MODELADO DE COMUNIDADES PARA LA RECOMENDACIÓN DE CONTENIDOS EN MUSEOS
COMMUNITY MODELLING FOR THE RECOMMENDATION OF CONTENTS IN MUSEUMS

TRABAJO FIN DE GRADO
CURSO 2020-2021

AUTORES
IAGO ZAMORANO CHOUCIÑO
MARCOS RAFAEL NÚÑEZ
VADYM BATSULA BILENKA

DIRECTORES
GUILLERMO JIMÉNEZ DÍAZ
MARÍA BELÉN DÍAZ AGUDO

GRADO EN INGENIERÍA INFORMÁTICA
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

MODELADO DE COMUNIDADES PARA LA RECOMENDACIÓN DE CONTENIDOS EN MUSEOS
COMMUNITY MODELLING FOR THE RECOMMENDATION OF CONTENTS IN MUSSEUM

TRABAJO DE FIN DE GRADO EN INGENIERÍA INFORMÁTICA
DEPARTAMENTO DE INGENIERÍA DE SOFTWARE E INTELIGENCIA ARTIFICIAL

AUTORES
IAGO ZAMORANO CHOUCIÑO
MARCOS RAFAEL NÚÑEZ
VADYM BATSULA BILENKA

DIRECTORES
GUILLERMO JIMÉNEZ DÍAZ
MARÍA BELÉN DÍAZ AGUDO

CONVOCATORIA: JUNIO 2021
CALIFICACIÓN:

GRADO EN INGENIERÍA INFORMÁTICA
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

15 DE JUNIO DE 2021

RESUMEN

Modelado de comunidades para exploración y recomendación de contenidos en museos

Las comunidades llevan consigo una implícita relación de similitud entre los individuos que las conforman. Las relaciones entre los individuos y el diferente método que tengamos en cuenta a la hora de precisar ciertas características comunes determinan las comunidades que se puedan llegar a encontrar.

Este trabajo pretende desarrollar una serie de herramientas que permitan la detección de comunidades de usuarios, así como la visualización de las mismas. Se experimentará con diferentes técnicas de clustering explorando las diferentes comunidades resultantes. Se desarrollará una aplicación que permita la exploración de los diferentes resultados obtenidos, así como la experimentación a la hora de modificar las medidas de similitud y la influencia en los resultados.

Palabras clave

Clustering, similitud, arte, SPARQL, comunidades, distancia.

ABSTRACT

Community modeling for exploration and content recommendation in museums

The communities involve similarity relationship between individuals on it. Relationships between these people and the method we consider at the time of specifying various common features will determine the communities that may be found.

This works pretends to develop a set of tools that allow us to detect communities of users, as well as their visualization. There will be done experimentations with different clustering techniques exploring in the process the communities that will result from these experiments. An application will be developed with the purpose of exploring different results obtained, as well as testing different similarities and their influences.

Keywords

Clustering, similarity, art, SPARQL, communities, distance.

ÍNDICE DE CONTENIDOS

Resumen V

Abstract VII

Índice de contenidos IX

Índice de figuras XI

Índice de tablas XIII

Capítulo 1 - Introducción 1

1.1 Motivación 2

1.2 Objetivos 2

1.3 Plan de trabajo 3

Capítulo 2 - Estado de la cuestión 5

2.1 Técnicas de clustering 5

2.2 Herramientas de Linked Data 10

2.2.1 Wikidata 12

2.2.2 SPARQL 13

Capítulo 3 - Modelado de comunidades 15

3.1 Caso de estudio 15

3.2 Definición de medidas de similitud 16

3.2.1 Similitud de usuarios 17

3.2.2 Similitud de cuadros 19

3.3 Proceso de modelado 31

3.3.1 Búsqueda de pesos 32
3.3.2 Formación de comunidades 34
3.3.3 Visualización de las comunidades 35
3.3.4 Discusión de los resultados obtenidos 35
Capítulo 4 - Implementación 39
4.1 Preprocesamiento de los datos y primeros resultados 39
4.2 Arquitectura de la aplicación 43
4.2.1 Backend 44
4.2.2 Frontend 48
4.3 Tecnologías y recursos externos utilizados 54
4.3.1 Desarrollo 54
4.3.2 Coordinación 57
Capítulo 5 - Conclusiones y trabajo futuro 59
5.1 Conclusiones 59
5.2 Trabajo futuro 60
Capítulo 6 - Introduction 62
6.1 Motivation 63
6.2 Objectives 63
6.3 Work plan 64
Capítulo 7 - Conclusions and future work 67
7.1 Conclusions 67
7.2 Future work 68
Bibliografía 71
Apéndices 73

ÍNDICE DE FIGURAS

Figura 1 Modelo estructural del clustering jerárquico 7

Figura 2 Proceso de ajuste de centroides en el algoritmo de k-means 8

Figura 3 Esquema DBSCAN mínimo de 3 muestras por cluster 10

Figura 4 Estructura de la Red Semántica 11

Figura 5 Estructura de la página en Wikidata 12

Figura 6 Ejemplo de consulta SPARQL 13

Figura 7 Extensión de los conjuntos de obras valoradas por usuarios con obras similares 18

Figura 8 Los fusilamientos del tres de mayo 20

Figura 9 La maja desnuda 20

Figura 10 Depicts de la obra La Rendición de Breda de Velázquez 21

Figura 11 Superclases comunes entre conceptos de dos conjuntos de depicts 22

Figura 12 Construcción de los vectores de pesos de los conjuntos de depicts 23

Figura 13 Representación de una imagen segmentada con k-means 25

Figura 14 Representación de HSV 26

Figura 15 Calculo del MSE entre las imágenes de dos obras reescaladas 31

Figura 16 Esquema del algoritmo genético utilizado 34

Figura 17 Frecuencia edades usuarios 40

Figura 18 Frecuencia movimientos artísticos 40

Figura 19 Frecuencia polaridad sentimientos usuarios 40

Figura 20 k-means demográficos 41

Figura 21 Etiquetado de los clusters mediante FCA 42

Figura 22 Jerárquico aglomerativo demográficos 42

Figura 23 Esquema con los diferentes módulos de la aplicación 44

Figura 24 Estructura de la lógica del modelado 45
Figura 25 Estructura del cuerpo de una petición 48
Figura 26 Portada de la aplicación web 50
Figura 27 Interfaz para la visualización de la información de la técnica 51
Figura 28 Información de los datos demográficos de un cluster 52
Figura 29 Obras de arte más y menos populares dentro del cluster 52
Figura 30 Artistas mejor y peor valorados, movimiento artístico 53
Figura 31 Visualización de la interfaz generadora de cuadros similares 54

ÍNDICE DE TABLAS

Tabla 1 Pesos de las similitudes parciales entre cuadros propuestos por un profesional en arte . 36

Tabla 2 Pesos de las similitudes parciales entre cuadros encontrados con el algoritmo genético 37

Tabla 3 Pesos de las similitudes parciales entre cuadros con el peso de la Similitud por artista y movimiento al máximo 37

Capítulo 1 - Introducción

El ser humano es social por naturaleza. A lo largo de toda su historia, ha tratado de organizarse en grupos de individuos para, no solo intentar sobrevivir, si no poder compartir hábitos, gustos, cultura o rutinas con los miembros de dicho grupo. El sentimiento de pertenencia a una comunidad con la que se comparten diferentes características puede ser muy gratificante para las personas que la componen.

Son diversos los campos de estudio que sacan provecho del concepto de comunidad a la hora de realizar estudios sobre la población, ya que resulta más sencillo encontrar estructura o caracterizar grupos de personas con características comunes que estudiarlos individualmente.

Es precisamente en la idea de compartir elementos o características donde reside la potencia de poder modelar o detectar comunidades, ya que, si somos capaces de reunir a un grupo de individuos en torno a una idea, afición, costumbre o valor común, además de promover la socialización y comunicación, lo cual siempre es positivo para el progreso como persona, será más fácil dirigirles mensajes, ofrecerles productos o recomendarles cualquier tipo de contenido.

Por otro lado, vivimos en un momento en el que la adquisición de datos a partir de sensores, tecnologías móviles o simplemente navegación por internet vive un momento álgido. Esto, junto con las distintas técnicas para la extracción de la información conforman lo que se conoce como Big-Data. La aplicación de estas técnicas permite entre otras muchas, la detección de comunidades por medio de mecanismos de aprendizaje automático.

Por ello, con este trabajo buscamos desarrollar una serie de herramientas de detección y visualización de comunidades. Estas herramientas tendrán distintas aplicaciones como encontrar una serie de comunidades de usuarios de museos en base a una serie de valoraciones acerca de distintas obras, junto con ciertas características demográficas de los mismos, con el objetivo de poder así estudiar sus gustos con mayor facilidad y que estas comunidades puedan ser de utilidad a la hora de optimizar nuestras herramientas para realizarles recomendaciones, tanto de obras que podrían resultarles interesantes, como de personas con las que con bastante seguridad compartirán gustos en el arte.

1.1 Motivación

El objetivo de este trabajo es desarrollar herramientas para encontrar, modelar, caracterizar y visualizar una serie de comunidades de personas en museos de arte. Partiendo de un caso de estudio de un conjunto de usuarios de los que se tienen datos demográficos y una serie de opiniones acerca de diversas obras de arte, la idea es poder encontrar comunidades de usuarios explorando las similitudes y diferencias entre dichos usuarios, tratando de encontrar rasgos comunes, ya sean a nivel demográfico o de valoración artística. Aunque las herramientas son específicas para nuestro dominio (arte), las técnicas de modelado se pueden aplicar a otros campos adaptando las medidas de similitud en las que se basan las comunidades.

Aplicando distintos tipos de modelado y algoritmos el objetivo es identificar y visualizar estos grupos para encontrar un mecanismo que nos permita concretar dichos grupos, para que luego puedan ser utilizados con otros fines como pueden ser el de la recomendación de contenidos del museo, trazado de rutas que más puedan agradar al usuario en función de sus opiniones o la interacción de personas en la misma comunidad.

1.2 Objetivos

Los objetivos más relevantes de nuestro trabajo en cuanto contribución son los siguientes:

- Realizar un estudio comparativo de las técnicas de clustering, sus diferencias y similitudes para tratar de averiguar cuál puede ser la más adecuada, tanto de cara a llegar al objetivo de manera óptima, con el menor coste de tiempo posible, como para formar comunidades en función de la situación y el contexto en que se quieran aplicar.
- Definir una medida de similitud entre cuadros que podamos utilizar para relacionar personas en base a su opinión/valoración sobre ciertas obras. En este proceso se definen una serie de similitudes parciales basadas en diferentes características de los cuadros, sobre las cuales utilizaremos unas medidas de peso concretas según ciertos criterios, para finalmente combinarlas en una similitud general. De manera paralela, se experimentará con el impacto que tiene la medida de similitud en el proceso de detección de comunidades.

- Desarrollar una interfaz a partir de un caso de estudio con datos reales que nos permita configurar variaciones de esta medida de similitud entre cuadros, permitiendo visualizarlos, y pudiendo ajustar los pesos de las similitudes parciales para ver como varían los resultados en función de los mismos.
- Modelar una serie de comunidades utilizando técnicas de clustering que definan la distancia entre usuarios en función de las distintas medidas de similitud, con el fin de encontrar otros usuarios que hayan valorado de la misma manera las mismas obras u otras similares.
- Experimentar con herramientas de visualización existentes y estudiar su adecuación en base las características de las comunidades detectadas en el dominio del trabajo. Proponer alguna de ellas que permita caracterizar las comunidades en base a distintos criterios ya sean demográficos, de gustos sobre arte o visualizar de alguna manera su cohesión.

Los objetivos a nivel de aprendizaje personal son los siguientes:

- Enfrentarnos durante el desarrollo del trabajo con conceptos y técnicas de programación o de diseño que no nos resulten familiares o que no hayamos estudiado en el grado para poder así ampliar nuestros conocimientos y competencias.
- Descubrir y utilizar diferentes herramientas nuevas que nos ayuden a desarrollar nuestras ideas, tanto por parte técnica como teórica.

1.3 Plan de trabajo

Una vez hemos expuesto cual es la motivación de este trabajo y cuáles son los objetivos que nos gustaría cumplir, pasamos a ver como hemos organizado el progreso del mismo.

En las primeras fases del proyecto, procuramos familiarizarnos con las diferentes técnicas de clustering, estudiando las características y el funcionamiento de cada una de ellas. Las

conclusiones y resumen de la revisión de las técnicas de clustering se han incluido en el Capítulo 2 (Estado de la cuestión).

Una vez conocimos en detalle dichas técnicas, las aplicamos sobre un conjunto de datos demográficos para tratar de entender qué nivel de relevancia podían tener a la hora de definir comunidades para así poder descartar ciertas de ellas, y mantener las que resulten satisfactorias. Posteriormente, como el resultado del análisis de los datos demográficos no resultaba del todo relevante, decidimos cambiar el enfoque y considerar los datos de valoraciones de los usuarios sobre las obras, que finalmente serían los que utilizaríamos para modelar las comunidades. Para trabajar en base a los datos de opinión de usuario, decidimos plantear una medida de similitud entre las obras, la cual estaría compuesta por diversas medidas parciales que considerasen distintos atributos de los cuadros como se detalla en el Capítulo 3 (Modelado de comunidades).

Para poder visualizar y comprender a fondo la medida de similitud entre obras, hemos desarrollado una interfaz que permitiera visualizar los cuadros más similares a uno dado, ajustando las medidas de similitud empleadas. Esta parte se detalla en el Capítulo 4 (Implementación), tanto la parte relativa a su funcionamiento (backend) como la visual (frontend).

Además, hemos buscado algún método o criterio para poder definir con qué pesos comparamos las obras a la hora de crear las comunidades, ya sea desarrollando alguna herramienta para su búsqueda u obteniendo información relevante para su decisión

Para llevar cabo la detección de comunidades aplicamos las medidas de similitud establecidas entre los cuadros para encontrar usuarios que valoren de forma parecida las obras. Finalmente, representaremos los datos más relevantes de estas comunidades de usuarios utilizando diferentes herramientas de visualización.

Capítulo 2 - Estado de la cuestión

Para poder entender el proceso de detección de comunidades primero hay que explicar el contexto en el que se encuentran las distintas técnicas y procesos utilizados.

La abrumadora cantidad de datos que hoy por hoy se generan (algo más de 2,5 exabytes según IBM), hace de ellos, una entrada de información de gran volumen. La naturaleza de los datos abarca casi cualquier campo hoy en día. Desde datos médicos hasta datos bancarios. Esta gran cantidad de datos es lo hoy se conoce por Big Data. Sin embargo, el Big Data no solo consta de la generación y adquisición de datos sino también de las técnicas que se aplican para extraer información relevante de los datos. Algunas de esas técnicas son las denominadas técnicas de clustering. En concreto, aquellas técnicas de clustering capaces de revelar cierta estructura dentro de los individuos. Por esta razón, durante el desarrollo de este trabajo se han aplicado distintas técnicas de clustering [1].

Para poder aplicar estas técnicas de clustering es necesario la recuperación de datos. El paradigma Linked Data ofrece numerosas herramientas para poder acceder a distintos datos utilizando en nuestro beneficio las redes semánticas sobre las que se definen.

2.1 Técnicas de clustering

La utilización del modelo de aprendizaje no supervisado es fundamental, al permitir el aprendizaje automatizado para entrenar la máquina y así poder conocer y agrupar los conjuntos de datos. Esto es necesario, frente al aprendizaje supervisado, porque a priori no tenemos información sobre la clase a la que pertenecen los elementos, por tanto, debemos descubrir patrones y características que permitan agrupar los datos de la manera más precisa posible.

Existen dos técnicas básicas de exploración en el aprendizaje no supervisado: las de agrupamiento o clustering, basadas en los individuos y las de reducción de dimensionalidad, que se basan en las variables [2].

En las técnicas de reducción de dimensionalidad, el objetivo es reducir un conjunto de variables en subconjuntos que difieran lo mínimo posible de sus originales. Muchos factores resultantes vienen de la combinación de varias variables. Las variables originales que más

impacten en un factor serán las que se relacionen entre sí resultando en subgrupos comunes [3]. La reducción de dimensionalidad puede presentar los siguientes problemas:

- Obtener una cantidad de dimensiones tan elevada que no contenga los suficientes ejemplos para caracterizar ciertos conjuntos.
- A la hora de encontrar variables que impacten en los factores, puede haber algunas que no ofrezcan ninguna contribución y confundan el sistema.
- Su visualización no es tan sencilla como otros algoritmos.

En este caso, la técnica que nos concierne es el clustering. De cara a afrontar el trabajo que se nos presenta, los principales problemas que nos encontramos son:

- Seleccionar las variables más importantes, las que realmente tengan impacto en el aprendizaje automatizado, dado que es bastante probable que haya varias de ellas irrelevantes y que no ofrezcan ninguna utilidad al proceso, ya sea porque sean demasiado escuetas o porque no sean lo suficientemente descriptivas.
- La representación que pueden tener algunos atributos. Esto complica la preparación del algoritmo de clustering, dado que puede haber variables entre sí con formatos no representativos o que no puedan relacionarse de manera directa, ya sea por formato, o por tipo; lo cual fuerza a refactorizar muchos de ellos de manera coherente y eficiente.

El objetivo final del proceso clustering es generar conjuntos de individuos lo más parejos entre sí dentro del conjunto y lo más diferentes posible con los de otros grupos. Las medidas de similitud que definen estas distancias entre individuos y conjuntos son las resultantes de resolver los problemas mencionados anteriormente y definiendo los atributos que mejor caractericen a los objetos.

Existen tres principales familias de técnicas de agrupamiento: clustering jerárquico y clustering basado en particiones y clustering basado en densidad.

En cuanto al agrupamiento jerárquico (Figura 1 Modelo estructural del clustering jerárquicoFigura 1), llamamos de esta manera al método que agrupa los datos según la distancia

entre cada uno de ellos y buscando que los que están dentro de un mismo cluster son los más similares. Un ejemplo claro es el aglomerativo, que comienza definiendo tantos clusters como individuos haya y a partir de este punto ir agrupando los más cercanos entre sí hasta formar un solo conjunto. El punto más importante de esto es definir la distancia que va a medir la proximidad entre elementos [4]. Habitualmente se utilizan las métricas Minkowski. Una vez elegida la métrica, se designan los puntos de referencia que se medirán. En este caso tenemos tres opciones:

- Centroide: Mide la distancia entre los puntos medios de cada cluster.
- Enlace simple: Se tiene en cuenta la medida entre los puntos más cercanos entre conjuntos.
- Enlace completo: Se toman los puntos más alejados de los cluster.

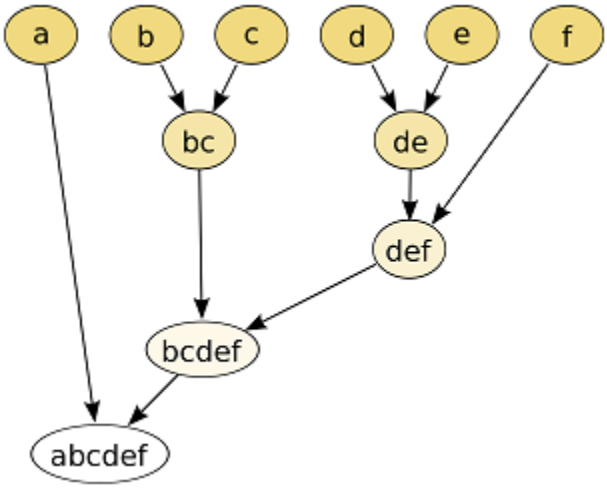

Figura 1 Modelo estructural del clustering jerárquico

En el caso del algoritmo basado en particiones, su objetivo es dividir el espacio en k clusters en los cuales se sitúan los individuos de la mejor manera posible. El más común es k-means (Figura 2), y su manera de funcionar es la siguiente: Tras fijar el valor k mencionado anteriormente, se generan k puntos aleatorios en el espacio llamados centroides. A continuación, se asignan los objetos que buscamos agrupar al centroide más cercano en el área y se vuelve a calcular el centroide de ese cluster en base a la nueva asignación de participantes del mismo. Este proceso se

repite hasta que los centroides se mantengan en el sitio, lo cual en ese momento significa que hemos llegado a la disposición más optimizada.

