Máster en Ingeniería de Sistemas y Control

TRABAJO DE FIN DE MÁSTER

APLICACIÓN DE TÉCNICAS DE MINERÍA DE TEXTOS EN INFORMES DE RADIODIAGNÓSTICO

Autor:

D. Juan Pueyo Candil

Directoras:

Dra. Raquel Dormido Canto

Dra. Natividad Duro Carralero

Curso Académico: 2014 – 2015

Convocatoria de Defensa: Septiembre 2015

Máster en Ingeniería de Sistemas y Control

TRABAJO DE FIN DE MÁSTER

APLICACIÓN DE TÉCNICAS DE MINERÍA DE TEXTOS EN INFORMES DE RADIODIAGNÓSTICO

PROYECTO TIPO A

Autor:
D. Juan Pueyo Candil

Directoras:
Dra. Raquel Dormido Canto
Dra. Natividad Duro Carralero

HOJA DE CALIFICACIONES

Autorización

Autorizo a la Universidad Complutense y a la Universidad Nacional de Educación a Distancia a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firmado: Juan Pueyo Candil

RESUMEN

En el presente documento se muestran los resultados del Trabajo Fin de Máster realizado. Este trabajo se centra en la aplicación de diversas técnicas de minería de textos en más de 500.000 informes radiológicos reales, cedidos por la Unidad Central de Radiodiagnóstico de la Comunidad de Madrid.

El objetivo principal de este proyecto es encontrar información relevante en dichos informes que no puede ser obtenida por los medios que actualmente existen en los hospitales. Para ello, se aplican técnicas de preprocesamiento de la información, de análisis semántico y de *clustering* documental. El trabajo realizado muestra los resultados favorables que se han obtenido en un caso real, encontrando mediante análisis semántico, el número de reacciones alérgicas a los contrastes que se han producido durante las pruebas radiológicas. También se muestran los resultados positivos de aplicar técnicas de *clustering,* con la finalidad de encontrar grupos de informes afines por el tipo de enfermedad. Por último, para la visualización de los resultados, se han desarrollado gráficos dinámicos e interactivos que facilitan la labor de validación de resultados por parte del centro.

Entre las conclusiones finales obtenidas, se puede destacar que, en minería de textos, mediante la aplicación de técnicas de análisis semántico, se pueden obtener resultados con una exactitud cercana al 100%, y que además, al aplicarlo junto con técnicas de *clustering*, permite obtener grupos con mayor precisión.

PALABRAS CLAVE

Análisis semántico, c*lustering* documental, informes de radiodiagnóstico, minería de datos, minería de textos, texto desestructurado, reacciones alérgicas, HTML5, JavaScript.

ÍNDICE

1. INTRODUCCIÓN 1
   1.1. MOTIVACIÓN Y ORIGEN 1
   1.2. OBJETIVOS 2
   1.3. PLANTEAMIENTO DEL PROBLEMA 3
   1.4. ORGANIZACIÓN DE LA MEMORIA 4
2. METODOLOGÍA Y CONDICIONANTES 5
   2.1. METODOLOGÍA 5
      2.1.1. Fases del Proyecto 5
   2.2. LIMITACIONES Y CONDICIONANTES 8
3. ESTADO DEL ARTE 11
   3.1. MINERÍA DE DATOS 12
      3.1.1. KNOWLEDGE DISCOVERY IN DATABASES 12
      3.1.2. TÉCNICAS DE MINERÍA DE DATOS 16
      3.1.3. MINERÍA DE TEXTOS 17
   3.2. APRENDIZAJE AUTOMÁTICO 21
      3.2.1. APRENDIZAJE NO SUPERVISADO 22
      3.2.2. AGRUPAMIENTO O CLUSTERING 23
      3.2.3. ALGORITMOS DE CLUSTERING 28
   3.3. R COMO HERRAMIENTA DE MINERÍA DE DATOS 33
   3.4. TECNOLOGÍAS WEB 37
      3.4.1. HTML5 37
      3.4.2. CSS3 38
      3.4.3. GRAFICOS VECTORIALES SVG 38
      3.4.4. JAVASCRIPT 41
   3.5. VISUALIZACIÓN DE DATOS MEDIANTE JAVASCRIPT Y HTML5 43
      3.5.1. APLICACIONES WEB PARA LA VISUALIZACIÓN 43
      3.5.2. API Y LIBRERÍAS PARA LA VISUALIZACIÓN WEB 48
      3.5.3. ARCHIVOS DE ALMACENAMIENTO JSON 53
      3.5.4. ATOM COMO EDITOR DE CÓDIGO FUENTE 54
4. DESARROLLO Y RESULTADOS DEL PROYECTO 55
   4.1. DATOS UTILIZADOS 55

4.2. CARGA DE INFORMES EN R 56
4.2.1. Tipos de datos 57
4.2.2. Campos agregados 57
4.2.3. Almacenamiento de los datos en un *Data Frame* 58
4.2.4. Eliminación de informes irrelevantes 59
4.2.5. Generación del Corpus 60
4.2.6. Creación de la Term Document Matrix 61
4.3. ANÁLISIS DE TÉRMINOS 63
4.3.1. Frecuencia de aparición de términos 63
4.3.2. Asociaciones de términos 66
4.4. CASO DE USO: REACCIONES ALÉRGICAS 68
4.4.1. Lenguaje utilizado en los informes 68
4.4.2. Proceso de obtención de informes con reacción alérgica 69
4.4.3. Eliminación de informes con reacción alérgica previa 72
4.4.4. Resumen del proceso 74
4.4.5. Obtención del contraste utilizado 75
4.4.6. Resultados derivados importantes 76
4.5. CLUSTERING DOCUMENTAL 78
4.5.1. Aplicación de algoritmos de *clustering* en informes de reacciones alérgicas 78
4.5.2. Aplicación de algoritmos de *clustering* en informes en general 88
4.5.3. Resultados derivados importantes 97
4.6. REPRESENTACIÓN DE DATOS 98
4.6.1. GRÁFICO DE DISTRIBUCIÓN DE PRUEBAS 98
4.6.2. GRÁFICOS DE BURBUJAS 104
4.6.3. Árbol jerárquico de clústeres 107
5. CONCLUSIONES Y TRABAJO FUTURO 110
5.1. Conclusiones 110
5.2. Trabajo futuro 112
6. LISTA DE REFERENCIAS Y BIBLIOGRAFÍA 114
7. LISTADO DE SIGLAS, ABREVIATURAS Y ACRÓNIMOS 119

LISTA DE FIGURAS Y TABLAS

FIGURAS:

Figura 3-1: fases del proceso de KDD 13
Figura 3-2: fases de un proceso de minería de textos 17
Figura 3-3: visión general del proceso de aprendizaje no supervisado con *clustering* 23
Figura 3-4: Imagen de una pitahaya o también conocida como fruta del dragón. Fuente: Wikipedia 25
Figura 3-5: Imagen de una garcinia mangostana o también conocida como mangostán. Fuente: Wikipedia 25
Figura 3-6: dendrograma de países por consumo de alimentos. Fuente [15]. 29
Figura 3-7: ejemplo del resultado de *k-means* con k = 3 31
Figura 3-8: imagen del entorno R Studio. 33
Figura 3-9: dendrograma de familias de paquetes de R. Fuente: r-statistics [22]. 35
Figura 3-10: gráfico SVG generado con Inkscape. Fuente [25]. 39
Figura 3-11: comparación de calidad entre SVG y mapa de bits. 40
Figura 3-12: porción de código para la representación de un tomate en SVG 40
Figura 3-13: vista de Madrid en 3D con Google Maps 41
Figura 3-14: Ejemplos de Google Fusion Tables. Fuente: [30]. 44
Figura 3-15: Manhattanhenge. Fuente: NYCHenge.com [32]. 45
Figura 3-16: reservas mundiales de petróleo con CartoDB. Fuente: Azavea Atlas [33]. 46
Figura 3-17: ejemplo de visualización con Tableau Public. Fuente: Tableau Software [34]. 47
Figura 3-18: nube de palabras interactiva creada con Many Eyes 47
Figura 3-19: gráfico de burbujas creado con Many Eyes 48
Figura 3-20: algunos ejemplos de gráficos de Google Chart Tools. Fuente: CTO Insights [37]. . 49
Figura 3-21: árbol de palabras interactivo. API Chart y JavaScript. Fuente: Google [34]. 49
Figura 3-22: algunos gráficos generados con FusionCharts. Fuente: FusionCharts [38]. 50
Figura 3-23: árbol de asociaciones con JavaScript Infovis Toolkit. Fuente: R-bloggers [39]. 51
Figura 3-24: árbol interactivo. Fuente: JavaScript Infovis Toolkit [40]. 51
Figura 3-25: ejemplo de gráfico jerárquico creado con D3. Fuente: D3.js [41]. 52
Figura 3-26: diagrama de Voronoi con los aeropuertos de Estados Unidos. Fuente: D3.js [41]. 53
Figura 3-27: ejemplo de código de un archivo JSON 54
Figura 4-1: diferencia entre los tiempos de proceso de los dos métodos 58
Figura 4-2: nube de palabras de un conjunto de informes 64
Figura 4-3: mapa de palabras de un conjunto de informes en los que todos contenían la palabra “mama” 65
Figura 4-4: gráfico de barras que muestra la frecuencia relativa 66
Figura 4-5: gráfico de relaciones entre términos con “reacción” y “alérgica” 67
Figura 4-6: gráfico de relaciones entre términos con “reacción” y “alérgica” 67
Figura 4-7: árbol de inferencia realizado con RWeka para el análisis de la palabra “refiere” 73

Figura 4-8: dendrograma obtenido con *clustering* jerárquico y distancia *manhattan* 80
Figura 4-9: gráfico de análisis del número de clústeres a buscar 81
Figura 4-10: resultado de aplicar el algoritmo *k-means* con 2 clústeres 81
Figura 4-11: relación entre elementos en cluster y exponente de pertenencia..... 83
Figura 4-12: gráfico de criterios para la creación de clústeres por el algoritmo Fuzzy C-Means 84
Figura 4-13: gráfico de calor devuelto por el algoritmo de propagación de afinidad 86
Figura 4-14: dendrograma obtenido con el *clustering* aglomerativo 87
Figura 4-15: dendrograma obtenido al aplicar el algoritmo de *clustering* jerárquico 88
Figura 4-16: zoom del dendrograma obtenido en el *clustering* izquierdo..... 89
Figura 4-17: cálculo del número óptimo de clústeres para el algoritmo *k-means* 90
Figura 4-18: cálculo del número de clústeres para el algoritmo *k-means* 91
Figura 4-19: gráfico generado por el algoritmo *k-means* con una distribución de 9 clústeres .. 92
Figura 4-20: gráfico generado por el algoritmo *k-means* con una distribución de 5 clústeres .. 92
Figura 4-21: gráfico generado por el algoritmo *k-means* con una distribución de 3 clústeres .. 93
Figura 4-22: distribución de términos con el algoritmo Fuzzy C-Means..... 96
Figura 4-23: ejemplo de datos de entrada para el gráfico BiPartite 99
Figura 4-24: código en R para el volcado de datos 100
Figura 4-25: ejemplo de creación dinámica del documento HTML 101
Figura 4-26: ejemplo del gráfico de pruebas por hospital 102
Figura 4-27 ejemplo del gráfico de pruebas por hospital 102
Figura 4-28: ejemplo del gráfico de pruebas por hospital 103
Figura 4-29: gráfico que muestra las diferentes agrupaciones creadas por los algoritmos de *clustering* 106
Figura 4-30: gráfico interactivo que permite hacer zoom en las diferentes agrupaciones 106
Figura 4-31: imagen completa del árbol mostrando un total de 50 informes..... 108
Figura 4-32: imagen del árbol en la que se observan algunos grupos cerrados..... 109
Figura 4-33: imagen del árbol en la que se observa la colocación manual de los elementos .. 109

TABLAS:

Tabla 2-1: primera parte del diagrama de Gantt del proyecto 6
Tabla 2-2: segunda parte del diagrama de Gantt del proyecto 7
Tabla 3-1: ejemplo de almacén de datos sin valores en la matriz 27
Tabla 3-2: ejemplo de almacén de datos con valores en la matriz 27
Tabla 3-3: compatibilidad de navegadores con algunas propiedades de CSS 38
Tabla 4-1: matriz de confusión del total de informes y las reacciones alérgicas encontradas ... 74
Tabla 4-2: tipos de contraste encontrados en los informes 76
Tabla 4-3: reacciones alérgicas distribuidas por tipo de prueba 76
Tabla 4-4: reacciones alérgicas distribuidas por hospital 77
Tabla 4-5: matriz de confusión de los 72 informes en el análisis semántico 79
Tabla 4-6: resultados obtenidos con los algoritmos de *clustering* jerárquico 79
Tabla 4-7: matriz de confusión al aplicar el algoritmo *k-means* 82
Tabla 4-8: matriz de confusión al aplicar el algoritmo Fuzzy C-Means 83
Tabla 4-9: distribución de elementos en los clústeres por el algoritmo de propagación de afinidad 85
Tabla 4-10: primeros caracteres de 10 de los informes agrupados por el *clustering* jerárquico 89
Tabla 4-11: distribución de elementos en cada grupo con el algoritmo *k-means* 94
Tabla 4-12: informes contenidos en el clúster 1 del resultado de 3 clústeres con *K-means* 94
Tabla 4-13: informes contenidos en el clúster 1 del resultado de 3 clústeres con *K-means* 95

1. INTRODUCCIÓN

1.1. MOTIVACIÓN Y ORIGEN

Actualmente nos encontramos en un momento en el que las tecnologías están influyendo en gran medida en la vida de las personas. El constante control y registro de las actividades y del rastro de datos que una persona va dejando a lo largo de su vida está provocando que esta inmensa cantidad de datos necesite ser tratada, no sólo desde el punto de vista económico, sino también desde otros muchos ámbitos como por ejemplo el de la salud, a modo de análisis y predicción.

Como decía en un artículo José Luis Sampedro, economista español Premio Nacional de las Letras Españolas en 2011, "los métodos de estudio deben adaptarse a la naturaleza del objeto estudiado"[1]. En dicho artículo, Sampedro expone la diferencia a la hora de desmontar y volver a montar un reloj, un gato y la República de Madagascar. Evidentemente, un reloj se puede desmontar y volver a montar, e incluso si se trata de repararlo podría volver a funcionar. En el caso de un gato, se podría diseccionar, pero no podría volver a la vida. Finalmente, en el caso de un país, no se puede hablar ni de desmontar. Con estos ejemplos, Sampedro, basándose en ideas también expuestas por Ramón Llull, explica en su artículo la existencia de tres grupos de objetos de estudio bien diferenciados: mecánico, biológico y social. Hoy en día no sólo existen esos grupos sino que gracias a la información que existe y que se almacena día a día, al continuo avance de la tecnología, y a la capacidad de cómputo con la que se puede trabajar gracias a sistemas que se encuentran "en la nube", se puede actuar sobre todos estos grupos y mejorar como sociedad. Es por ello que en este proyecto se lleva a cabo un análisis de informes de radiodiagnóstico con la finalidad de aplicar la ciencia y la tecnología para obtener resultados que puedan ayudar a la sociedad.

La idea de este proyecto surge inicialmente por el trabajo realizado en la asignatura de Minería de Datos del máster, lo que daba opción a profundizar en algoritmos de *clustering* sobre señales de fusión nuclear, pero a través de un contacto en la Unidad

Central de Radiodiagnóstico (UCR por sus siglas en español), surgió la idea de aplicar dichas técnicas de *clustering* a los informes radiológicos de la UCR. La UCR es una empresa pública de la Comunidad de Madrid creada en el 2008 con el objetivo de centralizar los estudios radiológicos de 6 hospitales de Madrid (Infanta Sofía, Infanta Cristina, Infanta Leonor, Henares, Sureste y Tajo) en un único punto, el Hospital Infanta Sofía. La UCR posee más de 4,5 millones de informes hasta la fecha [2], en los cuales existe información que la UCR es incapaz de extraer ya que se encuentra inmersa en los textos de los informes radiológicos redactados por los radiólogos que han realizado los estudios. A raíz de esto nace la idea de analizar los textos de estos informes para intentar obtener información que no se puede obtener por los medios que actualmente posee el centro de proceso de datos de la UCR, realizando análisis semántico o aplicando algoritmos de *clustering*.

1.2. OBJETIVOS

En este proyecto se han establecido cuatro objetivos principales: el primero de ellos consiste en demostrar que el análisis semántico en los informes de radiodiagnóstico puede aportar información de importancia que no se puede obtener por otros métodos. En concreto, se ha llevado a cabo un análisis semántico para buscar los pacientes que han sufrido una reacción alérgica al contraste utilizado durante alguna prueba radiológica.

El segundo objetivo es el de aplicar algoritmos de *clustering* a los informes de radiodiagnóstico de reacciones alérgicas y comparar los resultados de clasificación con los realizados en el análisis semántico, esperando que por ambos métodos se obtengan resultados similares.

El tercer objetivo, también relacionado con los algoritmos de *clustering*, es el de aplicar estos algoritmos a los informes de radiodiagnóstico en general para evaluar su validez en las agrupaciones creadas de cara a su uso posterior en otros estudios.

Finalmente se ha establecido un objetivo adicional, que es el de representar los resultados estadísticos y de *clustering* de una forma útil y al mismo tiempo atractiva utilizando las tecnologías de JavaScript y HTML5.

1.3. PLANTEAMIENTO DEL PROBLEMA

En el contexto clínico, la práctica del diagnóstico por imagen ha avanzado notablemente en los últimos años, consiguiendo importantes avances en el campo del almacenamiento digital de imágenes y de informes radiológicos. Sin embargo, en la mayoría de los casos, esta información se almacena sin ningún tipo de estructura ni en base a un estándar que sea mundialmente aceptado. En este punto entra en juego la Unidad Central de Radiodiagnóstico de la Comunidad de Madrid. La UCR se crea en 2008 con el objetivo de crear una red de comunicación entre 6 hospitales de Madrid: Hospital Infanta Sofía (Hospital del Norte o San Sebastián de los Reyes), Hospital Infanta Cristina (Hospital de Parla), Hospital Infanta Leonor (Hospital de Vallecas), Hospital del Henares (Hospital de Coslada), Hospital del Sureste (Hospital de Arganda del Rey) y Hospital del Tajo (Hospital de Aranjuez). El objetivo principal de crear esta red es el de permitir que un paciente que haya sido atendido en uno de estos hospitales pueda ser atendido en otro y el radiólogo disponga del historial clínico del paciente, reduciendo así los tiempos de espera de algunos centros y mejorando la atención del servicio de urgencias de estos hospitales.

Actualmente la UCR dispone de más de 4,5 millones de informes de radiodiagnóstico almacenados de forma digital en su base de datos, los cuales han sido elaborados por el personal de radiología de estos hospitales en sus respectivos centros. Estos informes están formados por diferentes campos, entre los que se incluye la información clínica y el propio informe médico, el cual posee información que sería de carácter relevante para la UCR pero que al tratarse de un texto elaborado por el personal de radiología y sin seguir un estándar establecido, se trata de un claro caso de elevados volúmenes de información, completamente desestructurados, que además pueden incluir información de otros departamentos hospitalarios e incluso fuentes públicas (datos

demográficos, epidemiológicos, etc.). Analizando esta situación, surge la necesidad de llevar a cabo estudios sobre los informes que puedan devolver resultados significativos, como el número de pacientes que han sufrido alguna enfermedad o por ejemplo, el de crear agrupaciones de pacientes en base a enfermedades sufridas y analizar el porcentaje de posibilidad de pertenencia de otros pacientes a dichos grupos y por tanto de prever una posible futura enfermedad.

1.4. ORGANIZACIÓN DE LA MEMORIA

El presente documento está dividido en cinco capítulos: Introducción, Metodología, Estado del Arte, Desarrollo y Conclusiones.

El primero de ellos es una introducción al trabajo realizado durante todo el tiempo de realización del proyecto.

En el segundo de los apartados se analiza la metodología seguida en el proyecto y los condicionantes que han influido en el desarrollo del proyecto.

En el tercer capítulo se presenta el estado del arte actual, haciendo hincapié en los algoritmos de *clustering* y en la representación de gráficos mediante JavaScript y HTML5.

En el cuarto capítulo se expone el desarrollo de todo el proyecto, comenzando por explicar el origen de los datos y el tratamiento previo que se ha realizado sobre los mismos. Seguidamente, se explica en detalle cada uno de los apartados desarrollados, comenzando por un caso de uso en concreto, como es la búsqueda de pacientes con reacción alérgica durante las pruebas de radiodiagnóstico. Posteriormente se explica el trabajo realizado sobre *clustering* documental y finalmente se detalla el desarrollo de gráficos realizado bajo HTML5 y JavaScript.

En el quinto apartado se muestran las conclusiones derivadas de la realización del proyecto, así como los objetivos alcanzados y las ideas para trabajo futuro.

2. METODOLOGÍA Y CONDICIONANTES

2.1. METODOLOGÍA

El desarrollo de este proyecto se ha llevado a cabo bajo un proceso incremental para poder afrontar todas las tareas que se querían realizar. Al mismo tiempo se ha empleado una metodología iterativa en cada una de las tareas ya que tienen una base común pero que según aumenta la complejidad, adoptan un desarrollo independiente y por ello era necesario aplicar el proceso incremental. En el caso del análisis de las reacciones alérgicas y en el estudio de *clustering* de textos, la base de carga y tratamiento previo de la información es común en ambos casos. Para el apartado de presentación gráfica también tiene como base esta información por lo que se puede decir que la primera fase del proyecto se diseñó para que fuera válida en el resto de casos. El motivo de elegir una metodología iterativa fue debido al desconocimiento de los resultados que se podían obtener y que influían notablemente en el desarrollo del proyecto y en la consecución de las tareas fijadas.

2.1.1. Fases del Proyecto

El proyecto se ha realizado en 5 fases que se detallan a continuación:

- **Fase 1**: en esta primera fase se realizó una investigación sobre el estado del arte en minería de textos aplicados a informes de radiodiagnóstico, así como también en minería de textos en general. Después se realizó una investigación sobre el estado del arte de los algoritmos de *clustering* aplicados a textos desestructurados y de las posibles técnicas a emplear.
- **Fase 2**: en la segunda fase se llevó a cabo una investigación sobre herramientas a utilizar y equipo informático necesario. Fue necesario realizar pruebas con la herramienta R con textos de prueba descargados de Internet [3] para analizar con qué volumen de información se podían obtener respuestas en un tiempo reducido.

- **Fase 3**: esta tercera fase se centró en los métodos de carga de los informes de radiodiagnóstico en R, así como su tratamiento posterior. Este proceso se explica en detalle en el apartado 4.2.
- **Fase 4**: durante esta fase se realizaron estudios estadísticos que luego sirvieron de base para su representación en la Fase 8. Se explica en el apartado 4.3.
- **Fase 5**: esta fase correspondió a la investigación sobre la obtención de resultados de las reacciones alérgicas producidas durante la realización de las pruebas de radiodiagnóstico. Se detalla en el apartado 4.4.
- **Fase 6**: fase en la que se trabajó con los algoritmos de *clustering* aplicados a textos. Los estudios realizados se muestran en el apartado 4.5
- **Fase 7**: en esta fase se llevó a cabo el estudio del estado del arte de la representación gráfica mediante JavaScript y HTML5.
- **Fase 8**: durante esta fase se desarrollaron las herramientas gráficas utilizadas en el proyecto. Su desarrollo se detalla en el apartado 4.6.
- **Fase 9**: elaboración de la memoria.

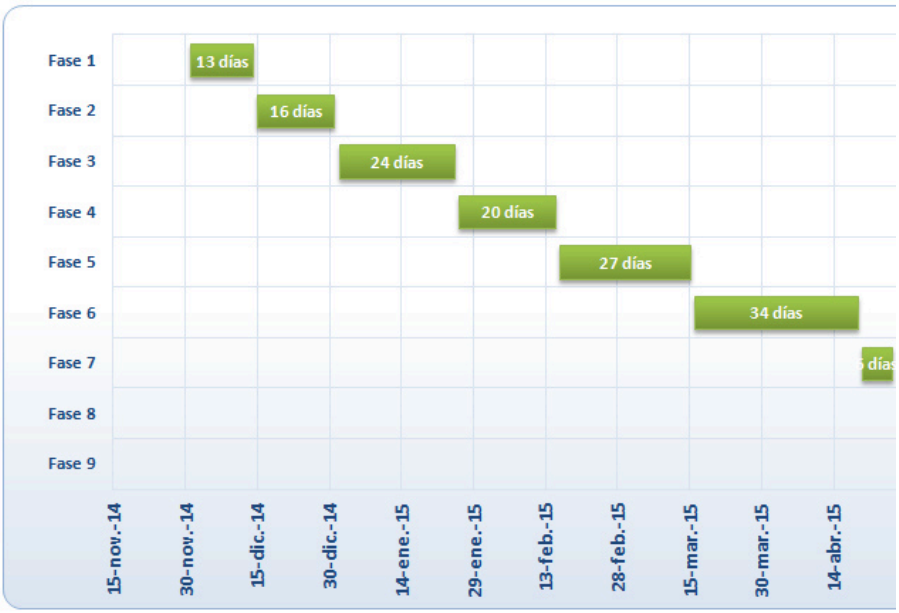


Tabla 2-1: primera parte del diagrama de Gantt del proyecto

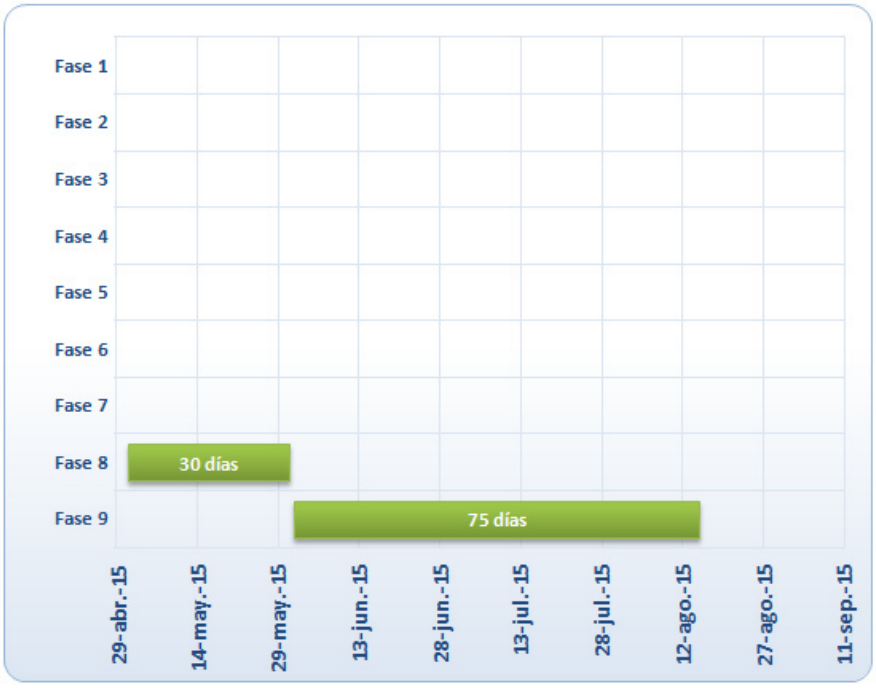


Tabla 2-2: segunda parte del diagrama de Gantt del proyecto

2.2. LIMITACIONES Y CONDICIONANTES

Como ya se comentó anteriormente, se eligió R como herramienta para el desarrollo de la parte de minería de textos y se probó durante la fase 2 del proyecto con diversos textos médicos descargados de Internet. Durante las pruebas realizadas se optó por utilizar equipos que al menos fueran Intel Core i5 con 8 GB de memoria RAM. Para la mayor parte de cálculos de tiempos y de obtención de resultados se empleó un equipo con las siguientes características principales:

- **Procesador**: Intel Core I5-4690K 3,50GHz. Es un procesador de cuarta generación con 4 núcleos y de los que mejor rendimiento ofrecen dentro de la gama Intel Core I5 actuales, aceptando velocidades de memoria RAM superiores a los 3 GHz y permitiendo incluso *overclocking* y poder llegar a 3,90GHz sin calentarse en exceso. Posee una memoria caché de 6MB.
- **Memoria RAM**: 8GB 2,400 GHz con disipadores de calor.
- **Unidad SSD**: Crucial MX100 256GB SSD SATA3. El SSD (del inglés *Solid State Drive*) puede llegar a alcanzar los 550 MB/s de velocidad en lectura y los 330 MB/s en escritura, frente a los 90 MB/s aproximados de un disco duro.
- **Placa Base**: Asrock Z97 Extreme4. Diseñada específicamente para ofrecer un alto rendimiento.
- **Sistema Operativo**: Windows 8.1.
- **Software**:
    - R versión 3.2.1
    - R Studio versión 0.99.467
    - Atom versión 1.0.3

Aun habiendo elegido este equipo, ciertas tareas requerían un tiempo de proceso elevado. Por ejemplo, después de haber depurado y optimizado al máximo toda la carga de informes de radiodiagnóstico desde el archivo original a la memoria RAM,

para volcar los datos se tardaba un tiempo superior a 1 minuto. En equipos más básicos, sin SSD, con un disco duro de 7.200 revoluciones por minuto, con placas base no orientadas a rendimiento y con procesadores Intel Core I3 de primera generación, el tiempo podía llegar a triplicarse o cuadruplicarse.

