto para as coleções, iniciando assim o quarto movimento da coleção “Últimas aquisições - novas direções”. Após a visita, Katz apresentou uma proposta de catalogação das duas coleções e sugeriu a criação de um ‘Centro de Altos Estudos em Museologia’ no Brasil” (COSTA, 2007), mas somente a ideia de transformar as casas em museu prevaleceu. A partir deste momento, Ema volta seus olhares para obras que “‘coroassem’ a coleção ou, melhor dizendo, seu ‘museu’” (COSTA, 2007), momento que adquiriu sua primeira obra de Marc Chagall, *À la Campagne*, em 1972, obra que se tornaria a mais valiosa da coleção. Seguindo a mesma lógica, Ema passa a buscar obras de altíssimo valor em leilões da casa Sotheby’s de Londres: “de uma só vez,

adquiriu telas de Chaim Soutine, Pierre-Auguste Renoir e Maurice de Vlaminck. Essas obras entraram para a coleção justamente no final de 1978, quando ela acabara de registrar oficialmente sua Fundação” (COSTA, 2007).

**Imagem 5: Detalhe da sala de música da Casa Museu Ema Klabin**



*Foto de Nelson Kon, 2022.*

Na Imagem 5 vemos em destaque a tela *À la Campagne* de Marc Chagall de 1925, à sua esquerda o relógio de piso francês do século XVIII.

## **6.2. A documentação da Coleção Ema Klabin**

O primeiro inventário da coleção foi realizado por Marcelo Mattos Araújo por insistência dos parentes de Ema Klabin, entre os anos de 1989 e 1991 (COSTA, 2007), resultando em uma lista dos itens da coleção separadas por ambientes. Em 1997, 3 anos após o falecimento de Ema (1994), Paulo de Freitas Costa inicia os trabalhos de catalogação e pesquisa da coleção que deu origem a sua dissertação de mestrado e posteriormente no livro “Sinfonia de Objetos” publicado em 2007 pela editora Iluminuras, que se apresenta como um estudo sobre o colecionismo brasileiro e sobre a coleção de Ema Klabin.

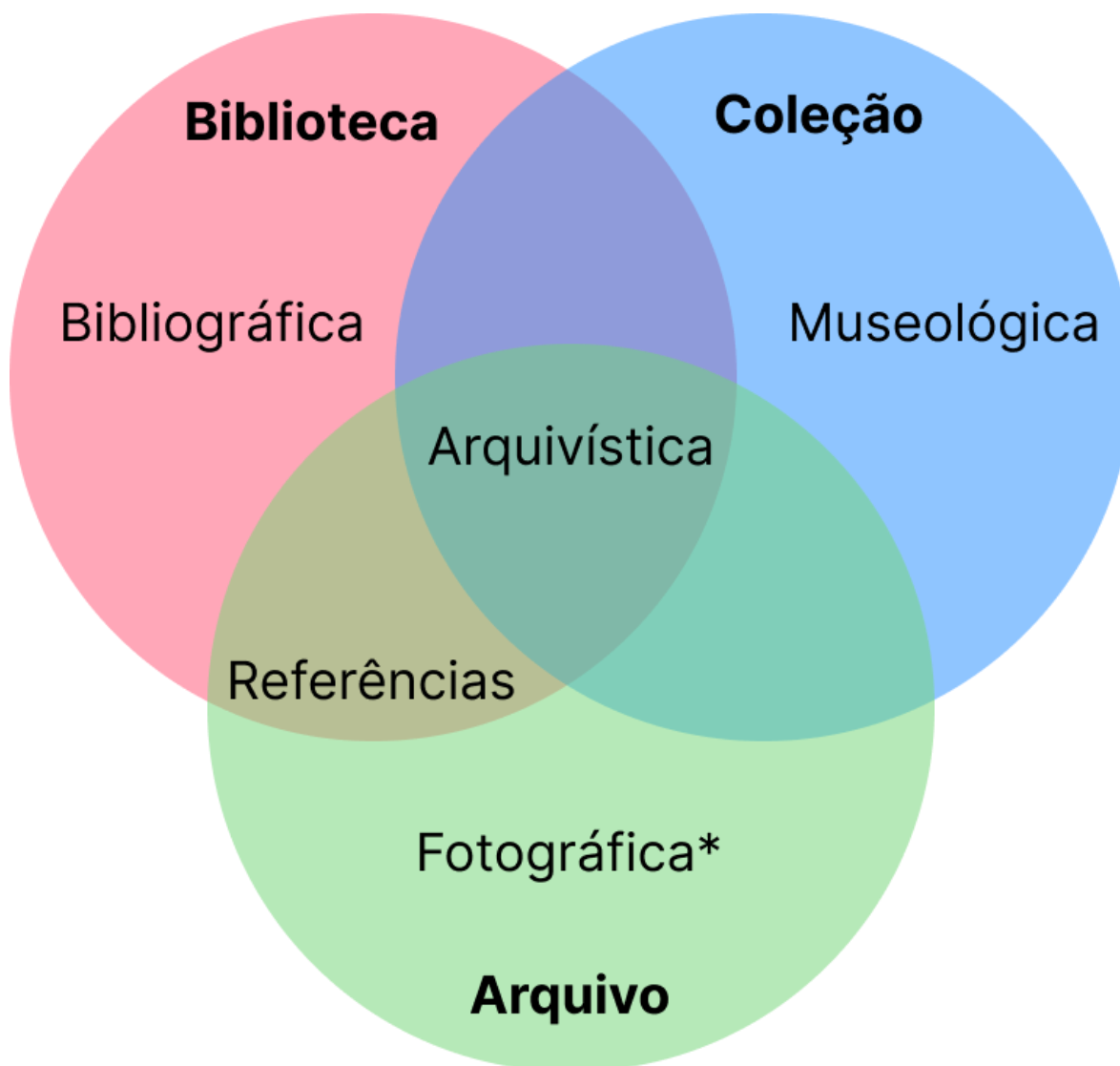
Em 1999 foram criadas cinco bases de dados para armazenar todas informações das coleções de itens museológicos, livros, fotografias pessoais e arquivos, sendo separadas em: “Museológico” para as informações sobre os itens da coleção de artes visuais e artes decorativas com atualmente 1708 itens; “Arquivístico” para os documentos pessoais de Ema, bem como suas correspondências, notas fiscais de compra de obras, negociações com marchands, etc, com 5261 itens; “Fotográfico” para as fotografias pessoais de Ema Klabin com 2132 fotografias; “Bibliográfico”, que armazena as informações sobre a coleções de livros e livros raros de Ema com 3554 publicações; e por fim, “Referências” que armazena as informações sobre livros, artigos e outras referências textuais que foram adquiridas ou recebidas como doação pela já então estabelecida Casa Museu Ema Klabin, ou seja, enquanto as informações do BD Bibliográfico são sobre os livros que pertenceram a Ema, as informações do BD Referências são sobre livros que foram adquiridos pela Casa Museu Ema Klabin após a sua fundação.

Todos os itens da coleção possuem um número de tombo —o principal metadado para identificação entre as demais entradas na BD —, que é composto pela letra M, indicando a base Museológica, seguido pelo hífen (-) para separar a indicação da BD do número sequencial de entrada no registro. A numeração é composta por quatro algarismos de 0 a 9, possibilitando a entrada de até 9999 itens. Por exemplo: M-0001, M-0002, (...), M-1660. As outras bases de dados seguem a mesma estrutura: A para arquivístico, B para bibliográfico, F para fotográfico e R para referências.

Essas cinco bases se organizam em três grupos: “Coleção”, “Biblioteca” e “Arquivo”, em que “Coleção” é composta pelas BD Museológica e Arquivística, “Biblioteca” é composta

pela BD Bibliográfica, e “Arquivo” pelas BD Fotográfica, Arquivística e Referências. A imagem a seguir demonstra essa organização, e as BDs marcadas com um asterisco (\*) não são públicas.

Imagem 6: Grupos e bases de dados da Casa Museu Ema Klabin.



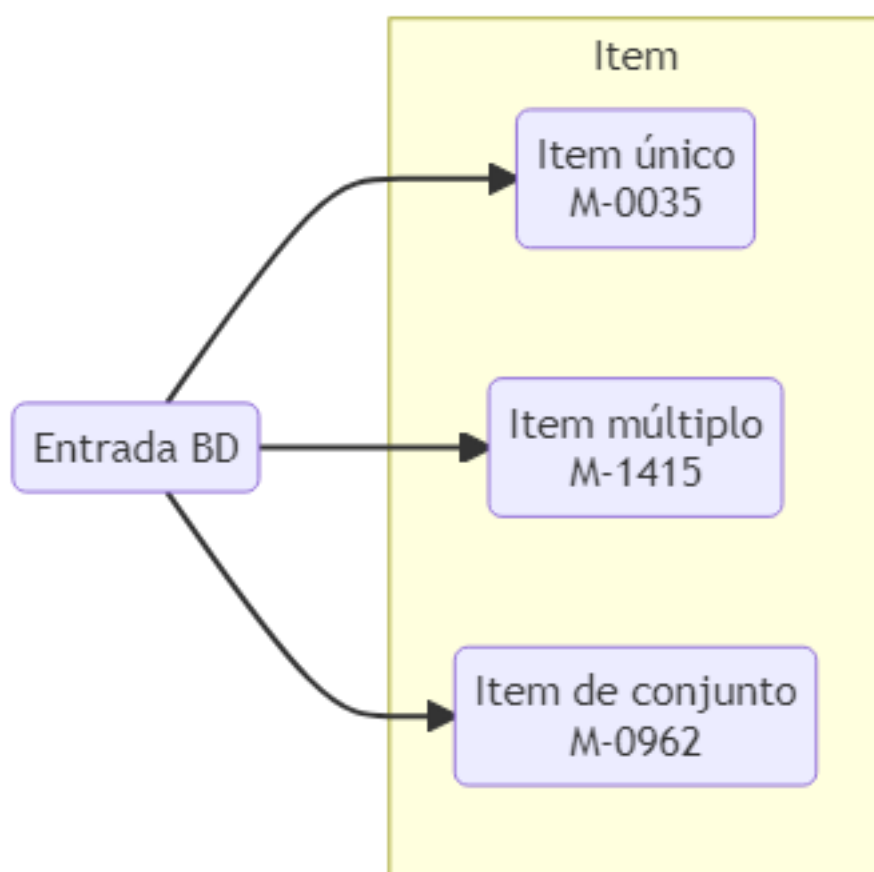
*Grupos e bases de dados da Casa Museu Ema Klabin.*

**Fonte:** Elaborado pelo autor.

Devido a heterogeneidade da Coleção alguns itens são simples, como uma peça esculpida em pedra ou um desenho em uma folha de papel, enquanto outros são itens compostos por diversos outros itens, como os faqueiros e os jogos para chá. Ou até mesmo objetos

compostos por duas ou mais peças de materiais diferentes, como no caso dos frascos para rapé que possuem o corpo e a tampa —alguns ainda possuem uma terceira peça que seria a base. Ou ainda, as sopeiras, que possuem corpo e tampa e fazem parte dos conjuntos de porcelanas chinesas de exportações, ou seja, duas peças distintas que constituem um único item pertencendo a um conjunto de itens. Com exceção do primeiro caso (item simples), a sintaxe de numeração não dá conta da real característica física da coleção. O esquema abaixo ilustra a estrutura atual de numeração:

**Figura 2: Estrutura de numeração da Casa-Museu Ema Klabin.**

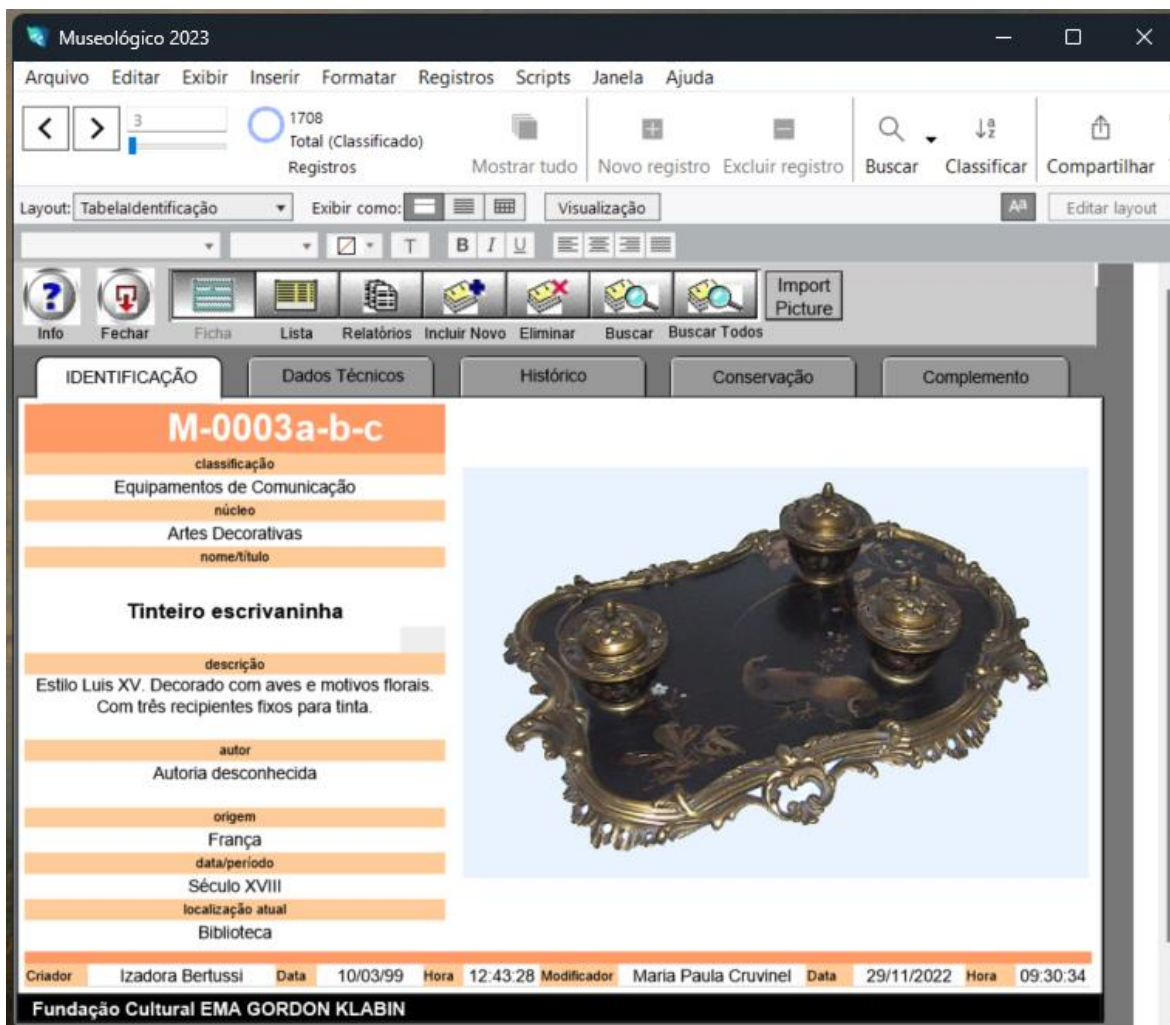


**Fonte:** Elaborado pelo autor.

Em alguns casos, a numeração de itens múltiplos na documentação da coleção é feita utilizando letras:



Imagem 7: Tela do FileMaker, software utilizado atualmente para gestão das BDs da Casa Museu Ema Klabin, demonstra a numeração de um item múltiplo utilizando sistema alfanumérico.



*Fonte: Tela do Filemaker demonstrando numeração com sistema alfanumérico.*


Esta sintaxe de numeração dos itens não corresponde às práticas recomendadas por modelos e guias como Cataloguing Cultural Objects: A Guide to Describing Cultural Works and Their Images (CCO) e Categories for the Description of Works of Art (CDWA) mantido pelo Getty Vocabulary Program (GVP). O CCO recomenda o registro separado das partes do todo e um registro para o todo em si (BACA, 2006), e que o catalogador faça o registro dessas relações entre os objetos. Para isso, o guia define três tipos diferentes de relações intrínsecas, ou relações hierárquicas parte-todo: grupos ou coleções, séries,

componentes ou obras arquitetônicas. Já a diretriz CDWA apresenta uma lista de possibilidades recomendadas que definem o nível de catalogação de um trabalho artístico: item, volume, álbum, grupo, subgrupo, coleção, série, conjunto, múltiplos, componentes, caixa, fundo, portfólio, suíte, complexo, agrupamento de objetos, performance e itens (HARPRING; BACA, 2022). O CDWA, assim como o CCO, define essa informação como obrigatória, e é desejável que as partes sejam registradas separadamente do todo, com a indicação de suas relações intrínsecas.

Um outro problema é a representação dos itens nas BDs Filemaker, que estão incompletas e não estruturadas. Informações como Material e Técnica do item estão em um único campo chamado MaterialTécnica, e as informações armazenadas neste campo não aderem a nenhum vocabulário controlado, o que torna difícil a recuperação da informação. O metadado sobre as dimensões da peça são armazenados todos no mesmo campo “Dimensões” sem um padrão definido em relação a ordem de escrita da altura, largura e profundidade e não há padrão para unidade de medida utilizada, temos registros em milímetros e em centímetros. Todos os metadados são dos tipos “text” ou “number”, ou seja, aceitam valores textuais (string) ou numéricos, o que torna difícil a inclusão de informações com estruturas mais complexas como “arrays” ou “objects”, como por exemplo as informações sobre o estado de conservação e intervenções dos itens, que são armazenadas nos campo Descrição e Intervenções na aba Conservação da BD Museológica, e que são escritas em linguagem natural, como no exemplo a seguir:

Imagem 8: Exemplo de descrição do estado de conservação de um item da Coleção Ema Klabin.



Identificação		Dados Técnicos		Histórico		CONSERVAÇÃO		Complemento	
<b>M-0021</b>	Classificação	Artes Visuais - Gravura							
	Núcleo	Arte Européia até 1900							
	Nome/Título	<b>VIRGEM COM MENINO (MADONA DA PÊRA)</b>							
Autor	Jerome Wierix (Antuérpia, Bélgica, 1553)				Origem	Flandres, Bélgica			
Data/Período	Século XVI				Localização	Biblioteca			
									
Estado Atual	Bom	Nível		Foto	não	Cromo	não	<b>Ir para Restauração</b>	
Recomend.									
Descrição	1997: restaurado. 11/abr/97: manchas.								
Intervenções	1997: gravura e moldura restaurados por Sophia Haberland e Ateliê Época de restauração. Passepartout de fibra de algodão, Ph neutro; papel filtro neutro e película de terfane; colas neutras, removíveis. Ver documentação escrita. 2004 - Remoção mecânica do 2o. suporte; remoção química da fita adesiva e da cola; remoção dos papéis colados no verso; limpeza mecânica; banhos para remoção de produtos químicos anteriores; suavização localizada das manchas; banhos para remoção do produto e desacidificação; consolidação das laterais com papel japonês; reconstituição do suporte; planificação e montagem articulada. (Ilele Baldini)								
Trânsito									
Criador	Izadora Bertussi	Data	23/07/99	Hora	15:41:16	Modificador	Maria Paula Cruvinel	Data	21/10/2022
								Hora	11:56:06
<b>Fundação Cultural EMA GORDON KLabin</b>									

*Fonte: Aba do estado de conservação de uma peça no Filemaker*

Este metadado poderia ser facilmente representado por uma matriz de objetos, em que o objeto contém as propriedades data, observação, e usuário, em uma arquitetura da informação baseada em eventos, em que o usuário adicionaria novos estados de conservação sempre com a mesma estrutura.