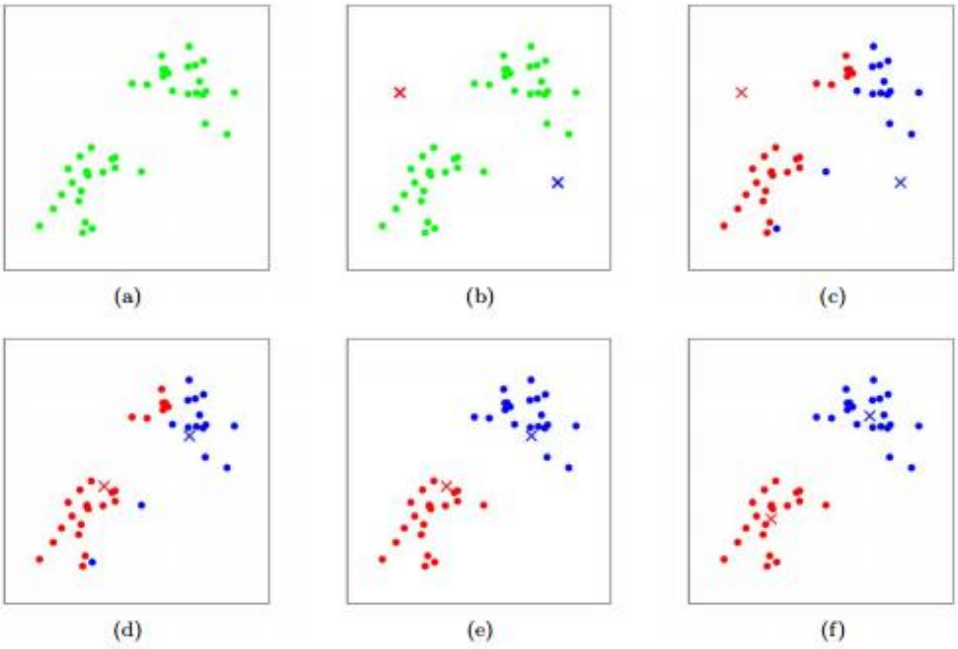

Figura 2 Proceso de ajuste de centroides en el algoritmo de k-means

Por otro lado, para el aprendizaje supervisado se necesitan previamente ejemplos ya clasificados, para así poder el sistema aprender funciones que permitan establecer reglas más precisas de clasificación.

El aprendizaje supervisado se divide, según su variable, de las siguientes dos maneras:

- Si la variable es numérica, se trata de un problema de regresión.
- Si, por el contrario, la variable es categórica, el problema es de clasificación.

Entre las técnicas más relevantes del aprendizaje supervisado, se encuentra el k-NN (k nearest neighbors). Se trata de un algoritmo basado en criterios de vecindad, el cual, a través de unos factores de clasificación dados inicialmente, se van reagrupando los individuos en base a la cercanía con otros que sean más parecidos [5].

Por otro lado, existe la técnica basada en árboles de decisión, la cual, a raíz de un conjunto inicial, se construye un árbol donde cada nodo pregunta por el valor de una variable que servirá para clasificar nuevas entradas que vayan ocurriendo.

De cara a la utilización de medidas de similitud, debemos encontrar técnicas que puedan trabajar con matrices de distancia entre los diferentes individuos. Los algoritmos de clustering que hemos estudiado anteriormente, tratan los datos como puntos en el espacio, entre los que mide la distancia mediante diferentes métodos. En este caso no contamos con puntos en el espacio sino con una herramienta que mide la distancia entre dos individuos, por tanto, necesitamos utilizar métodos que puedan calcular clusters con ella.

En este sentido, consideramos primeramente el método de k-medoids [6], que es una modificación de la técnica k-means. La principal diferencia es que k-medoids utiliza uno de los datos como centro del cluster, en vez de calcular el punto medio representativo de cada grupo en el espacio. Es por ello por lo que permite trabajar con una medida de distancia personalizada sin necesidad de definir a unos individuos en puntos concretos del espacio.

Otro método que se puede adecuar con facilidad es DBSCAN [7](Density Based Spatial Clustering of Applications with Noise) que pertenece a la familia de métodos basados en densidad. Esta técnica itera sobre los datos agrupando en un mismo cluster todos los puntos que estén a una distancia igual o menor que un valor épsilon (Figura 3). Es por ello por lo que se da el caso en que ciertos puntos de los datos no sean incluidos en ningún cluster al no estar dentro del rango mínimo de proximidad con respecto a ningún otro y sean etiquetados como ruido. En este sentido entra en juego el segundo parámetro con el que trabaja DBSCAN, el tamaño mínimo de los clusters. Este último valor define cuantos valores son necesarios para poder definir un cluster. Obsérvese que, si este último valor se define a 1, cada dato que anteriormente se etiquetaba como ruido al no estar cerca de ningún otro dato, conformara su propio cluster. Este método supone una ventaja con respecto a los anteriores al crear clusters densos en los que sus datos están con un mínimo de proximidad entre ellos, ya que los anteriores incluyen todos los datos en un cluster independientemente de la distancia que los separe.

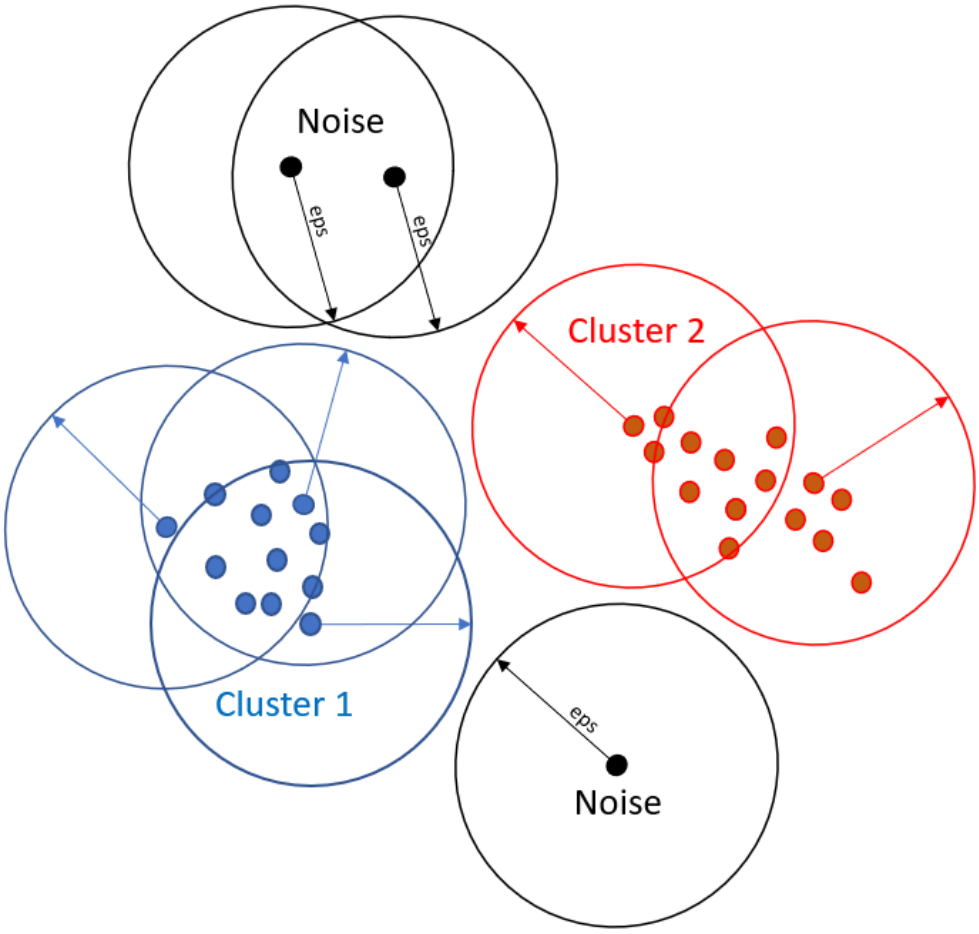

Figura 3 Esquema DBSCAN mínimo de 3 muestras por cluster

2.2 Herramientas de Linked Data

La Web Semántica [8] es una extensión de la World Wide Web que procura darle una estructura más organizada al conocimiento en la misma. Creada en 1990 en el CERN por Tim Berners-Lee y Robert Cailliau, la World Wide Web se basa en una colección de documentos de texto no estructurados, pensado para ser utilizados e interpretado por personas. Al carecer de una organización clara, realizar búsquedas precisas por significado o relaciones no es posible, reduciéndose las posibilidades a encontrar meras coincidencias textuales.

Es por ello por lo que en 2001 Tim Berners-Lee propone la idea de la Web Semántica, buscando dar una estructura más clara al conocimiento que contiene la web y que así pueda también ser accedido, interpretado y utilizado por agentes software y no solo por personas. De esta iniciativa surgen distintas tecnologías que tratan de dotar a la web de una estructura semántica

mediante diferentes formalismos como son los URIs (Uniform Resource Identifier), RDFs (Resource Description Framework) o las ontologías entre otras. Así mismo, surgen otras herramientas que hacen uso de estos estándares como son las bases de conocimiento, los razonadores o lenguajes de consulta como SPARQL.

Con la aplicación de esta colección de formalismos conseguimos dar una cierta estructura a la web, anotando parte de los recursos presentes en ella y estableciendo relaciones entre ellos, estableciendo así una Red Semántica.

Se define la Red Semántica como una forma de representar el conocimiento mediante un grafo dirigido en al cual los nodos son conceptos y los arcos contienen una etiqueta que define la relación especifica entre cada nodo. Son precisamente estas tripletas sujeto-propiedad-objeto las que definen la red (Figura 4). Los sujetos no son más que recursos en el ámbito de la red mientras que los objetos pueden ser o bien otro recurso o bien un literal. Por último, la propiedad es la etiqueta de la relación, cuyo objetivo es caracterizarla.

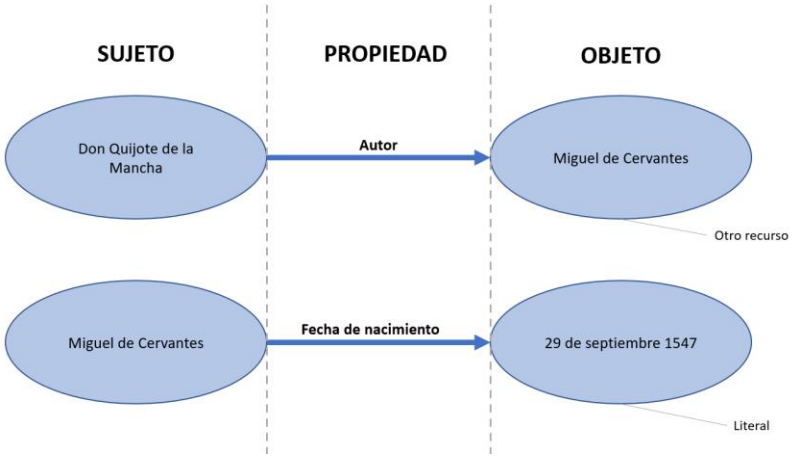

Figura 4 Estructura de la Red Semántica

Por último, el concepto de Linked Data [9] surge de esta colección de datos interrelacionados que siguen una estructura determinada por algún estándar de representación. Todos estos conceptos nos serán muy útiles a la hora de realizar nuestro proyecto, ya que nos permitirán recuperar diferentes propiedades de obras de arte para extender el conocimiento que tenemos de ellos que podamos utilizar a la hora de formar comunidades.

2.2.1 Wikidata

Wikidata es una base de conocimiento libre, colaborativa, multilingüe, secundaria que recolecta datos estructurados para proporcionar soporte a Wikipedia, Wikimedia Commons, otras wikis del movimiento Wikimedia o a cualquier usuario en el mundo [10].

La plataforma funciona como un repositorio centralizado de almacenamiento de datos que puede ser accedido por cualquiera. Está compuesto principalmente de entidades, cada una con su etiqueta y sus alias, una descripción (estos campos se ofrecen en todos los idiomas posibles) y un identificador único formado por la letra Q sucedida por un identificador numérico, por ejemplo, La rendición de Breda de Velázquez está identificada en Wikidata por Q1133420.

Además de los campos mencionados anteriormente, cada entidad de Wikidata cuenta con un conjunto de tuplas propiedad-objeto a los que conocemos como sentencias (statements). Junto con la entidad que las contiene, estas tuplas forman las tripletas que estudiábamos anteriormente, estableciendo una relación entre la entidad y el objeto (se puede tratar de un valor concreto o de otra entidad Wikidata). Dicha relación viene caracterizada por la propiedad que une ambas entidades.

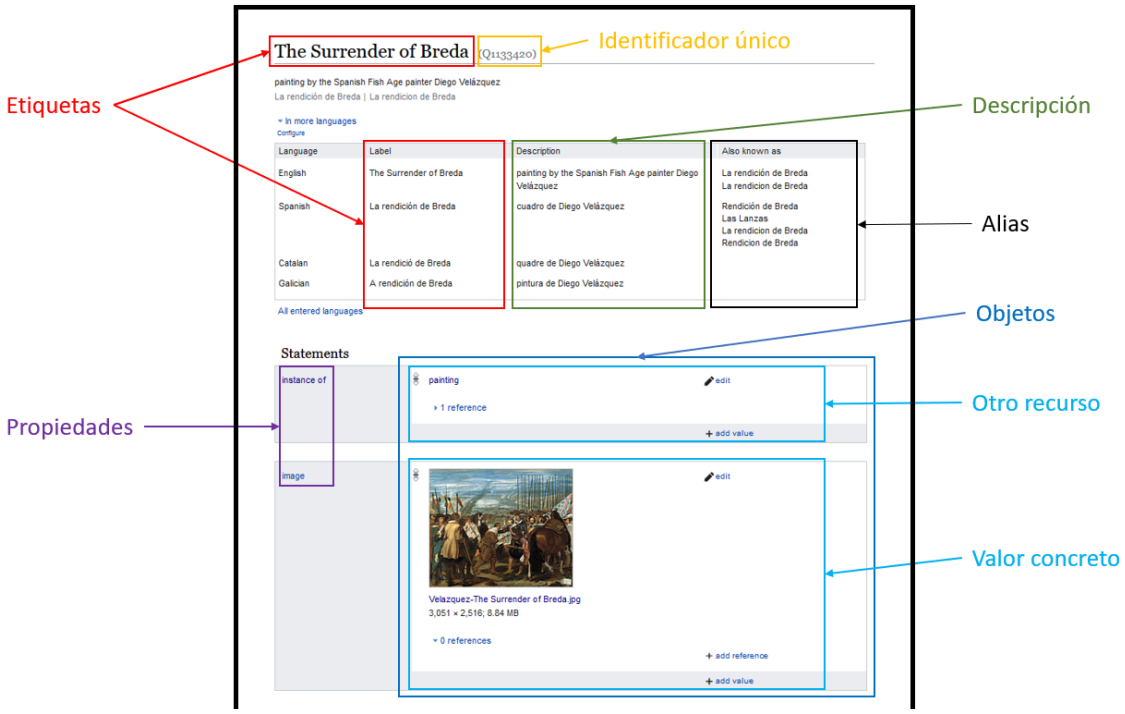

Figura 5 Estructura de la página en Wikidata

2.2.2 SPARQL

SPARQL (acrónimo recursivo para SPARQL Protocol and RDF Query Language) es un lenguaje de consulta sobre distintas estructuras RDF. En este sentido, nos permite buscar y recuperar cualquier información dentro de una estructura de datos enlazados como es Wikidata, permitiendo definir distintos parámetros en la consulta como las propiedades deseadas, restricciones, idioma, condiciones o límites.

En lo que compete a este trabajo, Wikidata ofrece un servicio de consultas SPARQL que permite acceder a su base de datos enlazados con una sintaxis sencilla e intuitiva conocido como WDQS (Wikidata Query Service). Observando el ejemplo expuesto a continuación (Figura 6), podemos distinguir los siguientes componentes en una consulta: la cláusula SELECT lista las variables a devolver por la consulta; el modificador DISTINCT asegura que no se devuelvan elementos duplicados; el campo WHERE establece las restricciones sobre la consulta, normalmente en forma de tripletas que recuperan los valores de las variables de la cláusula inicial; los prefijos wd y wdt son los propios de Wikidata que recuperan las entidades y propiedades respectivamente. Estas son, entre muchas otras, las principales componentes de una consulta SPARQL y son las que básicamente necesitaremos a la hora de recuperar propiedades de nuestras obras, aunque existen multitud de cláusulas más que definen diferentes operaciones más como ordenamientos, agrupamientos o la posibilidad de imponer condiciones basadas en los valores de las variables.

```
SELECT DISTINCT ?artist ?artistLabel WHERE {
  wd:Q1133420 wdt:P170 ?artist.
  SERVICE wikibase:label { bd:serviceParam wikibase:language"[AUTO_LANGUAGE],en". }
}
LIMIT 100
```

Figura 6 Ejemplo de consulta SPARQL

Con consultas similares a la anterior, tendremos la posibilidad de recuperar información de Wikidata a través de la biblioteca SPARQLWrapper, que lanza las búsquedas deseadas sobre el servicio de consulta deseado.

Una vez revisadas las distintas técnicas de clustering llegamos a la conclusión de que las más adecuadas a aplicar en nuestro caso son DBSCAN y k-medoids, ya que tal y como comentamos anteriormente, son las que mejor se adaptan a nuestra propuesta de formar comunidades en base a medidas de similitud. Por otro lado, herramientas como Wikidata Query Service o SPARQL serán de gran utilidad para obtener información acerca de los cuadros de nuestro caso de estudio.

A continuación, veremos cómo aplicar las técnicas y herramientas estudiadas en este capítulo a la hora de modelar grupos de usuarios. El siguiente capítulo detallará el proceso de detección y visualización de comunidades.

Capítulo 3 - Modelado de comunidades

El proceso de modelado de comunidades es el núcleo de este trabajo. En este capítulo repasaremos con detalle la propuesta que hemos desarrollado a lo largo del proyecto, profundizando en cada una de las herramientas que hemos diseñado y como, juntas, componen una aplicación final para poder detectarlas y visualizarlas.

El objetivo es encontrar ciertas características de los usuarios que nos permitan diferenciar grupos lo suficientemente heterogéneos entre sí como para poder obtener información relevante acerca de los individuos o de sus gustos. Tal y como se ha comentado en el Frontend para llevar a cabo esos grupos o clusters de usuarios, se han utilizado diversas técnicas de agrupamiento o clustering. Sin embargo, antes de poder aplicar algoritmos de clustering debemos definir la similitud entre dos personas. Se considera similitud a la medida que proporciona información de cuanto de parecidos o diferentes son dos usuarios. En este aspecto, decidimos definir inicialmente una medida de similitud entre las obras de arte, con el objetivo de poder utilizarla así a la hora de comparar usuarios en el proceso de agrupamiento. Basando la similitud de usuarios en la similitud de las obras que han valorado, conseguiremos formar clusters de usuarios con gustos artísticos comunes, dejando de lado en este caso las consideraciones demográficas.