En el equipo descrito se han podido realizar la mayoría de tareas planificadas, aunque en un tiempo elevado. Por ejemplo crear un corpus a partir de todos los datos de los que se disponía, podía llevar 1 hora de proceso. En el caso de los algoritmos de *clustering* es donde ha habido que decantarse por trabajar sólo con grupos de informes, como por ejemplo los de reacciones alérgicas o con los de intervalos concretos de fechas, ya que realizar un clúster con el algoritmo jerárquico con 50.000 informes podía llevar más de 24 horas. Es evidente que para este tipo de tareas hay que escalar a campos como el de *Big Data*.

Tal y como se comenta más en detalle en el apartado 3.3 de este documento, R es un lenguaje interpretado que gracias a RStudio se pueden ejecutar las líneas que se deseen y en el orden que interese. RStudio además permite poder almacenar en disco el entorno de trabajo, es decir, que se puede almacenar en el disco duro el contenido de las variables y de las estructuras de datos que se encuentran en memoria RAM, lo cual agiliza mucho el trabajo. Por ejemplo, si crear un corpus lleva una hora, después de ese proceso se puede almacenar el corpus creado en disco. Recuperarlo posteriormente lleva tan solo unas decenas de segundos y reduce en gran cantidad el tiempo total de proceso.

Otro de los condicionantes del proyecto ha sido la imposibilidad de tener los datos accesibles en todo momento. En un principio la UCR cedió un conjunto de unos 1.000 informes radiológicos que permitió ir realizando algunas pruebas. Posteriormente se firmó un acuerdo de confidencialidad con la Comunidad de Madrid y la UCR, lo que permitió que la UCR volcara todos sus informes correspondientes al año 2013 y después de un proceso para anonimizar los datos, eliminando nombres, apellidos, direcciones, etc., se cedieran para poder trabajar con ellos. Desgraciadamente las elecciones del 24 de marzo de 2015 llevaron a un cambio de Gobierno en la

Comunidad de Madrid lo que no solo paralizó la posible cesión de datos de otros años, sino que además hubo que devolver los que se tenían del año 2013, aunque al menos se ofreció la posibilidad de realizar pruebas con dichos datos en el centro de control que la UCR posee en el Hospital Santa Sofía.

3. ESTADO DEL ARTE

En la actualidad nos encontramos en la revolución del tratamiento masivo de datos, o también conocido como *Big Data*. Estamos ante un momento clave en el desarrollo y la aplicación de tecnologías informáticas y de comunicaciones. La sociedad ha pasado en menos de 40 años por la aparición del ordenador personal, el nacimiento de las redes, y por supuesto de la red Internet y su World-Wide Web, el movimiento de las redes sociales, y actualmente, por la importancia de todos y cada uno de los datos existentes. Antiguamente, cuando los dispositivos de almacenamiento externo tenían un coste elevado, era clave decidir si algún dato se almacenaba o se descartaba y se perdía para siempre. Hoy sin embargo cualquier dato tiene valor, y si se trata de datos antiguos, van adquiriendo valor según pasa el tiempo. Esta enorme avalancha de datos provenientes de smartphones, tabletas, sistemas de posicionamiento (*Global Positioning System* o GPS por sus siglas en inglés), pulseras de actividad, globos sonda, satélites, etc., están generando la necesidad de adaptar tanto los sistemas de procesamiento como los de almacenamiento y de comunicación. Por ejemplo Facebook genera al día 600 Terabytes de información, alcanzando actualmente un volumen total de 300 Petabytes, lo que le acerca cada vez más a almacenar su primer Exabyte [4]. Acumular tales cantidades de información y hacerlas accesibles al mismo tiempo para todo el mundo requiere una importante inversión en investigación tecnológica, de ahí que Google, Twitter, Facebook o cualquier empresa que maneje estos volúmenes de datos se acaba convirtiendo por propia necesidad en pionero tecnológico.

Dejando a un lado los sistemas de almacenamiento y comunicaciones, hay que incidir en la importancia del tratamiento de estos datos, es por ello que la minería de datos, la minería de textos y el aprendizaje automático son áreas que cobran vital importancia.

3.1. MINERÍA DE DATOS

Existen diferentes definiciones sobre lo que es minería de datos (*data mining* en inglés). Según Han and Kamber[5], es el proceso para descubrir información relevante a partir de grandes cantidades de datos. Es un campo interdisciplinar que contribuye en multitud de áreas, tales como estadística, aprendizaje automático, reconocimiento de patrones, bioinformática, etc. y que es empleado en un gran número de ámbitos como finanzas, telecomunicaciones, medicina y redes sociales.

Cada día son más las empresas que solicitan diversos servicios de consultoría para poder obtener información valiosa para su negocio dentro de sus propias bases de datos. Son conscientes del valor de la información que poseen, pero desconocen cómo sacarle provecho. A esto se le conoce como "conocimiento fundamental" (*know-how* en inglés), es decir, conocimientos técnicos sobre la realización y consecución de objetivos pasados para aprender de ellos y poder predecir lo que puede ocurrir en el futuro para hacerle frente. Poseer la infraestructura de almacenamiento y comunicación necesaria, así como sistemas que automaticen el proceso de obtención de dicha información, son objetivos que empiezan a estar a la orden del día para muchas empresas. La minería de datos y la minería de textos entran en juego en este punto, donde se necesita obtener información importante a partir de una cantidad ingente de datos, realizarlo de forma automática, y transformar esa información en conocimiento.

3.1.1. KNOWLEDGE DISCOVERY IN DATABASES

El proceso conocido como *Knowledge Discovery in Databases* (KDD por sus siglas en inglés), es un proceso que consiste en la extracción de conocimiento a partir de bases de datos. Forma parte del proceso más general conocido como *Knowledge Discovery* (KD por sus siglas en inglés). La minería de datos es un subproceso dentro del KDD en el que se emplean diferentes técnicas, como la inteligencia artificial o el aprendizaje automático, para obtener una estructura de datos que facilite su

tratamiento y que permita identificar patrones que puedan ser interesantes y que eran desconocidos a priori.

En los últimos años el proceso KDD ha ido cobrando mucha importancia pero por razones de marketing en vez utilizar su nombre se ha utilizado el de Minería de Datos, lo que ha provocado que haya una confusión con estos términos y se piense que la minería de datos hace referencia al proceso completo, pero en realidad, minería de datos es una parte del todo el proceso.

Existen diversas opiniones sobre las fases que componen todo el proceso KDD. Según Hernández Orallo [6], las fases son las siguientes:

1. Buscar y analizar posibles fuentes de información.
2. Crear un almacén de datos capaz de almacenar toda la información obtenida para que pueda ser tratada.
3. Crear un sistema que permita la visualización de datos para analizar qué puede ser interesante como objeto de estudio.
4. Realizar una selección de los datos realmente válidos.
5. Seleccionar y aplicar una técnica de minería de datos apropiada.
6. Representar los patrones extraídos para su interpretación y evaluación.
7. Difundir el nuevo conocimiento.



Figura 3-1: fases del proceso de KDD

En la figura 3-1 se muestran las fases más importantes de todo el proceso KDD, las cuales se detallan a continuación:

3.1.1.1. Fase de búsqueda de información

Esta primera fase es de gran importancia a la hora de poder obtener conocimiento válido en las siguientes fases, ya que la información se puede obtener de muy diversas fuentes y con formatos muy diferentes. En un principio esta fase no se debe automatizar para poder ir analizando si se dispone de información suficiente o es necesario ir buscando otras fuentes que completen la información inicial. Esta etapa también comprende el proceso para almacenar la información en un sistema que permita el tratamiento de los datos, así como el almacenamiento en estructuras de datos que faciliten la obtención de conocimiento.

Los detalles del proceso seguido con los datos utilizados en este proyecto se detallan en los apartados 4.1 y 4.2.

3.1.1.2. Fase de preparación de datos

Una vez obtenida la información y estar almacenada en estructuras de datos, en necesario realizar una criba de datos, intentando descartar todos aquellos que no sean de interés para permitir que la extracción de conocimiento sea más sencilla y devuelva resultados más cercanos a los esperados, así como también para acelerar el proceso evitando procesar datos irrelevantes.

Por un lado se trata de eliminar aquellas tuplas erróneas, repetidas o que se salgan de los parámetros establecidos, como por ejemplo por no estar dentro de un rango de fechas determinado. Por otro lado hay que llevar a cabo un análisis de los atributos de los datos, eliminando aquellos que carezcan de interés, resumiendo algunos atributos, creando nuevos atributos a partir de otros extrayendo la información que se necesita, etc.

Para no ralentizar los procesos siguientes, es importante que todo el tratamiento de los datos se realice en esta fase y se dejen preparados para la siguiente fase.

3.1.1.3. Fase de minería de datos

En esta fase se trata de decidir qué tipo de conocimiento se desea extraer, ya que dependiendo del tipo de patrón a descubrir, se deben utilizar unas técnicas de minería de datos u otras. Puede darse el caso en el que no se sepa exactamente qué es lo que se busca, por lo que en esta situación, hay que aplicar diversas técnicas analizando los datos obtenidos aplicando metodologías iterativas.

Por otro lado la fuente de información es importante a la hora de elegir una técnica de minería de datos, ya que hay técnicas cuya efectividad depende en gran medida del volumen de datos original o de si hay atributos vacíos en algunas tuplas.

Las técnicas de minería de datos que se emplean para la obtención de conocimiento se pueden dividir en dos grupos:

- **Técnicas de verificación**: son técnicas iterativas en la que se parte de una hipótesis sobre la existencia de una información de interés. El sistema trata de comprobar esa hipótesis y se analizan los resultados devueltos por el sistema. Si estos no son los esperados, se redefine la hipótesis inicial y se vuelve a iniciar el proceso.
- **Técnicas de descubrimiento**: estas técnicas tratan de descubrir de forma automática patrones que puedan ser de interés. Dependiendo de la información de origen, se pueden aplicar unas u otras.

3.1.1.4. Fase de evaluación y validación

En la fase anterior se generan diferentes modelos que necesitan ser validados para lo cual se emplean criterios de evaluación de hipótesis. En el caso de las técnicas de descubrimiento, se emplean conjuntos de datos de entrenamiento y diferentes

técnicas para elegir los datos que sirven de entrenamiento en los casos en los que estos sean escasos. Para evaluar la precisión y la exactitud de los resultados se elabora una matriz de resultados, donde se anotan los verdaderos positivos y negativos así como los falsos positivos y negativos. La exactitud de los resultados se obtiene dividiendo la suma de todos los resultados verdaderos entre todos los resultados negativos. La precisión se calcula dividiendo los verdaderos positivos entre la suma de los falsos positivos y negativos.

3.1.2. TÉCNICAS DE MINERÍA DE DATOS

En cuanto a las técnicas empleadas en minería de datos, existen bastantes diferencias entre unos autores y otros, pero en general se pueden indicar las siguientes:

- **Análisis de secuencias**: consiste en encontrar patrones que siguen una secuencia concreta, del tipo "si se da *p* entonces se da *q*".
- **Clasificación**: consiste en la creación de grupos en base al valor que obtiene un atributo dentro de todos los posibles valores, los cuales son conocidos.
- ***Clustering***: trata de encontrar grupos igual que en la clasificación, pero en este caso se desconocen los grupos que se pueden formar, las clases y los posibles valores de los atributos.
- **Predicción**: busca encontrar el posible valor futuro de diferentes atributos en función de los valores pasados.
- **Reconocimiento de patrones**: trata de encontrar patrones en los que el valor de un atributo viene originado por el valor de otros, del tipo "si existe *p* y *q* entonces existe *r*".
- **Reglas de asociación**: se considera que existe una asociación entre dos atributos cuando la frecuencia de que tengan el mismo valor en diferentes tuplas se da repetidamente.

3.1.3. MINERÍA DE TEXTOS

Se considera minería de textos a una parte de la minería de datos en que se analizan textos escritos con lenguaje natural, con la intención de extraer información útil donde puede parecer que no existe [7]. La principal diferencia entre la minería de textos y la minería de datos es que en minería de datos se analiza información estructurada, mientras que en minería de textos se analiza texto desestructurado y almacenado en bases de datos sin ningún criterio. Creando una analogía con el proceso KDD, algunos autores nombran a la minería de textos como el proceso *Knowledge Discovery in Text* (KDT por sus siglas en inglés). Según Ah-Hwee Tan [8], el 80% de la información que poseen las empresas en sus bases de datos se encuentra en documentos de texto, por lo que la importancia de la minería de textos a nivel comercial es todavía mayor que la de la minería de datos.

Según Tan [8], en todo proceso de minería de textos hay que pasar por una serie de fases para poder obtener conocimiento de la fuente de texto original.

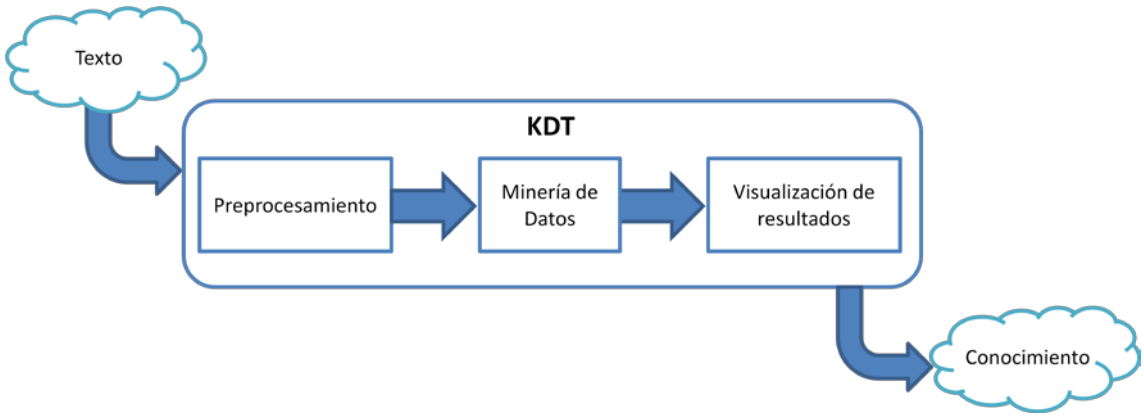


Figura 3-2: fases de un proceso de minería de textos

Como se muestra en la figura 3-2, las fases de un proceso de minería de textos son tres. Estas fases se detallan en los siguientes apartados.

3.1.3.1. Fase de preprocesamiento

Se trata de una etapa imprescindible en minería de textos ya que en ella se realizan transformaciones en el texto para permitir aplicar posteriormente las técnicas propias de minería de textos. Algunas de las transformaciones que se suelen realizar en esta fase son:

- Eliminación de signos de puntuación.
- Eliminación de espacios en blanco innecesarios.
- Supresión de palabras que carecen de significado como son artículos, preposiciones, algunos adjetivos, algunos adverbios, etc.
- Lematización de palabras, más conocido por su término en inglés, *stemming*, que consiste en sustituir todas las palabras de la misma familia por una única palabra de esta familia. Este proceso reduce en gran medida las diferentes palabras existentes lo que aumenta el rendimiento de las técnicas de minería de textos, sin embargo, en algunos casos, su aplicación puede generar resultados equívocos, por lo que algunos autores no están a favor de su empleo.
- Según Etxeberría [9], en los textos se encuentran lo que él denomina "segmentos repetidos". Es decir, conjuntos de palabras que unidas tienen un significado especial y que por tanto no se pueden separar porque perderían su significado. Es en esta fase de preprocesamiento cuando se deben encontrar tales "segmentos repetidos" para aislarlos y evitar su separación.

Ah-Hwee Tan denomina a las estructuras obtenidas en este proceso como "formas intermedias" de texto, y las clasifica en dos tipos:

- **Estructurada**: cuando los datos son representados siguiendo el modelo relacional.

- **Semiestructurada**: cuando los datos se forman utilizando grafos conceptuales. Estos grafos pueden representar conceptos y las relaciones entre ellos, o también pueden representar documentos y las relaciones entre los mismos.

Algunas de las técnicas que se aplican en esta fase son:

- Análisis de texto.
- Búsqueda de clases o categorías.
- Técnicas de extracción de información, como extracción de palabras o de partes concretas del texto.
- Técnicas de procesamiento de lenguaje natural (*Natural Processing Language* o NLP por sus siglas en inglés), como etiquetado de términos o la lematización.

3.1.3.2. Fase de minería de texto

En esta fase se analizan las "formas intermedias" con el objetivo de encontrar patrones que son desconocidos al comienzo del proceso, utilizando diversas técnicas de forma individual o en conjunto. Estas técnicas se dividen en dos grupos, descriptivas y predictivas. Algunas de estas técnicas son [10]:

- **Técnicas descriptivas**:
  - Visualización de documentos.
  - Extracción de características inmersas en el propio texto, como puede ser el de diferenciar cualquier clase de elemento, como personas, lugares, etc.
  - Aprendizaje automático no supervisado, donde se encuentra principalmente la técnica de *Clustering* documental, que se comenta en detalle en el punto 3.2.2 de este documento y que da forma a una parte importante de este proyecto.
  - Reglas de asociación, en donde se buscan las asociaciones entre términos que más se suceden a lo largo de los documentos. En la figura 4-5 se puede

ver un ejemplo de reglas de asociación entre diferentes términos que se han analizado en este proyecto.

- **Técnicas Predictivas**:
  - Clasificación, utilizada para clasificar elementos en categorías predefinidas. En este proyecto no se ha realizado este paso por salirse del alcance del mismo pero sería interesante poder catalogar a los pacientes en grupos de riesgo para prevenir enfermedades.
  - Resumen de textos, que consiste en reducir el texto a la parte fundamental, donde supuestamente se encuentra la información relevante. En este proyecto este paso se ha realizado principalmente en el caso de uso sobre la búsqueda de reacciones alérgicas.

En el punto 4 de este documento, se mostrará cómo se han llevado a cabo estas dos fases en el tratamiento de los informes de radiodiagnóstico, y en concreto en la extracción del número de reacciones alérgicas producidas en los pacientes durante la prueba.

3.1.3.3. Fase de visualización de resultados

En esta última fase se representan los resultados de la forma más sencilla posible para el usuario pero sin ocultar información para que ésta pueda ser validada. Actualmente los métodos más utilizados son la representación de gráficos dinámicos a través de un navegador web, ya que permiten una potencia gráfica y de representación que no poseen otras herramientas. Una parte de este proyecto ha consistido en el desarrollo de gráficos para mostrar resultados estadísticos y de *clustering* en un entorno web. Este trabajo se detalla en el apartado 4.6 de este documento.

3.2. APRENDIZAJE AUTOMÁTICO

El aprendizaje automático (*machine learning* en inglés) es un conjunto de algoritmos y técnicas de inteligencia artificial que tienen como objetivo alcanzar un aprendizaje por inducción a través de ejemplos de la vida real. Trata de permitir la autonomía de un sistema para su automejora continua con la menor intervención humana posible. Está inmerso en multitud de ámbitos de la sociedad y se ha convertido en una importante área de investigación en los últimos años, principalmente en campos como el estudio económico, reconocimiento de patrones, minería de datos o estadística entre otros. Dentro del aprendizaje automático hay que distinguir varios tipos de aprendizaje, de los cuales los más empleados son:

- **Aprendizaje supervisado**: de forma resumida, se puede decir que este modelo de aprendizaje se basa en un entrenamiento de datos, es decir, se trata de construir un algoritmo que, a partir de entradas de datos que se sabe qué salidas debe devolver, devuelva dichas salidas mediante modificaciones o ajustes de los parámetros de las fórmulas del algoritmo. Dentro de este tipo de aprendizaje existen diversos tipos y reglas a tener en cuenta. Un ejemplo de este tipo de algoritmo sería un sistema de reconocimiento automático de razas de personas en fotografías y el entrenamiento podrían ser imágenes de personas de las cuales se conoce su raza.

- **Aprendizaje no supervisado**: este aprendizaje es el que realmente nos interesa en este proyecto ya que las técnicas de *clustering* pertenecen a esta clase de aprendizaje, el cual parte de la idea de que no se conocen las salidas que el algoritmo tiene que devolver. Dicho de otra manera, estos algoritmos buscan agrupar la información en base a multitud de criterios diferentes, independientes o combinados, intentado mostrar resultados que a priori eran desconocidos.

- **Aprendizaje semisupervisado**: es una combinación de los dos tipos de aprendizaje ya comentados.

3.2.1. APRENDIZAJE NO SUPERVISADO

Como ya se ha comentado, el aprendizaje no supervisado se aplica cuando se desconoce la salida que se puede obtener a partir de unos datos de entrada que en principio están desestructurados o desagrupados.

Según [11], este tipo de aprendizaje puede ser interesante en:

- Agrupación de datos desagrupados siguiendo algún criterio predefinido.
- Generación de jerarquías para organizar los datos a partir de datos en un mismo nivel.
- Reducción de la dimensión de la información, reduciendo el número de atributos que caracterizan a dichos datos.
- Representación gráfica de datos que a priori, dada su complejidad, no pueden ser representados.

En la figura 3-3 se puede ver un proceso típico de aprendizaje no supervisado en el que se parte de un conjunto de datos desagrupados y posiblemente desestructurados (por lo que en esta fase puede existir la posibilidad de que los datos tengan que ser tratados para poder ser utilizados en la fase siguiente), en la cual el algoritmo elegido crea una serie de grupos o clústeres en base a diferentes criterios que se le pueden indicar al algoritmo, aunque no todos los algoritmos lo admiten y no contemplan dichas opciones. Una vez creados los clústeres, estos deben ser revisados por personal humano para su revisión, evaluando la eficacia de dicho algoritmo. Es por ello que en cualquier ámbito en el que se utilice este tipo de aprendizaje, lo normal es utilizar diferentes algoritmos, probar las salidas que devuelven en función de las entradas, y elegir el que se considere más apropiado en cada caso. Puede que esto de lugar a confusión con el aprendizaje supervisado, sin embargo, esta característica que los diferencia, se refiere al tipo de entrenamiento del algoritmo, no a la revisión final que hay que realizar y que es la misma en los dos casos.

Además del agrupamiento o *clustering*, el cual se detallará en el siguiente apartado, existe otro tipo de aprendizaje no supervisado que son las redes neuronales no

supervisadas. Dado que estos sistemas de cálculo no se van a emplear en este proyecto, simplemente indicar que una red neuronal supervisada necesita de la intervención humana para modificar los parámetros de sus funciones matemáticas, mientras que las redes neuronales no supervisadas lo hacen de forma autónoma siguiendo diferentes técnicas, como por ejemplo “ensayo y error”.

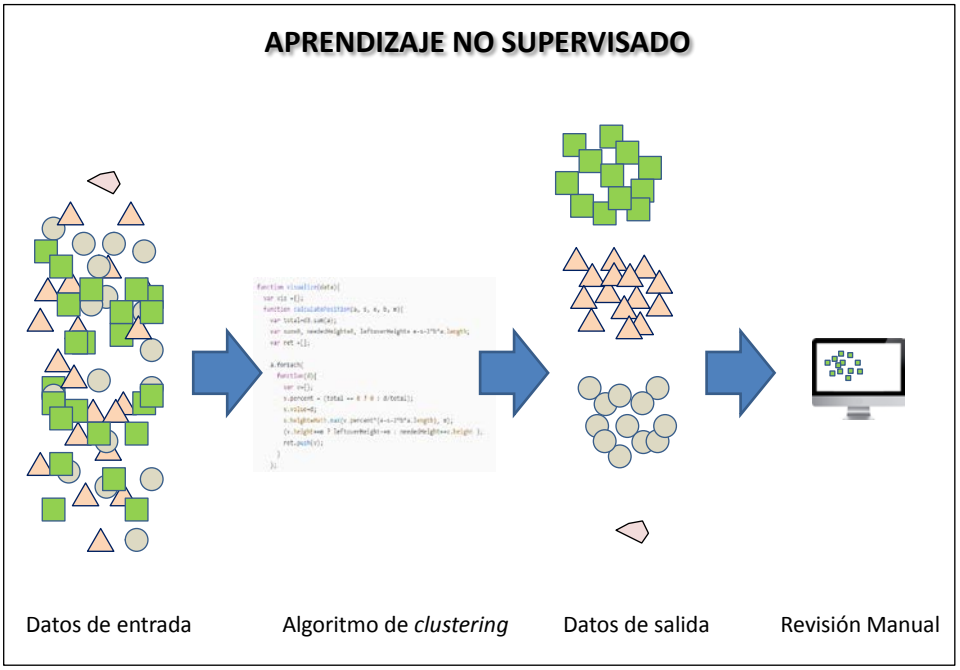


Figura 3-3: visión general del proceso de aprendizaje no supervisado con *clustering*

3.2.2. AGRUPAMIENTO O CLUSTERING

Desde el comienzo de los tiempos el ser humano ha utilizado la clasificación como método para identificar las cosas que nos rodean, identificándolas y clasificándolas gracias a los sentidos que posee. Después, gracias al estudio, a la ciencia y a la tecnología, muchas de esas clasificaciones han sido declaradas como erróneas o han tenido que volverse a definir, pero hasta ese momento, esa agrupación ha servido como el principal método de identificación. Un claro ejemplo de esto es la diferencia entre fruta, verdura e infrutescencias, un error de clasificación que es latente en la actualidad y que ha dado lugar a diferentes creencias populares [12], como que una

fruta es todo aquello comestible que se obtiene de los árboles, aunque esta definición se olvide de frutas tan conocidas como el melón y la sandía. Otra creencia hace referencia a su dulzura, como que la fruta es todo lo que ofrecen los árboles y las plantas y que tiene carácter dulce, pero en este caso los cítricos como la lima y el limón son los que se quedan fuera. Si se hace referencia a su carácter gastronómico, hay creencias populares que clasifican la fruta como todo aquello que suele tener un sabor dulce y que se come en el postre o entre horas, sin embargo, no deja de ser una creencia popular. Hay gente que simplemente no sabe definir qué es una fruta y solo es capaz de poner ejemplos de aquello que conoce como fruta. Con la verdura ocurre lo mismo, mucha gente cree que es todo aquello comestible que proviene de la tierra y no de árboles, cuando realmente no es así.

Teniendo en cuenta una definición que se da a la fruta, “producto del desarrollo del ovario de una flor después de la fecundación. En él quedan contenidas las semillas" [13], hay ejemplos de frutas como el tomate, el pepino, los pimientos, los jalapeños o la calabaza que son frutas, y ni se obtienen de ningún árbol, ni tienen por qué ser dulces. Si en un principio la clasificación de fruta se hubiera realizando por tener semillas en su interior, puede que hoy en día no existiera ese error, que incluso se puede encontrar en el Diccionario de la lengua española, donde define fruta como "fruto comestible de ciertas plantas cultivadas; p. ej. la pera, la guinda, la fresa, etc." [13]. y la fresa en realidad no es una fruta si no una infrutescencia, al igual que el higo y la piña [14]. En diversas ocasiones algunos países han establecido prohibiciones para la importación de cualquier tipo de fruta, y en esas prohibiciones se han dejado fuera frutas como el tomate. Si a un niño se le pregunta si un tomate es una fruta o una verdura lo más probable es que conteste que es una verdura, y si se le pregunta por qué, es muy probable que diga que así se lo han enseñado en el colegio [12].

El propósito de esta introducción sobre lo que es o no es una fruta, es el de dejar patente que el ser humano, cuando se encuentra ante algo nuevo, lo percibe por todos sus sentidos e intenta buscar parecidos entre lo nuevo y lo que ya conoce para intentar identificar ese objeto nuevo, recurriendo para ello a la clasificación. Siguiendo con el

ejemplo de la fruta, si a una persona se le presenta una bandeja con lo que él considera frutas y además se añaden dos frutas que no conozca, como podría ser la fruta del dragón (fruta tailandesa con forma de piña rosa que en su interior contiene una pulpa blanca con semillas negras) y el mangostán (fruta tailandesa de piel gruesa y



Figura 3-4: Imagen de una pitahaya o también conocida como fruta del dragón. Fuente: Wikipedia

de color morado en cuyo interior se encuentra la parte comestible que recuerda a una cabeza de ajos), lo más probable es que considere que son frutas, mientras que si se le presentan junto a un brócoli y una coliflor, lo más probable es que lo califique como verdura.



Figura 3-5: Imagen de una garcinia mangostana o también conocida como mangostán. Fuente: Wikipedia

Con este ejemplo se trata de demostrar que el ser humano tiene la necesidad de clasificar para definir y entender, y en muchas ocasiones esa clasificación puede estar influenciada por el entorno o directamente no ser correcta o no serlo en el momento de realizarla y, posteriormente, gracias a la intervención de la ciencia y la tecnología, esa clasificación puede ser corregida.

Dejando de lado el ejemplo anterior y pasando a realizar una definición más técnica, el agrupamiento (en inglés *cluster analysis* o simplemente *clustering*) es un método para identificar grupos homogéneos llamados clústeres [15]. En general, los objetos que pertenecen a un clúster son bastante parecidos entre sí ya que comparten varios atributos, mientras que los elementos de diferentes clústeres suelen ser bastante diferentes. Por ejemplo, si se quisiera agrupar los pacientes que visitaron un conjunto de hospitales durante los últimos años, primero hay que analizar qué variables se quieren tener en cuenta, como la edad (que se puede establecer en intervalos de 20 en 20), y a cuál de estos hospitales acudió el paciente. Una vez establecidas estas características que pueden diferenciar a unos pacientes de otros, se tiene que elegir un algoritmo de *clustering* (jerárquico, k-medias, aglomerativo, etc.) y, si se da el caso, algunos otros parámetros, como pueden ser el método de medida de distancias entre cada uno de estos individuos (distancia euclidiana, distancia Manhattan, distancia máxima, etc.). En la tabla 3-1 y en la tabla 3-2 se muestran dos tipos de almacenamiento de datos diferentes. En el primer caso no se utilizan valores en los elementos de la matriz, simplemente se marcan con valores 1 y 0 cuando se da el caso. Por ejemplo en la primera fila se indica que el paciente 1 ha visitado únicamente el hospital A y que su edad está en el rango 0-20. En el segundo caso sí que se almacenan los valores en las celdas de la matriz. La diferencia existente entre estas dos variantes radica en el tratamiento que hay que hacer de los datos a la hora de recorrer los elementos de la matriz. Generalmente, el primer caso, aunque requiere mayor número de filas y columnas, suele dar mejores resultados en cuanto a rendimiento se refiere y permite que los algoritmos sean más versátiles independientemente de la información que se quiera agrupar.

