

Museu Virtual da Emigração e das Comunidades
Projecto Integrado de Engenharia de Linguagens
Relatório Final

Bruno Azevedo and Miguel Costa

*Mestrado em Engenharia Informática,
Departamento de Informática,
Universidade do Minho*

De: 7 de Novembro de 2011
Até: 24 de Julho de 2012

Resumo

Este documento representa toda a documentação feita para o desenvolvimento do Museu Virtual da Emigração e das Comunidades. Neste momento contém essencialmente o caso de estudo definido, análise de requisitos, alguma especificação UML e as principais decisões tomadas até ao momento.

Conteúdo

1	introdução	3
2	Contextualização	4
3	Objectivos pretendidos	4
3.1	Modelo de referência internacional OAIS	5
3.1.1	SIP e o Processo de Ingestão	5
3.1.2	AIP e armazenamento de projetos	5
3.1.3	DIP e a disseminação/publicação de conteúdos	6
4	Caso de Estudo	6
5	Levantamento de Requisitos	7
5.1	Especificação dos Requisitos	8
5.1.1	análise das normas internacionais para a catalogação de obras de arte	8
5.1.2	Dados	8
5.1.3	Pedidos	8
5.1.4	Relações	8
6	Ambiente de Trabalho	8
6.1	Linguagens de Programação	8
6.1.1	Yii Framework	8
6.2	Ferramentas	8
7	Concepção/desenho da Resolução	9
7.1	Base de Dados	9
7.1.1	Contextualização	9
7.1.2	Modelo de Dados	10
7.1.3	Escolha do Motor de Base de Dados	28
7.2	Algoritmos	28
7.2.1	Fotografias	29
7.2.2	Zincogravuras	30
7.3	Plataforma Web	32
7.3.1	Back office	32
7.3.2	Front office	34
7.3.3	Salas de Exposição do Museu	36
8	Trabalho futuro	39
9	Conclusões	39
A	Schema XSD da DSL para especificar uma sala	40

1 introdução

No âmbito da UCE30 Engenharia de Linguagens do Mestrado em Engenharia Informática, é pretendido para o Projeto Integrado que se desenvolva um Museu Virtual (MV).

Este MV terá ainda uma Ontologia¹ associada para a informação que estiver guarda na base de dados esteja relacionada entre si. Através desta Ontologia será possível implementar uma característica importante no projeto, que é gerar automaticamente salas de exposição. Assim, o utilizador poderá fazer uma visita orientada por tema/história.

Os conhecimentos que foram e serão adquiridos nos vários módulos desta UCE30, quando aplicados neste projeto irão permitir-nos fazer ferramentas mais genéricas, eficientes e permitir que este MV possa comunicar com outros já existentes. Isto porque irão ser adoptados modelos internacionais para a recolha e partilha de informação.

¹Ontologia é um termo que foi adoptado pela comunidade de Inteligência Artificial para referir conceitos e termos para descrever alguma área do conhecimento ou construir uma representação desse conhecimento.

2 Contextualização

Hoje em dia, devido ao grande processamento que os computadores possuem, é possível processar enormes quantidades de informação. O problema talvez seja agora arranjar mecanismos de comunicação entre serviços, isto é, encontrar uma linguagem para partilhar informação que qualquer programa consiga entender sem grande esforço.

Devido à dificuldade de transferência de informação, o XML acabou por se tornar uma ótima opção e é considerado o standard para a resolução deste problema, logo, no nosso Museu Virtual (MV), terá de ter mecanismos de recolha de informação de um ficheiro XML de peças para serem expostas e, ainda, exportar essa informação também em XML.

Os Museus para exporem as suas peças, tem de definir regras de catalogação para o tipo de peça que é pretendido descrever, devido à necessidade de uniformizar estas regras, foram criadas diversas normas para os mais diversos objectos que podem ser apresentados num museu, por exemplo, desde monumentos até cartas ou roupa. No nosso caso, vamos adoptar essencialmente os modelos internacionais do CDWA² e do CCO³. Estes modelos contêm uma grande quantidade de elementos que se podem usar, em que alguns deles são obrigatórios, mas para além destes, de acordo com o que precisamos, dos que não eram obrigatórios, seleccionamos aqueles que nos pareceram mais relevantes e úteis para o nosso caso de estudo.

3 Objectivos pretendidos

O objectivo da realização deste projecto é desenvolver um Sistema para Geração Automática de Salas de Exposições em Museus Virtuais com base em Ontologias, ou seja, criação de um WebSite que permita ser gerido como um Museu, seja possível alguns utilizadores colocarem peças para exposição, outros com direitos de administração gerirem as salas e o museu e, como finalidade óbvia, mostrar a qualquer pessoa que entre no WebSite, as peças que estão arquivadas.

Por outras palavras, os objectivos principais são:

- arquivar o espólio do museu,
- gerar automaticamente os espaços de aprendizagem na forma de salas de exposição do museu,
- guiar as visitas recebendo delas informação.

Quando falamos em "arquivar o espólio", estamos a referir-nos a receber meta-informação sobre cada objecto, classificá-lo tendo em conta a ontologia e armazenar essa meta-informação depois de classificada.

O MV vai ter várias salas de exposição, em que cada uma delas transmite o conhecimento que relacionada todos os objectos que pertencem à sala. O tema ou conhecimento que é pretendido ser transmitido, será descrito por uma ontologia. É esta ontologia que define os objectos a serem expostos e as relações entre eles.

Dizer "guiar as visitas recebendo delas informação", é a nossa solução basear-se em apresentar o sítio web com base na ontologia da exposição e permitir que o visitante possa fazer comentários e até fornecer conhecimento que pode ser arquivada no MV, quer acrescentando novos objectos ou ligações, quer corrigindo informações incorrectas.

O nosso repositório para o MV tem de seguir o modelo de referência Internacional OAIS⁴ que indica como se deverá estruturar a Figura 1.

Neste repositório terá de haver três grandes processos:

- ingestão - processo que permite ingerir conteúdo no sistema.

²Categories for the Description of Works of Art (CDWA) - http://www.getty.edu/research/publications/electronic_publications/cdwa/introduction.html

³Cataloging Cultural Objects (CCO) - <http://cco.vrafoundation.org/>

⁴Open Archival Information System - consiste numa organização de pessoas e sistemas que tem a responsabilidade de preservar a informação e torna-la disponível para uma comunidade designada.

- administração - gestão de utilizadores, gestão dos arquivos armazenados...
- disseminação - processo que permite ver o conteúdo armazenado. Pode ser através de uma página web ou da criação de um ficheiro xml.

3.1 Modelo de referência internacional OAIS

O modelo que se vai usar para a recepção, disponibilizar e gerir a informação é o OAIS e possui a estrutura que é ilustrada na Figura 1.

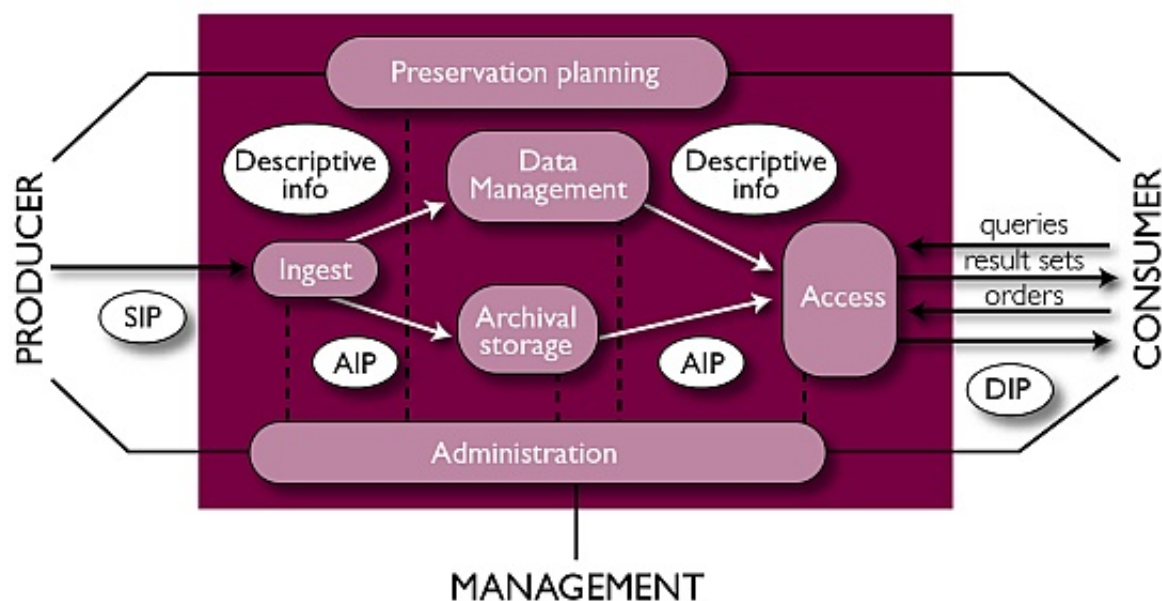


Figura 1: Modelo de referência internacional OAIS

A estrutura baseada neste modelo inclui 3 actores distintos:

- Produtor - corresponde aos utilizadores que irão submeter informação no repositório.
- Administrador - realiza todas as tarefas de manutenção do sistema.
- Consumidor - utilizador que consulta e pesquisa a informação do repositório.

3.1.1 SIP e o Processo de Ingestão

Neste processo, o ideal é conseguir enriquecer o repositório através do preenchimento de um formulário e ainda através de um ficheiro XML que respeite o Schema do CDWA-Lite.

Quando se inserir alguma informação no MV, essa informação irá sofrer um processo até ficar completamente armazenado no sistema para se conseguir relacionar com outras obras presentes no museu.

3.1.2 AIP e armazenamento de projetos

Neste repositório, vamos ter uma solução híbrida, com a metainformação guardada na Base de Dados relacional e a outra informação que corresponde a outros ficheiros incluídos no SIP que estejam relacionados com as obras, como imagens, documentos word, etc, guardados num Sistema de Ficheiros.

3.1.3 DIP e a disseminação/publicação de conteúdos

Um DIP pode ser disponibilizado de duas formas, ou através de um Website criado para o efeito ou então através de um ficheiro XML.

No Website será possível ver o espelho do museu, ver a sua informação e ainda entrar em salas que serão geradas automaticamente para mostrar obras que tinham algum fio de ligação entre elas.

4 Caso de Estudo

Depois de um grande impasse sobre o Museu que iríamos tratar, optamos por um projeto já com algum desenvolvimento que é o Museu da Emigração e das Comunidades de Fafe.

Dentro deste Museu existe uma grande variedade de peças de arte, em que grande parte estão relacionadas com a emigração de portugueses no início do século XX. Muitas das relíquias pertenceram a "Brasileiros retorna viagem", pessoas que emigraram para o Brasil, enriqueceram e quando voltaram para Portugal queriam ser importantes na sua terra, para isso mandavam construir edifícios (escolas, hospitais...) e alguns até tinha o seu próprio jornal/almanaque.

Do vasto material que poderíamos tratar, optámos pelas **fotografias** e pelas **zincogravuras**.

As fotografias foram tiradas pelo fotógrafo e pintor Gérald Bloncourt, de nacionalidade Haitiana, que tirou muitas fotografias ao dia a dia dos emigrantes portugueses em França.



Quanto às zincogravuras, elas foram usadas para diversos fins, como cartões em almanaques, capas de jornais, fotografias de famílias e retratos, logótipos...



Figura 2: RANILON - Correia plana de plástico e couro



Figura 3: "Director deste Almanaque, esposa e seus 10 filhos" - O Desforço

5 Levantamento de Requisitos

Na explicação do que era pretendido fazer, já descrevemos um pouco os requisitos necessários, fazendo uma síntese dos objectivos temos então:

1. Basear a especificação no estudo e análise das normas internacionais: CCO e CDWA.
2. Criar um modelo para uma Base de Dados Relacional que permita guardar a informação proveniente de documentos com os formatos internacionais já mencionados.
3. Tendo em conta o caso de estudo, encontrar uma solução para inserir a informação no repositório de forma a que as diferentes obras se possam relacionar entre si.

