



Visualização e Navegação de Conteúdos Audiovisuais

ANDRÉ MANUEL CARNEIRO PINHO

novembro de 2018

VISUALIZAÇÃO E NAVEGAÇÃO EM CONTEÚDOS AUDIOVISUAIS

André Manuel Carneiro Pinho



Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

2018

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha da Unidade Curricular de Tese / Dissertação (TEDI), do 2º ano, do Mestrado em Engenharia Eletrotécnica e de Computadores

Candidato: André Manuel Carneiro Pinho, N.º 1110367, 1110367@isep.ipp.pt

Orientação científica: Paula Viana, pmv@isep.ipp.pt



Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

21 de Novembro de 2018

Resumo

Cada vez mais são disponibilizados conteúdos audiovisuais na *internet* e, conseqüentemente, o acesso a este tipo de informação tem aumentado continuamente. A disponibilização de ferramentas de visualização adequadas é essencial para que os utilizadores consigam tirar o devido proveito deste tipo de conteúdo. Neste contexto, estudos, que têm como objetivo analisar novas formas de visualização de informação que permitam facilitar o processo de cognição e comunicação entre o utilizador e um conjunto de dados, têm sido desenvolvidos.

Nesta dissertação analisaram-se três dos principais tipos de técnicas de visualização que permitem a análise e consumo de conteúdo audiovisuais – as redes, as hierarquias e as séries temporais. As redes constituem uma importante ferramenta de análise e extração de informação uma vez que permitem constatar as relações entre os diferentes elementos da visualização. Por sua vez, as hierarquias permitem melhorar a experiência de navegação uma vez que é possível encontrar a informação pretendida mais rapidamente através da categorização dos dados. As séries temporais surgiram da necessidade de representar informação em função do tempo, o que permite perceber a ordem cronológica de eventos ou acontecimentos.

De forma a testar os conceitos desenvolvidos, e após um levantamento das entidades detentores de conteúdos, em que se destacam o arquivo da Rádio e Televisão de Portugal e a plataforma cultural Europeia que reúne mais de 50 milhões de objetos culturais sobre temas como arte e a Primeira Guerra Mundial, optou-se por efetuar a prova de conceito com este último acervo. Uma das razões principais para esta opção está relacionada com o facto de a plataforma Europeia disponibilizar APIs abertas de acesso à informação.

O protótipo desenvolvido nesta dissertação permite o acesso rápido e eficiente a recursos multimédia, sendo apoiado por um algoritmo de associação

de conteúdos, com base num conjunto de metadados, de forma a garantir que o utilizador consegue identificar os objetos culturais do seu interesse e as suas relações.

Palavras-chave

Conteúdo multimédia, Europeana, Interfaces gráficas, Visualização de informação

Abstract

Audiovisual content is available on the internet and access to this type of information has been continuously increasing. The availability of visualization tools is fundamental so that the users can get the proper use of this type of content. Studies have been made aiming at analyzing new forms of data visualization that facilitate the process of cognition and communication between the user and a dataset.

This dissertation analyzes three of the main types of visualization techniques that allow the analysis and consumption of audiovisual contents - the networks, the hierarchies and the time series. The networks are an important tool for analyzing and extracting information since they allow to verify the associations between the different elements of the visualization. In turn, hierarchies improve the browsing experience since it is possible to find information faster by categorizing the data. The time series arose from the need to represent information over time, which allows perception of the chronological order of events.

A survey of the entities holding audiovisual content has been made to test the concepts developed in which stand out the Portuguese Radio and Television archive and the Europeana cultural platform that brings together more than 50 million cultural objects on topics such as World War I and art. The latter was used to carry out the proof of concept due to the availability of open APIs to access the information.

The prototype developed in this dissertation allows fast and efficient access to multimedia resources and is supported by a content association algorithm, based on a set of metadata, in order to ensure that the user can identify the cultural objects of his interest and its associations.

Keywords

Multimedia content, Europeana, Graphic interfaces, Information visualization

Conteúdo

1	Introdução	1
1.1	Enquadramento e Motivação	1
1.2	Objetivos	2
1.3	Estrutura do Relatório	2
2	Visualização de Informação	5
2.1	Técnicas de Visualização	6
2.1.1	Hierarquia	6
2.1.2	Temporal	9
2.1.3	Redes	11
2.2	Visualização Multimédia	13
2.2.1	Navegação de imagens e vídeos	14
2.2.2	<i>Video Exploration</i>	15
2.2.3	<i>VideoMap</i>	17
2.2.4	<i>Hue Sphere Image Browser</i> e <i>Honeycomb Image Browser</i>	18
3	Ferramentas de Visualização	21
3.1	D3.js	21
3.2	Vega	24
3.3	ECharts	25
3.4	Vis.js	28
4	Repositórios Multimédia	31
4.1	Emissoras	31

4.1.1	<i>British Broadcasting Corporation</i>	31
4.1.2	<i>Deutsche Welle</i>	32
4.1.3	Rádio e Televisão de Portugal	33
4.2	<i>Open Data</i>	34
4.2.1	Europeana	34
4.2.2	<i>Wikimedia Commons</i>	35
5	<i>Europeana Data Model</i>	37
5.1	Descrição	37
5.2	Estrutura	40
5.2.1	Registos	40
5.2.2	<i>Datasets</i>	40
5.2.3	Identificadores	40
5.3	Classes de Dados	41
5.3.1	<i>ore:Aggregation</i>	41
5.3.2	<i>edm:EuropeanaAggregation</i>	41
5.3.3	<i>edm:ProvidedCHO</i>	41
5.3.4	<i>ore:Proxy</i>	42
5.3.5	<i>edm:WebResource</i>	42
5.4	API	42
5.4.1	Utilização	43
5.4.2	API de Pesquisa	44
5.4.3	API de Registos	45
5.4.4	API SPARQL	45
5.4.5	API <i>Open Archives Initiative Protocol for Metadata Harvesting</i>	46
5.4.6	API de Entidades	46
5.4.7	API de Anotações	47
6	Interface de Visualização e Navegação Semântica	49

6.1	Arquitetura do Sistema	49
6.2	Funcionamento	50
6.2.1	Interações	58
6.2.2	Sumarização de conteúdo	58
6.3	Resultados	61
6.3.1	Formulário de pesquisa	61
6.3.2	Visualização principal	62
6.3.3	Visualização complementar	66
7	Conclusão	69
7.1	Trabalho futuro	70
	Referências Documentais	79
	Anexo A. Classes do Modelo de Dados Europeana	81

Esta página foi intencionalmente deixada em branco.

Lista de Figuras

Figura 2.1	PhenoTree [1]	7
Figura 2.2	<i>Treemap</i> [2]	9
Figura 2.3	<i>Timeline</i> [3]	10
Figura 2.4	Diagrama em arco [4]	11
Figura 2.5	Diagrama nó-ligação [5]	12
Figura 2.6	Diagrama de cordas [6]	13
Figura 2.7	Disposição de imagens segundo a sua relevância [7]	14
Figura 2.8	<i>Stream Explorer</i> [8]	16
Figura 2.9	<i>Programme Explorer</i> [8]	16
Figura 2.10	Mapa [9]	17
Figura 2.11	<i>Hue Sphere Image Browser</i> [10]	18
Figura 2.12	<i>Honeycomb Image Browser</i> [10]	19
Figura 3.1	D3.js [11]	23
Figura 3.2	Vega [12]	25
Figura 3.3	ECharts [13]	26
Figura 3.4	Vis.js [14]	29
Figura 4.1	Arquivo de conteúdos BBC	32
Figura 4.2	Pesquisa no arquivo DW	33
Figura 4.3	Pesquisa no arquivo RTP	33
Figura 4.4	Pesquisa no arquivo Europeia	35
Figura 4.5	Pesquisa no arquivo Wikimedia	36

Figura 5.1	Exemplo de uma representação RDF	38
Figura 5.2	Formulário de registo da API Europeia	44
Figura 6.1	Arquitetura do protótipo de visualização	50
Figura 6.2	Interface de visualização e navegação semântica	51
Figura 6.3	Funcionamento geral do protótipo	52
Figura 6.4	Fluxograma do nível zero da visualização	54
Figura 6.5	Fluxograma da construção das associações na visuali- zação	55
Figura 6.6	Visualização de um nível e complexidade grau 2	57
Figura 6.7	Visualização de um nível e complexidade grau 3	57
Figura 6.8	Visualização de dois níveis e complexidade grau 2 . . .	57
Figura 6.9	Fluxograma da criação da sumarização de cores	59
Figura 6.10	Fluxograma da extração da cor dominante de um <i>frame</i>	60
Figura 6.11	Formulário de pesquisa	61
Figura 6.12	Associação por provedor	62
Figura 6.13	Associação por país	63
Figura 6.14	Visualização de profundidade 1 e complexidade 5 . . .	64
Figura 6.15	Visualização de profundidade 1 e complexidade 5 . . .	64
Figura 6.16	Sumarização das cores de cada <i>frame</i>	67

Lista de Tabelas

Tabela 5.1	<i>Namespaces</i> do EDM [15]	39
------------	---	----

Esta página foi intencionalmente deixada em branco.

Lista de Acrónimos

API Application Programming Interface

BBC British Broadcasting Corporation

CHO Cultural Heritage Object

CSS Cascading Style Sheets

DCMI Dublin Core Metadata Initiative

DOM Document Object Model

DW Deutsche Welle

EDM Europeana Data Model

HSL Hue, Saturation, Lightness

HTML HyperText Markup Language

IRI Internationalized Resource Identifier

ISEP Instituto Superior de Engenharia do Porto

JSON JavaScript Object Notation

JSON-LD JavaScript Object Notation for Linked Data

MEEC Mestrado em Engenharia Eletrotécnica e de Computadores

OAIS Open Archival Information System

OAI-PMH Open Archives Initiative Protocol for Metadata Harvesting

PNG Portable Network Graphics

RDF Resource Description Framework

REST Representational State Transfer

RTP Rádio e Televisão de Portugal

SPCA Sparse Principal Component Analysis

SVG Scalable Vector Graphics

TEDI Tese/Dissertação

W3C World Wide Web Consortium

XML Extensible Markup Language

Capítulo 1

Introdução

O desenvolvimento de áreas de investigação como as técnicas de visualização e tecnologias multimédia permitiu novas formas de apresentar informação que podem ser aplicadas em diversos contextos. A forma como os conteúdos multimédia são mostrados tem impacto na transmissão da informação pretendida e por isso é importante estudá-los.

1.1 Enquadramento e Motivação

O conteúdo multimédia é atualmente responsável por grande parte do tráfego na *internet* e esta tendência não tem mostrado sinais de abrandamento [16]. O seu consumo está dependente dos resultados que são retornados pelos motores de busca utilizados diariamente, no entanto, a forma como são mostrados é muitas das vezes insatisfatória [5]. Desta forma, têm surgido cada vez mais formas para visualizar este tipo de conteúdos, que oferecem diferentes funcionalidades e que permitem uma navegação mais acessível e interativa [17]. A opção por este tema teve em conta a necessidade de organizar, filtrar e tornar mais agradável a navegação dos variados conteúdos multimédia com que somos confrontados aquando da pesquisa deste tipo de informação. O acesso a conteúdos multimédia de uma forma mais prática e

intuitiva é relevante tanto para uma utilização pessoal como profissional.

1.2 **Objetivos**

O principal objetivo desta dissertação consiste em analisar as diversas abordagens à visualização e navegação de conteúdos, com especial foco nas técnicas que se podem adequar a conteúdos audiovisuais. Pretende-se também desenvolver um protótipo que disponibilize uma interface gráfica que deverá permitir a procura e visualização deste tipo de informação, utilizando um repositório multimédia como fonte de dados.

1.3 **Estrutura do Relatório**

Este documento encontra-se estruturado nas seguintes partes:

- Capítulo 1, denominado “Introdução”, é apresentada uma breve contextualização ao trabalho proposto, assim como os seus objetivos e a sua estrutura;
- Capítulo 2, denominado “Visualização de Informação”, é apresentado o estado da arte da visualização de informação: os conceitos e técnicas associados, as suas aplicações e casos de estudo relacionados;
- Capítulo 3, denominado “Ferramentas de Visualização”, abordam-se algumas ferramentas de visualização utilizadas para desenvolver interfaces de navegação e análise de conjuntos de dados;
- Capítulo 4, denominado “Repositórios Multimédia”, é feita uma análise dos repositórios multimédia existentes, desde portais agregadores a estações televisivas;
- Capítulo 5, denominado “*Europeana Data Model*”, é descrito o modelo de dados multimédia utilizado para a disponibilização de conteúdos no

portal agregador Europeana;

- Capítulo 6, denominado “Interface de Visualização e Navegação Semântica”, é descrito o trabalho realizado na concepção do protótipo construído nesta dissertação e as suas funcionalidades. São também apresentados os resultados obtidos e é feita uma análise às várias interações do utilizador e a forma como elas afetam a visualização;
- Capítulo 7, denominado “Conclusão”, são reunidas as considerações finais do trabalho realizado e apresentadas sugestões para trabalho futuro.

Esta página foi intencionalmente deixada em branco.

Capítulo 2

Visualização de Informação

Atualmente, é possível obter elevadas quantidades de informação a partir de diferentes fontes, como por exemplo, um artigo numa folha de jornal ou um *outdoor* publicitário à margem da estrada. Contudo, a forma como o conteúdo é apresentado deve ser tida em consideração de modo a que seja possível transmitir a informação da melhor forma possível. Utilizando o exemplo anterior, é possível perceber que não seria muito agradável nem útil utilizar um *outdoor* numa auto-estrada para dar a conhecer as notícias do dia. No entanto, este tipo de aplicação é bastante eficaz em campanhas publicitárias.

O ser humano consegue absorver mais informação através da visão do que através de todos os outros sentidos em conjunto [18], pelo que a eficaz utilização de uma imagem ou de um gráfico, pode traduzir-se num mecanismo rápido e preciso no auxílio à tomada de decisão. Assim, o objetivo da visualização de informação consiste em transmitir dados de uma forma mais acessível do que seria possível utilizando apenas frases e números [19]. A sua utilização permite avaliar, com maior facilidade, grandes quantidades de dados bem como compreender as suas características e quaisquer anomalias.

2.1 Técnicas de Visualização

A representação efetiva de informação é um desafio em que a solução passa, na maioria das vezes, por utilizar atributos para codificar o conteúdo de relevo dessa informação [20]. Estes atributos podem ser figuras geométricas, cores e transparências.

Existem diversas técnicas de apresentar informação, sendo as mais comuns os gráficos de duas e três dimensões como o gráfico de área ou o gráfico de barras. No entanto, existem outros métodos mais complexos que agregam diferentes princípios de visualização [21] e que serão abordados nesta secção.

2.1.1 Hierarquia

A utilização de uma lista vertical nem sempre é a forma mais conveniente de apresentar os resultados de uma pesquisa, especialmente quando são retornados diferentes aspetos da mesma informação. Usando como exemplo a pesquisa do termo “Porto”, podem ser apresentados resultados relativos à cidade do norte de Portugal, ao vinho de igual nome ou ao clube de futebol (Futebol Clube do Porto).

A disposição de informação segundo uma hierarquia, permite melhorar a experiência de navegação dos resultados no sentido em que é possível encontrar a informação pretendida mais rapidamente [22].

Radial Tree

A técnica de visualização *Radial Tree* consiste numa representação de uma estrutura idêntica à de uma árvore, sendo constituída por diversas hierarquias [23]. As hierarquias são formadas por nós, geralmente de igual tamanho e as ligações entre os vários níveis são feitas do interior da visualização para a sua periferia.

Baytas et al [1] aplicou este conceito no desenvolvimento de uma ferramenta de visualização chamada PhenoTree que permite a participação de

Treemap

A visualização *Treemap* consiste na representação hierárquica de grupos de informação, geralmente sob a forma de retângulos [25].

Ao contrário da *Radial Tree* onde é originada uma figura semelhante à de uma árvore, com origem num ponto central e propagação para a periferia. Numa visualização *Treemap*, os vários níveis da hierarquia são sucessivamente agrupadas até não ser mais possível categorizar a informação.

A utilização de diferentes cores para as diferentes categorias e o seu tamanho permite identificar quais são os elementos de maior dimensão e consequentemente, os que têm um maior relevo na estrutura de dados em análise [26]. Uma das mais comuns utilizações desta visualização consiste na representação dos diretórios de um sistema operativo.

Esta técnica encontra-se representada na figura 2.2 onde a sua aplicação permite identificar as hierarquias do *software* Flare, uma biblioteca para criar animações compatíveis com *Adobe Flash Player*.