PACIENTE	HOSP A	HOSP B	HOSP C	EDAD 0-20	EDAD 20-40	EDAD 40-60	EDAD 60-100
1	1	0	0	1	0	0	0
2	1	0	0	0	0	1	0
3	0	1	0	0	1	0	0
4	0	0	1	0	0	0	1
5	0	1	0	0	1	0	0
6	1	0	0	0	1	0	0
7	0	0	1	0	0	1	0

Tabla 3-1: ejemplo de almacén de datos sin valores en la matriz

PACIENTE	HOSPITAL	EDAD
1	A	0-20
2	A	40-60
3	B	20-40
4	C	60-100
5	B	20-40
6	A	20-40
7	C	40-60

Tabla 3-2: ejemplo de almacén de datos con valores en la matriz

Todo este conjunto de variables existentes es lo que dan como resultado un amplio conjunto de posibles resultados diferentes a la hora de agrupar la información y que ni mucho menos son todos válidos dependiendo del caso de estudio. En este punto, tal y como se muestra en la figura 3-3, es labor humana el decidir la bonanza del algoritmo y decidir el más idóneo para cada caso.

3.2.3. ALGORITMOS DE CLUSTERING

3.2.3.1. ALGORITMOS JERÁRQUICOS

Los algoritmos de *clustering* jerárquico agrupan la información en estructuras jerárquicas o de árbol, creando un grupo principal que es la raíz del árbol del cual cuelgan otras agrupaciones, las cuales a su vez contienen a otros grupos, y así sucesivamente hasta llegar a los propios elementos a clasificar, que serían las hojas del árbol. Dentro de estos algoritmos hay dos métodos de crear las agrupaciones, el método ascendente y el descendente. En el ascendente se parte de que todos los elementos pertenecen a un grupo en el que solo ellos son miembros. Estos grupos comienzan a formar parejas y solo uno de ellos sube de jerarquía, y así sucesivamente hasta llegar al grupo que incluye a todos los demás. Estos algoritmos tienen una complejidad de $O(n^3)$, muy elevada para conjuntos de datos muy grandes. Los algoritmos descendentes sin embargo realizan el proceso contrario, parten del nodo raíz como único grupo, dividiéndolo poco a poco en otros grupos en base a su diferencia hasta llegar a que cada grupo esté formado por un único elemento. La complejidad de estos algoritmos es de $O(2^n)$, radicalmente peor que en el caso de los otros algoritmos, es por esto que los algoritmos descendentes no se emplean habitualmente.

Para representar los resultados de este tipo de algoritmos se suelen utilizar los dendrogramas, que unen mediante líneas los elementos finales o grupos que forman otro grupo hasta llegar al grupo principal que es el nodo raíz. Si la representación está realizada correctamente, en un dendrograma importa la lejanía existente entre cada línea de división, que expresa la distancia existente entre cada uno de los grupos, así como la longitud de las líneas, que muestra el porcentaje de similitud, por lo que a mayor separación entre líneas, mayor distancia o diferencia entre grupos. Sobre este tema de cálculo y representación de distancias se han realizado diversos estudios, como los explicados en el artículo "Construcción de dendrogramas de taxonomía numérica mediante el coeficiente de distancia x^2: una revisión" [16]. Estos gráficos pueden representarse de forma horizontal o forma vertical. En la figura 3-6 se muestra

una clasificación realizada con *clustering* jerárquico en base a los datos de consumo de carnes rojas, carnes blancas, huevos, cereales, leche, pescado, fécula, nueces, frutas y verduras, de los países europeos. Del dendrograma se puede extraer que existen seis grupos principales en una agrupación de similitud del 69%, que son:

- Países Balcánicos: Albania, Bulgaria, Romania, y Yugoslavia.
- Países Mediterráneos: Grecia e Italia.
- Países de la Península Ibérica: España y Portugal.
- Países de Europa del Oeste: Austria, Holanda, Suiza, Bélgica, Alemania del Oeste, Irlanda, Francia y Reino Unido.
- Países escandinavos: Dinamarca, Noruega, Finlandia y Suecia.
- Países de Europa del Este: Checoslovaquia, Alemania del Este, Polonia, U.S.S.R. y Hungría.

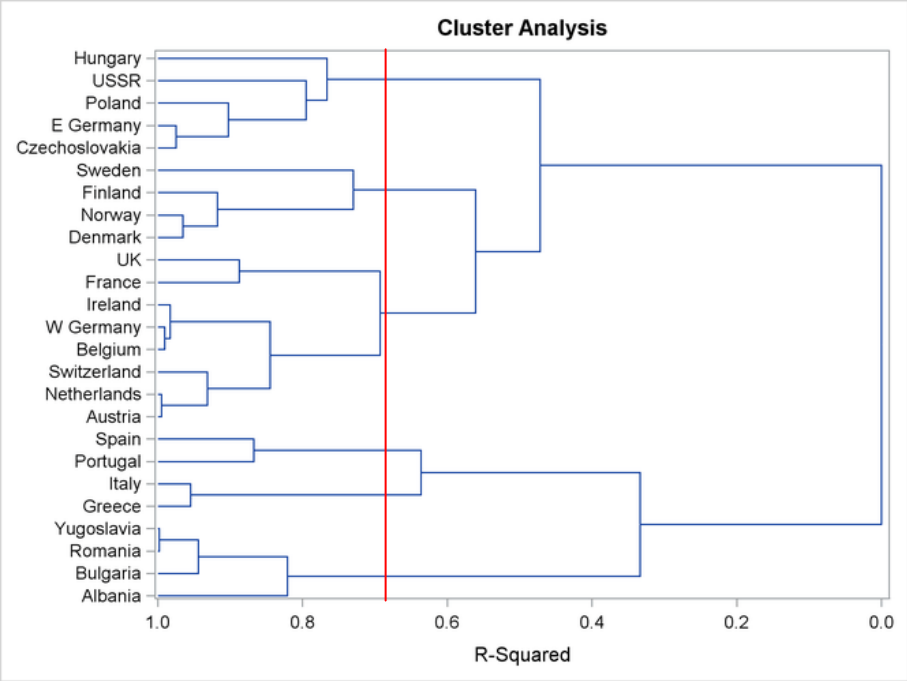


Figura 3-6: dendrograma de países por consumo de alimentos. Fuente [17].

Viendo este gráfico se demuestra la potencia que tiene un dendrograma a la hora de mostrar resultados de clasificación. Analizando los datos devueltos, podemos ver que la alimentación de España y Portugal es similar en un 85% y que la de Italia y Grecia en un 94%. El grupo formado por estos cuatro países es similar en un 63%.

En estos algoritmos jerárquicos, se utilizan diferentes métricas para calcular la distancia entre dos elementos. Las más utilizadas según [18] son:

- Euclídea: $d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
- Euclídea al cuadrado: $d = \sum_{i=1}^{n}(x_i - y_i)^2$
- Manhattan: $d = \sum_{i=1}^{n}|x_i - y_i|$
- Máxima: $d = \max_i|x_i - y_i|$
- Correlación de Person: mide la similitud entre formas siguiendo la siguiente fórmula:

  $$d = 1 - r$$

  donde

  $$r = Z(x) * Z(y)/n$$

  que es el producto escalar de los *z-scores* de x e y. El *z-score* de un vector se obtiene restando del vector su media y dividiéndola por su desviación típica.

En cuanto a las desventajas de estos algoritmos, ya se ha comentado que estos algoritmos tienen una complejidad computacional muy elevada para grandes cantidades de datos. Además, por cómo se genera el árbol resultante, no existe la posibilidad de corregir errores en la clasificación.

3.2.3.2. ALGORITMOS PARTICIONALES

Los algoritmos particionales se caracterizan por crear agrupaciones finales sin establecer una jerarquía. Son bastante eficientes con grandes volúmenes de datos, sin

embargo presentan el inconveniente de tener que elegir a priori el número de clústeres que tienen que devolver. Actualmente se han realizado diversos estudios para calcular de forma eficiente el valor del número de grupos [19]. Dentro de estos algoritmos hay que destacar dos de ellos, *K-Means* y *Fuzzy C-Means*.

ALGORITMO K-MEANS

Se trata de un sencillo algoritmo con buenos resultados y que es de los más utilizados dentro del grupo de algoritmos de aprendizaje no supervisado. Su complejidad es NP si no se establece el número de clústeres, mientras que si se indica un valor concreto, la complejidad es $O(n * k * i * a)$, donde *n* es el número de objetos, *k* el número de clústeres, *i* el número iteraciones y *a* el número de atributos.

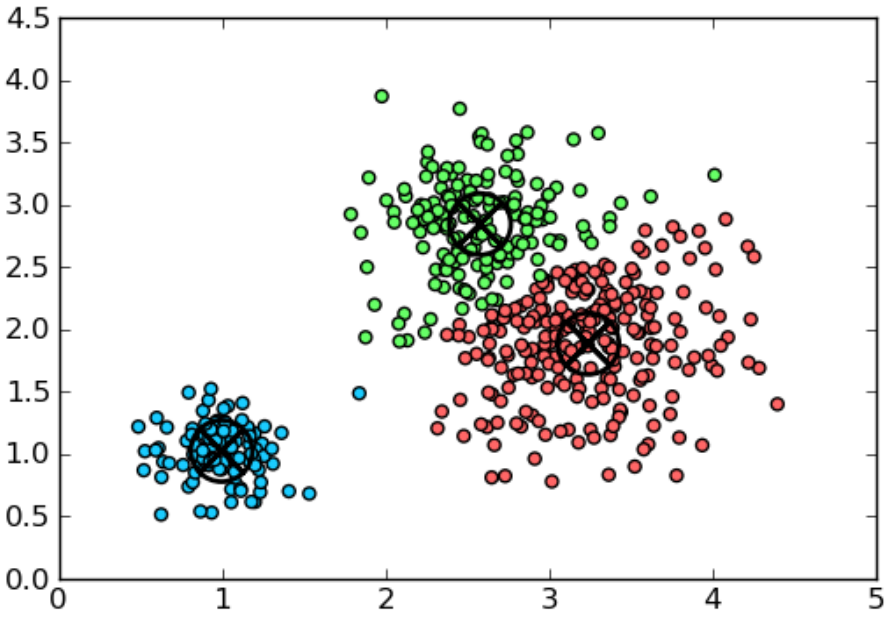


Figura 3-7: ejemplo del resultado de *k-means* con k = 3

Como principal inconveniente se encuentra la característica típica de los algoritmos particionales ya comentada, que es la necesidad de indicar previamente el número de agrupaciones deseadas realizando pruebas aleatorias o recurriendo a otros algoritmos [19], ya que la búsqueda del baricentro, o también llamado *centroide*, difiere en gran

medida en función del número de clústeres. En la figura 3-7 se puede observar el resultado devuelto por el algoritmo con un número de 3 clústeres. En la imagen se puede apreciar también cada uno de los *centroides* marcados por una cruz en el centro de cada agrupación.

Otro de los inconvenientes es que este algoritmo es muy sensible a objetos fuera de los valores normales (*outliers* en inglés), aunque existen algoritmos que intentan resolver estos problemas como *K-Medoids* [20] o PAM [21].

ALGORITMO FUZZY C-MEANS

Se trata de un algoritmo basado en el algoritmo *K-Means*, con la diferencia de que es difuso, es decir, utiliza una función de pertenencia y cada elemento no tiene el atributo de pertenecer o no a un grupo, sino que cada elemento tiene un porcentaje de pertenencia a cada uno de todos los grupos existentes. En general este método mejora el cálculo del *centroide* y por tanto de los elementos que pertenecen a cada grupo en comparación con *K-Means*, sin embargo, dado que el número de clústeres también tiene que ser indicado antes de la ejecución, un número de clústeres erróneo puede dar resultados equívocos.

3.3. R COMO HERRAMIENTA DE MINERÍA DE DATOS

R es un software y entorno de programación que se distribuye bajo licencia GNU (*Gnu Not* UNIX por sus siglas en inglés) GPL (*General Public License* por sus siglas en inglés) y tiene versiones para Windows, UNIX y MacOS [22]. Está enfocado en el análisis estadístico y gráfico, aportando una gran cantidad de herramientas y librerías. Es un lenguaje orientado a objetos que permite interactuar con librerías escritas en otros lenguajes como C, C++ o Java. Al ser un software libre, ha sido empleado en multitud de análisis científicos y concretamente en los últimos años ha sufrido su mayor auge gracias al fenómeno *Big Data*, ya que la comunidad de usuarios de R ha aportado librerías y funcionalidades que le han hecho más popular en este campo. Gracias a esta comunidad que crece día a día, en Internet se pueden encontrar cantidad de ejemplos y de proyectos de mejora.

En cuanto a su sintaxis se puede decir que es moderna y muy abierta, permitiendo instrucciones complejas que no se pueden realizar en otros lenguajes, especialmente en el tratamiento de vectores y matrices, incluyendo además otros objetos específicos para ciertos tipos de datos como pueden ser los *dataframes* o las *listas*. Por el contrario, este abanico tan grande de posibilidades también eleva la complejidad de su aprendizaje para iniciados en la programación.

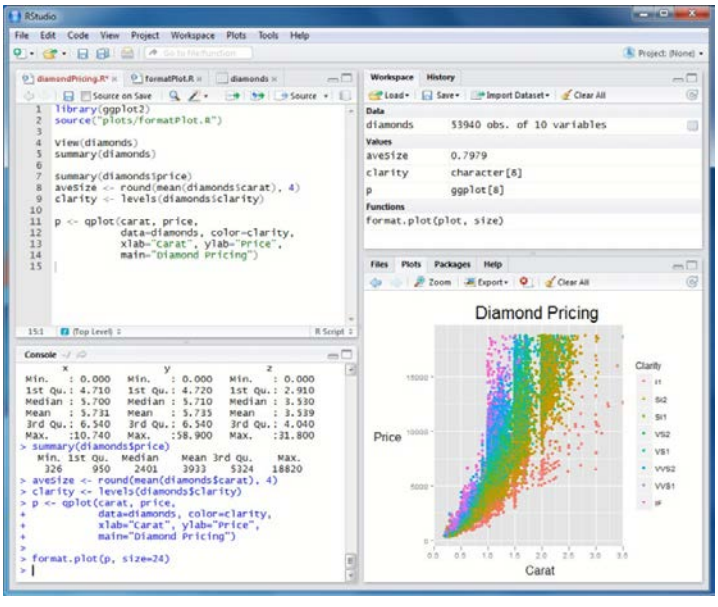


Figura 3-8: imagen del entorno R Studio.

Durante los últimos años han surgido diferentes entornos para trabajar con R. El más conocido y utilizado hoy en día es R Studio [23], que también es libre y permite trabajar de manera más fluida con R en entornos Windows, Linux y MacOS. RStudio Inc. ofrece otra herramienta llamada Shiny para el desarrollo de aplicaciones web con R, sin embargo a día de hoy los resultados todavía son bastante pobres.

Otra característica que hay que destacar de R es que se trata de un lenguaje interpretado y que mediante R Studio se permite que la ejecución del código no siga un orden de ejecución concreto. Aunque se puede ejecutar todo el código que se está editando, el trabajo normal con R consiste en ir ejecutando las líneas que interesen en cada momento, lo que ayuda en gran medida en la depuración y en las tareas de investigación porque se pueden ir probando resultados sobre la marcha, ejecutando líneas de código que se encuentran en otras partes del archivo.

Dentro del paquete de instalación de R se incorporan diversos paquetes básicos para su uso, sin embargo en Internet se ofrecen multitud de paquetes para realizar operaciones. En el momento de la redacción de este documento, el número de paquetes ascendía a casi 7.000. A continuación se muestran algunos que han sido empleados en este proyecto:

- *Clustering*: *cluster*, *fpc*
- Representación gráfica: ggplot2
- Operaciones adicionales con matrices: *Matrix*, *RTextTools*
- Tratamiento de archivos JSON: *RJSONIO*
- Tratamiento de textos: *TM*, *data*, stringr

En el apartado de minería de datos, R es considerado actualmente como una de las mejores herramientas en cuanto a la existencia de librerías, estructuras internas de datos y algoritmos ya desarrollados para el tratamiento de textos.

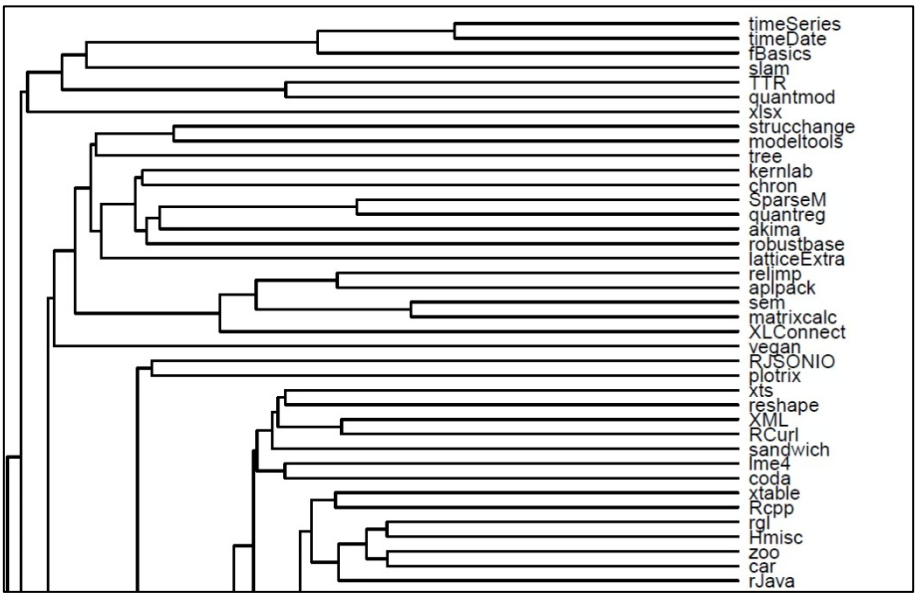


Figura 3-9: dendrograma de familias de paquetes de R. Fuente: r-statistics [24].

En cuanto a las estructuras de datos, R tiene varias estructuras que son necesarias y que se han utilizado en este proyecto:

- ***Data frame***: es una lista de variables almacenada en una matriz, asignando una variable a cada fila y permitiendo su identificación con un nombre. Si no se indica el nombre, R asigna un número para cada fila. Muchas librerías de R utilizan esta estructura para minería de textos.
- ***Corpus***: es una estructura de datos diseñada específicamente para el tratamiento de textos. En esta estructura generalmente se almacena el texto extraído que se quiere analizar. Muchos algoritmos necesitan esta estructura para extraer información o realizar operaciones.
- ***Document term matrix***: es una estructura parecida a un *data frame* destinada a almacenar todas las apariciones de términos en un conjunto de documentos, almacenando un término en el nombre de las filas y los identificadores de los documentos en las columnas. Si el término *x* aparece en el documento *y*, en la celda correspondiente a la fila *x* y columna *y*, se almacenará un uno, y en caso contrario, un cero.

- ***Term document matrix*:** es la matriz traspuesta a la anterior. Esta estructura existe porque algunos algoritmos utilizan esta y no la anterior. Su aplicación se describe en el apartado 4.2.6.

En relación a los paquetes existentes para la minería de textos, el paquete *TM* es el más utilizado debido a la cantidad de métodos que ofrece, como por ejemplo para calcular la frecuencia relativa y absoluta de apariencia de términos en un texto, para eliminar números, letras o palabras concretas, impresión de documentos almacenados en corpus, etc.

3.4. TECNOLOGÍAS WEB

3.4.1. HTML5

Desde el nacimiento de Internet ha existido la necesidad de presentar la información a través del navegador web. En sus orígenes surgió el *Standard Generalized Markup Language* (SGML por sus siglas en inglés), sistema para la elaboración de lenguajes de etiquetado [25] que se empleó para crear la primera versión del lenguaje HTML (del inglés *HyperText Markup Language*) que se convirtió en un estándar por primera vez en 1993 con su versión 2.0. A partir de ese momento, HTML es el estándar de facto y poco a poco ha ido evolucionando hasta nuestros días. El penúltimo estándar se publicó en diciembre de 1999 como HTML 4.01., incorporando importantes cambios que revolucionaron el diseño de las páginas web. 15 años después, en octubre de 2014, fue publicada la última versión hasta la fecha por la World Wide Web Consortium (W3C por sus siglas en inglés), que dado el tiempo transcurrido entre esta y la versión anterior, y por los numerosos cambios importantes que aportaba esta versión, también se modificó su nombre pasando a ser HTML5.

HTML5, a diferencia de sus anteriores versiones, prácticamente no incorpora elementos de diseño gráfico, en realidad aporta los elementos estructurales necesarios para diseños innovadores pero sin aportar nada de diseño, dejando esto en manos de las Hojas de Estilo en Cascada, que actualmente se encuentran en su versión 3 y en continuo auge. Dicho de otra manera, si hoy en día alguien decide aprender HTML5 para diseñar una página web, no podrá mostrar otra cosa que texto sin formatear, imágenes descolocadas, tablas sin diseño y poco más, ya que HTML5 está diseñado para crear la estructura interna de un sitio web. Es por ello que HTML5 necesita de CSS (del inglés *Cascading Style Sheets*) y casi se podría decir que también de JavaScript.

3.4.2. CSS3

CSS es un lenguaje creado para definir la presentación gráfica del lenguaje HTML. CSS3 es la última versión de las Hojas de Estilo en Cascada aunque todavía no se ha publicado como estándar. Al igual que con HTML, la W3C es la encargada de los estándares, cuyo última versión publicada es la 2.1, aunque actualmente se publican multitud de recomendaciones hacia el próximo estándar [26]. Estas recomendaciones deben ser aceptadas por los navegadores web para que los desarrolladores puedan presentar la información de forma correcta, aunque no todos los navegadores lo hacen de la misma manera ni al mismo tiempo. En la tabla 3-3 se muestra un ejemplo de compatibilidad que diversos sitios web ofrecen como consulta.

CSS3 Properties

	MAC						WIN								
	CHROME		FIREFOX	OPERA		SAFARI		CHROME	FIREFOX	OPERA	IE				
	25		20	12	15	5.1	6	25	15	12	6	7	8	9	10
RGBA	✓		✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	91%
HSLA	✓		✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	91%
Box Sizing	✓		✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	35%
Background Size	✓		✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	90%
Multiple Backgrounds	✓		✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	90%
Border Image	✓		✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	85%
Border Radius	✓		✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	91%
Box Shadow	✓		✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	91%
Text Shadow	✓		✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	76%
Opacity	✓		✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	91%
CSS Animations	✓		✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	66%
CSS Columns	✓		✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	85%
CSS Gradients	✓		✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	84%

Tabla 3-3: compatibilidad de navegadores con algunas propiedades de CSS

3.4.3. GRAFICOS VECTORIALES SVG

Los gráficos vectoriales redimensionables (*Scalable Vector Graphics* o SVG por sus siglas en inglés) son gráficos generados por vectores bidimensionales, por lo que se

pueden redimensionar sin pérdida de calidad a diferencia de los gráficos de trama. SVG es una recomendación de la W3C desde 2001 por lo que a día de hoy lo soportan todos los navegadores de forma implícita y sin complementos adicionales. La información que almacena internamente un archivo de trama es un conjunto de atributos por pixel, como la combinación de colores rojo, verde y azul, la luminosidad, transparencia, etc. por lo que a mayor resolución de imagen, mayor espacio necesita para almacenarse. Los gráficos vectoriales en cambio, no almacenan información por cada pixel, sino que almacenan información de elementos geométricos como líneas, círculos, rectángulos, etc., y que mediante fórmulas matemáticas, representa el resultado en la pantalla, por lo tanto no es necesario almacenar tanta información, permitiendo además que el gráfico se pueda redimensionar sin pérdida de calidad, porque con volver a realizar los cálculos necesarios, se puede presentar el gráfico con otro tamaño. En la Figura 3-10 se muestra el gráfico vectorial de un tomate creado con Inkscape, herramienta de código libre para la creación de gráficos vectoriales que permite la exportación a formato SVG de HTML. En la figura 3-11 se muestra la diferencia de calidad entre la imagen redimensionada a 250% en formato SVG (parte superior) y una imagen redimensionada con una herramienta de diseño gráfico.

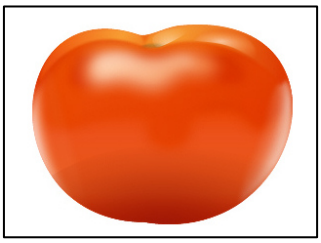


Figura 3-10: gráfico SVG generado con Inkscape. Fuente [27].

El código necesario para representar el tomate está compuesto de 1995 líneas de código, por lo que se puede deducir que estos gráficos SVG tan complejos tienen que ser realizados con herramientas de diseño vectorial. Como curiosidad, en la figura 3-12 se muestra un pequeño fragmento del código utilizado para representarlo.