Na imagem anterior também podemos ver os campos de “Criador” e “Modificador” no rodapé da tela, o campo de Modificador também apresenta problemas, pois registra somente o último modificador, ou seja, não existe no sistema uma forma de armazenar todo o histórico de atualizações nas BDs, o que torna impossível uma auditoria dos dados.

Somente os metadados Classificação e Núcleo utilizam vocabulários controlados para a indexação. Em Classificação foi utilizado o Tesouro de Objetos do Patrimônio Cultural nos Museus Brasileiros, de Helena Dodd Ferrez (2016), e Núcleo utiliza uma taxonomia estabelecida internamente pela equipe da casa museu: “Os núcleos foram estabelecidos pela equipe da Fundação e se baseiam no papel que os objetos representam dentro da coleção, independentemente de sua função original, visando facilitar a compreensão do público”

(EXPLORE, 2019). A não utilização de vocabulários controlados em outros metadados como material, técnica ou local de criação torna impossível a recuperação da informação por meio de filtros baseados nesses campos.

Um outro problema nas BDs é que não há uma conexão direta entre elas, quando muito, há a indicação do número de tombo do Arquivístico referente ao documento de compra da obra no campo Doc. Aquisição da BD Museológica, ou indicação em linguagem natural do número de tombo de uma referência. Informações que poderiam ser compartilhadas entre as bases em forma de linguagens documentárias como a localização interna dos itens na casa, a localização geográfica de origem do objeto e a autoria, operam de maneira isolada entre as BDs o que torna difícil a sua manutenção.

Imagem 9: Histórico de uma peça da Coleção Ema Klabin.

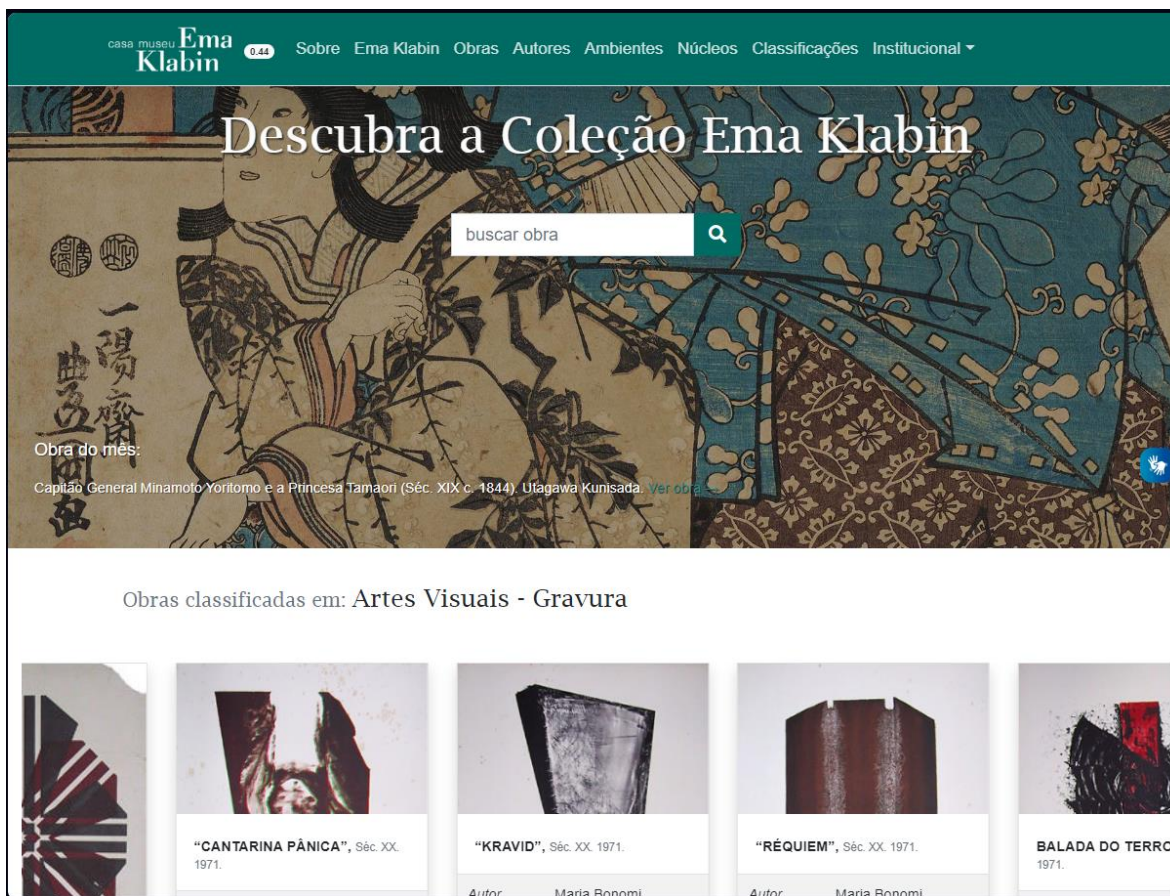
Identificação	Dados Técnicos	HISTÓRICO	Conservação	Complemento
<b>M-0023</b>	Classificação: Artes Visuais - Gravura Núcleo: Arte Europeia até 1900 Nome/Título: <b>JOVEM SENTADO NO CHÃO COM UMA PERNA ESTENDIDA</b>			
Autor: Rembrandt van Rijn (Leiden, Holanda,	Origem: Holanda			
Data/Período: Século XVII - 1646	Localização: Biblioteca			
Loc. Original: Biblioteca. Na parede da direita, logo à entrada.	Número Anterior: 154			
Procedência:				
Data/Entrada: 1947	Doc. Aquisição: A-0020, 0192.0			
Parecer Técnico: 1947, Armando Balloni, Rio de Janeiro (A-0020)				
Exposições: 2004, Pinacoteca do Estado, São Paulo, "Universos Sensíveis".				
Referência Bibliográfica: B-196. Catálogo Geral de Gravuras de Rembrandt. (HARRIS, Bruce & Seena, ed. "The Complete Etchings of Rembrandt". New York, Bounty Books, 1970. N. 196, pág. 148, reprod.)				
Referência Audiovisual:				
Criador: Izadora Bertussi	Data: 24/06/99	Hora: 10:25:26	Modificador: Maria Paula Cruvinel	Data: 21/10/2022
		Hora: 12:04:54		
Fundação Cultural EMA GORDON KLABIN				

Fonte: Aba do histórico de uma peça no Filemaker

Em 2019, com o lançamento do EXPLORE (<https://emaklabin.org.br/explore>), a Casa Museu Ema Klabin passa a disponibilizar a coleção para consulta online. O EXPLORE é resultado do Trabalho de Conclusão de Curso da Pós-Graduação em Comunicação e Design

Digital pela Escola Superior de Propaganda e Marketing (2018) e consiste em duas partes principais: plugin e tema para WordPress. O plugin é responsável por delimitar a arquitetura da Informação no sistema, o *back-end*, já o tema é responsável pela apresentação desta informação, o *front-end*. No *back-end* foi estabelecido dois tipos principais de conteúdos a serem inseridos: Autores e Obras. Também foi estabelecido três taxonomias para organizar as obras: Ambientes, Núcleos e Classificações. Em “Autores” foi inserido todos os metadados relacionados aos agentes criadores dos objetos da coleção. “Obras” armazena os metadados relacionados aos itens da coleção: número de tombo, título, autor, data, material, técnica, medidas, classificação, ambiente e núcleo. Em “ambientes” foi criado um termo para cada ambiente da casa: sala de jantar, galeria, biblioteca, salão, quarto principal, quarto azul, sala de música, hall, vestíbulo, jardim principal e jardim interno, e relacionado à localização das obras na casa. Em “núcleos” estão os termos utilizados pela curadoria para organizar os itens da coleção: artes decorativas, prataria, arte asiática, arte europeia até 1900, arte brasileira - século XX, mobiliário europeu, arte brasileira até 1900, moda, mobiliário brasileiro, arte europeia - século XX, antiguidade clássica, arte pré-colombiana, arte africana e arte das Américas. Em “classificações” foi utilizado o Tesouro de Objetos do Patrimônio Cultural nos Museus Brasileiros (FERREZ, 2016).

Imagem 10: Página principal do Explore.



Fonte: Página inicial do site <https://emaklabin.org.br/explore>

Imagem 11: Página de detalhe de um item do Explore.



Após o lançamento do site foi diagnosticada a necessidade de atualizar as bases de dados para que as informações estejam em padrões internacionais de descrição e representação de itens museológicos, como Object ID, e Linked-art criados pelos Grupos de Trabalhos do Comitê Internacional de Documentação do Conselho Internacional de Museus (CIDOC-ICOM), e que os processos museológicos passassem a aderir metodologias como *Standard Procedures for Collections Recording Used in Museums* (SPECTRUM) criada pela *Collections Trust*, recomendada pelo ICOM e utilizada por todas as instituições museológicas do Reino Unido e diversas outras ao redor do mundo (COLLECTIONS TRUST, 2017).

Tanto o EXPLORE quanto o Filemaker, não são soluções adequadas para a gestão de uma coleção complexa como a Coleção Ema Klabin. Embora o EXPLORE seja uma solução mais recente e desenvolvida internamente, será necessário passar por novos ciclos de desenvolvimento para que possa ser utilizado efetivamente como um *Collection Management System* (CMS).

O Elucidário.art consiste justamente nesta atualização do EXPLORE com foco na gestão da informação museológica. No capítulo seguinte apresentamos o app Elucidário.art e como ele foi desenvolvido para atender as necessidades da Coleção Ema Klabin.

## **7. Elucidario.art**

Em linhas gerais o Elucidário.art é um *Collection Management System*. A abreviação CMS é majoritariamente conhecida com um outro significado: *Content Management System*, devido a popularidade de plataformas como WordPress, Joomla, Drupal, etc. O Elucidário.art é um CMS para coleções de arte, ou seja, é um sistema de gerenciamento de coleções de arte, ou um *Content Management System* especializado. Portanto utilizaremos a abreviação CMS para nos referirmos ao Elucidário como um *Collection Management System*. Este termo também é utilizado por instituições como *Collections Trust* e ICOM para referir-se a esta modalidade de software.

Em suma, o Elucidário.art consiste em um plugin para WordPress que define um conjunto de funcionalidades para gerenciamento de coleções de arte. O plugin utiliza o modelo de



dados para aplicações Linked Art para definição das classes principais de conteúdo e se baseia nos procedimentos Spectrum para definição de seus fluxos de trabalho.

Para o entendimento das seções a seguir, é importante primeiro definirmos alguns conceitos:

**a11y (accessibility):** é a acessibilidade de um sistema, ou seja, a adaptação de um sistema para diferentes necessidades de acessibilidade. O termo a11y é uma abreviação de “*accessibility*” que contém 11 letras entre a letra “a” e a letra “y”.

**Administrador:** é o usuário que gerencia o sistema. É o usuário que tem acesso ao endereço ‘<https://exemplo.com/wp-admin>’ ou qualquer outra URI de gerenciamento do sistema.

***Application Programming Interface (API):*** é um conjunto de rotinas e padrões de programação para acesso a um aplicativo, software ou plataforma de forma programática.

***Back-end:*** é a parte do sistema que não é visível para o usuário. É a parte do sistema que gerencia os dados e a lógica de negócio.

**Bloco Gutenberg:** é uma funcionalidade do WordPress que facilita a reutilização de componentes de interface de usuário. Os blocos Gutenberg são utilizados para construir páginas e postagens no WordPress, e utilizam a biblioteca React para construção das interfaces.

**Componente React:** é um componente de interface de usuário desenvolvido utilizando a biblioteca React (META OPEN SOURCE, 2023). Os componentes são utilizados como blocos de construção no desenvolvimento de interfaces de usuário.

***Continuous Integration (CI):*** é uma prática de desenvolvimento de software onde desenvolvedores integram o código em um repositório compartilhado frequentemente, geralmente várias vezes ao dia. Cada integração é verificada por um processo de compilação automatizado (incluindo testes) para detectar erros de integração o mais rápido possível.

**Docker** (HYKES, 2013): é uma plataforma de código aberto para desenvolvimento, envio e execução de aplicativos. O Docker permite que os desenvolvedores empacotem um aplicativo com todas as partes de que ele precisa, como bibliotecas e outras dependências, e enviem tudo como um pacote. Ao fazer isso, graças à padronização, o Docker permite que o aplicativo seja executado em qualquer lugar, seja em um notebook, em um servidor físico, em uma máquina virtual ou em um ambiente em nuvem.

**Endpoint:** é um ponto de acesso de uma API, ou seja, uma URI que define um recurso e suas operações, por exemplo: “<https://exemplo.com/wp-json/lcdr/v1/objetos>”, em que /lcdr/v1/objetos é o *endpoint* para o recurso “objetos” da API.

**Front-end:** é o conjunto de tecnologias e ferramentas utilizadas para construção da interface de usuário. É a parte do sistema que é visível para o usuário, esta diretamente relacionada a *User Interface* (UI).

**Headless:** é utilizado para referir-se a sistemas que não possuem uma interface de usuário final (UI), mas possuem uma API Rest para leitura e escrita. Um sistema *headless* pode ser utilizado como *back-end* para um sistema *front-end*, e sua API pode ser utilizada para construção de interfaces de usuário em diferentes plataformas, como aplicações web, mobile, etc.

**i18n (internationalization):** é a internacionalização de um sistema, ou seja, a adaptação de um sistema para diferentes idiomas e culturas. O termo i18n é uma abreviação de “*internationalization*” que contém 18 letras entre a letra “i” e a letra “n”.

**i10n (localization):** é a localização de um sistema, ou seja, a adaptação de um sistema para uma cultura e idioma específico. O termo i10n é uma abreviação de “*localization*” que contém 10 letras entre a letra “i” e a letra “n”.

**Localhost:** é o endereço local do sistema, geralmente é acessado através do endereço “<http://localhost:8000>”.

**Pacotes:** no contexto de desenvolvimento de software, “pacote” é um termo utilizado para definir um conjunto de arquivos e diretórios que contém código-fonte, configurações, testes, arquivos, etc. Um pacote pode ser um plugin, um tema, uma biblioteca, um módulo,



entre outras coisas. No nosso contexto, os pacotes JavaScript utilizam o *Node Package Manager* (NPM) e o *Composer* quando há arquivos PHP no pacote para gerenciamento de dependências, e, dependendo do pacote, podem ser utilizados em diferentes ambientes, como no navegador, no Node.js, no React Native, etc. Neste contexto de pacotes JavaScript, cada pacote segue estruturas de pastas contendo um arquivo “package.json” que define as suas configurações, como nome, versão, dependências, *scripts*, etc. Este mesmo conceito também se aplica para o PHP, onde cada pacote segue estruturas de pastas contendo um arquivo “composer.json” que, assim como o “package.json”, define as suas configurações, como nome, versão, dependências, *scripts*, etc.

**User Interface (UI):** é a interface do usuário, ou seja, a parte do sistema que é visível e possibilita a interação com o usuário, está diretamente relacionada ao *front-end*.

**Representational State Transfer API (REST-API):** é um conjunto de padrões de comunicação hipermídia entre sistemas que utiliza o protocolo HTTP para realizar requisições e respostas.

**Rota (Route):** é um endereço de um recurso em um sistema. Por exemplo, o endereço “<https://exemplo.com>” é uma rota para o recurso “home” do sistema.

**Usuário final:** é o usuário que interage com o sistema. É o usuário que acessa o endereço ‘<https://exemplo.com>’.

**WAI-ARIA:** é um conjunto de atributos que podem ser adicionados a tags HTML para melhorar a acessibilidade de um sistema. O termo WAI-ARIA é uma abreviação de “*Web Accessibility Initiative - Accessible Rich Internet Applications*” (W3C, 2022).

## 7.1. O Repositório

As principais linguagens de programação utilizadas no código-fonte do Elucidário.art são PHP 8.2 (PHP, 2023), TypeScript (MICROSOFT, 2023), JavaScript (MDN, 2022) e JSON (BRAY; IETF, 2017), e, estão estruturadas em um repositório utilizando a arquitetura *monorepo*, ou seja, um repositório que contém múltiplos projetos (NARWHAL TECHNOLOGIES INC., 2022). Optamos por utilizar esta arquitetura para facilitar o desenvolvimento e a manutenção do código-fonte, pois desta forma podemos reutilizar

pacotes em diferentes projetos, como em bibliotecas, plugins, temas, etc. Esta arquitetura também pode ser chamada de micro-serviços (*microservices*), onde cada pacote é um serviço que pode ser desenvolvido, testado e publicado de maneira independente, o que facilita a manutenção e estimula a evolução independente de cada pacote.

Utilizamos o Git (TORVALDS; HAMANO, 2005) para o controle de versão dos pacotes, o GitHub (GITHUB, 2008) para hospedagem do repositório e o Versionamento Semântico (*Semantic Versioning* ou *SemVer*) (SEMVER, 2023) para definição das versões públicas dos pacotes. O SemVer define três tipos diferentes de incrementação de versão: *major*, *minor* e *patch*. O *major* é incrementado quando há mudanças incompatíveis na API, ou seja, quando a forma de usar as suas funções, métodos, classes, e etc, são alteradas; o *minor* é incrementado quando há adição de funcionalidades compatíveis com versões anteriores; e o *patch* é incrementado quando há correção de *bugs*, também compatíveis com versões anteriores.

**Figura 3: Exemplo do Versionamento Semântico (SemVer)**



1.4.2  
major minor patch

*Fonte: Elaborado pelo autor, com base no SemVer.org*

Para assegurarmos a qualidade do código-fonte, e se seus objetivos estão sendo atingidos, criamos um conjunto de testes automatizados para cada um dos pacotes, configurados especificamente para cada linguagem de programação e ambiente. Estes testes rodam automaticamente no ambiente local a cada nova alteração no código-fonte adicionada ao controle de versão utilizando a biblioteca Husky (TYPICODE, 2018), e em um ambiente de integração contínua (CI) (sincronização remota) utilizando o GitHub Actions (GITHUB, 2018).

O repositório pode ser acessado no link <https://github.com/hgodinho/elucidario> e é organizado da seguinte forma:

```
elucidario
├── packages
├── apps
├── publications
└── references
```

O diretório “*packages*” contém os pacotes que podem ser reutilizados, tanto por outros pacotes, como por aplicações. Todos os pacotes definidos nesta pasta seguem o padrão de nome “@elucidario/pkg-<nome-pacote>”.

No diretório “*apps*”, se encontram as aplicações, como um ambiente de desenvolvimento completo utilizando Docker para testes locais e o site da documentação disponível em <http://elucidario.art/doc>. Os pacotes nesta pasta seguem o padrão de nome “@elucidario/app-<nome-pacote>”.

O diretório “*publications*” contém as publicações referentes ao Elucidário.art, como a dissertação de mestrado e outros artigos desenvolvidos ao longo da pesquisa. Os pacotes nesta pasta seguem o padrão de nome “@elucidario/pub-<nome-pacote>”.

No diretório “*references*” contém referências utilizadas no desenvolvimento de todo o ecossistema do Elucidário.art e estão organizadas em um arquivo JSON para cada referência seguindo o formato *Citation Style Language* (CSL) (D’ARCUS, 2010).