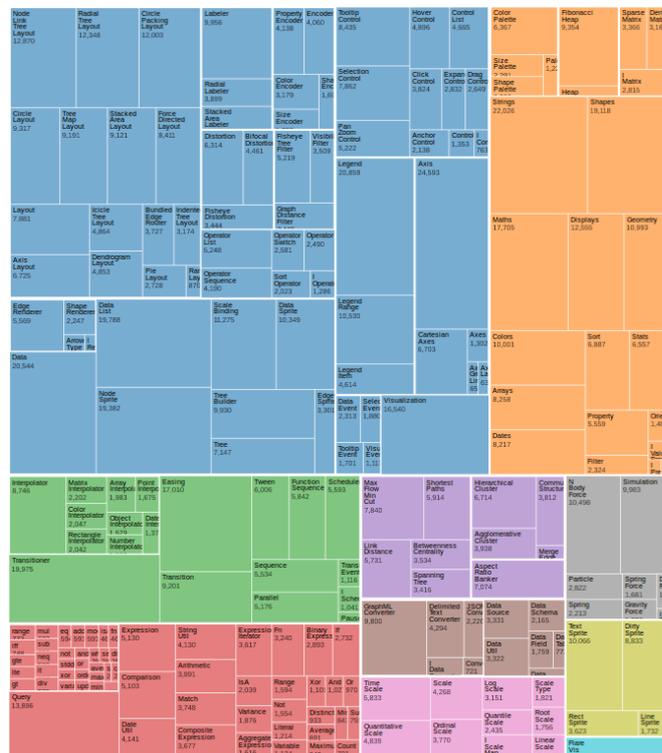


Figura 2.2: Treemap [2]

2.1.2 Temporal

Por vezes é necessário representar informação em função do tempo, de forma a perceber a ordem cronológica de eventos ou acontecimentos [27]. Este tipo de visualização pode ser aplicado nas mais diversas áreas de estudo, sendo bastante útil nas finanças e na meteorologia.

Timeline

A *timeline* é um método de representação que proporciona uma melhor compreensão da informação que está associada a um período de tempo, sendo que é utilizada nas mais diversas áreas como na gestão de projetos, visualização de registos médicos e na cobertura de eventos [28]. O seu princípio de funcionamento consiste em organizar a informação num eixo horizontal de acordo com a sua ocorrência no tempo.

A sua utilização em conjuntos de dados multimédia significa uma mais-valia porque este tipo de representação permite que os seus utilizadores consigam compreender que tipos de eventos ocorreram ao longo do tempo, em que momento, a sua duração e ainda que ocorrências podem ter sido afetados por outros eventos [29]. O recurso a imagens, sons ou vídeos permite que a experiência de utilização seja mais agradável e interativa uma vez que os utilizadores têm a possibilidade de descobrir mais informação sobre um determinado acontecimento, podendo traduzir-se assim numa maior utilidade [30].

Na figura 2.3 encontra-se representado um exemplo de uma *timeline* sobre a contribuição das mulheres na história do computador [3]. O gráfico contém diferentes objetos multimédia, como por exemplo, imagens, mapas e sons.

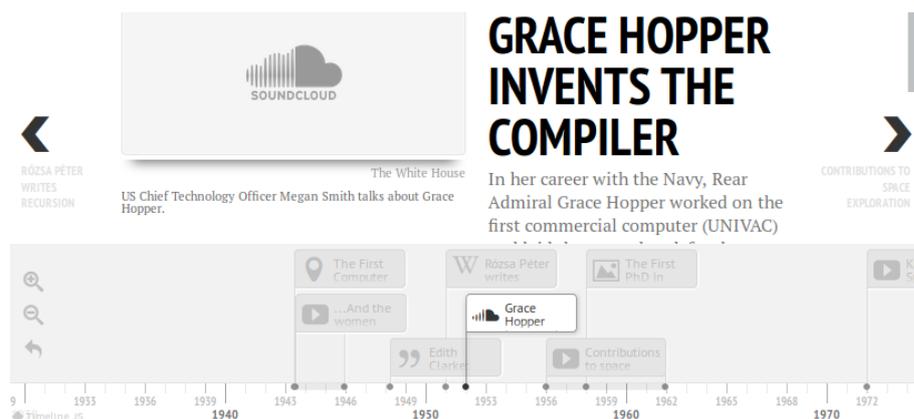


Figura 2.3: *Timeline* [3]

Diagrama em Arco

O diagrama em arco consiste na representação horizontal de pontos consecutivos, geralmente segundo uma linha reta [31]. Esta visualização é constituída também por ligações, representadas por um ou mais semi-círculos, que interligam os diferentes pontos. Através deste método é possível identificar sequências ou repetições podendo ser utilizada para visualizar estruturas de dados altamente estruturadas [4]. Uma aplicação deste tipo de visualização consiste na identificação de diferentes passagens de uma música. Cada arco

relaciona duas passagens iguais o que significa que têm as mesmas notas. Quanto mais notas estiverem relacionadas, mais acentuada fica a espessura do arco. Na figura seguinte encontra-se aplicada esta técnica de visualização para a composição *Für Elise* de Beethoven onde se verifica que a música começa e acaba exatamente com a mesma passagem.

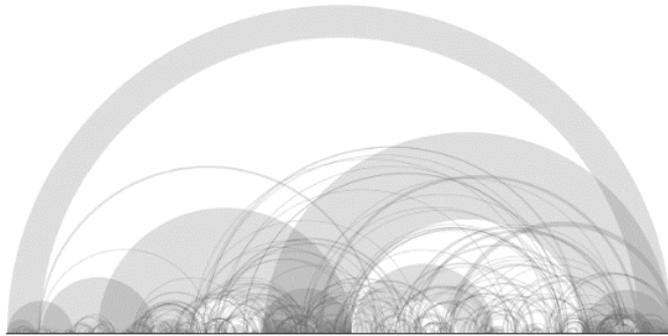


Figura 2.4: Diagrama em arco [4]

2.1.3 Redes

Por vezes é necessário ter uma visão geral sobre a forma como diferentes elementos se relacionam num meio, principalmente quando é necessário entender o comportamento destes. A representação de uma estrutura de dados numa rede constitui uma importante ferramenta de análise e extração de informação sobre as relações entre os diferentes elementos da visualização, que pode ser definida de diferentes formas.

Diagrama Nó-Ligação

O diagrama nó-ligação é uma forma de visualização que permite mostrar associações entre diferentes entidades. Nesta representação, as entidades são normalmente desenhadas como um nó arredondado e estão associadas entre si através de linhas [32]. Os nós podem ter diferentes cores dependendo da sua função na visualização.

Esta técnica foi utilizada por Rashid et al [5] no desenvolvimento de uma

diversos pontos. Assim, esta técnica de visualização aplica *Hierarchical Edge Bundling* para agrupar e moldar ligações adjacentes o que permite realçar as associações entre as partes do *dataset* sem obstruir a área de representação. Quanto mais ligações existirem entre duas partes, mais espesso será o arco que as liga. A figura 2.6 ilustra a aplicação desta técnica na representação de viagens da plataforma Uber nas várias zonas da cidade de San Francisco, nos Estados Unidos da América.

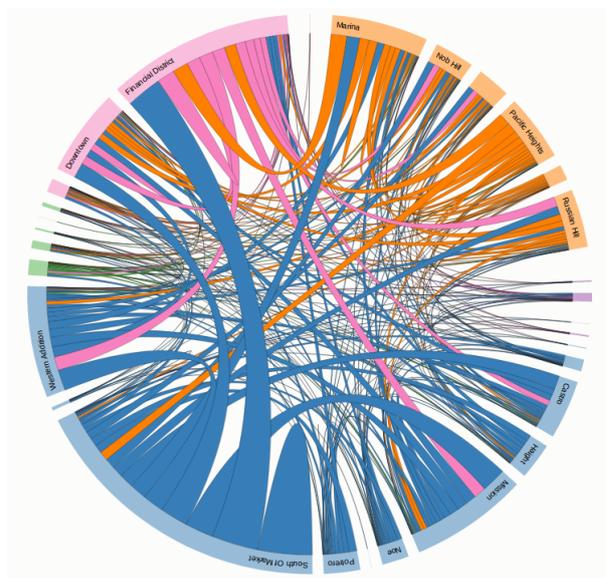


Figura 2.6: Diagrama de cordas [6]

2.2 Visualização Multimédia

O desenvolvimento de áreas de investigação como as tecnologias de apresentação e multimédia levou ao aparecimento de novas formas de visualização que podem ser aplicadas em diversos contextos. Os exemplos a seguir representam alguns dos progressos realizados por essas áreas de estudo, com especial foco em conteúdo audiovisual.

2.2.1 Navegação de imagens e vídeos

O trabalho desenvolvido por Ren et al [7] consiste numa interface gráfica interativa para explorar grandes conjuntos de imagens e vídeos. No que diz respeito a conteúdos capturados de forma amadora ou doméstica, sabe-se que os metadados geralmente disponíveis nestes dois tipos de dados estão limitados à data de captura, o que faz com que os dispositivos de armazenamento guardem fotografias e *clips* de vídeo de uma forma desorganizada. A alternativa passa por gerar anotações de uma forma automática através de ferramentas de análise e *machine learning*.

Neste exemplo foi implementado um *design* interativo de procura e navegação, centrado no utilizador e focado na interação humana com o conteúdo em vez de analisar a forma como os utilizadores criam relações semânticas entre o conteúdo. Para isso, as imagens são dispostas segundo um *ranking* onde as mais relevantes apresentam um maior tamanho. O *ranking* é definido através das escolhas anteriores do utilizador. É-lhe apresentado um conjunto de imagens e assim que uma delas é escolhida, esta é deslocada para o centro e as restantes são organizadas segundo a sua semelhança com a imagem central. Esta relação encontra-se representada na figura 2.7.



Figura 2.7: Disposição de imagens segundo a sua relevância [7]

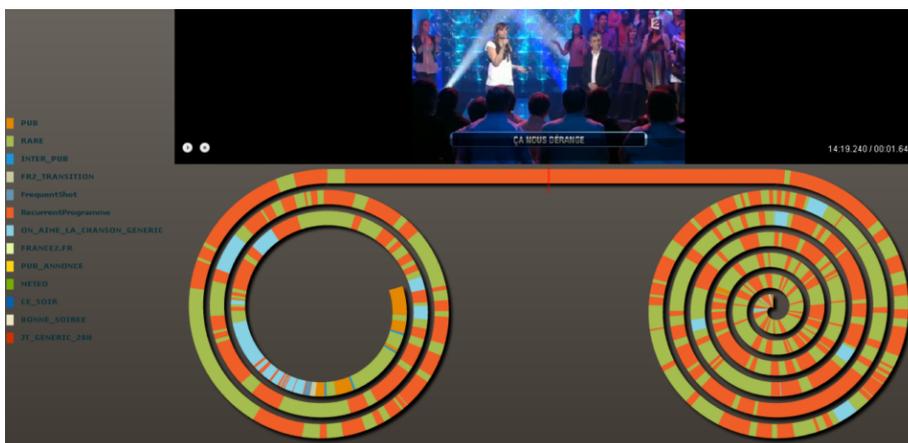
Esta abordagem permite a apresentação eficiente de conteúdo e traduz-se numa forma interativa de aceder e explorar vastos repositórios de dados. Devido à sua natureza, esta interface permite interagir com os objetos multimédia sem dificuldades o que permite chegar a um maior número de utilizadores.

2.2.2 *Video Exploration*

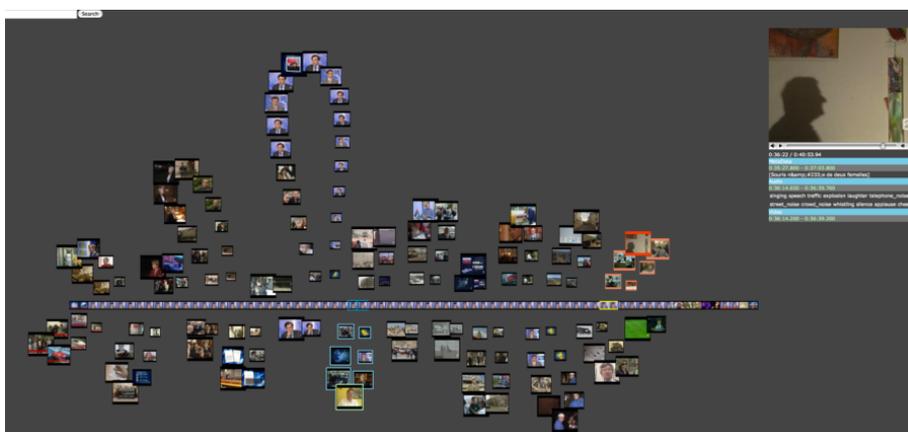
O trabalho desenvolvido por Viaud et al [8], propõe duas ferramentas para interagir com conteúdos de vídeo - *Stream Explorer* e *Programme Explorer*. Os seus objetivos são, respetivamente:

- Identificar e explorar os diferentes conteúdos de um vídeo;
- Disponibilizar e permitir a interação com o conteúdo de programas televisivos.

No caso do *Stream Explorer*, o vídeo é dividido por classes, representadas por diferentes cores segundo uma *timeline* com a forma de uma cassette de vídeo. Utilizando como exemplo um canal de televisão francês, são identificadas pelo menos três classes - publicidade, conteúdo muito frequente e conteúdo pouco frequente. A interface possui um leitor de vídeo que permite navegar os diferentes segmentos de vídeo, representado na figura 2.8.

Figura 2.8: *Stream Explorer* [8]

O *Programme Explorer* foi desenhado de forma a estruturar o programa televisivo segundo um *backbone* (espinha dorsal) que agrega o maior grupo, ou grupos de imagens similares. Estas imagens são depois dispostas horizontalmente enquanto, em cada lado do *backbone*, são desenhados laços com os diferentes tópicos do programa. O resultado final, na figura 2.9, permite realçar a estrutura do vídeo e mostrar uma visão geral do seu conteúdo.

Figura 2.9: *Programme Explorer* [8]

2.2.3 VideoMap

O recurso a mapas para representar a relação entre conteúdos foi utilizada por Ma et al [9] no desenvolvimento do seu trabalho “VideoMap”, que utiliza este meio para apresentar conteúdo de vídeo. Esta adaptação representa um auxílio na visualização deste tipo de *media* uma vez que permite construir várias relações entre os diferentes constituintes do vídeo, como as personagens, os locais e os acontecimentos.

Na figura 2.10 encontra-se representado um exemplo deste trabalho utilizando o filme “The Matrix”. As cenas deste filme são representadas por blocos de diferentes tamanhos e cores que refletem a sua duração. Os eventos são identificados por pontos vermelhos e ligados por uma linha da mesma cor por ordem cronológica (figura 2.10.a).

Aumentando a proximidade da visualização, como representado na figura 2.10.b, faz aparecer as personagens e os seus nomes sendo que a sua associação a um evento é representada com uma linha cinzenta. Na figura 2.10.c são representadas as diferentes *storylines*, cada uma com a sua cor.

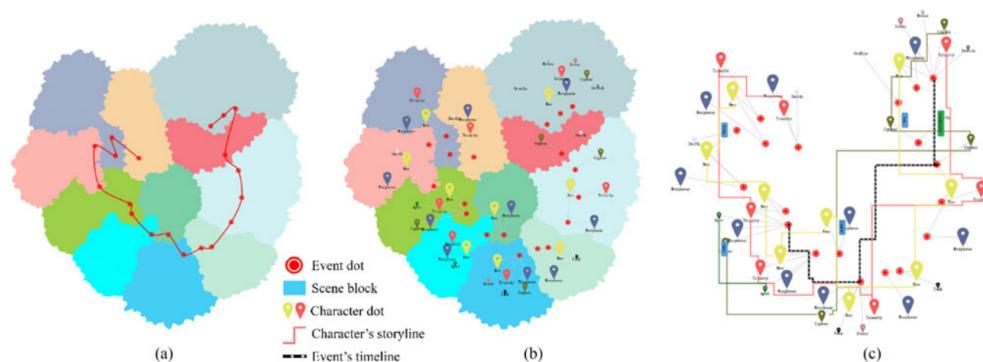


Figura 2.10: Mapa [9]

O exemplo exposto anteriormente poderia ser adaptado para representar as relações entre diferentes vídeos. Para isso, os blocos coloridos dariam lugar às formas dos países produtores desse tipo de conteúdos e os pontos vermelhos representariam os diferentes vídeos ligados de forma similar, ou

seja, cronologicamente.

2.2.4 *Hue Sphere Image Browser e Honeycomb Image Browser*

Estes dois trabalhos realizados por Gerald Schaefer demonstram abordagens efetivas para organizar grandes coleções de imagens [10]. Para isso, são utilizadas as características de cada imagem para dispô-las na visualização sendo que imagens similares são colocadas próximas umas das outras. O recurso a elementos de navegação permite que o utilizador explore as coleções de forma interativa.

Hue Sphere Image Browser é um sistema de navegação de imagens que dispensa a utilização de algoritmos de processamento complexos, como os histogramas. Em vez disso, as imagens são dispostas de acordo com o formato HSL (*hue, saturation, lightness*) - tonalidade, saturação e luminosidade, respetivamente. Tal como o próprio nome indica, os objetos são dispostos numa esfera onde o utilizador é levado a explorar a coleção através de um movimento rotativo (figura 2.11).



Figura 2.11: *Hue Sphere Image Browser* [10]

Honeycomb Image Browser partilha muitas funcionalidades com o sis-

tema anterior, nomeadamente o processo de obtenção da cor média de cada imagem. No entanto, a disposição das imagens é feita segundo um conjunto de hexágonos que fazem lembrar o interior de uma colmeia, tal como representado na figura 2.12.