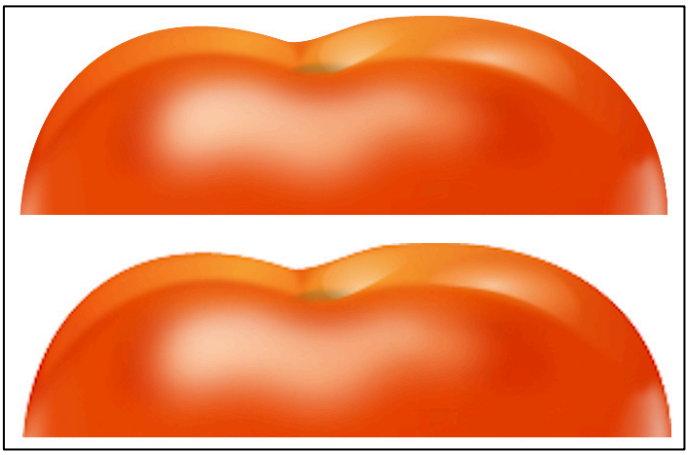


Figura 3-11: comparación de calidad entre SVG y mapa de bits.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- Created with Inkscape (http://www.inkscape.org/) -->

<svg
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   xmlns:cc="http://creativecommons.org/ns#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:svg="http://www.w3.org/2000/svg"
   xmlns="http://www.w3.org/2000/svg"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   width="266.97751"
   height="205.18167"
   id="svg3863">
  <defs
     id="defs3865">
    <linearGradient
       id="linearGradient4168">
      <stop
         id="stop4170"
         style="stop-color:#a21907;stop-opacity:1"
         offset="0" />
      <stop
         id="stop4172"
         style="stop-color:#9c1901;stop-opacity:0.203922"
         offset="0.72458541" />
      <stop
         id="stop4174"
         style="stop-color:#9b1a00;stop-opacity:0"
         offset="1" />
    </linearGradient>
    <clipPath
       id="clipPath3999">
      <path
         d="m 446.78572,93.25504 c 3.25282,1.17209 10.03657,-0.45667 12.4999
         id="path4001"
         style="fill:url(#linearGradient4003) #000000;fill-opacity:1;fill-ru
    </clipPath>
    <linearGradient
       x1="190.75587"
       y1="81.827255"
       x2="190.75587"
       y2="303.54019"
       id="linearGradient4003"
       xlink:href="#linearGradient3169"
       gradientUnits="userSpaceOnUse"
       gradientTransform="translate(280,0)" />
    <clipPath
       id="clipPath4005">
      <path
         d="m 446.78572,93.25504 c 3.25282,1.17209 10.03657,-0.45667 12.4999
         id="path4007"
         style="fill:url(#linearGradient4009) #000000;fill-opacity:1;fill-ru
    </clipPath>
```

Figura 3-12: porción de código para la representación de un tomate en SVG

3.4.4. JAVASCRIPT

JavaScript o también conocido como JS, es un lenguaje de programación orientado a objetos basado en el estándar ECMAScript [28]. Este estándar fue publicado por primera vez en 1997 basándose precisamente en JavaScript de NetScape y JScript de Microsoft. En 2015 se ha publicado la sexta versión del estándar [29] y la mayoría de los navegadores incluyen una implementación, por lo que en principio, todos los navegadores deberían poder interpretar cualquier lenguaje que se adapte al estándar, como es el caso de JavaScript, sin embargo JS crece a pasos agigantados y aunque el núcleo de JavaScript sí se adapta al estándar, otras partes no lo hacen, por lo que hay que adaptar el código a cada navegador.



Figura 3-13: vista de Madrid en 3D con Google Maps

JavaScript es un lenguaje interpretado que se ejecuta en el lado del cliente dentro del navegador web, proporcionando dinamismo a las páginas web y notables mejoras en la apariencia e interactividad de la interfaz de usuario. Hoy en día es probablemente el lenguaje más popular en el desarrollo web, aunque no fue hasta la introducción de AJAX (en inglés *Asynchronous JavaScript And XML*) cuando creció su popularidad, ya que desde ese momento se permitía la comunicación asíncrona con el servidor y la modificación de elementos HTML en el documento web sin necesidad de recargar la

página del navegador, lo que potenciaba en gran medida la creación de aplicaciones web interactivas (*Rich Internet Applications* o RIA por sus siglas en inglés) y también la creación de gráficos dinámicos e interactivos. Diversas aplicaciones web han ayudado a popularizar y a avanzar en el desarrollo de AJAX y JavaScript, como es el caso de Gmail o Google Maps de Google, dos de las RIA más avanzadas que existen en la actualidad [30].

3.5. VISUALIZACIÓN DE DATOS MEDIANTE JAVASCRIPT Y HTML5

Como ya se comentó en la introducción del Estado Del Arte, en la actualidad nos encontramos en un momento en el que las personas son las protagonistas principales de la revolución informativa en cuanto a la creación y consumo de datos se refiere, donde las redes sociales, los smartphones, las tabletas, las pulseras de actividad, los GPS, etc., provocan una inmensa cantidad de información que además de almacenarse hay que tratar, manipular y por tanto mostrar. Esa representación de información después de haber sido procesada, de resultados de estudios, y de hasta una simple estadística, provoca la necesidad de crear diferentes tipos de gráficos en función de las necesidades y del público al que va dirigido. Por ejemplo un niño asimilará mejor un gráfico simple explicado por algún dibujo animado que le resulte conocido; alguien habituado al uso de los dispositivos móviles puede buscar un gráfico interactivo que, además de mostrar diferentes tipos de información según se trabaja con el mismo, permita la posibilidad de hacer zoom, o de ir navegando a través de él, mostrando y ocultando información según sus deseos. Sea cual sea el gráfico, éste se tiene que poder visualizar tanto en un ordenador como en un dispositivo móvil, independientemente del fabricante del dispositivo y del sistema operativo que utilice, por lo que el mejor recurso multiplataforma del que disponemos hoy en día es el navegador web, y como tecnología de desarrollo JavaScript.

3.5.1. APLICACIONES WEB PARA LA VISUALIZACIÓN

Existen multitud de herramientas web que ofrecen diferentes posibilidades de visualización de datos e incluso la opción de crear una simple aplicación para dispositivos móviles donde se pueda visualizar dicho gráfico. Si bien la mayoría ofrecen la típica representación de gráficos de barras o de sectores, algunas otras ofrecen diferentes posibilidades como nubes de palabras, árboles o grafos.

3.5.1.1. Google Fusion Tables

Fusion Tables [31] es una aplicación de Google Inc. que se encuentra en fase experimental y que está destinada a la creación, manipulación y publicación de gráficos online. Su enfoque principal es la sencillez para importar grandes volúmenes de datos, filtrarlos y publicarlos en una página web o compartirlos en una red social. Siguiendo con la idea de la publicación, otro de sus puntos fuertes es el de permitir fusionar tablas de diferentes usuarios dentro de una misma tabla y actualizar los gráficos en tiempo real. Por otro lado, Fusion Tables aporta otra finalidad, que es la de mostrar información sobre mapas de ciudades de forma automática, como por ejemplo las localizaciones de cada una de las franquicias de una cadena de restaurantes o diferentes datos estadísticos en función del punto del mapa que se seleccione.

En cuanto a su tecnología, Fusion Tables utiliza principalmente JavaScript por el hecho de poderse visualizar en la mayoría de dispositivos y sistemas operativos.

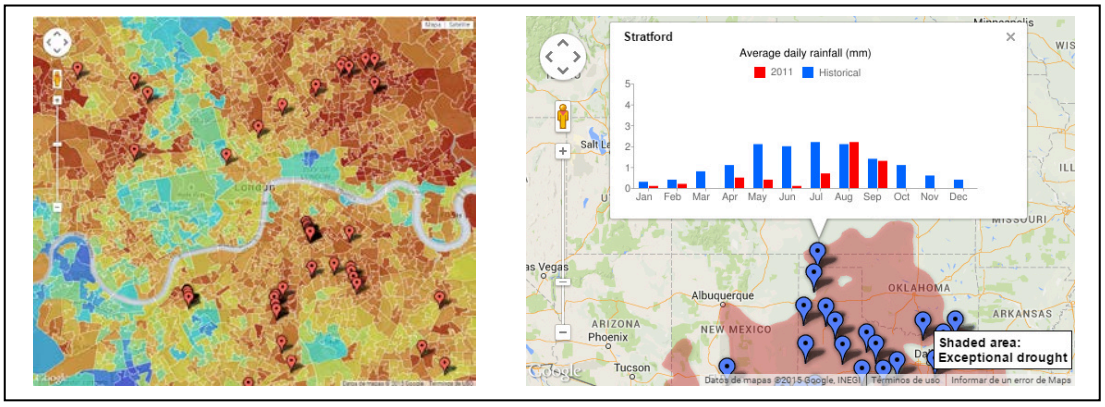


Figura 3-14: Ejemplos de Google Fusion Tables. Fuente: [32].

3.5.1.2. CartoDB

CartoDB [33] es un SIG (sistema de información geográfica) que se encuentra en una plataforma de computación en la nube y que proporciona herramientas para crear y visualizar mapas a través de un navegador, crear aplicaciones que manejen y representen datos en un mapa, entre otras. Su uso es gratuito hasta 50 MB de datos y puede utilizarse desde sus servidores o desde servidores externos. La tecnología más

empleada en la mayoría de sus servicios es JavaScript y como base de datos PostgreSQL.

En la figura 3-15 se muestra una imagen del mapa interactivo sobre los Solsticios de Verano de Manhattan[1] para que sus ciudadanos puedan saber a qué hora y en qué calle de Manhattan se podrá ver el Sol.



Figura 3-15: Manhattanhenge. Fuente: NYCHenge.com [34].

Otro ejemplo de lo que se puede realizar con CartoDB es representar en un mapa datos estadísticos en función del momento. En el siguiente ejemplo se muestra una imagen estática del mapa interactivo que describe el volumen de las reservas de petróleo existentes en el mundo.

[1] Fenómeno que ocurre dos veces al año y durante el cual el Sol se va alineando con las calles de Manhattan (Nueva York). Se le conoce por el nombre de Manhattanhenge por su neologismo con las ruinas de Stonehenge en Inglaterra.

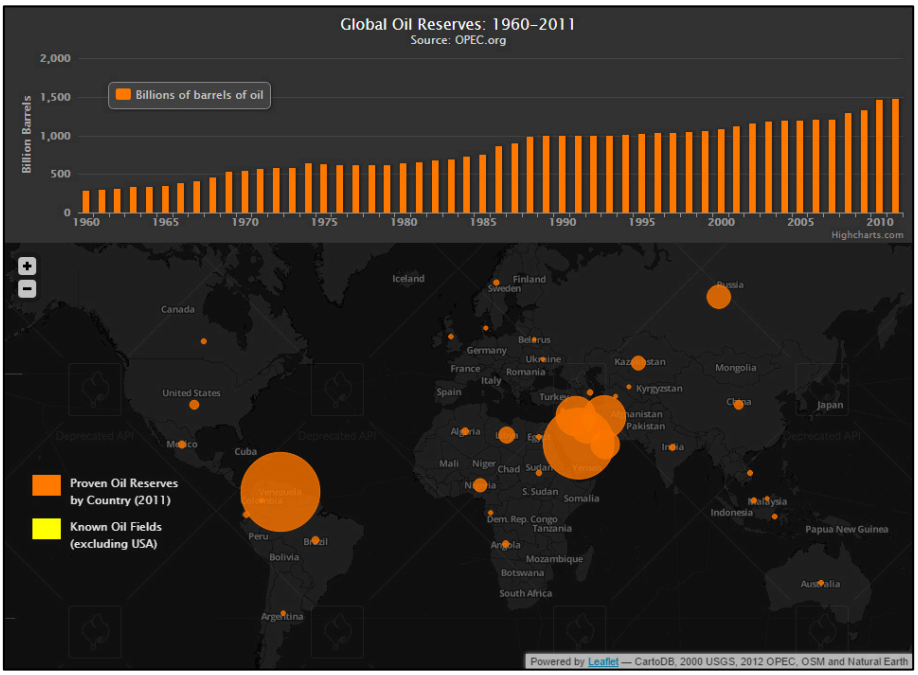


Figura 3-16: reservas mundiales de petróleo con CartoDB. Fuente: Azavea Atlas [35].

3.5.1.3. Tableau Public

Siguiendo en la línea de las anteriores herramientas, hay que hacer referencia a Tableau Public [36], herramienta gratuita de escritorio y online para la visualización de datos mediante gráficos que combinan una interfaz gráfica amigable, dinámica y eficiente, con los elementos tradicionales de las herramientas de *business inteligence*, tales como por ejemplo el modelo organizativo de variables utilizando dimensiones y medidas, o las conexiones con otros sistemas de gestión de información.

A pesar de tener una herramienta de escritorio, dentro de las tecnologías, cabe destacar que la representación de datos se realiza mediante JavaScript, y que para la versión online se puede encontrar una API de JavaScript desarrollada por Tableau Software, la cual permite integrar gráficos de negocio dentro de una página web.

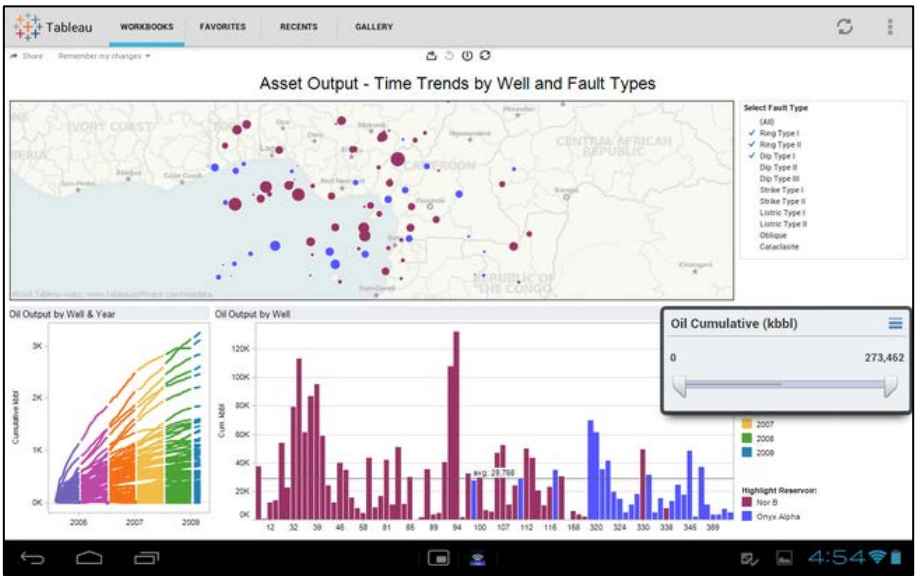


Figura 3-17: ejemplo de visualización con Tableau Public. Fuente: Tableau Software [36].

3.5.1.4. Many Eyes

Many Eyes [37] es una aplicación web creada por IBM, que al igual que Fusion Tables, permite al usuario crear, gestionar y publicar gráficos generados a partir de datos subidos por los usuarios. Esta herramienta está orientada principalmente a la publicación y por lo tanto no se pueden crear gráficos de forma privada.

Para la creación de las visualizaciones, ofrecen un sencillo asistente mediante el cual se puede crear un potente gráfico en tres sencillos pasos.



Figura 3-18: nube de palabras interactiva creada con Many Eyes



Figura 3-19: gráfico de burbujas creado con Many Eyes

3.5.2. API Y LIBRERÍAS PARA LA VISUALIZACIÓN WEB

Además de las herramientas mostradas en el apartado anterior, existen varias API y librerías disponibles para desarrolladores web que desean crear sus propias herramientas de visualización. A continuación se describirán las más conocidas y utilizadas en la actualidad.

3.5.2.1. Google Chart Tools

Google Chart Tools [38] es una API desarrollada por Google Inc. para generar gráficos en formato de red portátiles (PNG, por sus siglas en inglés), lo que hace muy sencilla su inserción en una página web utilizando una simple etiqueta <*img*>, sin necesidad de instalar ningún entorno ni componente adicional. Para gráficos más complejos como un árbol de palabras es necesario utilizar JavaScript.

Dentro del amplio abanico de gráficos que se pueden utilizar, 29 en total, se encuentran los típicos gráficos de barras, sectores o lineales y además histogramas, árboles de palabras y mapas entre otros.

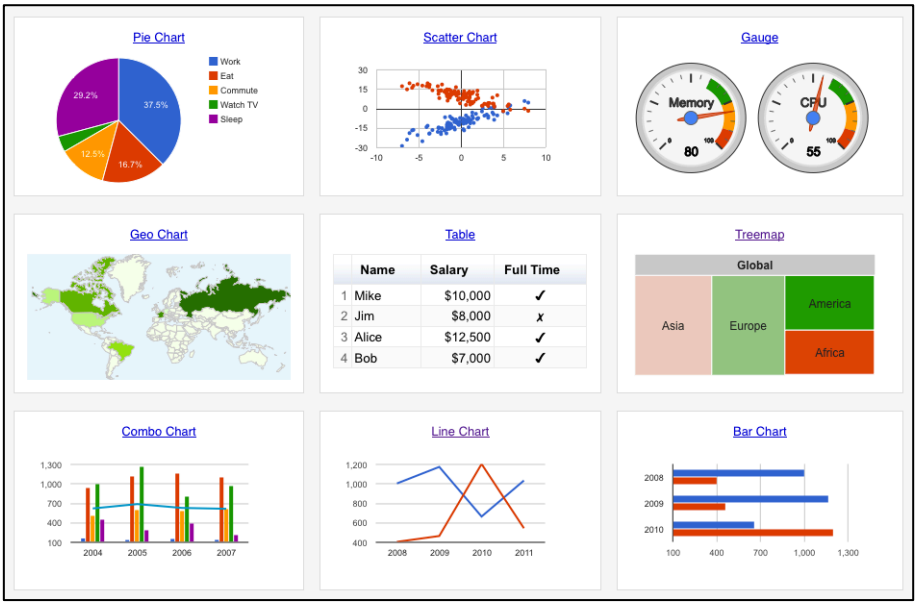


Figura 3-20: algunos ejemplos de gráficos de Google Chart Tools. Fuente: CTO Insights [39].

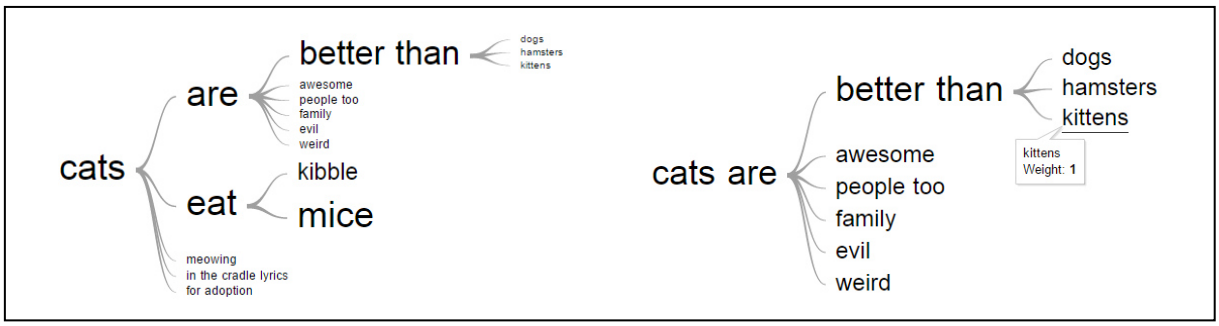


Figura 3-21: árbol de palabras interactivo. API Chart y JavaScript. Fuente: Google [34].

3.5.2.2. FusionCharts

FusionCharts [40] es una librería que inicialmente permitía crear gráficos en formato Adobe Flash con una calidad y apariencia visual sin competencia. Debido a la poca compatibilidad de Adobe Flash con los dispositivos móviles, actualmente está obsoleta y ofrece una librería en JavaScript, la cual, aunque no tiene tanta potencia como la anterior, sigue destacando frente a cualquier gráfico generado con otras

herramientas, y al desarrollar los gráficos en JavaScript, se pueden visualizar en cualquier plataforma, incluso en antiguas versiones de Internet Explorer.

Esta librería no dispone de ninguna versión gratuita aunque se puede utilizar una versión de prueba sin límite de uso que añade una marca de agua en los gráficos generados.

En la siguiente imagen se puede observar la calidad de los gráficos y el buen acabado y detalles que ofrecen.



Figura 3-22: algunos gráficos generados con FusionCharts. Fuente: FusionCharts [40].

3.5.2.3. JavaScript Infovis Toolkit

Se trata de una librería basada en tecnología JavaScript y Python bajo licencia MIT que permite la creación de gráficos interactivos que se pueden insertar en una página web, tales como árboles de conexión, árboles jerárquicos, mapas relacionales, Árboles Hiperbólicos, etc. Sus características principales son:

- Permite interaccionar con los datos en tiempo real.
- Es compatible con la mayoría de exploradores por haberse desarrollado en JavaScript.
- Es Open Source.
- Representa estructuras de datos complejas gracias a la interactividad de los gráficos, lo que no sería posible con gráficos que no fueran interactivos.

A diferencia de lo que ocurre con otras de las herramientas explicadas anteriormente, utiliza archivos JavaScript Object Notation (JSON, por sus siglas en inglés) para el intercambio de datos y su representación.

En la figura 3-23 se muestra un árbol interactivo en el que se representa las dependencias de los empleados de una empresa. En la figura 3-24 se puede observar un grafo interactivo que permite al usuario modificar la posición de cada uno de los nodos del grafo.

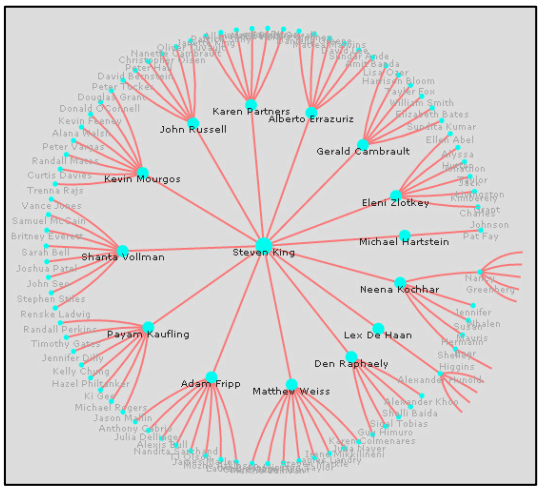


Figura 3-23: árbol de asociaciones con JavaScript Infovis Toolkit. Fuente: R-bloggers [41].

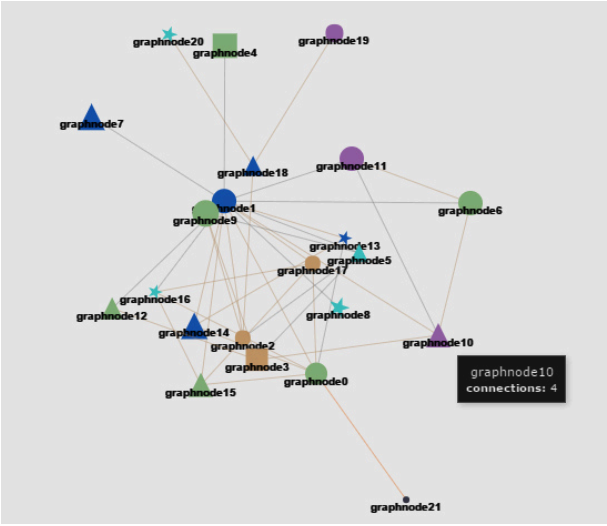


Figura 3-24: árbol interactivo. Fuente: JavaScript Infovis Toolkit [42].

3.5.2.4. D3.js

D3.js [43] es una librería desarrollada en JavaScript para la creación de gráficos dinámicos e interactivos que puedan ser utilizados en un navegador web. Su enfoque principal es el de aprovechar los últimos avances en las siguientes tecnologías que ya han sido explicadas anteriormente, SVG, HTML5 y CSS3.

Una de las características por las que hay que destacar a esta librería es la de que permite una gran libertad a la hora de modificar los aspectos visuales finales de los gráficos, así como su tratamiento dentro de una página web. Se basa en un método único de selección y adición de datos, por lo que no es necesario estructurar los datos de entrada en función de la visualización, lo que convierte a esta librería en una de las más versátiles.

En la figura 3-25 y en la figura 3-26 se muestran diferentes gráficos creados con D3.js.

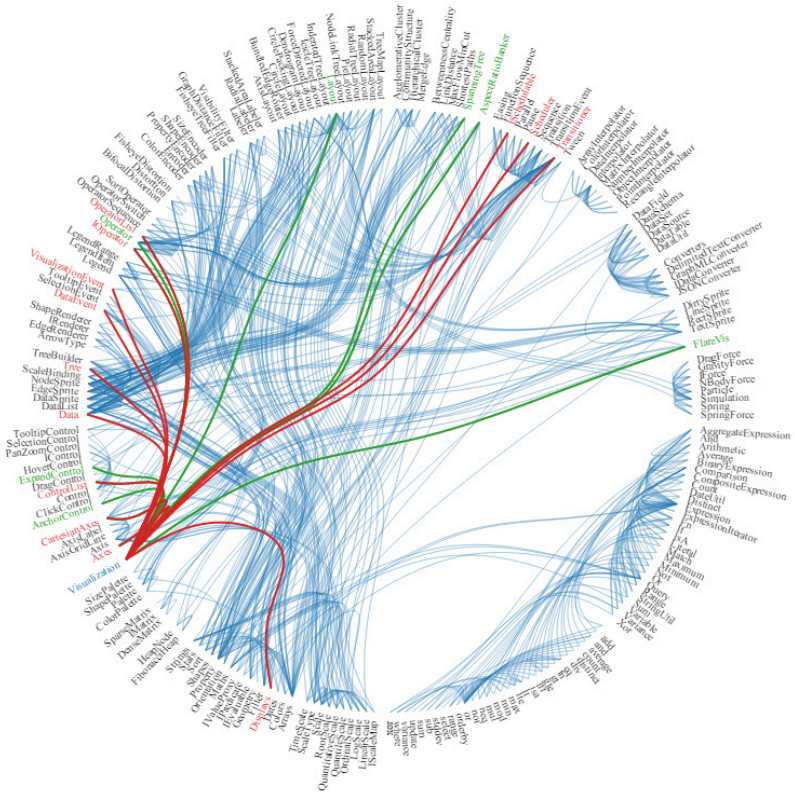


Figura 3-25: ejemplo de gráfico jerárquico creado con D3. Fuente: D3.js [43].

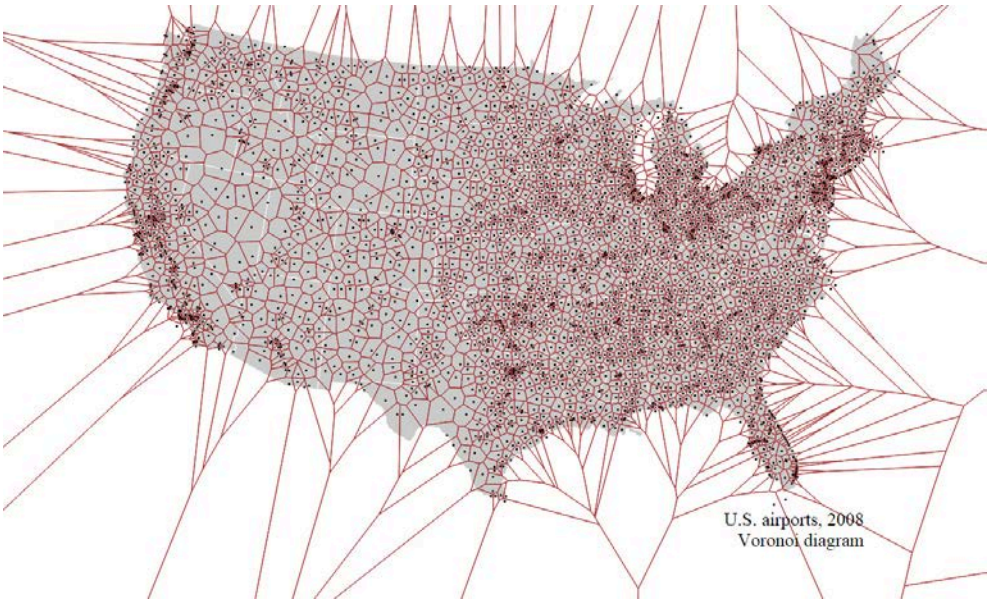


Figura 3-26: diagrama de Voronoi con los aeropuertos de Estados Unidos. Fuente: D3.js [43].

3.5.3. ARCHIVOS DE ALMACENAMIENTO JSON

Los archivos JSON son ficheros en texto plano para enviar datos mediante JavaScript sin la necesidad de utilizar eXtensible Markup Language (XML, por sus siglas en inglés). Estos archivos tienen un tamaño más pequeño y carecen de la estructura de XML, por lo que ofrecen mejores resultados para los analizadores sintácticos de JavaScript. Tienen una sintaxis simple formada principalmente por conjuntos de atributos y valores. En este proyecto interesa principalmente la sintaxis de objetos y arrays, donde se volcarán todos los datos de resultados. En la figura 3-27 se muestra un ejemplo de la estructura de datos de estos archivos, donde se ha almacenado la descripción de la guía docente y el número de alumnos ficticio de dos asignaturas del máster. Un almacenamiento como el del ejemplo permitiría representar en un gráfico de burbujas todas las asignaturas del máster, donde cada burbuja sería una asignatura y el tamaño de la misma estaría en relación al número de alumnos matriculados.