5.1 Especificação dos Requisitos

5.1.1 análise das normas internacionais para a catalogação de obras de arte

5.1.2 Dados

5.1.3 Pedidos

5.1.4 Relações

6 Ambiente de Trabalho

Para um projecto de uma dimensão considerável como este, é necessário procurar boas ferramentas que nos facilitem a execução das tarefas e ainda usar linguagens de programação que sejam "eficazes" para a implementação do nosso sistema.

6.1 Linguagens de Programação

O nosso sistema baseia-se essencialmente numa plataforma Web e a linguagem adoptada foi o PHP. No entanto, desenvolver um Website agradável requer o uso de outras linguagens como por exemplo JavaScript essencial para a interação do utilizador com a página.

De experiência que tínhamos da utilização do PHP, havia tarefas chatas e repetitivas, por isso procuramos uma framework que nos facilitasse essas tarefas e acabamos por escolher a Yii Framework.

6.1.1 Yii Framework

Esta framework adopta a metodologia MVC, já tem operações de caching, autenticação, controlo de acessos por utilizadores diferentes, scaffolding e testing. Foi essencialmente por causa destas características que escolhemos utilizar esta framework.

Extensões utilizadas

A Yii Framework tem uma grande biblioteca de extensões para integrar no site, nós tiramos proveito de algumas:

- **Gii** - A partir da Base de Dados criamos todos o código essencial para começar a trabalhar, models e as operações CRUD.
- **Yii-user** - essencial para controlo de utilizadores no site (registar, alterar dados pessoais e eliminar utilizadores).
- **comment** - extensão para controlar os comentários numa página.
- **cssmenu** - Para criar uma barra de menus mais funcional.
- **gmap** - Inserção do Google Maps nas nossas páginas.
- **Etooltip** - Mostrar as descrições das imagens.

Algumas destas extensões tiveram de criar tabelas na Base de Dados para guardar a informação que precisavam, por exemplo dos utilizadores e dos comentários.

6.2 Ferramentas

As ferramentas usadas foram:

- **Eclipse** - Essencialmente para criar as páginas Web.
- **Vim** - Para a criação das scripts em perl.
- **OxygenXML** - Para todas as tarefas que estivessem relacionadas com o XML.
- **TexMaker** - Fazer toda a documentação em latex.

7 Concepção/desenho da Resolução

7.1 Base de Dados

Neste capítulo, iremos apresentar uma contextualização sobre a necessidade desta base de dados e a nossa solução final de base de dados. Para isso, enunciaremos as tabelas criadas e explicaremos a sua razão de ser.

7.1.1 Contextualização

Uma das funcionalidades exigidas neste projeto é o arquivamento do espólio do museu. Arquivar o espólio significa receber meta-informação sobre cada objeto do espólio (eventualmente o próprio objeto se ele for digital), classificá-lo à luz de uma ontologia definida à partida (mas que poderá ir evoluindo) e armazenar a meta-informação após classificação.

Outra grande funcionalidade exigida é a geração automática de espaços de aprendizagem na forma de salas de exposição do museu. Cada sala de exposição de um Museu Virtual (MV) é um sítio Web através do qual se podem ver os objetos em exposição, ou a sua descrição, devidamente relacionados entre si de forma a transmitir o conhecimento correspondente ao tema dessa exposição. Estes espaços de aprendizagem também necessitam de ser armazenados.

Finalmente, a terceira grande funcionalidade passa por guiar as visitas pelas salas de exposição recebendo delas informação. Guiar a visita significa apresentar o sítio criado com base na ontologia da exposição e as indicações relativas à disposição dos objetos na sala e permitir que o visitante da página possa deixar comentários.

O repositório que irá armazenar toda esta informação, será constituído por uma base de dados relacional assim como num conjunto de diretorias no servidor da aplicação. Neste capítulo, ir-nos-emos focar no desenvolvimento da base de dados.

A especificação do modelo de dados relacional foi baseado no estudo das seguintes normas *Categories for the Description of Works of Art (CDWA)* e *Cataloging Cultural Objects (CCO)* e *CDWA Lite*.

O CDWA descreve o conteúdo de bases de dados de arte articulando uma *framework* conceptual para descrever e aceder a informação sobre obras de arte, arquitetura, outro materiais de cultura, grupos e coleções de obras e imagens relacionadas. O CDWA inclui 532 categorias e subcategorias.

O CCO é um guia ao invés de um conjunto de elementos de metadados. Os elementos que abrange são referentes a áreas de informação num registo de catalogação que podem ser mapeadas para vários elementos de metadados tais como CDWA ou CDWA Lite. Basicamente, é um documento lato que inclui regras para formatação de dados, sugestões para informação obrigatória, requisitos para vocabulário controlado e problemas de apresentação.

Para facilitar a partilha de registos complacentes com CDWA através do protocolo *Open Archives Initiative ? Protocol for Metadata Harvesting (OAI-PMH)*, o Getty Research Institute criou o schema XML CDWA Lite baseado nas categorias consideradas essenciais (core) do CDWA que descreve um formato para registos essenciais de obras de arte e material cultural baseados no CDWA e CCO.

Após uma análise a todas estas normas e estruturas, chegámos a um modelo final que passaremos a descrever na secção seguinte.

7.1.2 Modelo de Dados

O modelo representa uma simplificação do que é sugerido pelo CDWA, apenas representando os metadados core e outros que achámos que seriam interessantes e importantes para a descrição de uma obra de arte. Trata-se então de uma estrutura capaz de representar tudo que é necessário para cobrir os requisitos apresentados.

Para representar a primeira funcionalidade enunciada anteriormente, descreveremos cada uma das tabelas criadas assim como a(s) categoria(s) do CDWA que pretendem representar.

As chaves primárias são representadas por atributos sublinhados e as chaves estrangeiras por atributos em itálico.

Tabelas com origem em entidades e relacionamentos 1:N

As seguintes tabelas foram criadas com base nos metadados que pretendemos armazenar:

- **Object_Work_Records** {*id_object_Work_Records*, *displayCreator*, *displayMeasurements*, *displayMaterialsTech*, *displayCreationDate*, *RecordType*}

Correspondente à categoria **Object/Work**. Esta tabela identifica o foco lógico da discussão. Representa uma obra de arte (objeto) no repositório, descreve o que é a obra e possibilita a procura de obras de um determinados tipo.

Atributos:

- ***id_object_Work_Records***: é chave primária, já que identifica univocamente cada registo. Inteiro que é auto incrementado;
- ***displayCreator***: string que identifica o nome, informação biográfica breve e papéis do criador ou criadores no desenho e produção da obra, apresentada numa sintaxe adequada para exibição ao end-user e incluindo quaisquer indicações de incerteza, ambiguidade e nuance.
- ***DisplayMeasurements***: string que representa a informação sobre as dimensões, tamanho ou escala da obra. Apresentada numa sintaxe adequada para exibição ao end-user e incluindo quaisquer indicações de incerteza, ambiguidade e nuance.
- ***DisplayMaterialsTech***: string que representa uma indicação das substâncias ou materiais usados na criação de uma obra, assim como quaisquer técnicas de implementação, produção ou manufatura, processos ou métodos incorporados no seu fabrico. Apresentada numa sintaxe adequada para exibição ao end-user e incluindo quaisquer indicações de incerteza, ambiguidade e nuance.
- ***DisplayCreationDate***: uma string que representa uma descrição concisa da data ou âmbito de datas associadas com a criação, desenho, produção, exibição, performance, construção ou alteração da obra ou os seus componentes. Apresentada numa sintaxe adequada para exibição ao end-user e incluindo quaisquer indicações de incerteza, ambiguidade e nuance.
- ***RecordType***: é chave estrangeira da tabela RecordType. Inteiro que identifica o tipo de de registo da obra.

- **Object_Work_Types** {*id_type*, *type*, *termsource*, *termsourceID*}

Correspondente à categoria **Object/Work Type**. Esta tabela representa um termo que identifica o tipo específico de objeto ou obra descrita.

Atributos:

- ***id_type***: é chave primária, já que identifica univocamente cada tipo. Inteiro que é auto incrementado;
- ***type***: string que representa um termo que identifica o tipo específico de objeto ou obra descrita.
- ***termsource***: string que identifica a fonte do termo.
- ***termsourceID***: string que representa o identificador do termo na fonte.

- **Object_Work_Titles** {*id_object_Work_Titles*, *title*, *pref*, *type*, *lang*, *langtermsource*, *Object_Work_Record*}

Correspondente à categoria **Titles or Names**. Esta tabela representa os títulos ou nomes dados a uma obra de arte, arquitetura, ou grupo, assim como o tipo do título e as datas quando o título era válido.

Atributos:

- **id_object_Work_Titles:** é chave primária, já que identifica univocamente cada título. Inteiro que é auto incrementado;
- **title:** string que identifica os títulos ou nomes dados a uma obra de arte, arquitetura, cultura material.
- **pref:** string que representa uma indicação de um título ser o preferido da obra ou não.
- **type:** string que representa o tipo do título ou nome atribuído a uma obra.
- **lang:** string que representa a língua do título ou nome.
- **langtermsource:** string que representa a fonte do termo que identifica a língua.
- **Object_Work_Record:** é chave estrangeira da tabela *Object_Work_Record*. Inteiro que identifica a obra.

- **SourceTitles** {id_sourceTitles, sourceTitle}

Esta tabela representa a fonte para o título, genericamente uma fonte publicada.

Atributos:

- **id_sourceTitles:** é chave primária, já que identifica univocamente cada fonte. Inteiro que é auto incrementado;
- **sourceTitle:** string que identifica a fonte para o título, geralmente uma fonte publicada.

- **IndexingMeasurements** {id_indexingMeasurements, *Object_Work_Record*}

Correspondente à categoria **Measurements**. Esta tabela representa as dimensões, tamanho, forma, escala, formato, ou configuração de armazenamento, incluindo volume, peso, área ou tempo de execução.

Atributos:

- **id_indexingMeasurements:** é chave primária, já que identifica univocamente cada grupo de medidas. Inteiro que é auto incrementado;
- **Object_Work_Record:** é chave estrangeira da tabela *Object_Work_Record*. Inteiro que identifica a obra.

- **Measurements** {id_measurements, value, unit, type, *IndexingMeasurement*}

Correspondente às categorias **Dimensions Type**, **Dimensions Value** e **Dimensions Unit**. Esta tabela representa as dimensões ou outras medidas de outros aspetos de uma obra.

Atributos:

- **id_measurements:** é chave primária, já que identifica univocamente cada grupo de medidas. Inteiro que é auto incrementado;
- **value:** string que representa o valor numérico de uma dimensão em específico de uma obra;
- **unit:** string que representa unidade de medida usada;
- **type:** string que representa o tipo de dimensão de uma área em específico ou parte de uma obra.
- **Object_Work_Record:** é chave estrangeira da tabela *Object_Work_Record*. Inteiro que identifica a obra.

- **QualifierMeasurements** {id_qualifierMeasurements, qualifierMeasurement}

Correspondente à categoria **Dimensions Qualifier**. Esta tabela representa uma palavra ou frase que elabora na natureza da obra quando necessário, tal como quando as medidas são próximas.