Todos os pacotes nos diretórios apps e packages foram construídos levando em conta os seguintes princípios de design, ou técnicas de programação:

- a11y (*accessibility*) - quando aplicável o pacote deve seguir as regras de acessibilidade apropriadas para o contexto;
- i10n (*localization*) - quando aplicável o pacote deve implementar o suporte a localização dos idiomas português, espanhol e inglês, seguindo esta ordem de prioridade;
- i18n (*internationalization*) - quando aplicável o pacote deve implementar o suporte a internacionalização, e o processo de localização deve ser devidamente documentado;

Os pacotes principais do diretório *packages* são:

- “@elucidario/pkg-mdorim” - Pacote que define o modelo de dados utilizado no sistema;
- “@elucidario/pkg-design-system” - Pacote que define o sistema de design utilizado no aplicativo;
- “@elucidario/pkg-core” - Pacote principal do Elucidário.art, integra os demais pacotes e define as funcionalidades principais do sistema, como banco de dados, rotas, páginas administrativas, configurações, usuários, permissões de usuários, etc;

A seguir apresentamos os principais pacotes individualmente, descrevendo-os em três partes:

1. Escopo de uso do pacote;
2. Descrição de suas funcionalidades, implementação e estrutura. As descrições contém diagramas UML e tabelas para melhor representação de seu conteúdo;
3. Como os testes foram executados e seus resultados para garantir a qualidade de cada pacote.

### **9.1.1 @elucidario/pkg-mdorim**

#### **Escopo**

O Modelo de Dados para Organização e Representação da Informação Museológica (Mdorim) foi desenvolvido para ser utilizado pelo plugin Elucidário.art como modelo de dados para armazenamento, tratamento e transferência de dados em suas operações com foco na documentação museológica. O modelo utiliza como base principal o Linked Art e os procedimentos Spectrum para definição das entidades e fluxos de trabalhos utilizados no sistema. O modelo estende a estrutura de usuários padrão do WordPress e apresenta novos tipos e capacidades, além de entidades exclusivas para o registro de configurações do sistema, histórico de edições e duas interfaces: uma para mapeamento e outra para tradução dos metadados do modelo.

#### **Descrição**

O Mdorim é composto por três partes principais:

1. Modelo: Modelo de dados para representação das entidades e fluxos de trabalho baseados nos contextos de uso do sistema: leitura e escrita;
2. MySQL Schema: Esquemas de tabelas MySQL para armazenamento dos dados;
3. Interfaces: Interface para tradução dos rótulos e descrições dos metadados do modelo para português com habilitação da tradução para outros idiomas e interface de mapeamento dos dados em Linked Art para outros modelos.

O modelo de dados é composto por um conjunto de metadados que definem as entidades e fluxos de trabalho utilizados pelo sistema. Os metadados são definidos utilizando o JSON-Schema (DROETTBOOM, 2013), que permite a definição e validação de tipos de dados e suas propriedades.

Um esquema em JSON-Schema pode ser definido da seguinte forma:

**Figura 4: Exemplo de JSON-Schema descrevendo os metadados para validação da representação de uma pessoa**

```
{
  "$schema": "http://json-schema.org/draft-04/schema",
  "title": "Person",
  "type": "object",
  "properties": {
    "name": {
      "type": "string"
    },
    "age": {
      "description": "Age in years",
      "type": "integer",
      "minimum": 0
    }
  },
  "required": ["name"],
  "additionalProperties": false
}
```

**Fonte:** Elaborado pelo autor.

Tudo que está posicionado à esquerda dos dois pontos “:”, é uma propriedade do objeto, também chamada de chave (*key*); e, tudo que está posicionado à direita é o valor da propriedade. No exemplo acima, a chave `$schema` define a versão do JSON-Schema utilizado, em nosso modelo utilizamos a versão `draft-4` por ser a versão utilizada pelo

WordPress. As propriedades `title` e `type` define o título e o tipo de dado definido no esquema, como no exemplo estamos definindo um tipo `Person`, o valor de `type` é `object`. Em `properties` definimos as propriedades, em que cada uma é definida por um outro tipo de dado: `name` é uma `string` e `age` é uma `integer`. A chave `required` define as que são obrigatórias, em que `name` é obrigatório e `age`, como não está definido, é opcional. `additionalProperties` define se o objeto pode ter propriedades adicionais que não foram definidas previamente no esquema, neste caso definimos como `false`, ou seja, não pode ter propriedades adicionais.

Uma pessoa pode ter sua representação validada utilizando este esquema da seguinte forma:

**Figura 5: Exemplo de JSON representando uma pessoa**

```
{
  "name": "Henrique Godinho",
  "age": 33
}
```

**Fonte:** Elaborado pelo autor.

O JSON-Schema também apresenta propriedades que permitem a reutilização de definições. A chave `definitions` pode ser utilizada para descrever diversos metadados diferentes, enquanto a chave `$ref` permite a referência por meio de uma URI à um metadado definido previamente. No exemplo abaixo, definimos um esquema primário que contém as definições dos metadados utilizados no sistema, e em seguida definimos um tipo `Person` que utiliza a propriedade `$ref` para referenciar suas propriedades `name` e `age`:

**Figura 6: Exemplo de JSON-Schema contendo a propriedade “definitions”**

```
{
  "$schema": "http://json-schema.org/draft-04/schema",
  "$id": "http://example.com/definitions.json",
  "definitions": {
    "name": {
      "type": "string"
    },
    "age": {
      "type": "integer",
      "minimum": 0
    }
  }
}
```

```
}  
}
```

**Fonte:** Elaborado pelo autor.

**Figura 7: Exemplo de JSON-Schema contendo a propriedade “\$ref” se referenciando ao exemplo anterior**

```
{  
  "$schema": "http://json-schema.org/draft-04/schema",  
  "$id": "http://example.com/person.json",  
  "title": "Person",  
  "type": "object",  
  "properties": {  
    "name": {  
      "$ref":  
"http://example.com/definitions.json#/definitions/name"  
    },  
    "age": {  
      "$ref":  
"http://example.com/definitions.json#/definitions/age"  
    }  
  }  
}
```

**Fonte:** Elaborado pelo autor.

Perceba que agora em cada JSON também utilizamos a propriedade \$id como uma URI que identifica o esquema definido. Dessa forma conseguimos realizar a referência entre os esquemas por meio de suas URIs o que garante uma maior consistência das estruturas dos dados e evita a duplicação de definições.

Como vimos anteriormente, o Mdorim se baseia no Linked Art, portanto utilizamos suas entidades e propriedades como base para a definição do modelo para uso no aplicativo Elucidário.art. As nomenclaturas do modelo, como nome de entidades, objetos e propriedades foram mantidas em inglês para manter a consistência entre o idioma das linguagens de programação e a API do modelo. O modelo é dividido em quatro partes e dois contextos.

1. **Entidades:** Definição das entidades utilizadas no modelo (Linked Art);
2. **Procedimentos:** Definição dos fluxos de trabalho utilizados no modelo (Spectrum);

3. **Histórico de edições:** Definição do histórico de edições das propriedades e relações das entidades;
4. **Configurações:** Definição das configurações do sistema;

Seguindo o 5º princípio do Linked Art: “*Use REST / Don’t break the web*” (LINKED ART, 2021), o modelo é dividido em dois contextos: leitura e escrita, que se referem ao tipo de requisição feita à API do modelo. No contexto de leitura, as requisições HTTP são feitas utilizando o método GET e no contexto de escrita, as requisições são feitas utilizando os métodos POST, PUT ou DELETE (criação, edição e remoção respectivamente).

O Linked Art expõe somente o método GET (LINKED ART, 2021), ou seja, somente leitura, mas o Elucidário.art adiciona os outros métodos para obtermos uma interação completa com o modelo.

No contexto de leitura os dados podem ser retornados em dois formatos diferentes, o primeiro segue o formato do WordPress, em que as entidades relacionadas são definidas por meio de IDs e uma propriedade `_links` contendo um array com a URI da entidades relacionadas e/ou uma propriedade `_embedded` que contém todos dados de uma entidade relacionada (WORDPRESS, [s.d.]). O formato do WordPress é o padrão da REST-API do Elucidário.art, portanto não é necessário adicionar o cabeçalho `Accept` na requisição. O segundo segue o formato JSON-LD utilizando o perfil do Linked Art como modelo de interoperabilidade, isso significa que as relações com outras entidades estão definidas por meio de um objeto `Ref` que contém a URI, um rótulo e o tipo da entidade relacionada. A requisição feita ao *endpoint* deve conter um cabeçalho `Accept` com o valor `application/ld+json;profile="https://linked.art/ns/v1/linked-art.json"` (LINKED ART, 2021).

**Figura 8: Resposta a uma requisição contendo o metadado `classified_as` da entidade `Object` no contexto de leitura do WordPress utilizando somente a propriedade `_links`**

```
{
  "classified_as": [1, 2, 3],
  "_links": {
    "concepts": [
      {
        "id": 1,
```



```

        "href": "http://exemplo.com/wp-json/lcdr/v1/concepts/1"
      },
      {
        "id": 2,
        "href": "http://exemplo.com/wp-json/lcdr/v1/concepts/2"
      },
      {
        "id": 3,
        "href": "http://exemplo.com/wp-json/lcdr/v1/concepts/3"
      }
    ]
  }
}

```

**Fonte:** Elaborado pelo autor.

Neste contexto, o Elucidário.art retorna uma lista de ID das entidades Concepts registradas. Para cada ID é possível acessar a URI da entidade relacionada por meio da propriedade `_links`.

**Figura 9: Resposta a uma requisição contendo o metadado `classified_as` da entidade Object no contexto de leitura do content-type WordPress utilizando as propriedades `_links` e `_embedded`**

```

{
  "classified_as": [2],
  "_links": {
    "concepts": [
      {
        "id": 2,
        "href": "http://exemplo.com/wp-json/lcdr/v1/concepts/2",
        "embeddable": true
      }
    ]
  },
  "_embedded": {
    "concept": {
      "entity_id": 2,
      "name": "name",
      "guid": "936DA01F-9ABD-4d9d-80C7-02AF85C822A8",
      "author": 1,
      "status": "publish",
      "created": "2023-01-01 00:00:00",
      "modified": "2023-01-01 00:00:00",
      "type": "Concept",
      "label": "Conceito de exemplo",
      "identified_by": [

```

```

    {
      "type": "Name",
      "content": "Conceito de Exemplo"
    }
  ]
}
}
}

```

**Fonte:** Elaborado pelo autor.

Quando um objeto da propriedade link for marcado como embeddable, o Elucidário.art retorna os dados da entidade relacionada na mesma resposta da requisição, na propriedade \_embedded.

**Figura 10: Resposta a uma requisição contendo o metadado classified\_as da entidade Object no contexto de leitura do content-type Linked Art**

```

{
  "classified_as": [
    {
      "id": "http://exemplo.com/wp-json/lcdr/v1/concepts/1",
      "type": "Concept",
      "_label": "Exemplo 1"
    },
    {
      "id": "http://exemplo.com/wp-json/lcdr/v1/concepts/2",
      "type": "Concept",
      "_label": "Exemplo 2"
    },
    {
      "id": "http://exemplo.com/wp-json/lcdr/v1/concepts/3",
      "type": "Concept",
      "_label": "Exemplo 3"
    }
  ]
}

```

**Fonte:** Elaborado pelo autor.

A resposta da requisição no formato Linked Art retorna um array do objeto Ref que contém, cada um dos elementos, a URI, a label e o tipo da entidade relacionada.

Já no contexto de escrita a definição das relações são definidas pelas IDs das entidades:

**Figura 11: Requisição de escrita contendo o metadado `classified_as` da entidade `Object` no content-type `WordPress`**

```
{
  "classified_as": [1, 2, 3, 4]
}
```

**Fonte:** Elaborado pelo autor.

Neste exemplo, a requisição de escrita para o metadado `classified_as` contém apenas um array de IDs numéricas que representam as ID de cada `Concept` sendo referenciado.

O modelo estende o sistema de criação de usuários do WordPress e apresenta quatro novos tipos de usuários além do Admin (`admin`) e Público (`public`), são eles: Curadoria (`curator`), Museologia (`museologist`), Assistência (`assistant`) e Pesquisa (`researcher`), o modelo também introduz quatro novos grupos de capacidades de usuários, sendo eles: `entities`, `procedures`, `mapping` e `options`. Cada grupo de capacidade é composto por uma lista de capacidades para operações CRUD (*Create, Read, Update and Delete*) de cada entidade, procedimento, mapeamento e configuração do sistema, respectivamente. O quadro a seguir apresenta os tipos de usuário e os grupos de capacidades de cada um:

**Quadro 49: Tipos de usuário e capacidades por tipo de usuário.**

Capacidades	Admin	Curadoria	Museologia	Assistência	Pesquisa	Público
sistema	sim	-	-	-	-	-
create_users, delete_users, list_users	sim	sim	sim	-	-	-
see	todos	todos	todos	todos	todos	entities, mapping
create	todos	todos	todos	entities, procedures, mapping	entities, mapping	-
edit_own	todos	todos	todos	entities, procedures, mapping	entities, mapping	-
edit_others	todos	todos	todos	entities, procedures, mapping	entities, mapping	-

Capacidades	Admin	Curadoria	Museologia	Assistência	Pesquisa	Público
edit_published	todos	todos	todos	entities	-	-
delete_own	todos	todos	todos	entities, procedures, mapping	-	-
delete_others	todos	todos	todos	-	-	-
publish	todos	todos	todos	-	-	-

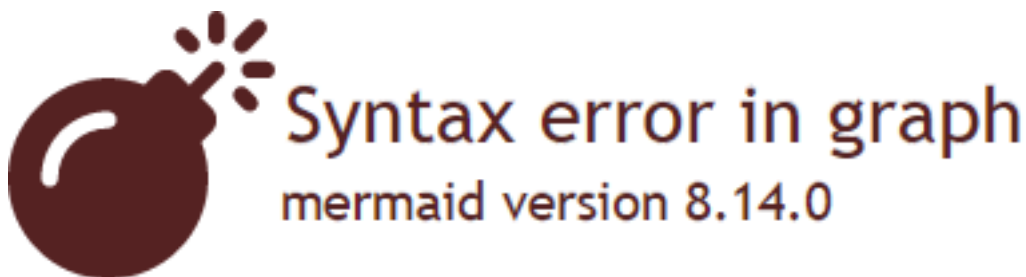
**Fonte:** Elaborado pelo autor.

As capacidades no modelo são definidas concatenando o tipo de capacidade com o grupo a que pertence, por exemplo: `create_entity` para criar uma entidade, `see_procedure` para ler um procedimento, `edit_mapping` para editar um mapeamento e `delete_option` para deletar uma configuração do sistema.

O usuário `admin` tem acesso total ao sistema e tem todas as capacidades de administrador do WordPress, como adicionar e remover plugins—inclusive o Elucidário.art, portanto somente este usuário tem este nível de acesso—, modificar configurações de domínio, estrutura de links e entre outras. Os tipos `curator` e `museologist` são usuários com acesso total ao modelo, mas somente a capacidade de criar, deletar e listar usuários é herdada do WordPress. Já `assistant` pode ver, criar, editar a própria, editar o de outro usuário, editar publicado e deletar o próprio, enquanto o `researcher` pode somente ver, criar, editar o próprio e editar o de outros. O último tipo de usuário é o `public` e tem somente acesso de leitura aos dados, portanto os usuários que tem algum tipo de permissão superior a `public` precisam estar autenticados e ter a devida autorização para realizar qualquer ação além de ler os dados que tem acesso. O usuário `public` não precisa ser realmente definido no sistema, mas

A seguir apresentamos uma visão geral do modelo com todas as entidades, procedimentos, opções, mapeamentos e usuários. A figura 12 foi gerada utilizando a ferramenta Mermaid (SVEIDQVIST; VINOD, [s.d.]).

### **Figura 12: Mapa mental do Mdorim**



**Fonte:** Elaborado pelo autor.

A figura 12 demonstra o Mdorim como um todo, em cinza vemos as entidades principais do modelo que foram baseadas no Linked Art, e ainda no cinza e com formato circular vemos os procedimentos que também tem sua rota no modelo. Em preto vemos a entidade Option e suas possibilidades de uso. Em marrom os usuários e os níveis de permissão, em vermelho a entidade History, por fim, em roxo o mapeamentos e suas propriedades.

A URI para realizar uma requisição a API do modelo segue a mesma lógica do WordPress (WORDPRESS, [s.d.]), utilizamos a rota principal /wp-json/ e adicionamos uma nova rota /lcdr/v1/ para identificar a API do Elucidário.art, em que lcdr é uma abreviação e v1 é a versão da API, ficando com a rota principal do modelo em:

{protocolo}://{domínio}/wp-json/lcdr/v1/. O protocolo pode ser tanto http quanto https, embora seja recomendado o uso de https para garantir a segurança das informações. O domínio é o endereço do site, por exemplo: elucidario.art ou emaklabin.org.br, e as rotas definem as entidades e as ações que podem ser realizadas, por exemplo, /lcdr/v1/objects para obter todos os Objects armazenados no sistema.

As rotas finais são definidas pelo idioma selecionado nas configurações do WordPress, por exemplo, a entidade Concept com o WordPress configurado em português teria a rota lcdr/v1/conceitos para requisições de todos os conceitos armazenados no sistema e lcdr/v1/conceitos/{id} para requisições de um conceito específico. A tabela a seguir apresenta as rotas definidas para cada entidade do modelo, juntamente com os métodos HTTP e uma descrição.

**Quadro 50: Endpoints do modelo.**

Entidade	Rotas	Métodos	Descrição
Concept,	/lcdr/v1/{entidade}	GET,	Coleção de entidades Linked

Entidade	Rotas	Métodos	Descrição
Digital, Event, Group, Object, People, Place, Provenance, Set, Textual, Visual		POST, PUT, DELETE	Art. Substitua {entidade} pela nome da entidade no idioma configurado.
Concept, Digital, Event, Group, Object, People, Place, Provenance, Set, Textual, Visual	/lcdr/v1/{entidade}/{id}	GET, POST, PUT, DELETE	Entidade Linked Art específica. Substitua {entidade} pela nome da entidade no idioma configurado.
Procedure	/lcdr/v1/procedimentos	GET, POST, PUT, DELETE	Coleção de procedimentos. A string <code>procedimentos</code> pode ser alterada para o idioma configurado.
Procedure	/lcdr/v1/procedimentos/{id}	GET, POST, PUT, DELETE	Procedimento específico. A string <code>procedimentos</code> pode ser alterada para o idioma configurado.
Mapping	/lcdr/v1/mapeamentos	GET, POST, PUT, DELETE	Coleção de mapeamentos. A string <code>mapeamentos</code> pode ser alterada para o idioma configurado.
Mapping	/lcdr/v1/mapeamentos/{id}	GET, POST, PUT, DELETE	Mapeamento específico. A string <code>mapeamentos</code> pode ser alterada para o idioma configurado.
Option	/lcdr/v1/opcao/{name}	GET, POST, PUT, DELETE	Opção específica do sistema. A string <code>opcao</code> pode ser alterada para o idioma configurado.
History	/lcdr/v1/{entidade}/{id}/edicoes	GET	Histórico de edições de uma entidade. Substitua {entidade} pelo tipo de entidade no idioma configurado, e {id} para a ID da entidade. A string <code>edicoes</code> pode ser alterada para o idioma configurado.

**Fonte:** Elaborado pelo autor.

As entidades principais do modelo seguem as mesmas do Linked Art (Ver capítulo 6), portanto possuem todas suas propriedades e mais as descritas a seguir. São elas: Concept, Digital, Event, Provenance, Group, Person, Object, Place, Set, Textual e Visual.