```
{"name": "UNED", "children":  [{
  "name": "Master",
  "children": [{
    "name": "Minería de Datos",
    "children": [{
      "name": "El objetivo de esta guía es orientar al alumno en el estudio de la asignatura.",
      "size": "30"
    }
    ]
  }, {
    "name": "Procesado de Señales",
    "children": [{
      "name": "Esta guía proporciona unas orientaciones generales sobre la asignatura",
      "size": "20"
    }
    ]
  }
  ]
 }]
}
```

Figura 3-27: ejemplo de código de un archivo JSON.

3.5.4. ATOM COMO EDITOR DE CÓDIGO FUENTE

Atom [44] es un editor de código fuente creado en GitHub por una pequeña comunidad de desarrolladores. Nace con el objetivo de ofrecer al público un editor rápido, capaz de entender todos los lenguajes de programación existentes y al mismo tiempo gratuito, ya que Sublime Text, otro editor de texto muy parecido, es de pago. Tal y como ellos dicen, Atom es un editor que se puede piratear y su código fuente está disponible dentro de la página web del proyecto en GitHub [45].

Además de poder modificar su apariencia con temas diferentes que se pueden descargar de su página web, existe la posibilidad de añadir paquetes que le aporten nuevas funcionalidades, como por ejemplo una ventana con el mapa del código fuente en la parte derecha, un paquete para encriptar y desencriptar el código fuente, paquetes para trabajar con lenguajes antiguos como COBOL o Fortran, etc.

En definitiva, se trata de un editor gratuito y potente que sirve para editar todos los archivos fuente de este proyecto, evitando así tener una herramienta para cada tipo de lenguaje.

4. DESARROLLO Y RESULTADOS DEL PROYECTO

En el siguiente apartado se explicará de forma detallada cuál ha sido el origen de los datos y cuáles han sido las tareas realizadas en este proyecto, así como los métodos utilizados, las pruebas realizadas y las comparaciones de resultados que se han realizado en cada caso.

4.1. DATOS UTILIZADOS

Como ya se adelantó en el apartado 2.2, los datos con los que se ha trabajado fueron cedidos por la Unidad Central de Radiodiagnóstico de la Comunidad de Madrid. Los datos correspondían a todos los informes radiológicos que la UCR había realizado en sus 6 hospitales durante el año 2013, en total 554.173 informes, y fueron cedidos después de un proceso para anonimizar la información.

Los campos que contenía cada uno de estos informes eran:

- ***patient_id_anon***: número de identificación de paciente anonimizado.
- ***acc_number_anon***: número de acción anonimizada.
- ***type***: tipo de prueba realizada.
- ***time_capture***: fecha de realización de la prueba.
- ***priority***: prioridad (normal o urgente).
- ***referral_service***: servicio que solicita la prueba.
- ***clinic_information***: información clínica. No es relevante para el proyecto.
- ***report***: informe elaborado por un radiólogo. Es la información sobre la que se basa todo este proyecto.
- ***sex***: sexo del paciente (F o M).
- ***time_birth_date***: fecha de nacimiento.

- ***type_SERAM***: código que identifica la prueba para la Sociedad Española de Radiología.
- ***patient_origin***: servicio del que proviene el paciente (consultas externas, hospitalización, urgencias)
- ***performed_location***: código que indica el lugar en el que se ha elaborado la prueba, y la prueba realizada. Por ejemplo TJRM indica que se ha realizado una resonancia magnética en el Hospital del Tajo.
- ***referral_institution***: nombre del centro que solicita la prueba y que puede ser un hospital o un ambulatorio, y no tiene por qué coincidir con el hospital en el que se ha realizado la prueba.
- ***institution***: hospital asignado al centro de referencia.
- ***time_report***: fecha en la que se ha realizado la prueba.

Campos agregados en el proyecto:

- ***center***: hospital donde se realiza la prueba. Se extrae del campo "performed_location".
- ***test***: prueba realizada. Se extrae del campo "performed_location".
- ***id***: número identificativo propio del informe.

4.2. CARGA DE INFORMES EN R

Los datos suministrados por la Unidad Central de Radiodiagnóstico se encontraban en formato CSV (del inglés C*omma-Separated Values*) con el punto y coma como separador, por lo que la primera tarea que se llevó a cabo fue la de sustituir el punto y coma por un separador más universal como es la barra vertical (|) o tubería (*pipe* en inglés). Este proceso se realizó modificando el separador de listas de Windows, que por defecto tiene establecido el punto y coma, lo que hace que

Microsoft Excel utilice dicho símbolo para separar las columnas. Aunque pueda parecer un proceso rudimentario no requiere ningún tiempo de proceso porque es un simple cambio de configuración, mientras que hacer la sustitución con R haciendo una búsqueda secuencial de dichos signos, puede llevar varias horas y también puede provocar que se sustituyan algunos signos que no son separadores de listas y por tanto puedan provocar que los datos se queden corruptos.

El archivo válido definitivo contenía los 554.173 informes correspondientes al año 2013 y ocupaba 408.888 KB en disco.

4.2.1. Tipos de datos

Todos los datos importados del archivo CSV se trataron como cadenas de texto en R, salvo las fechas, para las que se creó un nuevo tipo de datos específico para fechas con el formato dd/mm/aaaa, ya que el tipo de datos estándar para fechas de R utiliza un formato diferente y provocaba errores en la importación. Para los nuevos campos que se agregaron también se utilizó un tipo de datos para cadenas de texto.

4.2.2. Campos agregados

Para realizar la mayoría de las tareas que se pretendían llevar a cabo en este proyecto, era necesario trabajar con el hospital de referencia y saber a qué tipo de prueba se había sometido el paciente. Dicha información no se encontraba en ningún campo específico sino que se encontraba codificada en otro campo, por lo que realizar su extracción durante los procesos de *clustering* por ejemplo, aumentaba el tiempo de todo el proceso en un 30%, así que se optó por extraer esta información una única vez y almacenarla por separado en dos campos individuales que se podían consultar en cualquier momento. Uno de estos campos almacena el hospital en el que se realiza la prueba y el otro almacena el tipo de prueba realizada. El campo *id* se agregó porque al modificar estructuras con el *Corpus* y la *Term Document Matrix*, los datos sufrían una

nueva ordenación y se perdía el orden original, por lo que se agregó este valor correlativo también durante la carga de datos.

4.2.3. Almacenamiento de los datos en un *Data Frame*

Los datos se cargaron en un *data frame* desde el archivo CSV. En un primer momento se realizó primero la carga de datos, se crearon los dos campos nuevos indicados en el apartado anterior y posteriormente se recorrieron todos los informes extrayendo la información del hospital y de la prueba. Finalmente se convirtieron todos los textos de los informes a letra mayúscula y se sustituyeron todas las letras acentuadas por letras sin acentuar. El problema de realizarlo de esta manera es que el tiempo total de carga podía llegar a los 4 minutos, 1 para la carga, 2 para la extracción de la información y 1 minuto para el tratamiento de los textos de los informes. A pesar de utilizar un disco SSD, el tiempo de lectura de disco sigue siendo alto (en equipos con disco duro, el tiempo total superaba los 6 minutos) y el procesador del equipo no se encuentra a pleno rendimiento.

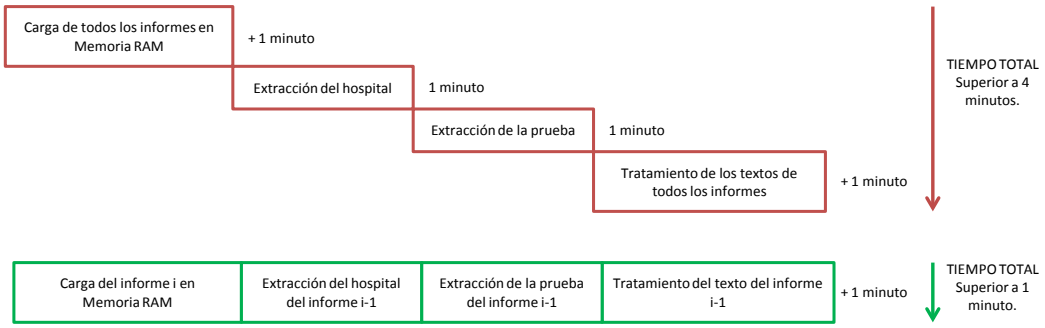


Figura 4-1: diferencia entre los tiempos de proceso de los dos métodos

Por este motivo se buscó un método para aprovechar el tiempo de lectura de disco de cada informe para utilizar el procesador para extraer la información del hospital y de la prueba, pero esta tarea requería la modificación del código fuente de la función de

importación de datos que posee R, sin embargo, utilizando una función propia, se logró que al realizar la importación de cada informe se llamara a dicha función para que el procesador realizara la extracción de la información del informe leído previamente y que ya se encontraba en memoria RAM, mientras se leía el siguiente informe de disco a memoria RAM. Dado el buen funcionamiento de este sistema, también se aprovechó para transformar todos los textos de los informes a letra mayúscula. Una vez realizado esto, el tiempo se redujo notablemente, realizando toda la carga, extracción y tratamiento del texto en poco más de 1 minuto. En la figura 4-1 se muestra un esquema de los dos procesos.

4.2.4. Eliminación de informes irrelevantes

Antes de nada hay que indicar que ningún informe es irrelevante ya que todos aportan información importante, pero dependiendo de cada caso, puede que interesen unos informes u otros. Todas las tareas que se han llevado a cabo en este proyecto se han centrado en enfermedades o problemas existentes, por lo que todos los informes que han dado un resultado negativo en la prueba realizada, para nuestro caso carecían de relevancia. Para eliminar estos informes y así reducir el número de ellos en los procesos y por tanto también el tiempo de proceso, en un primer momento se optó por analizar la posibilidad de realizar un análisis semántico de los informes para intentar identificarlos, debido a que no existe ningún campo que indique si la prueba ha dado un resultado positivo o negativo, sin embargo, después de tener una reunión con personal de la UCR, se extrajeron todos los informes que tuvieran menos de 20 palabras. Después de analizar unos 400 informes, todos ellos tenían un resultado negativo. El motivo viene dado porque en una prueba en la que no se encuentra nada raro o que da un resultado negativo, el texto del informe no aporta ningún detalle especial, para lo cual no se utilizan más que unas pocas palabras. En los casos en los que un paciente se somete a una misma prueba que dio positivo en una prueba anterior, el informe se hace desde cero y haciendo referencia a la anterior prueba, es decir, que en estos informes no se hace un simple añadido, por lo cual, un

paciente que dio positivo en una prueba y sigue dando positivo en las pruebas siguientes, su informe contiene un texto elaborado y explicado y por tanto es considerado como positivo.

El número de informes que se descartaron por tener menos de 20 palabras fue de 175.506, lo que corresponde a un 31% del total, dejando 378.667 informes con resultado positivo, un 69% del total, y con los que se llevaron a cabo los diversos estudios de este proyecto.

4.2.5. Generación del Corpus

Como ya se explicó en el apartado 3.3, una de las estructuras de R necesarias para utilizar funciones de minería de textos es el corpus. A la hora de la creación de esta estructura también se realizaron las siguientes operaciones con los textos de los informes [46],[47]:

- Eliminación de las *stopwords*: así se llaman las palabras que no aportan información importante a la hora de realizar minería de textos, como por ejemplo los artículos, las preposiciones, algunos adjetivos o algunos adverbios. R, a través de la librería TM, ofrece un conjunto de *stopwords* para el idioma español, pero se consideró que no estaba bien elaborada por lo que se utilizaron las *stopwords* en español de Google Code [48].
- Eliminación de espacios en blanco extra: todas las apariciones de más de un espacio en blanco seguido se sustituyeron por un solo espacio en blanco. El motivo de realizar esto es porque algunas funciones de tratamiento de textos pueden no funcionar correctamente al encontrarse con más de un espacio en blanco seguido.
- Eliminación de símbolos de puntuación: debido a que en las tareas de *clustering* solo se tienen en cuenta las palabras existentes, se optó por eliminar todos los símbolos de puntuación, ya que no aportaban nada importante para el caso. Para lo único que podrían haber sido interesantes hubiera sido para el

análisis semántico de los textos en la búsqueda de las reacciones alérgicas, sin embargo, como se verá más adelante, no tenían un aporte semántico relevante y se tomó la decisión de eliminarlos.

El propio corpus tiene una opción para eliminar estos símbolos de forma automática pero sin saber exactamente cuál era el motivo, el símbolo punto no se eliminaba de forma correcta y hubo que recurrir a eliminarlo de otra manera.

- Eliminación de los números: en un principio se optó por no eliminar los números de los textos ya que se consideró que sería importante tener en cuenta los números. No es lo mismo un tumor de 1 centímetro que de 5, pero dada la complejidad que esto aportaba al análisis, se decidió eliminar los números. Para un estudio futuro podría ser interesante analizar la importancia de estos valores y ver la relación entre el potencial que puede aportar y el incremento de proceso para su tratamiento.

4.2.6. Creación de la Term Document Matrix

La Term Document Matrix (TDM por sus siglas en inglés) así como su traspuesta, la estructura Document Term Matrix (DTM por sus siglas en inglés), ya fueron mencionadas en el apartado 3.3. Son estructuras necesarias para la aplicación de ciertas funciones de minería de textos. Para almacenar la información en estas estructuras es necesario tener la información almacenada en una estructura corpus, motivo adicional para haber utilizado un corpus.

A la hora de crear una TDM se puede especificar la frecuencia absoluta mínima y máxima de los términos a guardar en la TDM, sin embargo para la mayoría de tareas se específico 1 como frecuencia mínima e infinito como frecuencia máxima.

Otro de los parámetros que se pueden especificar a la hora de crear la TDM es la longitud mínima y máxima de las palabras a agregar. Aunque después de haber

eliminado las *stopwords* no sería necesario eliminar nada más, se pasó a eliminar todas aquellas palabras con menos de 3 letras.

4.3. ANÁLISIS DE TÉRMINOS

El análisis de términos que se ha realizado ha consistido en una búsqueda de apariciones de términos, tanto relativa como absoluta, y una búsqueda de asociaciones de términos.

4.3.1. Frecuencia de aparición de términos

El recuento de palabras que aparecen en un informe o en el total de ellos puede ofrecer información muy valiosa. Si por ejemplo se analiza en cuántos informes aparece al menos una vez la palabra "litiasis" (frecuencia relativa) podríamos saber cuántos informes han sido elaborados a partir de una prueba de litiasis. Si buscamos información sobre una espondilolistesis de columna, y analizamos la prueba que se ha realizado, se puede saber por ejemplo cuántas resonancias y cuantas radiografías se han realizado por una espondilolistesis, comparar, y hacer un cálculo económico por el coste que supone cada prueba, tanto a nivel económico como por tiempo de redacción de un informe por parte de un radiólogo. Si a estos datos se añade además el hospital que ha realizado la prueba, esto puede dar lugar a información de gran valor para la UCR como se verá en el caso de la búsqueda de reacciones alérgicas.

Otro de los puntos fuertes de las apariciones de términos no es buscar información concreta, sino buscar los términos que más aparecen dentro de un grupo de informes. Esa búsqueda en muchas ocasiones orienta sobre los pasos que se pueden dar en otros tipos de búsquedas.

Dentro de estas búsquedas hay que distinguir entre la frecuencia relativa y la absoluta. La primera de ellas contabiliza en cuántos informes aparece una palabra al menos una vez mientras que la segunda contabiliza todas y cada una de las veces que aparece un término en un informe. Si por ejemplo existen dos informes y en el primero de ellos aparece la palabra reacción una vez y en el segundo aparece dos veces, la frecuencia relativa será 2 y la absoluta 3.

Para mostrar información sobre estas frecuencias se han utilizado diferentes gráficos que se muestra a continuación:

4.3.1.1. Nube de palabras

Se trata de un gráfico muy útil que de forma rápida muestra la frecuencia absoluta de todos los términos con los que se ha trabajado. Este gráfico muestra con diferentes tamaños y colores un conjunto de palabras en función del número de apariciones. Dado que la nube de palabras utiliza una TDM, es a la hora de crear esta TDM cuando se establecen las frecuencias mínimas y máximas.



Figura 4-2: nube de palabras de un conjunto de informes

En la figura 4-2 se muestra un ejemplo de una nube de palabras elaborada a partir de un conjunto de informes donde se puede observar cómo los términos que más aparecen son “alteraciones”, “tamaño”, “estudio” y “normal”, que puede que no aporten mucha información, sin embargo si se sigue analizando el gráfico hay palabras

como “mama”, “tórax” o “riñones” que sí que pueden dar pie a realizar búsquedas más concretas sobre un cáncer de mama o un cáncer de pulmón. Por ejemplo partiendo de este gráfico se podría trabajar con la TDM extrayendo todos los informes que incluyeran la palabra “mama” y volver a crear otra nube de palabras y así ver qué términos son los que más aparecen dentro de este grupo de informes. En la figura 4-3 se muestra un ejemplo.



Figura 4-3: mapa de palabras de un conjunto de informes en los que todos contenían la palabra “mama”

En la figura 4-4 se muestra otro gráfico utilizado para mostrar las frecuencias relativas. En este caso se ha utilizado un conjunto de informes en el que todos ellos incluyen la palabra “reacción”. Como se puede ver en el gráfico, el 90% de esos informes también incluyen la palabra “alérgica”, y el 70% incluyen la palabra “contraste”.

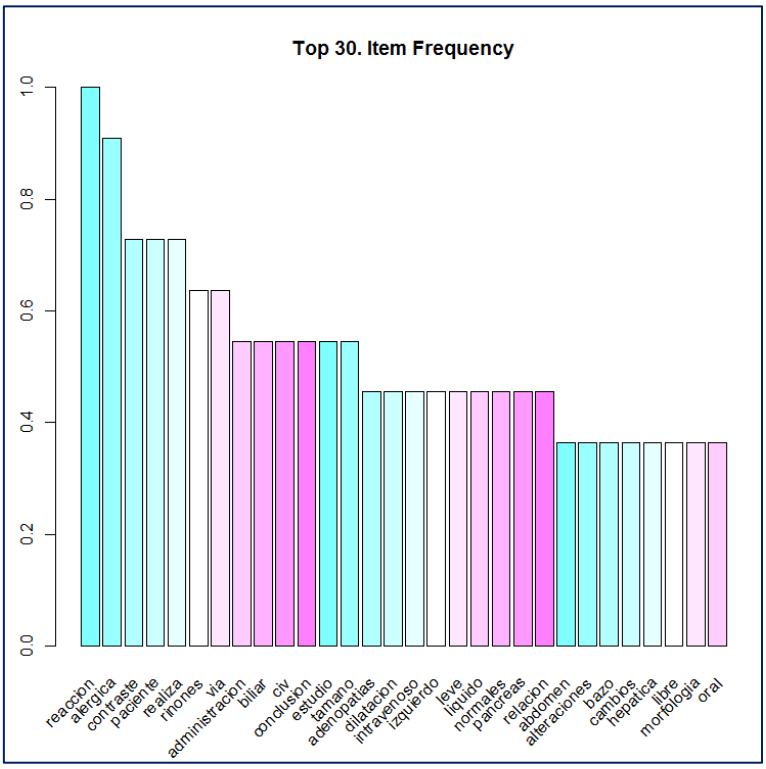


Figura 4-4: gráfico de barras que muestra la frecuencia relativa

4.3.2. Asociaciones de términos

Las asociaciones de términos son un punto fundamental en la minería de textos desestructurados ya que muestra las relaciones existentes entre diferentes términos, dando lugar a asociaciones que a priori pueden ser desconocidas. Los resultados devueltos siguen la estructura siguiente: "si aparece la palabra *x* o aparecen las palabras *x* e *y*, entonces aparece la palabra *z*". En la figura 4-5 y en la figura 4-6 se muestran dos gráficos utilizados para el análisis de relaciones entre términos. En el primer gráfico se puede observar que las palabras "reacción" y "alérgica" son las que más aparecen, representado por el tamaño de los círculos, sin embargo "reacción" no se relaciona con otros términos mientras que "alérgica" sí. Por ejemplo si aparecen las palabras "alérgica" y "paciente" entonces aparece la palabra "realiza".

Además, al aparecer "paciente" también aparece la palabra "contraste". El segundo gráfico muestra la misma información pero en forma de tabla.

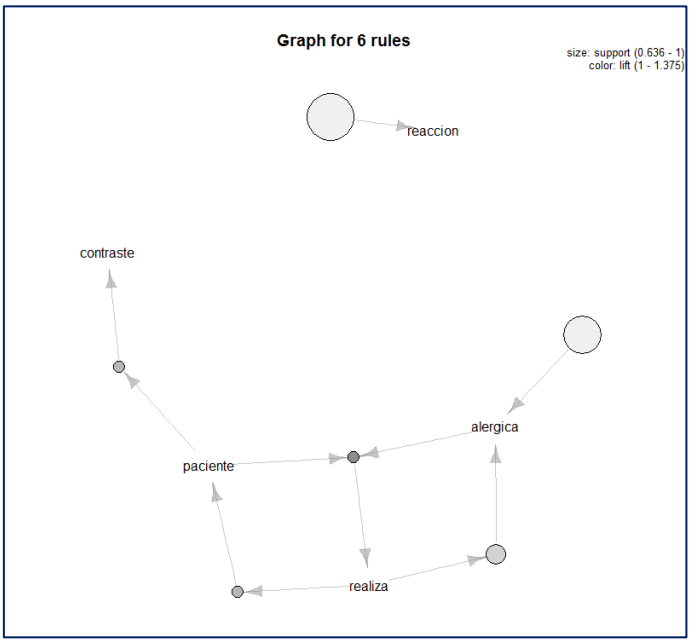


Figura 4-5: gráfico de relaciones entre términos con “reacción” y “alérgica”

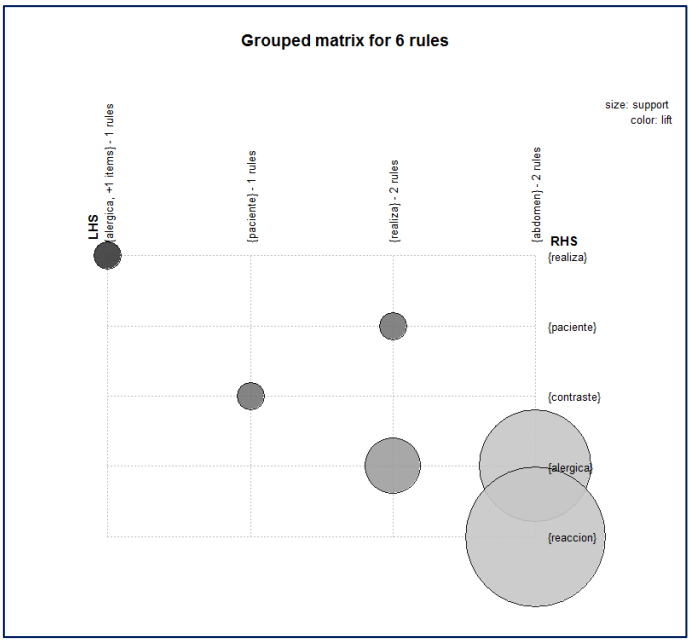


Figura 4-6: gráfico de relaciones entre términos con “reacción” y “alérgica”

4.4. CASO DE USO: REACCIONES ALÉRGICAS

A lo largo de este documento se ha mencionado en diversas ocasiones el tema de las reacciones alérgicas ya que gran parte del tiempo dedicado se ha centrado sobre este trabajo, el cuál surge a partir de una necesidad de la UCR, debido a que con los medios de que disponían no eran capaces de obtener cuántos pacientes habían sufrido una reacción alérgica al contraste utilizado durante la prueba. Este dato es de suma importancia ya que hay personas que han fallecido en el propio hospital durante la administración del contraste debido a la reacción alérgica sufrida. Es muy raro que surja una complicación de esta gravedad, lo normal es que si se sufre una reacción, esta sea de carácter leve, sin embargo, según comentaba el director de la UCR, la necesidad de tener ese dato viene también por la sospecha que tenían en relación a que el número de alergias iban en aumento.

Por otro lado, también es importante el tipo de contraste que se le aplicó al paciente y que el radiólogo debería haber indicado en el informe, ya que permite realizar una estadística sobre los tipos de contraste, que principalmente son tres [49]:

- Contraste yodado: es intravenoso y se utiliza en las pruebas de rayos X y tomografía axial computarizada (TAC por sus siglas).
- Gadolinio: es intravenoso y se utiliza en las pruebas de resonancia magnética nuclear (RM o RMN por sus siglas).
- Sulfato de bario: se toma por vía oral o rectal y se utiliza en las pruebas de rayos X y TAC.

4.4.1. Lenguaje utilizado en los informes

Es importante recalcar que este proyecto se ha basado en informes radiológicos que están escritos en español y por personal de radiología, porque si no fuera así, los procesos no tendrían por qué funcionar, ya que no era objetivo de este proyecto elaborar algoritmos universales y adaptables al idioma.

Que los informes estén escritos por radiólogos también es un punto importante a tener en cuenta ya que se puede decir que tienen un lenguaje propio para escribir los informes. Esto tiene su explicación ya que los radiólogos tratan de escribir un lenguaje que sea entendido por su entorno de forma rápida. No tratan de escribir una novela o de hacer una redacción atractiva, tratan de describir el resultado de la prueba con el menor número de palabras posible pero sin perder información, y la mejor manera de hacer esto es que todos escriban más o menos igual, utilizando los mismos sustantivos, verbos y adjetivos para hacer referencia a lo mismo, sin recurrir a sinónimos que puedan desviar el objetivo de la redacción del documento. Todo esto ayuda en gran medida a la hora de tratar los textos, pero evidentemente los métodos utilizados no tienen por qué funcionar en textos escritos por un veterinario o un topógrafo.

Indicar que a día de hoy existen proyectos abiertos en diversos organismos mundiales para crear una estandarización en los métodos para redactar los informes, lo que abriría un nuevo foco de trabajo para el tratamiento de los textos de radiodiagnóstico. Desde luego, una de las cosas que puede dejar claro el trabajo realizado en este proyecto es que si los textos estuvieran estructurados, el trabajo sería más sencillo y rápido.

4.4.2. Proceso de obtención de informes con reacción alérgica

El primer paso que se dio para buscar estos informes fue el de buscar cuántos informes contenían las palabras “alergia”, “alérgica” y derivadas de estas, ya fueran en solitario o relacionadas unas con otras. Por otro lado también se realizaron búsquedas de informes que tuvieran la palabra “reacción”, tanto en solitario como relacionadas con las palabras “alergia” y “alérgica”. Finalmente se realizaron búsquedas con palabras como “contraste”, “yodo”, “yodado”, “gadolinio”, “bario”, etc.

Utilizando algoritmos de asociaciones de términos como los comentados en el apartado 4.3.2 se obtuvieron los siguientes resultados:

1. La palabra “alergia” no mostraba ningún tipo de asociación con los informes que indicaban una reacción alérgica durante la prueba, ya que aparecía en algunos de estos informes y en otros que hablaban de otros tipos de alergias, como alergia al polen, a los frutos secos, etc. Dado que no se encontró ningún tipo de relación única con los informes buscados, se descartó el uso de esta palabra.

2. La palabra “reacción” existía en 640 informes, demasiados para tener algo que ver exclusivamente con una reacción alérgica.

3. Al realizar una búsqueda de la palabra “alérgica” (72 informes) se obtuvieron dos grupos de informes: el primero, un 5% nada más, informaba sobre alergias que no tenían que ver con la prueba; en el 95% restante todos los informes explicaban que se había producido una reacción alérgica durante la prueba, sin embargo, en algunos de ellos, el radiólogo comentaba que no iba a proceder con la administración del contraste debido a que el paciente había indicado que en alguna prueba anterior había sufrido una reacción. Esta búsqueda por tanto se acercaba bastante a lo buscado ya que no se perdía ningún informe sin reacción alérgica, pero incluía informes que no se debían de tener en cuenta, como los que hacían referencia a otras alergias, o los que indicaban que el paciente había tenido una reacción alérgica anterior, ya que esto en sí no es una reacción en dicha prueba, si no en otra anterior, o incluso nunca la ha tenido ya que el paciente puede haberse equivocado, o incluso tener miedo a sufrir una reacción y mentir sobre la reacción sufrida.

4. Realizando pruebas de asociaciones entre términos se observó que la palabra “reacción” tenía una asociación relevante con una única palabra, “alérgica”. Esta relación se puede observar en la figura 4-5 y en la figura 4-6.

5. A la vista de los resultados anteriores y de las relaciones entre “reacción” y “alérgica”, se realizó una búsqueda de la cadena “reacción alérgica”, que devolvió 35 resultados que eran exclusivamente informes que indicaban una reacción alérgica, ya sea en la propia prueba o en otra anterior.

6. Una vez realizado esto se procedió a crear el corpus y la TDM de este caso y del explicado en el punto 2 y se observó que realizando una búsqueda de la cadena "reacción alérgica" antes de crear el corpus se perdían informes, ya que algunos radiólogos habían escrito un símbolo de puntuación o varios espacios seguidos entre las palabras "reacción" y "alérgica". También se perdía algún caso en el que el radiólogo añadía una palabra. Observado esto se optó por no buscar la cadena "reacción alérgica" y buscar solo informes con la palabra "alérgica" que reducía el número de informes de 378.667 a 72, lo que permitía crear el corpus y la TDM de forma casi instantánea y con mejores resultados que a partir de la búsqueda de la cadena completa.

7. Creado el corpus y la TDM de estos 72 informes, se buscaron aquellos que tuvieran la palabra "reacción". El resultado devuelto fue de 35 informes, es decir, que 35 informes contenían las palabras "reacción" y "alérgica" y todos ellos hacían referencia a una reacción alérgica en la prueba o en un caso anterior sin falsos positivos.

8. Este resultado fue corroborado por la UCR y por un lingüista, ya que según comentaron, un radiólogo no debería referirse a este tipo de situaciones utilizando "reacción" sin la palabra "alérgica", ya que por sí sola, la palabra "reacción" no aporta información suficiente y necesita de un adjetivo para complementarse, que en este caso es "alérgica", y que por suerte carece de sinónimos que aporten una información similar al sentido de la frase empleada.

9. Una vez que se lograron obtener todos los informes que indicaban una reacción alérgica, 35 en total durante el año 2013, se realizó una revisión manual de cada uno de ellos. Solamente faltaba buscar un método para descartar aquellos en los que no se aplicaba contraste por haber sufrido el paciente una reacción alérgica en una prueba anterior, aunque en realidad no era imprescindible ya que el número de informes no era elevado para su consulta manual.

4.4.3. Eliminación de informes con reacción alérgica previa

Este punto no era algo imprescindible, porque revisar manualmente un número de informes tan reducido no suponía mucho tiempo para la UCR, sin embargo, con el objetivo de intentar automatizar todo el proceso, finalmente sí se llevo a cabo. Los pasos que se realizaron fueron los siguientes:

1. Se separaron de forma manual en dos grupos. En el primero se agruparon los informes que hacían referencia a una reacción durante la prueba y en el segundo aquellos en los que el paciente era quien informaba al radiólogo de la reacción en otra prueba anterior. En total había 22 informes en el primer grupo y 50 en el segundo.
2. Se analizaron los dos grupos y se comparó la semántica de los informes. Se observó cómo en todos los informes la información relevante a la reacción alérgica se encontraba en la frase donde se encontraban las palabras "reacción" y "alérgica" o en las seis o siete palabras inmediatamente anteriores o posteriores a esta cadena. Este descubrimiento tiene lógica ya que aunque cada radiólogo tenga un lenguaje propio, en principio siempre técnico, realiza sus explicaciones sobre lo ocurrido en lo que rodea a las palabras "reacción" y "alérgica". Por este motivo los textos de los informes se truncaron a los caracteres comprendidos desde los 45 anteriores a la palabra alérgica y los 65 posteriores.
3. Una vez realizado esto, se analizó la semántica de estas cadenas de 120 caracteres y se descubrió que los radiólogos en la mayoría de los informes empleaban el verbo "referir" para indicar que era el paciente quien informaba de una reacción anterior. Comentando este punto con la UCR, confirmaron que es el verbo que suelen utilizan cuando un paciente quiere informar de algo, y como ya se ha comentado anteriormente, los radiólogos intentan no recurrir a sinónimos y así facilitar la lectura de los mismos.

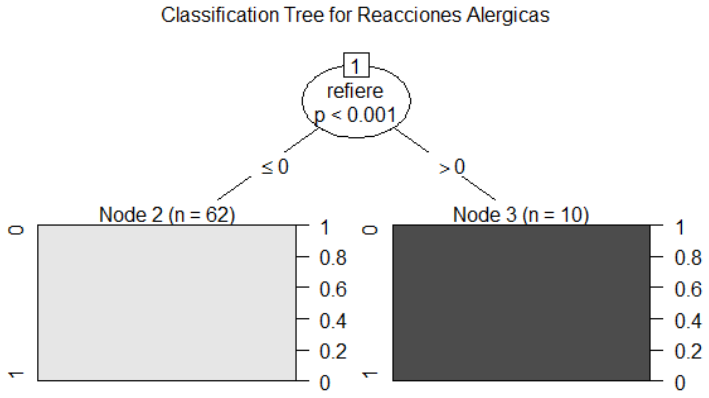


Figura 4-7: árbol de inferencia realizado con RWeka para el análisis de la palabra "refiere"

En la Figura 4-7 se muestra el árbol de inferencia realizado con RWeka sobre la palabra "refiere" en dichos informes, donde se puede observar que el valor de corte es 0 y la distribución de informes con esa palabra. En cualquier caso, para intentar alcanzar el 100% de los informes, se siguieron analizando los textos buscando palabras que aportasen información exclusiva sobre una reacción previa, siempre dentro de los 120 caracteres, y se observó que además del verbo "referir" también se solían utilizar palabras como "previa" o "previas" y "antecedente" o "antecedentes", así como la cadena "no administra", que en realidad sería la cadena "no se administra" pero el pronombre "se" se eliminó como *stopword* cuando se creó el corpus. Descartando por tanto todos los informes que incluyeran alguna de estas palabras en los caracteres que rodeaban a las palabras "reacción" y "alérgica" solo se obtuvo 1 falso positivo. En la tabla 4-1 se muestra la matriz de confusión del total de los 554.173 informes, donde se observan los buenos resultados obtenidos sin falsos negativos y un solo falso positivo. Este falso positivo se trataba de un informe que hacía referencia a una reacción alérgica que había sufrido el paciente ese mismo día, por lo que se puede considerar un caso aislado y que por ello no ha cumplido las condiciones que eliminan estos informes.

MATRIZ DE CONFUSIÓN

		PREDICCIÓN	
		NEGATIVO	POSITIVO
REAL	NEGATIVO	554.150	1
	POSITIVO	0	22

Tabla 4-1: matriz de confusión del total de informes y las reacciones alérgicas encontradas

Como conclusión, indicar que la exactitud de los resultados es:

$$EX = (554.150 + 22)/(554.150 + 1 + 0 + 22) = 0{,}9999 = 99{,}99\%$$

Y que la precisión de los resultados es:

$$PR = 22/(1 + 22) = 0{,}9565 = 95{,}65\%$$

4. Finalmente y como último análisis, se realizó el mismo proceso utilizando el texto completo del informe pero dado que un informe normalmente contiene de media unas 300 palabras, aumentaba la probabilidad de que las palabras buscadas aparecieran por el resto del informe haciendo referencia a otro tema que no fuera la reacción alérgica, por lo que se descartaban informes válidos. En concreto se obtenían 17 más el mismo falso positivo que en el caso anterior, es decir, 17 frente a 22.

4.4.4. Resumen del proceso

Para dejar claro cuál ha sido el proceso seguido, se escribe a continuación un resumen de todos los pasos seguidos.

1. Carga de los 554.173 informes correspondientes al año 2013.
2. Eliminación de informes con menos de 20 palabras, por lo que se quedan 378.667 informes.

3. Obtención de todos los informes que contienen la palabra “alérgica”, dando lugar a 72 informes.
4. Reducción del texto de los informes a los 120 caracteres que rodean a la palabra alérgica.
5. Creación del corpus y de la TDM.
6. Obtención de los informes que tiene la palabra “reacción”, lo que deja el número de informes en 35.
7. Eliminación de los informes según lo explicado en el punto 4.4.3.

4.4.5. Obtención del contraste utilizado

En relación a la búsqueda del contraste utilizado durante la prueba, se realizaron diferentes búsquedas en los 22 informes anteriores pero el tipo de contraste no se encontraba dentro de los caracteres cercanos a las palabras "reacción" y "alérgica", por lo que hubo que realizar la búsqueda en todo el texto del informe, lo que impedía obtener resultados correctos porque en algunos informes no se especificaba el tipo de contraste utilizado, simplemente se indicaba si era oral o intravenoso, ya fuera con toda la palabra o utilizando una abreviatura como CIV (Contraste IntraVenoso), pero estos adjetivos podían aparecer más veces en el texto y podían generar confusión y un alto número de falsos positivos. Aunque existía la posibilidad de buscar un adjetivo cercano a la palabra “contraste”, dada la complejidad del proceso y lo poco que podía aportar, se optó por descartarlo. En la tabla 4-2 se muestran los resultados de los tipos de contraste encontrados.

TERMINO	APARICIONES
CIV	6
Endovenoso	1
Gadolinio	1
Intravenoso	2
No se especifica	8
Paramagnético	1
Yodado	4

Tabla 4-2: tipos de contraste encontrados en los informes

4.4.6. Resultados derivados importantes

Un apartado interesante y que aporta información de gran importancia es el análisis de todos los campos y su fusión con datos demográficos. Dado que los datos suministrados por la UCR están anonimizados, ciertas pruebas no se pueden realizar, pero con campos como el sexo, la prueba o el hospital de referencia sobre todo, se pueden obtener datos que requieren atención especial y que se detallan a continuación:

1. De los 22 informes, 11 corresponden a mujeres y 11 a hombres.
2. De las 22 pruebas, 13 corresponden a TAC, lo que en un principio puede tener lógica ya que el TAC es una prueba que se suele hacer con contraste en más ocasiones que el resto de pruebas.

PRUEBA	REACCIONES
TAC	13
RM	3
RX	1
OTROS	5

Tabla 4-3: reacciones alérgicas distribuidas por tipo de prueba

3. Y el dato más importante: de las 22 reacciones alérgicas, 9 se produjeron en el mismo hospital, mientras que el resto se repartieron por el resto de hospitales. En la siguiente tabla se puede ver el resultado, en la cual, por motivos evidentes de confidencialidad, no se especifica el nombre del hospital en el que se produjeron las reacciones alérgicas.

HOSPITAL	REACCIONES
H1	1
H2	2
H3	2
H4	3
H5	5
H6	9

Tabla 4-4: reacciones alérgicas distribuidas por hospital

Este dato lleva a analizar si existe algún motivo por el cual se obtiene dicho resultado o es simple casualidad. Hay que tener en cuenta que solo se han analizado informes de un año, por lo que la casualidad puede jugar un papel importante, pero también hay que tener en cuenta que por ejemplo existen diferentes contrastes elaborados por diferentes laboratorios y que no todos tienen el mismo precio, lo que lleva a la posibilidad de que en dicho hospital se esté administrando un contraste más económico pero de peor calidad y el cual provoque más reacciones alérgicas en los pacientes.

4.5. CLUSTERING DOCUMENTAL

Uno de los objetivos buscados en este punto del proyecto es el de evaluar los resultados de aplicar algoritmos de *clustering* a los informes de reacciones alérgicas obtenidos en el punto anterior, analizando si estos algoritmos pueden separar en diferentes clústeres las reacciones alérgicas producidas durante la prueba o en pruebas anteriores. Otro de los objetivos buscados es el aplicar algoritmos de *clustering* a los informes de radiodiagnóstico en general para analizar si los clústeres devueltos por estos algoritmos son agrupaciones válidas y pueden servir de punto de partida para otros estudios.

4.5.1. Aplicación de algoritmos de *clustering* en informes de reacciones alérgicas

En los siguientes apartados se van a describir los algoritmos de *clustering* utilizados así como los resultados obtenidos, después de haber sido aplicados en los 72 informes devueltos al buscar la palabra “alérgica”, como ya se explicó en el punto 4.4.2. Ya se demostró que el texto relevante sobre la reacción alérgica se encuentra dentro de los 120 caracteres que rodean a la palabra “alérgica”, por lo que será sobre estos textos parciales sobre los que se apliquen los algoritmos.

Hay que recordar que dentro de los 72 informes existen 22 verdaderos positivos y 50 verdaderos negativos. Los algoritmos de *clustering* simplemente dividirán los informes en dos grupos, sin identificar cuál es uno u otro. Esa tarea se tendrá que realizar de forma manual. La matriz de confusión obtenida con el análisis semántico atendiendo exclusivamente a los 72 informes se muestra en la tabla 4-5.

MATRIZ DE CONFUSIÓN

		PREDICCIÓN	
		NEGATIVO	POSITIVO
REAL	NEGATIVO	49	1
	POSITIVO	0	22

Tabla 4-5: matriz de confusión de los 72 informes en el análisis semántico

La exactitud de los resultados es:

$$EX = (49 + 22)/(49 + 1 + 0 + 22) = \ 0{,}9861 \ = \ 98{,}61\%$$

La precisión de los resultados es:

$$PR = 22/(1 + 22) = 0{,}9565 = 95{,}65\%$$

4.5.1.1. Clustering Jerárquico

A continuación se van a detallar los resultados que se obtuvieron al aplicar el algoritmo de *clustering* jerárquico en los 72 informes. Debido a que este algoritmo permite modificar el método de cálculo de distancias, se aplicó el algoritmo con 5 tipos de medición diferentes.

MEDIDA DE DISTANCIAS	GRUPO 1	GRUPO 2
BINARIA	70	2
EUCLÍDEA	70	2
MÁXIMA	71	1
MANHATTAN	69	3
CABERRA	70	2

Tabla 4-6: resultados obtenidos con los algoritmos de *clustering* jerárquico

Como se puede observar en la tabla anterior, independientemente de si el grupo 1 es o no el que hace referencia a las reacciones durante la prueba, los resultados están muy lejos de ser tenidos en cuenta, más aún si se compara con los resultados obtenidos con el análisis semántico. La siguiente imagen muestra el dendrograma obtenido al aplicar el algoritmo jerárquico con la distancia *manhattan*, donde se pueden observar los dos grupos principales que se obtuvieron.

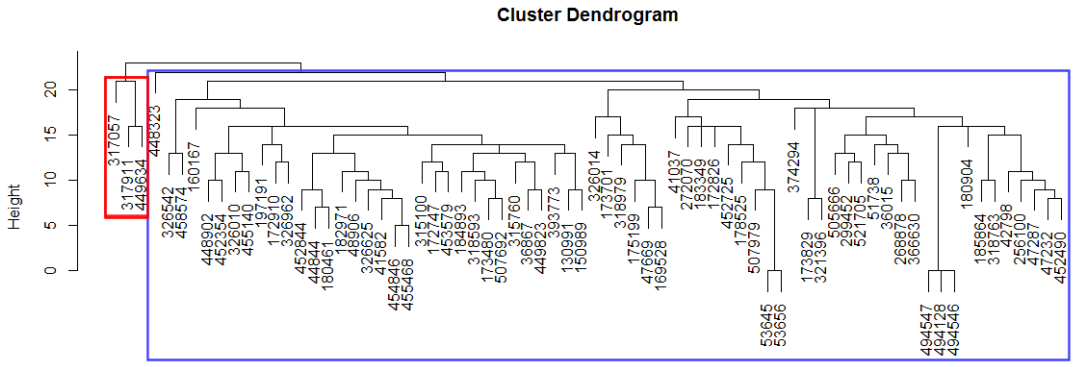


Figura 4-8: dendrograma obtenido con *clustering* jerárquico y distancia *manhattan*

Dejando a un lado el resultado devuelto, observando el dendrograma se aprecian pequeños grupos de 8 o 10 informes pero no se puede observar la existencia de dos grupos claramente diferenciados que son los que realmente se buscan.

4.5.1.2. K-means

A pesar de buscar un número exacto de 2 clústeres, antes de aplicar el algoritmo se realizó un análisis del número óptimo de clústeres que se pueden aplicar [50]. En la figura 4-9 se muestra el gráfico que se obtuvo, donde se busca una disminución pronunciada en la suma de cuadrados antes de la disminución más suave. En este caso, la disminución más pronunciada se produce entre 1 y 2 clústeres, por lo que el número de clústeres óptimo es 2, que es justo lo que se necesitaba.

Procediendo de la misma manera que con el algoritmo jerárquico, se aplicó el algoritmo *k-means* con los 72 informes y 2 clústeres. En la figura 4-10 se muestra el resultado devuelto por el algoritmo, donde se pueden ver dos grupos bien diferenciados, uno de 34 elementos y otro de 38. Aunque cualquiera de los dos se aleja de los 22 positivos reales, se procedió a realizar la matriz de confusión.

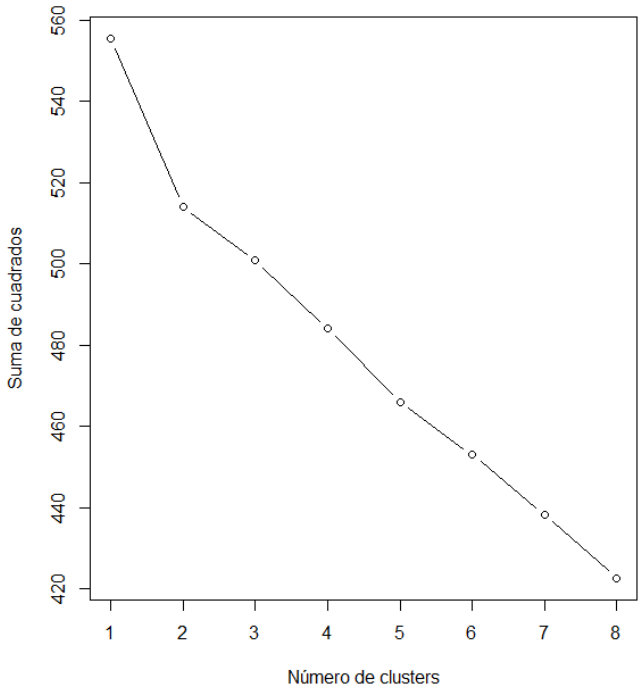


Figura 4-9: gráfico de análisis del número de clústeres a buscar

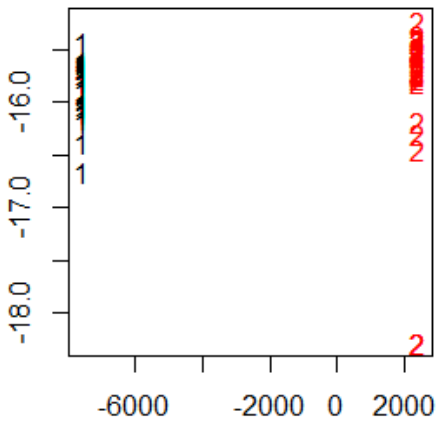


Figura 4-10: resultado de aplicar el algoritmo *k-means* con 2 clústeres

MATRIZ DE CONFUSIÓN

		PREDICCIÓN	
		NEGATIVO	POSITIVO
REAL	NEGATIVO	34	16
	POSITIVO	0	22

Tabla 4-7: matriz de confusión al aplicar el algoritmo *k-means*

La exactitud de los resultados es:

$$EX = (34 + 22)/(34 + 16 + 0 + 22) = 0{,}7778 = 77{,}78\%$$

La precisión de los resultados es:

$$PR = 22/(16 + 22) = 0{,}5789 = 57{,}89\%$$

Como se puede observar, la exactitud del resultado supera el 77% ya que no se ha obtenido ningún falso negativo, o dicho de otra manera, que todos los informes que informaban de una reacción alérgica durante la prueba están dentro del mismo grupo, lo cual es un buen resultado, pero en ese mismo grupo hay 16 informes más que son falsos positivos, por lo que la exactitud no llega al 60%.

4.5.1.3. Fuzzy C-Means

Otro de los algoritmos que se aplicaron fue el algoritmo *Fuzzy C-Means*. Al ser difuso, cada elemento pertenece a todos los grupos pero con un grado de pertenencia, por lo que este algoritmo permite introducir un exponente de pertenencia que permite ajustar el resultado. Como se desconocía a priori un valor apropiado para este exponente, se codificó un pequeño algoritmo para calcular el número de elementos que se asignaban a un clúster en función de dicho exponente. En la figura 4-11 se muestra el resultado de dicho cálculo. El valor que más se acercaba a los 50 elementos era de 52, con un valor de exponente de 1,01.

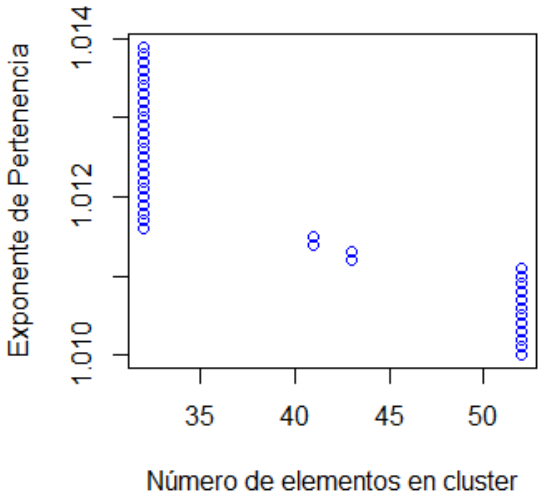


Figura 4-11: relación entre elementos en cluster y exponente de pertenencia

Una vez calculado el exponente, se aplicó el algoritmo con los 72 informes. El resultado se muestra en la siguiente tabla de confusión:

MATRIZ DE CONFUSIÓN

		PREDICCIÓN	
		NEGATIVO	POSITIVO
REAL	NEGATIVO	9	41
	POSITIVO	11	11

Tabla 4-8: matriz de confusión al aplicar el algoritmo Fuzzy C-Means

La exactitud de los resultados es:

$$EX = (9 + 11)/(9 + 11 + 11 + 41) = 0{,}2778 = 27{,}78\%$$

La precisión de los resultados es:

$$PR = 11/(11 + 41) = 0{,}2115 = 21{,}15\%$$

Como se puede observar, los resultados son bastante peores que los devueltos por el algoritmo *k-means*.

Para comprobar cuál fue el criterio seguido por el algoritmo para realizar dicha clasificación [46] se puede observar la Figura 4-12, donde se indica que el primer grupo se creó en base a los informes que utilizaban las palabras "alérgica", "reacción", "contraste" y "paciente" por ser las más repetidas. Como en estas palabras no hay referencia alguna a palabras que relacionen el informe con haber tenido o no una reacción durante la prueba, los resultados devueltos si coinciden son por casualidad.

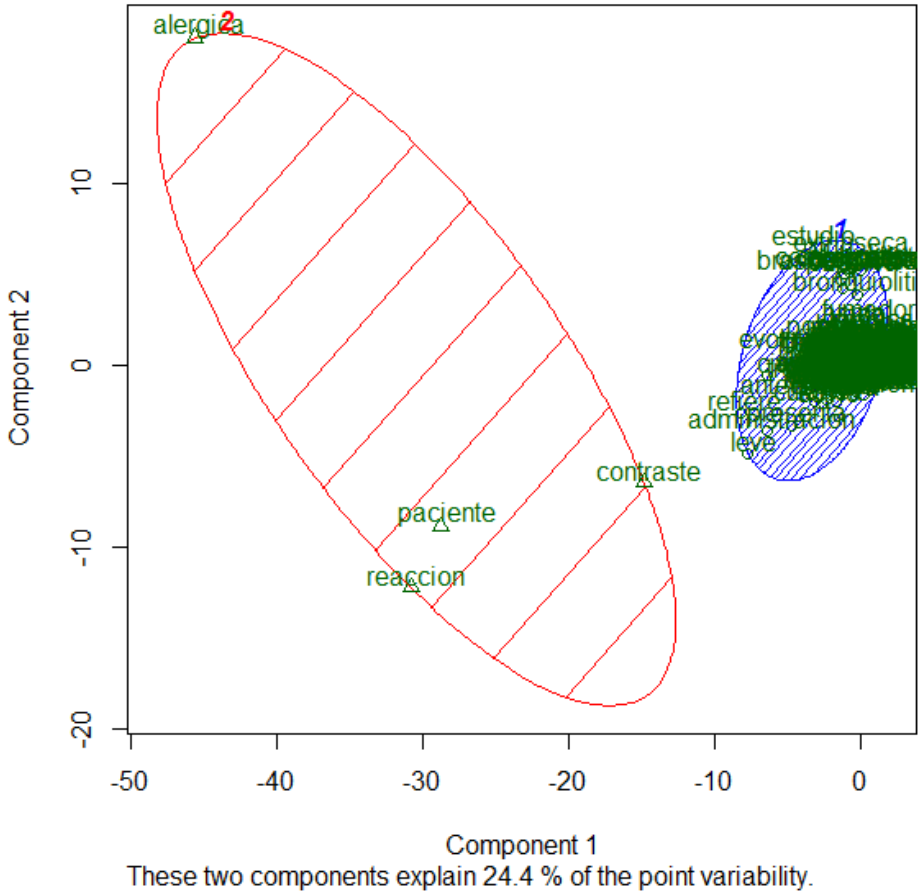


Figura 4-12: gráfico de criterios para la creación de clústeres por el algoritmo Fuzzy C-Means

4.5.1.4. Clustering por Propagación de Afinidad

Otro de los algoritmos con los que se suelen obtener buenos resultados en el campo de la minería de textos, es el *Clustering* por Propagación de Afinidad. Debido a esto se eligió este algoritmo para realizar las mismas pruebas que con los algoritmos anteriores.

Este tipo de algoritmos establecen por sí solos el número de clústeres más apropiados en función de la métrica de distancias que se elija. Después de realizar diversas pruebas y no obtener ningún resultado cercano a lo que se buscaba, se optó por establecer los parámetros recomendados por Brendan J. Frey y Delbert Dueck [51] para la medida de similitud, que aunque mejoraban los resultados previos, tampoco se acercaban a lo esperado. En total, el algoritmo devolvió 8 clústeres, por lo que se decidió analizar su distribución y ver la posibilidad de combinar varios clústeres para intentar obtener un resultado mejor. En la siguiente tabla se muestra la distribución de positivos y negativos reales realizada por el algoritmo en cada clúster:

CLUSTER	ELEMENTOS	POSITIVOS	NEGATIVOS
1	5	0	5
2	4	4	0
3	15	0	15
4	1	0	1
5	1	0	1
6	3	0	3
7	21	6	17
8	20	12	8
TOTAL		**22**	**50**

Tabla 4-9: distribución de elementos en los clústeres por el algoritmo de propagación de afinidad

Como se puede apreciar en la tabla, en los clústeres 1 a 6 solo hay informes de un tipo, sin embargo en los clústeres 7 y 8 están prácticamente todos los positivos mezclados con los negativos, con lo cual, el resultado no merece ser tenido en cuenta.

En la figura 4-13 se muestra un gráfico de calor generado por este algoritmo. En este gráfico se muestra una única zona clara que corresponde al gráfico completo, lo que indica que el algoritmo no ha encontrado grupos claramente definidos. Si así fuera y hubiera encontrado los dos grupos que se buscaban, se observaría un gráfico con cuatro secciones, dos naranjas y dos amarillas.

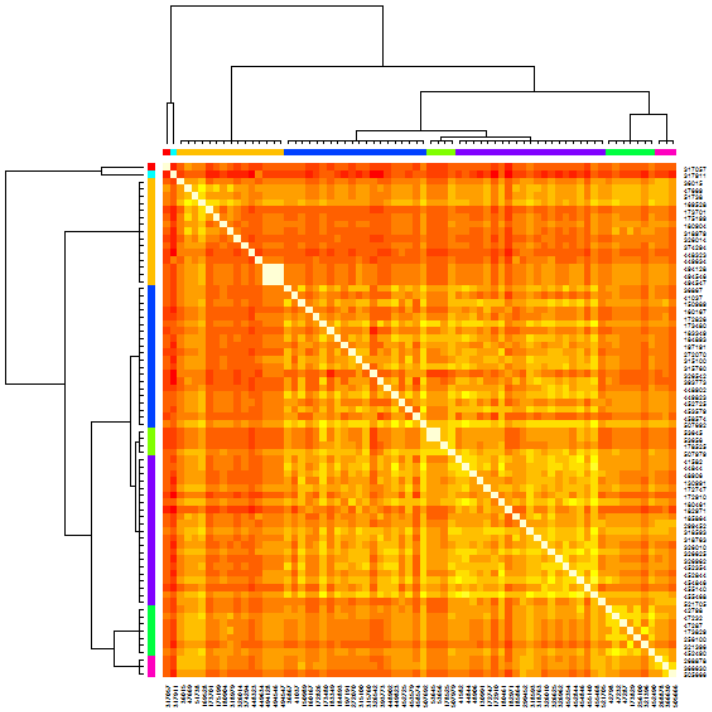


Figura 4-13: gráfico de calor devuelto por el algoritmo de propagación de afinidad

4.5.1.5. Clustering Aglomerativo

El *Clustering* Aglomerativo está basado en el *Clustering* por Propagación de Afinidad y se recomienda utilizarlo [52] en los casos en los que el anterior algoritmo no da buenos resultados, sin embargo, con este algoritmo tampoco se obtuvo un resultado próximo a lo buscado. En la siguiente figura se puede observar el dendrograma devuelto, donde hay 2 elementos en un grupo y 70 en el segundo, y como la división comienza en la parte superior del gráfico. Para que hubiera dos grupos bien definidos, las líneas verticales hasta la primera división deberían tener una

altura que como mínimo ocupara un tercio del gráfico, y sin embargo, dichas líneas son casi inapreciables.

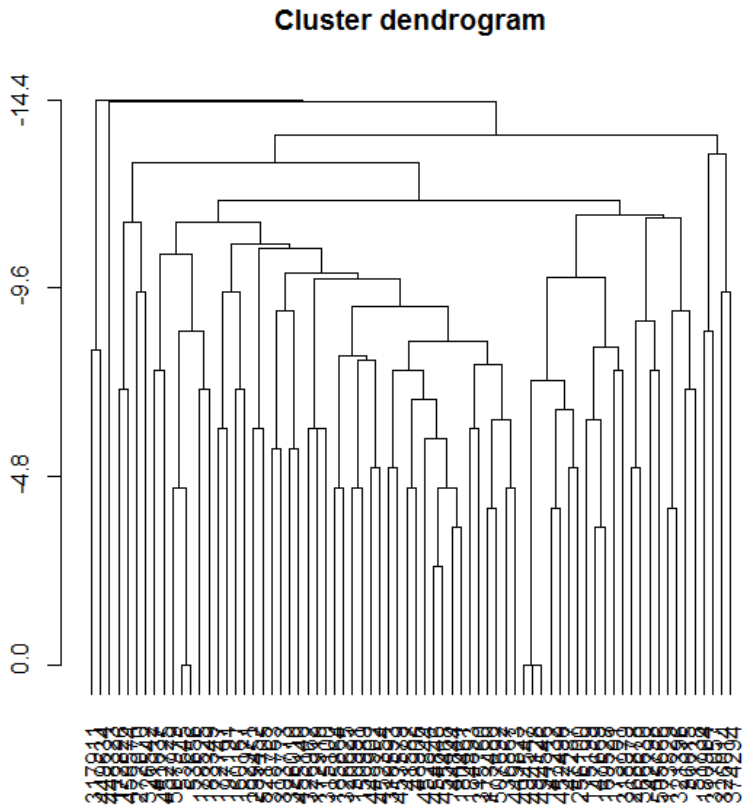


Figura 4-14: dendrograma obtenido con el *clustering* aglomerativo

4.5.2. Aplicación de algoritmos de *clustering* en informes en general

Dado que los resultados obtenidos de aplicar los algoritmos de *clustering* con la intención de separar en dos clústeres los informes de reacciones alérgicas no devolvieron los resultados esperados, se procedió a aplicar los mismos algoritmos en un conjunto de informes aleatorios para analizar los resultados y su posible aplicación en la agrupación de informes por enfermedades. Por lo ya comentado en el apartado 2.2, Limitaciones Y Condicionantes, en este análisis se utilizaron solamente 250 informes aleatorios, ya que el tiempo de proceso con 50.000 informes superaba las 24 horas.

4.5.2.1. Clustering Jerárquico

En la aplicación del algoritmo de *clustering* jerárquico se obtuvieron multitud de datos pero de difícil interpretación debido a la cantidad de clústeres que se crearon.

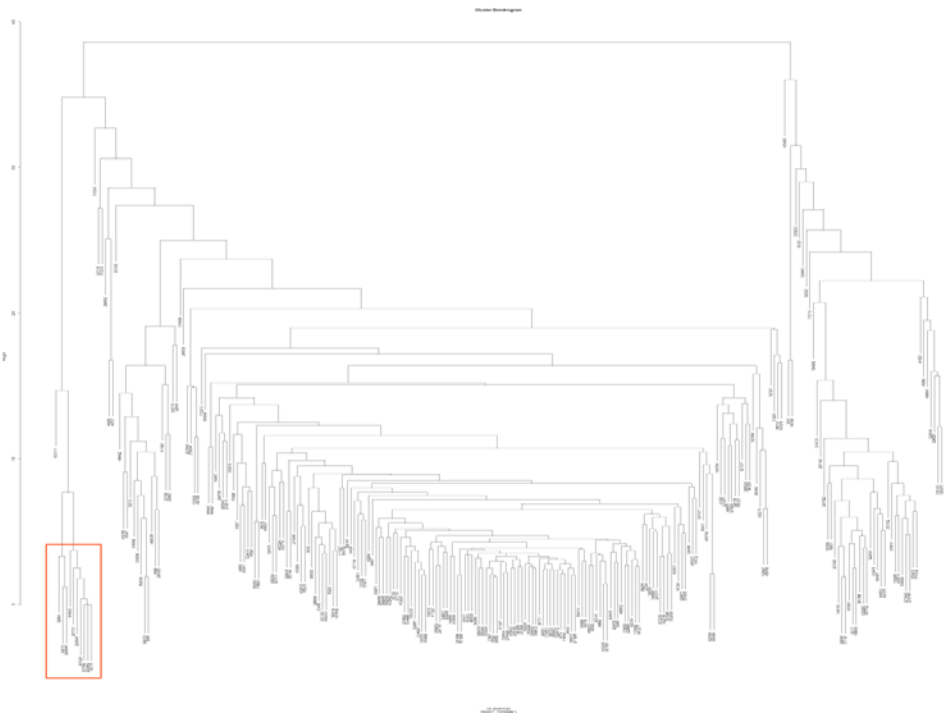


Figura 4-15: dendrograma obtenido al aplicar el algoritmo de *clustering* jerárquico

Observando el dendrograma obtenido, se analizó manualmente el grupo de informes que se encontraba en la parte inferior izquierda, y que estaba compuesto por 10 informes.

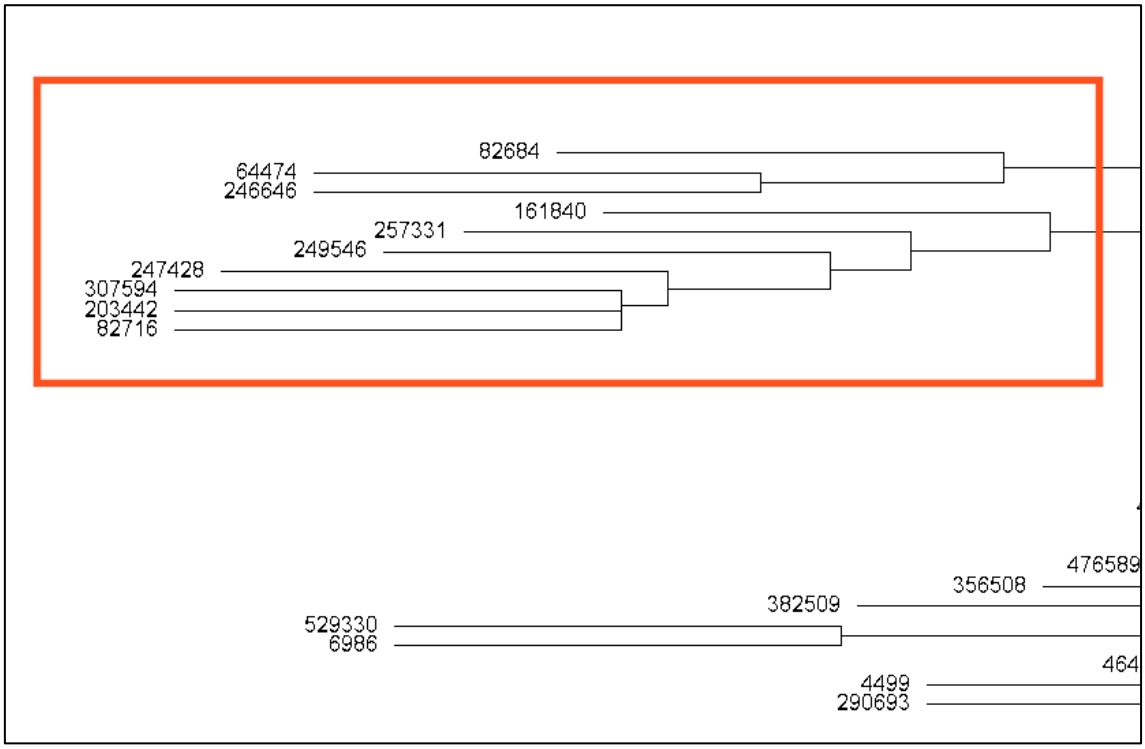


Figura 4-16: zoom del dendrograma obtenido en el *clustering* izquierdo

Los primeros caracteres de estos 10 informes están recogidos en la siguiente tabla:

ID	TEXTO DE INFORME
82684	RESULTADOS DE LA DENSITOMETRIA:ZONA: FEMUR IZQUIERDO FEMUR IZQUIERDO.REGION: CUELLO TOTAL.OMS:
64474	RESULTADOS DE LA DENSITOMETRIA:ZONA: FEMUR IZQUIERDO FEMUR IZQUIERDO.REGION: CUELLO TOTAL.OMS:
246646	RESULTADOS DE LA DENSITOMETRIA:ZONA: FEMUR IZQUIERDO FEMUR IZQUIERDO.REGION: CUELLO TOTAL.OMS:
161840	RESULTADOS DE LA DENSITOMETRIA:ZONA: ANTEBRAZO IZQ. ANTEBRAZO IZQ..REGION: RADIO UD RADIO
257331	RESULTADOS DE LA DENSITOMETRIA:ZONA: COLUMNA AP.REGION: L1-L4.OMS: NORMAL.DMO: 1,131 G/CM2.T-SCOR
249546	RESULTADOS DE LA DENSITOMETRIA:ZONA: COLUMNA AP.REGION: L1-L4.OMS: OSTEOPOROSIS.DMO: 0,678 G/CM2.T-
247428	RESULTADOS DE LA DENSITOMETRIA:ZONA: COLUMNA AP.REGION: L1-L4.OMS: OSTEOPENIA.DMO: 0,962 G/CM2.T-
307594	RESULTADOS DE LA DENSITOMETRIA:ZONA: COLUMNA AP.REGION: L1-L4.OMS: OSTEOPENIA.DMO: 0,970 G/CM2.T-
203442	RESULTADOS DE LA DENSITOMETRIA:ZONA: COLUMNA AP.REGION: L1-L4.OMS: OSTEOPENIA.DMO: 0,900 G/CM2.T-
82716	RESULTADOS DE LA DENSITOMETRIA:ZONA: COLUMNA AP.REGION: L1-L3.OMS: OSTEOPENIA.DMO: 0,887 G/CM2.T-

Tabla 4-10: primeros caracteres de 10 de los informes agrupados por el *clustering* jerárquico

Analizando el contenido de los informes, se pudo observar la similitud existente entre estos informes. Tomando como referencia el dendrograma y analizando el grupo de los informes 82716, 203442, 307594, 247428 se ve como todos ellos son prácticamente iguales en los primeros caracteres. En el caso del informe número 247428 es algo diferente en el resto del texto que lo forma, aunque dicho texto no se muestra en la tabla.

4.5.2.2. K-means

Después de haber analizado el dendrograma devuelto por el algoritmo de *clustering* jerárquico, surgió la dificultad de elegir por ensayo y error el número de clústeres que debe buscar el algoritmo *K-means*, por lo que se recurrió al mismo algoritmo de cálculo del número óptimo de clústeres que se utilizó en la búsqueda de las reacciones alérgicas. En la siguiente figura se muestra el gráfico devuelto por el algoritmo habiendo establecido un máximo de 200 clústeres.

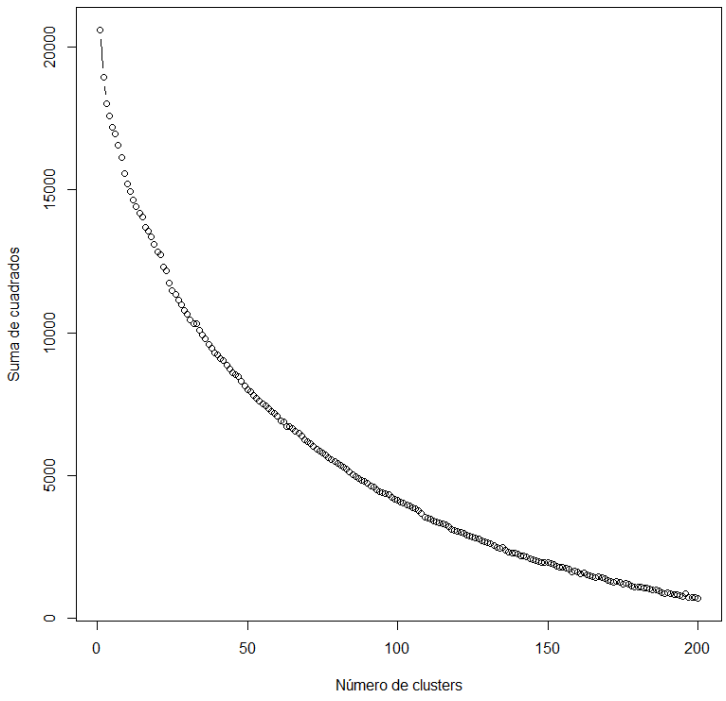


Figura 4-17: cálculo del número óptimo de clústeres para el algoritmo *k-means*

Observando el gráfico anterior, se puede apreciar la caída más pronunciada en los primeros 30 clústeres, por lo que se volvió a ejecutar dicho algoritmo haciendo enfoque en ese grupo. El gráfico generado se puede ver en la figura siguiente:

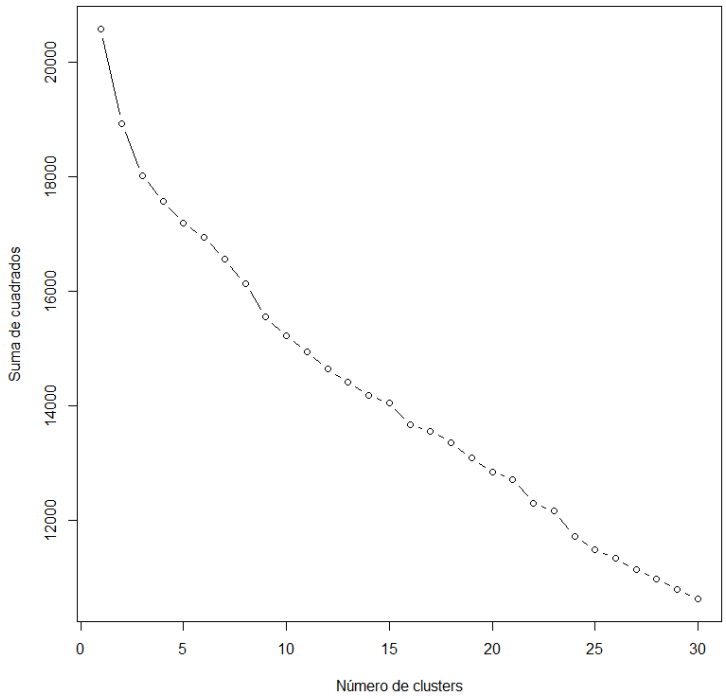


Figura 4-18: cálculo del número de clústeres para el algoritmo *k-means*

El gráfico que se generó sugería tres posibilidades en base a los decrementos pronunciados. La primera que se puede observar se encuentra entre los valores 2 y 3, por lo que el valor 3 sería el número óptimo de clústeres. La segunda posibilidad se encuentra entre los valores 4 y 5, por lo que el valor 5 sería el número de clústeres óptimo, sin embargo, si se toma como caída pronunciada la curva existente entre el valor 1 y 9, sería 9 el número de clústeres óptimos. Por este motivo se ejecutó el algoritmo *k-means* con estas tres valores de clústeres y se compararon los gráficos de distribución generados por el algoritmo. En primer lugar se analizó el gráfico devuelto con 9 clústeres, el cual se puede ver en la figura 4-19, donde se observan 4 clústeres claramente separados, los numerados como 2, 4, 5 y 8. Por otro lado se observan los

grupos 7 y 9, y muy cercanos a ellos, el resto de grupos, los numerados como 1, 3 y 6. En la figura 4-20 se muestra el gráfico generado por el algoritmo para 5 clústeres.

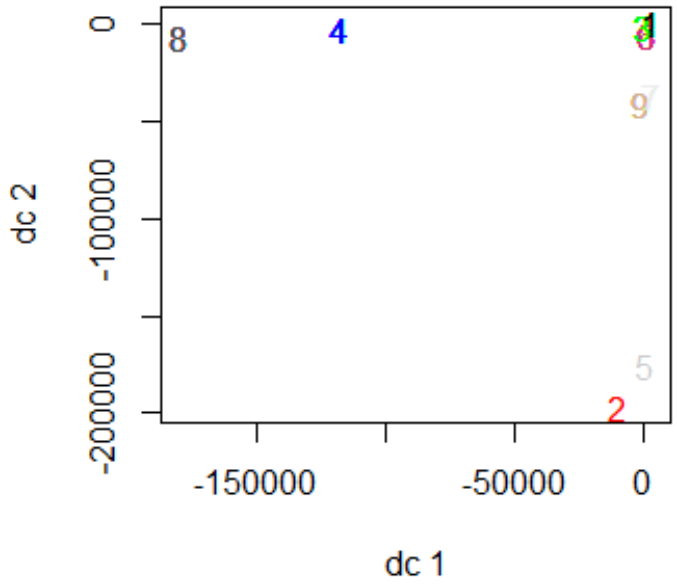


Figura 4-19: gráfico generado por el algoritmo *k-means* con una distribución de 9 clústeres

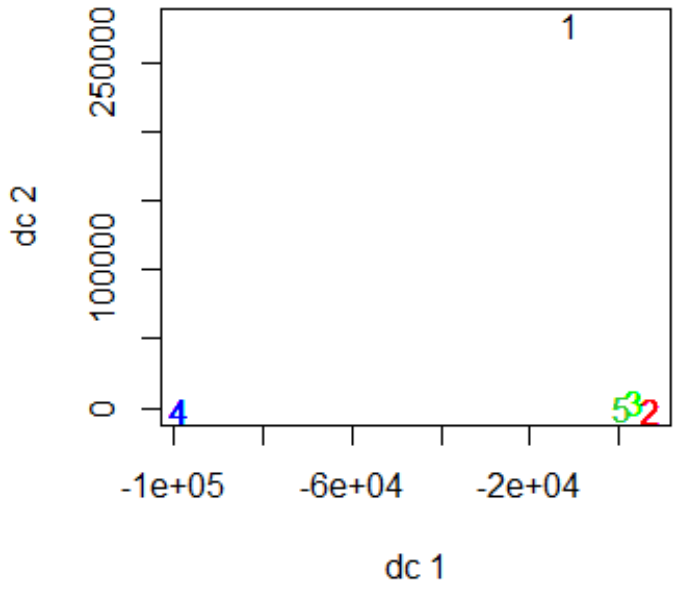


Figura 4-20: gráfico generado por el algoritmo *k-means* con una distribución de 5 clústeres

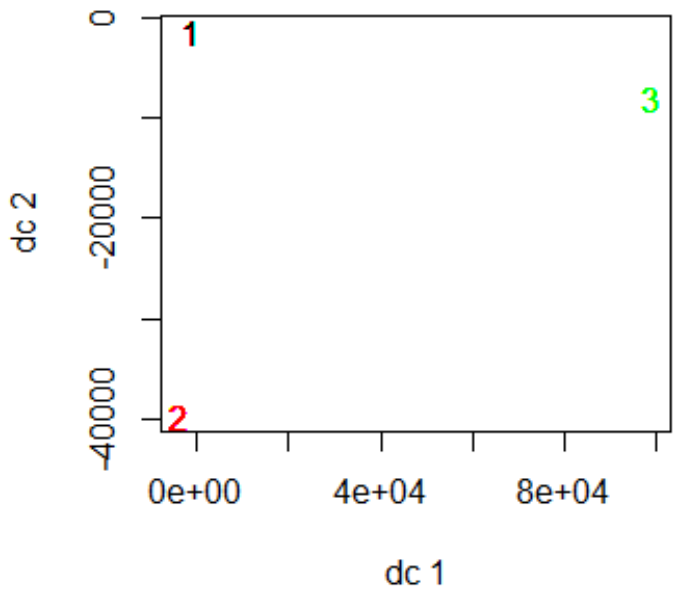


Figura 4-21: gráfico generado por el algoritmo *k-means* con una distribución de 3 clústeres

En este caso, se aprecian dos clústeres bien definidos frente al resto, los numerados como 1 y 4, y los tres restantes más agrupados en la parte inferior derecha. Finalmente, en el gráfico generado para 3 clústeres que se muestra en la figura 4-21, se encontraron tres grupos bien definidos.

En base a los resultados obtenidos, se analizaron manualmente algunos de los grupos devueltos en los casos de 3 y 9 clústeres. En el primer caso, el clúster más pequeño fue el único que tenía informes que informaran sobre una prueba o resultado similar, sin embargo el resto no. En el segundo caso, donde los clústeres eran más pequeños salvo uno, los informes de la mayoría de los clústeres correspondían a las mismas pruebas o resultados similares. En la tabla 4-11 se muestran los elementos asignados por el algoritmo a cada grupo. De ellos se analizaron en detalle el clúster número 1 del primer caso y el clúster número 3 del segundo caso ya que tenían algunos informes en común.

3 CLUSTERES		9 CLÚSTERES	
NÚMERO	ELEMENTOS	NÚMERO	ELEMENTOS
1	11	1	4
2	212	2	1
3	27	3	7
		4	168
		5	13
		6	30
		7	3
		8	23
		9	1
TOTAL	250	TOTAL	250

Tabla 4-11: distribución de elementos en cada grupo con el algoritmo *k-means*

CLÚSTER 1 DEL RESULTADO DE 3 CLÚSTERES		
id	report	test
64474	RESULTADOS DE LA DENSITOMETRIA:ZONA: FEMUR IZQUIERDO FEMUR IZQUIERDO.REGION: CUELLO TOTAL.OMS: OSTEOPENIA	DT
82684	RESULTADOS DE LA DENSITOMETRIA:ZONA: FEMUR IZQUIERDO FEMUR IZQUIERDO.REGION: CUELLO TOTAL.OMS: OSTEOPENIA	DT
82716	RESULTADOS DE LA DENSITOMETRIA:ZONA: COLUMNA AP.REGION: L1-L3.OMS: OSTEOPENIA.DMO: 0,887 G/CM2.T-	DT
161840	RESULTADOS DE LA DENSITOMETRIA:ZONA: ANTEBRAZO IZQ. ANTEBRAZO IZQ..REGION: RADIO UD RADIO 33%.OMS: N/A	DT
203442	RESULTADOS DE LA DENSITOMETRIA:ZONA: COLUMNA AP.REGION: L1-L4.OMS: OSTEOPENIA.DMO: 0,900 G/CM2.T-	DT
246646	RESULTADOS DE LA DENSITOMETRIA:ZONA: FEMUR IZQUIERDO FEMUR IZQUIERDO.REGION: CUELLO TOTAL.OMS: OSTEOPENIA	DT
247428	RESULTADOS DE LA DENSITOMETRIA:ZONA: COLUMNA AP.REGION: L1-L4.OMS: OSTEOPENIA.DMO: 0,962 G/CM2.T-	DT
249546	RESULTADOS DE LA DENSITOMETRIA:ZONA: COLUMNA AP.REGION: L1-L4.OMS: OSTEOPOROSIS.DMO: 0,678 G/CM2.T-	DT
257331	RESULTADOS DE LA DENSITOMETRIA:ZONA: COLUMNA AP.REGION: L1-L4.OMS: NORMAL.DMO: 1,131 G/CM2.T-SCORE: -	DT
307594	RESULTADOS DE LA DENSITOMETRIA:ZONA: COLUMNA AP.REGION: L1-L4.OMS: OSTEOPENIA.DMO: 0,970 G/CM2.T-	DT

Tabla 4-12: informes contenidos en el clúster 1 del resultado de 3 clústeres con *K-means*

CLÚSTER 3 DEL RESULTADO DE 9 CLÚSTERES		
id	**report**	**test**
82716	RESULTADOS DE LA DENSITOMETRIA:ZONA: COLUMNA AP.REGION: L1-L3.OMS: OSTEOPENIA.DMO: 0,887 G/CM2.T-	DT
161840	RESULTADOS DE LA DENSITOMETRIA:ZONA: ANTEBRAZO IZQ. ANTEBRAZO IZQ..REGION: RADIO UD RADIO 33%.OMS: N/A	DT
203442	RESULTADOS DE LA DENSITOMETRIA:ZONA: COLUMNA AP.REGION: L1-L4.OMS: OSTEOPENIA.DMO: 0,900 G/CM2.T-	DT
247428	RESULTADOS DE LA DENSITOMETRIA:ZONA: COLUMNA AP.REGION: L1-L4.OMS: OSTEOPENIA.DMO: 0,962 G/CM2.T-	DT
249546	RESULTADOS DE LA DENSITOMETRIA:ZONA: COLUMNA AP.REGION: L1-L4.OMS: OSTEOPOROSIS.DMO: 0,678 G/CM2.T-	DT
257331	RESULTADOS DE LA DENSITOMETRIA:ZONA: COLUMNA AP.REGION: L1-L4.OMS: NORMAL.DMO: 1,131 G/CM2.T-SCORE: -	DT
307594	RESULTADOS DE LA DENSITOMETRIA:ZONA: COLUMNA AP.REGION: L1-L4.OMS: OSTEOPENIA.DMO: 0,970 G/CM2.T-	DT

Tabla 4-13: informes contenidos en el clúster 1 del resultado de 3 clústeres con *K-means*

En las tablas anteriores se encuentran parte de los informes que se analizaron en detalle, que como se puede observar, tanto en el primer caso como en el segundo, todos los informes son muy parecidos y hacen referencia a la misma prueba, aunque en el segundo caso tienen una similitud mayor que en el primer caso.

4.5.2.3. Fuzzy C-Means

Para finalizar el estudio de *clustering* documental aplicado a los informes en general se aplicó el algoritmo *Fuzzy C-Means* con el objetivo de obtener un gráfico que mostrara la distribución de términos en una clasificación de 9 clústeres. Dicho gráfico se puede observar en la siguiente imagen, donde se pueden ver 9 elipses, que hacen referencia a cada uno de los clústeres. Como se aprecia en la figura, todos los clústeres comparten un gran número de palabras, de ahí que la separación en clústeres venga determinada por aquellas palabras que aparecen solo en algunos informes, como es el caso de las palabras “tscore” y “columna”.

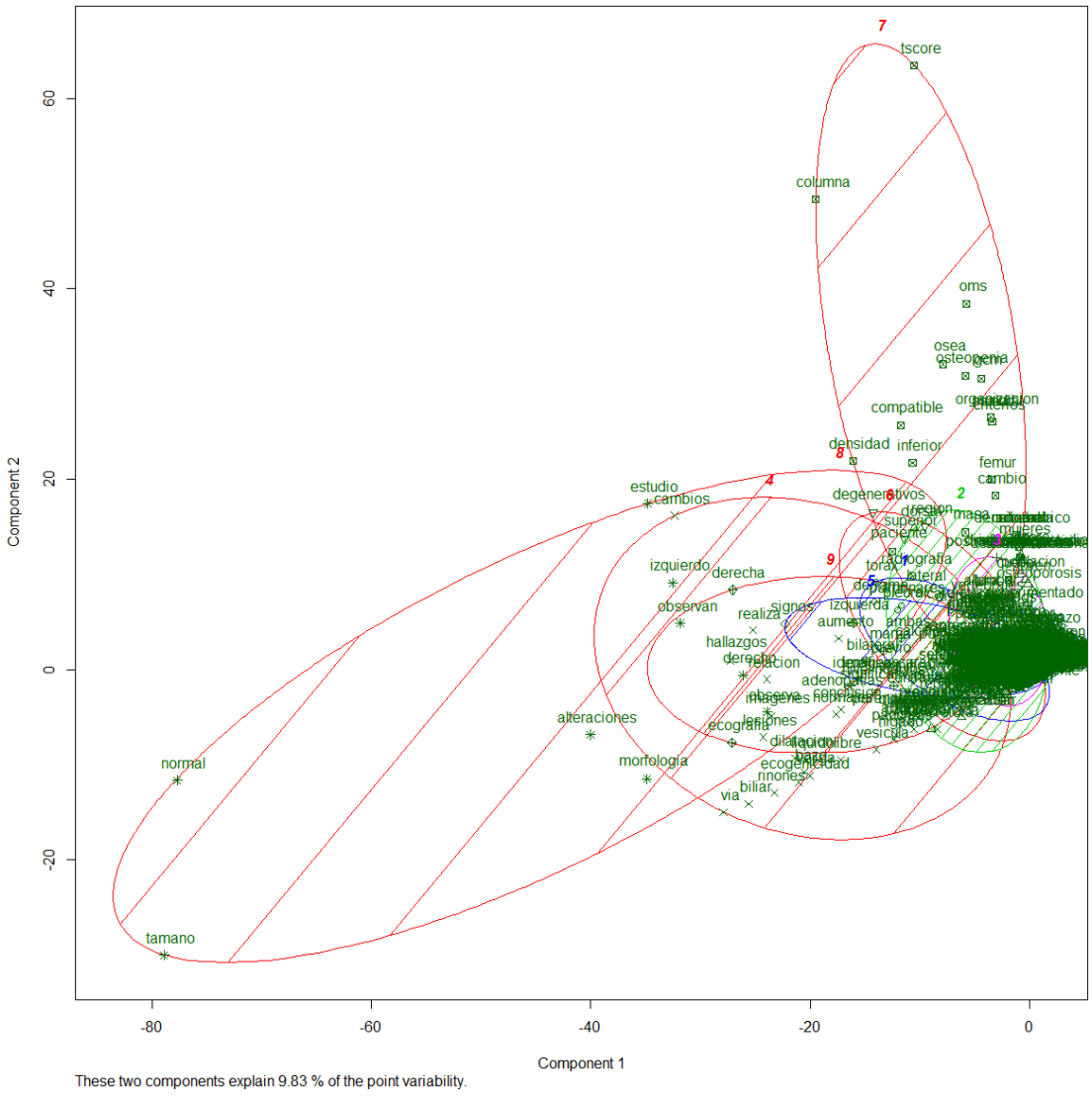


Figura 4-22: distribución de términos con el algoritmo Fuzzy C-Means

4.5.3. Resultados derivados importantes

Independientemente de lo comentado en el apartado anterior, al analizar diversos informes para comprobar su pertenencia a diferentes clústeres, se han obtenido resultados que no se esperaban, como por ejemplo, que se han encontrado informes que eran prácticamente iguales, o incluso iguales al 100%. Si tenemos en cuenta que un paciente sufre una enfermedad y se somete a dos pruebas iguales a lo largo del año para comprobar la evolución, puede tener sentido que el radiólogo copie y pegue el mismo texto, en el caso de no encontrar nada nuevo, o que copie, pegue y añada texto nuevo, ya que de esta manera, en principio, puede hacer más sencilla la lectura de varios informes que traten el mismo problema, y si esos informes son trasladados a otro especialista, con el último informe tendría datos suficientes para entender los resultados del total de pruebas, sin embargo, hay que tener en cuenta que para los cálculos de presupuestos y para realizar una estimación de costes, la UCR dispone de una tabla de tarifas, en cual se estima el tiempo necesario para rellenar cada tipo de informe así como su coste horas/hombre. Por ejemplo realizar un informe sobre una radiografía de un esguince de tobillo puede llevar entre 5 y 10 minutos, mientras que un TAC coronario puede llevar entre 60 y 80 minutos. Por tanto, si un radiólogo copia y pega un texto de un informe anterior y sólo añade unas líneas, el tiempo empleado es muy inferior al estimado. Explicado este problema, queda ahora analizar por qué existen informes prácticamente iguales entre diferentes pacientes, que aunque no son más que meras especulaciones, invita a pensar que en algún caso se están copiando informes y se están modificando los valores numéricos para reducir el tiempo de redacción y no se están redactando los informes desde cero.

Por otro lado, esto también tiene otra lectura, algo ya comentado en el apartado 4.4.1, Lenguaje Utilizado En Los Informes, que es la necesidad de crear estándares y herramientas informáticas que permitan estructurar los informes para facilitar su redacción y al mismo tiempo facilitar su posterior lectura y el análisis de los datos.

4.6. REPRESENTACIÓN DE DATOS

En el apartado de representación de datos se han desarrollado tres gráficos. Para su elaboración se ha seleccionado la librería D3.js, ya comentada en el apartado 3.5.2.4, por ser de las más estandarizadas en la actualidad y por tener un mayor número de seguidores, lo que la convierte en una de las librerías con más actualizaciones o con más ideas aportadas por desarrolladores. Los gráficos que se han desarrollado son los siguientes:

- **Gráfico de distribución de pruebas**: muestra las pruebas realizadas en un hospital y cálculos sobre porcentajes. Se basa en el gráfico BiPartite [53].
- **Gráficos de burbujas**: diseñados para mostrar agrupaciones como el árbol jerárquico, pero en este caso cada grupo está incluido en una burbuja, la cual a su vez puede tener más burbujas. Está basado en los gráficos Circle Packing [54] y Zoomable Circle Packing [55].
- **Árbol jerárquico de clústeres**: diseñado para mostrar agrupaciones de elementos desde la raíz, que sería el grupo inicial, hasta las hojas, donde se encontrarían los elementos. Está basado en el gráfico dndTree publicado en GitHub en octubre de 2013 [56].

4.6.1. GRÁFICO DE DISTRIBUCIÓN DE PRUEBAS

4.6.1.1. Diseño

Los elementos que componen este gráfico son los siguientes:

- **Un documento HTML**: incluye la estructura base de la página, la estructura del gráfico, (dimensiones, márgenes y títulos) y las llamadas a la librería D3.js y al archivo JavaScript que desarrolla el gráfico. También incluye los datos del gráfico siguiendo la siguiente estructura para cada una de las filas de datos:

$$[Hospital, Prueba, Total]$$

- **Un archivo JavaScript**: incluye el código que representa el gráfico y toma los datos del documento HTML los cuales tienen la siguiente estructura. Está basado en el gráfico BiPartite y ha sido modificado para poder representar los datos de los hospitales que se han utilizado como ejemplo.

4.6.1.2. Almacén de datos

Este gráfico no utiliza un archivo de datos estructurados específico, como podría ser un documento XML o JSON, simplemente hay que seguir la estructura $[Hospital, Prueba, Total]$. Un ejemplo de cómo almacenar los datos podría ser el siguiente:

```
var hospitales=[
    ['Centro de Especialidades de Alcobendas','RX',9325],
    ['Hospital de Arganda','MG',2263],
    ['Hospital de Arganda','RM',6174],
    ['Hospital de Vallecas','DT',4671],
    ['Hospital de Vallecas','MG',2260],
    ['Hospital del Henares','JV',5881],
    ['Hospital del Henares','MG',2333],
    ['Hospital del Norte','DT',7409],
    ['Hospital del Norte','MG',5550],
    ['Hospital del Norte','RM',10911],
    ['Hospital del Sur','MG',2932],
    ['Hospital del Sur','RM',6356],
    ['Hospital del Tajo','MG',1851],
    ['Hospital del Tajo','RM',4611],
];

```

Figura 4-23: ejemplo de datos de entrada para el gráfico BiPartite

Como se puede observar en el código anterior, no existe un cálculo de los totales por prueba ni de los porcentajes de cada prueba en relación al total, ya que esos cálculos se realizan en el código JavaScript del gráfico.

4.6.1.3. Volcado de datos

Los datos se han obtenido de un *data frame* creado en R, el cual se ha convertido en un tipo *data table* para poder utilizar las librerías de cálculos de totales. Una vez calculados los totales de cada prueba en cada hospital, se agregan a una nueva columna y se eliminan las columnas duplicadas. Finalmente se sustituyen las siglas de los hospitales por su nombre completo.

Dada la complejidad del proceso, se ha considerado relevante incluir el código en R en este apartado.