3.1 Caso de estudio

Para llevar a cabo el modelado de comunidades es necesario el uso de datos tal y como se comenta en la sección anterior. En este caso se han usado los datos proporcionados por los directores del trabajo al inicio del proyecto. Antes de profundizar en el proceso de modelado de comunidades es importante que se haga una breve descripción de los distintos datos con los que se ha contado y su origen. Para el desarrollo del trabajo se ha contado con tres conjuntos de datos:

- Datos demográficos de los usuarios: este conjunto de datos reúne la información de un grupo de usuarios del Museo del Prado. El número de usuarios es de 171 y de ellos se conoce la edad, género y país. Aunque el origen de los datos es ficticio, estos están generados en base a datos reales procedentes del Museo del Prado.

- Datos de obras de arte: en este conjunto se puede encontrar cierta información sobre 30 obras de arte del Museo del prado. Cuenta con una serie de enlaces para recuperar cierta información acerca de las obras de arte, el título de la obra, autor, movimiento artístico y año en el que fue pintado. La mayoría de estos cuadros han sido extraídos de WikiArt.org.
- Datos de las emociones de los usuarios: por último, estos datos se componen por una serie de emociones que han provocado las obras de arte a los usuarios. Se cuenta con un total de 1759 emociones que de igual manera han sido extraídas de WikiArt.org.

Una vez conocidos los datos de partida, se procede a la visualización y limpieza de datos. El objetivo principal es hacerse una idea general de los datos con los que se cuenta y comprobar que los datos son válidos. Por ejemplo, en el caso de los enlaces se comprueba que todos contengan una URL valida. También se hace un pequeño análisis estadístico descriptivo para ver la naturaleza de los datos y una normalización de estos en los casos necesarios.

3.2 Definición de medidas de similitud

Para poder realizar grupos o clusters de usuarios primero se debe definir cuál será la medida de similitud a tener en cuenta. Lógicamente, para definir la similitud entre dos usuarios se puede hacer usos de multitud de características de estos.

En un principio se decidió usar como medidas de similitud los datos demográficos de los usuarios. Durante este proceso se utilizaron las tres variables: edad, género y nacionalidad. Todas estas variables fueron normalizadas para que pudieran ser usadas como distancia entre usuarios. Sin embargo, los resultados obtenidos tras aplicar diferentes algoritmos de clustering carecían de sentido ya que no existía ningún patrón por el que caracterizar a los grupos de usuarios.

Como respuesta al anterior resultado se decide que la medida de similitud entre dos usuarios se hará en base a los sentimientos que les han provocado ciertas obras de arte. La idea principal, es basarse en esos sentimientos generados por los cuadros, ya sean positivos o negativos para generar un grupo de usuarios con gustos similares.

Para llevar a cabo esta idea se han definido dos medidas de similitud principales: por un lado, la similitud entre cuadros y por otro, la similitud entre personas. El objetivo de la similitud entre cuadros es poder concretar qué cuadros son más similares entre sí. De esta forma podemos

crear una bolsa de cuadros similares bajo ciertos criterios. Una vez concretada la similitud entre cuadros, se utilizan los sentimientos de los usuarios para, en un primer lugar, saber que cuadros han valorado positiva y negativamente y, en un segundo lugar, utilizar la medida de similitud entre cuadros para crear una bolsa de cuadros que son del agrado o desagrado del usuario. Con todo esto ya podemos concretar una medida de similitud válida para los usuarios.

A continuación, se explica de manera detallada cada una de las similitudes principales definidas para la detección de comunidades de usuarios.

3.2.1 Similitud de usuarios

Como comentábamos al principio del capítulo, necesitamos poder definir una medida de similitud entre los distintos usuarios para poder agruparlos en comunidades, ya que las diferentes técnicas de clustering que utilizaremos para ello requieren conocer cuan cerca están los unos de los otros a la hora de definir estos grupos. Para ello hemos de considerar las valoraciones y opiniones que cada persona ha proporcionado sobre los cuadros. Con esta idea hemos definido una medida de similitud entre las obras tal que la podamos utilizar a la hora de poder comparar dos conjuntos de cuadros que dos usuarios han valorado de cierta manera.

Los usuarios han comentado sobre una serie de obras la emoción que esta les causa o inspira. A su vez estas emociones han sido etiquetadas con tres polaridades, siendo estas positive, negative y mixed con el objetivo de poder obtener conjuntos de cuadros que les gusten, no les gusten o les resulten indiferentes. A la hora de enfrentar dos usuarios para obtener su índice de similitud, es útil tener conjuntos de cuadros los más amplios posibles para obtener medidas más significativas, y es por ello por lo que a la hora de compararlos consideramos los conjuntos de cuadros que han valorado con cierta polaridad, dejando de lado las emociones concretas, con las que solo obtendríamos conjuntos pequeños y en muchos casos vacíos.

Una vez hemos decidido que llevaremos a cabo las comparaciones de usuarios enfrentando conjuntos de obras que han valorado con cierta polaridad, hemos de definir un método que, haciendo uso de las medidas de similitud entre cuadros que definiremos posteriormente, devuelva el índice de similitud entre ambos conjuntos de obras.

Una primera aproximación que tomamos fue considerar la similitud media de los cuadros de ambos conjuntos. La idea era tomar las obras de ambos de dos en dos y obtener así todas las similitudes de cada posible pareja, para después poder calcular la media. El problema que presentaba este método es que considerar la media centraba en exceso los resultados, obteniendo gran parte de los usuarios índices de similitud muy cercanos.

Para conseguir una medida algo más significativa y discriminante a la hora de comparar conjuntos definimos el método que describimos a continuación:

La idea principal es extender los conjuntos de obras valoradas en cierta polaridad iniciales de cada usuario (Figura 7). Para llevarlo a cabo usamos la medida de similitud de cuadros para recuperar los k (parametrizable) cuadros más similares a cada uno de los presentes en cada conjunto. Otra opción sería recuperar para cada cuadro aquellos que superen un cierto *umbral* (parametrizable) de similitud.

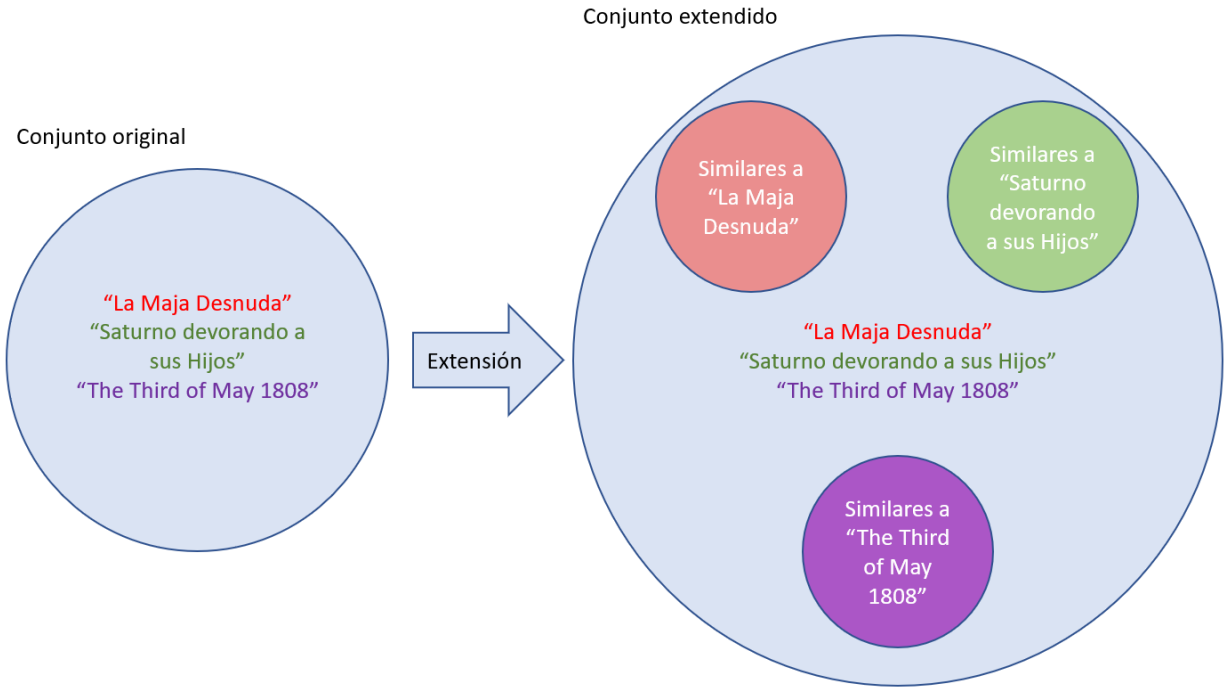

Figura 7 Extensión de los conjuntos de obras valoradas por usuarios con obras similares

Una vez hemos extendido ambos conjuntos de obras, podemos aplicar alguna medida de similitud entre los dos. En este caso aplicamos el coeficiente de similitud entre conjuntos de Jaccard que se calcula de la siguiente manera: Sean A y B dos conjuntos, definimos el coeficiente

de similitud de Jaccard como el cociente entre el cardinal de la intersección y el cardinal de la unión de A y B.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

El coeficiente de similitud de Jaccard nos da una idea de en qué medida dos conjuntos de solapan. Llevado a nuestro caso, es una medida muy indicativa de cuan parecidos son las valoraciones de dos usuarios. Al haber recuperado los cuadros más similares con respecto al conjunto original, dos usuarios con gustos parecidos tendrán un índice de solapamiento muy alto.

Sin embargo, dos usuarios más disonantes, obtendrán un porcentaje de solapamiento mucho menor ya que si sus cuadros no son similares, con la extensión de los conjuntos, no obtendrán tampoco demasiada coincidencia.

3.2.2 Similitud de cuadros

El proceso de detección de comunidades de usuarios hace uso de una medida de similitud entre cuadros previa. El objetivo de esta similitud es poder concretar cuanto de similares son los cuadros entre sí en base unas variables concretas. Para definir la medida de similitud entre dos cuadros se utilizan diversas similitudes parciales de manera ponderada. Es decir, adjudicando diversos pesos a cada una de las similitudes parciales. Para llevarlo a cabo, se construye una matriz de distancias entre los cuadros siendo 1 la similitud más alta y 0 la más baja entre un par de cuadros dados. Para ello, se han definido cinco medidas de similitud parcial. Con esta técnica se enriquece de manera sustancial la similitud entre cuadros. Sin embargo, surge una gran cuestión una vez concretadas y definidas las similitudes parciales de los cuadros. ¿Qué peso deben tener cada una de las variables al comparar dos cuadros? Para responder esta pregunta debemos entender que repercusión tiene el uso de los pesos.

Figura 8 Los fusilamientos del tres de mayo

Figura 9 La maja desnuda

Si por ejemplo queremos comparar los cuadros de la Figura 9 y la Figura 8 en base al autor exclusivamente, la variable autor tendrá el peso total de la similitud. Es decir, los posibles resultados que obtendremos serán la completa similitud (100%) en caso de coincidencia de autor o el 0% en caso contrario. Para este caso, se obtendría un 100% de similitud ya que ambos cuadros pertenecer a Francisco Goya. Sin embargo, si ahora tomamos dos atributos de los cuadros como autor y color dominante, y cada uno tiene un peso de 40% y 60% respectivamente veremos que los índices de similitud varían. En este caso, si el autor coincide, pero el color predominante no, obtendremos una similitud de un 40% entre esa pareja de cuadros. Como se puede observar, la repercusión a la hora de elegir el peso de cada una de las similitudes parciales es de vital importancia a la hora de comparar dos cuadros.

Cabe destacar que los distintos pesos que se apliquen a las similitudes parciales no llevan consigo una connotación ni positiva ni negativa. Gracias a este planteamiento, se pueden construir distintas medidas de similitud entre cuadros permitiendo generar comunidades de usuarios bajo ciertos criterios. Si por ejemplo quisiéramos detectar comunidades de usuarios en base al gusto por un autor exclusivamente, sería tan fácil como ajustar el peso al 100% de la variable autor. De esta manera se conseguiría definir comunidades que tienen como característica común su gusto por un autor en concreto.

En los siguientes subapartados se explica de manera detallada cómo se han utilizado los siguientes atributos de los cuadros como medidas de similitud parcial.

- Similitud por contenido
- Similitud por color dominante
- Similitud por tamaño
- Similitud por artista y movimiento
- Similitud por error cuadrático medio por píxel

3.2.2.1 Similitud por contenido

El contenido de un cuadro se define por los elementos que están representados en él, que llamaremos depicts. Podemos observar, por ejemplo, los depicts de la obra La rendición de Breda de Velázquez en la Figura 10. Se trata de una variable muy descriptiva y representativa de la temática del mismo, por lo que una comparativa entre ellos puede suponer una buena medida de similitud entre dos obras. Además, acceder a esta información es muy sencillo a través de Wikidata ya que se trata de una de las principales propiedades de las obras (P180).

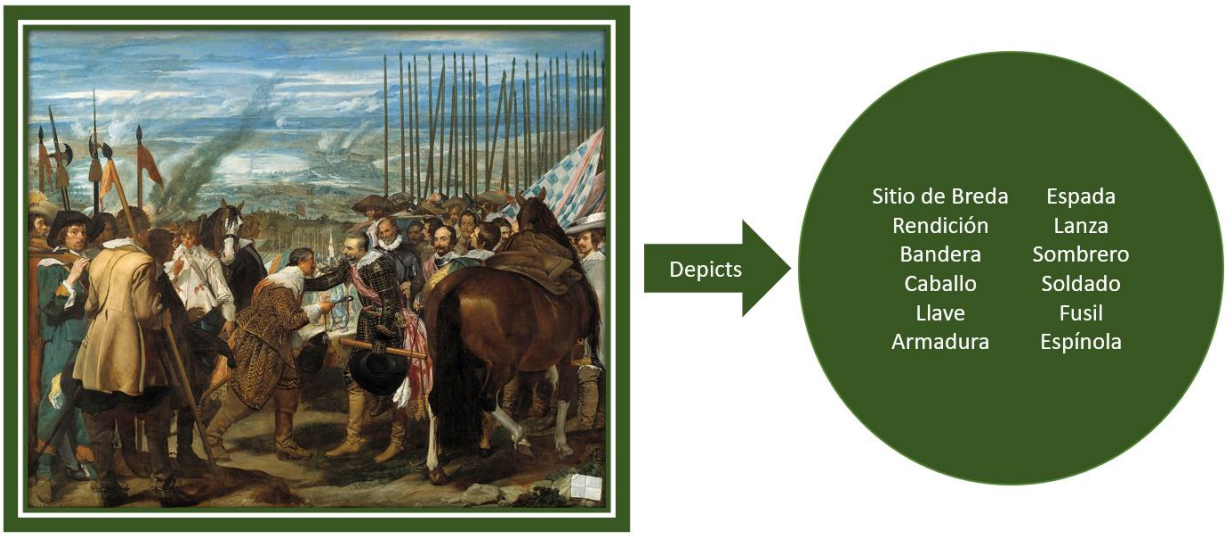

Figura 10 Depicts de la obra La Rendición de Breda de Velázquez

Para definir una medida de similitud entre dos conjuntos de conceptos, como son los representados en un cuadro, no debemos limitarnos a la coincidencia directa de términos ya que en la mayoría de los casos ésta será muy limitada y estaremos perdiendo información. Por ejemplo, si en dos cuadros podemos encontrar un fondo boscoso, pero en un caso se trata de un pinar y en el otro se trata de un encinar. No podemos desechar esa información porque simplemente solo consideremos la coincidencia directa de los conceptos pino y encina, deberíamos poder abstraernos hasta el concepto de árbol.

Es por ello por lo que la medida de similitud que hemos definido para los conjuntos de depicts de dos cuadros tiene en cuenta las superclases de cada concepto, en cierta profundidad. En este caso, hemos recuperado las propiedades "instance of" (P31 en wikidata) y "subclass of"

(P279 en wikidata) para utilizarlas como superclases, ya que representan los conceptos en un nivel abstracción mayor, conservando, sin embargo, la semántica de los mismos.

Debemos considerar también la profundidad en la que recuperaremos estos conceptos, es decir, cuanto vamos a escalar en la jerarquía de superclases de dos pares de conceptos a la hora de buscar coincidencias. También es importante que la medida de similitud entre dichos pares de conceptos tenga en cuenta en qué profundidad se ha hallado un término coincidente, ya que cuanto más subimos en el árbol de superclases, más generales serán los conceptos que recuperemos y nos exponemos a una probable pérdida de semántica si nos excedemos al subir en la jerarquía. Volviendo al ejemplo anterior (Figura 11), los conceptos pino y encina, coincidirán con total seguridad en el primer nivel ya que ambos son subclases de árboles. Sin embargo, considerando de nuevo el concepto del pino y en este caso el de un animal como puede ser un caballo, aunque no coincidan en primera instancia, lo acabarán haciendo eventualmente al ser, ambos, subclases de seres vivos. Teniendo esto en cuenta, debemos aplicar en nuestra medida de similitud algún mecanismo que pese las medidas en función de la altura en que los conceptos coincidan.

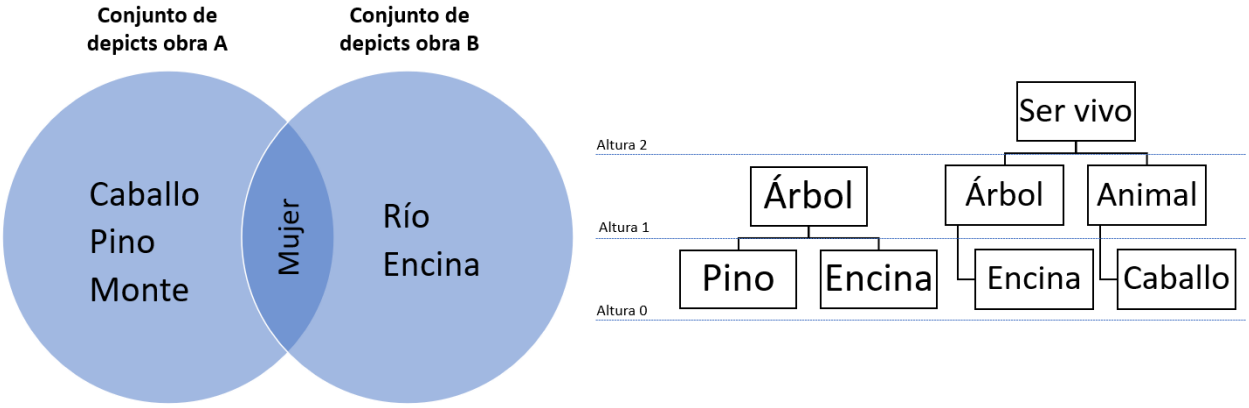

Figura 11 Superclases comunes entre conceptos de dos conjuntos de depicts

Con todo, planteamos la medida de similitud como la similitud del coseno entre dos vectores con pesos comprendidos en el rango [0,1] que construimos con el siguiente formato:

Sea A conjunto de depicts del primer cuadro y B el conjunto de depicts del segundo cuadro. Consideremos los siguientes conjuntos para tratar por separado los conceptos presentes en ambos cuadros y los que en cambio son exclusivos de cada uno:

1. Intersección: definida como $A \cap B$

2. Exclusivos de A*:* definido como A \ (A ∩ B)

3. Exclusivos de B*:* definido como B \ (A ∩ B)

4. Comunes de A y B*:* conjunto de antecesores comunes de los conceptos de Exclusivos de A y Exclusivos de B tomados de dos en dos

5. Descartes*:* conjunto de conceptos de A y B para los que se han encontrado antecesores comunes y se han incluido en el conjunto anterior

Considerando los anteriores conjuntos, y, tal y como podemos observar en la Figura 12, construimos los vectores insertando para cada concepto de los siguientes conjuntos los siguientes pesos:

- Para todos los conceptos del conjunto Intersección insertamos el valor máximo 1 en ambos vectores, indicando la presencia de cada concepto en ambas obras.
- Para todos los conceptos del conjunto Exclusivos de A, exceptuando los incluidos en el conjunto Descartes, insertamos el valor máximo 1 en el vector de pesos de A y el valor mínimo 0 en el vector de pesos de B
- Para todos los conceptos del conjunto Exclusivos de B, exceptuando los incluidos en el conjunto Descartes, insertamos el valor máximo 1 en el vector de pesos de B y el valor mínimo 0 en el vector de pesos de A
- Para todos los conceptos del conjunto Comunes, insertamos en cada vector el valor inverso a la profundidad en que coincidieron los conceptos en cada caso.