Figura 2.12: *Honeycomb Image Browser* [10]

Esta página foi intencionalmente deixada em branco.

Capítulo 3

Ferramentas de Visualização

A *visual analytics* é uma corrente de investigação emergente com foco na exploração e análise de dados [34]. Ao contextualizar os dados e resultados de uma análise numa visualização, obtém-se uma experiência poderosa que tanto pode apoiar como confrontar os resultados da análise, ajudando assim à interpretação de dados e validação de resultados.

A transformação de conjuntos de dados a partir do seu formato natural como letras e números, para uma visualização gráfica, é um processo que requer ferramentas adequadas ao tipo de utilização pretendida. Para além das ferramentas profissionais, normalmente acessíveis através da compra de uma licença, existem outras, disponibilizadas sem qualquer custo e que podem ser utilizadas por estudantes, investigadores e programadores. Estas alternativas representam uma maior flexibilidade para criar *designs* customizados e, com tal, são as que se adequam ao contexto desta dissertação.

3.1 D3.js

D3.js (ou simplesmente D3) é uma biblioteca *JavaScript* para criar e manipular visualizações interativas a partir de conjuntos de dados [35]. A sua integração com as normas *web* como *HyperText Markup Language* (HTML),

Scalable Vector Graphics (SVG) e *Cascading Style Sheets* (CSS) permite acessar aos mesmos meios dos *browsers* atuais sem recurso a nenhuma *framework* externa. Desta forma, é possível interligar os dados com a *Document Object Model*¹ (DOM) e em seguida aplicar transformações, como por exemplo, transformar um *array* de números num gráfico de barras.

A funcionamento desta ferramenta assenta em essencialmente três princípios:

- Seleção - Através da sua abordagem declarativa, D3 utiliza conjuntos de elementos ou seleções para iterar e modificar componentes da visualização. Por exemplo, para alterar a cor de todos os elementos do tipo parágrafo para branco, seria necessária apenas a seguinte linha:

```
d3.selectAll("p").style("color", "white");
```

- Propriedades dinâmicas - A possibilidade de mudar determinados elementos tendo em conta as suas propriedades é uma forma bastante útil de personalizar aquilo que é mostrado ao utilizador. Este princípio pode ser utilizado para, por exemplo, definir o tamanho de letra de um elemento a partir de um *array* de valores:

```
d3.selectAll("p")
  .data([4, 8, 15, 16, 23, 42])
  .style("font-size", function(value) { return value + "px"; });
```

- Movimentação - Com este princípio é possível entrar e sair de elementos da visualização para, por exemplo, remover conteúdo quando o conjunto de dados é atualizado. Para isso é utilizada a sintaxe *enter* e *exit* como definido no seguinte exemplo:

¹API para documentos HTML que define a estrutura lógica e a forma como o documento é acedido e manipulado [36]

```
d3.select("body")
  .selectAll("p")
  .data([4, 8, 15, 16, 23, 42])
  .enter().append("p")
  .text(function(value) { return "I'm number " + value + "!"; });
```

A sua arquitetura modular permite a utilização de *plugins*, disponibilizados pelos seus autores ou por terceiros e que permitem adicionar novas formas de criar visualizações.

A figura 3.1 representa um diagrama de cordas que permite visualizar relações de uma forma circular. Este exemplo [11], feito para um *blog*, dispõe em cada lado diferentes categorias de temas e no centro encontram-se as pessoas associadas a cada um dos lados.

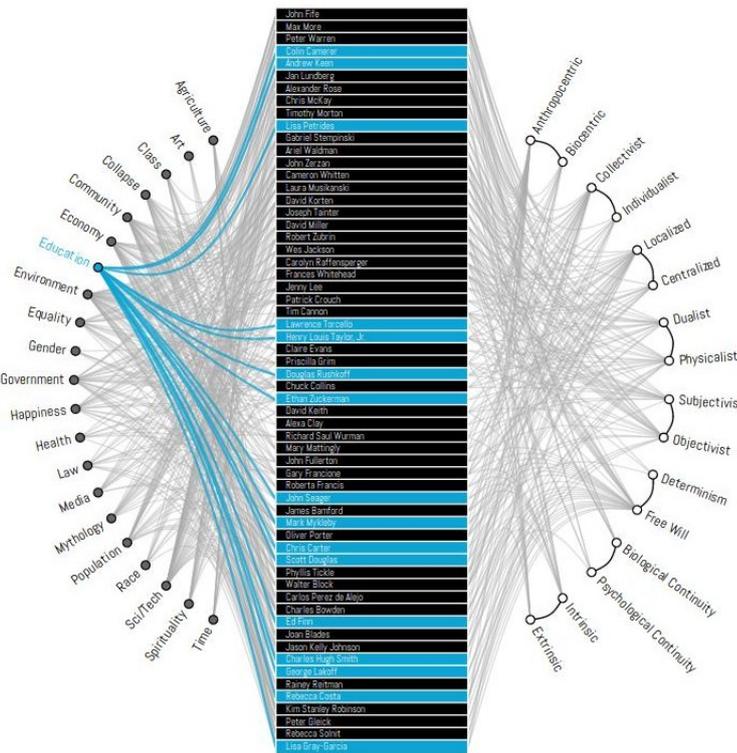


Figura 3.1: D3.js [11]

3.2 Vega

Vega é uma linguagem declarativa para criar visualizações interativas [37]. Através da sua especificação em *JavaScript Object Notation* (JSON) é possível determinar a aparência e o comportamento da visualização. As definições desta especificação incluem diversas propriedades, entre elas:

- `$schema` - Endereço para o conjunto de dados que dará origem à visualização;
- `description` - Descrição do que se pretende obter como por exemplo, "Gráfico de barras sobre a variação do preço do petróleo desde 2000";
- `config` - Valores predefinidos para o tipo de visualização, como por exemplo, configuração dos eixos ou estilo do título;
- `transforms` - Permite processar um *dataset* e definir novos valores como calcular o máximo e mínimo de um conjunto de dados.

A visualização é gerada depois de carregar o ficheiro com a especificação descrita anteriormente para a API Vega, como por exemplo:

```
vega.loader()  
  .load('specification.json')  
  .then(function(data) { render(JSON.parse(data)); });
```

Desta forma, Vega disponibiliza um meio para que os seus utilizadores possam desenvolver métodos para gerar visualizações. O seu objetivo é promover um ecossistema de componentes interoperáveis para desenvolver visualizações como ferramentas de *design* interativo ou de recomendação automática de gráficos.

Na figura 3.2 encontra-se representado um exemplo de aplicação desta ferramenta na construção de uma visualização sobre as ligações entre os maiores aeroportos nos Estados Unidos da América, em 2008 [12]. Na figura,

o ponto de origem que se encontra selecionado é o aeroporto internacional de Los Angeles.

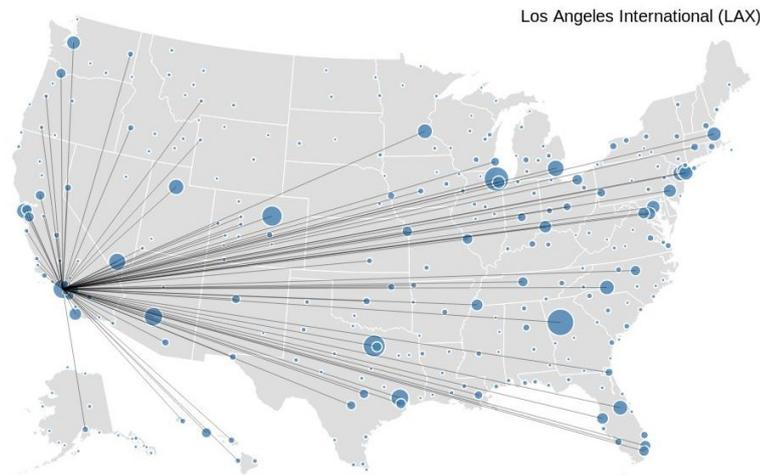


Figura 3.2: Vega [12]

3.3 ECharts

ECharts é uma *framework web* que permite a rápida implementação de visualizações interativas [38]. O seu objetivo é ser de fácil utilização e altamente eficiente de forma a que possa ser aplicada por utilizadores que não tenham competências técnicas de desenvolvimento *web*. Assim, é possível construir elementos como os gráficos de barras ou de área mas também customizar e estender as visualizações de forma a torná-las mais complexas.

À semelhança da ferramenta Vega, mencionada anteriormente, a declaração de elementos, estilos, animações e outras propriedades pode ficar definida num ficheiro JSON que é carregado para a biblioteca. Da lista de propriedades suportadas fazem parte, entre outras, o título da visualização, o tipo de animações e as configurações do componente a desenhar.

Na figura 3.3 encontra-se aplicada esta ferramenta num gráfico de dispersão, adaptado para mostrar as horas e os dias da semana com um formato semelhante ao de um relógio para representar os momentos de atividade ao

longo do tempo.

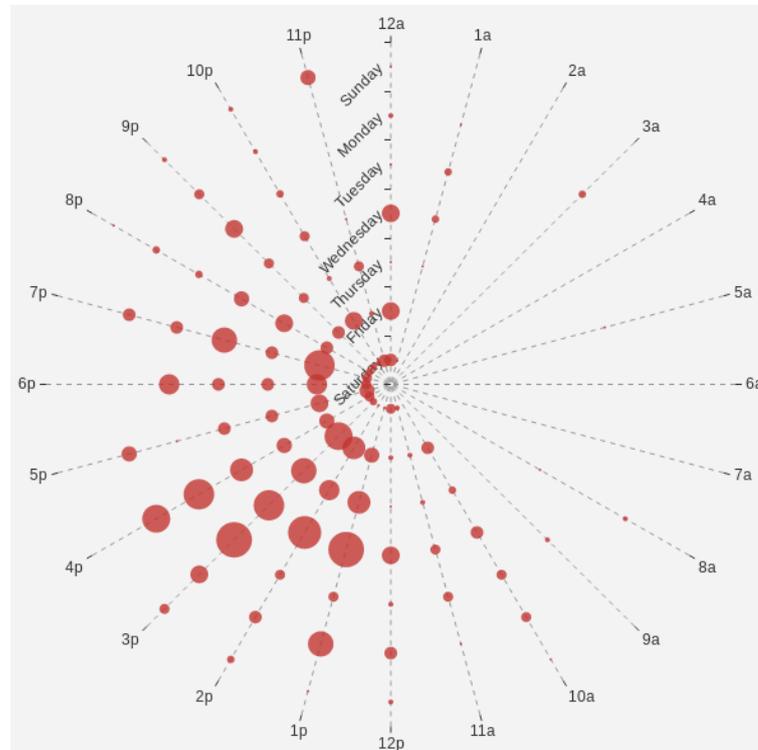


Figura 3.3: ECharts [13]

A inicialização desta biblioteca pode ser feita em apenas duas instruções, no entanto é necessário definir toda a estrutura da visualização previamente, como mostra o seguinte excerto de configurações:

```
var options = {
  title: {
    text: 'Registration moments',
  },
  angleAxis: {
    type: 'category',
    data: hours,
    splitLine: {
```

```
        show: true,
       LineStyle: {
            color: '#999',
            type: 'dashed'
        }
    },
    axisLine: {
        show: false
    }
},
radiusAxis: {
    type: 'category',
    data: days,
    axisLine: {
        show: false
    },
    axisLabel: {
        rotate: 45
    }
},
series: [{
    type: 'scatter',
    coordinateSystem: 'polar',
    data: data
}]
};

var myChart = echarts.init(document.getElementById('main'));
myChart.setOption(options);
```

3.4 Vis.js

Vis.js é uma biblioteca de visualização dinâmica, desenhada para ser fácil de utilizar e ao mesmo tempo capaz de processar e manipular grandes conjuntos de dados [39]. Disponibiliza atualmente vários modos de visualização, entre eles:

- Rede - Este modo permite construir redes de nós e ligações. A definição deste tipo de visualização encontra-se dividida em diversos módulos, que controlam a forma como cada um dos elementos é disposta. Dos módulos incluídos fazem parte métodos para gerir a criação dos nós, a interação do utilizador e o *layout* da visualização;
- *Timeline* - Constitui uma interface para visualizar um conjunto de dados ao longo do tempo de forma interativa, sendo possível navegar os vários períodos numa escala que varia entre os milissegundos e os anos. É possível fazer as diversas operações de eventos (criar, mover e apagar) e ainda aplicar estilos CSS.

A sua inicialização é bastante direta bastando para isso definir o elemento que irá mostrar a visualização, o *dataset* e as suas opções.

```
var container = document.getElementById('mygraph');  
var graph3d = new vis.Graph3d(container, data, options);
```

No caso de uma visualização do tipo rede, é também possível estender a utilização da visualização através de uma API que permite, por exemplo, editar elementos e detetar interações do utilizador.

A figura 3.4 ilustra uma rede de equipamentos utilizando esta biblioteca onde é possível verificar as diferentes ligações entre os vários nós [14]. Os diferentes tipos de nós são customizados com uma imagem representativa do equipamento.

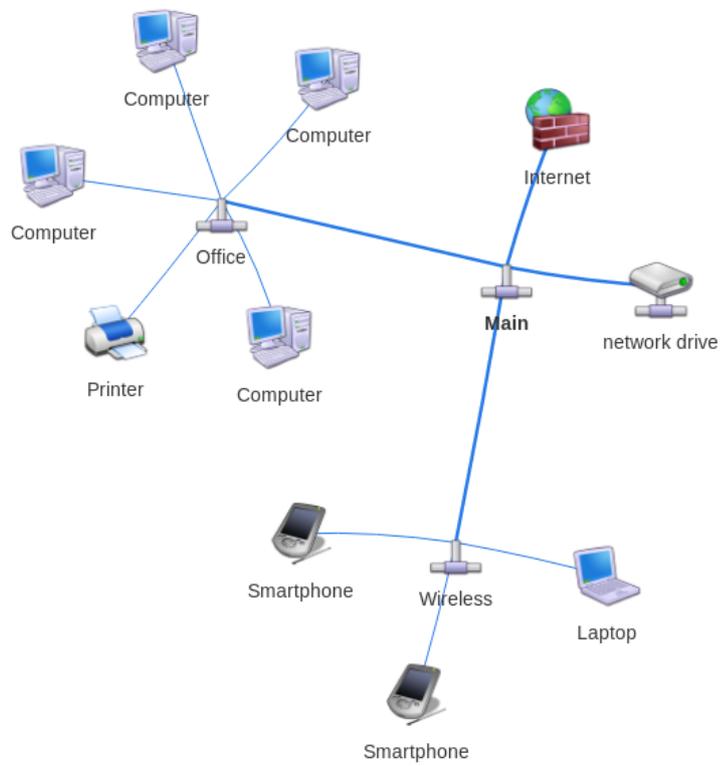


Figura 3.4: Vis.js [14]

Esta página foi intencionalmente deixada em branco.

Capítulo 4

Repositórios Multimédia

Todos os dias, diversos agregadores de conteúdos disponibilizam as suas coleções na *internet* de forma a permitir que qualquer pessoa tenha acesso a este tipo de informação. Este ato resulta da necessidade cada vez maior de consumir informação digital tanto para uso recreativo como para investigação. Nesta secção serão abordadas as instituições que têm feito este esforço e que permitem o livre acesso às suas coleções multimédia.

4.1 Emissoras

As emissoras de rádio e televisão são, possivelmente, o tipo de instituição que gera mais conteúdo multimédia diariamente. Desta forma, é importante identificar aquelas que disponibilizam a consulta das suas coleções.

4.1.1 *British Broadcasting Corporation*

British Broadcasting Corporation (BBC) é uma emissora pública do Reino Unido que distribuí conteúdos de rádio e televisão. Através dos seus arquivos *online*¹, é possível aceder a um conjunto de coleções que permite a consulta de imagens, áudios e vídeos que fazem parte desta instituição. Esta página

¹<http://www.bbc.co.uk/archive/collections.shtml>

não permite a procura de temas por palavras-chave sendo que as coleções existentes estão organizadas por categorias e dispostas verticalmente em duas colunas, como ilustra a figura 4.1.