Atributos:

- **id_qualifierMeasurements:** é chave primária, já que identifica univocamente cada qualificador. Inteiro que é auto incrementado;
 - **qualifierMeasurement:** string que representa uma palavra ou frase que elabora na natureza da obra quando necessário, tal como quando as medidas são próximas.
- **IndexingMaterialsTech** {id_indexingMaterialsTech, type, *Object_Work_Record*}
- Correspondente à categoria **Materials/Techniques**. Esta tabela representa materiais e técnicas indexadas com termos controlados para pesquisa.
- Atributos:
- **id_indexingMaterialsTech:** é chave primária, já que identifica univocamente cada grupo de materiais. Inteiro que é auto incrementado;
 - **type:** string que representa o papel que indica se os termos se referem a um meio ou um suporte para materiais, ou a uma técnica ou utensílio.
 - **Object_Work_Record:** é chave estrangeira da tabela *Object_Work_Record*. Inteiro que identifica a obra.
- **TermMaterialsTech** {id_termMaterialsTech, termMaterialsTech, termsource, termsourceID}
- Correspondente à categoria **Materials/Techniques Description**. Esta tabela representa um termo que indexa um material ou técnica.
- Atributos:
- **id_termMaterialsTech:** é chave primária, já que identifica univocamente cada termo. Inteiro que é auto incrementado;
 - **termMaterialsTech:** string que representa um termo que indexa um material ou técnica
 - **termsource:** string que identifica a fonte do termo.
 - **termsourceID:** string que representa o identificador do termo na fonte.
- **ExtentMaterialsTech** {id_extentMaterialsTech, extentMaterialsTech, *IndexingMaterialsTech*}
- Correspondente à categoria **Materials/Techniques Extent**. Esta tabela representa uma explicação da parte da obra à qual os materiais ou técnicas se aplicam.
- Atributos:
- **id_extentMaterialsTech:** é chave primária, já que identifica univocamente cada explicação. Inteiro que é auto incrementado;
 - **extentMaterialsTech:** string que representa uma explicação da parte da obra à qual os materiais ou técnicas se aplicam.
 - **IndexingMaterialsTech:** é chave estrangeira da tabela *IndexingMaterialsTech*. Inteiro que identifica o material/técnica.
- **ConservationTreatmentHistory** {id_conservationTreatmentHistory, *treatmentDate*, *Object_Work_Record*}
- Correspondente à categoria **Conservation/Treatment History**. Esta tabela representa o histórico dos procedimentos ou ações a que uma obra foi sujeita para repará-la, conservá-la ou estabilizá-la.
- Atributos:
- **id_conservationTreatmentHistory:** é chave primária, já que identifica univocamente cada procedimento. Inteiro que é auto incrementado;
 - **treatmentDate:** é chave estrangeira da tabela *TreatmentDates*. Inteiro que identifica a data de tratamento.
 - **Object_Work_Record:** é chave estrangeira da tabela *Object_Work_Record*. Inteiro que identifica a obra.

- **TreatmentTypes** {id_treatmentTypes, treatmentType}

Correspondente à categoria **Treatment Type**. Esta tabela representa o nome do tratamento de conservação ou procedimento de restauração técnica/científica realizado na obra.

Atributos:

- **id_treatmentTypes**: é chave primária, já que identifica univocamente cada nome. Inteiro que é auto incrementado;
- **treatmentType**: string que representa o nome do tratamento de conservação ou procedimento de restauração técnica/científica realizado na obra.
- **ConservationTreatmentDescriptions** {id_conservationTreatmentDescriptions, ConservationTreatmentDescription, *ConservationTreatmentHistory*}

Correspondente à categoria **Conservation/Treatment Description**. Esta tabela representa uma descrição em prosa de procedimentos ou ações a que uma obra foi sujeita para repará-la, conservá-la ou estabilizá-la.

Atributos:

- **id_conservationTreatmentDescriptions**: é chave primária, já que identifica univocamente cada descrição. Inteiro que é auto incrementado;
 - **ConservationTreatmentDescription**: string que representa uma descrição em prosa de procedimentos ou ações a que uma obra foi sujeita para repará-la, conservá-la ou estabilizá-la.
 - **ConservationTreatmentHistory**: é chave estrangeira da tabela ConservationTreatmentHistory. Inteiro que identifica o histórico.
 - **TreatmentDates** {id_treatmentDates, treatmentDate, *earliestDate*, *latestDate*}
- Correspondente à categoria **Treatment Date**. Esta tabela representa a data em que um determinado procedimento ou tratamento se realizou.
- Atributos:
- **id_treatmentDates**: é chave primária, já que identifica univocamente cada data. Inteiro que é auto incrementado;
 - **treatmentDate**: string que representa a data em que um determinado procedimento ou tratamento se realizou.
 - **earliestDate**: é chave estrangeira da tabela EarliestDates. Inteiro que identifica a data inferior.
 - **latestDate**: é chave estrangeira da tabela LatestDates. Inteiro que identifica a data mais posterior.

- **EarliestDates** {id_earliestDate, earliestDate, termsource}

Correspondente à categoria **Earliest Date**. Esta tabela representa um ano que amplamente delimita o início de um espaço de tempo entre datas.

Atributos:

- **id_earliestDate**: é chave primária, já que identifica univocamente cada ano. Inteiro que é auto incrementado;
- **earliestDate**: string que representa um ano que amplamente delimita o início de um espaço de tempo entre datas.
- **termsource**: string que identifica a fonte do termo.
- **LatestDates**{id_latestDate, latestDate, termsource}

Correspondente à categoria **Latest Date**. Esta tabela representa um ano que amplamente delimita o final de um espaço de tempo entre datas.

Atributos:

- **id_latestDate:** é chave primária, já que identifica univocamente cada ano. Inteiro que é auto incrementado;
 - **latestDate:** string que representa um ano que amplamente delimita o final de um espaço de tempo entre datas.
 - **termSource:** string que identifica a fonte do termo.
- **IndexingCreators** {id_indexingCreators, genderCreator}

Correspondente à categoria **Creator Description**. Esta tabela representa os criadores de uma obra.

Atributos:

 - **id_indexingCreators:** é chave primária, já que identifica univocamente cada criador. Inteiro que é auto incrementado;
 - **genderCreator:** string que identifica o sexo do criador.
- **NamesCreator** {id_namesCreator, nameCreator, type, termSource, termSourceID}

Correspondente à categoria **Creator Description**. Esta tabela representa os nomes, alcunhas ou outros identificadores de um indivíduo, grupo de pessoas, firma ou outra entidade que contribuiu para o desenho, criação, produção, manufatura ou alteração da obra.

Atributos:

 - **id_namesCreator:** é chave primária, já que identifica univocamente cada nome. Inteiro que é auto incrementado;
 - **nameCreator:** string que representa os nomes, alcunhas ou outros identificadores de um indivíduo, grupo de pessoas, firma ou outra entidade que contribuiu para o desenho, criação, produção, manufatura ou alteração da obra.
 - **type:** string que identifica se o nome corresponde a uma pessoa singular ou coletiva.
 - **termSource:** string que identifica a fonte do nome.
 - **termSourceID:** string que representa o identificador do nome na fonte.
- **SourceNamesCreator** {id_sourceNamesCreator, sourceNameCreator}

Esta tabela representa a fonte para o nome, genericamente uma fonte publicada.

Atributos:

 - **id_sourceNamesCreator:** é chave primária, já que identifica univocamente cada fonte. Inteiro que é auto incrementado;
 - **sourceNameCreator:** string que identifica a fonte para o nome, geralmente uma fonte publicada.
- **NationalitiesCreator** {id_nationalitiesCreator, nationalitycreator}

Correspondente à categoria **Creator Description**. Esta tabela representa a afiliação nacional ou cultural da pessoa singular ou coletiva.

Atributos:

 - **id_nationalitiesCreator:** é chave primária, já que identifica univocamente cada nacionalidade. Inteiro que é auto incrementado;
 - **nationalitycreator:** string que representa a afiliação nacional ou cultural da pessoa singular ou coletiva.
- **vitalDatesCreator** {id_vitalDatesCreator, vitalDatesCreator, birthDate, deathDate, termSource, *IndexingCreator*}

Correspondente à categoria **Creator Description**. Esta tabela representa uma descrição do tempo de vida da pessoa singular ou da existência da pessoa coletiva.

Atributos:

- **id_vitalDatesCreator:** é chave primária, já que identifica univocamente cada descrição. Inteiro que é auto incrementado;
 - **vitalDatesCreator:** string que representa uma descrição do tempo de vida da pessoa singular ou da existência da pessoa coletiva.
 - **birthDate:** string que identifica o ano de nascimento do criador.
 - **deathDate:** string que identifica o ano de morte do criador.
 - **termsource:** string que identifica a fonte do termo.
 - **IndexingCreator:** é chave estrangeira da tabela IndexingCreators. Inteiro que identifica o criador.
- **CreatorRoles** {id_rolesCreator, roleCreator, termsource, termsourceID}

Correspondente à categoria **Creator Description**. Esta tabela representa o papel do criador ou outro agente na criação ou produção da obra.

Atributos:

 - **id_rolesCreator:** é chave primária, já que identifica univocamente cada papel. Inteiro que é auto incrementado;
 - **roleCreator:** string que representa o papel do criador ou outro agente na criação ou produção da obra.
 - **termsource:** string que identifica a fonte do papel.
 - **termsourceID:** string que representa o identificador do papel na fonte.
- **Styles** {id_styles, style, termsource, termsourceID}

Correspondente à categoria **Styles/Periods Indexing Terms**. Esta tabela representa o termo que identifica estilo, período artístico ou histórico, movimento, grupo ou escola cujas características estão representadas na obra.

Atributos:

 - **id_styles:** é chave primária, já que identifica univocamente cada estilo. Inteiro que é auto incrementado;
 - **style:** string que representa o termo que identifica estilo, período artístico ou histórico, movimento, grupo ou escola cujas características estão representadas na obra.
 - **termsource:** string que identifica a fonte do estilo.
 - **termsourceID:** string que representa o identificador do estilo na fonte.
- **Cultures** {id_cultures, culture, termsource, termsourceID}

Correspondente à categoria **Object/Work Culture**. Esta tabela representa o nome da cultura, povo, ou nacionalidade da qual a obra é originária.

Atributos:

 - **id_cultures:** é chave primária, já que identifica univocamente cada cultura. Inteiro que é auto incrementado;
 - **culture:** string que representa o nome da cultura, povo, ou nacionalidade da qual a obra é originária.
 - **termsource:** string que identifica a fonte da cultura.
 - **termsourceID:** string que representa o identificador da cultura na fonte.
- **IndexingDates** {id_indexingDates, dateQualifier, *earliestDate*, *latestDate*, *Object_Work_Record*}

Correspondente à categoria **Creation Date**. Esta tabela representa um conjunto de anos que delimitam a duração da criação e produção da obra.

Atributos:

- **id_indexingDates:** é chave primária, já que identifica univocamente cada conjunto de anos. Inteiro que é auto incrementado;
 - **dateQualifier:** string que representa um esclarecimento do significado da data.
 - **earliestDate:** é chave estrangeira da tabela EarliestDates. Inteiro que identifica a data inferior.
 - **latestDate:** é chave estrangeira da tabela LatestDates. Inteiro que identifica a data mais posterior.
 - **Object_Work_Record:** é chave estrangeira da tabela Object_Work_Record. Inteiro que identifica a obra.
- **CreationPlaces** {id_creationPlaces, creationPlace, termsource, termsourceID, placeQualifier, *Object_Work_Record*}

Correspondente à categoria **Creation Place/Original Location**. Esta tabela representa a localização onde a criação, desenho ou produção da obra ou os seus componentes se realizou, ou a localização original da obra.

Atributos:

- **id_creationPlaces:** é chave primária, já que identifica univocamente cada localização. Inteiro que é auto incrementado;
 - **creationPlace:** string que representa a localização onde a criação, desenho ou produção da obra ou os seus componentes se realizou, ou a localização original da obra.
 - **termsource:** string que identifica a fonte da localização.
 - **termsourceID:** string que representa o identificador da localização na fonte.
 - **placeQualifier:** string que representa um esclarecimento do significado do lugar ou localização.
 - **Object_Work_Record:** é chave estrangeira da tabela Object_Work_Record. Inteiro que identifica a obra.
- **Provenance** {id_provenance, provenanceDescription, cost, legalStatus, *Object_Work_Record*, *Owner*, *ownershipDate*, *OwnershipPlace*}

Correspondente à categoria **Ownership/Collecting History**. Esta tabela representa a proveniência ou história dos proprietários de uma obra de arte, obra arquitetónica ou grupo desde a sua criação até ao presente.