Conjuntos	Intersección	Exclusivos A \ Descartes	Exclusivos B \ Descartes	Comunes	
Conceptos	Mujer	Monte	Río	Árbol	Ser vivo
Vector A	1	1	0	1	0.5
Vector B	1	0	1	1	0.5

Figura 12 Construcción de los vectores de pesos de los conjuntos de depicts

Una vez construidos sendos vectores de pesos, definimos la similitud por contenido como la similitud del coseno entre ambos vectores, cuyo valor viene dado por la siguiente formula:

$$sim(a,b) = \frac{A \cdot B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\,\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Con este método conseguimos no solo tener en cuenta significados más abstractos de los conceptos de cada conjunto de depicts, si no poder pesar su aportación al índice de similitud en función de la altura en la que encontremos las coincidencias de conceptos.

Un aspecto importante a tener en cuenta a la hora de llevar a cabo este proceso es la profundidad máxima en la jerarquía conceptual, es decir, la profundidad en que se buscan las coincidencias de los conceptos. El incremento de este valor puede conllevar que el número de coincidencias aumente considerablemente al abstraerse demasiado del concepto original, lo cual puede conllevar una perdida excesiva de semántica además de tener un gran coste computacional. Sin embargo, un valor muy bajo podría estar privándonos de encontrar coincidencias de mayor relevancia. Un valor con un coste asumible y que no peca de excesiva abstracción semántica es 3.

3.2.2.2 Similitud por color dominante

El color en un cuadro es otro de los atributos que definen la similitud entre cuadros, que, dependiendo del caso de prueba y el criterio del individuo, es más o menos importante. Con esto damos por hecho que el color es una propiedad imprescindible para definir la similitud que hay entre dos obras.

Dadas las infinitas tonalidades de colores, resulta imposible hacer una lista de los mismos que aparecen en un cuadro. Por ello, hemos concluido que la mejor forma de abordar este tema es obtener una paleta de los colores dominantes de cada obra de arte, partiendo de la imagen de la misma representada en el modelo RGB. Esto se consigue recabando una media de colores por píxel que contiene.

Para llegar a la información necesaria que nos permitirá realizar esta distinción y permitir comparar un cuadro con otros utilizamos k-means, de manera que el algoritmo, segmenta la imagen con k precisión. Esta solución permite que al establecer los k clusters que se elijan, a partir de una primera pasada agrupar ciertos colores inicialmente, de forma que con cada iteración los conjuntos se van moviendo y atrayendo las tonalidades que más tienden a la media, resultando finalmente en un resultado de los k colores principales medios que contiene la imagen. La paleta de colores que se logra puede ser variable dependiendo de la precisión de tonalidad que queremos obtener, siendo más precisa cuanto mayor índice k le demos al algoritmo, permitiendo de esta manera una mayor gama de colores, a costa de reducir finalmente el abanico de posibilidades similares en este aspecto, dada la comparación con un tono más específico. Esta decisión también implica que utilizar como color dominante el color medio general de una obra tampoco es suficientemente precisa, puesto que la combinación de muchos colores suele discurrir hacia un color generalmente oscuro y no muy representativo.

En la siguiente imagen (Figura 13), podemos ver como partiendo de una imagen original, el algoritmo la segmenta en los k colores, tres en este caso, más dominantes que contiene. De esta forma, se generan tres tonos principales que más se asemejan a la mayoría de colores cercanos a esa tonalidad.

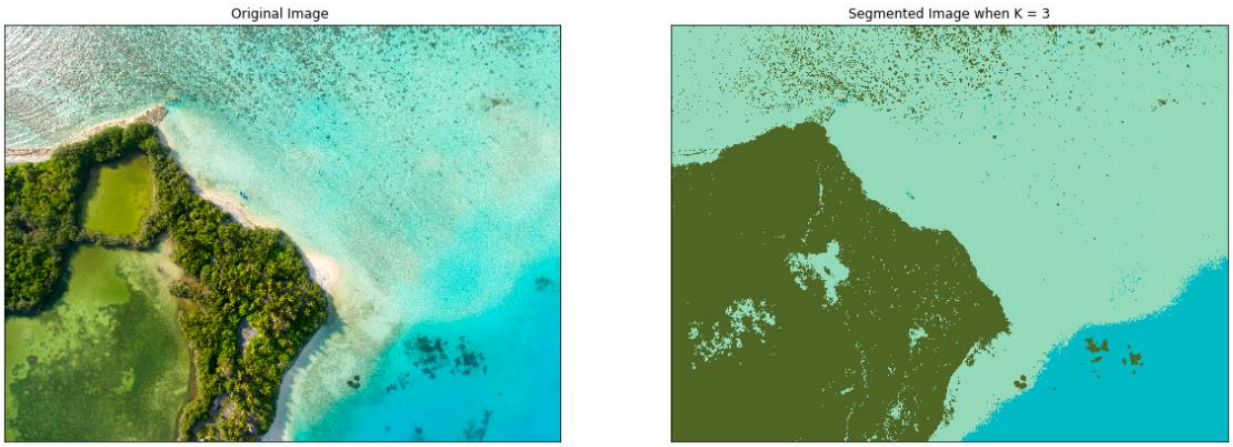

Figura 13 Representación de una imagen segmentada con k-means

A la pregunta ¿y cómo medimos de manera precisa cuán parecido es un cuadro a otro cuando se trata con este atributo?, la respuesta implica utilizar el modelo de color HSV.

HSV (Hue – Saturation – Value, o Matiz – Saturación – Valor en español) o HSB, que sustituye el Valor por Brightness (Brillo) está representado en un diagrama circular (Figura 14), estos valores se suelen comprender entre estas magnitudes:

- H (Matiz): Comprende toda la gama cromática y se representa mediante la posición en un círculo completo, es decir, en un rango de 0 a 360º. Sabemos que los colores básicos se localizan de la siguiente forma: 0º/360º coincide con el color rojo (1,0,0), 120º es el verde (0,1,0) y finalmente el azul se corresponde con 240º (0,0,1).
- S *(*Saturación): También denominado pureza, este parámetro representa la cantidad de color. Con un rango comprendido entre 0 y 100%, cuanto mayor es la cantidad, más coloración hay.
- V (Valor): Este atributo representa la luminosidad, definido por la altura en el eje blanco-negro. Sus valores, al igual que la saturación van del 0 al 100%, siendo 0 el negro y 100 el blanco.

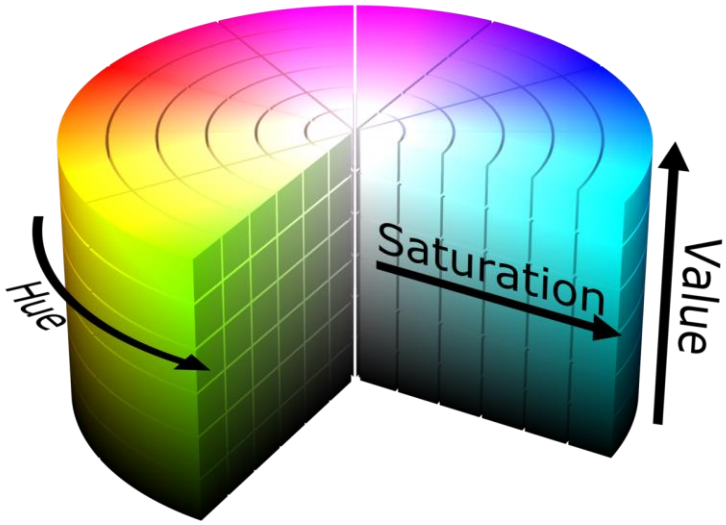

Figura 14 Representación de HSV

Esta medida HSV la obtenemos mediante la transformación desde el frecuente modelo RGB, ya que, si lo utilizamos directamente, es bastante probable que comparando los valores RGB de dos cuadros aparentemente muy diferentes, se encuentre una similitud entre colores, ya que únicamente teniendo en cuenta el círculo cromático habitual, se podrían clasificar en los mismos rangos de color, colores muy dispares entre sí realmente, por tanto necesitamos un método más preciso para medir la similitud entre dos colores.

Una vez decidido este plan de acción y obtenidas las medidas transformadas a este modelo procedemos a comparar el color dominante entre los cuadros. Los valores HSV transformados están normalizados en un rango del 0 al 1, ya que al estar expresados inicialmente en porcentajes y en grados, así obtenemos unos resultados más comprensibles a primera vista. Tras obtener estos datos en los cuadros, obtenemos la similitud de las obras dos a dos calculando la distancia euclídea entre los valores HSV de las mismas.

3.2.2.3 Similitud por tamaño

El tamaño o dimensiones de una obra de arte siempre ha sido uno de sus aspectos más llamativos. Existen claros ejemplos de ello, como la Batalla de Gettysburg del pintor Paul Philippoteaux con un área superior a los 2500m^2 o, por el contrario, Carlos Ortiz de Taranco de Federico Madrazo y Kuntz situado en el Museo del Prado y que apenas llega a los 90cm^2. Ambas obras de arte ponen de manifiesto que las dimensiones juegan un papel importante a la hora de provocar una reacción en el espectador. Por esta razón se ha utilizado la dimensión o tamaño de la obra de arte como medida de similitud parcial.

Al igual que con los depicts mencionados en el apartado Similitud por contenido, el acceso a al ancho y largo de las obras de arte se hace a través de Wikidata ya que su acceso resulta rápido y cómodo. Para llevar a cabo la similitud se recuperan dos propiedades de los cuadros: el ancho (P2049) y la altura (P2048). Una vez recuperados el ancho y altura de ambos cuadros se calcula el área de los dos cuadros. Por último, y con el objetivo de obtener un resultado simétrico, calculamos la razón dividiendo el área de mayor tamaño entre el área de menor tamaño. Este resultado nos permite medir a través de la razón entre áreas como de semejantes son dos cuadros.

En definitiva, con esta medida de similitud basada en el tamaño de las obras de arte aportamos mayor riqueza a la similitud global, ya que, tal y como dice Silvia Puig Pages en su

trabajo Relación corporal en la obra escultórica tamaño y proporción como elemento implicado en la percepción tridimensional: “las medidas de los objetos no es algo puramente métrico y neutro, sino que hay una manera de ser, un "carácter", una "personalidad", una tendencia psicológica, etc.., para cada tamaño” [11].

3.2.2.4 Similitud por artista y movimiento

Si pensamos en obras de arte inevitablemente pensamos en el artista que la creó, ya sea un autor reconocido o desconocido. La importancia del autor no solo sobre las obras que compone sino sobre los espectadores que las visualizan es sin duda uno de los aspectos más destacables a la hora de comparar dos cuadros. Tal y como se menciona en el artículo Los visitantes del Museo del Prado: nueva metodología de medición del turismo cultural de Laura Pena Alberdi y Celia Sánchez Vicente, el motivo principal de la visita al Museo del Prado se caracteriza por las obras permanentes ya sea por el interés de una obra u artista en concreto [12]. Esto nos indica la relación existente entre los gustos de los usuarios y las obras de un mismo autor. Esta relación puede darse por dos motivos: el gusto por las obras en sí mismas de un artista en concreto o bien por el sentimiento del espectador hacia el artista en particular tal y como dice Dr. Juan García Villar en su escrito “El artista, la pintura y el espectador”, “los espectadores…además saben interpretar y traducir ésta como una gramática pictórica, y, en definitiva, como la resonancia de los sentimientos y las reacciones reflejas del autor” [13].

Si se visualizan las distintas obras de un mismo autor, son muchos los factores por los que se podrían caracterizar las pinturas y relacionarlas con el autor. Los colores, texturas, técnicas, temáticas y patrones son algunos de los rasgos propios de un autor que se ven plasmados en sus obras. Por ejemplo, en las pinturas de El Greco (Doménikos Theotokópoulos) se pone de manifiesto la relación entre las pinturas y el autor. El estilo se caracteriza por figuras manieristas muy alargadas, con iluminación propia, muy expresivas, en ambientes indefinidos y usando los contrastes con los colores [14].

Como se ha podido comprobar, el autor del cuadro juega un papel fundamental a la hora de comparar dos obras de arte. Sin embargo, todos o casi todos los cuadros pertenecen al menos a un movimiento artístico. Son muchos los ejemplos de obras pertenecientes a un mismo

movimiento artístico. Al igual que la relación existente entre autor y obra también existe diferentes factores por los que caracterizar una obra y enmarcarla en un movimiento artístico concreto. Si hablamos del Impresionismo, hablamos de pintura al aire libre, la luz y hablamos, por ejemplo, de Claude Monet y Eduard Manet [15].

Teniendo en cuenta la influencia del autor y movimiento artístico a la hora de comparar dos cuadros se ha decidido generar una medida de similitud parcial que tenga en cuenta estos datos de los cuadros. Para recuperar esta información se ha utilizado Wikidata, al igual que en otras similitudes parciales. En concreto, se han utilizado las propiedades de los cuadros creador (P170) y movimiento artístico (P135).

Una vez recuperados el autor y el movimiento de ambos cuadros, aplicamos el siguiente calculo: en primer lugar, se comprueba si ambos autores coinciden. En caso afirmativo el índice de similitud para ese par de cuadros es de 1. En caso contrario, se comprueba si los cuadros pertenecen al mismo movimiento artístico. Si coincide el valor de la similitud será de un 0,85. Por lo tanto, la similitud entre dos cuadros según autor y movimiento podrá ser de: 0, 0,85 o 1. La decisión de fusionar autor y movimiento en una misma similitud viene dada por la relación entre autor, obra y movimiento. La relación entre obra, autor y movimiento tiene el suficiente peso como para llevar a cabo la fusión de ambas. Cabe destacar que, en caso de coincidencia por movimiento artístico se valora la similitud existente con un 0,85, ya que de esta forma se puede valorar la similitud a pesar de la no coincidencia de autor sin llegar al máximo de la similitud. Esto permite relacionar ambas variables y tratarlas como una.

3.2.2.5 Similitud por error cuadrático medio por píxel

Aparentemente aplicar el uso del error cuadrático medio (MSE) por píxel carece de sentido a la hora de comparar dos obras de arte, ya que simplemente aporta una medida de análisis estadístico. Sin embargo, son muchos los estudios, artículos y tecnologías que usan esta medida a la hora de comparar dos imágenes. Pero ¿qué es el error cuadrático medio? Para responder esta pregunta debemos trasladarnos al mundo de la estadística, en concreto a la inferencia estadística.

Imaginemos que tenemos una población de individuos no vacía. De cada individuo se sabe su peso (Kg) y altura (cm). Ahora imaginemos que queremos hacer una estimación en base al peso. El estimador predice ciertos valores para la altura. Una forma de medir la calidad de nuestro estimador es a través del error cuadrático medio. Este medidor calcula la diferencia entre los valores predichos por el estimador y el valor del parámetro. Para ello, se calcula la media de los cuadrados de las diferencias entre los valores que se estiman para una variable, en este caso la altura y el valor del parámetro [16].

Si extrapolamos este concepto al dominio de nuestro problema, el uso del error cuadrático medio nos proporciona un mecanismo para comparar dos imágenes. La idea principal es comparar el valor de cada píxel de una imagen con el valor del píxel correspondiente de otra imagen. Para ello, la imagen es transformada a un formato de punto flotante.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Sin embargo, esta medida que proporciona el MSE puede llegar a dar problemas ya que como se puede observar la diferencia esta al cuadrado. Esto supone que los errores grandes serán errores muy grandes. Para mitigar este resultado aplicamos una extensión de este cálculo, la raíz del error cuadrático medio (RMSE). Esta extensión aplica la raíz cuadrada al MSE. De esta forma la media no se ve tan afectada por los errores de mayor magnitud.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Por lo tanto, el RMSE nos permite calcular la diferencia entre los valores de cada píxel dadas dos imágenes. De nuevo surge otro problema. Las imágenes no tienen la misma dimensión.

Es por esto por lo que antes de aplicar el RMSE se redimensionan las imágenes al tamaño de la mayor de ellas tal y como se ilustra en la Figura 15. Por último, se calcula el RMSE por píxel permitiéndonos conocer como de diferentes son dos imágenes píxel a píxel.

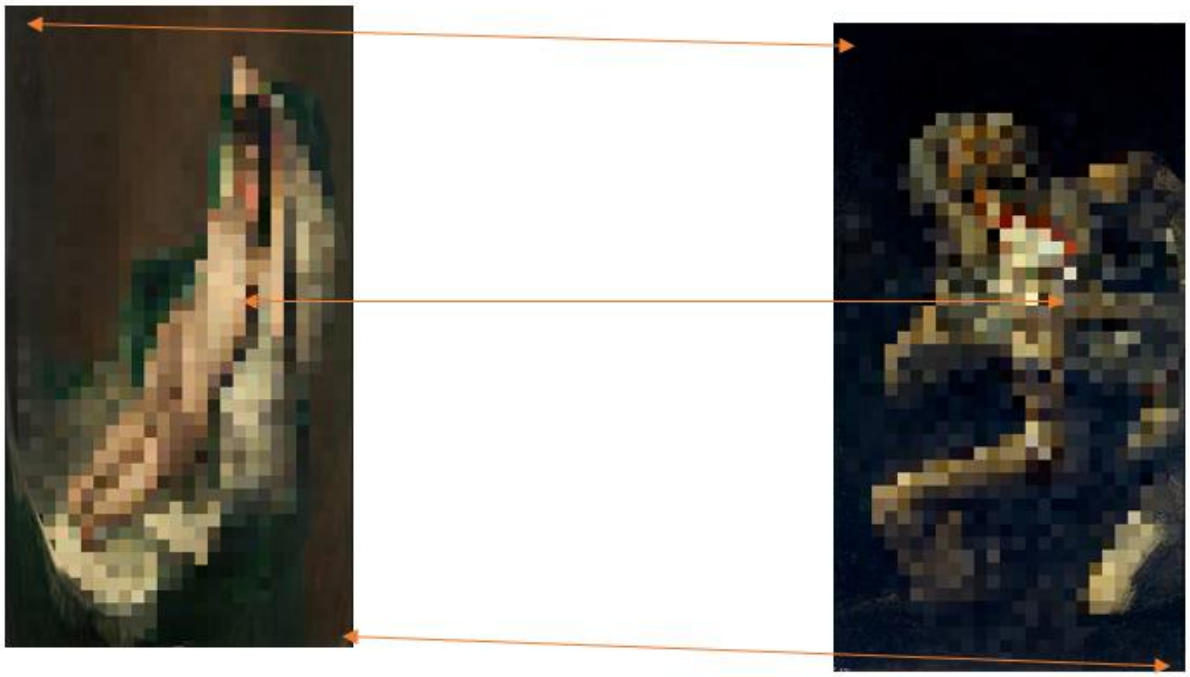

Figura 15 Calculo del MSE entre las imágenes de dos obras reescaladas

Esta medida de similitud parcial, al contrario que la similitud basada en el color dominante en el cuadro aporta una comparación entre imágenes objetiva y rigurosa en cuanto valor de los pixeles. Esto es así, ya que, mientras en la similitud basada en el color predominante se compara la media del valor de los pixeles, llegando incluso a comparar colores que no están presentes en el cuadro, con el RMSE la comparación de las imágenes a nivel de píxel es más objetiva.