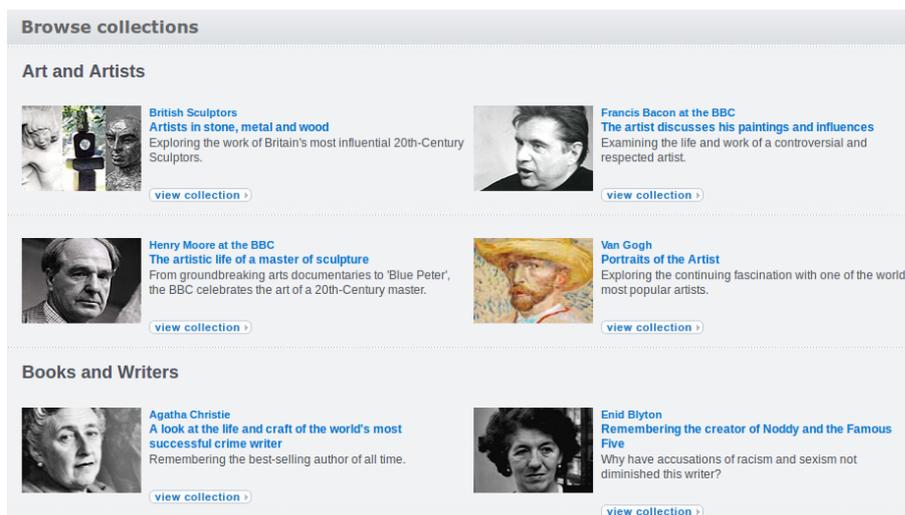


Figura 4.1: Arquivo de conteúdos BBC

4.1.2 Deutsche Welle

A Deutsche Welle (DW) é a emissora internacional da Alemanha para conteúdo televisivo e rádio. Através do seu *media center*², que conta com mais de 53 mil registos, é possível explorar conteúdo multimédia como imagens, áudio e vídeo.

Por definição, os resultados são apresentados segundo uma grelha paginada e ordenados pela data de publicação. Nos resultados são incluídos o título, data e descrição. É possível aplicar filtros como o tipo de *media* e a data de publicação.

A figura 4.2 ilustra o formato de apresentação de resultados no arquivo da DW depois de fazer uma procura por "Europe".

²<http://www.dw.com/en/media-center/s-100824>

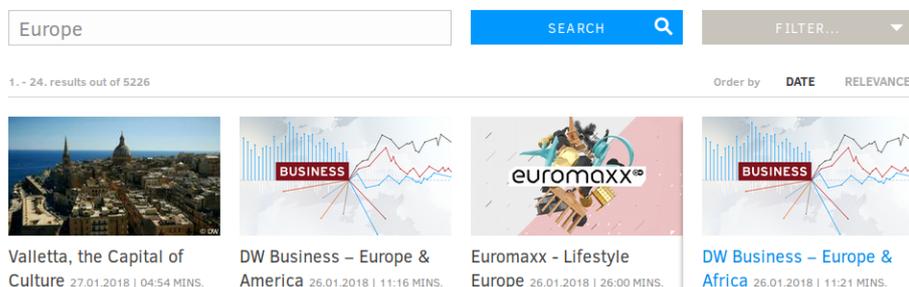


Figura 4.2: Pesquisa no arquivo DW

4.1.3 Rádio e Televisão de Portugal

A Rádio e Televisão de Portugal (RTP) disponibilizou, a 7 de Março de 2017, um arquivo³ com mais de 6500 conteúdos [40]. Este portal permite consultar vídeos, áudios, fotografias e textos, produzidos desde 1936 até ao momento, de uma forma fácil e gratuita.

Os resultados de uma pesquisa no arquivo da RTP são retornados por ordem de relevância segundo uma grelha vertical paginada. Cada um dos resultados inclui o seu título, data e uma pequena descrição. É também possível aplicar filtros de forma a poder especificar o formato do conteúdo ou o género.

A figura 4.3 ilustra o formato de apresentação de resultados no arquivo da RTP depois de fazer uma procura por "Fórmula 1".



Figura 4.3: Pesquisa no arquivo RTP

³<https://arquivos.rtp.pt/>

4.2 *Open Data*

A expressão *Open Data* ou dados abertos ganhou relevo em 2009 quando diversos governos mundiais anunciaram iniciativas para disponibilizar informação estatal de uma forma livre [41]. Este termo significa que os dados podem ser consultados, modificados e partilhados para qualquer fim e sem custos. Este movimento levou ao aparecimento de algumas plataformas agregadoras de conteúdos e que serão abordadas em seguida.

4.2.1 **Europeana**

Europeana é a plataforma digital da União Europeia para a herança cultural [42]. Através deste arquivo é possível aceder a mais de 53 milhões de itens que incluem imagem, texto, vídeo, áudio e objetos tridimensionais. As coleções de dados aqui presentes são normalmente cedidas por entidades parceiras como museus, universidades e bibliotecas. Todo o seu conteúdo pode ser consultado livremente, contando também com uma *Application Programming Interface* (API) para obtenção de conteúdo de uma forma computadorizada.

Os resultados duma pesquisa no portal Europeana são mostrados, por definição, segundo uma lista vertical paginada podendo ser também dispostos através de uma grelha. Estes são constituídos pelo título, data, nome da instituição que disponibilizou o conteúdo e o seu tipo. A interface de pesquisa permite também aplicar filtros para refinar os resultados tais como a instituição provedora e o tipo de conteúdo.

A figura 4.4 ilustra o formato de apresentação de resultados no arquivo da Europeana depois de fazer uma procura por "Leonardo Da Vinci".



Figura 4.4: Pesquisa no arquivo Europeana

4.2.2 *Wikimedia Commons*

Wikimedia Commons é uma organização sem fins lucrativos que disponibiliza livremente um repositório de conteúdo multimídia, lançado em Setembro de 2004 [43]. Contém mais de 50 milhões de arquivos e permite que os seus utilizadores possam consultar este tipo de informação em qualquer idioma. Existe também a possibilidade de contribuir ou consultar a informação desta organização através de uma API.

Os resultados de uma pesquisa de vídeos neste repositório são retornados segundo uma grelha e existe também a possibilidade de pesquisar dentro de uma subcategoria, como se verifica na figura 4.5

Subcategories

This category has the following 8 subcategories, out of 8 total.

- ▶ [Videos of 2018 from Portugal](#) (12 F)
- ▶ [Videos of rail transport in Portugal](#) (2 C, 38 F)
- V**
 - ▶ [Videos from Braga](#) (3 F)
 - ▶ [Videos from Coimbra](#) (2 F)

2

Media in category "Videos from Portugal"

The following 26 files are in this category, out of 26 total.

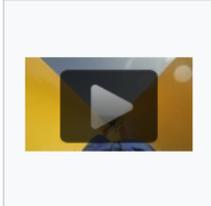
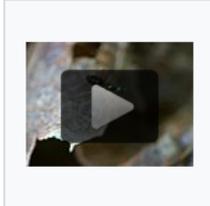
			
3D All-on-4 technique.ogv	Aquashow.webm	Ariasella pieltaini.ogv	Buildings in Sintra P1000251.ogv
24 s, 1,920 × 1,080; 1.61 MB	9.1 s, 1,920 × 1,080; 10.88 MB	2 min 27 s, 640 × 480; 39.65 MB	6.5 s, 1,280 × 720; 5.79 MB

Figura 4.5: Pesquisa no arquivo Wikimedia

Capítulo 5

Europeana Data Model

No âmbito desta dissertação pretende-se estudar a visualização de informação multimédia e por isso é necessário ter acesso a este tipo de conteúdos. Assim, tendo em conta os repositórios multimédia apresentados na secção 4, optou-se por utilizar a Europeana uma vez que disponibiliza o acesso aos conteúdos através de APIs. A sua completa documentação voltada para o desenvolvimento e investigação, foi um fator que pesou na altura da escolha.

5.1 Descrição

Esta organização recebe objetos culturais de diferentes origens como museus, bibliotecas e arquivos audiovisuais, sendo que estes podem utilizar diferentes *standards* para organizar os metadados dos seus objetos. Uma vez que esta informação tem que ser apresentada de forma a que tenha valor, foi desenvolvido o *Europeana Data Model* (EDM), que pretende facilitar a associação de objetos culturais de diferentes origens para que estes possam ter mais relevo quando acedidos através de um contexto intercultural e multilingue como é o caso da Europeana [15].

Até ao momento, o *World Wide Web Consortium* (W3C) introduziu um conjunto de normas que permitem representar e partilhar informação estru-

turada de forma a que possa ser acedida por equipamentos ligados à *internet* [44]. Uma dessas normas chama-se *Resource Description Framework* (RDF) e é utilizada para representar objetos utilizando três componentes - o objeto, o predicado e o sujeito - de forma a estabelecer relações com significado semântico. Por exemplo, o trio (“Os Maias”, autor, “Eça de Queirós”) descreve o livro ligando o seu identificador ao de Eça de Queirós através do conector “autor” que denota assim a relação entre a obra e o seu escritor.

Na figura seguinte é mostrada uma representação de como RDF expressa a informação dos objetos [45]. Aqui:

- O identificador dentro do círculo e escrito em fonte normal representa um recurso RDF;
- A *string* entre aspas representa um nome ou valor que pode incluir uma *tag* para identificar o idioma;
- A seta que liga dois recursos indica uma declaração RDF no formato de trio. O objeto está na origem da seta, o predicado é a propriedade escrita junto à linha e por fim o sujeito encontra-se onde a seta termina;
- O identificador em itálico mostra o tipo de recurso, se aparecer dentro de um círculo ou uma super propriedade se aparecer junto à seta.

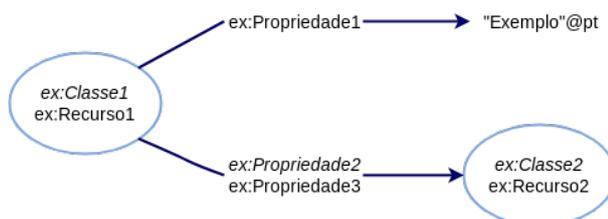


Figura 5.1: Exemplo de uma representação RDF

Esta intenção de disponibilizar diferentes tipos de conteúdos na *internet* levou ao aparecimento do princípio de *Linked Data* que se refere a um conjunto de práticas para interligar estruturas de dados de diferentes origens

na *web* [46]. Para isso, utiliza RDF para representar estas ligações. Esta intenção de criar um meio para normalizar a partilha de informação levou ao aparecimento da *Semantic Web* que influenciou a criação do EDM, nomeadamente, na adoção da RDF para representar os seus objetos culturais [44].

O modelo de dados Europeana pretende ser um meio para receber e interligar informação de diferentes provedores de conteúdos que podem incluir pinturas, textos e objetos tridimensionais [15]. Desta forma é possível enriquecer cada vez mais os objetos culturais devido às associações que se vão criando a partir da disponibilização de diferente informação a partir de várias fontes. Existe uma série de propriedades que o EDM utiliza para concretizar o seu objetivo e que podem ser introduzidas pelo próprio modelo de dados ou por *namespaces*¹. Os *namespaces* são utilizados na RDF de forma a abreviar os identificadores de classes e propriedades.

Na tabela seguinte encontra-se representados todos os *namespaces* que são utilizados no EDM.

Tabela 5.1: *Namespaces* do EDM [15]

Prefixo	Descrição
cc	Creative Commons
dc	Dublin Core
dcterms	Dublin Core Metadata Initiative (DCMI) Terms
edm	Europeana Data Model
foaf	Friend of a Friend Vocabulary
ore	Open Archives Initiative Object Reuse and Exchange
owl	OWL Web Ontology Language
rdaGr2	RDA Group 2 elements
rdf	Resource Description Framework
skos	Simple Knowledge Organization System
wgs84	WGS84 Geo Positioning

¹Conjunto de elementos dentro de uma classe em que cada elemento possui um identificador que pode ser partilhado com elementos de outras classes [47].

5.2 Estrutura

Para que este modelo de dados possa servir o seu propósito, foi necessário definir e organizar as diferentes partes que o constituem e que serão apresentadas de seguida.

5.2.1 Registos

O registo contém a informação que relaciona os metadados, a representação digital e outros tipos de recursos como agentes, lugares e épocas pertencentes ao *Cultural Heritage Object* (CHO) [15].

5.2.2 Datasets

Os registos são organizados por *datasets* que são definidos, segundo *Open Archival Information System* (OAIS), como um *Information Package* ou uma coleção de informação [15]. Esta unidade pode representar uma coleção pertencente à Europeia mas noutros cenários um *dataset* pode ser originado por um *provider*, que disponibiliza o conjunto diretamente à Europeia ou por um *data provider* que disponibiliza o *dataset* a um agregador.

5.2.3 Identificadores

No EDM são utilizados *Internationalized Resource Identifiers* (IRI) como identificadores para todos os tipos de recursos que fazem parte dos metadados de um CHO e para o próprio CHO [15]. Estes identificadores permitem que os recursos sejam diferenciáveis, ou seja, acedendo ao endereço http://data.europeana.eu/item/92056/BibliographicResource_1000093324568 leva a que seja mostrada tanto a página HTML do objeto no portal da Europeia como a respetiva resposta no caso de um pedido por API.

5.3 Classes de Dados

O modelo de dados descrito nesta secção permite que as diferentes entidades que disponibilizam conteúdos à Europeana possam estruturar os dados da forma que lhes seja conveniente e que respeite a informação do objeto [45]. Por sua vez, a Europeana irá manipular estes conteúdos de forma a agregá-los e enriquecê-los. Para isso conta com várias classes de recursos que resultam do conjunto de informação que lhe é disponibilizada [15].

5.3.1 *ore:Aggregation*

Este objeto de agregação contém o conjunto de recursos relacionados com um CHO e que, coletivamente, representam esse objeto na Europeana [15]. Dos recursos fazem parte as descrições do objeto recolhidas junto dos *providers* e a descrição feita pela Europeana. A relação entre o objeto e a agregação é feita pela propriedade *edm:aggregatedCHO*. A lista de propriedades desta classe encontra-se disponível no anexo A.1.

5.3.2 *edm:EuropeanaAggregation*

É uma sub-classe do objeto *ore:Aggregation* e representa as agregações feitas pela própria Europeana que contêm, tal como no objeto anterior, descrições sobre o CHO [45]. As propriedades desta classe estão descritas no anexo A.2.

5.3.3 *edm:ProvidedCHO*

Representa o *Cultural Heritage Object* sobre o qual a Europeana recolhe descrições [15]. Os metadados enviados pelos provedores não se encontram disponíveis neste objeto, no entanto, a única propriedade aqui definida (*owl:sameAs*) serve para identificar quando diferentes identificadores têm a mesma identidade.

5.3.4 *ore:Proxy*

A Europeana utiliza *proxies* como substitutos de objetos culturais dentro das agregações com a finalidade de fazer distinções sobre os CHOs [15]. Desta forma, é possível distinguir os metadados adicionados pela Europeana e os metadados gerados nos *providers*. O anexo A.3 lista as propriedades que podem ser encontradas nesta classe.

5.3.5 *edm:WebResource*

É a representação digital do CHO [15]. É possível que existam várias representações para o mesmo objeto. Quando assim é, as representações estão associadas ao *ore:Aggregation* pela propriedade *edm:hasView* ou alguma das suas sub-propriedades. As propriedades deste objeto, listadas no anexo A.4, descrevem as representações do CHO.

5.4 API

A Europeana disponibiliza o acesso às suas coleções através de uma API *Representational State Transfer* (REST) desde 2011. Desde então tem sido continuamente melhorada e cresceu de tal forma que deu origem a várias APIs mais especializadas. Estas interfaces permitem não só aceder às obras disponibilizadas na Europeana mas também contribuir com informação sobre os registos.

O termo REST foi inicialmente definido por Roy Fielding em 2000 na sua dissertação de doutoramento [48] e especifica uma arquitetura que pode ser aplicada no desenho de sistemas distribuídos e no desenvolvimento de serviços *web*. Esta arquitetura define uma série de requisitos:

- Deve haver uma clara separação entre cliente e servidor de forma a que problemas de interface gráfica não afetem o armazenamento e tratamento de dados;

- A comunicação entre cliente e servidor deve ser *stateless*, ou seja, cada pedido do cliente ao servidor deve conter toda a informação necessária para o seu processamento;
- De forma a retirar o máximo aproveitamento da rede, o sistema deve apresentar um sistema de *cache*. Assim, o cliente pode reutilizar as respostas anteriores do servidor a pedidos idênticos.

Através desta API REST, a Europeia disponibiliza o acesso de programadores e investigadores ao seu repositório para que estes possam enriquecer as suas aplicações e aumentar a partilha de conhecimento.

5.4.1 Utilização

A utilização da API Europeia requer que seja efetuado um registo num formulário² criado para o efeito e representado na figura 5.2. Neste formulário é necessário fornecer um endereço de correio eletrónico válido para onde serão enviadas as duas chaves de acesso à API. A chave pública é utilizada para a maior parte das funcionalidades enquanto que a chave privada será necessária para aceder a funcionalidades mais específicas que requerem uma autenticação adicional. Estas chaves são também utilizadas para recolher, anonimamente, informação acerca da sua utilização para que a organização possa obter estatísticas relevantes sobre a utilização da API e assim continuar a melhorá-la.