Atributos:

- **id_provenance:** é chave primária, já que identifica univocamente cada proveniência. Inteiro que é auto incrementado;
 - **provenanceDescription:** string que representa uma descrição em prosa da proveniência ou história dos proprietários de uma obra de arte, obra arquitetónica ou grupo.
 - **cost:** string que representa o valor monetário de uma obra numa determinada moeda na altura da transferência de propriedade.
 - **legalStatus:** string que representa o estatuto jurídico da obra.
 - **Object_Work_Record:** é chave estrangeira da tabela Object_Work_Record. Inteiro que identifica a obra.
 - **Owner:** é chave estrangeira da tabela Owners. Inteiro que identifica o proprietário.
 - **OwnershipDate:** é chave estrangeira da tabela OwnershipDates. Inteiro que identifica o período de posse da obra.
 - **OwnershipPlace:** é chave estrangeira da tabela OwnershipPlaces. Inteiro que identifica a localização da obra durante a posse.
- **TransferModes** {id_transferMode, transferMode, *Provenance*}

Correspondente à categoria **Transfer Mode**. Esta tabela representa o meio (modo de transferência) pelo qual uma obra entrou na coleção de uma pessoa singular ou coletiva.

Atributos:

- **id_transferMode:** é chave primária, já que identifica univocamente cada modo de transferência. Inteiro que é auto incrementado;
- **transferMode:** string que representa o meio (modo de transferência) pelo qual uma obra entrou na coleção de uma pessoa singular ou coletiva.
- **Provenance:** é chave estrangeira da tabela Provenance. Inteiro que identifica a proveniência.

- **Owners** {id_owner, owner, termsource, termsourceID, *OwnerRolesid_ownerRole*}

Correspondente à categoria **Owner/Agent**. Esta tabela representa o nome de uma pessoa singular ou coletiva que era proprietária ou estava na posse da obra, ou serviu como agente na sua transferência entre proprietários.

Atributos:

- **id_owner:** é chave primária, já que identifica univocamente cada nome. Inteiro que é auto incrementado;
- **owner:** string que representa o nome de uma pessoa singular ou coletiva que era proprietária ou estava na posse da obra, ou serviu como agente na sua transferência entre proprietários.
- **termsource:** string que identifica a fonte do nome.
- **termsourceID:** string que representa o identificador do nome na fonte.
- **OwnerRolesid_ownerRole:** é chave estrangeira da tabela OwnerRoles. Inteiro que identifica o papel do proprietário.

- **OwnerRoles** {id_ownerRole, ownerRole, termsource, termsourceID}

Correspondente à categoria **Owner/Agent Role**. Esta tabela representa o papel desempenhado por uma pessoa singular ou coletiva no que diz respeito à propriedade, posse ou transferência de proprietário da obra.

Atributos:

- **id_ownerRole:** é chave primária, já que identifica univocamente cada papel. Inteiro que é auto incrementado;
- **ownerRole:** string que representa o papel desempenhado por uma pessoa singular ou coletiva no que diz respeito à propriedade, posse ou transferência de proprietário da obra.
- **termsource:** string que identifica a fonte do papel.
- **termsourceID:** string que representa o identificador do papel na fonte.

- **OwnershipDates** {id_ownershipDate, ownershipDate, *earliestDate*, *latestDate*}

Correspondente à categoria **Ownership Date**. Esta tabela representa o período de tempo durante o qual pertenceu ou estava na posse de determinado proprietário ou agente.

Atributos:

- **id_ownershipDate:** é chave primária, já que identifica univocamente cada período de tempo;
- **ownershipDate:** string que representa o período de tempo durante o qual pertenceu ou estava na posse de determinado proprietário ou agente.
- **earliestDate:** é chave estrangeira da tabela EarliestDates. Inteiro que identifica a data inferior.
- **latestDate:** é chave estrangeira da tabela LatestDates. Inteiro que identifica a data mais posterior.

- **OwnershipPlaces** {id_ownershipPlaces, ownershipPlace, termsource, termsourceID}

Correspondente à categoria **Ownership Place**. Esta tabela representa o local onde a obra estava alojada aquando da posse por determinado proprietário.

Atributos:

- **id_ownershipPlaces:** é chave primária, já que identifica univocamente cada local. Inteiro que é auto incrementado;
 - **ownershipPlace:** string que representa o local onde a obra estava alojada aquando da posse por determinado proprietário.
 - **termsource:** string que identifica a fonte do do local.
 - **termsourceID:** string que representa o identificador do local na fonte.
- **ExhibitionHistory** {id_exhibitionHistory, exhibitionDescription, exhibitionObjectLabel, remarks, *exhibitionType*, *Object_Work_Record*}

Correspondente à categoria **Exhibition/Loan History**. Esta tabela representa um registo histórico de uma exposição pública de uma obra.

Atributos:

- **id_exhibitionHistory:** é chave primária, já que identifica univocamente cada exposição. Inteiro que é auto incrementado;
 - **exhibitionDescription:** string que representa uma descrição incluindo o título ou nome da exposição, a sua localização e outra informação pertinente.
 - **exhibitionObjectLabel:** string que representa um display que identifica a obra em exposição.
 - **remarks:** string que representa notas adicionais ou comentários pertinentes à informação.
 - **ExhibitionType:** é chave estrangeira da tabela ExhibitionTypes. Inteiro que identifica o tipo da exposição.
 - **Object_Work_Record:** é chave estrangeira da tabela Object_Work_Record. Inteiro que identifica a obra.
- **ExhibitionTitles** {id_exhibitionTitles, *ExhibitionHistory*, exhibitionTitle}

Correspondente à categoria **Exhibition Title or Name**. Esta tabela representa o título ou nome das exposições tal como formuladas pela instituição organizadora.

Atributos:

- **id_exhibitionTitles:** é chave primária, já que identifica univocamente cada título. Inteiro que é auto incrementado;
 - **exhibitionTitle:** string que representa o título ou nome das exposições tal como formuladas pela instituição organizadora.
 - **ExhibitionHistory:** é chave estrangeira da tabela ExhibitionHistory. Inteiro que identifica a exposição.
- **ExhibitionTypes** {id_exhibitionTypes, exhibitionType}

Correspondente à categoria **Exhibition Type**. Esta tabela representa uma indicação do tipo de exposição ou empréstimo.

Atributos:

- **id_exhibitionTypes:** é chave primária, já que identifica univocamente cada tipo. Inteiro que é auto incrementado;
 - **exhibitionType:** string que representa uma indicação do tipo de exposição ou empréstimo.
- **ExhibitionVenues** {id_exhibitionVenues, *ExhibitionHistory*, exhibitionVenue, *ExhibitionVenueDate*} Correspondente à categoria **Exhibition Venue**. Esta tabela representa uma apresentação dos nomes, locais e datas de quando a exposição ou obra estava em exibição.

Atributos:

- **id_exhibitionVenues:** é chave primária, já que identifica univocamente cada apresentação. Inteiro que é auto incrementado;
- **exhibitionVenue:** string que representa uma apresentação dos nomes, locais e datas de quando a exposição ou obra estava em exibição
- **ExhibitionHistory:** é chave estrangeira da tabela ExhibitionHistory. Inteiro que identifica a exposição.
- **ExhibitionVenueDate:** é chave estrangeira da tabela ExhibitionVenueDates. Inteiro que identifica o período de exposição.

- **ExhibitionVenueNames** {id_exhibitionVenueNames, exhibitionVenueName, exhibitionVenuePlace, nameTermSource, nameTermSourceID, placeTermSource, placeTermSourceID, *ExhibitionVenue*}

Correspondente à categoria **Venue Name/Place**. Esta tabela representa o nome da instituição, galeria, outra instalação e o a localização geográfica onde uma exposição teve lugar.

Atributos:

- **id_exhibitionVenueNames:** é chave primária, já que identifica univocamente cada nome. Inteiro que é auto incrementado;
 - **exhibitionVenueName:** string que representa o nome da instituição, galeria ou outra instalação onde uma exposição teve lugar.
 - **ExhibitionVenueName:** string que representa a localização geográfica onde uma exposição teve lugar.
 - **nameTermSource:** string que identifica a fonte do nome.
 - **nameTermSourceID:** string que representa o identificador do nome na fonte.
 - **placeTermSource:** string que identifica a fonte da localização geográfica.
 - **placeTermSourceID:** string que representa o identificador da localização geográfica na fonte.
 - **ExhibitionVenue:** é chave estrangeira da tabela ExhibitionVenues. Inteiro que identifica a apresentação do nome.
- **ExhibitionVenueDates** {id_ExhibitionVenueDate, exhibitionVenueDate, *earliestDate*, *latestDate*} Correspondente à categoria **Venue Date**. Esta tabela representa uma descrição da data ou âmbito de datas de quando a obra estava em exposição numa determinada venue.

Atributos:

- **id_ExhibitionVenueDate:** é chave primária, já que identifica univocamente cada descrição;
 - **exhibitionVenueDate:** string que representa uma descrição da data ou âmbito de datas de quando a obra estava em exposição numa determinada venue.
 - **earliestDate:** é chave estrangeira da tabela EarliestDates. Inteiro que identifica a data inferior.
 - **latestDate:** é chave estrangeira da tabela LatestDates. Inteiro que identifica a data mais posterior.
- **ExhibitionOrganizers** {id_exhibitionOrganizer, exhibitionOrganizerName, exhibitionOrganizerLocation, nameTermSource, nameTermSourceID, locationTermSource, locationTermSourceID, *ExhibitionHistory*}
- Correspondente à categoria **Exhibition Organizer**. Esta tabela representa o nome e localização da pessoa ou agência responsável pela exposição ou empréstimo.

Atributos:

- **id_exhibitionOrganizer:** é chave primária, já que identifica univocamente cada nome e localização. Inteiro que é auto incrementado;
- **exhibitionOrganizerName:** string que representa o nome da pessoa ou agência responsável pela exposição ou empréstimo.

- **exhibitionOrganizerLocation:** string que representa a localização da pessoa ou agência responsável pela exposição ou empréstimo.
- **nameTermsource:** string que identifica a fonte do nome.
- **nameTermsourceID:** string que representa o identificador do nome na fonte.
- **locationTermsource:** string que identifica a fonte da localização geográfica.
- **locationTermsourceID:** string que representa o identificador da localização geográfica na fonte.
- **ExhibitionHistory:** é chave estrangeira da tabela ExhibitionHistory. Inteiro que identifica a exposição.

- **Locations** {id_locations, *Object_Work_Record*, *LocationName*}

Correspondente à categoria **Current Location**. Esta tabela representa a localização do repositório atual de uma obra.

Atributos:

- **id_locations:** é chave primária, já que identifica univocamente cada localização. Inteiro que é auto incrementado;
- **Object_Work_Record:** é chave estrangeira da tabela Object_Work_Record. Inteiro que identifica a obra.
- **LocationName:** é chave estrangeira da tabela LocationsName. Inteiro que identifica o nome da localização.

- **LocationsName** {id_locationsName, locationName, type, termsource, termsourceID, locID, locIDtype}

Correspondente à categoria **Current Repository/ Geographic Location**. Esta tabela representa o nome do repositório que atualmente aloja a obra e a localização geográfica onde a obra se encontra.

Atributos:

- **id_locationsName:** é chave primária, já que identifica univocamente cada nome. Inteiro que é auto incrementado;
- **locationName:** string que representa o nome do repositório que atualmente aloja a obra e a localização geográfica onde a obra se encontra.
- **type:** string que representa o tipo do repositório.
- **termsource:** string que identifica a fonte do nome.
- **termsourceID:** string que representa o identificador do nome na fonte.
- **locID:** string que identifica o código que identifica univocamente o repositório.
- **locIDtype:** string que representa o tipo do identificador único do repositório.