3.3 Proceso de modelado

Las medidas de similitud que hemos definido nos sirven para poder medir las distancias entre usuarios, sin embargo, nos presentan una serie de cuestiones que hemos de resolver antes de aplicar alguna técnica de clustering sobre nuestros usuarios.

En primer lugar, hemos considerado nuestra medida de similitud entre obras como la unión de una serie de similitudes parciales, pero no hemos decidido que peso tendrá cada una de ellas en la similitud total. Definir que similitudes parciales tendrán más o menos peso tendrá un fuerte impacto en el resultado final del modelado de comunidades. Si, por ejemplo, damos mucho peso a la similitud por artista, acabaremos obteniendo grupos de admiradores de artistas concretos. Si, sin embargo, decidimos dar mayor peso a la similitud por depicts, observaremos que los usuarios se agruparan más en función de una temática común. Nuestra idea es que la decisión de los pesos quede como un parámetro más de nuestro proceso de modelado, por eso hemos incorporado al proyecto una interfaz para poder visualizar el efecto de dichos pesos a la hora de comparar distintas obras.

En segundo lugar, a la hora de comparar dos usuarios, hemos de decidir como recuperar los cuadros más similares a uno dado cuando queremos extender sus bolsas de cuadros. Como veíamos anteriormente, tenemos las posibilidades de fijar un número k de obras similares a ser recuperadas o de fijar un umbral de similitud para recuperar solo las obras que los superen para otra obra dada. Fijar un parámetro *k* de obras a recuperar nos permite tener una gran selección de cuadros similares para cada usuario, lo que nos da un mayor margen a la hora de compararlos, sin embargo, esta propuesta puede pecar de permisiva ya que se puede dar el caso de que estar recuperando obras con bajo índices de similitud entre ellas simplemente porque está entre los k más similares. A la hora de utilizar un umbral, podemos enfrentar la problemática de si este es demasiado alto, obtengamos un bajo número de obras y las similitudes de usuarios sean muy bajas, sin embargo, al igual que antes, si le otorgamos un valor muy bajo, podremos estar pecando de excesiva permisividad.

3.3.1 Búsqueda de pesos

Como hemos discutido en el apartado anterior, los pesos con los que apliquemos las distintas similitudes parciales entre obras tendrán un fuerte impacto en el resultado final de las comunidades. Estos podrían ser configurados a mano si se buscan cumplir requisitos simples como los expuestos anteriormente, pero si sin embargo se desean cumplir requisitos más complejos como puedan ser la búsqueda de comunidades lo más compactas posibles, comunidades de similar

tamaño, comunidades que cumplan alguna restricción demográfica o cualquier otro objetivo, hemos de proporcionar algún mecanismo capaz de satisfacer estas restricciones.

Por ellos hemos implementado un algoritmo genético cuya función es poder utilizarlo para buscar los pesos más adecuados para cumplir un determinado objetivo. Utilizando el vector de pesos como el cromosoma de nuestro algoritmo e implementando una función de fitness adecuada, consiguiendo que premie los individuos que más se ajusten a las restricciones y que castigue a los individuos que más se alejen, podremos obtener el vector de pesos más adecuado para nuestro modelo de comunidades.

Este algoritmo genético (Figura 16) sigue una estructura clásica de iterar a lo largo de múltiples generaciones procediendo al proceso de selección, el cruce de un cierto porcentaje de individuos, la mutación de parte de los alelos de cromosomas de la población y la reinserción de los mejores individuos de la generación anterior (elitismo). Esta diseñado como un algoritmo de minimización, por lo que la función de fitness a utilizar debe tenerlo en cuenta a la hora de evaluar cada cromosoma. Los cromosomas están implementados como vectores de números reales cuyo rango se define inicialmente para cada uno de los alelos (posiciones del vector) inicialmente. Entrando en detalle en algunos aspectos, se ha implementado el proceso de selección mediante un proceso de ruleta por fitness en el que se reparten las probabilidades en función de las puntuaciones de la función de fitness. Al inicio de cada iteración se aísla una pequeña elite que será reinsertada en la siguiente generación sin sufrir cruce ni mutaciones. Los cruces están implementados de tres formas: cruce de un punto, cruce de dos puntos (se parten los cromosomas por uno o dos puntos respectivamente y se intercambian los alelos de cada partición) y cruce uniforme (que intercambia los alelos de cada cromosoma con un 50% de probabilidad). Por último, la mutación se da según un porcentaje en el que para cada alelo se cambia su valor por uno elegido aleatoriamente dentro de su rango definido.

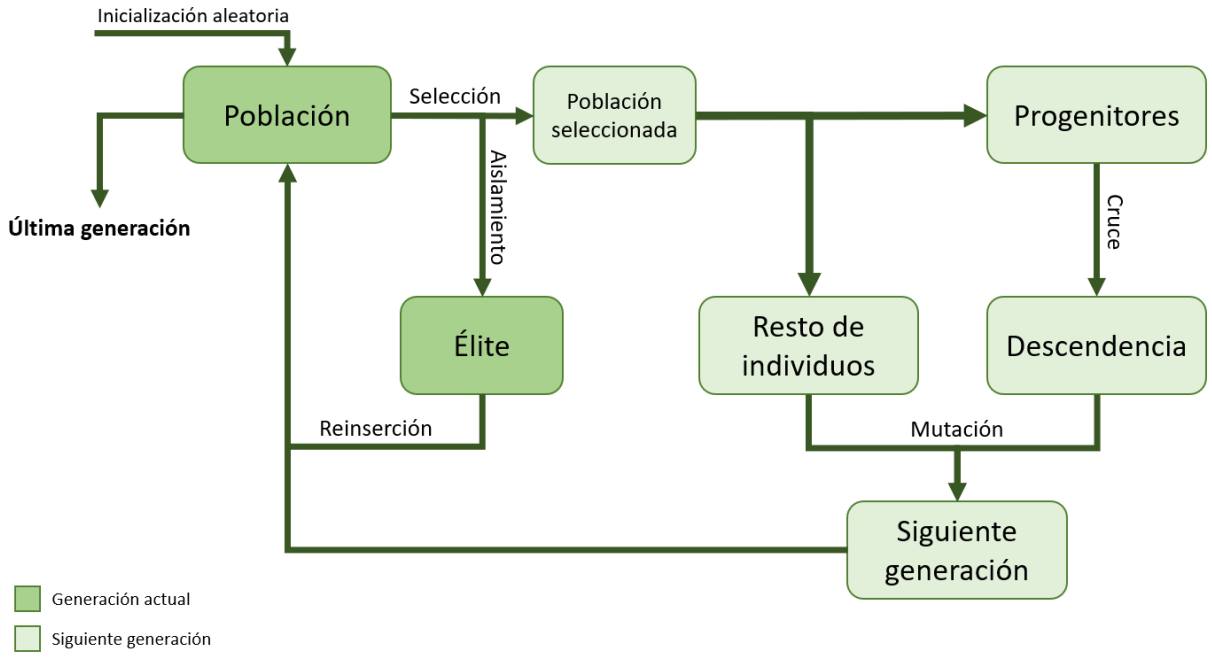

Figura 16 Esquema del algoritmo genético utilizado

Como podemos ver, el proceso de modelado ofrece una cierta flexibilidad en este aspecto, pudiendo generar múltiples tipos de comunidades definidas en función de diferentes criterios.

Con todo esto, decidimos ponerlo a funcionar con una función de fitness propia. En este caso uno de los directores nos proporcionó un conjunto de datos que contiene las respuestas de una serie de profesionales y amateurs en el campo del arte acerca de su percepción similitud de varios pares de obras. Con esta información definimos una función de fitness que midiera el error medio de nuestra similitud de obras pesada con el cromosoma dado con respecto a las respuestas de los profesionales y amateurs. De esta forma conseguiríamos unos pesos que se ajustasen con su percepción de similitud de obras y que podrían servir de punto de partida para modelar una serie de comunidades.

3.3.2 Formación de comunidades

Una vez definidos los parámetros pendientes, podremos empezar a aplicar las ya anticipadas técnicas de clustering. Para poder aplicar las distintas técnicas necesitamos definir la medida distancia entre dos usuarios, que por supuesto estará fundamentada en las medidas de similitud. En este caso decidimos tener en cuenta tanto las opiniones en polaridad positiva como negativa a la hora de comparar usuarios. Una vez obtenidas ambos índices, definimos la similitud

de dos usuarios como la media de ambas y la distancia que los separa como la unidad menos la media obtenida. Así pues, cuanto más similares sean menor será la distancia y cuanto menos se asemejen mayor será esta.

A la hora de formar las comunidades, finalmente hemos tenido en cuenta dos técnicas de clustering. Con sus diferencias, ofrecen resultados diferentes a la hora de tratar con los usuarios y sus similitudes. En el caso de DBSCAN, obtendremos clusters con un alto nivel de cohesión en lo que a gustos artísticos se refiere. Sin embargo, ofrece la desventaja de que en ocasiones puede etiquetar a un gran número de usuarios como ruido, no incluyéndolos en ninguna comunidad. En cambio, k-medoids incluye a todos los usuarios en algún cluster por muy lejanos que sean con respecto al resto de miembros del grupo, algo que puede causar que surjan comunidades más dispersas o con usuarios muy diferentes a los demás.

3.3.3 Visualización de las comunidades

Para poder visualizar las comunidades obtenidas mediante las técnicas de clustering, podemos recurrir a diferentes datos de los usuarios que la forman. En este caso hemos querido considerar, no sólo los gustos artísticos de los usuarios que conforman cada grupo, sino algo más de información que pueda ser de utilidad a la hora de estudiarlos. En esta línea, decidimos incluir para cada cluster formado una serie de datos acerca de la distribución demográfica del mismo, los cuadros y artistas con mayor número de valoraciones positivas y negativas de cada cluster, el movimiento más popular y los sentimientos más expresados.

3.3.4 Discusión de los resultados obtenidos

La flexibilidad que ofrece el mecanismo de modelado que hemos propuesto en este capítulo tiene como consecuencia que se puedan encontrar diversos resultados en función de cómo parametricemos la detección de comunidades. En este sentido, hemos decidido proponer una serie de comunidades que hemos detectado utilizando diferentes pesos para las similitudes de cuadros. En cuanto a la técnica de clustering elegida, consideramos utilizar DBSCAN por su capacidad de descartar individuos ruidosos a la hora de formar las comunidades, y, poder así obtenerlas con una mayor homogeneidad y similitud entre los individuos que las componen.

En la primera propuesta de detección de comunidades, utilizamos unos pesos propuestos por un profesional en arte tras experimentar con la herramienta de visualización de la similitud entre cuadros que desarrollamos. Nos comentó que, en su opinión, las similitudes parciales más determinantes son tanto la Similitud por artista y movimiento como la Similitud por contenido, mientras que, sin embargo, la Similitud por color dominante y la Similitud por tamaño debían ser las que menos influencia tuvieran en la medida final. Así pues, tras varias pruebas con la herramienta de visualización, decidió que los siguientes pesos son los más adecuados:

Similitud por contenido	35%
Similitud por color dominante	5%
Similitud por tamaño	5%
Similitud por artista y movimiento	35%
Similitud por error cuadrático medio por píxel	20%

Tabla 1 Pesos de las similitudes parciales entre cuadros propuestos por un profesional en arte

En cuanto a los resultados de esta propuesta, se obtuvieron 5 comunidades. En general, cada cluster de individuos tiene un movimiento artístico claramente representativo, estando representado cada uno de los presentes en el caso de estudio por al menos una comunidad. En cada una de las comunidades es fácil identificar al menos un cuadro que estuviera valorado positivamente por la mayoría de los usuarios. En contra, no existe tanta consonancia a la hora de encontrar cuadros valorados negativamente por una parte significativa de las comunidades. En cuanto a los datos demográficos, al no haberse tenido en cuenta en la formación de clusters, existe una gran variedad en la presencia de usuarios de diferentes edades y nacionalidades, así como una tendencia a una presencia equitativa de ambos géneros.

En la segunda propuesta de detección de comunidades, utilizamos el algoritmo genético que desarrollamos. Utilizando la función de fitness detallada anteriormente (Búsqueda de pesos), en la que mediamos la aptitud de cada cromosoma como el error medio entre la similitud determinada por los pesos del cromosoma y la similitud percibida por profesionales en arte entre diversos pares de cuadros, obtuvimos los siguientes pesos para las similitudes parciales entre obras:

Similitud por contenido	12%
Similitud por color dominante	4%
Similitud por tamaño	15%
Similitud por artista y movimiento	34%
Similitud por error cuadrático medio por píxel	35%

Tabla 2 Pesos de las similitudes parciales entre cuadros encontrados con el algoritmo genético

Como se puede observar en la Tabla 2 el resultado del algoritmo genético proporciona unos pesos en los que predominan la similitud por artista y movimiento y similitud por error cuadrático medio por píxel. Con estos pesos se obtuvieron 6 comunidades. Dos de ellas con más del doble de individuos que cualquiera del resto de clusters. Los artistas y movimientos están bastante repartidos de forma muy variable entre los cuadros, aunque vemos que, en cuanto a artistas, Goya y Zurbarán aparecen a menudo, y en cuanto a movimiento, lo hace Renacimiento del Norte.

En la tercera propuesta de detección de comunidades, decidimos probar darle un valor máximo al peso de las Similitud por artista y movimiento con la intención de que en cada cluster se pudiera identificar un artista o movimiento especialmente popular. Así, los pesos utilizados serían los siguientes:

Similitud por contenido	0%
Similitud por color dominante	0%
Similitud por tamaño	0%
Similitud por artista y movimiento	100%
Similitud por error cuadrático medio por píxel	0%

Tabla 3 Pesos de las similitudes parciales entre cuadros con el peso de la Similitud por artista y movimiento al máximo

El número de clusters generados con esta combinación de pesos es de 5. El tamaño de cada cluster es diferente, aunque entre tres de ellos la diferencia de individuos es mínima.

Mayoritariamente, se presentan varias obras de forma periódica, como por ejemplo The Surgeon o The Straw Manikin, de nuevo vemos que cuantas más obras tenga un autor más aparecen en los clusters. Se puede ver que dadas las opiniones que aparecen en cada cluster, un caso en el que haya más equidad numérica en artista y movimiento, seríamos capaces de detectar realmente clusters que giren en torno a ciertas obras, descartando las que, a pesar de tener muchas apariciones, no son relevantes en los clusters.

Cabe destacar que en los tres resultados mencionados siempre se genera un cluster en el que la mayoría de los usuarios son extranjeros. Es curioso este comportamiento teniendo en cuenta que la mayoría de los individuos proporcionados son de origen español y que para la generación de comunidades se han utilizado pesos muy diferentes. Por lo tanto, se podría concretar que las personas con nacionalidad extranjera son afines a cuadros, autores y movimientos artísticos concretos. Mas concretamente los cuadros Apparition of the Apostle Peter to Saint Peter Nolasco, St. Rufina of Seville y Still Life with Pots, el movimiento artístico Barroco y los artistas Francisco de Zurbaran y Diego Velazquez.

Una vez detallado el proceso de modelado y sus resultados, pasamos a explicar cómo hemos planteado el diseño y el desarrollo de la aplicación que lo implementa. En el siguiente capítulo veremos una explicación pormenorizada de la arquitectura y diseño de la aplicación, así como de las tecnologías y herramientas utilizadas durante la implementación.

Capítulo 4 - Implementación

El objetivo principal de este trabajo es el desarrollo de una herramienta que permita detectar comunidades de usuarios, experimentar con técnicas de similitud y su visualización. Sin embargo, este proyecto ofrece otras funcionalidades secundarias dignas de mencionar.

Para poder llevar a cabo la implementación se han usado diferentes técnicas y tecnologías de desarrollo software. Durante todo el desarrollo se han puesto en práctica conocimientos conocidos para desarrollo. Sin embargo, la naturaleza del proyecto ha obligado al estudio y familiarización de otras técnicas. Puesto que el desarrollo ha sido colaborativo, todos los participantes han afianzado e incrementado sus conocimientos en diferentes aspectos en el desarrollo software. Se hará mayor hincapié en este aspecto en los apéndices de la memoria.

Durante la implementación del proyecto se han utilizado diversas aplicaciones para la comunicación y el control de versiones. El uso de estas herramientas ha sido de vital importancia no solo por el déficit de comunicación presencial que ha supuesto la situación de pandemia actual, sino porque gracias a estas aplicaciones se ha podido llevar un control absoluto sobre las distintas etapas de desarrollo del proyecto. El código del proyecto está alojado en un repositorio Github (https://github.com/iagger/Modelado-de-comunidades) que utilizamos para compartirlo a lo largo de su desarrollo con instrucciones acerca de su uso.

En los siguientes apartados se explica de manera detallada el proceso de implementación de cada una de las partes.

4.1 Preprocesamiento de los datos y primeros resultados

En una primera fase de desarrollo tratamos con los datos que se nos proporcionaron inicialmente con el objetivo de familiarizarnos con ellos y poder generar unas primeras comunidades a partir de los de los datos demográficos de los usuarios. Para llevar a cabo este proceso se utilizan cuadernos o notebooks de Jupyter. Estos cuadernos permiten la programación (Python) y a su vez la compilación y ejecución de parte del código. Estos cuadernos están alojados dentro del proyecto. Concretamente en una carpeta nombrada como “notebooks”.

Utilizando los datos demográficos de los usuarios se busca poder realizar y aplicar las distintas técnicas de clustering es crucial que se realicen ciertas tareas previas. Sin este proceso previo todo el trabajo a posterior podría verse perjudicado ya que es posible que dentro del conjunto de datos existan valores incongruentes o nulos.

El preprocesamiento de datos está dividido en dos partes:

1. Visualización de los datos: se exploran cada uno de los archivos de datos proporcionados. El objetivo es visualizar los datos en crudo para conocer la naturaleza de los datos antes de su procesado.
2. Limpieza y resultados estadísticos descriptivos: para cada uno de los conjuntos de datos se realiza una exploración de los datos en profundidad. Para cada variable se comprueban si los datos son coherentes y si existen valores nulos o vacíos. Conjuntamente se muestran ciertos valores descriptivos de los conjuntos de datos. Tales como el número de individuos o la frecuencia. Para ello, se utilizan diferentes formas de visualización. Como los diagramas circulares ilustrados en las figuras Figura 17, Figura 19, Figura 18.

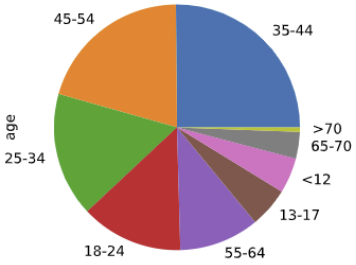

Figura 17 Frecuencia edades usuarios

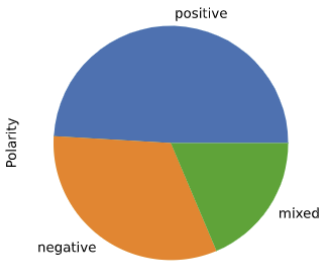

Figura 19 Frecuencia polaridad sentimientos usuarios

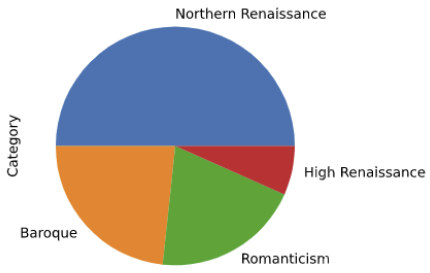

Figura 18 Frecuencia movimientos artísticos

Tras la visualizacion y limpieza de datos, tambien se han generado comunidades de usurios en una primera etapa del trabajo. Estas comunidades estan generadas con los datos de demograficos de los usuarios. Para poder aplicar las diferentes técnicas de *clustering* es necesario normalizar las variables que componen el dataset. La variable edad se normaliza usando rangos de edad. Para cada rango de edad se da un valor entre 1 y el número de rangos. Para el género se da el valor 0 para representar el femenino y 1 para el masculino. Por último, la variable pais se normaliza utilizando otro dataset. Este dataset contiene entre otra información el indice de alfabetismo, y la renta media de la población. Con esta información se calcula un valor para cada país.