²<https://pro.europeana.eu/get-api>

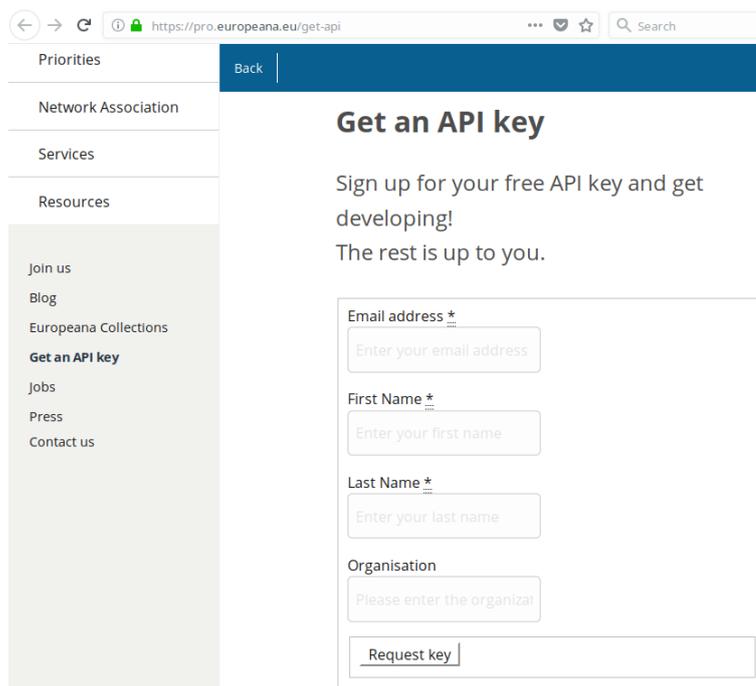
The image shows a web browser window with the URL <https://pro.europeana.eu/get-api>. The page has a dark blue header with a 'Back' button. On the left, there is a navigation menu with items: Priorities, Network Association, Services, Resources, Join us, Blog, Europeana Collections, Get an API key (highlighted), Jobs, Press, and Contact us. The main content area is titled 'Get an API key' and contains the text: 'Sign up for your free API key and get developing! The rest is up to you.' Below this is a registration form with the following fields: 'Email address *' (with placeholder 'Enter your email address'), 'First Name *' (with placeholder 'Enter your first name'), 'Last Name *' (with placeholder 'Enter your last name'), and 'Organisation' (with placeholder 'Please enter the organizat'). At the bottom of the form is a 'Request key' button.

Figura 5.2: Formulário de registo da API Europeana

Atualmente, a API REST da Europeana encontra-se dividida em seis interfaces que serão especificadas nas sub-seções seguintes.

5.4.2 API de Pesquisa

A API de Pesquisa disponibiliza uma forma de procurar metadados no repositório Europeana através de palavras-chave e é considerada a interface mais fácil de utilizar [49]. Através dos seus filtros, como o idioma ou o tipo de dados, é possível refinar a pesquisa de conteúdos. Esta interface tem como objetivo retornar todos os resultados que contenham as palavras-chave pesquisadas.

Cada chamada a esta API é um pedido HTTPS para o endereço <https://www.europeana.eu/api/v2/search.json>. Ao URL anterior deve ser adicionada também a chave pública (*wskey*), obtida através do registo prévio e o termo a pesquisar (*query*). Com estes dois parâmetros obtêm-se a forma mais básica de utilização desta API - <https://www.europeana.eu/api/v2/>

`search.json?wskey=CHAVE&query=TERMO`.

A resposta é sempre enviada em JSON e contém, para além dos resultados encontrados, uma série de campos com informação sobre o pedido que foi feito, como por exemplo, se este foi bem sucedido e quanto tempo demorou a ser atendido.

5.4.3 API de Registos

A API de registos permite aceder diretamente aos dados guardados pela Europeia, que se encontram modelados por EDM [50]. Esta interface permite assim obter um segmento do modelo de dados que normalmente contém um CHO, uma agregação de informação que relaciona os metadados com a representação digital e ainda determinados recursos contextuais como períodos e locais associados ao objeto cultural. Os pedidos a esta API devem ser direcionados para o endereço `https://www.europeana.eu/api/v2/record/ID_REGISTO.FORMATO` onde o ID do registo resulta da junção do identificador do *dataset* com o identificador do registo dentro do *dataset*. O formato diz respeito à forma como os resultados irão ser apresentados que podem ser em JSON, RDF ou *JavaScript Object Notation for Linked Data* (JSON-LD). Em caso de erro numa chamada a esta API será retornado o respetivo erro HTTP e a sua descrição.

5.4.4 API SPARQL

SPARQL é uma linguagem de pesquisa do formato de dados RDF [51]. Pode ser utilizada para fazer *queries* em diferentes fontes de dados e suporta agregação, negação e *subqueries*. A API SPARQL permite explorar associações entre dados da Europeia e fontes de dados externas como Geonames³, Wikidata⁴ e DBPedia⁵. O *endpoint* desta API encontra-se disponível no

³<http://www.geonames.org/>

⁴<https://www.wikidata.org>

⁵<https://www.dbpedia.org/>

endereço `http://sparql.europeana.eu` que por sua vez acede ao conteúdo do repositório Europeana através da base de dados Virtuoso - uma solução que permite a disponibilização de dados e integra com RDF [52]. Uma *query* SPARQL é normalmente constituída pelos prefixos dos *namespaces*, a escolha dos campos a pesquisar e as variáveis. A expressão seguinte mostra uma *query* que retorna a lista de provedores da Europeana.

```
PREFIX edm: <http://www.europeana.eu/schemas/edm/>
SELECT ?DataProvider
WHERE { ?Aggregation edm:dataProvider ?DataProvider }
```

5.4.5 API *Open Archives Initiative Protocol for Metadata Harvesting*

A tecnologia OAI-PMH especifica um protocolo baseada na colheita de metadados independentemente da aplicação que os alberga [53]. Através da sua implementação numa API é possível obter grandes quantidades de dados do repositório Europeana. Estes dados podem ser totalmente ou parcialmente obtidos e utilizados posteriormente no desenvolvimento de aplicações. Neste momento, esta API Europeana encontra-se na sua fase Beta pelo que, para ser utilizada, é necessário obter uma chave diferente daquela que é utilizada, por exemplo, na API de pesquisa. Os dados estão agrupados em *datasets* e disponíveis no formato RDF e *Extensible Markup Language* (XML). O endereço para aceder a esta interface encontra-se disponível em `http://oai.europeana.eu/` e as respostas são retornados em XML.

5.4.6 API de Entidades

Uma entidade pode ser um local ou uma pessoa mas também pode representar um conceito, uma obra ou um período de tempo [54]. Estas entidades fazem parte da *Europeana Entity Collection* e como tal encontram-se associadas a vocabulários como Geonames, DBPedia e Wikidata. A API de

entidades é a mais recente das APIs Europeana, tendo a primeira versão pública sido disponibilizada em Outubro de 2017. Esta interface permite obter informação sobre pessoas, tópicos ou locais que fazem parte da coleção de entidades Europeana e de momento suporta duas operações: recolha de conjuntos completos de metadados associados a uma entidade e sugestão de entidades com base nos *inputs* dos utilizadores. As chamadas à API são feitas por HTTPS para o endereço <https://www.europeana.eu/api/entities/> e as respostas formatadas em JSON-LD. Por se tratar de uma versão inicial, pode ser acedida utilizando a chave “apidemo”.

5.4.7 API de Anotações

No contexto da Europeana existem anotações que são melhorias integrais ou parciais de metadados que podem ser originadas pelos utilizadores. Esta API permite criar, obter e gerir anotações do repositório Europeana. Encontra-se de momento na sua fase Alpha, tal como a API de entidades, pelo que a sua utilização requer a utilização de uma chave diferente da que é utilizada em APIs mais estáveis. Os pedidos a esta API são feitos sobre HTTPS para os endereços <https://test-annotations.europeana.eu/annotation> (ambiente de testes) e <https://www.europeana.eu/api/annotations> (ambiente de produção). O primeiro *endpoint*, pode ser acedido através da chave “tester1” e permite adicionar, atualizar e remover anotações enquanto o último possibilita a procura e recolha de anotações utilizando a chave “apidemo”. Tal como na API de entidades as respostas apenas suportam o formato JSON-LD.

Esta página foi intencionalmente deixada em branco.

Capítulo 6

Interface de Visualização e Navegação Semântica

Esta secção tem como finalidade descrever o desenvolvimento do protótipo construído no âmbito desta dissertação. Depois de analisar as técnicas de visualização apresentadas na secção 2.2, decidiu-se que se iria desenvolver uma visualização baseada no diagrama nó-ligação de forma a que fosse possível comprovar e explorar as ligações entre os diferentes conteúdos através das interações do utilizador com o protótipo. Para isso recorreu-se à biblioteca de visualização Vis.js apresentada na secção 3.4 e que permite, através dos seus módulos, construir uma rede de nós e ligações onde é possível customizar os elementos e estilos [14].

6.1 Arquitetura do Sistema

Durante a análise dos requisitos do sistema de visualização a desenvolver, definiu-se que este deveria ser acessível pelo maior número possível de utilizadores. Assim, a utilização de tecnologias *web* como HTML e JavaScript para construir o protótipo tornou-se evidente em detrimento de outras alternativas como as aplicações móveis ou *desktop*. Desta forma, qualquer

dispositivo com um navegador *web* poderá aceder e comprovar o funcionamento do protótipo. Na análise dos requisitos ficou definido também que a pesquisa de objetos culturais seria feita através de palavras-chave. Assim, o processo de visualização de conteúdos é iniciado pelo utilizador quando introduz o termo que pretende pesquisar. Esta informação é então enviada para o servidor que irá inquirir a coleção Europeia através da sua API de pesquisa e retornar os resultados, se existirem, para o cliente. Esta sequência de ações encontra-se representada na figura 6.1.

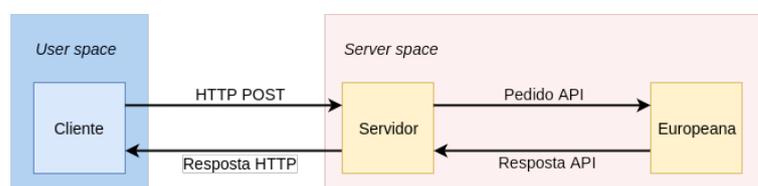


Figura 6.1: Arquitetura do protótipo de visualização

Tal como referido anteriormente, o sistema desenvolvido foi pensado para que pudesse ser utilizado por qualquer dispositivo com capacidades de processamento de páginas *web* como telemóveis, *tablets* ou computadores. Desta forma, evita-se que seja necessário instalar qualquer *software* adicional por parte do utilizador uma vez que um *browser* é suficiente para disponibilizar a visualização.

No que diz respeito ao *server space*, foi necessário instalar e configurar um servidor para servir páginas *web* (Apache) e uma linguagem de programação que fosse capaz de receber os pedidos do utilizador, processá-los e retornar as respetivas respostas (PHP).

6.2 Funcionamento

O protótipo desenvolvido no âmbito desta dissertação, visível na figura 6.2, é constituído essencialmente por duas partes:

- Formulário de pesquisa, localizado no lado esquerdo, onde o utilizador

é levado a especificar as palavras chave sobre o tema que deseja procurar. É também possível especificar o número de objetos visíveis por associação e o número de níveis da visualização;

- Área de visualização, que irá ser utilizada para apresentar os resultados da pesquisa sobre a forma de um diagrama nó-ligação.

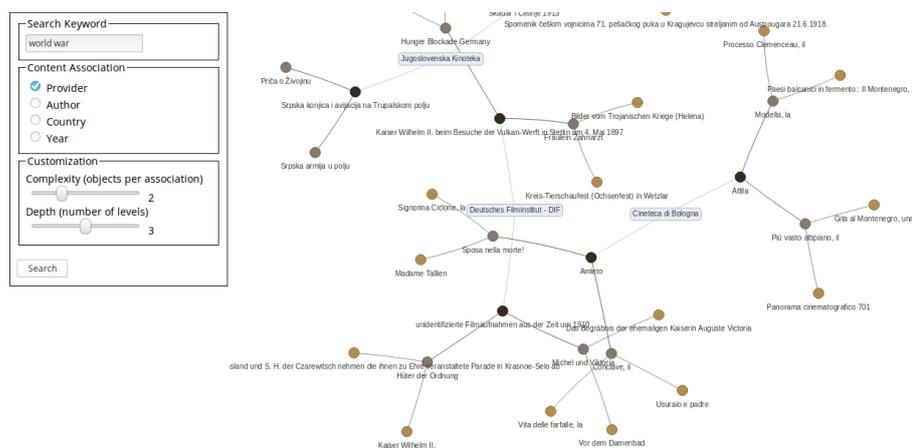


Figura 6.2: Interface de visualização e navegação semântica

Na primeira vez que a página é aberta não é apresentada nenhuma visualização e o único elemento visível é o formulário de pesquisa. Neste formulário o utilizador é levado a preencher o campo relativo às palavras chave que pretende procurar e, se assim entender, pode customizar a resposta através dos parâmetros de “Complexidade” e “Profundidade”. Estas duas propriedades irão controlar o número de objetos que são mostrados ao utilizador.

Assim que o formulário é submetido através do botão criado para esse efeito, os dados são enviados para o servidor que por sua vez irá chamar a API Europeia de pesquisa de forma a encontrar os objetos culturais associados ao termo digitado no formulário.

A resposta da API é então devolvida para a página que irá construir a visualização com os objetos retornados pela Europeia, caso estes existam. Se não forem encontrados resultados, será mostrada ao utilizador uma mensa-

gem de erro a indicar que não foram encontrados resultados para a pesquisa efetuada. Este funcionamento encontra-se representado no fluxograma da figura 6.3.

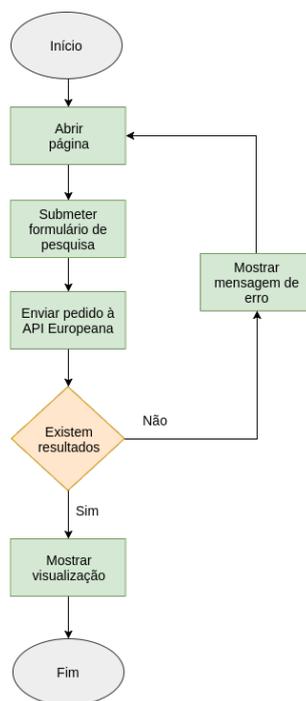


Figura 6.3: Funcionamento geral do protótipo

É no processamento dos resultados que se encontra o componente mais importante deste protótipo uma vez que é o responsável por iterar a resposta da Europeia e construir a estrutura de dados que dará origem à visualização através do Vis.js. Esta estrutura é constituída por duas partes: os nós e as ligações. Os nós são os objetos retornados pela Europeia e as ligações representam as associações entre os vários nós.

Para esta visualização definiu-se que os nós seriam dispostos segundo uma hierarquia, constituída por um nó “pai” e vários nós “filho”. A primeira propriedade identificada no modelo de dados Europeia que seria adequada para o nível superior desta hierarquia foi o *data provider*, ou seja, a organização que disponibilizou o objeto ao agregador. Desta forma, o utilizador

consegue identificar quem disponibilizou determinada obra e descobrir conteúdos dessa mesma organização.

Contudo, verificou-se que outras propriedades poderiam ser aplicadas para o mesmo fim tais como, o ano da obra, o seu autor e o país de origem. Estas quatro propriedades fazem parte do parâmetro do formulário que permite definir como é feita a associação dos objetos culturais. Por definição, é selecionada a opção que diz respeito ao *data provider*.

Outro dos parâmetros de pesquisa nesta interface é a profundidade, que define a quantidade de níveis que a visualização pode ter. Este parâmetro é tido em conta nas seguintes iterações dos objetos de forma a que o número de níveis de nós não ultrapasse o valor definido pelo utilizador. O mesmo acontece para o parâmetro da complexidade, ou seja, irão ser adicionados nós a cada “pai” até que o valor da complexidade seja atingido. Quando isso acontece, os nós começam a ser colocados no nível seguinte, se possível.

Depois de definido o nível superior da hierarquia, através do parâmetro “*Content Association*”, é possível começar a construir a base desta visualização. A resposta da Europeia é então iterada e em cada objeto cultural é identificada a propriedade de associação definida anteriormente. De seguida, o objeto é adicionado à lista de nós com um identificador de nó “pai” ficando desta forma concluído o nível zero da visualização (figura 6.4).