- **WorkIDs** {id_workIDs, workID, type, *Location*}

Correspondente à categoria **Object/Work Label/Identification**. Esta tabela representa qualquer identificador numérico ou alfa numérico atribuído à obra.

Atributos:

- **id_workIDs:** é chave primária, já que identifica univocamente cada identificador. Inteiro que é auto incrementado;
- **workID:** string que representa qualquer identificador numérico ou alfa numérico atribuído à obra.
- **type:** string que representa o tipo do identificador.
- **Location:** é chave estrangeira da tabela Locations. Inteiro que identifica a localização do repositório.

- **IndexingSubjects** {id_indexingSubjects, type, extentSubject, *Object_Work_Record*}

Correspondente à categoria **Subject Matter**. Esta tabela representa um conjunto de termos que refletem o que é descrito na obra e pela obra.

Atributos:

- **id_indexingSubjects**: é chave primária, já que identifica univocamente cada conjunto de termos. Inteiro que é auto incrementado;
- **type**: string que serve para distinguir entre o que é descrito na obra (descrição e identificação) ou o que é descrito pela obra (interpretação).
- **extentSubject**: string que representa a parte da obra às quais os termos se aplicam.
- **Object_Work_Record**: é chave estrangeira da tabela *Object_Work_Record*. Inteiro que identifica a obra.

- **SubjectTerms** {id_subjectTerms, subjectTerm, type, termsource, termsourceID}

Correspondente à categoria **General Subject Terms**. Esta tabela representa os termos que identificam, descrevem e/ou interpretam o que é descrito na e pela obra.

Atributos:

- **id_subjectTerms**: é chave primária, já que identifica univocamente cada termo. Inteiro que é auto incrementado;
- **subjectTerm**: string que representa o termo que identifica, descreve e/ou interpreta o que é descrito na e pela obra.
- **type**: string que representa o tipo do termo.
- **termsource**: string que identifica a fonte do termo.
- **termsourceID**: string que representa o identificador do termo na fonte.

- **Classifications** {id_classifications, classification, termsource, termsourceID}

Correspondente à categoria **Classification**. Esta tabela representa os termos utilizados para categorizar um obra, agrupando-a com outras obras com base em características semelhantes.

Atributos:

- **id_classifications**: é chave primária, já que identifica univocamente cada termo. Inteiro que é auto incrementado;
- **classification**: string que representa o termo utilizado para categorizar um obra, agrupando-a com outras obras com base em características semelhantes.
- **termsource**: string que identifica a fonte do termo.
- **termsourceID**: string que representa o identificador do termo na fonte.

- **DescriptiveNotes** {id_descriptiveNotes, descriptiveNote, *Object_Work_Record*}

Correspondente à categoria **Descriptive Note**. Esta tabela representa um texto breve que descreve o conteúdo e contexto da obra.

Atributos:

- **id_descriptiveNotes**: é chave primária, já que identifica univocamente cada texto. Inteiro que é auto incrementado;
- **descriptiveNote**: string que representa um texto breve que descreve o conteúdo e contexto da obra.
- **Object_Work_Record**: é chave estrangeira da tabela *Object_Work_Record*. Inteiro que identifica a obra.

- **DescriptiveNoteSources** {id_descriptiveNoteSources, descriptiveNoteSource}

Esta tabela representa a fonte para a nota descritiva, geralmente uma fonte publicada.

Atributos:

- **id_descriptiveNoteSources:** é chave primária, já que identifica univocamente cada fonte. Inteiro que é auto incrementado;
- **descriptiveNoteSource:** string que identifica a fonte para a nota descritiva, geralmente uma fonte publicada.

- **Inscriptions** {id_inscriptions, inscriptions, *Object_Work_Record*} Correspondente à categoria **Inscriptions/Marks**. Esta tabela representa uma descrição ou transcrição de qualquer inscrição física distinguível ou identificável, anotações, textos, marcas ou etiquetas que são afixadas, aplicadas, estampadas, escritas, inscritas ou anexadas à obra.

Atributos:

- **id_inscriptions:** é chave primária, já que identifica univocamente cada inscrição. Inteiro que é auto incrementado;
- **inscriptions:** string que representa uma descrição ou transcrição de qualquer inscrição física distinguível ou identificável, anotações, textos, marcas ou etiquetas que são afixadas, aplicadas, estampadas, escritas, inscritas ou anexadas à obra.
- **Object_Work_Record:** é chave estrangeira da tabela *Object_Work_Record*. Inteiro que identifica a obra.

- **RelatedWorks** {id_relatedWorks, relatedWorkRelType, *Object_Work_Record*} Correspondente à categoria **Related Works**. Esta tabela representa obras de arte ou arquitetônicas relacionadas com a obra a ser descrita e uma descrição do relacionamento entre as obras.

Atributos:

- **id_relatedWorks:** é chave primária, já que identifica univocamente cada obra relacionada. Inteiro que é auto incrementado;
- **relatedWorkRelType:** string que representa um termo que descreve a natureza do relacionamento entre a obra a ser descrita e a entidade relacionada.
- **Object_Work_Record:** é chave estrangeira da tabela *Object_Work_Record*. Inteiro que identifica a obra a ser descrita.

- **LinksRelatedWork** {id_linksRelatedWork, linkRelatedWork, linkscheme, *RelatedWork*}

Esta tabela representa uma referência uri/url que é universal e é acessível no ambiente worldwide web.

Atributos:

- **id_linksRelatedWork:** é chave primária, já que identifica univocamente cada referência. Inteiro que é auto incrementado;
- **linkRelatedWork:** string que representa uma referência uri/url que é universal e é acessível no ambiente worldwide web.
- **linkscheme:** string que representa o esquema do link.
- **Object_Work_Record:** é chave estrangeira da tabela *Object_Work_Record*. Inteiro que identifica a obra a ser descrita.

- **LabelRelatedWork** {id_labelRelatedWork, labelRelatedWork, *RelatedWork*}

Correspondente à categoria **Related Work Label/Identification**. Esta tabela representa uma identificação da obra relacionada, grupo, coleção ou série que terá significado para os end-users.

Atributos:

- **id_labelRelatedWork:** é chave primária, já que identifica univocamente cada identificação. Inteiro que é auto incrementado;
 - **labelRelatedWork:** string que representa uma identificação da obra relacionada, grupo, coleção ou série que terá significado para os end-users.
 - **RelatedWork:** é chave estrangeira da tabela RelatedWorks. Inteiro que identifica a obra relacionada.
- **LocationsRelatedWork** {id_locationsRelatedWork, locationRelatedWork, relWorkID, locID, locIDtype, termsource, *RelatedWork*}
 Esta tabela representa a localização atual da obra relacionada, geralmente um repositório.
 Atributos:
 - **id_locationsRelatedWork:** é chave primária, já que identifica univocamente cada localização. Inteiro que é auto incrementado;
 - **locationRelatedWork:** string que representa a localização atual da obra relacionada, geralmente um repositório.
 - **relWorkID:** string que representa o identificador único numérico ou alfa numérico da obra relacionada.
 - **locID:** string que representa um código único que identifica o repositório.
 - **locIDtype:** string que representa a fonte que fornece o locID.
 - **termsource:** string que identifica a fonte do termo.
 - **RelatedWork:** é chave estrangeira da tabela RelatedWorks. Inteiro que identifica a obra relacionada.
- **RightsWork** {id_rightsWork, rightsWork, type, *Object_Work_Record*}
 Correspondente à categoria **Copyright/Restrictions**. Esta tabela representa informação sobre direitos de gestão.
 Atributos:
 - **id_rightsWork:** é chave primária, já que identifica univocamente cada informação. Inteiro que é auto incrementado;
 - **rightsWork:** string que representa informação sobre direitos de gestão.
 - **type:** string que representa o tipo da informação de direitos de gestão.
 - **Object_Work_Record:** é chave estrangeira da tabela Object_Work_Record. Inteiro que identifica a obra a ser descrita.
- **RecordsID** {id_recordsID, recordID, type, *Object_Work_Record*}
 Correspondente à categoria **Object/Work Record ID**. Esta tabela representa um identificador único de registo no sistema.
 Atributos:
 - **id_recordsID:** é chave primária, já que identifica univocamente cada identificador. Inteiro que é auto incrementado;
 - **recordID:** string que representa um identificador único de registo no sistema.
 - **type:** string que representa o tipo do identificador.
 - **Object_Work_Record:** é chave estrangeira da tabela Object_Work_Record. Inteiro que identifica a obra a ser descrita.
- **RecordSources** {id_recordSources, recordSource, *Object_Work_Record*}
 Esta tabela representa a fonte de informação neste registo, geralmente o repositório ou outra instituição.
 Atributos:

- **id_recordSources:** é chave primária, já que identifica univocamente cada fonte. Inteiro que é auto incrementado;
 - **recordSource:** string que representa a fonte de informação neste registo, geralmente o repositório ou outra instituição.
 - **Object_Work_Record:** é chave estrangeira da tabela Object_Work_Record. Inteiro que identifica a obra a ser descrita.
- **RecordInfo** {id_recordInfo, type, *Object_Work_Record*}
Esta tabela representa informação de metadados deste registo.
Atributos:
 - **id_recordInfo:** é chave primária, já que identifica univocamente cada grupo de informação. Inteiro que é auto incrementado;
 - **type:** string que representa o tipo de informação.
 - **Object_Work_Record:** é chave estrangeira da tabela Object_Work_Record. Inteiro que identifica a obra a ser descrita.
- **RecordsInfoID** {recordInfoID, type, *RecordInfo*}
Esta tabela representa o identificador único para os metadados. Possui a mesma definição que o *Record-sID*, mas fora do contexto de um sistema local.
Atributos:
 - **recordInfoID:** é chave primária, já que identifica univocamente cada identificador. String definida pelo utilizador;
 - **type:** string que representa o tipo do identificador.
 - **RecordInfo:** é chave estrangeira da tabela RecordInfo. Inteiro que identifica a informação.
- **RecordInfoLinks** {id_recordInfoLinks, recordInfoLink, *RecordInfo*}
Esta tabela representa um link para os metadados (não se trata do mesmo link do objeto).
Atributos:
 - **id_recordInfoLinks:** é chave primária, já que identifica univocamente cada link. Inteiro que é auto incrementado;
 - **recordInfoLink:** string que representa um link para os metadados.
 - **RecordInfo:** é chave estrangeira da tabela RecordInfo. Inteiro que identifica a informação.
- **RecordsRelID** {recordRelID, type, *RecordInfo*}
Esta tabela representa um identificador único dos metadados do objeto relacionado.
Atributos:
 - **recordRelID:** é chave primária, já que identifica univocamente cada identificador. String definida pelo utilizador;
 - **type:** string que representa o tipo do identificador.
 - **RecordInfo:** é chave estrangeira da tabela RecordInfo. Inteiro que identifica a informação.
- **RecordMetadataLocations** {id_recordMetadataLocations, recordMetadataLocation, type, *RecordInfo*}
Esta tabela representa um apontador para outros metadados (administrativos, técnicos, estruturais).
Atributos:

- **id_recordMetadataLocations:** é chave primária, já que identifica univocamente cada apontador. Inteiro que é auto incrementado;
 - **recordMetadataLocation:** string que representa um apontador para outros metadados (administrativos, técnicos, estruturais).
 - **type:** string que representa o tipo do apontador.
 - **RecordInfo:** é chave estrangeira da tabela RecordInfo. Inteiro que identifica a informação.
- **RecordMetadataDates** {id_recordMetadataDates, recordMetadataDate, type, termsource, *RecordInfo*}
Esta tabela representa a data de criação ou a data de modificação do registo.
Atributos:
 - **id_recordMetadataDates:** é chave primária, já que identifica univocamente cada data. Inteiro que é auto incrementado;
 - **recordMetadataDate:** string que representa a data de criação ou a data de modificação do registo.
 - **type:** string que representa o tipo da data.
 - **termSource:** string que representa a fonte da data.
 - **RecordInfo:** é chave estrangeira da tabela RecordInfo. Inteiro que identifica a informação.
- **RecordTypes** {id_recordTypes, recordType}
Esta tabela representa um termo que estabelece se o registo representa um item individual ou uma coleção, série ou grupo de obras.
Atributos:
 - **id_recordTypes:** é chave primária, já que identifica univocamente cada termo. Inteiro que é auto incrementado;
 - **recordType:** string que representa um termo que estabelece se o registo representa um item individual ou uma coleção, série ou grupo de obras.
- **Resources** {id_resources, *LinkResource*, *ResourceID*, *ResourceViewDescription*, *Object_Work_Record*}
Esta tabela representa informação sobre imagens ou outros recursos que servem como substitutos visuais da obra.
Atributos:
 - **id_resources:** é chave primária, já que identifica univocamente cada grupo de informação. Inteiro que é auto incrementado;
 - **LinkResource:** é chave estrangeira da tabela LinkResources. Inteiro que identifica a referência.
 - **ResourceID:** é chave estrangeira da tabela ResourcesID. String que identifica o identificador do recurso.
 - **ResourceViewDescription:** é chave estrangeira da tabela ResourceViewDescription. Inteiro que identifica a descrição.
 - **Object_Work_Record:** é chave estrangeira da tabela Object_Work_Record. Inteiro que identifica a obra a ser descrita.
- **LinkResources** {id_linkResources, linkResource, type, formatResource}
Esta tabela representa uma referência uri/url que é universal e é acessível no ambiente worlwide web.
Atributos:
 - **id_linkResources:** é chave primária, já que identifica univocamente cada referência. Inteiro que é auto incrementado;

- **linkResource:** string que representa uma referência uri/url que é universal e é acessível no ambiente worlwide web.
 - **type:** string que indica o tipo do recurso.
 - **formatResource:** string que indica o formato do recurso.
- **ResourcesID** {resourceID, type}
Esta tabela representa um identificador único numérico ou alfa numérico do recurso.
Atributos:
 - **resourceID:** é chave primária, já que identifica univocamente cada identificador. String definida pelo utilizador.
 - **type:** string que representa o tipo de identificador.
 - **ResourceRelTypes** {id_resourceRelType, resourceRelType} Esta tabela representa o relacionamento entre o recurso e a obra descrita.
Atributos:
 - **id_resourceRelType:** é chave primária, já que identifica univocamente cada relacionamento. Inteiro que é auto incrementado;
 - **resourceRelType:** string que representa o relacionamento entre o recurso e a obra descrita.
 - **ResourceTypes** {id_resourceTypes, resourceType, termsource, termsourceID}
Esta tabela representa a identificação genérica do meio da imagem ou outro recurso.
Atributos:
 - **id_resourceTypes:** é chave primária, já que identifica univocamente cada identificação. Inteiro que é auto incrementado;
 - **resourceType:** string que representa a identificação genérica do meio da imagem ou outro recurso.
 - **termsource:** string que identifica a fonte do termo.
 - **termsourceID:** string que representa o identificador do termo na fonte.
 - **ResourceViewDescriptions** {id_resourceViewDescriptions, resourceViewDescription}
Esta tabela representa uma descrição dos aspetos espaciais, cronológicos ou contextuais da obra, tal como é vista neste recurso em específico.
Atributos:
 - **id_resourceViewDescriptions:** é chave primária, já que identifica univocamente cada descrição. Inteiro que é auto incrementado;
 - **resourceViewDescription:** string que representa uma descrição dos aspetos espaciais, cronológicos ou contextuais da obra, tal como é vista neste recurso em específico.

Tabelas com origem em entidades e relacionamentos N:M

As tabelas seguintes, resultam de relacionamentos N:M existentes entre as tabelas apresentadas anteriormente:

- **Object_Work_Types_Object_Work_Records** { *Object_Work_Type*, *Object_Work_Record* }
Resultante do relacionamento N:M entre as tabelas Object_Work_Records e Object_Work_Types.
- **SourceTitle_Object_Work_Titles** { *SourceTitle*, *Object_Work_Title* }
Resultante do relacionamento N:M entre as tabelas SourceTitles e Object_Work_Titles.

- **QualifierMeasurements_IndexingMeasurements** { *QualifierMeasurement*, *IndexingMeasurement* }
Resultante do relacionamento N:M entre as tabelas QualifierMeasurements e IndexingMeasurements.
- **TermMaterialsTech_IndexingMaterialsTech** { *TermMaterialsTech*, *IndexingMaterialsTech* }
Resultante do relacionamento N:M entre as tabelas TermMaterialsTech e IndexingMaterialsTech.
- **TreatmentTypes_ConservationTreatmentHistory** { *TreatmentType*, *ConservationTreatmentHistory* }
Resultante do relacionamento N:M entre as tabelas TreatmentTypes e ConservationTreatmentHistory.
- **Object_Work_Records_IndexingCreators** { *Object_Work_Record*, *IndexingCreator* }
Resultante do relacionamento N:M entre as tabelas Object_Work_Records e IndexingCreators.
- **NamesCreator_IndexingCreators** { *NameCreator*, *IndexingCreator* }
Resultante do relacionamento N:M entre as tabelas NamesCreator e IndexingCreators.
- **SourceNamesCreator_NamesCreator** { *SourceNamesCreator*, *NamesCreator* }
Resultante do relacionamento N:M entre as tabelas SourceNamesCreator e IndexingCreators.
- **IndexingCreators_NationalitiesCreator** { *IndexingCreator*, *NationalityCreator* }
Resultante do relacionamento N:M entre as tabelas IndexingCreators e NationalitiesCreator.
- **IndexingCreators_RolesCreator** { *IndexingCreator*, *CreatorRole* }
Resultante do relacionamento N:M entre as tabelas IndexingCreators e CreatorRoles.
- **Styles_Object_Work_Records** { *Style*, *Object_Work_Record* }
Resultante do relacionamento N:M entre as tabelas Styles e Object_Work_Records.
- **Cultures_Object_Work_Records** { *Culture*, *Object_Work_Record* }
Resultante do relacionamento N:M entre as tabelas Cultures e Object_Work_Records.
- **IndexingSubjects_SubjectTerms** { *IndexingSubject*, *SubjectTerm* }
Resultante do relacionamento N:M entre as tabelas IndexingSubjects e SubjectTerms.
- **Object_Work_Records_Classifications** { *Object_Work_Record*, *Classification* }
Resultante do relacionamento N:M entre as tabelas Object_Work_Records e Classifications.
- **DescriptiveNotes_DescriptiveNoteSources** { *DescriptiveNote*, *DescriptiveNoteSource* }
Resultante do relacionamento N:M entre as tabelas DescriptiveNotes e DescriptiveNoteSources.
- **ResourceRelTypes_Resources** { *ResourceRelType*, *Resource* }
Resultante do relacionamento N:M entre as tabelas ResourceRelTypes e Resources.
- **ResourceTypes_Resources** { *ResourceType*, *Resource* }
Resultante do relacionamento N:M entre as tabelas ResourceTypes e Resources.

Tabelas para a geração de salas de exposição Estas tabelas resultam da necessidade de armazenar os metadados relativos às exposições e às respectivas salas.

- **Exhibitions** {id_exhibition, name, description, image_path, deleted}

Armazena os metadados relativos a uma exposição.

Atributos:

- **id_exhibition:** é chave primária, já que identifica univocamente cada exposição. Inteiro que é auto incrementado;
- **name:** string que representa o nome dado à exposição;
- **description:** string que representa uma descrição do que trata a exposição;
- **image_path:** string que representa o nome atribuído automaticamente à imagem associada à exposição;
- **deleted:** inteiro que indica se a exposição foi removida ou não.

- **Rooms** {id_room, name, description, path, image_path, deleted}

Armazena os emta-dados relativos a uma sala de exposição.

Atributos:

- **id_room:** é chave primária, já que identifica univocamente cada sala. Inteiro que é auto incrementado;
- **name:** string que representa o nome dado à sala;
- **description:** string que representa uma descrição do que trata a sala;
- **path:** string que representa o caminho no repositório para o local onde a imagem foi armazenada;
- **image_path:** string que representa o nome atribuído automaticamente à imagem associada à sala;
- **deleted:** inteiro que indica se a sala foi removida ou não.

- **Exhibitions_Rooms** {Exhibitionsid_exhibition, Roomsid_room, ord_nr, ord_def_by_user}

Resultante do relacionamento N:M entre as tabelas Exhibitions e Rooms.

Atributos:

- **ord_nr:** inteiro que representa o número de ordenação de uma sala numa exposição;
- **ord_def_by_user:** inteiro que indica se o número de ordenação foi atribuído automaticamente ou definido pelo utilizador.

- **Exhibitions_SubjectTerms** {Exhibitionsid_exhibition, SubjectTermsid_subjectTerms}

Resultante do relacionamento N:M entre as tabelas Exhibitions e SubjectTerms.

7.1.3 Escolha do Motor de Base de Dados

A escolha do motor de Base de Dados a utilizar revelou-se imediata visto que estamos a utilizar o web server Apache XAMPP, que contém MySQL, PHP e Perl, daí que o motor escolhido foi MySQL. Para além disso, este motor fornece capacidades de escalabilidade, controlo de concorrência e é open source.

7.2 Algoritmos

Nesta secção são apresentados os diferentes casos referentes aos metadados já existentes no formato digital do Museu da Emigração e das Comunidades de Fafe e o respetivos processos de extração, transformação e carregamento para o repositório.

7.2.1 Fotografias

Todas as fotografias que nos foram disponibilizadas, tinham a sua metainformação num documento do Word e tivemos também acesso as imagens que representavam as fotografias. O nome dos ficheiros de imagem era o n.º de inventário que estava descrito no documento do Word, por isso tornou-se relativamente simples associar uma imagem com a respectiva metainformação.

Extração

A informação que existia no documento do Word tinha uma estrutura bem delineada por isso a parte de extração da informação acabou por não ser muito complexa.

Exemplo do que estava no documento Word:

Inventário	CGB003
Classificação Genérica	Fotografia
Objecto/Documento	Fotografia P/B
Título / Id. pelo Autor	3416/26a
Autor/Produção	Gérald Bloncourt
Material/Suporte	Papel
Técnica	Fotografia P/B ? Impressão digital
Dimensões	42 x 59,5 cm
Legenda	Nas montanhas - região de Chaves ? as casas de madeira e está frio. Esta mulher decorou as paredes com jornais e organizou as painéis e as tampas com decoração! Os utensílios estão presos, ela não os usa... Para mim, trata-se de um testemunho de resistência, de coragem! Um exemplo magnífico de "cultura"!... 1966
Datação	Impressão em 2009
Lugar de Produção	França
História do objecto	Sem moldura. Não foi exposto neste Museu.
Incorporação	Doação do autor ao Museu das Migrações no ano de 2009.

Para facilitar o trabalho de recolher toda a informação e depois inserir na Base de Dados, através do ficheiro do Word geremos um ficheiro XML com uma estrutura feita por nós.

```
<catalogo>
  <peca>
    <inventario>CGB001</inventario>
    <classificacao_generica>Fotografia</classificacao_generica>
    <objecto>Fotografia P/B</objecto>
    <titulo>1111/4</titulo>
    <autor>Gérald Bloncourt</autor>
    <suporte>Papel</suporte>
    <tecnica>Fotografia P/B ? Impressão digital</tecnica>
    <dimensoes>42 x 59,5 cm</dimensoes>
    <legenda>Emigrante português lendo um jornal do seu país num estaleiro de obras nos subúrbios de Pa
    <datacao>Impressão em 2009</datacao>
    <lugar_producao>França</lugar_producao>
    <historia>Sem moldura. Não foi exposto neste Museu.</historia>
    <incorporacao>Doação do autor ao Museu das Migrações no ano de 2009.</incorporacao>
  </peca>
  ...
</catalogo>
```

Transformação

A partir deste ficheiro XML criado por nós foi mais simples gerar outro ficheiro XML de acordo com o

Schema CDWA-Lite e foi a partir deste que a inserção na Base de Dados foi gerada. Assim, é possível aproveitar o código para inserir outras peças de acordo com a norma CDWA.