Una vez formalizados los datos, se aplican dos técnicas de clustering y una de etiquetado de clustering. Por un lado, se aplica el algoritmo de k-means y se estudia cual es el mejor número de clusters a través del índice de Davies-Bouldin que mide la compactación de los clusters [17]. Posteriormente, se aplica el etiquetado *FCA* para poder visualizar las características de cada uno de los clusters generados. Para visualizar los clusters generados se utiliza una gráfica de puntos donde cada punto representa a un individuo y cada color representa al cluster al que pertenece dicho individuo. Tal como se muestra en la Figura 20.

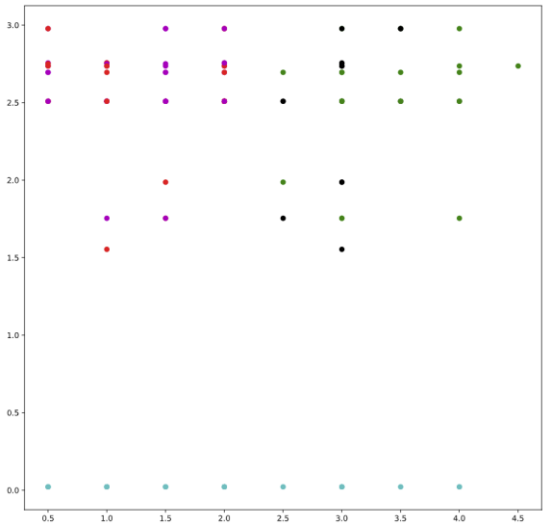

Figura 20 k-means demográficos

De igual manera gracias al etiquetado FCA se puede ilustrar la caracterización de los cluster generados tal y como se muestra en la Figura 21.

	userId	cluster	adult	youg-adult	senior	young	Spain	United States	Other	Mexico	France	Japan	Brazil	Italy	Germany	Argentina	Canada	male	female
0	1	2	X				X											X	
1	2	0		X			X												X
2	3	0		X			X												X
3	4	0		X			X												X
4	5	2	X				X											X	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
166	298	1	X											X				X	
167	299	1	X				X											X	
168	300	0		X			X												X
169	301	1	X				X											X	
170	306	2	X				X											X	

Figura 21 Etiquetado de los clusters mediante FCA

Paralelamente se aplica el algoritmo jerárquico aglomerativo con distintas estrategias que minimizan o maximizan ciertos aspectos y obteniendo así distintos resultados. Para cada una de estas estrategias también se ha estudiado la calidad de los clusters a través del índice de Davies-Bouldin. También se ilustran los clusters detectados a través del algoritmo jerárquico aglomerativo por medio de un dendrograma tal y como aparece en la Figura 22.

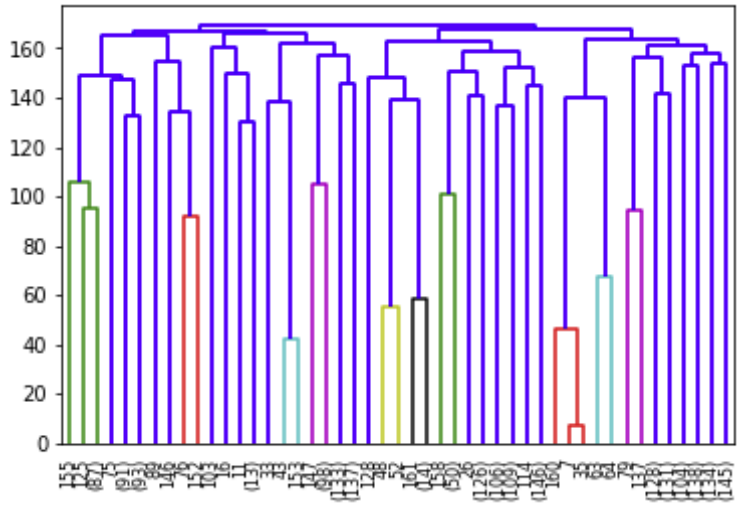

Figura 22 Jerárquico aglomerativo demográficos

Finalmente se selecciona la que obtiene mejor valor (índice de Davies-Bouldin más bajo) y de nuevo se aplica el FCA a los clusters generados en busca de caracterizar a los mismos.

Aunque esta parte del proyecto no arroja luz a la hora de modelar comunidades, si ha servido para familiarizarnos con las diversas técnicas de clustering y tecnologías.

4.2 Arquitectura de la aplicación

La aplicación que hemos implementado sigue un modelo cliente-servidor. Toda la lógica relacionada con el modelado de comunidades corre a cargo de la parte del servidor, así como las tareas de proporcionar respuesta y servicio a los distintos clientes con los que se comunique. Por otro lado, son los clientes los que mediante peticiones al servidor responden a las interacciones con el usuario y, en este caso permiten visualizar los resultados de distintas partes de la lógica del modelado de comunidades que se lleva a cabo en el servidor.

En esta línea, distinguimos dos partes claras en nuestra aplicación (Figura 23). La primera de ellas es la parte de backend, que refiere a todo lo relacionado con la lógica del servidor y la respuetas ante las peticiones de los clientes. En segundo lugar, la parte frontend implementa un cliente web que ofrece al usuario una interfaz gráfica con la que poder visualizar distintos resultados de la fase de modelado de comunidades. A continuación detallamos cada una de estas partes.

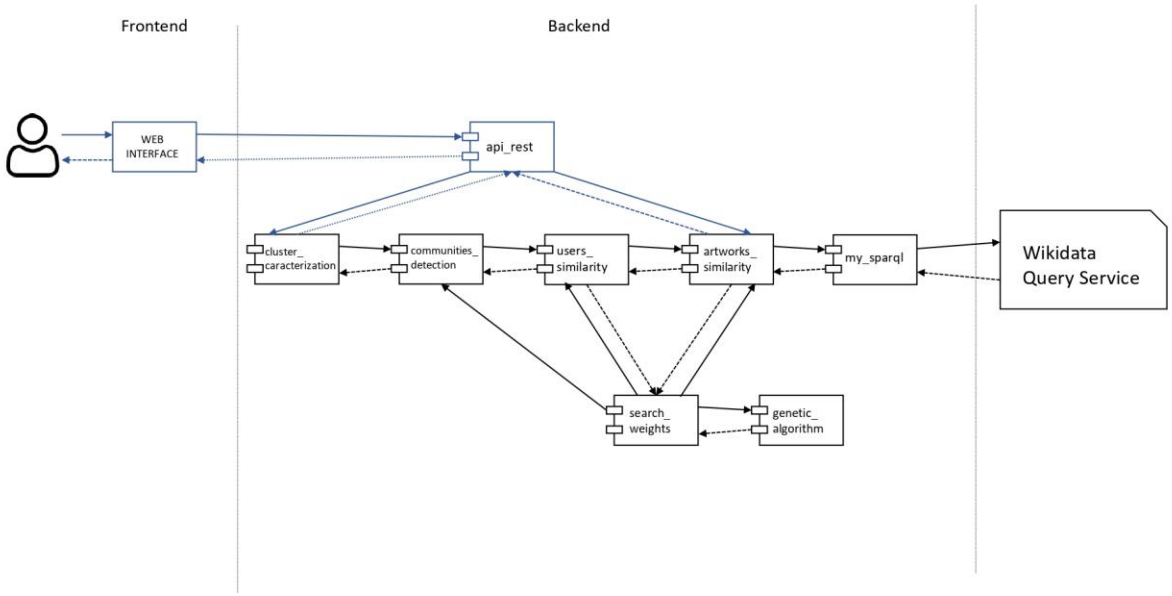

Figura 23 Esquema con los diferentes módulos de la aplicación

4.2.1 Backend

El backend de la aplicación contiene todos los módulos relacionados con la lógica del modelado de comunidades, así como los servicios REST que responden a las peticiones realizadas desde el frontend. Además, contiene también la lógica de consultas a Wikidata, de la cual depende parte del proceso de modelado a la hora de ampliar la información de la que se dispone acerca de algunos cuadros. A continuación, pasamos a detallar cada uno de los modulo contenidos en el backend y a detallar el servicio que cada uno ofrece.

4.2.1.1 Lógica del modelado

Como hemos visto en el capítulo Modelado de comunidades, el proceso de modelado consta de diversas etapas en las que se procesan los datos de los usuarios con diferentes objetivos. Incluimos desde la lógica de consultas a Wikidata para extender nuestra información la búsqueda de pesos adecuados y la formación de comunidades, pasando por la definición de las similitudes entre cuadros o entre usuarios. A continuación, detallamos los módulos relacionados con ella que encontramos en la aplicación (Figura 24).

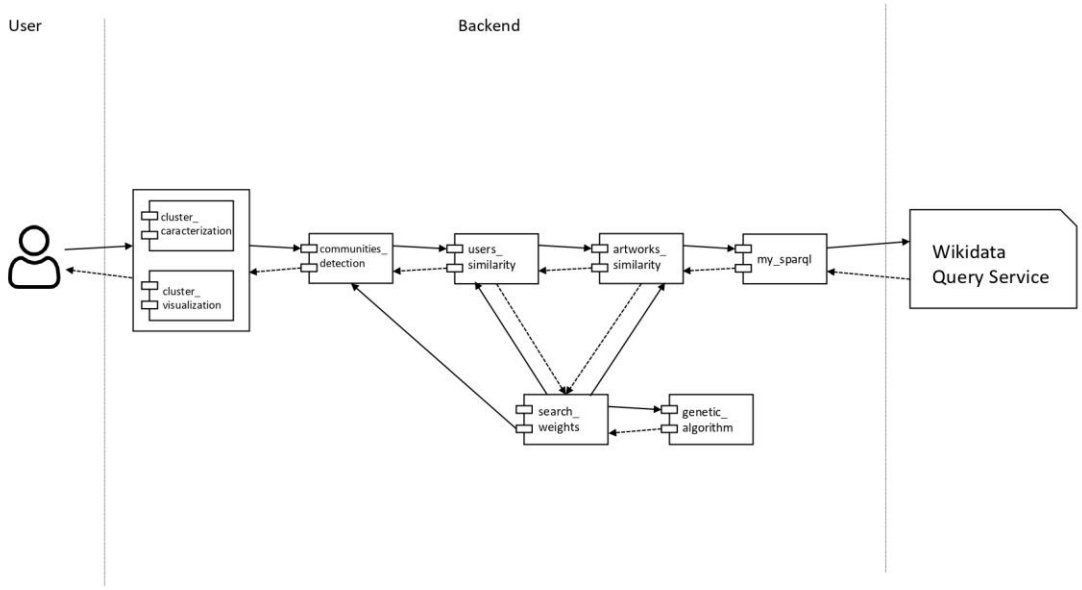

Figura 24 Estructura de la lógica del modelado

Yendo de menos a más, encontramos inicialmente el módulo my_sparql, cuyo objetivo es encapsular toda la lógica de nuestras consultas a Wikidata, así como ofrecer distintas utilidades para transformar los resultados al formato deseado. Al inicio del proyecto, preparábamos las consultas sobre distintas propiedades de los cuadros por separado, algo que resultaba repetitivo y engorroso. Observamos que las consultas que realizábamos tenían siempre la misma forma, la de recuperar cierta propiedad *P* para una determinada obra *Q*. Para, no solo facilitar el proceso de consulta, si no poder tener un sistema de cacheado en memoria de los resultados de las mismas (algo costosas en tiempo), decidimos implementar la clase PropertyRetreiver que realiza esta misma consulta para unos parámetros propiedad *P*, con la que se instancia y genera una cache o consulta una ya creada, y recupera dicha propiedad para la entidad especificada *Q* ya sea por haberla encontrado en la memoria cache o por lanzar la consulta a Wikidata mediante la librería SPARQLWrapper. Con esto conseguimos una manera rápida y limpia de recuperar propiedades de cuadros consiguiendo abstraernos de temas como las consultas SPARQL o de su persistencia en memoria.

Al igual que con las consultas SPARQL, era clave poder ofrecer un sistema de cacheado para el cálculo de similitudes entre obras, ya que en ocasiones calcularlas suponía un gran esfuerzo

computacional. En ese respecto decidimos desarrollar la clase CachedSimilarity con el objetivo de ofrecer una interfaz que abstrajese el proceso de cacheado y consulta de las similitudes a calcular. De nuevo esta clase busca en la memoria cache el resultado de la similitud de dos obras dadas, mandando calcular la misma en caso de no encontrarse mediante un método abstracto que ha de ser implementado por las clases que hereden de ella. En este sentido, hemos encapsulado cada una de las similitudes parciales entre obras en su propia clase que hereda de CachedSimilarity y que implementa en cada caso el método computeSimilarity.

Así pues, en el módulo artwork_similarity están contenidas todas las clases que encapsulan las distintas medidas de similitud parcial entre obras. Además, están definidos algunos métodos de utilidad que ofrecen diferentes utilidades basadas en dichas similitudes parciales.

Por otro lado, encontramos en el módulo users_similarity se define la similitud entre usuarios expuesta en el anterior capitulo. Al no ser posible realizar un cacheado de esta medida de similitud ya que depende totalmente de los pesos parciales, buscando una mejora de eficiencia, se calculan las obras más similares a cada una de las presentes en nuestro dataset en el momento de la instanciación. De esta manera evitamos tener que calcularlas para las obras presentes en los conjuntos de cada usuario en cada llamada.

De cara a la decisión de pesos, se ha definido en el módulo genetic_algorithm una implementación general de los algoritmos genéticos con cromosomas de números reales, con una sencilla interfaz para definir sus parámetros, rangos de los genes y una función de *fitness.* El objetivo es poder utilizar esta implementación para definir distintas funciones de fitness que puedan cumplir con las restricciones deseadas a la hora de buscar unos pesos adecuados.

A la hora de formar comunidades, podemos recurrir al módulo communities_detection que permite ejecutar algoritmos de clustering como k-medoids o DBSCAN para generar comunidades. Permite definir diversos parámetros para ejecutar dichos algoritmos o incluso colocar los pesos asociados a las similitudes parciales entre cuadros deseados. En su defecto, se tomarán los valores por defecto definidos en el fichero configuration.cfg.

Finalmente utilizan el módulo cluster_information, que implementa una serie de funciones para generar un infográfico que caracterice los resultados de la formación de comunidades, permitiendo visualizar diferentes aspectos de cada uno de los clusters creados.

4.2.1.2 Servicios REST

De cara a representar visualmente la información recopilada en base a las similitudes entre obras y usuarios, un paso inevitable es sintetizar todo exponiéndolo en un servidor en forma de datos que, posteriormente, se representan de manera gráfica con herramientas frontend (4.2.2).

De esta forma, mediante peticiones cliente-servidor, en este caso el cliente solicita cierto conjunto de datos que seguidamente el servidor, tras recibir la petición, los procesa y los envía formateados.

A través de Sanic hemos desarrollado las APIs necesarias para realizar las funciones que desempeñan nuestro trabajo.

Tal y como se explica en frontend, a través de la interfaz visual todos los parámetros de entrada se reciben y se tratan, para a continuación enviar una respuesta con los campos necesarios para que se pueda formatear y mostrar con HTML, Javascript y CSS.

A continuación, se definen los cuatro servicios disponibles en la restful API:

Para comenzar, se define como endpoint al punto final de comunicación con un servicio, en este caso es la ruta que, al acceder a la misma, inicia el proceso que ejecuta el servicio. El endpoint que recibe el primer servicio es la base de la aplicación: “/artworks”. Este servicio no necesita ningún parámetro de entrada y su único propósito es enviar una respuesta en formato JSON que contiene una lista de todos los cuadros guardados en el sistema, con varios atributos de los mismos.

El siguiente servicio, se implementa utilizando la lógica de modelado para generar los cuadros más similares a uno dado. El servicio se define con el siguiente endpoint: "/artworks/similarity/artworkID". En cuanto a los parámetros de entrada, se utiliza un query param, que se corresponde con el ID de un cuadro para complementar la ruta que define esta similitud. Por otro lado, existe la opción de proporcionarle un cuerpo a la petición, que ofrece la posibilidad de personalizar los pesos que definen las variables de similitud (Figura 25). El servicio devuelve un JSON que lleva los datos más relevantes del cuadro especificado y los *k* cuadros más similares, junto con sus datos y el % de similitud que tienen con el mismo.

Otro de estos servicios, tiene la finalidad de mostrar los clusters que se generan a través de

Figura 25 Estructura del cuerpo de una petición

las similitudes entre los usuarios y sus gustos artísticos. El endpoint definido para este servicio es: "/artworks/similarity/clusters". Esta visualización se realiza devolviendo un archivo con formato PDF que contiene todos los datos relevantes de cada cluster.

Por último, este servicio permite visualizar los clusters encontrados directamente en la ruta del endpoint "/artworks/similarity/clustersHTML" formateado con HTML, CSS y Javascript. El servicio devuelve todos estos clusters representados en formato JSON para que este formateo pueda realizarse y permitir una visualización cómoda.

4.2.2 Frontend

El *frontend* es la sección de código que encapsula toda la programación para la visualización e interacción del usuario con la aplicación. En concreto, la aplicación permite la visualización de las similitudes entre las obras de arte, así como los resultados de las comunidades

detectadas. Antes de exponer el proceso y técnicas de desarrollo utilizadas, es importante destacar las funcionalidades principales que ofrece la interfaz web:

1. Interfaz para la visualización de las similitudes entre las obras de arte: Esta funcionalidad tiene por objetivo ilustrar la comparación de cuadros bajo ciertos criterios. En concreto, se permite seleccionar una obra de arte y un peso para cada una de las similitudes parciales. Tras seleccionar el cuadro y determinar un peso concreto para cada similitud parcial, se muestran los cinco cuadros más similares. Esta herramienta permite ver la influencia de los pesos a la hora de decidir cuáles son los cuadros más similares a otro dado.

2. Interfaz para la visualización de comunidades usuarios: esta funcionalidad permite al usuario visualizar la información referente a las comunidades de usuarios detectadas. Para poder visualizar las comunidades de usuarios es necesario generarlas previamente. Esta tarea se lleva a cabo en el backend de la aplicación. Para generar comunidades de usuarios, previamente hay que definir unos pesos para las similitudes parciales entre cuadros. Tras definir unos pesos concretos, es necesario crear una matriz de distancias entre cuadros. Una vez generada la matriz de distancia entre cuadros, esta se utiliza para concretar la similitud entre usuarios. Una vez concretada la medida de similitud entre usuarios se usan diferentes algoritmos de detección de comunidades para la generación de comunidades de usuarios. Este proceso es muy costoso en cuanto al tiempo que necesita. Por esta razón las comunidades que se visualizan en la aplicación están generadas previamente. Estas comunidades fueron generadas tras fijar ciertos criterios en la elección de pesos, tal como se explica en el capítulo 3.2.2.5. En concreto, la interfaz solicitará al backend los datos referentes a la información de las comunidades detectadas.

Como ya se adelantó al comienzo de la sección, para el desarrollo del frontend se utiliza un patrón de desarrollo web. Este hace uso de tres lenguajes de programación: HTML, hojas de estilo CSS y JavaScript. De igual manera se han utilizado librerías externas que proporcionan herramientas tanto para la comunicación con el servicio REST como para la visualización de los

elementos de la interfaz. En concreto se utilizan las librerías: Bootstrap (CSS y JavaScript), JQuery, Morris y Raphael (JavaScript).