```
# ********************************************************************
# Calculating totals for graphic statistics
#
# ********************************************************************
library(data.table)

# data table creation
dt <- data.table(ucr.df)

# Calculating totals
dt[, COUNT := .N, by='center,test']

# Create data table with 3 fields
dt[,c("center","test","COUNT")]
dt.graph <- data.frame(dt$center, dt$test, dt$COUNT)

# Delete duplicated rows and name the columns
dt.graph <- dt.graph [!duplicated(dt.graph), ]
colnames(dt.graph) <- c("centro","test", "totales")

# Change de string of the hospitals
dt.graph$centro <- as.character(dt.graph$centro)
dt.graph$centro[dt.graph$centro == "AL"] <- "Centro de Especialidades de Alcobendas"
dt.graph$centro[dt.graph$centro == "AR"] <- "Hospital de Arganda"
dt.graph$centro[dt.graph$centro == "HE"] <- "Hospital del Henares"
dt.graph$centro[dt.graph$centro == "NO"] <- "Hospital del Norte"
dt.graph$centro[dt.graph$centro == "PA"] <- "Hospital del Sur"
dt.graph$centro[dt.graph$centro == "TJ"] <- "Hospital del Tajo"
dt.graph$centro[dt.graph$centro == "VA"] <- "Hospital de Vallecas"

# Order the graph by centro, test y totales
dt.graph <- dt.graph[order(dt.graph$centro, dt.graph$test, dt.graph$totales), ]
```

Figura 4-24: código en R para el volcado de datos

4.6.1.4. Creación dinámica del documento HTML

Para poder crear un gráfico que se actualizara en función de los datos y dado que estos se encuentran en el propio documento HTML, se ha optado por crear de forma dinámica el documento HTML desde R, tal y como ser haría desde un lenguaje como PHP, para ello se crea la parte de la cabecera del documento, posteriormente se agregan los datos del *data table* comentado anteriormente y se añade la parte final del

documento HTML. Un ejemplo del resultado de la creación dinámica del documento HTML se puede observar en la siguiente imagen.