Carregamento para o repositório

O carregamento para a nossa Base de Dados foi feita gerando comandos SQL e depois executados esses comandos na respectiva BD para o repositório. Os comandos SQL foram gerados a partir do ficheiro CDWA.

7.2.2 Zincogravuras

Os recursos fornecidos que continham os metadados de zincogravuras eram constituídos por um único documento Word contendo metadados referentes a 933 zincogravuras e uma pasta contendo as imagens de cada uma das zincogravuras com a particularidade do nome de cada ficheiro corresponder ao número de inventário da zincogravura correspondente o que facilita a associação entre ambos.

Extração

O documento Word fornecido continha os metadados bem estruturados, sendo que cada zincogravura fazia-se distinguir de outras através da utilização de múltiplas linhas em branco, desta forma detetámos o elemento de quebra para cada zincogravura. De seguida, era necessário identificar os atributos e respetivos valores para cada zincogravura. Esta parte também se revelou fácil, visto que cada atributo mantinha sempre a mesma label e desta forma detetámos elementos para extrair os metadados.

Para proceder à parte da extração criámos uma script utilizando a linguagem de programação Perl visto tratar-se uma linguagem que torna fácil este tipo de processos. Para extrair o texto do corpo do documento Word, utilizámos um módulo Perl (`Text::Extract::Word`) que o possibilita através de uma única chamada a uma função fornecida por esse módulo. As seguintes linhas de código ilustram essa extração:

```
my $file = Text::Extract::Word->new($origin); # Passed either a file name or an open file
handle, this constructor returns an instance that can be used to query the file contents.

my $body = $file->get_body(); # Returns the text for the main body of the Word document.
This excludes headers, footers, and annotations.
```

Transformação

Tendo acesso ao texto extraído do documento em memória, o processo de transformação pode ser iniciado.

Começámos por fazer uma limpeza de certos elementos textuais que dificultavam o parsing dos elementos identificados anteriormente. E de seguida efetuámos uma substituição das labels de cada atributo por outras mais limpas e mais simples sem caracteres especiais (ex: '/') ou acentos. Esta transformação também foi necessária porque as labels originais foram, possivelmente, escritas à mão e existiam alguns casos em que estas nem sempre eram iguais possuindo, por exemplo, um acento numa zincogravura e noutra não, ou abreviações de palavras numas e noutras não.

Este processo não foi obtido apenas com uma iteração, uma análise cuidada aos dados originais e observação dos resultados obtidos da transformação em cada nova iteração, levou a que pequenas alterações fossem feitas a este processo, obtendo no final o seguinte código:

```
$body =~ s/`+//g; # Limpa lixo
$body =~ s/\n[ \t]+\n//g; # Limpa lixo
$body =~ s/\n{2,}/\n\n/g; # Limpa \n a mais
$body =~ s/Denominação\T[ií]tulo/dt/g; #
$body =~ s/Autor\Produção/ap/g; #
$body =~ s/Datação/dat/g; #
$body =~ s/Dimensões/dim/g; #
$body =~ s/[ \t]*(Nº|Número) de Inventário/ni/g; #
$body =~ s/Descrição/desc/g; #
$body =~ s/Impress[ao] no/imp:/g; #
$body =~ s/Proveniência:?\Incorporação/pi/g; #
```

```
$body =~ s/&/&amp;/g; #
```

```
$body = trim($body);
```

Carregamento para documento XML Após a transformação dos metadados originais e estando estes acessíveis em memória, só restava o carregamento destes para uma estrutura. Como adotámos a norma CDWA e nos baseámos no schema CDWA Lite, optámos por carregá-los para um documento XML que fosse válido segundo esse schema.

Este processo consiste, então, na iteração de todas as linhas do documento Word que está em memória. Se o elemento de quebra não for detetado numa dessas linhas então trata-se de um atributo de uma zincogravura e este é inserido numa hash com a seguinte estrutura (label_atributo => valor_atributo), a utilidade desta hash será explicada de seguida.

No entanto, se for detetado o elemento de quebra, significa que todos os elementos de uma zincogravura já foram iterados e estes estão presentes na hash mencionada anteriormente. Então, esta hash é utilizada para alimentar uma função de escrita destes atributos de uma zincogravura no ficheiro XML final. Após esta escrita, a hash é ?limpa? para que os atributos da próxima zincogravura possam ser adicionados.

Eis a função que executa este processo:

```
# Escreve para um ficheiro xml ($destiny) o documento Word estruturado segundo o cdwa-lite.
Invoca funcoes auxiliares.
sub writeGravs {
    my $text_word = shift;
    my %cdwa = ();

    open (F, ">:utf8",$destiny);
    print F "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n";
    print F "<cdwaliteWrap xmlns=\"http://www.getty.edu/CDWA/CDWALite\"
>\n";

    # divide cada linha do documento word. ainda nao tem informacao do que é ou nao uma peça
    my @line_text_word = split "\n", $text_word;
    if (scalar @line_text_word != 0) {
        foreach(@line_text_word) {
            # se varios \s forem encontrados (elemento de quebra; o que distingue uma peça
            # de outra no documento) processa os elementos capturados ate a quebra e sao
            # impressos no formato XML
            if (/^\s*$/){
                print F "<cdwalite>\n";
                print F printGravura(\%cdwa);
                print F "</cdwalite>\n";
                %cdwa = ();
            }
            # caso contrario, trata-se de um campo da mesma peça da iteracao anterior
            elsif($_ =~ /([^\s:]+):(.*)/) {
                my $value = trim($2);
                %cdwa = getValue(\%cdwa, $1, $value);
            }
        }
    }
    print F "</cdwaliteWrap>";
    close F;
}
```

Finalmente falta apenas fazer uma pequena explicação da função que escreve uma zincogravura no ficheiro XML.

Nesta função é feito um mapeamento entre os atributos presentes na hash passada à função e os elementos correspondentes no schema CDWA Lite. Os atributos fornecidos no documento Word não possuem todos os elementos obrigatórios segundo o CDWA, por isso alguns elementos, que apesar de não possuírem valor, são escritos no ficheiro XML final.

É feita uma grande utilização de expressões regulares sobre os valores dos atributos, como por exemplo, no atributo correspondente às medidas, para que se possa saber quais os valores correspondentes à altura, largura ou mesmo diâmetro e qual a unidade desse valor. Também, para extrair os metadados referentes às obras relacionadas, ou seja, onde e quando é que a zincogravura foi impressa.

Carregamento para o repositório

Após criação do documento XML, resta apenas o carregamento destes metadados para o repositório, utilizando para isso o documento XML. Este carregamento é feito através de outra script Perl. Nesta script foi utilizada um módulo Perl (XML::Simple) para que fosse possível transformar o documento XML num objeto em memória, assim como a sua navegação. Como esta script será utilizada após a script correspondente à dos metadados das fotografias, são utilizados alguns contadores que iniciam com determinados valores e servirão para povoar os identificadores de algumas tabelas na base de dados. Este processo decorre através da iteração de cada elemento correspondente a uma obra de arte. Em cada iteração, é feito um mapeamento entre cada elemento (se este possuir valor) e uma instrução SQL (MySQL).

Ambas as scripts podem ser consultadas na pasta Código/Scripts.


7.3 Plataforma Web

7.3.1 Back office

Um dos principais requisitos é a parte de gestão e manutenção de toda a plataforma, nós com a ajuda da Yii Framework implementámos todas essas funcionalidades. O back office baseia-se essencialmente em formulários que permitem enriquecer, ou mesmo alterar/eliminar, o conteúdo disponível no MV.

Tendo em conta que usámos a metodologia MVC, algumas das tarefas necessárias foram geradas recorrendo à ajuda da própria framework. Através da Gii (uma extensão da Yii Framework) é possível gerar algumas tarefas básicas, designadas por CRUD (acrónimo de Create, Read, Update e Delete), para todas as tabelas que existiam na Base de Dados, foram geradas estas operações sendo assim possível logo à partida conseguir manusear a informação no repositório.

Apesar de terem sido gerados os formulários para as várias operações, foi necessário criar novos ou fazer alterações nos que já existiam. As alterações nos que já existiam foram essencialmente a nível de chaves estrangeiras, ou seja, em vez de se ter de colocar o valor número correspondente ao ID de algo que se pretendia associar, em alguns casos colocamos comboboxs para o utilizador seleccionar a opção pretendida.



Museu da Emigração e das Comunidades

[Início](#) | [Conteúdo](#) | [Exposições](#) | [Acerca](#) | [Contactos](#) | [Procurar](#) [Logout \(museu\)](#) | [Perfil](#)

[Home](#) » [Peças](#) » Criar Ficha Completa

Criar uma Ficha Completa para uma Peça

*Fields with * are required.*

Tipo de peça *

Título *

▼ Detalhes Título

Preferência

Tipo

Língua

Criador *

Data de Criação *

Data Inferior *
 ▼

Operações

- [Índice](#)
- [Criar](#)
- [Importar obras de arte](#)
- [Administração](#)

Figura 4: Formulário para inserir uma nova peça - Exemplo de um formulário criado por nós

Existe ainda uma página chama de "Administração" onde é possível ver todos os registos que existe de uma determinada tabela e ter acesso às operações de Ver, Editar e Eliminar.



Museu da Emigração e das Comunidades

Ínicio | Conteúdo | Exposições | | Acerca | Contactos | | Procurar
Logout (museu) | Perfil

[Home](#) » [Pecas](#) » [Administração](#)

Manage Object Work Records

You may optionally enter a comparison operator (<, <=, >, >=, <> or =) at the beginning of each of your search values to specify how the comparison should be done.

[Advanced Search](#)

Displaying 1-10 of 1014 result(s).

ID	Criador	Medidas/Dimensões	Materiais/Técnicas	Data de Criação	Tipo de Registo	
1	Gérald Bloncourt	42 x 59,5 cm	Fotografia P/B		item	
2	Gérald Bloncourt	42 x 59,5 cm	Fotografia P/B		item	
3	Gérald Bloncourt	42 x 59,5 cm	Fotografia P/B		item	
4	Gérald Bloncourt	42 x 59,5 cm	Fotografia P/B		item	
5	Gérald Bloncourt	59,5 x 42cm	Fotografia P/B		item	
6	Gérald Bloncourt	42 x 59,5 cm	Fotografia P/B		item	
7	Gérald Bloncourt	42 x 59,5 cm	Fotografia P/B		item	
8	Gérald Bloncourt	42 x 59,5 cm	Fotografia P/B		item	
9	Gérald Bloncourt	42 x 59,5 cm	Fotografia P/B		item	
10	Gérald Bloncourt	42 x 59,5 cm	Fotografia P/B		item	

Go to page: < Previous **1** 2 3 4 5 6 7 8 9 10 Next >

Operações
[Índice](#)
[Criar](#)
[Criar Ficha Completa](#)

Figura 5: Página de administração das peças

7.3.2 Front office

O front office e o back office partilham a mesma plataforma, no entanto as operações de back office só estão disponíveis caso o utilizador esteja logado.

Praticamente todas as páginas do site apresentam a seguinte estrutura:



Museu da Emigração e das Comunidades

Ínicio | Conteúdo | Exposições | | Acerca | Contactos | | Procurar
Logout (museu) | Perfil

[Home](#) » [Pecas](#)

Peças do Museu

Sort by: [ID](#) [Criador](#) [Medidas/Dimensões](#) [Materiais/Técnicas](#) [Tipo de Registo](#) [Títulos](#)

Displaying 1-10 of 1014 result(s).