**Quadro 51: Propriedades básicas de uma entidade Linked Art no Mdorim**

Nome	Tipo	Exigido	Descrição
entity_id	int	sim	ID da entidade
name	string	sim	Nome da entidade
guid	string	sim	Identificador Global Exclusivo (GUID) da entidade
author	int	sim	ID do autor da entidade
status	string	sim	Status da entidade no sistema
password	string	não	Senha para acesso à entidade, utilizada pela API
created	datetime	sim	Data de criação da entidade
modified	datetime	sim	Data de modificação da entidade
history	History	sim	Histórico de edição da entidade

**Fonte:** Elaborado pelo autor.

Também introduzimos uma entidade chamada *Option*, que representa uma opção do sistema. Esta entidade possui uma ID, um nome, um valor e um esquema em JSON-Schema para validação. As opções são utilizadas para definir as configurações do sistema, como por exemplo, o idioma padrão, o número de itens por página, etc. Cada opção tem seu esquema definido no código-fonte e vai variar de acordo com o tipo de dado que ela representa.

**Quadro 52: Entidade *Option* do modelo.**

Nome	Tipo	Exigido	Descrição
id	int	sim	ID da opção
name	string	sim	Nome da opção
value	string	sim	Valor da opção em formato string
schema	object	sim	Esquema (JSON-Schema) para validação da opção inserida

**Fonte:** Elaborado pelo autor.

Já os procedimentos possíveis entre as entidades são baseados no padrão Spectrum, em que cada procedimento é definido por uma série de etapas para documentar determinado evento a um objeto. Os procedimentos de Planejamento da documentação e Catalogação foram omitidos nesta versão do modelo, pois acreditamos que correspondem ao Elucidário.art como um todo e que sua representação pode se tornar bastante complexa. Uma opção futura para estes procedimentos seria explorar visualizações de dados em que poderíamos

representar o fluxo de trabalho de cada procedimento, por exemplo, utilizando um diagrama de Gantt.

#### Quadro 53: Entidade Procedure do modelo.

Nome	Tipo	Exigido	Descrição
id	int	sim	ID do procedimento
type	string	sim	Tipo de procedimento
description	string	não	Descrição do procedimento
author	int	sim	ID do usuário que criou o procedimento
created	datetime	sim	Data de criação do procedimento
modified	datetime	sim	Data de atualização do procedimento
status	string	sim	Status do procedimento, pode ser <b>draft</b> , <b>active</b> , <b>inactive</b> , <b>pending</b> ou <b>scheduled</b>
related_entities	array	não	Entidades relacionadas
schedule	object	não	Objeto Schedule que define o agendamento do procedimento
history	History	sim	Histórico de edição da entidade

**Fonte:** Elaborado pelo autor.

O histórico de edições do modelo consiste em uma entidade chamada History que intercepta as ações de criação, edição e remoção de entidades e registra informações que permitem a auditoria e recuperação de informações adicionadas anteriormente no caso de edições erradas. Uma entidade sempre conterá um objeto History descrevendo seu histórico de edições.

Diferentemente das outras entidades, History somente expõe o método GET para leitura, não sendo possível criar, editar ou remover um History, ou seja, o usuário nunca manipulará diretamente esta entidade, mas sim como consequência de uma ação realizada no sistema de forma programática.

O History é composto por uma série de objetos HistoryEvent que representam cada uma das ações realizadas na entidade. Cada HistoryEvent contém a data e hora da ação, o tipo de ação, o usuário que realizou a ação, o ID da entidade e o conteúdo da propriedade antes e depois da ação. O conteúdo é armazenado em formato JSON e é utilizado para recuperar o estado anterior da entidade.

#### Quadro 54: Entidade History do modelo.



Nome	Tipo	Exigido	Descrição
events	array	sim	Lista de <code>HistoryEvent</code>

**Fonte:** Elaborado pelo autor.

#### Quadro 55: Objeto `HistoryEvent` do modelo.

Nome	Tipo	Exigido	Descrição
id	int	sim	ID do evento de edição
timestamp	date	sim	Data e hora do evento
type	string	sim	Tipo do evento, podendo ser <code>Creation</code> ou <code>Edition</code>
user	string	sim	Usuário que realizou a edição
entity	int	sim	ID da entidade que passou pela edição
property	string	sim	Nome da propriedade que passou pela edição
related_event	int	não	ID de um evento de edição relacionado
current	any	sim	Valor atual da propriedade
previous	any	sim	Valor anterior da propriedade

**Fonte:** Elaborado pelo autor.

A interface de mapeamento também é baseada em JSON-Schema, com o intuito de oferecer ao usuário uma maneira de criar seus próprios mapeamentos dentro do Elucidário.art. A interface é utilizada para as funções de importação e exportação de dado, e também oferecem mais contexto à UI dando uma referência ao usuário de possíveis mapeamentos pertencentes ao campo que está preenchendo.

#### Quadro 56: Entidade `Mapping` do modelo.

Nome	Tipo	Exigido	Descrição
mapping_id	int	sim	ID do mapeamento.
name	string	sim	Nome do mapeamento em <i>kebab-case</i>
title	string	sim	Título do mapeamento
description	string	-	Descrição do mapeamento
author	int	sim	ID do autor do mapeamento
standard	string	sim	Nome do modelo que será mapeado
standard_uri	string	sim	URI do modelo que será mapeado
version	string	sim	Versão do mapeamento
created	datetime	sim	Data de criação do mapeamento
modified	datetime	sim	Data de atualização do mapeamento
mapping	<code>PropMap[ ]</code>	sim	Propriedades mapeadas

**Fonte:** Elaborado pelo autor.

**Quadro 57: Entidade PropMap do modelo.**

Nome	Tipo	Exigido	Descrição
map_id	int	sim	ID do mapeamento de uma propriedade.
description	string	-	Descrição do mapeamento
prop_name	string	sim	Nome da propriedade alvo
entity_type	string	sim	Tipo da propriedade alvo
external_prop_name	string	sim	Nome da propriedade de origem
external_prop_uri	string	sim	URI da propriedade de origem
external_prop_type	string	sim	Tipo de dado da propriedade de origem
map_value	string boolean integer array object	sim	Valor mapeados da propriedade
editable	boolean	não	Se a propriedade pode ser editada na UI do sistema
status	string	sim	Status do mapeamento, pode ser <b>active</b> ou <b>inactive</b>

**Fonte:** Elaborado pelo autor.

A propriedade `map_value` do objeto `PropMap` registra um valor padrão que pode ser utilizado para preencher automaticamente os campos de informação no momento da criação de uma entidade, por exemplo, se estamos tentando representar o metadado `title` do Dublin Core utilizando a propriedade `identified_by` do Linked Art, podemos definir um valor padrão para `classified_as` no objeto `Identifier` com a URI `http://purl.org/dc/elements/1.1/title` do Dublin Core e `aat:300417209` (*full titles*) no AAT:

```
{
  "identified_by": [
    {
      "type": "Identifier",
      "classified_as": [
        {
          "id": "http://purl.org/dc/elements/1.1/title",
          "_label": "Title",
          "type": "Type"
        },
        {
          "id": "http://vocab.getty.edu/aat/300417209",
          "_label": "Full titles",
          "type": "Type"
        }
      ]
    }
  ]
}
```

```
}  
  ]  
}  
]  
}
```

No Linked Art, a propriedade `identified_by` deve receber obrigatoriamente um valor para `content`, mas como no nosso mapeamento estamos criando valores pré-preenchidos, deixamos que o usuário preencha o restante das informações e usamos os valores preenchidos previamente para popular a UI.

Para armazenarmos todas as informações das entidades no sistema, criamos 8 novas tabelas no banco de dados MySQL do WordPress, que dá suporte a diferentes tipos de dados, como numéricos, datas e horas, textos, tipos espaciais e JSON (MYSQL, 2023). Optamos por criar novas tabelas para o Elucidário.art ao invés de usarmos as tabelas padrão do WordPress para isolarmos o seu funcionamento no sistema e diminuir as possibilidades de interferência com outros plugins e temas que possam ser instalados na mesma instância.

As tabelas foram definidas a partir das entidades, propriedades e relações do Mdorim. Levamos em conta as estruturas de seus metadados e como se dão as relações e repetições ao longo do modelo, desta forma optamos por criar uma única tabela para armazenar as entidades principais em uma estrutura polimórfica, em que cada linha da tabela pode ser uma entidade diferente. Esta decisão foi tomada para evitar uma arquitetura mais complexa, com uma tabela para cada entidade e mais uma tabela para cada relação possível entre as entidades do Linked Art, o que poderia acarretar em inúmeras classes e linhas de código extras, aumentando a dificuldade de manutenção, a possibilidade de erros, e possivelmente diminuindo a performance do sistema.

Dividimos as propriedades do Linked Art em dois tipos de acordo com suas características, os metadados que definem relações entre entidades e os metadados que definem um valor para uma entidade. As propriedades que definem uma relação entre entidades são armazenadas em uma tabela separada, em que cada linha representa uma relação entre duas entidades com um predicado específico, semelhante a uma tripla RDF: sujeito, predicado e objeto — em que nas colunas sujeito e objeto armazenamos as IDs das entidades relacionadas, e em predicado, armazenamos o nome da propriedade do Linked Art a que

esta relação pertence, como por exemplo: `classified_as`, `representations`, `took_place_at`, entre outras. As propriedades que definem um valor para uma entidade são armazenadas na própria tabela da entidade na coluna correspondente ao nome da propriedade, por exemplo: `identified_by`, `dimension`, `formed_by`, entre outras. Também criamos alguns metadados exclusivos para o devido funcionamento do sistema. No quadro a seguir listamos todas as propriedades em cada um dos três tipos:

**Quadro 58: Propriedades do Mdorim separadas por tipo.**

Tipo	Propriedades
<b>Internas:</b> propriedades utilizadas internamente pelo sistema para seu devido funcionamento	<code>entity_id</code> , <code>name</code> , <code>guid</code> , <code>author</code> , <code>status</code> , <code>password</code> , <code>created</code> , <code>modified</code> , <code>history</code>
<b>Valor:</b> propriedades do Linked Art utilizadas como colunas que armazenam valores de texto, números, datas, objetos, arrays, etc no Mdorim	<code>type</code> , <code>label</code> , <code>identified_by</code> , <code>referred_to_by</code> , <code>equivalent</code> , <code>attributed_by</code> , <code>dimension</code> , <code>digitally_available_via</code> , <code>created_by</code> , <code>contact_point</code> , <code>begin_of_existence</code> , <code>end_of_existence</code> , <code>timespan</code> , <code>part</code> , <code>produced_by</code> , <code>destroyed_by</code> , <code>removed_by</code>
<b>Relacionamento:</b> propriedades do Linked Art utilizadas como predicados no relacionamento entre entidades no Mdorim	<code>classified_as</code> , <code>representation</code> , <code>member_of</code> , <code>subject_of</code> , <code>part_of</code> , <code>conforms_to</code> , <code>access_point</code> , <code>digitally_carries</code> , <code>digitally_shows</code> , <code>used_for</code> , <code>carried_out</code> , <code>residence</code> , <code>took_place_at</code> , <code>caused_by</code> , <code>carried_out_by</code> , <code>used_specific_object</code> , <code>influenced_by</code> , <code>technique</code> , <code>digitally_shown_by</code> , <code>shown_by</code> , <code>about</code> , <code>represents</code> , <code>represents_instance_of_type</code> , <code>made_of</code> , <code>current_owner</code> , <code>current_custodian</code> , <code>current_permanent_custodian</code> , <code>current_location</code> , <code>shows</code> , <code>carries</code> , <code>approximated_by</code> , <code>language</code> , <code>digitally_carried_by</code> , <code>carried_by</code> , <code>refers_to</code> , <code>broader</code>

**Fonte:** Elaborado pelo autor, com base em Linked Art (2020).

Os metadados internos são utilizados para adicionar uma camada administrativa ao sistema, em que `entity_id` é utilizado para registrar a ID numérica e auto-incrementada da entidade; `name` é um texto em caixa baixa, sem acentos, sem caracteres especiais, e com espaços e pontuações substituídos por hífen — esta convenção também é conhecida pelo nome “*kebab case*”, resultando em strings como “tarsila-do-amaral”, “elucidario-art” ou “casa-museu-ema-klabin” — o campo `name` é utilizado pelo sistema para gerar a URI da entidade; `guid`, ou *Globally Unique Identifier*, é um identificador global exclusivo que é utilizado para identificar a entidade no sistema, diferentemente do WordPress que utiliza uma URI para identificar as entidades em suas tabelas, utilizaremos um inteiro de 128 bits (MICROSOFT, [s.d.]), por exemplo: “936DA01F-9ABD-4d9d-80C7-02AF85C822A8”,

esta decisão leva em consideração a real finalidade deste campo, de identificar globalmente e exclusivamente uma entidade. A forma como o WordPress utiliza a *guid* tem um problema no momento em que o utilizamos em ambientes diferentes, como produção ou desenvolvimento. Como o WordPress utiliza a URI para gerar a *guid*, na produção teríamos uma *guid* como por exemplo: “<https://exemplo.com/elucidario?objeto=2>”, e no desenvolvimento “<https://localhost:8080/elucidario?objeto=2>” o que viola a definição de *guid* (MICROSOFT, [s.d.]) uma vez que o domínio da URI é diferente em cada ambiente, exigindo, portanto, sempre um processo extra de “*find and replace*” no banco de dados quando for realizar uma migração de ambiente ou até mesmo de servidor de hospedagem ou domínio. O valor da *guid* deve ser imutável. Em *author* armazenamos a ID do usuário-autor da entidade; *status* armazena o status de publicação, segue o mesmo padrão do WordPress, “*publish*”, “*future*”, “*draft*”, “*pending*”, “*private*” e “*trash*”; O campo *password* define uma senha para esta entidade, utilizado pela API para controle do acesso; *created* e *modified* armazenam as datas de criação e modificação respectivamente; e, por fim, *history* armazena o histórico de edições da entidade.

Cada nova tabela adicionada seguiu o mesmo padrão de nomenclatura, utilizamos a abreviação “*lcdr*” e o prefixo definido nas configurações do arquivo “*wp-config.php*”. Geralmente esta configuração segue o padrão “*wp*”, mas pode ser customizada em cada instalação, por exemplo: se o prefixo definido for “*wp*”, a tabela de entidades será *wp\_lcdr\_entities* e a tabela de relações será *wp\_lcdr\_relationships*.

As descrições das tabelas a seguir são estruturadas da seguinte forma:

- **Coluna** - Nome da coluna;
- **Tipo** - Tipo de dado da coluna, em que o tamanho do campo é definido entre parênteses, por exemplo, *varchar(255)*, ou seja, uma *string* de 255 caracteres;
- **Extra** - Informações adicionais sobre a coluna, como se é uma coluna auto incrementada (*auto\_increment*), ou se é uma coluna que não pode ser nula (*not null*), também pode ser uma coluna que tem um valor padrão (*default*);
- **Chave** - Tipo de chave da coluna, se é uma chave primária (*PRI*), unitária (*UNIQUE*), ou um índice (*KEY*);
- **Descrição** - Descrição da coluna.

## Tabela wp\_lcdr\_entities

É a tabela que armazena as entidades do sistema. Cada entidade foi criada a partir dos *endpoints* do Linked Art.

**Quadro 59: Descrição da tabela MySQL wp\_lcdr\_entities do sistema.**

Coluna	Tipo	Extra	Chave	Descrição
entity_id	bigint(20) unsigned	auto_increment	PRI	Identificador único da entidade
name	varchar(255)	-	KEY	Slug da entidade, igual ao <i>post_name</i> da tabela <i>wp_posts</i>
guid	varchar(255)	-	UNIQUE	Identificador Único Global da entidade no sistema
author	bigint(20) unsigned	-	KEY	ID do autor da entidade, igual ao <i>post_author</i> da tabela <i>wp_posts</i>
status	varchar(20)	-	KEY	Status da entidade no sistema
password	varchar(255)	-	-	Senha para visualização da entidade
created	datetime	default: 0000-00-00 00:00:00	KEY	Data de criação da entidade, igual ao <i>post_date</i> da tabela <i>wp_posts</i>
modified	datetime	default: 0000-00-00 00:00:00	KEY	Data de modificação da entidade, igual ao <i>post_modified</i> da tabela <i>wp_posts</i>
type	varchar(60)	NOT NULL	-	Tipo da entidade
label	varchar(255)	-	-	Nome da entidade
identified_by	JSON	NOT NULL	-	Um array de objetos JSON que podem ser <i>Name</i> ou <i>Identifier</i>
referred_to_by	JSON	-	-	Um array de objetos JSON <i>Statement</i>
equivalent	JSON	-	-	Um array de JSON em que cada um é uma referência a um recurso externo equivalente
attributed_by	JSON	-	-	Um array de JSON <i>AttributeAssignment</i> que define um atributo a esta entidade
dimension	JSON	-	-	Um array de JSON

Coluna	Tipo	Extra	Chave	Descrição
				<i>Dimension</i> que descreve as dimensões
format	varchar(60)	-	-	Formato de mídia , por exemplo: <i>application/pdf</i> , <i>image/jpeg</i> , <i>video/mp4</i> , etc.
digitally_available_via	JSON	-	-	Um array de JSON <i>DigitalService</i>
created_by	JSON	-	-	Um objeto JSON <i>Creation</i>
contact_point	JSON	-	-	Um array de JSON <i>Identifier</i>
begin_of_existence	JSON	-	-	Um objeto JSON que pode ser <i>Birth</i> ou <i>Formation</i>
end_of_existence	JSON	-	-	Um objeto JSON que pode ser <i>Death</i> ou <i>Dissolution</i>
timespan	JSON	-	-	Um objeto JSON <i>TimeSpan</i>
part	JSON	-	-	Um array de objetos JSON que seja um evento de proveniência, como <i>Acquisition</i> , <i>Payment</i> , <i>TransferOfCustody</i> , <i>Encounter</i> , <i>RightAcquisition</i> , <i>Move</i> e <i>Promise</i>
produced_by	JSON	-	-	Um objeto JSON <i>Production</i>
destroyed_by	JSON	-	-	Um objeto JSON <i>Destruction</i>
removed_by	JSON	-	-	Um objeto JSON <i>PartRemoval</i>
defined_by	varchar(255)	-	-	-
content	longtext	-	-	-

**Fonte:** Elaborado pelo autor.

O tipo da classe é armazenada na coluna `type` e pode ser uma das seguintes opções: `Concept`, `Digital`, `Event`, `Provenance`, `Actor`, `Object`, `Place`, `Set`, `Textual` ou `Visual`. A coluna `label` armazena, como no Linked Art, um rótulo legível por humanos com foco nos desenvolvedores, este campo pode ser utilizado como título geral da página final, por exemplo, e pode ser gerado automaticamente baseado na coluna `identified_by`, ou definida manualmente pelo usuário.

Das colunas `identified_by` a `removed_by`, com exceção de `format` que discutiremos a seguir, são armazenadas em JSON e devem passar por devido processo de validação e codificação em JSON antes de serem inseridas no banco de dados. As colunas `format`, `defined_by` e `content` armazenam strings e passam por processos de escape e higienização antes de serem armazenadas. O escape é o processo de remover dados não desejáveis, como por exemplo a tag `<script>` do HTML que pode conter scripts maliciosos com intenções de gerar danos ao usuário ou ao sistema (WORDPRESS, [s.d.]). A higienização é o processo de remover outros tipos de dados que não são necessários ou que também podem gerar danos, como comandos SQL que podem remover ou alterar o banco de dados (WORDPRESS, [s.d.]).

A coluna `begin_of_existence` é uma união das propriedades `formed_by` da entidade `Groups` e `born` da entidade `People` do Linked Art, e a coluna `end_of_existence` é a união de `dissolved_by` e `died` de `Groups` e `People` respectivamente. O tipo de cada objeto inserido nessas colunas vai depender do tipo da entidade definido na coluna `type`, que é um dado obrigatório e define a estrutura da classe final da entidade e quais colunas e relações de fato esta entidade usa. Por exemplo, a entidade `Concept` utiliza somente as colunas `identified_by`, `referred_to_by`, `equivalent`, `attributed_by` e `created_by`. Da mesma forma, cada entidade pode possuir apenas um conjunto possíveis de relações entre outras entidades. Descrevemos no quadro a seguir as colunas e relações que cada entidade utiliza.