A continuación, se explican los rasgos más destacables de la implementación para las distintas opciones de visualización que se ofrecen:

1. **Portada de la aplicación:** el inicio de la aplicación cuenta con una interfaz en la permite al usuario elegir por medio de dos botones los dos servicios principales que ofrece la aplicación. Uno redirige al usuario a la interfaz para la visualización de las similitudes entre las obras de arte y el otro a interfaz para la visualización de comunidades usuarios respectivamente.

Figura 26 Portada de la aplicación web

Tal y como se muestra en la Figura 26, la interfaz hace uso de diferentes hojas de estilo para proporcionar una vista más agradable para el usuario.

2. **Visionado de comunidades**: esta interfaz del frontend permite al usuario visualizar de las distintas comunidades detectadas. En concreto, esta interfaz está dividida en tres grandes secciones. Por un lado, permite al usuario seleccionar uno de los resultados generados en el backend previamente. Cada uno de estos resultados han sido generados

con unos pesos distintos para las similitudes parciales entre cuadros. Para recuperar esta información es necesario que la interfaz haga uso del servicio REST. Particularmente el servicio que tiene como endpoint "/artworks/similarity/clustersHTML" y que devuelve un JSON que contiene la información de cada uno de los resultados generados. Por otro lado, muestra al usuario la información correspondiente al resultado seleccionado. Esta información está compuesta por el algoritmo utilizado para la generación de comunidades, los valores de los parámetros de ese algoritmo, una descripción del tipo de comunidades generadas y los pesos de cada una de las similitudes parciales entre cuadros tal y como se muestra en la Figura 27. Esta información permite al usuario conocer las características del método utilizado para la generación de las comunidades.

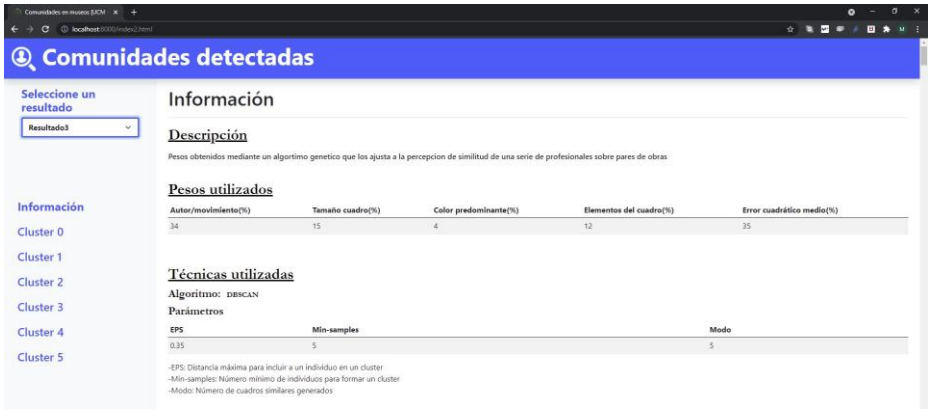

Figura 27 Interfaz para la visualización de la información de la técnica utilizada en el proceso de generación de comunidades

Por último, la interfaz de visualización de comunidades permite al usuario explorar la información detallada de cada uno de los clusters. Para representar esta información se han caracterizado cada uno de los clusters de la siguiente manera: En un primer lugar se encuentra el número de individuos que componen el cluster así como un histograma por cada variable demográfica (edad, nacionalidad y género) como se muestra en la Figura 28.

Figura 28 Información de los datos demográficos de un cluster

En segundo lugar, se encuentran las obras más y menos populares dentro del cluster. Para cada una de las obras se muestra el título, el número de valoraciones positivas y negativas de los usuarios. Toda esta información reside en un carrusel de imágenes para proporcionar al usuario una vista cómoda de esta información como se ilustra en la Figura 29.

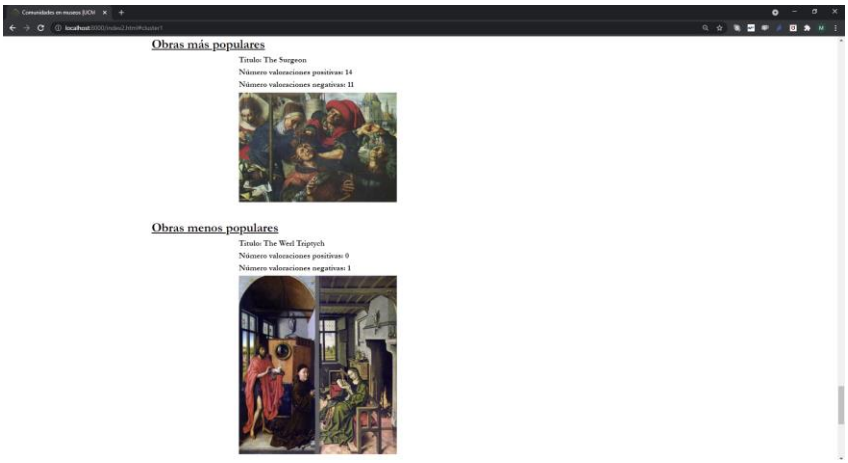

Figura 29 Obras de arte más y menos populares dentro del cluster

En tercer y último lugar, se visualizan los artistas valorados positiva y negativamente, movimiento artístico más gustado y los sentimientos más comunes dentro del cluster. Para representar la frecuencia de los sentimientos de los usuarios del cluster se utiliza un diagrama circular en el que cada color representa un sentimiento como se observa en la Figura 30.

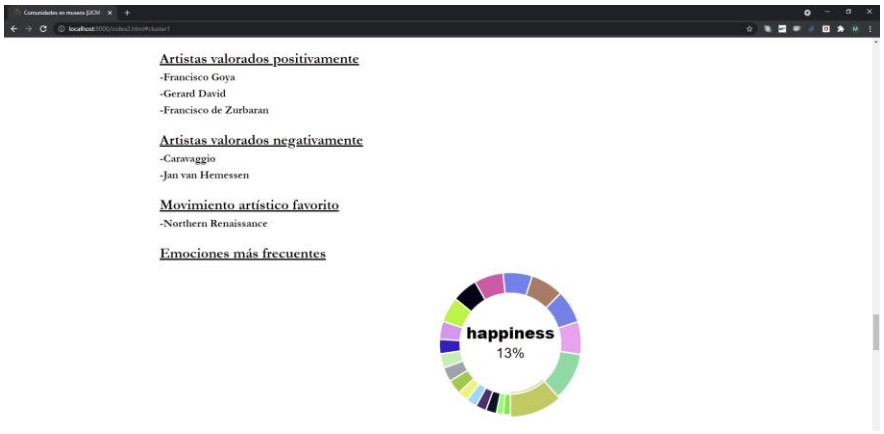

Figura 30 Artistas mejor y peor valorados, movimiento artístico más gustado y emociones más comunes de un cluster

3. **Visionado de similitud entre cuadros**: esta interfaz permite al usuario seleccionar una obra de arte, unos pesos para las similitudes parciales y muestra los cinco cuadros más similares al elegido por el usuario utilizando los pesos introducidos por el usuario. Esta interfaz esta divida claramente en tres secciones tal y como se muestra en la Figura 31. En la primera sección el usuario puede seleccionar un cuadro del conjunto de obras de arte proporcionado por los directores. De igual manera, se muestra el título, autor y movimiento artístico del cuadro. Permitiendo de esta manera ver las características principales de la obra de arte seleccionada. Para poder mostrar esta información la interfaz usa el servicio REST con endpoint "/artworks" que retorna un JSON con la información de los cuadros. En la segunda sección, el usuario puede introducir el peso para cada una de las similitudes parciales entre cuadros. Estas a su vez deben sumar uno. En la última sección, aparecen los cuadros más parecidos al seleccionado por el usuario generados con los pesos introducidos. Estos cuadros se muestran tras pulsar el botón "Genera similares". Cuando este es pulsado la interfaz hace uso del servicio REST con endpoint "/artworks/similarity/artworkID" pasándole la información introducida por el usuario. Es decir, el cuadro y los pesos seleccionados. Como respuesta se obtiene un JSON con los cinco cuadros más similares. Además de mostrar las imágenes correspondientes a las cinco obras más similares, también muestra el título, autor, movimiento artístico e índice de similitud. Todos estos elementos residen en un carrusel de imágenes facilitando así la visualización.

Figura 31 Visualización de la interfaz generadora de cuadros similares

4.3 Tecnologías y recursos externos utilizados

A la hora de desarrollar el código del proyecto tuvimos que hacer uso de diversas herramientas. En ocasiones las habíamos usado antes y conocíamos detalladamente como hacer uso de ellas, en otros casos tuvimos que estudiarlas y aprender a utilizarlas.

Al ser un proyecto grupal, se han usado diversos medios para la coordinación y comunicación. Estas herramientas han sido fundamentales para el buen desarrollo del proyecto. Ya que nos han permitido una comunicación directa y fácil entre los integrantes y un control sobre las versiones durante la etapa de implementación.

A continuación, enumeraremos las herramientas más destacables usadas para el desarrollo y la coordinación.

4.3.1 Desarrollo

Con respecto al apartado técnico es donde principalmente nos encontramos con nuevas tecnologías, lo cual nos ha supuesto un reto más en la implementación de nuestro proyecto, ya que no sólo nos limitamos a analizar y desarrollar lo relativo a las comunidades, sino también influye el proceso de adaptación que conlleva utilizar herramientas desconocidas.

4.3.1.1 Python

Ha sido el lenguaje principal del proyecto, predominante tanto en la fase de modelado como en el desarrollo de la aplicación servidor (api-rest) con la que trabaja la interfaz. Presente desde los primeros notebooks que desarrollamos hasta el grueso de los scripts de la lógica de modelado de comunidades, Python ofrece una gran versatilidad a la hora de trabajar con datos, así como una gran comodidad para usuarios más novatos. Además, Python ofrece multitud de librerías que nos han sido de gran utilidad a la hora de implementar el proyecto.

4.3.1.2 Numpy

Numpy es la librería de manejo de estructuras numéricas por excelencia de Python. Ofrece un gran soporte a la hora de tratar con vectores y matrices, así como una cómoda e intuitiva interfaz para manejarlos. Numpy ofrece además una gran cantidad de métodos para obtener información o modificar este tipo de estructuras de forma limpia y eficiente.

4.3.1.3 Pandas

Pandas es una librería para la manipulación y análisis de datos en Python. Su principal estructura, los DataFrame, permiten cargar, estudiar, recorrer y modificar los datos de forma rápida y sencilla.

4.3.1.4 Scikit-Learn

Scikit-Learn es una librería para Python de software libre que ofrece una rica selección de utilidades relacionadas con el aprendizaje automático. Muchos de sus métodos de clustering, de preprocesamiento o evaluación han sido utilizados a la hora de formar y evaluar las comunidades.

4.3.1.5 SPARQLWrapper

Una librería Python que ofrece la posibilidad de realizar consultas SPARQL sobre diferentes motores de búsqueda. De gran utilidad a la hora de recuperar información en Wikidata acerca de las distintas obras.

4.3.1.6 Filecache

FileCache es una librería Python que ofrece una cómoda interfaz para guardar información en memoria de manera consistente. El objeto FileCache se maneja igual que un diccionario, con

la salvedad de que la información va siendo salvada en la carpeta con la que se instancia. Muy a la hora de salvar el resultado de las consultas SPARQL (muy costosas en tiempo) o para evitar ejecutar cálculos costosos en exceso.

4.3.1.7 Colorsys

Es una librería de Python que permite la conversión entre modelos de color. Su uso se limita a convertir las coordenadas de una expresión a otra (por ejemplo: Conversión de RGB a HSV pasando por parámetro los valores de Red, Green y Blue).

4.3.1.8 Sanic

Sanic es un framework asíncrono enfocado a las tecnologías web y servidor. Está hecho para ser el intermedio perfecto entre complejidad de uso y gran amplitud de posibilidades a desarrollar en funcionalidad. La elección de este framework para representar nuestro trabajo de manera visual, en conjunción de la funcionalidad web que creamos con Sanic y el apartado gráfico que le dan las tecnologías que comentamos en el apartado siguiente (4.3.1.9), se ha visto afectada por el beneficio que ofrece la escalabilidad y velocidad de esta opción.

4.3.1.9 HTML, CSS y JavaScript

HTML, CSS y JavaScript componen los tres lenguajes de programación web más usados en la actualidad. HTML es un lenguaje basado en etiquetas y estaría en la parte más alta conceptualmente. Es principalmente el código de los elementos presentes o no en la interfaz. Para una buena praxis el HTML debe ir acompañado de una o varias hojas de estilo CSS. CSS es un lenguaje de programación que permite de manera sencilla modificar los estilos de los componentes de la interfaz. Cabe destacar el uso de la librería Bootstrap que contiene entre otros ficheros, hojas de estilo encapsuladas en clases. Es muy útil ya que basta con añadir la clase al componente HTML para aplicar ese estilo.

Por último, se necesita del uso de JavaScript para poder realizar ciertos procesos. JavaScript es un lenguaje de programación que permite entre otras cosas realizar pequeños procedimientos y la comunicación con servidores. Es importante destacar el uso de la librería JQuery.js para la comunicación con el servidor. Esta librería ofrece un servicio muy cómodo para hacer peticiones al servidor.

4.3.1.10 Jupyter notebook

Jupyter Notebook es un entorno de desarrollo interactivo basado en web que permite crear documentos de código Python, permitiendo visualizar resultados de distintos bloques de código, así como incluir comentarios de texto en formato Markdown con toda la flexibilidad que este lenguaje ofrece.

4.3.1.11 PyCharm y Visual Studio Code

Ambos son entornos de desarrollo que permiten trabajar con Python, que ofrecen una gran flexibilidad y comodidad a la hora de programar, depurar e identificar errores. Visual Studio Code además permite incluir diferentes plugins de utilidad para visualizar todo tipo de archivos de datos, trabajar con notebooks ofreciendo una interfaz limpia y sencilla.

4.3.1.12 Postman

En conjunción con el servicio que creamos con *Sanic*, hemos utilizados Postman para poder enviar y recibir peticiones y así, poder trabajar en base a lo que nos trasmite la herramienta.

4.3.2 Coordinación

Al tratarse de un proyecto colaborativo, se han usado diferentes tecnologías que permiten el control de versiones, el desarrollo conjunto de un documento y la comunicación entre los participantes.

4.3.2.1 Github

A través de un repositorio en GitHub, pudimos compartir, coordinar y visualizar el progreso del proyecto tanto entre los componentes del equipo como con los directores del TFG. Así además será accesible para cualquiera que quiera acceder a él para consultarlo o para trabajo futuro.

4.3.2.2 Google Meet

Es la herramienta de comunicación que más hemos utilizado, tanto para comunicarnos con los directores como para las reuniones semanales de los miembros del equipo. Con las dificultades para mantener reuniones presenciales, ha sido de gran utilidad contar con ella.

4.3.2.3 Microsoft Office Word

A la hora de redactar la memoria utilizamos Microsoft Word. Se trata de una de las herramientas cómodas y con más funcionalidades en lo que a la escritura de documentos concierne. Además, ofrece la posibilidad de la edición colaborativa y simultánea del documento, por lo que es muy útil evitar tener que estar compartiendo el documento constantemente.

Capítulo 5 - Conclusiones y trabajo futuro

En este apartado culminamos recapitulando en torno a las ideas y procesos que se nos han ido presentando a lo largo de todo el desarrollo del trabajo, de manera que podamos plasmar una reflexión general sobre ello y posteriormente comentar las propuestas que tenemos a futuro por si se da el caso de retomar el trabajo.

5.1 Conclusiones

Este trabajo ha permitido la creación de un mecanismo capaz de detectar y visualizar comunidades de usuarios, experimentar y observar la relación existente a la hora de definir una medida de similitud, en nuestro caso, entre usuarios y su efecto al detectar comunidades. Hemos podido comprobar que priorizar los gustos en lo que concierne a al arte a la hora de encontrarlas es de gran utilidad para que se formen grupos con características similares, lo que combinado con las herramientas y tecnologías adecuadas puede tener diversas aplicaciones como comentaremos en el siguiente apartado.

A la hora de encontrar comunidades utilizando los datos de nuestro caso de estudio, hemos observado comunidades muy ruidosas, con individuos muy diferentes entre sí en algunos casos o directamente sin ser incluidos en ninguna de ellas. Este problema se podría explicar teniendo en cuenta la naturaleza de nuestros datos, ya que el número de usuarios con que hemos trabajado es algo limitado y quizá tengan gustos artísticos muy variados y heterogéneos.

Es importante destacar la relación detectada entre las comunidades encontradas y la medida de similitud entre usuarios. En nuestro trabajo, se ha definido la similitud entre usuarios en base a sus opiniones sobre ciertas obras de arte. Por lo tanto, también se ha definido una medida de similitud entre las obras de arte. Esta medida de similitud está compuesta por otras similitudes parciales que usan algunas de las características de los cuadros. El peso de cada una de las similitudes parciales entre cuadros es fundamental y tiene una relación directa a la hora de detectar distintas comunidades.

El uso de diferentes técnicas de clustering ha influido directamente en los resultados arrojados. Aunque se ha podido experimentar con distintas técnicas de clustering (k-means, jerárquico aglomerativo), la naturaleza de nuestro problema nos ha permitido la experimentación con otras técnicas distintas. En concreto, se ha usado k-means y DBSCAN. Estas han permitido comprobar las diferentes comunidades detectadas al usar cada una de ellas. Hemos podido comprobar que mientras que k-means tiene como resultado clusters más heterogéneos, DBSCAN ha conseguido clusters con individuos más similares entre sí. Esto esta provocado ya que uno incluye a cada usuario en un cluster (k-means) y el otro no. Gracias a esto se ha podido estudiar el carácter de cada una de las técnicas, pudiendo ser usada cada una de ellas en función de un propósito determinado.

La herramienta de detección que hemos desarrollado es altamente parametrizable y por tanto puede ofrecer multitud de posibles soluciones a la hora de encontrar diferentes grupos de usuarios. Por ello tanto el algoritmo genético como la interfaz de visualización de las similitudes entre obras son un complemento idóneo a la hora de decidir los pesos más adecuados.

Centrándonos en nuestro trabajo, consideramos que en líneas generales hemos cumplido con la mayoría nuestros objetivos. Hemos tenido la posibilidad de trabajar con diversas técnicas de clustering, lo cual nos ha permitido comprender en profundidad las diferencias entre ellas. Además, nos hemos familiarizado con varios entornos de trabajo y diferentes herramientas, lo que ha contribuido a nuestro aprendizaje. En lo concerniente a los objetivos del desarrollo de la aplicación también hemos cumplido a grandes rasgos con los objetivos propuestos consiguiendo llegar a un resultado satisfactorio.

5.2 Trabajo futuro

Considerando que se han cumplido los objetivos principales del trabajo, es importante destacar las distintas opciones de crecimiento del proyecto o posibles variantes del mismo que pudieran arrojar una mayor riqueza al proyecto. A continuación, se comentan las tareas más destacables.

Aunque se ha desarrollado una aplicación que permite visualizar comunidades y experimentar con los pesos de las similitudes parciales entre cuadros, un objetivo es diseñar y

construir una aplicación que permita la recomendación de contenido explícitamente. El usuario respondería a unas breves preguntas sobre ciertas obras de arte y se le incluiría en uno o en varios clusters. Permitiendo, así, la recomendación personalizada de contenido de un museo. A su vez, esta funcionalidad podría expandirse permitiendo al usuario cambiar de cluster libremente.