Go to page: < Previous **1** 2 3 4 5 6 7 8 9 10 Next >

ID: 1
Títulos: 3417/22 - Porto 1966
Criador: Gérald Bloncourt
Medidas/Dimensões: 42 x 59,5 cm
Materiais/Técnicas: Fotografia P/B
Data de Criação:
Tipo de Registo: item

ID: 2
Títulos: 3416/20a - Chaves 1966
Criador: Gérald Bloncourt
Medidas/Dimensões: 42 x 59,5 cm
Materiais/Técnicas: Fotografia P/B
Data de Criação:
Tipo de Registo: item

ID: 3
Títulos: 3406/12 - Lisboa 1966
Criador: Gérald Bloncourt
Medidas/Dimensões: 42 x 59,5 cm
Materiais/Técnicas: Fotografia P/B
Data de Criação:

Operações
[Criar](#)
[Criar Ficha Completa](#)
[Importar obras de arte](#)
[Administração](#)

Figura 6: Esquema das páginas

A nossa página está dividida em 4 áreas:

1. - Cabeçalho
2. - Menu
3. - Conteúdo da página
4. - Operações disponíveis, só aparece caso o utilizador esteja logado.

Os registos principais das peças que existem são as obras de arte e um utilizador exigente não se limita a ficar satisfeito em ver apenas a descrição de uma obra de arte, vai querer ver outras peças relacionadas, fazer comentários e até partilhar através das redes sociais, por isso uma página para mostrar uma peça de arte está estruturada apresentada a seguir.


Começa por apresentar os detalhes da obra de arte.

[Início](#)
[Conteúdo](#)
[Exposições](#)
[Acerca](#)
[Contactos](#)
[Procurar](#)

Logout (museu)
Perfil

[Home](#) » [Peças](#) » 3416/26a - Chaves 1966

3416/26a - Chaves 1966 (2)

ID	2
Criador	Gérald Bloncourt
Medidas/Dimensões	42 x 59,5 cm
Materiais/Técnicas	Fotografia P/B
Data de Criação	
Tipo de Registo	item
Tipo de Peça	Fotografia
Classificações	photographs
Titulos	3416/26a - Chaves 1966
Legenda	Nas montanhas - região de Chaves – as casas de madeira e está frio. Esta mulher decorou as paredes com jornais e organizou as panelas e as tampas com decoração! Os utensílios estão presos, ela não os usa... Para mim, trata-se de um testemunho de resistência, de coragem! Um exemplo magnífico de "cultura"!... 1966
Descrições	
Locais	creationLocation: França currentRepository: Museu da Emigração e das Comunidades (Fafe, Portugal) : Chaves
Peças/Obras/Trabalhos Relacionados	
Recursos	
Tags	fotografia, frança, gerald bloncourt, fotografia preto e branco, emigracao

Operações

[Índice](#)
[Exportar](#)
[Criar](#)
[Criar Ficha Completa](#)
[Importar obras de arte](#)
[Actualizar](#)
[Eliminar](#)
[Administração](#)
[Atribuir Título Novo](#)
[Atribuir Título Existente](#)
[Atribuir Legenda](#)
[Atribuir Descrição](#)
[Atribuir Local](#)
[Associar imagem](#)

Figura 7: Detalhes de uma obra

Apresenta as peças que estão de alguma forma relacionadas com a que está a ser vista.



Figura 8: Área onde mostra obras relacionadas

Existe ainda uma área onde se pode partilhar a obra de arte ou fazer comentários.

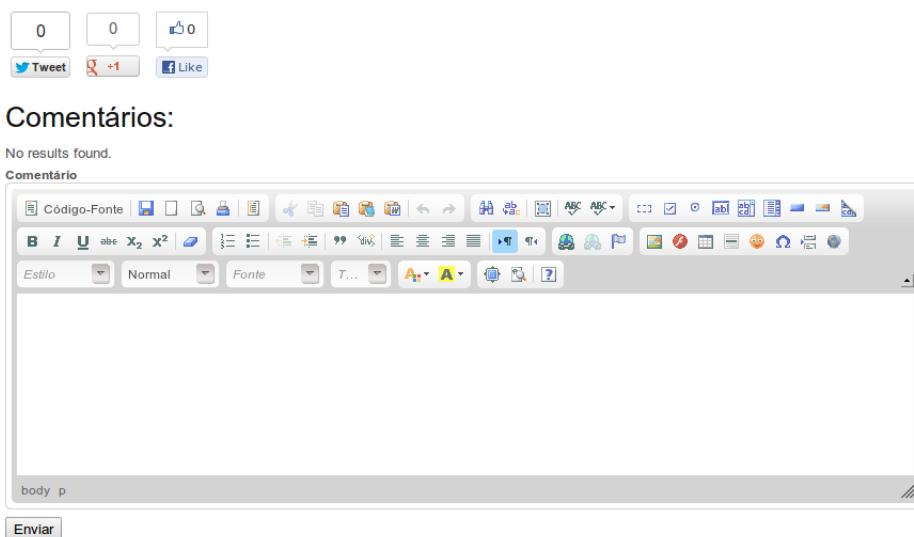


Figura 9: Área social - partilhar e fazer comentários

Os comentários são importantes porque podem servir para um utilizador comum ajudar a enriquecer o repositório.

7.3.3 Salas de Exposição do Museu

Um dos objetivos propostos para este projeto passava pela geração automática de espaços de aprendizagem na forma de salas de exposição do museu. É então possível a criação de exposições que são um conjunto de salas e cuja navegação sobre essas salas segue uma ordem que pode ser definida pela ordem de criação da sala ou pelo utilizador. Essa mesma ordem é exibida aquando da listagem das salas de uma exposição. Cada exposição possui um nome, uma descrição do tema da exposição, uma imagem e um conjunto de tags que descrevem o que é exibido na exposição.

Numa sala de exposição é possível ver os objetos em exposição devidamente relacionados entre si de forma a transmitir o conhecimento correspondente ao tema dessa exposição. Este tema, ou conhecimento a transmitir,

é descrito por um conjunto de funções, em que cada uma delas, extrai os objetos do arquivo de acordo com determinados argumentos definidos para essas funções e cria os links apropriados. Também é possível definir o layout pretendido para uma sala de exposição. Assim como a exposição, uma sala possui um nome, uma descrição do tema da sala e uma imagem associada ao tema.

Geração de Salas de Exposição

Para a geração de salas definimos uma DSL⁵ em XML que permite a especificação dos metadados de uma sala, tais como a exposição a que pertence, o nome, a descrição e os objetos que serão exibidos na sala, sendo cada objeto constituído por um tipo de objeto (indica o tipo dos objetos a apresentar em que cada tipo corresponde a uma função que aplica determinados filtros às obras) e uma lista de argumentos (que complementam os filtros utilizados na função do tipo selecionado). O input desta DSL e os restantes metadados, a imagem, o número de ordenação da sala e o layout pretendido para a exibição da sala são capturados através de um formulário.

Neste formulário, antes de os metadados serem submetidos, uma validação do input da DSL segundo o schema XSD criado (o schema XSD pode ser consultado em anexo) é realizada, assim como uma validação da não existência da exposição mencionada e da não existência de uma sala com o nome especificado na exposição especificada.

Após validação, é iniciado um processo de transformação do input da DSL para aquela que será a sala em formato PHP especificada na DSL, assim como o armazenamento dos seus metadados na base de dados.

Este processo consiste na criação de um ficheiro PHP que será a sala criada. Nela constarão os metadados especificados, sendo que estes serão simplesmente impressos na sala PHP, no entanto os objetos que serão exibidos na sala não seguem esse procedimento. Para que o respetivo código de extração das obras pretendidas para um objeto seja possível, um documento XML foi criado, que contém um mapeamento entre os tipos de objetos existentes na DSL e um extrato de código (este ficheiro pode ser consultado em componentes/conceitos.xml). Esse código é o responsável pela extração das obras do arquivo e apresentação dos mesmos no site. O código possui embutido em si mesmo, palavras reservadas (envolvidas por dois caracteres '%') correspondentes a determinados argumentos necessários à devida extração dos objetos. Um exemplo de um mapeamento existente no documento XML é apresentado de seguida:

```
<conceitos>
  <conceito>
    <key>PecasCriador</key>
    <value>
      &lt; ;br&gt;&lt; ;br&gt; ;
      &lt; ;h5&gt;Peças de %Criador%&lt; ;/h5&gt; ;
      &lt; ;?php
      $dataProvider = IndexingCreators::
                          getObjectWorkRecords_NameCreator( '%Criador%' );
      $dataProvider->pagination->pageSize=%NrItens%;

      $this->widget( 'zii.widgets.CListView', array(
        'dataProvider'=>$dataProvider,
        'template'=>'{sorter}{summary}{pager}{items}{pager}{summary}',
        'pager'=>array(
          'maxButtonCount'=>'10',
        ),
        'itemView'=>'/object_Work_Records/_view',
      ));
    &gt;
  </value>
</conceito>
...
</conceitos>
```

O armazenamento dos metadados na base de dados tem em conta o número de ordenação da sala definido

⁵Uma linguagem de programação ou linguagem de especificação dedicada um problema de domínio específico.

ou não pelo utilizador. Isto é, se o utilizador não o definir então a nova sala será a consequente à última sala já existente nessa exposição, ou seja, a ordem é a da data de criação das salas.

Já se o utilizador definir um número de ordenação este é atribuído à sala. Caso o número esteja compreendido no âmbito de salas existentes, isto é, se o utilizador especificar que a sala será a sala número 100 quando só existem 5 salas nessa exposição, então esta sala será a última dessa exposição. Caso o número esteja compreendido no âmbito de salas existentes, esse número é atribuído à sala e as restantes salas cujo número de ordenação seja igual ou superior a esse número sofrem uma alteração, passando 1 unidade acima, ou seja, existe uma reordenação das salas.

Finalmente, à sala gerada é anexado um sistema de navegação entre salas, sistema este que consiste num link para a sala imediatamente anterior a essa sala e outro para aquela imediatamente posterior.

Informação que se pode colocar nas salas

O conteúdo de uma sala é inserido através de código XML, na parte dos objectos há várias possibilidades de informação que podemos querer:

- **PecasCriador** - Mostra as peças de um determinado criador.
- **PecasLocal** - Mostra as peças de um determinado local.
- **PecasData** - Mostra as peças de uma determinada data.
- **PecasTag** - Peças que tenham associadas uma tag.
- **PecasCriadorTag** - Peças de um criador que tenham uma determinada tag.

Neste momento são estes os tipos de objectos que se pode colocar numa sala.

Layout das Salas de Exposição

Dependendo da informação que se pretenda colocar numa sala, pode ser útil variar o layout para uma apresentação mais agradável ao utilizador, por isso foram criados por enquanto 3 layouts diferentes:

Título	
Descrição	
Parte 1	Parte 2

Figura 10: Template 1

O template 1 baseia-se em ter duas colunas com informação.

Título	
Descrição	
Parte 1	Parte 2
Parte 3	

Figura 11: Template 2

O template 2, é idêntico ao anterior mas aqui é possível ainda adicionar mais informação depois das duas colunas.

Título
Descrição
Parte 1

Figura 12: Template 3

Por último, este template é o mais simples e baseia-se em ter a informação toda seguida.

8 Trabalho futuro

Muitas funcionalidades apesar de estarem implementadas há melhoramentos que se podem fazer, tanto a nível de back office como na geração das salas de exposição.

Ao nível do back office pode-se fazer melhoramentos na parte de enriquecimento da Base de Dados (criação de mais operações para associar informação) e ainda na ingestão de novas peça, tanto pelo formulário como pelo upload de um ficheiro XML (no formato CDWA).

A geração de salas também pode ser melhorada fazendo um formulário em vez de se inserir código XML. A parte das visitas guiadas talvez outro mecanismo de controlo seria mais apropriado.

Algo que não ficou muito bem defenido é a ontologia que controla a navegação no Website. Seria importante fazer uma abordagem diferente da que nós utilizamos.

9 Conclusões

Síntese do Documento.

Estado final do projecto. Análise crítica dos resultados.

Sugestões para trabalho futuro.

A Schema XSD da DSL para especificar uma sala