**Quadro 60: Colunas e predicados (relacionamentos) de cada Entidade do Mdorim.**

Entidade	Colunas usadas	Predicados permitidos
<i>Concept</i>	<code>identified_by</code> , <code>referred_to_by</code> , <code>equivalent</code> , <code>attributed_by</code> , <code>created_by</code>	<code>classified_as</code> , <code>referred_to_by</code> , <code>representation</code> , <code>member_of</code> , <code>subject_of</code> , <code>broader</code>
<i>Digital</i>	<code>identified_by</code> , <code>referred_to_by</code> , <code>equivalent</code> , <code>attributed_by</code> , <code>dimension</code> , <code>digitally_available_via</code> , <code>created_by</code>	<code>classified_as</code> , <code>referred_to_by</code> , <code>representation</code> , <code>member_of</code> , <code>subject_of</code> , <code>part_of</code> , <code>conforms_to</code> , <code>access_point</code> , <code>digitally_carries</code> , <code>digitally_shows</code> , <code>used_for</code>
<i>Event</i>	<code>identified_by</code> , <code>referred_to_by</code> , <code>equivalent</code> , <code>attributed_by</code> , <code>timespan</code>	<code>classified_as</code> , <code>referred_to_by</code> , <code>representation</code> , <code>member_of</code> , <code>subject_of</code> , <code>part_of</code> , <code>took_place_at</code> , <code>caused_by</code> , <code>carried_out_by</code> , <code>used_specific_object</code> , <code>influenced_by</code> , <code>technique</code>



Entidade	Colunas usadas	Predicados permitidos
<i>Provenance</i>	identified_by, referred_to_by, equivalent, attributed_by, timespan, part	classified_as, referred_to_by, representation, member_of, subject_of, part_of, took_place_at, caused_by, carried_out_by, used_specific_object, influenced_by, technique
<i>Actor</i>	identified_by, referred_to_by, equivalent, attributed_by, contact_point, begin_of_existence, end_of_existence	classified_as, referred_to_by, representation, member_of, subject_of, carried_out, residence
<i>Object</i>	identified_by, referred_to_by, equivalent, attributed_by, dimension, produced_by, destroyed_by, removed_by	classified_as, referred_to_by, representation, member_of, subject_of, part_of, used_for, took_place_at, caused_by, carried_out_by, used_specific_object, influenced_by, technique, made_of, current_owner, current_custodian, current_permanent_custodian, current_location, shows, carries
<i>Place</i>	identified_by, referred_to_by, equivalent, attributed_by	classified_as, referred_to_by, representation, member_of, subject_of, part_of, approximated_by
<i>Set</i>	identified_by, referred_to_by, equivalent, attributed_by, dimension, created_by, timespan	classified_as, referred_to_by, representation, member_of, subject_of, took_place_at, caused_by, carried_out_by, used_specific_object, influenced_by, technique
<i>Textual</i>	identified_by, referred_to_by, equivalent, attributed_by, dimension, created_by, timespan	classified_as, referred_to_by, representation, member_of, subject_of, part_of, used_for, took_place_at, caused_by, carried_out_by, used_specific_object, influenced_by, technique, about, language, digitally_carried_y, carried_by, refers_to
<i>Visual</i>	identified_by, referred_to_by, equivalent, attributed_by, dimension, created_by, timespan	classified_as, referred_to_by, member_of, subject_of, part_of, used_for, took_place_at, caused_by, carried_out_by, used_specific_object, influenced_by, technique, digitally_shown_by, shown_by, about, represents, represents_instance_of_type

**Fonte:** Elaborado pelo autor, com base em Linked Art (2020).

Perceba que a propriedade `referred_to_by` aparece tanto nas colunas, quanto nos predicados, isso se dá pois ela permite em seu registro tanto uma relação quanto um valor, por exemplo, uma entidade pode ser descrita por um objeto `Statement` ou por uma referência a uma entidade `Textual`, uma vez que ambas derivam da classe do CRM *E33\_Linguistic\_Object* definidas como tipo [CITAÇÃO DO GITHUB ISSUES, AGUARDAR RESPOSTA MACA@ECA].

### Tabela *wp\_lcdr\_relationships*

É a tabela que armazena as relações entre as entidades do sistema.

#### Quadro 61: Descrição da tabela MySQL *wp\_lcdr\_relationships* do sistema.

Coluna	Tipo	Extra	Chave	Descrição
rel_id	bigint(20) unsigned	auto_increment	PRI	Identificador único da relação
subject	bigint(20) unsigned	NOT NULL	KEY	Identificador único da entidade que é o sujeito da relação
predicate	varchar(50)	NOT NULL	KEY	Predicado da relação, pode ser qualquer uma das propriedades listadas como propriedade de relacionamento no Quadro 34.
object	bigint(20) unsigned	NOT NULL	KEY	Identificador único da entidade que é o objeto da relação
rel_order	int(11)	NOT NULL	KEY	Ordem da relação, para o caso de uma entidade ter mais de uma relação

**Fonte:** Elaborado pelo autor.

A coluna *rel\_id* armazena a ID única da relação, campo utilizado para facilitar comandos SQL como edição e remoção; *subject* e *object* armazenam as IDs das entidades presentes na tabela *wp\_lcdr\_entities*; a coluna *predicate* armazena o tipo de relação possível entre as duas entidades, por exemplo, se a entidade “A” é uma parte da entidade “B”, a coluna *predicate* armazenará o valor *part\_of*; e, por fim, a coluna *rel\_order* que armazena a ordem de exibição da relação na UI no caso de haver mais de uma relação entre entidades e predicados iguais. Por exemplo, se a entidade “A” é uma parte da entidade “B” e a entidade “C” também é uma parte da entidade “B”, a coluna *rel\_order* armazenará o valor “0” para a relação entre “A” e “B” e o valor “1” para a relação entre “C” e “B”, usando indexação a partir do valor “0” como padrão em linguagens de programação.

Criamos também outras duas tabelas, uma para definir o histórico de edições de cada registro e uma para definir as configurações do sistema.

### Tabela *wp\_lcdr\_history*

Armazena o histórico de edições das entidades.

#### Quadro 62: Descrição da tabela MySQL *wp\_lcdr\_history* do sistema.

Coluna	Tipo	Extra	Chave	Descrição
history_id	bigint(20) unsigned	auto_increment	PRI	Identificador único do registro
type	varchar(20)	NOT NULL	KEY	Tipo de evento no histórico de edição, podendo ser <i>Creation</i> ou <i>Edition</i>
timestamp	datetime	NOT NULL default: 0000-00-00 00:00:00	KEY	Data e hora do evento
entity_id	bigint(20) unsigned	NOT NULL	KEY	Identificador único da entidade que sofreu a alteração.
procedure_id	bigint(20) unsigned	NOT NULL	KEY	Identificador único da entidade que sofreu a alteração.
user_id	bigint(20) unsigned	NOT NULL	KEY	Identificador único do usuário que realizou a alteração.
property	varchar(200)	-	-	Nome da propriedade que sofreu a alteração.
related_event	bigint(20)	-	-	Identificador único de outro evento a que este evento está relacionado.
current	longtext	-	-	Valor atual da propriedade que sofreu a alteração.
previous	longtext	-	-	Valor antigo da propriedade que sofreu a alteração.

**Fonte:** Elaborado pelo autor.

A coluna `history_id` armazena a ID do evento de edição, enquanto `type` contextualiza que tipo de evento é este, podendo ser *Creation*, *Edition* ou *Exclusion*. Em `timestamp` registramos o carimbo de data e hora em que o evento ocorreu. As colunas `entity_id` e `user_id` armazenam a ID da Entidade que sofreu a edição e a ID do Usuário que realizou a edição, respectivamente. Em `property` armazenamos o nome da propriedade que foi editada, `related_event` registra a ID de outro evento de edição, caso este evento seja uma edição de uma edição, como por exemplo o ato de voltar a um ponto anterior. Em `previous` e `current` armazenamos o valor anterior e atual da propriedade editada.

### **Tabela `wp_lcdr_options`**

É a tabela que armazena as configurações do sistema, segue uma estrutura semelhante à tabela “`wp_options`” do WordPress.

### **Quadro 63: Descrição da tabela MySQL `wp_lcdr_options` do sistema.**

Coluna	Tipo	Extra	Chave	Descrição
id	bigint(20)	auto_increment	PRI	Identificador único do registro
name	varchar(64)	-	UNIQUE	Nome da configuração
value	JSON	-	-	Valor da configuração

**Fonte:** Elaborado pelo autor.

Em `id` armazenamos a ID da opção, `name` e `value` registram o nome e o valor da opção, respectivamente. Nesta tabela armazenamos as opções que podem ser reutilizadas por todo o sistema, como *tokens* de autenticação com APIs externas, por exemplo.

### Tabela `wp_lcdr_procedures`

**Quadro 64: Descrição da tabela MySQL `wp_lcdr_procedures` do sistema.**

Coluna	Tipo	Extra	Chave	Descrição
procedure_id	bigint(20) unsigned	auto_increment	PRI	ID do procedimento
type	varchar(64)	-	KEY	Tipo de procedimento
guid	varchar(255)	-	UNIQUE	Identificador Único Global do procedimento no sistema
description	longtext	-	-	Descrição do procedimento
author	bigint(20) unsigned	-	-	Usuário que criou o registro
created	timestamp	-	-	Data e hora da criação do registro
modified	timestamp	-	-	Data e hora da última atualização do registro
status	varchar(30)	KEY	-	Status do registro ( <i>draft</i> , <i>active</i> , <i>inactive</i> , <i>deleted</i> , <i>pending</i> , <i>scheduled</i> )
schedule	JSON	-	-	Um objeto <i>Schedule</i> que configura o agendamento de uma ação

**Fonte:** Elaborado pelo autor.

Tabela para armazenar as entidades *Procedure* que descrevem os procedimentos

*Spectrum*, o `type` demonstra qual procedimento do *Spectrum* está sendo criado e, o sistema se encarrega de, por meio da interface de mapeamento, definir quais metadados e como devem ser preenchidos para cada procedimento e em cada entidade. Por exemplo, o procedimento para Entrada de Objetos define as informações de identificação, descrição e de entrada do objeto como obrigatórias, e as de Entrada de Empréstimos como opcionais, dependendo do tipo de entrada de objeto sendo descrita, o sistema se encarrega de buscar

como os metadados devem ser preenchidos usando a interface de mapeamento e as tabelas descritas a seguir, e retorna quais metadados do Linked Art devem ser preenchidos e em qual esquema, o Object criado em seguida é armazenado na tabela “wp\_lcdr\_entity” e a relação entre Object e Procedure na tabela wp\_lcdr\_procedure\_entity. Em description registramos uma descrição do procedimento, e em created e modified registramos as datas de criação e modificação do procedimento, respectivamente. author registra a ID do usuário que criou o procedimento e em status os valores podem ser draft, active, inactive, deleted, pending, scheduled. Por fim a coluna schedule armazena um objeto Schedule opcional que define o agendamento de um procedimento, por exemplo, se o procedimento deve ser realizado em uma data específica, ou se deve ser repetido a cada semana, mês ou ano, por exemplo.

#### **Tabela wp\_lcdr\_procedure\_entity**

**Quadro 65: Descrição da tabela MySQL wp\_lcdr\_procedure\_entity do sistema.**

Coluna	Tipo	Extra	Chave	Descrição
rel_id	bigint(20) unsigned	auto_increment	PRI	Identificador único da relação
entity_id	bigint(20) unsigned	NOT NULL	KEY	ID da entidade
procedure_id	bigint(20) unsigned	NOT NULL	KEY	ID do procedimento
rel_order	int(11)	NOT NULL	KEY	Ordem da relação

**Fonte:** Elaborado pelo autor.

#### **Tabela wp\_lcdr\_mapping**

É a tabela que registra as informações de mapeamento do modelo de dados para outros modelos externos. Esta tabela é utilizada pela interface de mapeamento do sistema, e permite que o usuário adicione novos mapeamentos de acordo com sua necessidade. Os mapeamentos são majoritariamente utilizados nas funções de exportação e importação de dados do sistema.

**Quadro 66: Descrição da tabela MySQL wp\_lcdr\_mapping do sistema.**

Coluna	Tipo	Extra	Chave	Descrição
mapping_id	bigint(20) unsigned	auto_increment	PRI	Identificador único do registro
name	varchar(255)	NOT NULL	KEY	Nome do mapeamento em

Coluna	Tipo	Extra	Chave	Descrição
				kebab-case
title	varchar(255)	NOT NULL	KEY	Título do mapeamento
description	longtext	-	-	Descrição do mapeamento
author	bigint(20) unsigned	-	KEY	ID do autor do mapeamento
version	varchar(20)	-	-	Versão do mapeamento
created	datetime	NOT NULL default: 0000-00-00 00:00:00	KEY	Data e hora de criação do registro
modified	datetime	NOT NULL default: 0000-00-00 00:00:00	KEY	Data e hora da última modificação do registro

**Fonte:** Elaborado pelo autor.

A coluna `mapping_id` é a ID do mapeamento; `name` registra, assim como em “`wp_lcdr_entities`”, o nome do mapeamento em *kebab-case*; `title` registra o título do mapeamento; a descrição do mapeamento pode ser registrada em `description`; `author` registra a ID única do usuário; e, por fim, `created` e `modified` registram as datas de criação e modificação do mapeamento, respectivamente.

### Tabela `wp_lcdr_prop_map`

Tabela para armazenar as propriedades mapeadas do modelo de dados para outros modelos externos.

#### Quadro 67: Descrição da tabela MySQL `wp_lcdr_prop_map` do sistema.

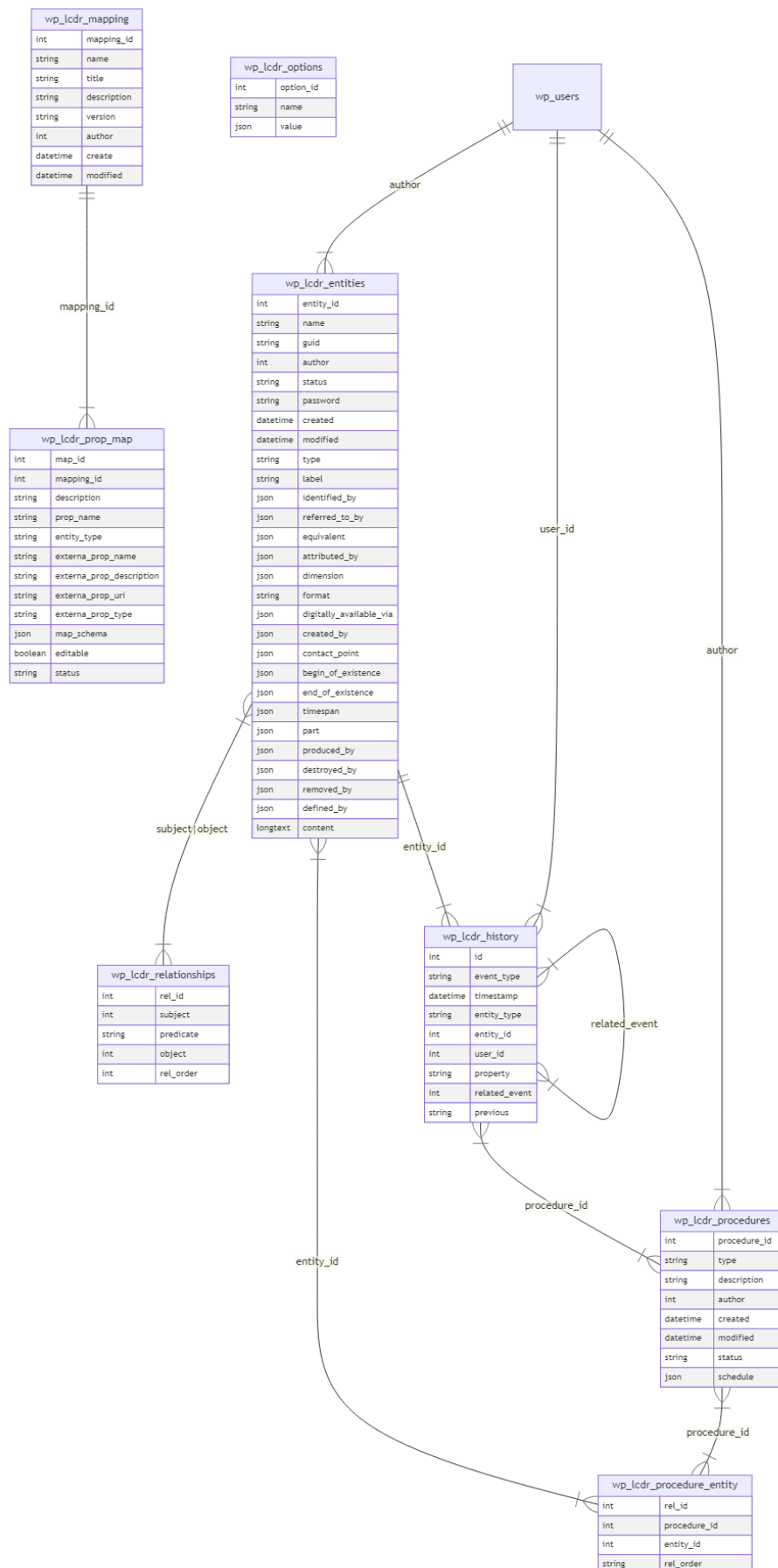
Coluna	Tipo	Extra	Chave	Descrição
map_id	bigint(20) unsigned	auto_increment	PRI	Identificador único do registro
mapping_id	bigint(20) unsigned	NOT NULL	KEY	ID de uma entidade <i>Mapping</i> do sistema
description	longtext	-	-	Descrição do mapeamento
prop_name	varchar(50)	NOT NULL	KEY	Nome da propriedade sendo mapeada
entity_type	varchar(50)	NOT NULL	KEY	Tipo de entidade mapeada
external_prop_name	varchar(100)	NOT NULL	KEY	Nome da propriedade externa sendo mapeada
external_prop_description	longtext	-	-	Descrição da propriedade externa
external_prop_uri	varchar(255)	NOT NULL	-	URI da propriedade externa

Coluna	Tipo	Extra	Chave	Descrição
external_prop_type	varchar(20)	NOT NULL	-	Tipo de dados da propriedade externa
map_value	JSON	NOT NULL	-	Valores padrão do mapeamento, em formato JSON
editable	tinyint(1)	NOT NULL, default: 1	-	Como um boolean, registra se o mapeamento pode ou não ser editado no contexto de edição das entidades
status	varchar(20)	NOT NULL	-	Status do mapeamento, podendo ser <i>active</i> ou <i>inactive</i>

**Fonte:** Elaborado pelo autor.

A coluna `map_id` registra a ID do mapeamento de uma propriedade; já `mapping_id` registra a ID do mapeamento (`wp_lcdr_mapping`) ao qual a propriedade pertence; `description` registra uma descrição opcional sobre o mapeamento; `prop_name` e `entity_type` registram o nome da propriedade mapeada e o tipo da entidade que utiliza determinada propriedade; `external_prop_name`, `external_prop_description`, `external_prop_uri` e `external_prop_type` registram o nome, a descrição, a URI e o tipo da propriedade mapeada, respectivamente; `editable` é um valor booleano que registra se o mapeamento pode ser editado ou não no contexto de edição de uma entidade, isso habilita o usuário a sobrescrever o valor do mapeamento no momento da edição de uma entidade; em `status` registramos o status do mapeamento, podendo ser *active* ou *inactive*; e, por fim, em `map_value` registramos os valores padrão do mapeamento.