Esta funcionalidad, a su vez, podría extenderse prácticamente de manera natural a ofrecer también la posibilidad de recomendar a otros usuarios con gustos semejantes para motivar a las personas a socializar y compartir opiniones y afición.

Por otro lado, sería interesante perfeccionar la manera en la que se clasifica a los usuarios de forma que dependiendo de su perfil, siendo aficionado o profesional en el arte 2 ejemplos de perfiles básicos, el sistema de recomendación asigne pesos específicos por defecto a las variables de similitud que realizan la comparación y posterior asignación.

El párrafo anterior nos lleva a la posibilidad de que a medida que se necesite más precisión y los perfiles aumenten, también se implementen más variables de similitud entre las obras para ofrecer un sistema de clasificación lo más completo posible.

Es inevitable que, si queremos hacer nuestro trabajo realmente útil y escalable, deberemos publicar el proyecto en un servidor online, de manera que pueda acceder cualquier persona de manera remota y así poder explorar las posibilidades de la interfaz y el desarrollo. Esto a su vez implica constante mantenimiento tanto por parte del backend, como del frontend; buscando que además de solucionar problemas, se optimice el rendimiento, se mejore la interfaz visual para ser lo más accesible posible y también que sea lo más actual posible gráficamente para atraer la atención de los usuarios.

Tras una entrevista realizada a un profesional del arte también nos planteamos la posibilidad de añadir más similitudes parciales entre las obras de arte. Utilizar la textura, luminosidad y técnica presente en los cuadros enriquecería la similitud entre cuadros.

Por último, poder determinar unos pesos concretos para cada una de las similitudes parciales entre usuarios. Obtener la opinión de diferentes grupos de usuarios para determinar qué pesos utilizar para detectar comunidades.

Capítulo 6 - Introduction

Humans are social by nature. Along the history, they tried to organize in groups to not only surviving, but also sharing habits, tastes, culture or routines between its members. The feeling of being a member of a community which has same features may be gratifying for these individuals.

Social science is taking benefits from the community concept at the time of performing studies about population, as it results on an easier way to find structures or characterizing groups of persons with common features, instead of studying them individually.

Precisely, the idea is to share elements or features where resides the power or being able to model and detect communities, as if we are capable to gather a group of individuals around an idea, hobby, habit or common value, in addition to promoting the socialization and communication, which it's always positive to progress as a person, will be easier to send information, offer products or recommend any kind of contents.

On the other hand, we live in a moment where acquiring data via sensors, mobile technologies or simply surfing the internet is in a hectic moment. All this, in addition to different techniques of data extraction are called Big-Data. These techniques allow, among many other things, detecting communities through machine learning mechanisms.

Therefore, this work seek to developing tools to detect and visualize communities. These tools will have different applications, like generating a series of user's communities from a museum based on a set of ratings about different artworks, alongside with some personal features of them, having the objective of being able to study their taste easily and using these communities to optimize our tools to recommend contents for them, both artworks that they might like and people that surely have their same tastes.

6.1 Motivation

This work's objective is to find, model and characterize a series of communities of people from art museums. Starting from a set of users which have demographic data and diverse opinions about different artworks, the idea is to define communities of users exploring similarities and differences between these being said users, trying to find common traits, either on demographic level or by artistic assessment. Although the tools we will use are focused on art in this case, they can be applied to other fields adapting the community's similarity metrics.

Applying various modeling types and algorithms our intention is to identify and visualize these groups to find a mechanism that allows us to specify user's groups to be used later in other purposes like recommending museum's contents, tracing the best routes to visit the artworks based on their opinions or interacting with people in the same community.

6.2 Objectives

Our work's most relevant objectives are the following:

- Comparing different clustering techniques, their differences and similarities to try to find out which one is the most suitable, both to get to our objective on an optimized way and with the lowest cost of time possible, and to form communities depending on the situation and the context in which it is to be applied.
- Defining a similarity measure between artworks that we can use to relate people in base of their opinion about artworks. In this process we will define a set of partial similarities based on different features of the paintings, which we will use along with specific weight measures, to combine them both into a general similarity measure. Also, experimenting with the impact that the similarity metric has in the community detection process.
- To develop an interface to interact with mentioned similarity between paintings, allowing us to visualize them and adjusting the weights of the partial similarities to monitor these new results.

- Modeling a set of communities using clustering techniques to define distances between users based on similarity metrics, with the purpose of finding other users that rated the same way these paintings or their similar.
- Experimenting with existing visualizing tools and deciding which is the most convenient. Proposing it to characterize the communities in base of various criteria either demographic, art tastes or on any way their cohesion.

Our personal learning objectives are:

- Facing during our development, concepts and programming techniques that are not familiar to us to allow us expanding our horizons and knowledge.
- Discovering and using different tools which will help us to develop our ideas, both technical and theoretical.

6.3 Work plan

Once we exposed what is our motivation in this work and which are the objectives we would like to achieve, let's see how we organized the development.

In the early stages of the project we tried to familiarize with different clustering techniques, studying their features and how they work. Conclusions and summary of our clustering techniques review are included in chapter 2.

Once we got to know in detail each of these techniques, we applied them into our set of demographic data to try to understand how relevant could they be when defining communities and discarding some of them, only keeping the satisfactory ones. Later, as the results of the demographic data analysis weren't propitious, we decided to focus on it in another way, considering user's ratings on artworks, which later on would finally be the feature we will use to model communities. In order to work around these opinions, we decided to define a similarity measure between paintings, which is composed by diverse partial similarities formed by various artwork's features as we explain in chapter 3.

To be able to visualize and understand thoroughly similarity measure, we developed an interface that allows us to adjust weights in real time and see a chosen painting's other most similar ones. This part is explained in chapter 4, both the backend and frontend.

Seeking for a method or criteria to define which weights are the most relevant to compare artworks and create communities, either a developing a tool that searches for it or helps us with relevant information that could help us deciding.

Once we established the similarities between paintings, we applied them to find users that rated the same way these artworks. Finally, we represent visually the most relevant data of these user's communities.

Capítulo 7 - Conclusions and future work

In this section we culminate by recapitulating around ideas and processes that have been appearing to us throughout the entire project, so that we can translate it into a general thought. Afterwards, we will comment on proposals we have in order to maintain and expand our work, in case we take it up again.

7.1 Conclusions

This work allowed us creating a mechanism capable to detect and visualize user's communities, experimenting and observing the existent relationship when defining a similarity metric, in this case, between users and its effect detecting communities. We proved that prioritizing art tastes is so useful to form groups with similar features, which in combination with the appropriate technologies and tools may have numerous applications as we explain in the next paragraph.

At the time we found communities using our study case's data, we found out that there are noisy communities, with very different individuals or not even included anywhere. This problem may be explained considering the data nature, as the number of users that we worked with is limited and they may have so varied art tastes.

It's important to note the detected relationship between found communities and the user's similarity metric. In our work, it has been defined this similarity based on people's artistic opinions. Therefore, an artwork similarity metric has been defined as well. This metric is compound by partial similarities that use some artworks features. Each partial similarity's weight essential and has direct relationship with detecting various communities.

Usage of different clustering techniques has influenced in the results. Although we experimented with different clustering techniques (k-means, hierarchical agglomerative), the nature of our problem has permitted us trying some different techniques. Specially, we used k-medoids and DBSCAN. These allowed checking on the different detected communities. We proved that k-medoids results on heterogeneous clusters, while DBSCAN has got *clusters* with more similar individuals. This was caused because k-medoids includes each user in a different

cluster and DBSCAN not. Thanks to this we could study the nature of each technique, allowing us to use them when needed.

The detection tool we developed is highly parametrizable and that offers plenty of possible solutions finding different groups of users. For this reason, both the genetic algorithm and the visualization interface are the best component deciding when weights are the adequate.

Focusing on our work, we considered that in general terms we mostly accomplished our objectives. We had the opportunity to work with various clustering techniques, which allowed us to deeply understand their differences. Also, we got familiarized with different work environments and tools, with help us learning. Concerning our development objectives we reached our goals with satisfactory results.

7.2 Future work

Considering we reached our principal objectives, it is important to note different growing options and possible variants to enhance our project. Then, we will comment the most relevant tasks to do.

Although it has been developed an application that visualizes communities and allows to experiment with different weights for the partial similarities, an important objective is to design and build and application to recommend contents to the users. The user has to ask short questions about artworks so the system can situate him in the most convenient set and recommend him custom contents from a museum. This functionality may be extended allowing the user changing his cluster freely.

This functionality also could be extended naturally to offer the possibility to meet users with same art tastes than him to motivate socialization, share opinions and hobbies.

On the other hand, would be interesting to improve the way users get classified, considering their profile, having professional and amateur as two examples of this feature. The system will assign specific weights to compare artworks and generate more precise communities.

The previous paragraph leads us to the possibility of increasing the number of profiles and precision to create groups, so that would need to also increase the similarity variables to offer the most complete classification system possible.

It is necessary that if we want our work to be really useful and scalable, we will need to publish the project on an online server, so that anyone can access to it remotely and explore all the interface and development's possibilities. This also implies to constantly maintain the backend and frontend, solving bugs, optimizing its performance and improving the interface to make it as accessible and graphically attractive to draw the attention of users.

After interviewing an art professional we considered adding more partial similarities between artworks. Using texture, brightness and present techniques would enhance similarities.

Finally, be able to determine concrete weights for every partial similarity. Obtain opinions of different groups of people to decide which weights to use to detect communities.

BIBLIOGRAFÍA

[1] D. E. Holmes, Big Data: una breve introducción, Antoni Bosch editor, 2018.

[2] J. B. y. P. V. D.B. Millán, Aprendizaje automático, 2006.

[3] H. Chung, Clustering, Dimensionality reduction and Side Information, 2006.

[4] T. Pang-Ning, M. Steinbach, A. Karpatne y V. Kumar, Introduction to Data Mining, 2006.

[5] T. Hastie, R. Tibshirani y F. J.H., The elements of statistical learning : data mining, inference, and prediction, 2001.

[6] «K-Medoids scikit-learn-extra,» scikit-learn-extra, [En línea]. Available: https://scikit-learn-extra.readthedocs.io/en/latest/modules/cluster.html#k-medoids.

[7] «DBSCAN scikit-learn,» scikit-learn, [En línea]. Available: https://scikit-learn.org/stable/modules/clustering.html#dbscan.

[8] D. Allemang y J. Hendler, Semantic Web for the Working Ontologist : Effective Modeling in RDFS and OWL, Elsevier Science & Technology, 2011.

[9] «W3C Semantic Web Standards,» W3C, [En línea]. Available: https://www.w3.org/standards/semanticweb/data. [Último acceso: Mayo 2021].

[10] «Wikidata:Introduction,» Wikimedia, [En línea]. Available: https://www.wikidata.org/wiki/Wikidata:Introduction. [Último acceso: 23 05 2021].

[11] S. P. Pages, Relación corporal en la obra escultórica, tamaño y proporción como elemento implicado en la percepción tridimensional, Universitat de Barcelona, 2005.

[12] L. P. A. y. C. S. Vicente, «LOS VISITANTES DEL MUSEO DEL PRADO: NUEVA METODOLOGÍA DE MEDICIÓN DEL TURISMO CULTURAL,» *Estudios Turísticos,* nº 168 , pp. 85-98, 2006.

[13] D. J. G. Villar, *EL ARTISTA, LA PINTURA Y EL ESPECTADOR,* Universidad de Granada, 2012.

[14] J. L. Muñoz, El Greco y Santa Olalla, 1993.

[15] J. L. Porras, Impresionismo.

[16] M. Á. G. Villegas, Inferencia estadística, Ediciones Díaz de Santos, 2005-01-01.

[17] K. Sharma, Microarray Analysis, Momentum Press, 2015-06-19.

APÉNDICES

Apéndice A - Reparto del trabajo

A lo largo del desarrollo del proyecto, hemos procurado mantener un reparto equitativo de las cargas de trabajo. Para ello hemos concertado reuniones de equipo semanales para planificar y repartir las tareas equilibradamente. Además, hemos intentado que en todo momento cada uno de los miembros del equipo estuviera cómodo con el trabajo que se le asignaba, teniendo en cuenta los campos en los que más soltura y facilidad tenía, con el objetivo de alcanzar la mayor productividad y eficiencia posibles.

La primera fase del trabajo consistió principalmente en la documentación acerca del ámbito en el que íbamos a trabajar. Esta fase se realizó de manera conjunta, ya que preferimos que esto fuera un esfuerzo grupal para que todos tuviéramos una visión lo más amplia y clara posible acerca de las técnicas y tecnologías con que íbamos a trabajar. A lo largo de esta fase, preparamos diversas presentaciones acerca de las técnicas de clustering que pudimos poner en común mediante reuniones telemáticas para discutir sobre ellas y resolver posibles dudas.

En cuanto al proceso de estudio de datos, modelado de comunidades, desarrollo de las interfaces web y toda la fase de implementación que conllevó, sí realizamos una división más clara de las cargas de trabajo, como detallaremos a continuación.

Vadym Batsula Bilenka

En lo referido a la lógica de modelado de comunidades e implementación del trabajo participó en partes de ambas fases, en paralelo con la investigación y aprendizaje del lenguaje de programación que se utiliza y al aprendizaje automático.

Respecto a la fase de modelado, aportó la parte que establece la similitud entre cuadros en base a los colores que este contiene, analizando cuál es la mejor manera y más precisa de realizar esto, y concluyendo finalmente en la decisión de utilizar el modelo de color HSV y la técnica de clustering k-means.

Se encargó a implementar de manera completa el servidor de microservicios que comunica la implementación del modelado de comunidades con el frontend que representa la información obtenida. Esto conllevó el análisis de los diferentes frameworks que permiten la cómoda realización de esta tarea, para seleccionar el que más se ajustara a nuestras necesidades. La elección terminó siendo Sanic, que le siguió el aprendizaje conveniente para la correcta utilización de esta herramienta. Implementó los servicios necesarios para recibir los datos que vienen de la lógica de modelado para tratarlos, si es necesario, y enviarlos posteriormente a la parte que se ocupa de hacer visible nuestro trabajo, el frontend, satisfaciendo sus necesidades a la hora de formatear la salida de los servicios para que éste, pudiera realizar su tarea de manera cómoda y eficaz.

En cuanto a la aportación en la memoria, se encargó a todas las transcripciones al inglés íntegras. Los apartados que precisan esta tarea son: resumen, introducción y sus subsecciones motivación, objetivos y plan de trabajo, así como las conclusiones y los planes de trabajo futuro.

Contribuyó en pequeñas aportaciones al Capítulo 1 (Introducción) y a las tecnologías de desarrollo del Capítulo 4 (Implementación).

En el Capítulo 2 (Estado de la cuestión) realizó la recopilación y explicación de todas las tecnologías de aprendizaje automático, tanto las que se utilizan en este trabajo, como las que no.

También es autor de la Similitud por color dominante en el Capítulo 3 y de la sección Servicios REST del Capítulo 4.

Finalmente, escribió, dentro del Capítulo 5 (Conclusiones y trabajo futuro), la introducción al mismo y el apartado 2, que se refiere a los planes de trabajo futuro.

Marcos Rafael Núñez

En lo referente al desarrollo de la aplicación contribuyo tanto en la lógica del modelado, como en el frontend de la aplicación. Su participación ha estado presente en todas las fases de proyecto.

Durante el desarrollo de la lógica del modelado investigó distintitas formas posibles para comparar dos cuadros. Desarrolló íntegramente las similitudes parciales por tamaño del cuadro, así como la similitud parcial basada en el error cuadrático medio por píxel. De igual manera, investigó, decidió e implementó la forma de utilizar los gustos de los usuarios por los cuadros para

generar comunidades entre los usuarios. Decidió utilizar la similitud entre cuadros para generar una bolsa de cuadros para comparar dos usuarios.

Después de acordar grupalmente la manera de visualizar las comunidades generadas, implemento de manera íntegra toda la parte del frontend de la aplicación. Para ello ha tenido que hacer una primera etapa de estudio de los lenguajes utilizados ya que nunca los había usado. Para implementar la vista de la aplicación desarrolló varios ficheros HTML, así como distintas hojas de estilo y código en JavaScript. También ha utilizado librerías externas con las que ha podido crear una vista agradable y fácil de usar.

Para la generación de comunidades, se reunió con un profesor de Historia del arte de la Universidad Complutense de Madrid para determinar qué pesos dar a las distintas similitudes parciales de cuadros, así como posibles cambios para realizar como trabajo futuro.

Con respecto a la memoria ha participado en la mayoría de las secciones que la componen. Principalmente ha contribuido en las secciones referentes al modelado de comunidades, así como en la implementación.

Su aportación con respecto al Capítulo 1 - (Introducción) ha sido la elaboración parcial de todos los puntos que componen este capítulo.

Con respecto al Capítulo 3 - (Modelado de comunidades) su aportación ha sido el desarrollo completo de la sección 3.1 (Caso de estudio) así como las subsecciones 3.2.2.3, 3.2.2.4, 3.2.2.5 correspondientes a las similitudes parciales de las obras de arte basadas en el tamaño, artista y movimiento y error cuadrático medio por pixel. Por último, contribuyó parcialmente en la sección 3.3.4 (Discusión de resultados)

En el capítulo referente a la implementación redactó íntegramente la sección dedicada al frontend de la aplicación. Es decir, la sección 4.2.2.

Por último, participo en el desarrollo del Capítulo 5 - y de todas las subsecciones que lo componen.

Iago Zamorano Chouciño

En lo relativo al desarrollo de la aplicación contribuyó principalmente en la parte de la lógica del modelado de comunidades, participando en todas sus fases.

Inicialmente, se encargó de crear la herramienta de consultas SPARQL para poder realizar consultas con mayor facilidad y rapidez. Posteriormente, investigo la posibilidad de utilizar alguna librería que permitiera cachear la información tanto de las consultas como de otras operaciones algo costosas en tiempo como serían, a la postre, los cálculos de similitud entre los cuadros. Así pues, empezó a implementar dichas operaciones con el apoyo de la librería FileCache que proporciona un soporte de almacenamiento persistente para este tipo de casos.

A continuación, desarrolló íntegramente la similitud parcial por contenido entre cuadros, así como la similitud por artista y movimiento. También aportó con el desarrollo de los módulos de detección de comunidades en los cuales se llevan a cabo las técnicas de clustering. Fue el responsable del desarrollo completo del algoritmo genético para la búsqueda de pesos, así como de la propuesta de uso de dicho algoritmo.

Desarrollo también los módulos encargados de caracterizar los clusters para generar reportes con los distintos datos que los caracterizan y que estos puedan ser utilizados por el frontend a la hora de visualizarlos mediante la interfaz web. También creó una herramienta de visualización alternativa que genera dichos reportes exportándolos en un archivo pdf.

En cuanto a la memoria, se encargó de redactar diversas secciones. En concreto aquellas relacionadas con las fases del proceso de modelado de cuyo desarrollo se encargó. Por tanto, participó en gran medida en el Capítulo 3 (Modelado de comunidades), redactando todo aquello relacionado con la Similitud de usuarios, y el Proceso de modelado, así como la Similitud por contenido.

En el Capítulo 1 (Introducción), contribuyó en cierta medida en el desarrollo de todos sus puntos ya que este en general fue un apartado redactado en conjunto y revisado y corregido por todos los autores.

En el Capítulo 2 (Estado de la cuestión), se encargó de redactar el segundo apartado (Herramientas de Linked Data) íntegramente, así como de explicar el funcionamiento de las técnicas de clustering DBSCAN y k-medoids en el primer apartado (Técnicas de clustering).

Por último, en el Capítulo 4 (Implementación), fue el responsable de escribir acerca de la Arquitectura de la aplicación, detallando el funcionamiento de la Lógica del modelado. Además, contribuyó en la explicación de las Tecnologías y recursos externos utilizados.