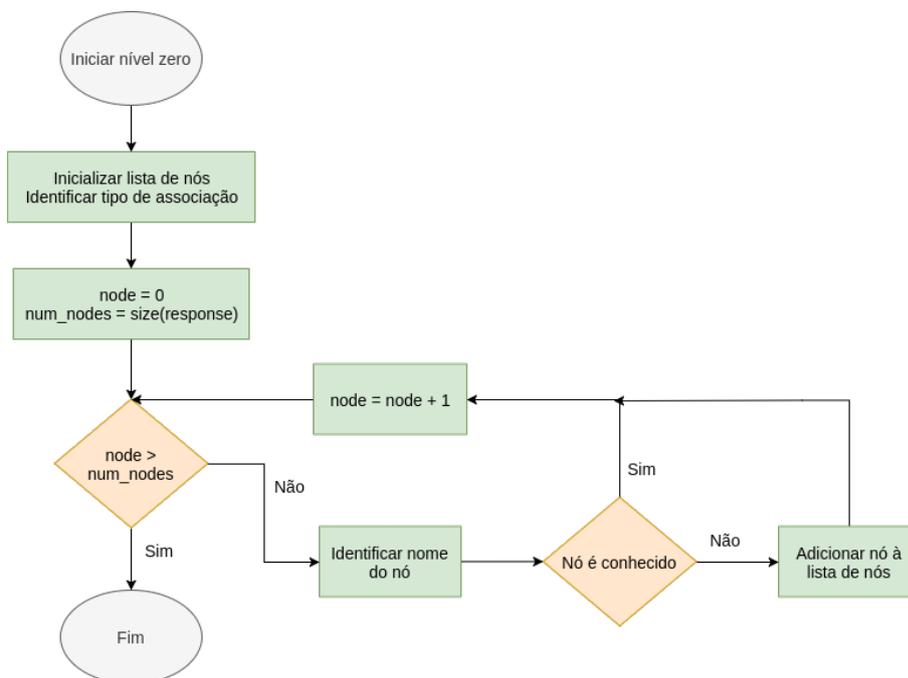


Figura 6.4: Fluxograma do nível zero da visualização

A estrutura de dados da visualização é então inicializada da seguinte forma:

```

data.nodes = [{
    label: 'Provedor A'
}]
  
```

Para definir a restante estrutura de dados, é utilizado um ciclo para preencher os níveis de dados em falta. A resposta Europeia é novamente iterada e para cada objeto cultural é identificado o seu “pai” através do parâmetro de associação. Se este nó superior não tiver atingido a sua capacidade máxima, o objeto é adicionado à lista de nós. Uma vez que neste passo já é definida uma relação entre os nós é necessário atualizar a lista de ligações da estrutura de dados (figura 6.5).

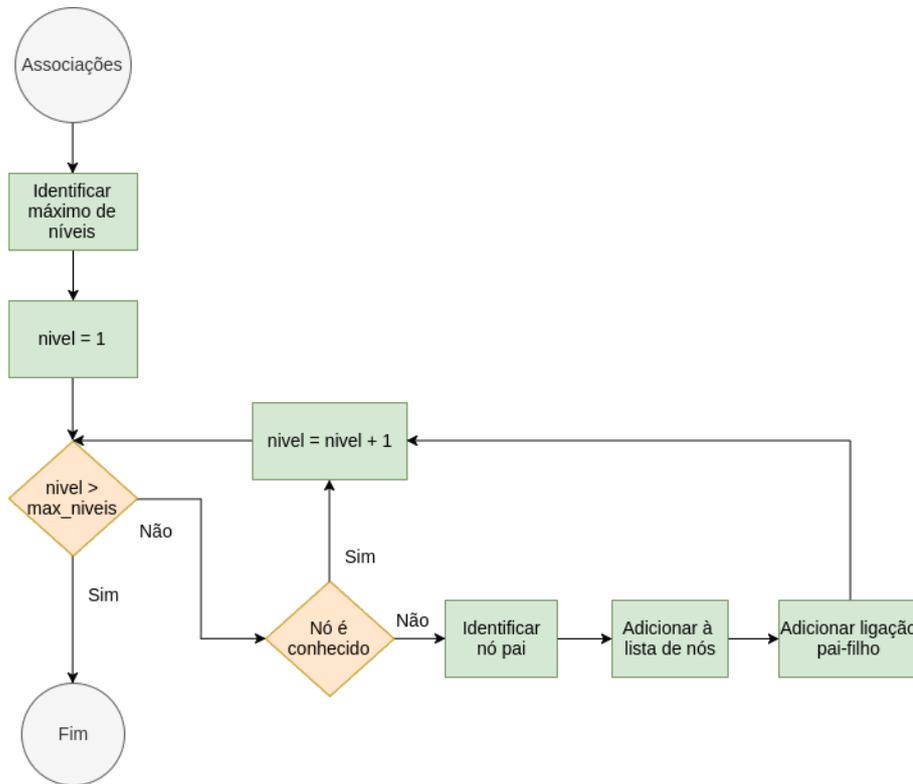


Figura 6.5: Fluxograma da construção das associações na visualização

Utilizando pseudo-código, a estrutura de dados da visualização seria idêntica ao que é apresentado de seguida.

```

data.nodes = [{
  label: 'Objeto 1',
  provider: 'Provedor A'
}, {
  label: 'Objeto 2',
  provider: 'Provedor A'
}]
  
```

```

data.links = [{
  from: 'Provedor A',
  to: 'Objeto 1'
}]
  
```

```
}, {  
  from: 'Provedor A',  
  to: 'Objeto 2'  
}]
```

Com o pseudo-código anterior seria possível gerar uma visualização de apenas um nível, semelhante à da figura 6.6. Esta imagem, tal como apresentada, é também definida por uma complexidade de grau 2 o que significa que não seria possível adicionar mais nenhum nó à visualização. De forma a disponibilizar mais objetos culturais do mesmo provedor, o utilizador tem duas opções:

- Aumentar para uma complexidade de grau 3;
- Aumentar para uma profundidade de nível 2.

A primeira hipótese irá disponibilizar mais um espaço no “Provedor A” fazendo com que seja possível estabelecer uma ligação entre este nó e o terceiro objeto cultural (figura 6.7). A outra opção permite adicionar um segundo nível à visualização. Assim, a ligação do próximo objeto cultural não será desenhada a partir do nó “zero” mas sim a partir dos nós do primeiro nível (figura 6.8). A escolha do nó de origem tem em conta se este se encontra associado com o novo objeto, ou seja, se o provedor é o mesmo e se pode alocar a ligação, uma vez que o parâmetro da complexidade é aplicável a qualquer nó da visualização.

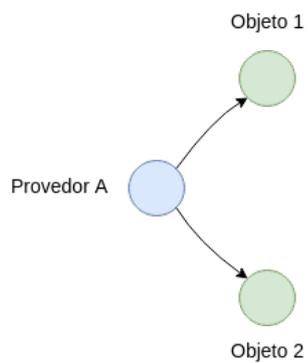


Figura 6.6: Visualização de um nível e complexidade grau 2

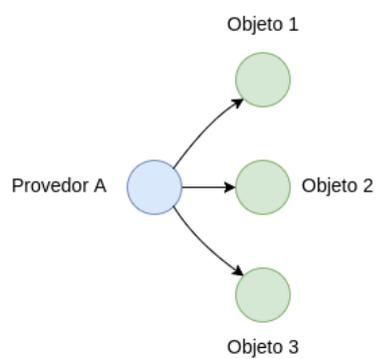


Figura 6.7: Visualização de um nível e complexidade grau 3

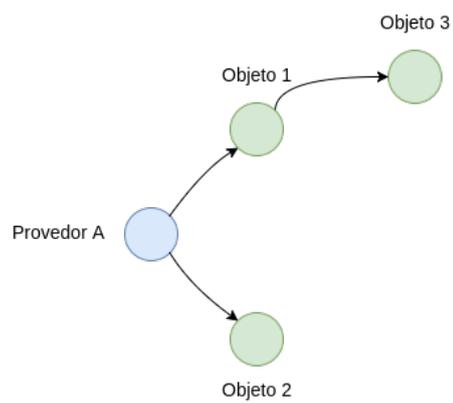


Figura 6.8: Visualização de dois níveis e complexidade grau 2

6.2.1 Interações

A escolha da biblioteca Vis.js foi baseada em vários fatores, entre eles, a possibilidade de construir uma rede de nós e ligações de uma forma nativa através do módulo para esse efeito. Esta ferramenta também permite definir interações do utilizador com a visualização. Aquelas que foram integradas no protótipo desenvolvido foram:

- *Clique* - Através do clique do rato sobre um nó ou neste caso, do clique duplo do rato, é despoletado um evento que faz abrir a página Europeia do objeto numa nova janela;
- *Zoom* - Utilizando o *scroll* do rato é possível aumentar ou diminuir a área da visualização que é mostrada;
- *Arrasto* - A opção de arrasto permite mover nós sobre a visualização e é útil quando é necessário estudar as ligações de uma determinada associação;
- *Hover* - Representa a sobreposição do cursor do rato sobre um elemento da visualização, ou neste caso, sobre um nó. Esta interação permite desencadear um evento que faz mostrar informação adicional sobre o objeto cultural, se esta existir.

6.2.2 Sumarização de conteúdo

A possibilidade de interagir com a rede de nós e mostrar informação complementar sobre um objeto cultural é algo que pode facilmente tornar-se numa experiência de visualização mais agradável e útil. No caso do evento de *hover*, foi implementado um elemento visual que mostra a sumarização do objeto cultural em cores, dispostas verticalmente.

Esta nova visualização é obtida através da captura sucessiva das *frames* do vídeo e do seu respetivo processamento para encontrar a cor dominante de

cada *frame*. A captura das *frames* do vídeo é feita com recurso à ferramenta *ffmpeg*¹, que permite manipular conteúdo de vídeo a partir de um ficheiro e guardar o seu resultado noutros formatos. As *frames* foram capturadas a cada 2 segundos e cada uma delas é depois guardada no formato *Portable Network Graphics* (PNG). De seguida, cada imagem é analisada de forma a extrair a sua cor dominante que é depois utilizada para gerar código HTML que dará origem à sumarização do vídeo em cores. Este processo encontra-se visível no fluxograma da figura 6.9.

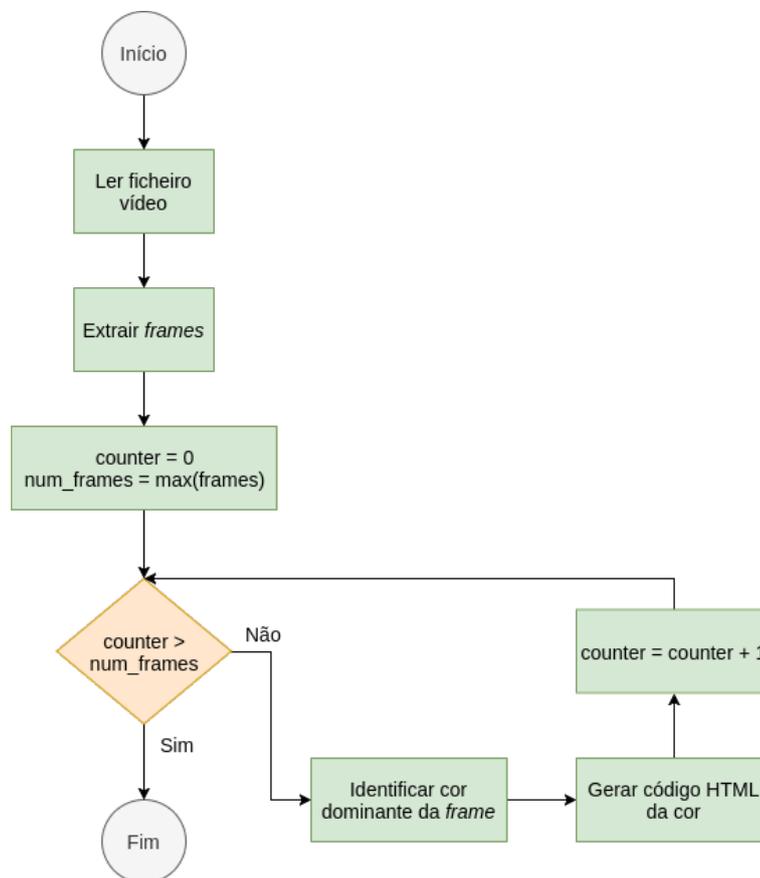


Figura 6.9: Fluxograma da criação da sumarização de cores

Para obter a cor dominante de cada *frame* foi utilizada a extensão *Graphics Draw* do PHP, uma biblioteca que disponibiliza uma série de métodos para

¹<https://ffmpeg.org/>

criar e manipular imagens de diferentes formatos. O algoritmo responsável por esta tarefa (figura 6.10) começa por identificar a altura e largura da imagem para depois iterar cada um dos seus pixels. Depois, é obtida a cor de cada pixel e guardada numa estrutura de dados a ocorrência dessa cor ao longo da imagem. Por fim, a estrutura de dados é ordenada de forma decrescente de frequência das cores, ou seja, na primeira posição da estrutura encontra-se a cor dominante da imagem enquanto que na última posição está a cor que menos vezes foi identificada.

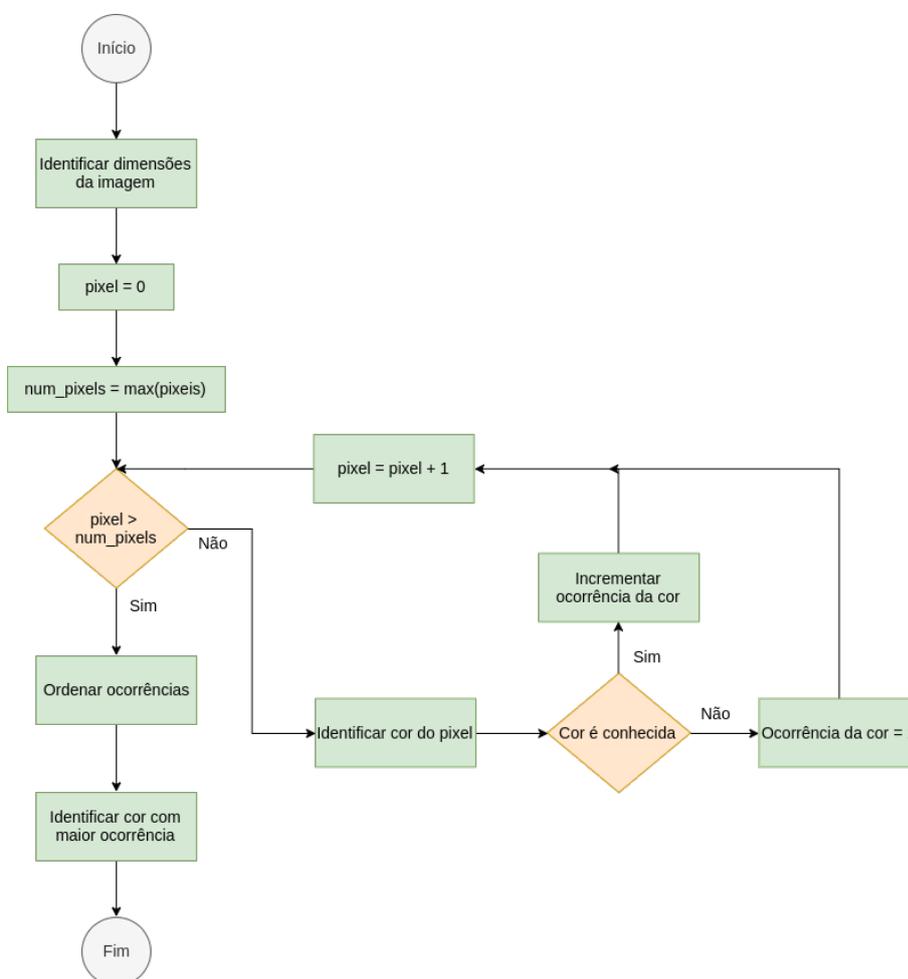


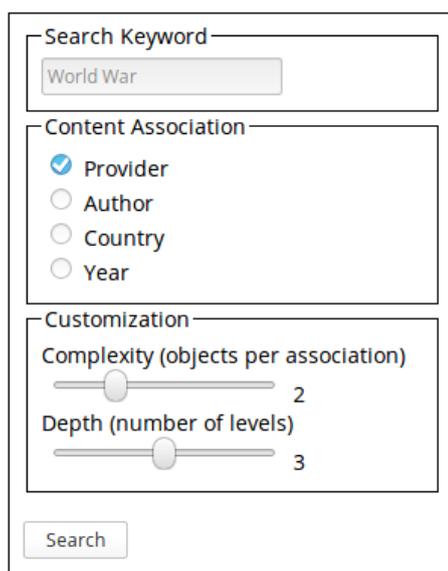
Figura 6.10: Fluxograma da extração da cor dominante de um *frame*

6.3 Resultados

Nesta secção são apresentados os testes realizados na validação do protótipo desenvolvido e os resultados obtidos. A descrição de cada teste será acompanhada por uma captura da ecrã do protótipo e, caso se aplique, um extrato da estrutura de dados da visualização.

6.3.1 Formulário de pesquisa

A primeira ação do utilizador quando acede ao protótipo é pesquisar sobre objetos culturais que pretende visualizar. Para isso, é necessário preencher um formulário para o efeito que se encontra disposto verticalmente do lado esquerdo da página (figura 6.11)



O formulário de pesquisa é composto por três seções principais:

- Search Keyword:** Um campo de texto contendo o valor "World War".
- Content Association:** Um grupo de botões de opção com o seguinte conteúdo:
 - Provider
 - Author
 - Country
 - Year
- Customization:** Duas barras deslizantes:
 - Complexity (objects per association):** Uma barra deslizante com o valor 2.
 - Depth (number of levels):** Uma barra deslizante com o valor 3.

Um botão "Search" está localizado na base do formulário.

Figura 6.11: Formulário de pesquisa

Este formulário já se encontra parcialmente preenchido com valores predefinidos para os parâmetros que o utilizador não é obrigado a especificar, como o tipo de associação, o grau de complexidade e o nível de profundidade da visualização. Os valores destes parâmetros são, respetivamente:

- Associação por provedor;

- Complexidade de grau 2;
- Profundidade de nível 3.