Podemos representar as tabelas definidas anteriormente em um diagrama de entidade e relacionamento, como mostrado na figura a seguir:



**Figura 13:**  
**Diagrama ER**  
**das tabelas**  
**MySQL do**  
**Mdorim.**

**Fonte:**  
 Elaborado pelo  
 autor.



A tabela `wp_users` é a tabela padrão do WordPress para armazenamento das informações sobre os usuários do sistema, esta tabela possui uma relação de um para vários com a tabela `wp_lcdr_entities`, ou seja, um usuário pode ter várias entidades, mas uma entidade só pode ter um usuário, este tipo de relação também se aplica à tabela `wp_lcdr_history`, em que um usuário pode ter várias edições, mas uma edição só pode ter um usuário. A tabela `wp_lcdr_relationships` pode ter uma relação de vários para vários com a tabela `wp_lcdr_entities`, ou seja, uma entidade pode ter várias relações, e uma relação pode ter duas entidades. As tabelas `wp_lcdr_mapping` e `wp_lcdr_prop_map` possuem relação de vários para um, em que uma entidade Mapping pode conter diversos PropMap, enquanto o PropMap se relaciona com somente um Mapping. Já a tabela `wp_lcdr_options` não possui relação com nenhuma outra tabela e opera de forma independente.

O Mdorim também é utilizado pelo plugin para gerar a UI administrativa para edição das entidades no WordPress, portanto utilizaremos os esquemas definidos em JSON-Schema para gerar automaticamente os componentes da UI. Como o JSON-Schema não foi criado para ser utilizado por usuários finais, mas sim por desenvolvedores e máquinas, não possui suporte para localização (i10n) dos valores definidos nas propriedades `title` e `description`, portanto esses valores foram definidos em inglês. Fato que impôs um desafio ao seu uso na UI, uma vez que devemos dar suporte à outros idiomas como português e espanhol. Para resolver esse problema, criamos arquivos JSON paralelos que mapeiam cada propriedade do Linked Art à um objeto JSON que permite a internacionalização (i18n) dos textos para a UI:

#### Quadro 48: JSON-UI

```
{
  "$schema": "http://json-schema.org/schema",
  "$id": "https://elucidario.art/mdorim/schemas/json-ui/schema.json",
  "title": "JSON-UI",
  "description": "JSON-UI is a JSON-based declarative schema for
describing user interfaces, use with JSON-Schema to describe data
structures and JSON-UI to describe the user interface for the data
structure.",
  "type": "object",
  "patternProperties": {
    "^_?[a-z][a-z0-9_]*$": {
      "type": "object",
```

```

        "properties": {
            "label": {
                "type": "array",
                "description": "The label of the data model, use an
array of localized strings to support multiple languages.",
                "items": {
                    "$ref": "#/definitions/localizedString"
                }
            },
            "description": {
                "type": "array",
                "description": "The description of the data model,
use an array of localized strings to support multiple languages.",
                "items": {
                    "$ref": "#/definitions/localizedString"
                }
            }
        },
        "required": ["label", "description"],
        "additionalProperties": false
    },
    "definitions": {
        "localizedString": {
            "type": "object",
            "properties": {
                "lang": {
                    "type": "string",
                    "description": "The language of the string, use RFC
5646 language tags, e.g. pt-BR, en-US, en-GB, fr-FR, etc."
                },
                "content": {
                    "type": "string",
                    "description": "The localized content of the string."
                }
            },
            "required": ["lang", "content"],
            "additionalProperties": false
        }
    }
}

```

Fonte: Elaborado pelo autor.

Em definitions, definimos um objeto localizedString que contém as propriedades lang e content, ambas obrigatórias. A propriedade lang deve seguir a RFC 5646<sup>[1]</sup> e a propriedade content deve ser uma string no idioma definido em lang.

A propriedade `patternProperties` define o padrão de nome de cada propriedade do Linked Art para o mapeamento, no caso utilizamos o padrão `Regex[1 ^_?[a-z][a-z0-9_]*$`, em que:

- `^_?` - a propriedade pode iniciar opcionalmente com o caractere “\_”;
- `[a-z]` - a propriedade deve iniciar com uma letra minúscula;
- `[a-z0-9_]*$` - a propriedade pode conter letras minúsculas, números e o caractere “\_”;

Com este padrão conseguimos selecionar qualquer nome de propriedade do Linked Art, e cada chave deve conter um objeto que define as seguintes propriedades:

- `label`: array de objetos `localizedString` que definem o nome da propriedade em diferentes idiomas;
- `description`: array de objetos `localizedString` que definem a descrição da propriedade em diferentes idiomas;
- `messages`: array de objetos que possuem as propriedades `code`, `type` e `content`, em que `code` é o código da mensagem, `type` é o tipo da mensagem (*error*, *success*, *warning*, *info*) e `content` é um array de objetos `localizedString` que definem a mensagem em diferentes idiomas;
- `component`: *string* que define o nome do componente React que será utilizado para renderização e edição do metadado, esta propriedade tem prioridade acima do tipo definido em “type” no JSON-Schema.

Utilizando este esquema, o metadado “`identified_by`” do Linked Art pode ser traduzido da seguinte forma:

```
{
  "label": [
    {
      "lang": "pt-BR",
      "content": "Identificado por"
    },
    {
      "lang": "es-ES",
      "content": "Identificado por"
    }
  ],
}
```

```

    {
      "lang": "en-US",
      "content": "Identified by"
    }
  ],
  "description": [
    {
      "lang": "pt-BR",
      "content": "Identificador do recurso"
    },
    {
      "lang": "es-ES",
      "content": "Identificador del recurso"
    },
    {
      "lang": "en-US",
      "content": "Resource identifier"
    }
  ]
}

```

Esta interface de tradução é utilizada somente no código, em que novos arquivos JSON são adicionados para cada idioma que se deseja dar suporte. Cada novo idioma deve conter uma suite de testes adicionais para validação do formato.

## Testes

Os testes do Mdorim foram definidos utilizando as bibliotecas Jest (NAKAZAWA, 2011) e jest-json-schema (AMERICAN EXPRESS, [s.d.]), pra realização dos testes unitários. O Jest é uma ferramenta para criar e rodar testes automatizados em JavaScript, que permite a extensão de suas funcionalidades através de plugins. Uma dessas funcionalidades é a criação de *matchers*, que são funções que permitem a validação de valores, como por exemplo, a validação de uma *string*:

Figura 14: Exemplo de teste utilizando o Jest

```

// js
expect("banana").toBe("banana");

```

**Fonte:** elaborado pelo autor.

Em que *expect* é uma função que espera um valor qualquer como parâmetro, nela definimos o valor que esperamos receber, e em seguida utilizamos o *matcher toBe* para validar se o valor recebido é igual ao valor esperado.

O `jest-json-schema` é um plugin para o Jest criado para realizar a validação de arquivos JSON com formato JSON-Schema, esta biblioteca adiciona um novo *matcher* ao Jest, o `toMatchSchema`, que permite a validação de um objeto JSON com um JSON-Schema.

Dessa forma podemos utilizar os próprios métodos de importação do JavaScript para importar os arquivos JSON-Schema e realizar a validação dos objetos JSON com o JSON-Schema correspondente, como no exemplo abaixo:

Figura 15: Exemplo de teste utilizando o `jest-json-schema`

```
// js
import { matchersWithOptions } from "jest-json-schema";

import mdorim from "../lib/mjs/index";

expect.extend({
  matchersWithOptions({
    schemas: [mdorim.schemas.mapping],
  })
});

expect(mdorim.mapping.mapping_test).toMatchSchema(mdorim.schemas.mapping);
```

**Fonte:** elaborado pelo autor.

Em um primeiro momento extendemos a função `expect` e adicionamos um novo *matcher* fornecido pela biblioteca `jest-json-schema`, em seguida importamos o arquivo JSON-Schema que queremos utilizar para validar o objeto JSON, e por fim utilizamos o *matcher* `toMatchSchema` para validar o objeto JSON com o JSON-Schema correspondente.

Cada `expect` deve retornar `true` para que o teste seja considerado válido, caso contrário o teste é considerado inválido, neste caso o Jest retorna uma mensagem de erro indicando o motivo do teste ter falhado e o erro faz com que a Integração Contínua seja interrompida. Portanto é necessário que todos os testes sejam válidos para que a Integração Contínua seja concluída com sucesso.

### 9.1.2 @elucidario/pkg-design-system

#### Escopo

O *Design System* ou Sistema de Design (SD) do Elucidário.art consiste em um conjunto de regras, definições, padrões de design e bibliotecas de componentes utilizados para construir

a identidade visual e as interfaces de usuário do plugin. Nele definimos fontes, cores, padrões, espaçamentos, alinhamentos, layouts, etc. O SD é utilizado para construir a interface de administração do plugin, acessível para o usuário logado e com permissão para visualização e edição.

### **Descrição**

Os conjuntos de regras e definições delimitam o design em padrões consistentes para reutilização ao longo de toda a aplicação, nele definimos fontes, cores, espaçamentos, tamanhos, alinhamentos, etc, juntamente com as bibliotecas de componentes que são conjuntos de componentes React que podem ser utilizados para construir as interfaces de usuário do plugin. Os componentes são construídos utilizando o padrão de design definido no SD, dessa forma conseguimos garantir a consistência do design ao longo da aplicação, além de facilitar a manutenção e a criação de novos componentes.

Todas as dependências do Elucidário.art são open-source, assim como a fonte utilizada, a Inter (ANDERSSON, 2017) desenvolvida para uso em telas.

Figura 16: Fonte Inter.

Inter Regular  
Sample

ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz () & ?! @  
1234567890 .,:; /→

Inter Bold  
Sample

ABCDEFGHIJ  
KLMNOP  
QRSTUVWXYZ

*Fonte: Exemplo da fonte Inter (ANDERSSON, Rasmus, 2017).*

Como vimos no “@elucidario/pkg-mdorim”, os metadados do modelo são descritos em JSON-Schema que estipula sete tipos de dados primitivos:

1. `string` - sequência de caracteres;
2. `null` - valor nulo;
3. `number` - número;
4. `integer` - número inteiro;
5. `boolean` - valor booleano, `true` ou `false`, 1 ou 0;
6. `array` - lista de valores;
7. `object` - objeto JSON.

Para a definição do SD utilizaremos o mesmo formato definido no Mdorim com algumas camadas extras para melhorar a experiência de desenvolvimento. Sendo uma delas o mapeamento entre os tipos de dados primitivos e tags HTML com seus atributos. Para componentes mais complexos, como array desenvolvemos um componente React exclusivo que permite a adição, remoção e reordenação de itens. O quadro a seguir descreve o mapeamento realizado, em que “schema type” é o tipo de dado definido no JSON-Schema; “schema format” representa o formato de dado de uma string no JSON-Schema; “schema extra” representa qualquer outra propriedade do vocabulário JSON-Schema; “html tag” é a tag html relacionada no mapeamento, pode vir acompanhada de um sinal “>” representando uma hierarquia entre tags html ou componentes React; “input type” é o tipo de input HTML utilizado; “attributes” são os atributos da tag html; “descrição” é uma breve descrição do mapeamento realizado.

**Quadro 68: Mapeamento entre JSON-Schema e HTML para a definição do SD do Elucidário.art**

Schema tipo	Schema formato	Schema extra	HTML tag	Input tipo	Atributos	Descrição
string	-	-	input	text	-	Texto livre
string	date-time	-	input	date, datetime, datetime-local, time, week	-	Data e hora
string	uri	-	input	url	-	URI
string	email	-	input	email	-	E-mail
string	ip	-	input	text	REGEX pattern	IP
string	uuid	-	input	text	REGEX pattern	UUID
string	hex-color	-	input	color	-	Cor em código hexadecimal
string	-	enum	select > option	-	-	Usa-se a tag select para o elemento pai e uma tag option para cada item do enum
number   integer	-	minimum   maximum	input	number	min   max   step	Número



Schema tipo	Schema formato	Schema extra	HTML tag	Input tipo	Atributos	Descrição
		multipleOf				
boolean	-	-	input	checkbox	-	Caixa de seleção
object	-	-	fieldset > <qualquer outra propriedade>	-	-	Objeto composto por propriedades
array	-	-	fieldset > componente Array	-	-	Array de itens

**Fonte:** Elaborado pelo autor.

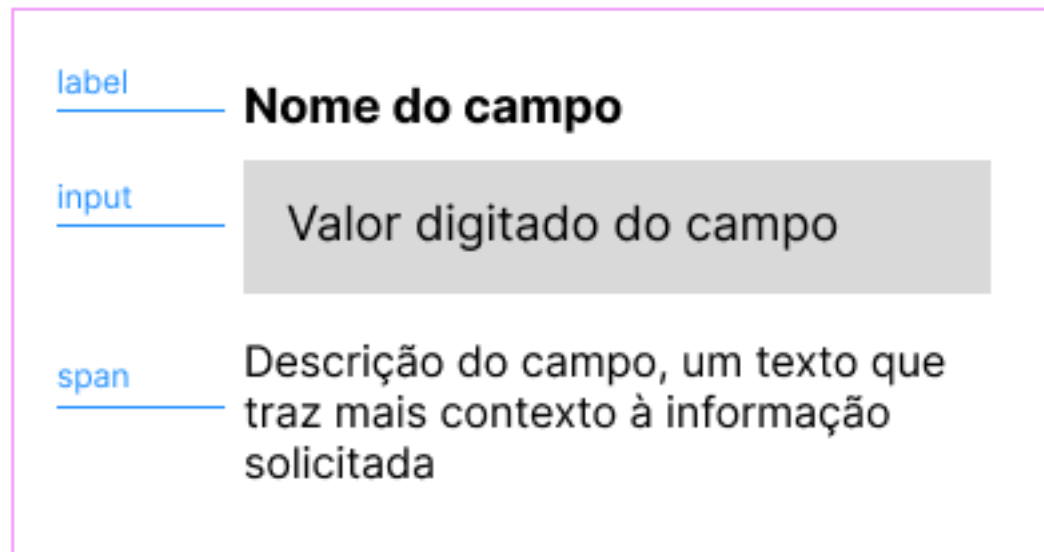
Criamos um componente chamado `Field`, construído a partir da tag `fieldset` do HTML, este componente agrega e padroniza todos os requisitos para o devido funcionamento de um componente de formulário React, e utiliza os mapeamentos descritos anteriormente para renderizar a UI. O componente `Field` resolve os seguintes requisitos:

- nome do campo;
- descrição;
- validação de dados;
- feedback de validação;
- a11y (acessibilidade):
- contraste de cores;
- navegação por teclado;
- WAI-ARIA;
- i18n (internacionalização):
- tradução de textos;
- tradução de mensagens de erro;

Podemos representar a anatomia do componente `Field` da seguinte forma:

**Figura 17: Anatomia do componente “Field”.**

## Field

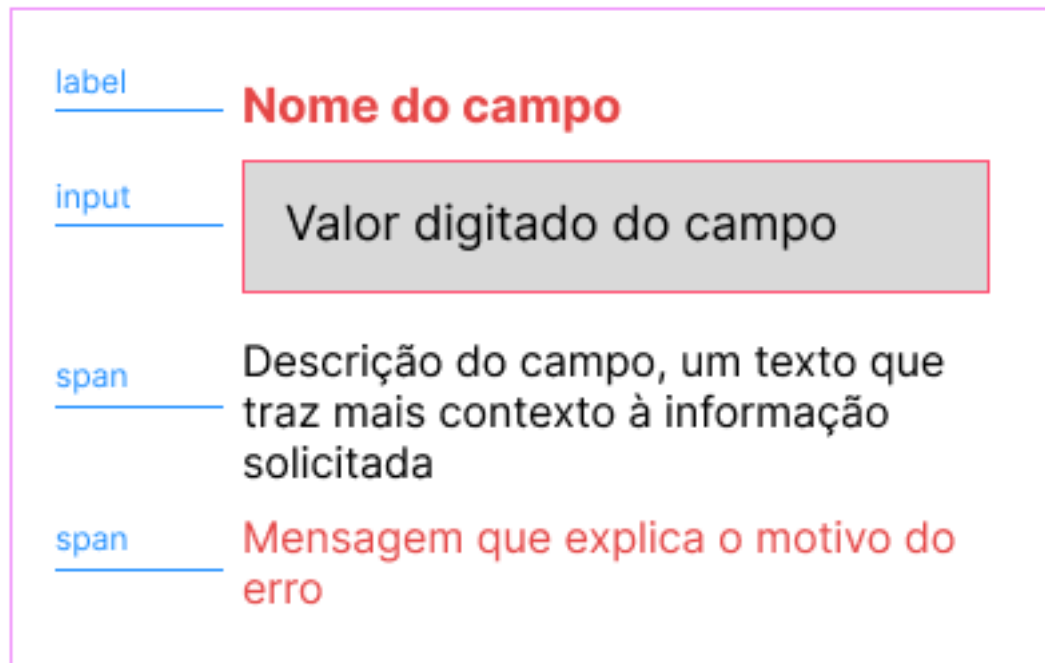


*Fonte: Elaborado pelo autor.*

Para a validação de dados, além do feedback visual é necessário que o *Field* informe ao usuário a mensagem da validação, seja sucesso ou erro:

**Figura 18: Feedback de erro do componente “Field”.**

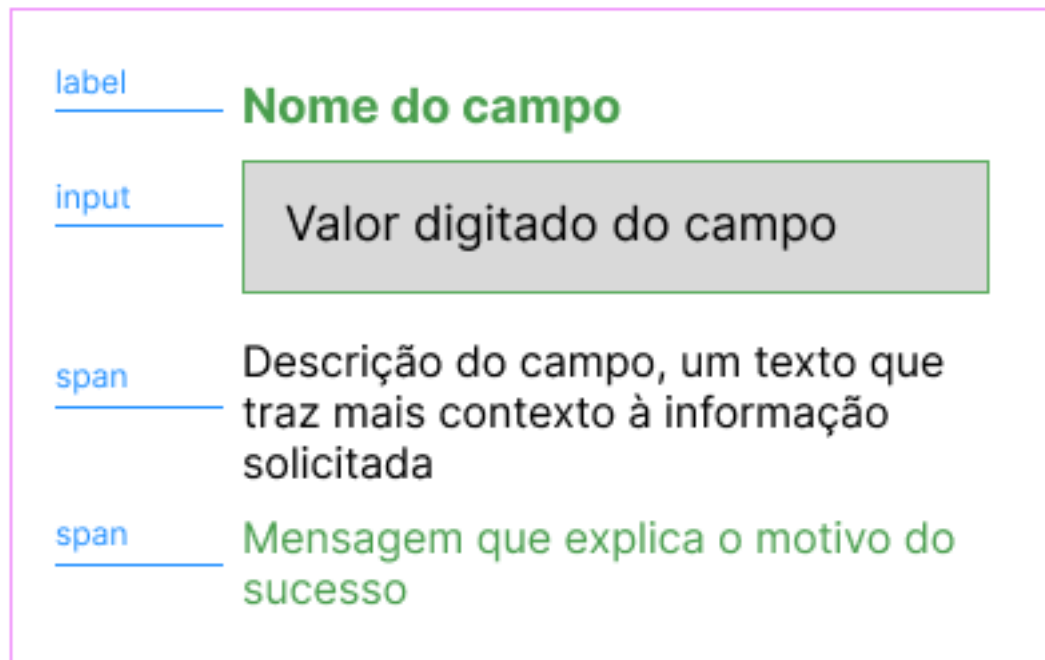
## Field - error



*Fonte: Elaborado pelo autor.*

**Figura 19: Feedback de sucesso do componente “Field”.**

### Field - success

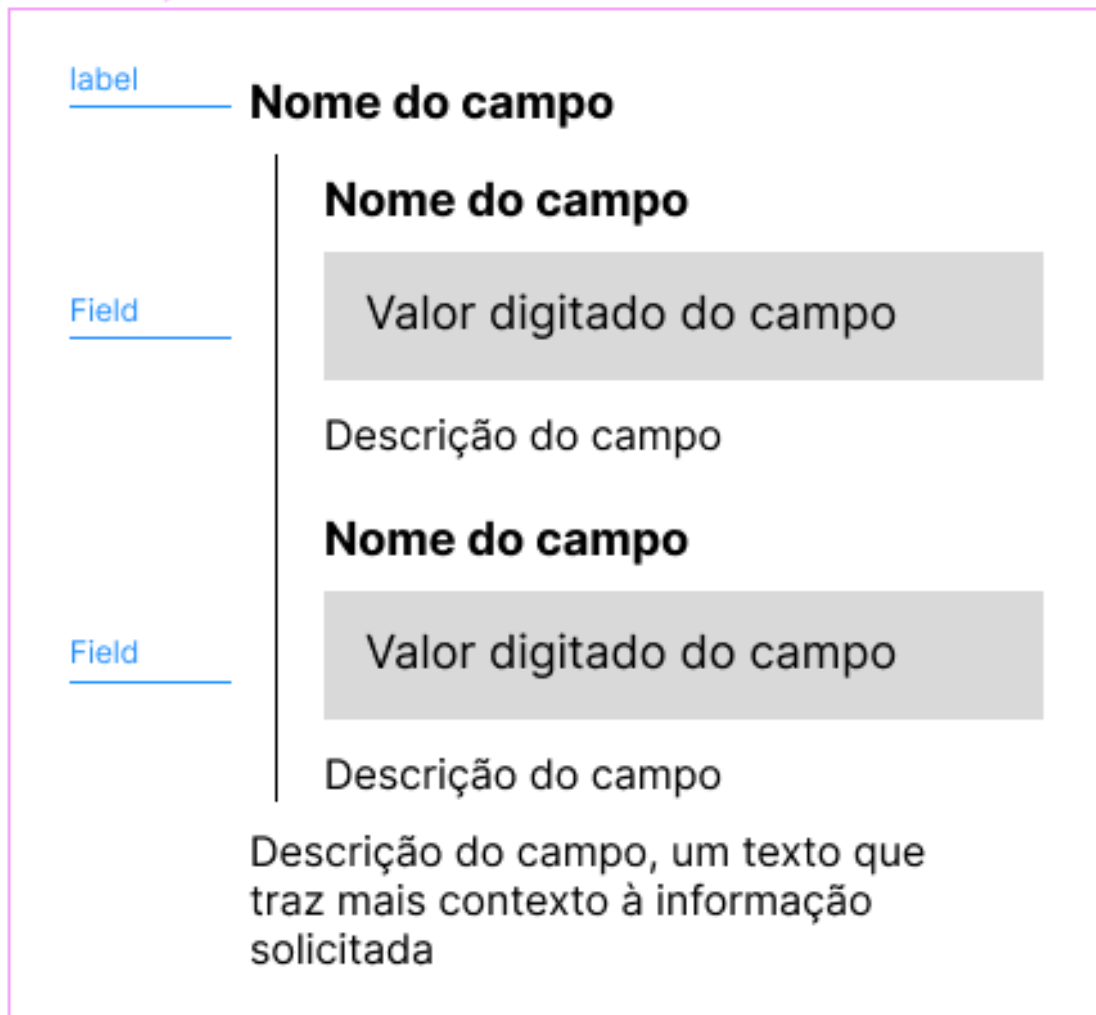


*Fonte: Elaborado pelo autor.*

No caso de object o *Field* é capaz de aninhar outros *Fields*:

**Figura 20: Anatomia do componente “Field” com “object”.**

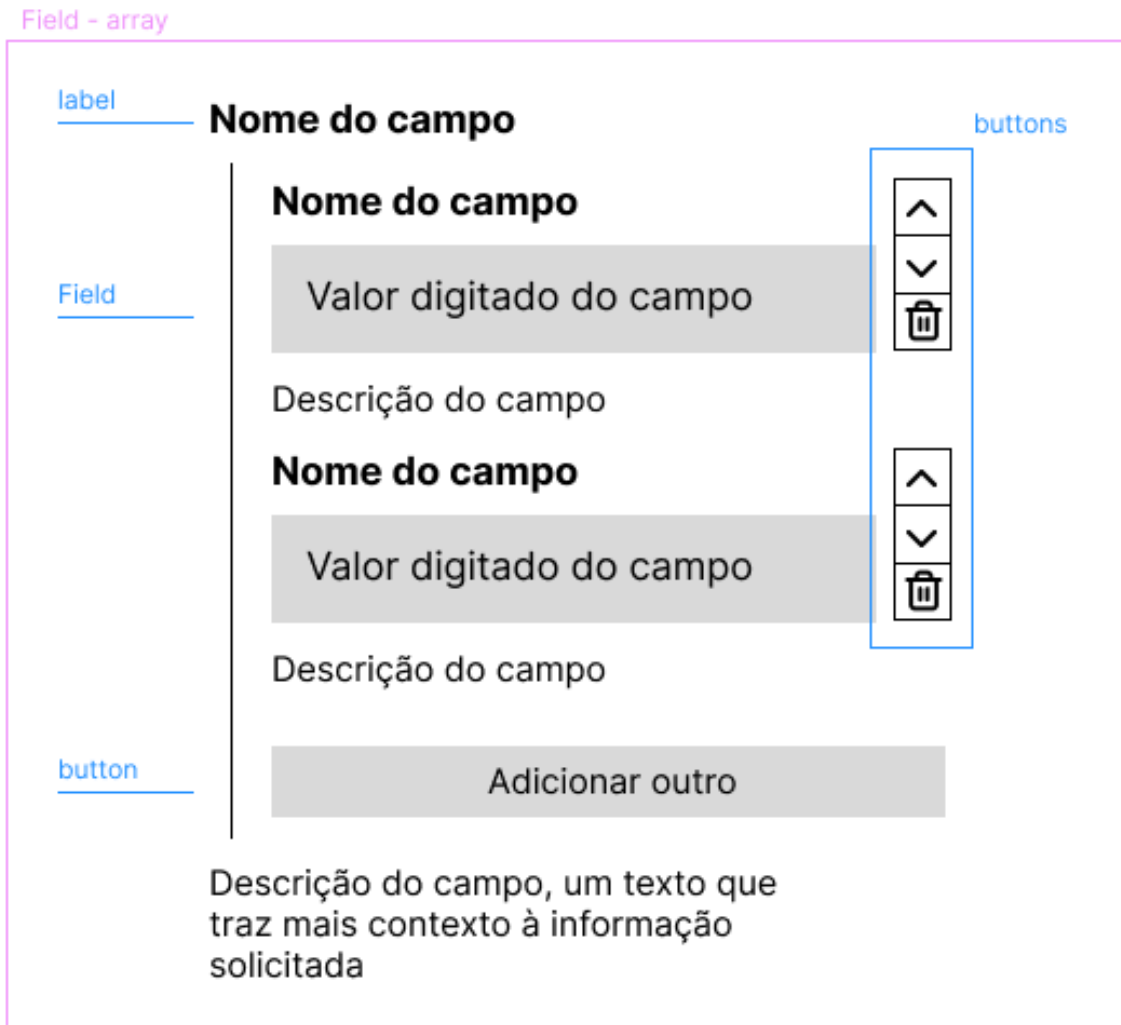
## Field - object



*Fonte: Elaborado pelo autor.*

Assim como o array é capaz de adicionar, remover e reordenar *Fields*:

**Figura 21: Anatomia do componente “Field” com “array”.**

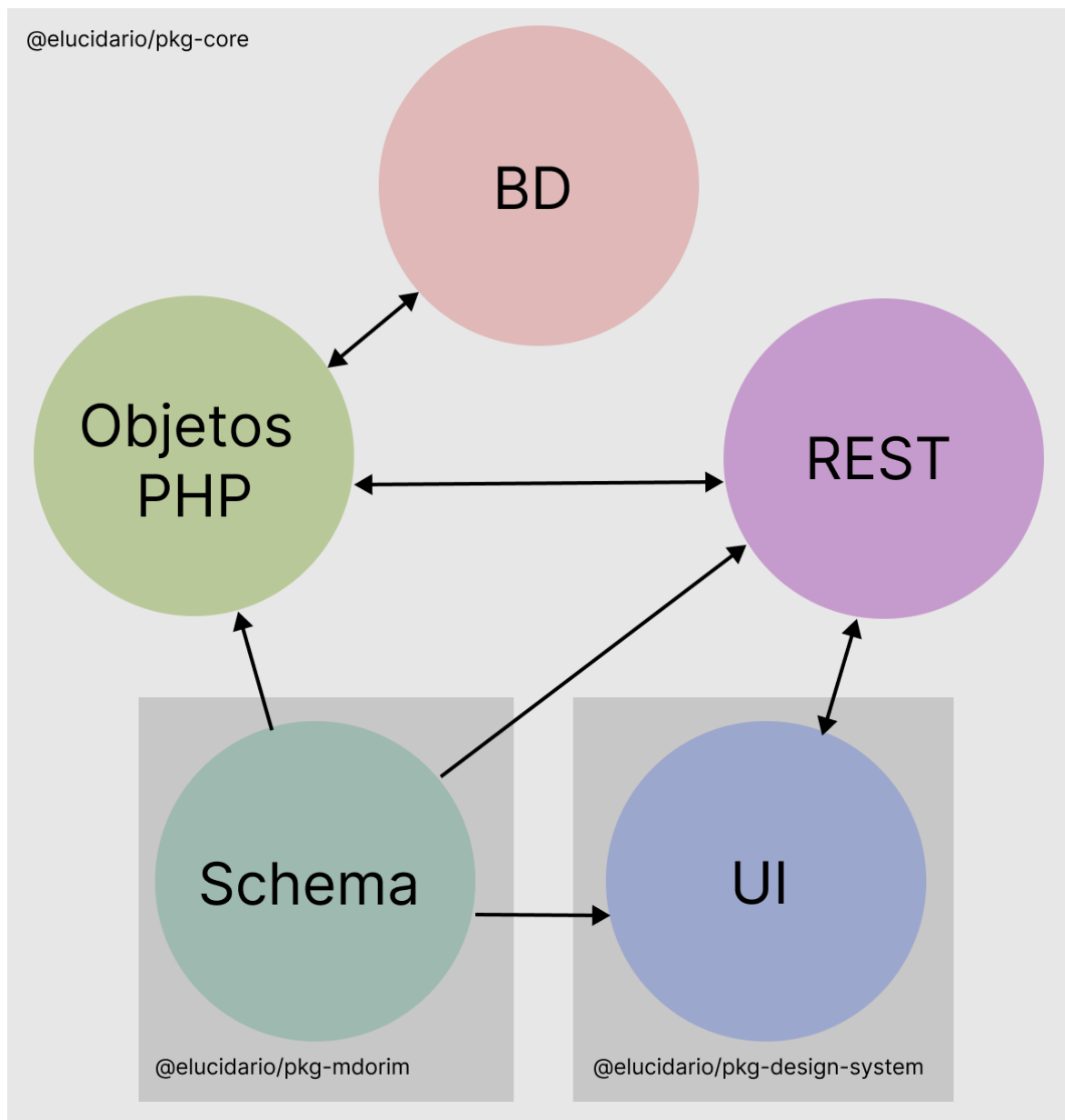


Fonte: Elaborado pelo autor.

### 9.1.3 @elucidario/pkg-core

Este pacote é o plugin final. Ele pode ser instalado em diversas instâncias do WordPress, como em um ambiente local (*localhost*), uma rede privada organizacional, em uma operadora de serviços *cloud* ou em um serviço de hospedagem compartilhada de sites. Nele definimos as principais funcionalidades do sistema como banco de dados, rotas, páginas administrativas, configurações, etc. Este pacote é o principal do Elucidário.art e agrega os demais pacotes.

**Figura 22:** @elucidario/pkg-core



*Fonte: Elaborado pelo autor.*

## 7.2. O Elucidário.art na Casa Museu Ema Klabin

Falar sobre a implementação do sistema na Casa Museu Ema Klabin.

## 8. Considerações finais

AMERICAN EXPRESS. **jest-json-schema**. [s.d.]. Disponível em:

<https://github.com/americanexpress/jest-json-schema>. Acesso em: 20 ago. 2023.

ANDERSSON, Rasmus. **The Inter font family**. 2017. Acesso em: 21 ago. 2023.

BACA, Murtha (ORG.). **Cataloguing Cultural Objects (CCO)**. Estados Unidos da América: Visual Resources Association, 2006. Disponível em:  
<https://www.vraweb.org/s/CatalogingCulturalObjectsFullv2.pdf>.

BACA, Murtha; HARPRING, Patricia; LANZI, Elisa; MCRAE, Linda; WHITESIDE, Ann. **Cataloging Cultural Objects: A Guide to Describing Cultural Works and Their Images**. Chicago: American Library Association, 2006. Disponível em:  
<https://vraweb.org/wp-content/uploads/2020/04/CatalogingCulturalObjectsFullv2.pdf>.  
Acesso em: 29 jul. 2022.

BARRETO, Aldo de Albuquerque. Uma história da Ciência da Informação. *Em: Para entender a Ciência da Informação*. Salvador: EDUFBA, 2007. Disponível em:  
<https://repositorio.ufba.br/bitstream/ufba/145/1/Para%20entender%20a%20ciencia%20da%20informacao.pdf>. Acesso em: 15 jul. 2022.

BERGMANN, Sebastian. **PHPUnit**. 2001. Disponível em: <https://phpunit.de>. Acesso em: 30 jul. 2023.

BERNES-LEE, Tim. **Linked Data**. 2022. Disponível em:  
<https://www.w3.org/DesignIssues/LinkedData.html>. Acesso em: 3 ago. 2022.

BEVILACQUA, Gabriel Moore Forell. **SPECTRUM 4.0: O padrão para gestão de coleções de museus do Reino Unido**. São Paulo: Secretaria de Estado de Cultura; Associação de Amigos do Museu do Café; Pinacoteca do Estado de São Paulo, 2014. v. 2  
Disponível em: [https://spectrum-pt.org/wp-content/uploads/2021/03/Spectrum\\_PT\\_NET.pdf](https://spectrum-pt.org/wp-content/uploads/2021/03/Spectrum_PT_NET.pdf). Acesso em: 20 maio. 2023.

BEVILACQUA, Gabriel Moore Forell; MONTEIRO, Juliana; CONTE, Marília Bona; BOTALLO, Marilúcia (ORG.). **Declaração de Princípios de Documentação em Museus e Diretrizes Internacionais de Informação sobre Objetos de Museus: Categorias de Informação do Comitê Internacional de Documentação**. São Paulo: Secretaria de Estado de Cultura de São Paulo; Associação de Amigos do Museu do Café; Pinacoteca do Estado de São Paulo, 2014. Disponível em: <https://cidoc.mini.icom.museum/wp->



content/uploads/sites/6/2020/03/CIDOC-Declaracao-de-principios.pdf. Acesso em: 5 mar. 2021.

BRASIL. **Lei no 11.904 de 14 de janeiro de 2009. Regulamenta dispositivos da Lei no 11.904, de 14 de janeiro de 2009, que institui o Estatuto de Museus, e da Lei no 11.906, de 20 de janeiro de 2009, que cria o Instituto Brasileiro de Museus - IBRAM**, 14 jan. 2009. Disponível em: [http://www.planalto.gov.br/ccivil\\_03/\\_ato2007-2010/2009/lei/11904.htm](http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2009/lei/11904.htm). Acesso em: 10 mar. 2021.

BRASIL. **Decreto no 8.124 de 17 de outubro de 2013. Regulamenta dispositivos da Lei no 11.904, de 14 de janeiro de 2009, que institui o Estatuto de Museus, e da Lei no 11.906, de 20 de janeiro de 2009, que cria o Instituto Brasileiro de Museus - IBRAM.**, 17 out. 2013. Disponível em: [http://www.planalto.gov.br/ccivil\\_03/\\_ato2011-2014/2013/decreto/d8124.htm](http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2013/decreto/d8124.htm). Acesso em: 10 mar. 2021.

BRAY, T. Ed.; IETF, Internet Engineering Task Force. **RFC 8259 - The JavaScript Object Notation (JSON) Data Interchange Format**. 2017. Disponível em: <https://www.rfc-editor.org/rfc/rfc8259>. Acesso em: 2 jun. 2023.

BUCKLAND, Michael K. Information as Thing. **Journal of the American Society for Information Science (1986-1998)**, New Jersey, v. 42, 1991.

CAPURRO, Rafael. Epistemologia e Ciência da Informação. **V Encontro Nacional de Pesquisa em Ciência da Informação**, Belo Horizonte, 2003. Disponível em: [http://www.capurro.de/enancib\\_p.htm](http://www.capurro.de/enancib_p.htm). Acesso em: 22 jul. 2022.

CAPURRO, Rafael; HJORLAND, Birger. O conceito de informação. **Perspectivas em Ciência da Informação**, Belo Horizonte, v. 12, p. 148–207, 2007. DOI: <https://doi.org/10.1590/S1413-99362007000100012>. Acesso em: 22 jul. 2022.

CARO-CASTRO, Carmem. Vocabulários estruturados, Web Semântica y Linked Data: oportunidades y retos para los profesionales de la documentación. **II Seminário de Estudos da Informação: Arquivologia, Biblioteconomia e Ciência de Informação: Identidades, Contrastes e Perspectivas de Interlocução**, Niterói, 2011.

CETIC.BR. **Pesquisa sobre o uso das tecnologias de informação e comunicação nos equipamentos culturais brasileiros: TIC Cultura 2016.** 2016. Disponível em: <https://www.cetic.br/pt/tics/cultura/2016/geral/>. Acesso em: 18 jul. 2022.

CETIC.BR. **Pesquisa sobre o uso das tecnologias de informação e comunicação nos equipamentos culturais brasileiros: TIC Cultura 2018.** 2018. Disponível em: <https://www.cetic.br/pt/tics/cultura/2018/geral/>. Acesso em: 18 jul. 2022.

CETIC.BR. **Pesquisa sobre o uso das tecnologias de informação e comunicação nos equipamentos culturais brasileiros: TIC Cultura 2020.** 2021. Disponível em: <https://www.cetic.br/pt/tics/cultura/2020/geral/>. Acesso em: 18 jul. 2022.

CETIC.BR. **Pesquisa sobre o uso das tecnologias de informação e comunicação nos equipamentos culturais brasileiros: TIC Cultura 2022.** 2023. Disponível em: <https://www.cetic.br/pt/pesquisa/cultura/indicadores/>. Acesso em: 20 ago. 2023.

COLLECTIONS TRUST. **Standard Procedures for Collections Recording Used in Museums. Spectrum 5**, 2017. a. Disponível em: <https://collectionstrust.org.uk/spectrum/>. Acesso em: 5 jul. 2022.

COLLECTIONS TRUST. **Primary procedures. Spectrum 5**, 2017. b. Disponível em: <https://collectionstrust.org.uk/spectrum/primary-procedures/>. Acesso em: 20 maio. 2023.

COLLECTIONS TRUST. **Appendix - information requirements. Spectrum 5**, 2017. c. Disponível em: <https://collectionstrust.org.uk/spectrum/information-requirements/>. Acesso em: 21 maio. 2023.