6.3.2 Visualização principal

Ao preencher automaticamente parte do formulário, é possível reduzir o tempo que o utilizador teria que despendar para iniciar a procura de objetos culturais. Desta forma, o utilizador apenas tem que introduzir o tema que deseja procurar, podendo no entanto customizar a resposta que lhe é retornada através dos parâmetros descritos na subsecção anterior. A alteração destes parâmetros faz com que o processamento da resposta Europeia se adapte de forma a respeitar as escolhas do utilizador. Por exemplo, no caso da escolha de uma associação por país, a visualização não seria inicializada com o desenho dos nós dos provedores (figura 6.12) mas sim dos países de origem (figura 6.13).

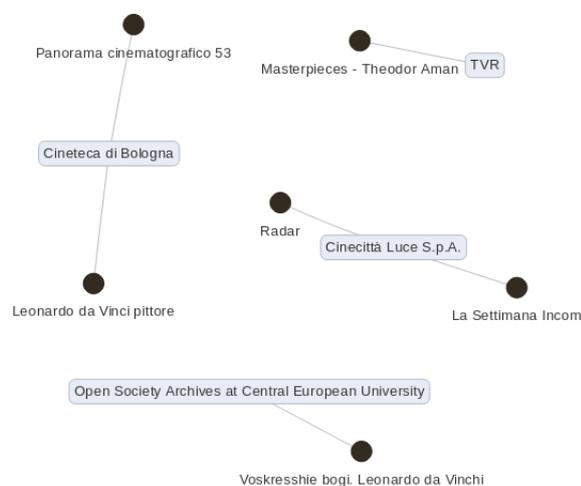


Figura 6.12: Associação por provedor



Figura 6.13: Associação por país

De forma a modificar o número de nós apresentados e a forma como são desenhados é necessário alterar os parâmetros de complexidade e profundidade. Tal como definido na secção 6, estas propriedades definem, respetivamente, o número de nós “filho” associados a cada nó “pai” e o número de níveis da visualização. Nas figuras 6.14 e 6.15 encontram-se representadas duas formas de configuração destas propriedades. A primeira, define-se por uma profundidade mínima, de nível 1 e uma complexidade máxima de grau 5, que resulta numa visualização multidireccional uma vez que os nós “filho” são dispostos à volta de cada nó “pai”, criando uma forma similar à de uma estrela. A segunda figura é definida pelo inverso, ou seja, uma profundidade máxima com 5 níveis e uma complexidade de grau 1. Com estas duas propriedades é obtida uma visualização unidireccional uma vez que os nós “filho” são quase que concatenados uns aos outros de forma a respeitar a escolha do utilizador.

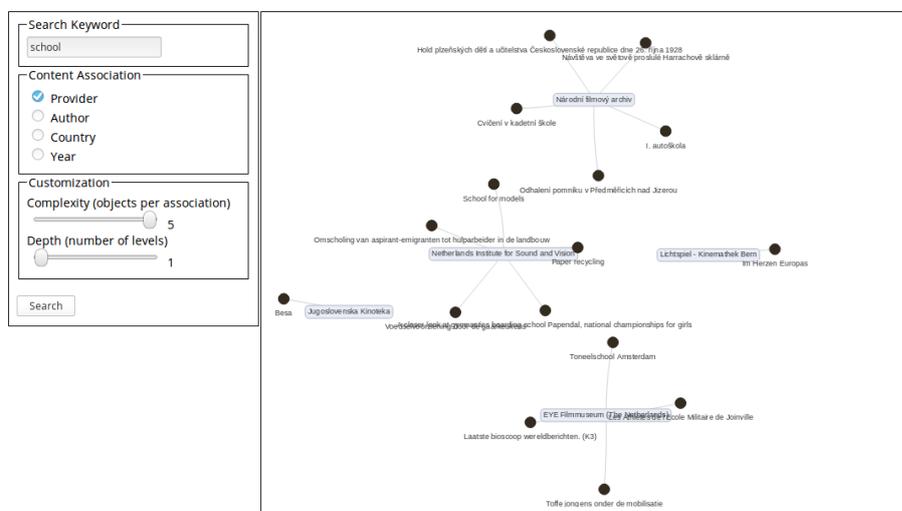


Figura 6.14: Visualização de profundidade 1 e complexidade 5

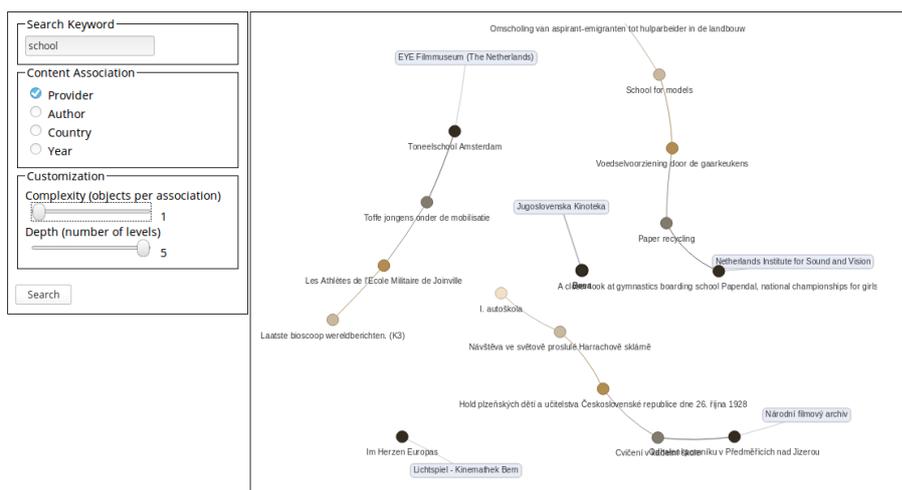


Figura 6.15: Visualização de profundidade 1 e complexidade 5

A estrutura de dados que dá origem às duas figuras anteriores é idêntica, nomeadamente na definição dos nós da visualização. No entanto, as ligações entre os nós têm que ter em conta as escolhas do utilizador e por isso, a definição das ligações é diferente nos dois testes. No teste da figura 6.14 as ligações são feitas sempre do nó “pai” para os nós “filho” tal como definido no excerto de código seguinte.

```
data.nodes = [{
  label: 'Objeto 1',
  parent: 'Provedor A'
}, (...), {
  label: 'Objeto 5',
  parent: 'Provedor A'
}]
```

```
data.links = [{
  from: 'Provedor A',
  to: 'Objeto 1'
}, (...), {
  from: 'Provedor A',
  to: 'Objeto 5'
}]
```

No teste da figura 6.15 as ligações atravessam os vários níveis da profundidade, sendo que um nó no nível dois pode ser “filho” de um nó de nível um e “pai” de um nó de nível três.

```
data.nodes = [{
  label: 'Objeto 1',
  parent: 'Provedor A'
}, (...), {
  label: 'Objeto 5',
  parent: 'Provedor A'
}]
```

```
data.links = [{
  from: 'Provedor A',
```

```
        to: 'Objeto 1'
    }, {
        from: 'Objeto 1',
        to: 'Objeto 2'
    }, (...), {
        from: 'Objeto 4',
        to: 'Objeto 5'
    }
}]
```

6.3.3 Visualização complementar

Para além da interface de visualização principal, onde são apresentados os objetos culturais e as suas relações, foi implementada uma visualização complementar que aplica o princípio de sumarização de conteúdo definido na secção 6.2.2. Esta visualização não se encontra disponível para todo o repositório de coleções Europeia, uma vez que isso implicaria o *download* e processamento de todos os objetos culturais, tornando-se assim numa tarefa que demoraria uma quantidade de tempo significativa. Em vez disso, para efeitos de demonstração, foi obtido um dos objetos disponíveis donde foram extraídas as suas *frames* e conseqüentemente, a sua cor dominante.

De forma a comprovar o seu funcionamento, é feita uma *query* com as palavras “educar para a diversidade” onde está contido o nome do objeto cultural processado previamente - “Educar para a diversidade : Cabo Verde”. Os restantes parâmetros de pesquisa não são alterados, ficando assim nos seus valores predefinidos. Esta visualização complementar é mostrada quando o cursor do rato é colocado sobre o respetivo nó (figura 6.16)



Figura 6.16: Sumarização das cores de cada *frame*

Esta página foi intencionalmente deixada em branco.

Capítulo 7

Conclusão

A realização desta dissertação tinha como objetivo analisar abordagens para navegação de conteúdos multimédia, com especial foco em recursos vídeo. Pretendia-se portanto, identificar formas diferentes e inovadoras de explorar grandes repositórios de dados, que permitam uma utilização mais apelativa deste tipo de conteúdo e que se distinguisse daquelas que são mostradas diariamente na generalidade dos *sites* de partilha de vídeos.

Para isso, foi feito o levantamento das técnicas de visualização que podem ser utilizadas para mostrar conteúdo multimédia, como é o caso das hierarquias que, através da categorização, reduz o tempo de procura de informação. Outra técnica abordada foi a utilização de um diagrama de rede ou diagrama nó-ligação que permite mostrar as associações entre os diversos objetos multimédia o que resulta numa experiência de navegação mais interativa em comparação com a tradicional lista vertical.

Depois de estudar as técnicas de navegação de conteúdo multimédia, foi desenvolvida uma interface gráfica de procura e navegação de objetos culturais, utilizando o repositório de coleções Europeia. Este desenvolvimento consiste na aplicação de um diagrama nó-ligação para apresentar os resultados da pesquisa. A forma como os vários recursos são dispostos na visualização pode ser customizada através dos parâmetros de complexidade

e profundidade. Foi também incluída uma visualização complementar que consiste na sumarização das cores do vídeo, extraídas a cada dois segundos do vídeo.

O protótipo desenvolvido e os seus constituintes permitem uma consulta rápida e eficiente de recursos multimédia, apoiada por um algoritmo de associação de objetos culturais, de forma a garantir que o utilizador consegue identificar os conteúdos que lhe interessam e as suas relações.

7.1 Trabalho futuro

O trabalho aqui descrito e os resultados obtidos são promissores no sentido em que é possível definir diferentes formas de apresentar conteúdos multimédia que podem tornar a experiência de utilização mais agradável.

Como formas de enriquecimento da interface de visualização, seria interessante implementar mecanismos de identificação de diferentes associações entre nós em simultâneo para, por exemplo, conseguir saber quais os objetos que têm o mesmo realizador ao mesmo tempo que se explora a visualização por provedor.

A existência de um *player* de vídeo embutido na interface gráfica seria uma melhoria bastante conveniente do ponto de vista do utilizador uma vez que, para aceder ao vídeo, não necessitaria de aceder à página da Europeia como acontece nesta versão do protótipo. Isto ajudaria a reduzir eventuais dispersões causadas pela mudança de contexto.

Para complementar a experiência de utilização com um ambiente de navegação mais interativo e imersivo, poderá ser utilizada uma visualização tridimensional onde o utilizador é colocado no centro da rede de nós e, através de uma vista “em primeira pessoa” - semelhante à navegação “*Street View*” do Google Maps - movimenta-se pelas ligações até encontrar o conteúdo que mais se relaciona com a sua pesquisa.

Por fim, seria importante realizar um estudo com um público-alvo de

forma a quantificar em que medida uma visualização deste tipo contribui para uma experiência de descoberta de conteúdo mais rica e interativa. Este estudo iria comprovar com maior precisão a adesão efetiva a esta alternativa, em detrimento dos métodos de navegação mais tradicionais.

Esta página foi intencionalmente deixada em branco.

Referências Documentais

- [1] I. M. Baytas, K. Lin, F. Wang, A. K. Jain, and J. Zhou, “PhenoTree: Interactive Visual Analytics for Hierarchical Phenotyping From Large-Scale Electronic Health Records,” *IEEE Transactions on Multimedia*, 2016.
- [2] “D3 Treemap - Flare visualization toolkit. Disponível em: <https://beta.observablehq.com/@mbostock/d3-treemap>. Acedido a 25-10-2018.”
- [3] N. U. K. Lab, “Timeline - Women in computing. Disponível em: <https://timeline.knightlab.com>. Acedido a 23/09/2017 .”
- [4] M. Wattenberg, “Arc diagrams: Visualizing structure in strings,” *Proceedings - IEEE Symposium on Information Visualization, INFO VIS*, vol. 2002-Janua, pp. 110–116, 2002.
- [5] U. Rashid, M. Viviani, and G. Pasi, “A graph-based approach for visualizing and exploring a multimedia search result space,” *Information Sciences*, vol. 370-371, pp. 303–322, 2016.
- [6] “Chord Diagram - Uber Rides by Neighborhood. Disponível em: <https://bost.ocks.org/mike/uberdata>. Acedido a 24/10/2018.”

- [7] K. Ren, R. Sarvas, and J. Čalíc, “Interactive search and browsing interface for large-scale visual repositories,” *Multimedia Tools and Applications*, vol. 49, no. 3, pp. 513–528, 2010.
- [8] M.-l. Viaud, O. Buisson, A. Saulnier, and C. Guenais, “Video Exploration: From Multimedia Content Analysis to Interactive Visualization,” *Proceedings of the international conference on Multimedia - MM '10*, no. December 2013, p. 1311, 2010.
- [9] C. X. Ma, Y. J. Liu, G. Zhao, and H. A. Wang, “Visualizing and Analyzing Video Content with Interactive Scalable Maps,” *IEEE Transactions on Multimedia*, 2016.
- [10] G. Schaefer, “Approaches for interactive browsing of large image datasets,” in *2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT)*, pp. 1–4, IEEE, oct 2016.
- [11] A. Anderson, “The Concept Map – The Conversation. Disponível em: <http://www.findtheconversation.com/the-concept-map>. Acedido a 03/10/2018.”
- [12] “Vega - Airport Connections Example. Disponível em: <https://vega.github.io/vega/examples/airport-connections>. Acedido a 04/10.2018.”
- [13] “ECharts Demo - Scatter plot. Disponível em: <https://ecomfe.github.io/echarts-examples/public/index.html#chart-type-scatter>. Acedido a 06/10/2018.”
- [14] Visjs.org, “Vis.js - Network. Disponível em: <http://visjs.org/examples/network/nodeStyles/images.html>. Acedido a 29/09/2017.”

- [15] Europeana, “Introduction to the Europeana APIs. Disponível em: <https://pro.europeana.eu/resources/apis/intro#edm> <http://labs.europeana.eu/api>. Acedido a 13/10/2018.”
- [16] R. Farahbakhsh, N. Crespi, Á. Cuevas, R. Cuevas, and R. González, “Understanding the evolution of multimedia content in the internet through bittorrent glasses,” *IEEE Network*, vol. 27, no. 6, pp. 80–88, 2013.
- [17] C. J. Downie and T. Peng, “Visualization of online datasets,” in *2017 {IEEE} 15th {International} {Conference} on {Software} {Engineering} {Research}, {Management} and {Applications} ({SERA})*, pp. 239–246, jun 2017.
- [18] C. Ware, *Information Visualization: Perception for Design*. 2013.
- [19] C. Kattmann and S. Tenbohlen, “Visualization of power quality data,” in *2017 IEEE Manchester PowerTech*, pp. 1–5, IEEE, jun 2017.
- [20] E. G. Caldarola and A. M. Rinaldi, “Improving the Visualization of WordNet Large Lexical Database through Semantic Tag Clouds,” in *2016 IEEE International Congress on Big Data (BigData Congress)*, pp. 34–41, IEEE, jun 2016.
- [21] D. A. Keim, “Information Visualization and Visual Data Mining,” *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS*, vol. 7, no. 1, 2002.
- [22] Y. G. Jiang, J. Wang, Q. Wang, W. Liu, and C. W. Ngo, “Hierarchical Visualization of Video Search Results for Topic-based Browsing,” *IEEE Transactions on Multimedia*, vol. PP, no. 99, pp. 2161–2170, 2016.
- [23] A. Arce-Orozco, L. Camacho-Valerio, and S. Madrigal-Quesada, “Radial Tree in Bunches: Optimizing the Use of Space in the Visualization of

- Radial Trees,” in *2017 International Conference on Information Systems and Computer Science (INCISCOS)*, pp. 369–374, IEEE, nov 2017.
- [24] “Fenótipo in Artigos de apoio Infopédia, Porto Editora. Disponível em: <https://www.infopedia.pt/dicionarios/lingua-portuguesa/fenotipo>. Acedido a 06/10/2018.”
- [25] E. C. Oliveira, L. C. Oliveira, A. Cardoso, L. Mattioli, and E. A. Lammounier Júnior, “Meta-model of information visualization based on Treemap,” *Universal Access in the Information Society*, vol. 16, pp. 903–912, nov 2017.
- [26] B. Shneiderman, “Tree visualization with tree-maps,” *ACM Transactions on Computer Graphics*, vol. 11, no. 1, pp. 92–99, 1992. Tree visualization with tree-maps, 1992.
- [27] B. Chen, H. Zhou, and X. Chen, “E-Embed: A time series visualization framework based on earth mover’s distance,” *Journal of Visual Languages & Computing*, vol. 48, pp. 110–122, oct 2018.
- [28] R. Lipka, “Scalable timeline visualization,” in *2016 9th International Conference on Human System Interactions (HSI)*, pp. 150–156, IEEE, jul 2016.
- [29] A. Imawan and J. Kwon, “A timeline visualization system for road traffic big data,” in *2015 IEEE International Conference on Big Data (Big Data)*, pp. 2928–2929, IEEE, oct 2015.
- [30] R. Barisic, “Cross-curricular education using interactive visualizations,” *Procedia - Social and Behavioral Sciences*, vol. 47, no. Figure 1, pp. 2157–2161, 2012.

- [31] J. Cardinal, M. Hoffmann, V. Kusters, C. D. Tóth, and M. Wettstein, “Arc diagrams, flip distances, and Hamiltonian triangulations,” *Computational Geometry*, vol. 68, pp. 206–225, mar 2018.
- [32] B. Saket, C. Scheidegger, and S. Kobourov, “Comparing Node-Link and Node-Link-Group Visualizations From An Enjoyment Perspective,” vol. 35, no. 3, 2016.
- [33] A. Jalali, “Reflections on the use of chord diagrams in social network visualization in process mining,” in *2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)*, pp. 1–6, IEEE, jun 2016.
- [34] N. Cao, Y. Wu, D. Gotz, D. Kiem, and Y.-P. Tan, “Guest Editorial : Visual Analytics in Multimedia — Opportunities and Research Challenges,” *Ieee Transactions on Multimedia*, vol. 18, no. 11, pp. 2133–2134, 2016.
- [35] D3js.org, “D3.js - Data-Driven Documents. Disponível em: <https://d3js.org>. Acedido a 05/08/2017.”
- [36] J. Robie, “What is the Document Object Model? Disponível em: <https://www.w3.org/TR/WD-DOM/introduction.html>. Acedido a 03/10/2018.”
- [37] “Vega - A Visualization Grammar. Disponível em: <https://vega.github.io/vega/about>. Acedido a 04/10/2018.”
- [38] D. Li, H. Mei, Y. Shen, S. Su, W. Zhang, J. Wang, M. Zu, and W. Chen, “ECharts: A declarative framework for rapid construction of web-based visualization,” *Visual Informatics*, vol. 2, pp. 136–146, jun 2018.
- [39] Visjs.org, “Vis.js - A dynamic, browser based visualization library. Disponível em: <http://visjs.org>. Acedido a 29/09/2017.”

- [40] A. Lopes, “Novo site RTP Arquivos. Disponível em: <http://media.rtp.pt/extra/pessoas/novo-site-rtp-arquivos>. Acedido a 19/01/2018,” 2017.
- [41] “Open Data - Introduction. Disponível em: <http://opendatahandbook.org/guide/en/introduction>. Acedido a 21/10/2018.”
- [42] Europeana, “Europeana - a European Cultural Heritage Platform for all. Disponível em: <https://ec.europa.eu/digital-single-market/en/europeana-european-digital-library-all>. Acedido a 21/01/2018.”
- [43] “What is Wikimedia Commons? Disponível em: <https://commons.wikimedia.org/wiki/Commons:Welcome>. Acedido a 21/10/2018.”
- [44] C. Meghini and W. P. Meeting, “The Europeana Data Model , current status,” no. May 2014, 2010.
- [45] A. Isaac, C. Meghini, M. Dekkers, S. Gradmann, R. Clayphan, J. Molendijk, J. Purday, and V. Charles, “Europeana Data Model Primer,” pp. 1–35, 2013.
- [46] C. Bizer, T. Heath, and T. Berners-Lee, “Linked data-the story so far,” *International journal on Semantic Web and Information Systems*, vol. 5, no. 3, pp. 1–22, 2009.
- [47] Oxford Dictionaries, “Definition of namespace. Disponível em: <https://en.oxforddictionaries.com/definition/namespace>. Acedido a 13/10/2018.”
- [48] R. Fielding, *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, 2000.

Referências Documentais

- [49] “Search API - Europeana Pro. Disponível em: <https://pro.europeana.eu/resources/apis/search>. Acedido a 28/10/2018.”
- [50] “Record API - Europeana Pro. Disponível em: <https://pro.europeana.eu/resources/apis/record>. Acedido a 28/10/2018.”
- [51] “SPARQL - Europeana Pro. Disponível em: <https://pro.europeana.eu/resources/apis/sparql>. Acedido a 28/10/2018.”
- [52] “OpenLink Virtuoso. Disponível em: <https://virtuoso.openlinksw.com>. Acedido a 15/10/2018.”
- [53] “OAI-PMH Service - Europeana Pro. Disponível em: <https://pro.europeana.eu/resources/apis/oai-pmh-service>. Acedido a 28/10/2018.”
- [54] “Entity API - Europeana Pro. Disponível em: <https://pro.europeana.eu/resources/apis/entity>. Acedido a 28/10/2018.”

Esta página foi intencionalmente deixada em branco.

Anexo A. Classes do Modelo de Dados Europeana

A.1. ore:Aggregation

Propriedade	Descrição
edm:aggregatedCHO	The identifier of the source object e.g. the Mona Lisa itself. This could be a full linked open data URI or an internal identifier.
edm:dataProvider	The name of the data provider of the object (i.e. the organisation providing data to an aggregator).
edm:hasView	The URL of a web resource which is a digital representation of the CHO. This may be the source object itself in the case of a born digital cultural heritage object.
edm:intermediateProvider	The name of the intermediate organization that selects, collates, or curates data from a Data Provider that is then aggregated by a Provider from which Europeana aggregates. The Intermediate Provider is distinct from both the Data Provider and the Provider in the data supply chain.
edm:isShownAt	The URL to the digital object on the provider's web site in its full information context.
edm:isShownBy	The URL of a web view of the object.

Anexo A.

edm:object	The URL of a representation of the CHO which will be used for generating previews for use in the Europeana portal. This may be the same URL as edm:isShownBy.
edm:provider	The name of the provider of the object (i.e. the organisation providing data directly to Europeana).
dc:rights	Information about rights held in and over the resource.
edm:rights	The rights statement that applies to the digital representation, as given (for example) in edm:object or edm:isShownAt/By, when these resources are not provided with their own edm:rights.
edm:ugc	Indicates whether the record has been contributed by the user community. It is a mandatory property for objects that are user generated or user created that have been collected by crowdsourcing or project activity. The property is used to identify such content and can only take the value "true" (lower case).

A.2. edm:EuropeanaAggregation

Propriedade	Descrição
dc:creator	A creator definitions. This field has always the value "Europeana".
edm:aggregatedCHO	The ID of the record corresponding to the CHO of this aggregation
edm:collectionName	This property is deprecated and edm:datasetName should be used instead.
edm:datasetName	This property holds the identifier given to the dataset in Europeana.

Anexo A.

edm:country	This is the name of the country in which the Provider is based or “Europe” in the case of Europe-wide projects.
edm:hasView	This property relates a ORE aggregation about a CHO with a web resource providing a view of that CHO. Examples of view are: a thumbnail, a textual abstract and a table of contents.
edm:isShownBy	An unambiguous URL reference to the digital object on the provider’s web site in the best available resolution/quality.
edm:landingPage	The URL of Europeana HTML object page. It captures the relation between an aggregation representing a cultural heritage object and the Web resource representing that object on the provider’s web site.
edm:language	A language assigned to the resource with reference to the Provider.
edm:preview	The URL of a thumbnail representing the digital object, generated by Europeana.
edm:rights	Information about copyright of the digital object as specified by isShownBy and isShownAt.
ore:aggregates	Aggregations, by definition, aggregate resources. The ore:aggregates relationship expresses that the object resource is a member of the set of aggregated resources of the subject (the Aggregation). This relationship between the Aggregation and its Aggregated Resources is thus more specific than a simple part/whole relationship, as expressed by dc-terms:hasPart for example.

Anexo A.

A.3. ore:Proxy

Propriedade	Descrição
dc:contributor	An entity responsible for making contributions to the resource.
dc:coverage	The spatial or temporal topic of the resource, the spatial applicability of the resource, or the jurisdiction under which the resource is relevant.
dc:creator	An entity primarily responsible for making the resource. This may be a person, organisation or a service.
dc:date	A point or period of time associated with an event in the lifecycle of the resource.
dc:description	A description of the resource.
dc:format	The file format, physical medium or dimensions of the resource.
dc:identifier	An unambiguous reference to the resource within a given context.
dc:language	A language of the resource.
dc:publisher	An entity responsible for making the resource available. Examples of a publisher include a person, an organisation and a service.
dc:relation	The name or identifier of a related resource, generally used for other related CHOs. The recommended best practice is to identify the resource using a formal identification scheme.
dc:rights	Name of the rights holder of the CHO or more general rights information. (Note that the controlled edm:rights property relates to the digital objects and applies to the edm:WebResource and/or edm:Aggregation).

Anexo A.

dc:source	A related resource from which the described resource is derived in whole or in part, i.e. the source of the original CHO.
dc:subject	The topic of the resource.
dc:title	A name given to the resource. Typically, a Title will be a name by which the resource is formally known.
dc:type	The nature or genre of the resource. Type includes terms describing general categories, functions, genres, or aggregation levels for content.
dcterms:alternative	An alternative name for the resource. This can be any form of the title that is used as a substitute or an alternative to the formal title of the resource including abbreviations or translations of the title.
dcterms:conformsTo	An established standard to which the described resource conforms.
dcterms:created	Date of creation of the resource.
dcterms:extent	The size or duration of the resource.
dcterms:hasFormat	A related resource that is substantially the same as the pre-existing described resource, but in another format.
dcterms:hasPart	A related resource that is included either physically or logically in the described resource.
dcterms:hasVersion	A related resource that is a version, edition, or adaptation of the described resource. Changes in version imply substantive changes in content rather than differences in format.
dcterms:isFormatOf	A related resource that is substantially the same as the described resource, but in another format.
dcterms:isPartOf	A related resource in which the described resource is physically or logically included.

Anexo A.

dcterms:isReferencedBy	A related resource that references, cites, or otherwise points to the described resource.
dcterms:isReplacedBy	A related resource that supplants, displaces, or supersedes the described resource.
dcterms:isRequiredBy	A related resource that requires the described resource to support its function, delivery or coherence.
dcterms:issued	Date of formal issuance (e.g., publication) of the resource.
dcterms:isVersionOf	A related resource of which the described resource is a version, edition, or adaptation. Changes in version imply substantive changes in content rather than differences in format.
dcterms:medium	The material or physical carrier of the resource.
dcterms:provenance	A statement of any changes in ownership and custody of the resource since its creation that are significant for its authenticity, integrity and interpretation. This may include a description of any changes successive custodians made to the resource.
dcterms:references	A related resource that is referenced, cited, or otherwise pointed to by the described resource.
dcterms:replaces	A related resource that is supplanted, displaced, or superseded by the described resource.
dcterms:requires	A related resource that is required by the described resource to support its function, delivery or coherence.
dcterms:spatial	Spatial characteristics of the resource.
dcterms:tableOfContents	Table Of Contents. A list of subunits of the resource.
dcterms:temporal	Temporal characteristics of the resource.
edm:currentLocation	The geographic location and/or name of the repository, building, site, or other entity whose boundaries presently include the resource.

Anexo A.

edm:europaProxy	Flag whether the proxy is an Europeana proxy. See chapter 6.2 Europeana proxies and data enrichment in the EDM primer.
edm:hasMet	edm:hasMet relates a resource with the objects or phenomena that have happened to or have happened together with the resource under consideration. We can abstractly think of history and the present as a series of “meetings” between people and other things in space-time. Therefore we name this relationship as the things the object “has met” in the course of its existence. These meetings are events in the proper sense, in which other people and things participate in any role.
edm:hasType	This property relates a resource with the concepts it belongs to in a suitable type system such as MIME or any thesaurus that captures categories of objects in a given field (e.g., the “Objects” facet in Getty’s Art and Architecture Thesaurus). It does not capture aboutness.
edm:incorporates	This property captures the use of some resource to add value to another resource. Such resources may be nested, such as performing a theater play text, and then recording the performance, or creating an artful edition of a collection of poems or just aggregating various poems in an anthology.
edm:isDerivativeOf	This property captures a narrower notion of derivation than edm:isSimilarTo, in the sense that it relates a resource to another one, obtained by reworking, reducing, expanding, parts or the whole contents of the former, and possibly adding some minor parts.

Anexo A.

edm:isNextInSequence	Relates two resources S and R that are ordered parts of the same resource A, and such that S comes immediately after R in the order created by their being parts of A.
edm:isRelatedTo	The most general contextual property defined in EDM. Contextual properties have typically to do either with the things that have happened to or together with the object under consideration, or what the object refers to by its shape, form or features in a figural or encoded form.
edm:isRepresentationOf	This property associates a resource to another resource that it represents.
edm:isSimilarTo	The most generic derivation property, covering also the case of questionable derivation. Is Similar To asserts that parts of the contents of one resource exhibit common features with respect to ideas, shapes, structures, colors, words, plots, topics with the contents of the related resource.
edm:isSuccessorOf	This property captures the relation between the continuation of a resource and that resource. This applies to a story, a serial, a journal etc. No content of the successor resource is identical or has a similar form with that of the precursor. The similarity is only in the context, subjects and figures of a plot. Successors typically form part of a common whole – such as a trilogy, a journal, etc.
edm:realizes	This property describes a relation between a physical thing and the information resource that is contained in it, visible at it or otherwise carried by it, if applicable.

Anexo A.

edm:type	The Europeana material type of the resource. All digital objects in Europeana have to be classified as one of the five Europeana material types using upper case letters: TEXT, IMAGE, SOUND, VIDEO or 3D. A broad classification of objects into five material types that users may find useful for filtering purposes: text, image, sound, video, 3D.
edm:unstored	This property should not be used and is only included here for backward compatibility with ESE.
edm:wasPresentAt	This property associates the people, things or information resources with an event at which they were present.
edm:year	A point of time associated with an event in the life of the original analog or born digital object.
ore:proxyFor	Proxy objects are used to represent a resource as it is aggregated in a particular aggregation. The ore:proxyFor relationship is used to link the proxy to the aggregated resource it is a proxy for. The subject of the relationship is a proxy object, and the object of the relationship is the aggregated resource.
ore:proxyIn	Proxy objects must also link to the aggregation in which the resource being proxied is aggregated. The ore:proxyIn relationship is used for this purpose. The subject of the relationship is a proxy object, and the object of the relationship is the aggregation

Anexo A.

A.4. edm:WebResource

Propriedade	Descrição
dc:creator	An entity primarily responsible for making the web resource. This may be a person, organisation or a service.
dc:description	An account or description of this digital representation.
dc:format	The file format, physical medium or dimensions of the resource.
dc:rights	Free text information about the rights in this object.
dc:source	A related resource from which the web resource is derived in whole or in part.
dcterms:conformsTo	An established standard to which the web resource conforms.
dcterms:created	Date of creation of the web resource.
dcterms:extent	The size or duration of the digital resource.
dcterms:hasPart	A related resource that is included either physically or logically in the web resource.
dcterms:isFormatOf	A related resource that is substantially the same as the described resource, but in another format.
dcterms:isReferencedBy	A related resource that references, cites, or otherwise points to the described resource. In IIIF, dcterms:isReferencedBy can be used to connect a edm:WebResource to a IIIF manifest URI.
dcterms:isPartOf	A related resource in which the described web resource is physically or logically included.

Anexo A.

dcterms:issued	Date of formal issuance (e.g., publication) of the resource.
edm:isNextInSequence	Where one CHO has several web resources, shown by multiple instances of the edm:hasView property on the ore:Aggregation this property can be used to show the sequence of the objects. Each web resource (apart from the first in the sequence) should use this property to give the URI of the preceding resource in the sequence.
edm:rights	The value in this element will indicate the usage and access rights that apply to this digital representation. The rights statement specified at the level of the web resource will "override" the statement specified at the level of the aggregation. The value in this element is a URI taken from the set of those defined for use in Europeana. A list of these can be found at http://pro.europeana.eu/web/available-rights-statements
owl:sameAs	Provide the URI of another web representation of the same resource.
edm:codecName	The name of a device or computer program capable of encoding or decoding a digital data stream or signal, e.g. "h264".
ebucore:hasMimeType	The main MIME type as defined by IANA: e.g. image/jpeg
ebucore:fileByteSize	The size of a media file in bytes.
ebucore:duration	The duration of a media file in ms.

Anexo A.

ebucore:width	The width of a media file in pixels.
ebucore:height	The height of a media file in pixels.
edm:spatialResolution	The spatial resolution of a media resource.
ebucore:sampleSize	The size of an audio sample in bits. Also called bit depth.
ebucore:sampleRate	The frequency at which audio is sampled per second. Also called sampling rate.
ebucore:bitRate	To provide the bitrate at which the MediaResource can be played in kilobits/second.
ebucore:frameRate	The frame rate of the video signal in frame per second.
edm:hasColorSpace	Whether an image is a coloured image.
edm:componentColor	A significant color present in an image. The colors are taken from the CSS3 standard color palette and are expressed as a hexadecimal binary value.
ebucore:orientation	The orientation of a Document or an Image i.e. landscape or portrait.
ebucore:audiochannelNumber	The total number of audio channels contained in the MediaResource.
svcs:has_service	The identifier of the Service require to consume the WebResource.