COLLECTIONS TRUST. **Object name. Spectrum 5**, 2017. d. Disponível em: <https://collectionstrust.org.uk/resource/object-name/>. Acesso em: 22 maio. 2023.

COLLECTIONS TRUST. **Object Entry. Spectrum 5**, 2017. e. Disponível em: <https://collectionstrust.org.uk/wp-content/uploads/2017/11/Object-entry.pdf>. Acesso em: 4 jun. 2023.

COLLECTIONS TRUST. **Acquisition and accessioning. Spectrum 5**, 2017. f. Disponível em: <https://collectionstrust.org.uk/wp-content/uploads/2017/11/Acquisition-and-accessioning.pdf>. Acesso em: 4 jun. 2023.

COLLECTIONS TRUST. **Location and movement control. Spectrum 5**, 2017. g. Disponível em: <https://collectionstrust.org.uk/wp-content/uploads/2017/11/Location-and-movement-control.pdf>. Acesso em: 4 jun. 2023.

COLLECTIONS TRUST. **Inventory. Spectrum 5**, 2017. h. Disponível em: <https://collectionstrust.org.uk/wp-content/uploads/2017/11/Inventory.pdf>. Acesso em: 4 jun. 2023.

COLLECTIONS TRUST. **Cataloguing. Spectrum 5**, 2017. i. Disponível em: <https://collectionstrust.org.uk/wp-content/uploads/2017/11/Cataloguing.pdf>. Acesso em: 4 jun. 2023.

COLLECTIONS TRUST. **Object exit. Spectrum 5**, 2017. j. Disponível em: <https://collectionstrust.org.uk/wp-content/uploads/2017/11/Object-exit.pdf>. Acesso em: 4 jun. 2023.

COLLECTIONS TRUST. **Loan in (borrowing object). Spectrum 5**, 2017. k. Disponível em: <https://collectionstrust.org.uk/wp-content/uploads/2017/11/Loans-in-borrowing-objects.pdf>. Acesso em: 4 jun. 2023.

COLLECTIONS TRUST. **Documentation planning. Spectrum 5**, 2017. l. Disponível em: <https://collectionstrust.org.uk/wp-content/uploads/2017/11/Documentation-planning.pdf>. Acesso em: 4 jun. 2023.

COLLECTIONS TRUST. **Loans out (lending objects). Spectrum 5**, 2017. m. Disponível em: <https://collectionstrust.org.uk/wp-content/uploads/2017/11/Loans-out-lending-objects.pdf>. Acesso em: 4 jun. 2023.

COLLECTIONS TRUST. **Introduction do Spectrum. Spectrum 5**, 2022. Disponível em: <https://collectionstrust.org.uk/spectrum/spectrum-5/>. Acesso em: 19 maio. 2023.

COLLECTIONS TRUST. **Spectrum Timeline**. [s.d.]. Disponível em:  
<https://collectionstrust.org.uk/resource/spectrum-timeline/>. Acesso em: 20 maio. 2023a.

COLLECTIONS TRUST. **#RethinkingCataloguing**. [s.d.]. Disponível em:  
<https://collectionstrust.org.uk/blog/rethinkingcataloguing/>. Acesso em: 20 maio. 2023b.

CONSELHO INTERNACIONAL DE MUSEUS - ICOM. **Código de ética lusófono**. , 2009. Disponível em: [http://icom.org.br/wp-content/themes/colorwaytheme/pdfs/codigo%20de%20etica/codigo\\_de\\_etica\\_lusofono\\_iii\\_2009.pdf](http://icom.org.br/wp-content/themes/colorwaytheme/pdfs/codigo%20de%20etica/codigo_de_etica_lusofono_iii_2009.pdf). Acesso em: 22 jul. 2021.

CONSELHO INTERNACIONAL DE MUSEUS, ICOM. **Lista de Controle da Norma Object ID**. , 1999. Disponível em: [https://icom.museum/wp-content/uploads/2020/12/ObjectID\\_portuguese.pdf](https://icom.museum/wp-content/uploads/2020/12/ObjectID_portuguese.pdf). Acesso em: 29 jul. 2022.

COSTA, Paulo. **Sinfonia de Objetos**. São Paulo: Iluminuras, 2007.

CYPRESS. **Cypress**. 2017. Disponível em: <https://www.cypress.io>. Acesso em: 30 jul. 2023.

D'ARCUS, Bruce. **Citation Style Language Repository**. 2010. Disponível em:  
<https://github.com/citation-style-language>. Acesso em: 30 jul. 2023.

DERVICHE, André. Pandemia e crise estimulam a inovação em museus. São Paulo, 2021. Disponível em: <https://jornal.usp.br/atualidades/pandemia-e-crise-estimulam-inovacao-em-museus/>. Acesso em: 5 jul. 2022.

DOMINO, DocumentandO Museu IberoamericaNO. , 2022.

DROETTBOOM, Michael. **What is a schema?** 2013. Disponível em: <https://json-schema.org/understanding-json-schema/about.html>. Acesso em: 1 jul. 2023.

DROETTBOOM, Michael. **Understanding JSON Schema**. [s.l.] : Space Telescope Science Institute, 2020. Disponível em: <https://json-schema.org/understanding-json-schema/>. Acesso em: 27 maio. 2023.

EXTREMEPROGRAMMINGROADMAP. **Extreme Programming**. 2006. Disponível em: <http://xp.c2.com/ExtremeProgramming.html>. Acesso em: 3 jun. 2023.

FIELDING, Roy T. Representational State Transfer (REST). *Em: Architectural Styles and the Design of Network-based Softwares Architectures*. Irvine: Information and Computer Science - University of California, 2000. Disponível em: [https://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm). Acesso em: 3 ago. 2022.

GITHUB. **Github**. 2008. Disponível em: <https://github.com>. Acesso em: 13 ago. 2023.

GITHUB. **Github Actions**. 2018. Disponível em: <https://docs.github.com/en/actions>. Acesso em: 30 jul. 2023.

GODINHO, Henrique. **Explore a Coleção! - Casa Museu Ema Klabin**. 2019a. Disponível em: <https://emaklabin.org.br/explore>. Acesso em: 5 jul. 2022.

GODINHO, Henrique. **Núcleos (Explore a Coleção! - Casa Museu Ema Klabin)**. 2019b. Disponível em: <https://emaklabin.org.br/explore/nucleos>. Acesso em: 5 jul. 2022.

HARPRING, Patricia; BACA, Murtha (ORG.). **Categories for the Descriptions of Works of Art**. 2022. Disponível em: [https://www.getty.edu/research/publications/electronic\\_publications/cdwa/](https://www.getty.edu/research/publications/electronic_publications/cdwa/). Acesso em: 20 ago. 2023.

HYKES, Solomon. **Docker**. 2013. Disponível em: <https://www.docker.com>. Acesso em: 30 jul. 2023.

ICOM, Intenational Concoul of Museums-Brasil. **Nova definição de Museu - ICOM**. 2022. Disponível em: [https://www.icom.org.br/?page\\_id=2776](https://www.icom.org.br/?page_id=2776). Acesso em: 29 maio. 2023.

INTERNATIONAL COMMITTEE FOR DOCUMENTATION, CIDOC-ICOM. **DocumentandO Museu IberoamericaNO - DOMINO Working Group**. 2018. Disponível em: <https://cidoc.mini.icom.museum/working-groups/domino/>. Acesso em: 2 ago. 2022.

JSON-LD WORKING GROUP. **JSON-LD 1.1 - A JSON-based Serialization for Linked Data**. 2020. Disponível em: <https://www.w3.org/TR/json-ld11/>. Acesso em: 3 ago. 2022.

LE COADIC, Yves-François. **A Ciência da Informação**. Brasília: Briquet de Lemos, 1994.

LINKED ART. **Linked Art**. 2021a. Disponível em: <https://linked.art/>. Acesso em: 20 maio. 2023.

LINKED ART. **Community - Linked Art**. 2021b. Disponível em: <https://linked.art/community/>. Acesso em: 20 maio. 2023.

LINKED ART. **Linked Art Profile of CIDOC-CRM**. 2021c. Disponível em: <https://linked.art/model/profile/>. Acesso em: 20 maio. 2023.

LINKED ART. **API - Linked Art**. 2021d. Disponível em: <https://linked.art/api/1.0/>. Acesso em: 21 maio. 2023.

LINKED ART. **Linked Art API Endpoints**. 2021e. Disponível em: <https://linked.art/api/1.0/endpoint/>. Acesso em: 21 maio. 2023.

LINKED ART. **Linked Art API: Concept**. 2021f. Disponível em: <https://linked.art/api/1.0/endpoint/concept/>. Acesso em: 2 jun. 2023.

LINKED ART. **Linked Art API: Shared Data Structures**. 2021g. Disponível em: <https://linked.art/api/1.0/shared/>. Acesso em: 2 jun. 2023.

LINKED ART. **Linked Art API: Dimension Structure**. 2021h. Disponível em: <https://linked.art/api/1.0/shared/dimension/>. Acesso em: 2 jun. 2023.

LINKED ART. **Linked Art API: Identifier Structure**. 2021i. Disponível em: <https://linked.art/api/1.0/shared/identifier/>. Acesso em: 2 jun. 2023.

LINKED ART. **Linked Art API: Digital Object**. 2021j. Disponível em: [https://linked.art/api/1.0/endpoint/digital\\_object/](https://linked.art/api/1.0/endpoint/digital_object/). Acesso em: 18 jun. 2023.

LINKED ART. **Linked Art API: Monetary Amount Structure**. 2021k. Disponível em: <https://linked.art/api/1.0/shared/money/>. Acesso em: 14 ago. 2023.

LINKED ART. **Linked Art API: Name Structure**. 2021l. Disponível em:

<https://linked.art/api/1.0/shared/name/>. Acesso em: 14 ago. 2023.

LINKED ART. **Linked Art API: TimeSpan Structure**. 2021m. Disponível em:

<https://linked.art/api/1.0/shared/timespan/>. Acesso em: 14 ago. 2023.

LINKED ART. **Linked Art API: Assigned Relationship Structure**. 2021n. Disponível

em: <https://linked.art/api/1.0/shared/assignment/>. Acesso em: 14 ago. 2023.

LINKED ART. **Linked Art API: Reference Structure**. 2021o. Disponível em:

<https://linked.art/api/1.0/shared/reference/>. Acesso em: 14 ago. 2023.

LINKED ART. **Linked Art API: Event**. 2021p. Disponível em:

<https://linked.art/api/1.0/endpoint/event/>. Acesso em: 14 ago. 2023.

LINKED ART. **Linked Art API: Group**. 2021q. Disponível em:

<https://linked.art/api/1.0/endpoint/group/>. Acesso em: 14 ago. 2023.

LINKED ART. **Linked Art API: Person**. 2021r. Disponível em:

<https://linked.art/api/1.0/endpoint/person/>. Acesso em: 14 ago. 2023.

LINKED ART. **Linked Art API: Physical Object Representation**. 2021s. Disponível em:

[https://linked.art/api/1.0/endpoint/physical\\_object/](https://linked.art/api/1.0/endpoint/physical_object/). Acesso em: 14 ago. 2023.

LINKED ART. **Linked Art API: Place**. 2021t. Disponível em:

<https://linked.art/api/1.0/endpoint/place/>. Acesso em: 14 ago. 2023.

LINKED ART. **Linked Art API: Provenance Activity**. 2021u. Disponível em:

[https://linked.art/api/1.0/endpoint/provenance\\_activity/](https://linked.art/api/1.0/endpoint/provenance_activity/). Acesso em: 14 ago. 2023.

LINKED ART. **Linked Art API: Set**. 2021v. Disponível em:

<https://linked.art/api/1.0/endpoint/set/>. Acesso em: 14 ago. 2023.

LINKED ART. **Linked Art API: Textual Work**. 2021w. Disponível em:

[https://linked.art/api/1.0/endpoint/textual\\_work/](https://linked.art/api/1.0/endpoint/textual_work/). Acesso em: 14 ago. 2023.

LINKED ART. **Linked Art API Visual Work**. 2021x. Disponível em:

[https://linked.art/api/1.0/endpoint/visual\\_work/](https://linked.art/api/1.0/endpoint/visual_work/). Acesso em: 14 ago. 2023.

LINKED ART. **Linked Art API Protocol**. 2021y. Disponível em:  
<https://linked.art/api/1.0/protocol/>. Acesso em: 15 ago. 2023.

LINKED ART. **API Design Principles and Requirements**. 2021z. Disponível em:  
<https://linked.art/api/1.0/principles/>. Acesso em: 15 ago. 2023.

LINKED ART. **Linked Art API: Statement Structure**. Disponível em:  
<https://linked.art/api/1.0/shared/statement/>. Acesso em: 2023-08-14. .

LINKED ART. **Linked Art API: Type Structure**. Disponível em:  
<https://linked.art/api/1.0/shared/type/>. Acesso em: 14 ago. 2023b.

MADURO, Nuno. **Pest**. 2021. Disponível em: <https://pestphp.com>. Acesso em: 30 jul. 2023.

MATOS, Alexandre. Nota sobre a tradução do SPECTRUM em Portugal. *Em*:  
**SPECTRUM 4.0: o padrão para gestão de coleções de museus do Reino Unido**. São Paulo: Secretaria de Estado de Cultura; Associação de Amigos do Museu do Café; Pinacoteca do Estado de São Paulo, 2014. v. 2. Disponível em: [https://spectrum-pt.org/wp-content/uploads/2021/03/Spectrum\\_PT\\_NET.pdf](https://spectrum-pt.org/wp-content/uploads/2021/03/Spectrum_PT_NET.pdf). Acesso em: 19 maio. 2023.

MDN, Mozilla. **JavaScript**. 2022. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em: 2 jun. 2023.

MDN WEB DOCS. **Working with JSON**. 2023. Disponível em:  
<https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>. Acesso em: 21 maio. 2023.

META OPEN SOURCE. **React: The library for web and native user interfaces**. 2023. Disponível em: <https://react.dev>. Acesso em: 19 jun. 2023.

MICROSOFT. **TypeScript 5.1**. 2023a. Disponível em: <https://www.typescriptlang.org>. Acesso em: 2 jun. 2023.

MICROSOFT. **Description of the database normalization basics**. 2023b. Disponível em:  
<https://learn.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description>. Acesso em: 2 jun. 2023.



MICROSOFT. **Guid Estrutura**. [s.d.]. Disponível em: <https://learn.microsoft.com/pt-br/dotnet/api/system.guid?view=net-7.0>. Acesso em: 10 ago. 2023.

MYSQL. **Chapter 11 Data Types**. 2023. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>. Acesso em: 30 jul. 2023.

NAKAZAWA, Cristoph. **Jest**. 2011. Disponível em: <https://jestjs.io>. Acesso em: 30 jul. 2023.

NARWHAL TECHNOLOGIES INC. **monorepo.tools**. 2022. Disponível em: <https://monorepo.tools>. Acesso em: 1 jun. 2023.

PADILHA, Renata Cardozo. **Documentação Museológica e Gestão de Acervo**. Florianópolis: FCC Edições, 2014. v. 2

PETTY, Brian. **dbdelta supporting foreign key**. 2014. Disponível em: <https://core.trac.wordpress.org/ticket/19207#comment:3>. Acesso em: 2 jun. 2023.

PHP, The PHP Group. **PHP**. 2023. Disponível em: <https://www.php.net>. Acesso em: 2 jun. 2023.

SANDERSON, Robert. **Shout It Loud: LOUD - EuropeanaTech 2018 Keynotes**. 2018. Disponível em: <https://youtu.be/r4afi8mGVAY>. Acesso em: 20 maio. 2023.

SARACEVIC, Tefko. Ciência da Informação: origem, evolução e relações. **Perspectivas em Ciência da Informação**, Belo Horizonte, v. 1, p. 41–62, jan./jul. 1996. Disponível em: <http://portaldeperiodicos.eci.ufmg.br/index.php/pci/article/view/235/22>. Acesso em: 15 jul. 2022.

SCHWABER, Ken; SUTHERLAND, Jeff. **The 2020 Scrum Guide**. 2020. Disponível em: <https://scrumguides.org/scrum-guide.html>. Acesso em: 3 jun. 2023.

SEMVER. **Versionamento Semântico 2.0.0**. 2023. Disponível em: <https://semver.org/lang/pt-BR/>. Acesso em: 28 jul. 2023.

SHADOW. **Violation of 1NF in MySQL - Response**. 2017. Disponível em: <https://stackoverflow.com/a/42602932/11085794>. Acesso em: 2 jun. 2023.

SILVA, Camila Aparecida Da. **Esquema de metadados para descrição de obras de arte em museus brasileiros: uma proposta**. 2020. Programa de Pós-Graduação em Ciência da Informação - Escola de Comunicações e Artes / Universidade de São Paulo, São Paulo, 2020.

SPORNY, Manu. **What is JSON-LD?** 2012. Disponível em: <https://www.youtube.com/watch?v=vioCbTo3C-4>. Acesso em: 22 maio. 2023.

SVEIDQVIST, Knut; VINOD, Sidharth. **Mermaid Diagramming and charting tool**. [s.d.]. Acesso em: 6 set. 2023.

TORVALDS, Linus; HAMANO, Junio. **Git**. 2005. Disponível em: <https://git-scm.com>. Acesso em: 13 ago. 2023.

TYPICODE. **Husky Commit**. 2018. Disponível em: <https://typicode.github.io/husky/>. Acesso em: 9 ago. 2023.

VEGA-ALMEIDA, Rosa Lidia; FERNÁNDEZ-MOLINA, Juan Carlos; LINARES, Radamés Colimbié. Coordenadas paradigmáticas, históricas y epistemológicas de la Ciencia de la información: una sistematización. **Information Research**, [S. l.], v. 14, 2009. Disponível em: <https://informationr.net/ir/14-2/paper399.html>.

W3C JSON-LD WORKING GROUP. **JSON for Linking Data**. 2014. Disponível em: <https://json-ld.org>. Acesso em: 3 ago. 2022.

W3C, Word Wide Web Consortium. **WAI-ARIA Overview**. 2022. Disponível em: <https://www.w3.org/WAI/standards-guidelines/aria/>. Acesso em: 25 jun. 2023.

WELLS, Don. **Extreme Programming: a gentle introduction**. 1999. Disponível em: <http://www.extremeprogramming.org>. Acesso em: 3 jun. 2023.

WELLS, Don. **Agile Software Development: A gentle introduction**. 2009. Disponível em: <http://www.agile-process.org>. Acesso em: 3 jun. 2023.

WIKI EXTREME PROGRAMMING. **Relentless Testing**. 2009. Disponível em: <http://wiki.c2.com/?RelentlessTesting>. Acesso em: 3 jun. 2023.

WIKI EXTREME PROGRAMMING. **Extreme Programming For One**. 2014. Disponível em: <http://wiki.c2.com/?ExtremeProgrammingForOne>. Acesso em: 3 jun. 2023.

WORDPRESS. **Escaping Data**. [s.d.]. Disponível em: <https://developer.wordpress.org/apis/security/escaping/>. Acesso em: 10 ago. 2023a.

WORDPRESS. **Sanitizing Data**. [s.d.]. Disponível em: <https://developer.wordpress.org/apis/security/sanitizing/>. Acesso em: 10 ago. 2023b.

WORDPRESS. **Linking and Embedding**. [s.d.]. Disponível em: <https://developer.wordpress.org/rest-api/using-the-rest-api/linking-and-embedding/>. Acesso em: 15 ago. 2023c.

WORDPRESS. **WordPress Rest API Key Concepts**. [s.d.]. Disponível em: <https://developer.wordpress.org/rest-api/key-concepts/>. Acesso em: 16 ago. 2023d.