```
<!DOCTYPE html>
  <meta charset="utf-8">
  <style>
    body{width:1200px;margin:100px auto;}
    svg text{font-size:12px;}
    rect{shape-rendering:crispEdges;}
  </style>
  <body>
  <script src="http://d3js.org/d3.v3.min.js"></script>
  <script src="biPartite.js"></script>
  <script>
  var hospitales=[['Centro de Especialidades de Alcobendas','RX',9325],
     ['Hospital de Arganda','MG',2263],
     ['Hospital de Vallecas','DT',4671],
     ['Hospital de Vallecas','VA',539],
     ['Hospital del Henares','JV',5881],
     ['Hospital del Henares','US',22979],
     ['Hospital del Norte','DT',7409],
     ['Hospital del Norte','VA',946],
     ['Hospital del Sur','MG',2932],
     ['Hospital del Sur','US',15097],
     ['Hospital del Tajo','MG',1851],
     ['Hospital del Tajo','RM',4611],
     ['Hospital del Tajo','US',11724],
];
  var width = 2000, height = 610, margin ={b:0, t:80, l:500, r:50};
  var svg = d3.select("body")
    .append("svg").attr("width",width).attr("height",(height+margin.b+margin.t))
    .append("g").attr("transform","translate("+ margin.l+","+margin.t+")");
  var data = [
    {data:bP.partData(hospitales,2), id:"hospitales", header:["Hospital","Prueba",
    "Pruebas Realizadas por Hospital"]},
  ];
  bP.draw(data, svg);
  </script>
  </body>
```

Figura 4-25: ejemplo de creación dinámica del documento HTML

4.6.1.5. Ejemplos

A continuación se muestran tres ejemplos de visualización del gráfico dinámico, el cual va modificando los cálculos de forma dinámica en las columnas de la izquierda o de la derecha.

- **Ejemplo 1**: muestra de los totales de cada prueba en cada hospital. Es la apariencia del gráfico cuando se carga la página web.

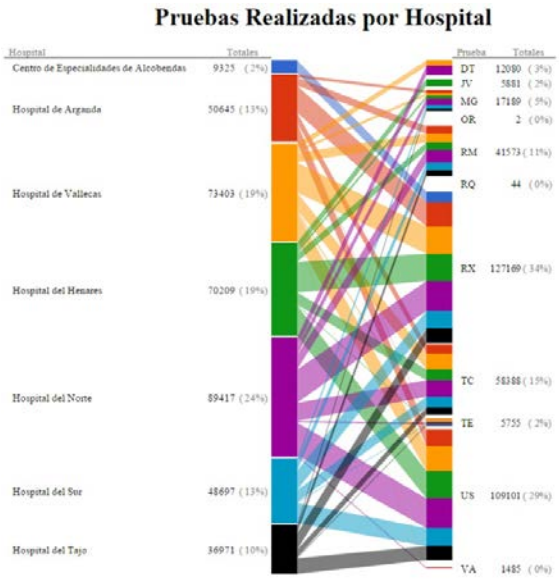


Figura 4-26: ejemplo del gráfico de pruebas por hospital

- **Ejemplo 2**: en este caso se ha colocado el cursor encima de la etiqueta "Hospital del Henares", por lo que el gráfico muestra en la derecha las pruebas realizadas en este hospital.

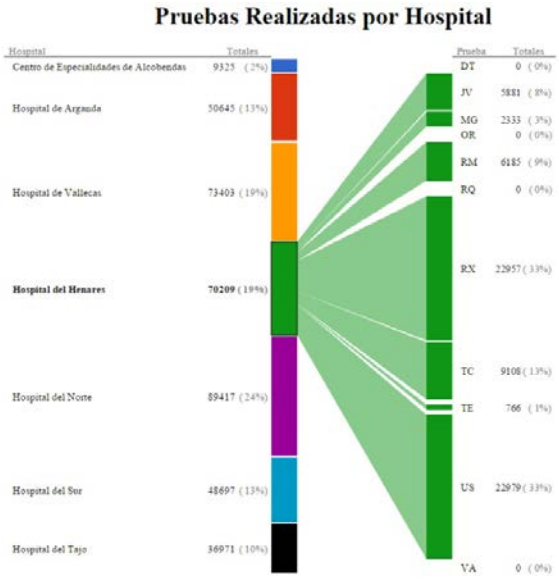


Figura 4-27 ejemplo del gráfico de pruebas por hospital

- **Ejemplo 3**: en esta ocasión se ha colocado el cursor del ratón encima de una de las pruebas. De esta manera el gráfico actualiza la parte izquierda indicando cuántas radiografías se han realizado en cada centro.

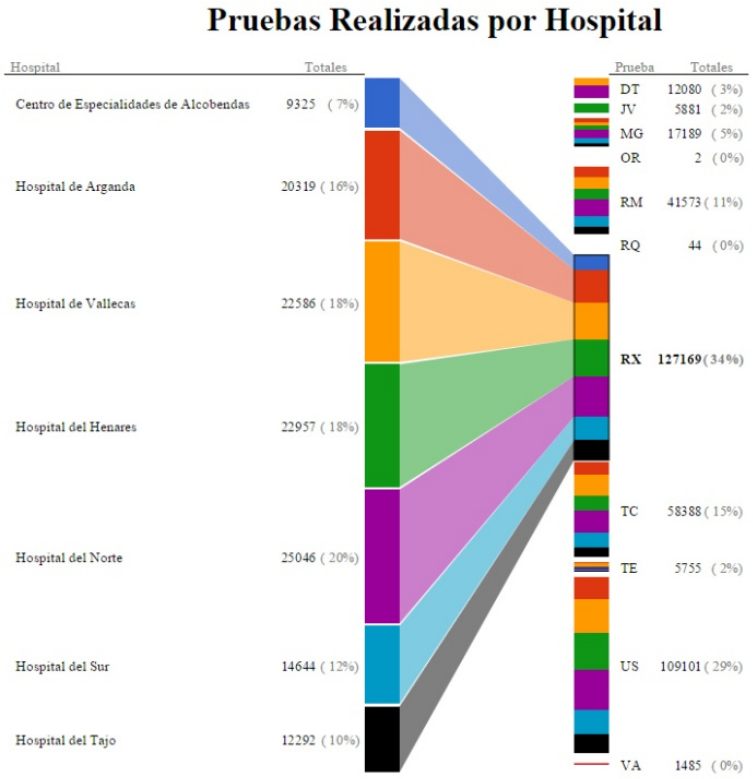


Figura 4-28: ejemplo del gráfico de pruebas por hospital

Este gráfico está accesible en la siguiente dirección web:

http://juanpueyo.tk/uned/graficos/estadisticas/

Por seguridad, el acceso ha sido protegido. Para acceder es necesario utilizar como usuario *uned* y como contraseña *tfmmaster*.

4.6.2. GRÁFICOS DE BURBUJAS

4.6.2.1. Diseño

Estos gráficos han sido elaborados partiendo de los diseños de dos gráficos existentes, Circle Packing y Zoomable Circle Packing, aprovechando su diseño gráfico y adaptando su representación para textos grandes como son los informes radiológicos, así como la alimentación de datos que se ha adaptado para archivos JSON.

Cada gráfico está formado por:

- **Un documento HTML**: incluye la estructura HTML y los estilos CSS de la página, así como el diseño completo del gráfico y sus estilos. Para el diseño del gráfico, a partir de los resultados obtenidos con los algoritmos de *clustering*, se ha generado un archivo JSON, y utilizando la librería D3js, se ha creado una burbuja por cada clúster devuelto por los resultados de R, y en esas burbujas se han ido incluyendo las burbujas en función del resto de grupos creados hasta llegar a los propios informes, los cuales se muestran colocando el cursor encima de la burbuja.
- **Un archivo JSON**: incluye los datos devueltos por los resultados que devuelven los algoritmos de *clustering* de R.

4.6.2.2. Almacén de datos

Como ya se ha comentado, los datos se almacenan en un archivo JSON, el cual tiene la siguiente estructura:

CLUSTER	ID Informe	Texto Informe
1	aaaaaaaaa	…
	bbbbbbbb	…
	cccccccccc	…
2	ddddddddd	…
	eeeeeeeee	…

El cual se vuelca siguiendo la siguiente sintaxis:

```
{"name": "ROOT",
 "children": [
        {"name": "cluster01",
         "children": [
             {"name": "ID informe",
              "children": [
                  {"name": "Texto informe 01"},
                  {"name": "Texto informe 02"}
             ]}
         ]}
]}
```

4.6.2.3. Volcado de datos

Para el volcado de datos se ha creado una función recursiva que recorre cada uno de los grupos creados por los algoritmos de *clustering* ejecutados en R.

```
library(RJSONIO)

makeList<-function(x){
  if(ncol(x)>2){
    listSplit<-split(x[-1],x[1],drop=T)
    listSplit
    lapply(names(listSplit),function(y){list(name=y,children=makeList(listSplit[[y]]))})
  }else{
    lapply(seq(nrow(x[1])),function(y){list(name=x[,1][y],size=x[,2][y])})
  }
}
```

4.6.2.4. Ejemplos

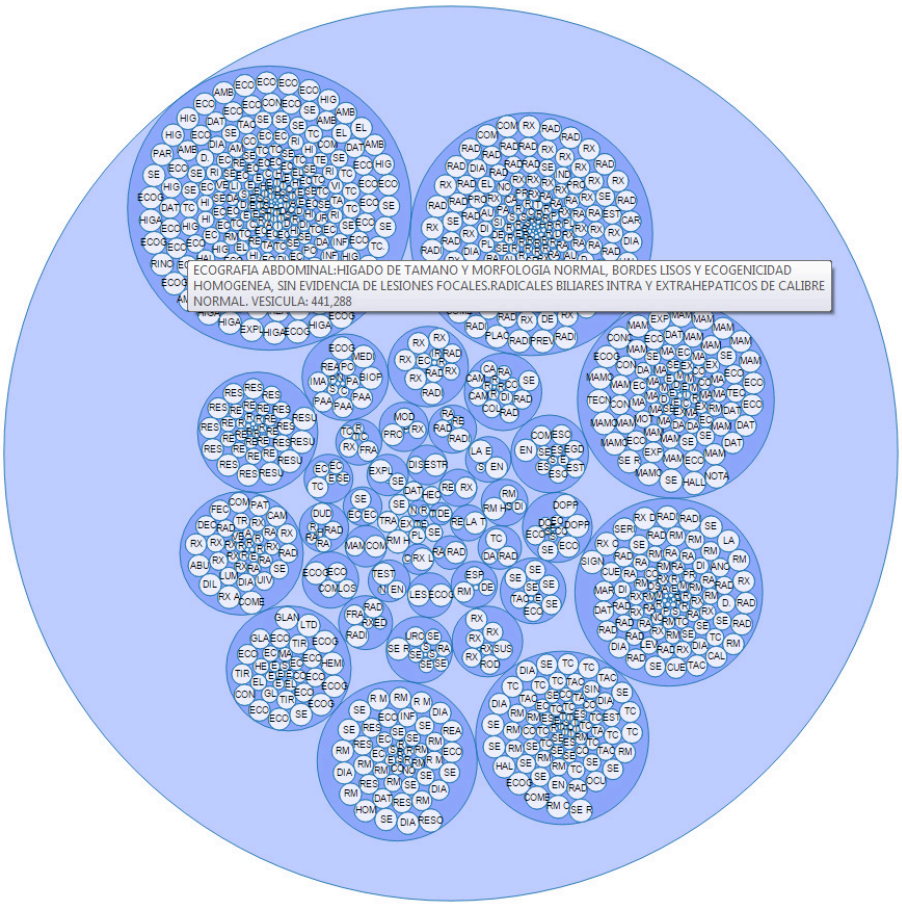


Figura 4-29: gráfico que muestra las diferentes agrupaciones creadas por los algoritmos de *clustering*

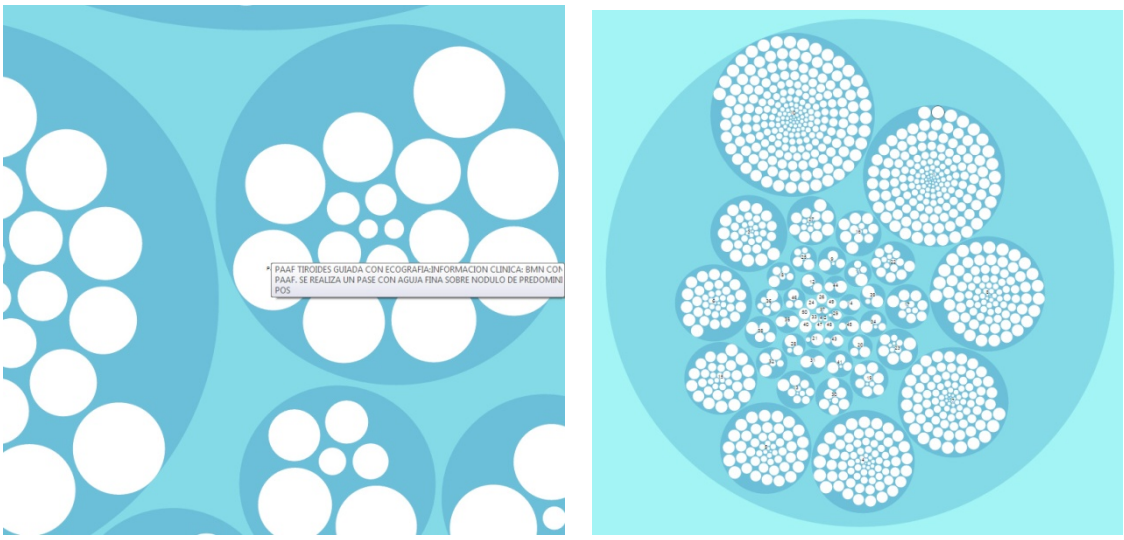


Figura 4-30: gráfico interactivo que permite hacer zoom en las diferentes agrupaciones

Estos gráficos están accesibles en las siguientes direcciones web:

- http://juanpueyo.tk/uned/graficos/cluster/
- http://juanpueyo.tk/uned/graficos/clusterzoom

Por seguridad, el acceso ha sido protegido. Para acceder es necesario utilizar como usuario *uned* y como contraseña *tfmmaster*.

4.6.3. Árbol jerárquico de clústeres

4.6.3.1. Diseño

Se trata de un árbol generado a partir del gráfico dndTree, el cual se ha adaptado para poder representar textos de gran longitud, como son los informes médicos, y se ha modificado parte del código para reducir el espacio ocupado en la pantalla a la hora de su representación.

Cada gráfico está formado por:

- **Un documento HTML**: incluye la estructura HTML y los estilos CSS del árbol y de la página.
- **Librería de JavaScript dndTree.js**: librería que incluye el código para generar el árbol, así como para permitir la interactividad con el mismo. Parte de esta librería se ha tenido que modificar para permitir la representación del texto de los informes médicos, ya que en el diseño original se producía un solapamiento de la información entre diferentes informes. También se han modificado las funciones que realizan el trazado de las líneas entre cada uno de los clústeres y sus elementos, las cuales eran demasiado largas (por tratar de adaptarse a los textos largos) y generaban un gráfico de poca utilidad.

  Al igual que en los gráficos de burbujas, este gráfico también toma los datos de archivos JSON.
- **Un archivo JSON**: incluye los datos devueltos por los resultados que devuelven los algoritmos de *clustering* de R.

4.6.3.2. Archivo de datos JSON

La librería dndTree.js utiliza el formato de archivo JSON para obtener los datos, sin embargo su estructura difiere de la utilizada para los gráficos de burbujas, por lo que se optó por la opción de modificar la función de la librería para aprovechar el mismo archivo JSON utilizado en dichos gráficos y no tener que modificar el código de R.

4.6.3.3. Ejemplos

En la siguiente imagen se muestra la misma información que se mostraba con los gráficos de burbujas. En este caso, además de poder visualizar toda la información en una sola pantalla, queda más patente la agrupación en clústeres y resulta más sencillo leer los informes médicos de cada uno de los grupos y poder comparar entre ellos.



Figura 4-31: imagen completa del árbol mostrando un total de 50 informes

Al tratarse de un gráfico interactivo, existe la posibilidad de hacer zoom con la rueda del ratón en cualquier parte del gráfico para leer el texto del informe. También se puede hacer clic en las agrupaciones para abrir o cerrar los grupos.

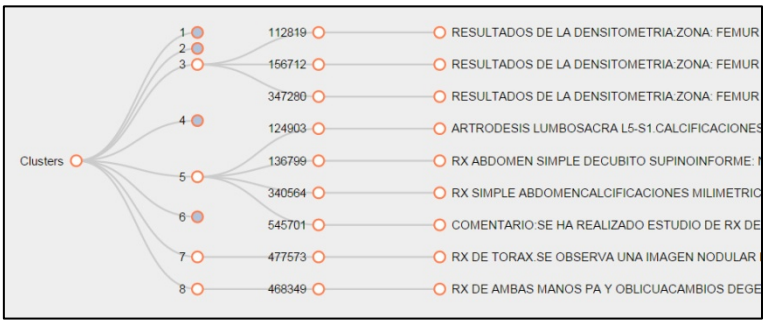


Figura 4-32: imagen del árbol en la que se observan algunos grupos cerrados

Otra de las opciones que permite el gráfico es la de arrastrar elementos desde su grupo original a otro grupo cualquiera pero dado que se trataría de una clasificación dependiente del usuario, no se ha creado una exportación de los datos del diseño modificado por el usuario.

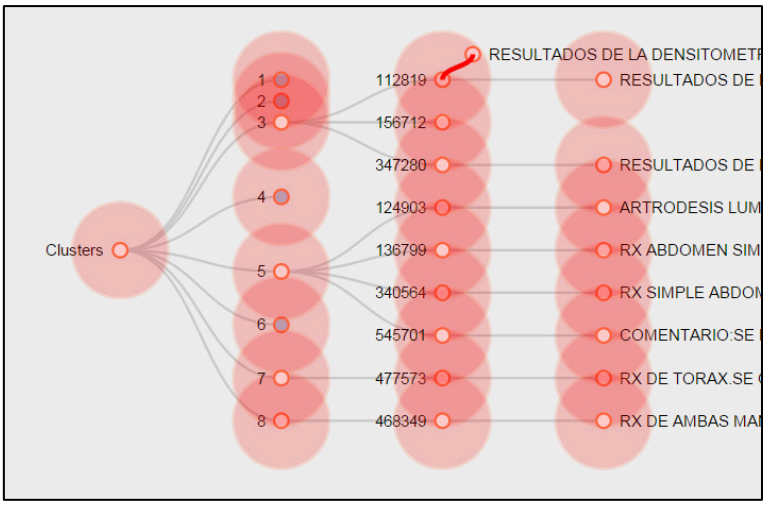


Figura 4-33: imagen del árbol en la que se observa la colocación manual de los elementos

Este gráfico está accesible en la siguiente dirección web:

http://juanpueyo.tk/uned/graficos/arbol/

Por seguridad, el acceso ha sido protegido. Para acceder es necesario utilizar como usuario *uned* y como contraseña *tfmmaster*.

5. CONCLUSIONES Y TRABAJO FUTURO

5.1. Conclusiones

El estudio realizado en este proyecto se encuentra enmarcado en el área de la minería de datos y más concretamente en el área de la minería de textos desestructurados, tanto a nivel de investigación como a nivel de presentación gráfica de resultados. Al comienzo de este proyecto se establecieron cuatro objetivos, tres relacionados con la minería de textos y uno relacionado con el diseño de gráficos. A continuación se detallan las conclusiones alcanzadas en relación a cada uno de los objetivos que se buscaban en este proyecto.

El primer objetivo que se estableció estaba relacionado con la búsqueda de resultados realizando un análisis semántico en los informes de radiodiagnóstico. Para ello se utilizó un caso real en el que se buscaban aquellos informes que dieran a conocer una reacción alérgica al contraste durante la realización de la prueba dentro del volumen de informes reales de los que se disponía, en total 554.173 informes. Después de realizar un tratamiento previo de dichos informes y de preparar los textos para su análisis semántico, se logró no solo encontrar todos los informes que se buscaban, sino que también se logró separar de forma automática los informes que describían una reacción alérgica durante la prueba, de aquellos informes en los que era el radiólogo el que comentaba que el paciente hacía referencia a que en algún momento había sufrido una reacción alérgica a algún tipo de contraste. Los resultados obtenidos tenían una exactitud de un 99,99% y de una precisión de un 95,65%, lo que demostraba que estos resultados eran mucho mejores de lo esperado en un principio. Una vez obtenidos estos resultados, se realizaron diversas clasificaciones que dieron lugar a la obtención de información de caracter importante, especialmente cuando se descubrió que más de la mitad de las alergias producidas tuvieron lugar en el mismo hospital.

El segundo objetivo que se estableció hacía referencia a la posibilidad de obtener los mismos resultados con el análisis semántico que con técnicas de *clustering* documental, más concretamente en la separación mediante estas técnicas de los informes que hacían referencia a una alergia al contraste a lo largo de la prueba de los

que hacían referencia a una alergia anterior. Se realizaron pruebas con diversos algoritmos de *clustering*, se modificaron diversos parámetros, se crearon algoritmos para buscar los mejores valores para estos parámetros y se utilizaron técnicas para desagrupar y volver agrupar, sin embargo no se lograron resultados que pudieran considerarse válidos. Concretamente, el algoritmo que mejor exactitud y precisión devolvió fue el algoritmo *k-means* con un 77,78% y un 57,89% respectivamente, porcentajes demasiado bajos para ser tenidos en cuenta.

Como tercer objetivo del proyecto se buscaba la posibilidad de crear agrupaciones de informes utilizando técnicas de *clustering* que pudieran servir de punto de partida para otros estudios o para buscar enfermedades o resultados similares. En este punto sí se obtuvieron buenos resultados con los algoritmos que se utilizaron, ya que se lograron agrupaciones de informes que tenían una relación muy estrecha o que eran muy similares. Esta búsqueda también dio lugar a encontrar informes iguales de pacientes diferentes, lo que da pie a pensar en la posibilidad de que algunos informes estén siendo copiados para ganar tiempo. Como conclusión a este punto, hay que dejar claros los buenos resultados obtenidos, aunque dependiendo de la exactitud que se busque puede que sea necesario un elevado trabajo manual de validación. En este punto del proyecto también cabe destacar que hubo que recurrir a utilizar grupos pequeños de informes debido a los altos tiempos de proceso que se requerían, lo que deja patente que para tratar un volumen de informes como del que se disponía requeriría realizar un escalado a técnicas de *Big Data*.

Finalmente, el último objetivo que se propuso se trataba de un desarrollo de software en entorno web. Dicho desarrollo ha servido para demostrar que la presentación de resultados es un punto importante, ya que facilita en gran medida la labor de validación de resultados, sobre todo si los gráficos son dinámicos y permiten realizar modificaciones en los mismos.

En definitiva, en minería de textos, el análisis semántico es quizás una de las tareas más importante a realizar, que por sí sola en algunos casos puede dar resultados con una exactitud cercana al 100%, y que además, en los casos en los que se quieran

aplicar técnicas de *clustering* documental, el análisis semántico debe ser una técnica que se aplique en conjunto con estas.

5.2. Trabajo futuro

El trabajo desarrollado en este proyecto es una pequeña parte de lo que comprende la minería de textos, más aún si se piensa en que ciertas técnicas no se pueden aplicar si no se utilizan tecnologías de *Big Data*. Por este motivo existen multitud de ideas que se pueden llevar a cabo partiendo del trabajo aquí realizado. A continuación se enumeran algunas de ellas:

- Llevar a cabo las técnicas aplicadas en este proyecto en informes de radiodiagnóstico en otros idiomas.
- Escalar el trabajo realizado a sistemas Spark o Hadoop para aplicar las técnicas de *clustering* a todos los informes en conjunto.
- Mediante las técnicas de *clustering* crear grupos que se relacionen por la enfermedad detectada y posteriormente crear algoritmos que analicen los grupos obtenidos y analizar la posibilidad de que un paciente pertenezca a cada uno de estos grupos de enfermedades y estudiar si esta vía puede llevar a predecir enfermedades.
- En base a los resultados obtenidos, fusionar estos datos con datos demográficos para obtener datos estadísticos desconocidos hasta ahora.
- Debido a la complejidad a la que se elevaba el trabajo a realizar, no se tuvieron en cuenta los valores numéricos de los informes. Un trabajo interesante sería el de desarrollar algoritmos capaces de encontrar los valores numéricos asociados a las enfermedades para así poder catalogar las enfermedades en diferentes niveles de importancia o gravedad.
- Otra tarea que podría ser interesante y que ya se ha realizado en otro tipos de textos es lo que se conoce en inglés como *stemming*, que consiste en la

sustitución de todas las palabras de una misma familia por una única palabra, y así analizar los resultados de aplicar las técnicas de *clustering* a estos textos.

6. LISTA DE REFERENCIAS Y BIBLIOGRAFÍA

[1] J. L. Sampedro, «El reloj, el gato y Madagascar,» *Revista de Estudios Andaluces,* pp. 119-126, 1983.

[2] UCR, «ti3CEIEC | Blog de Inteligencia Económica del CEIEC,» 05 2004. [En línea]. Available: http://ti3.ceiec.es/wp-content/uploads/2014/05/MarisaGonzalez_UCR.pdf. [Último acceso: 2015].

[3] R Data Mining, «Free Datasets,» [En línea]. Available: http://www.rdatamining.com/resources/data.

[4] P. Vagata y K. Wilfong, «Facebook Code,» 10 4 2014. [En línea]. Available: https://code.facebook.com/posts/229861827208629/scaling-the-facebook-data-warehouse-to-300-pb/. [Último acceso: 2015].

[5] J. Han y M. Kamber, «Data Mining: Concepts and Techniques,» *Morgan Kaufmann Publishers Inc.,* 2000.

[6] j. Hernández Orallo, M. Ramírez Quintana y C. Ferri Ramírez, Introducción a la minería de datos, Valencia: Pearson Education, 2004.

[7] V. Ramanathan y T. Meyyappan, «Survey of Text Mining,» *International Conference on Technology and Business Management,* 2013.

[8] A.-H. Tan, «Text Mining: The state of the art and the challenges,» *proceedings, PAKDD Workshop on Knowledge discovery from Advanced Databases (KDAD'99),* pp. 71-76, 1999.

[9] J. Etxeberría Murgiondo, Análisis de Datos y Textos, Madrid: RA-MA, 1995.

[10] H. Andreas, N. Andreas y P. Gerhard, «A Brief Survey of Text Mining,» 2005.

[11] F. Fernández Rebollo y D. Borrajo Millán, «OpenCourseWare Universidad Carlos III de Madrid,» 27 02 2009. [En línea]. Available: http://ocw.uc3m.es/ingenieria-informatica/aprendizaje-automatico/material-de-clase-1/aa-ocw-agrupacion.pdf. [Último acceso: 2015].

[12] D. Crowe, «David Crowe: Tomatoes, HIV and AIDS,» 2004, [En línea]. Available: http://davidcrowe.ca/SciHealthEnv/TomatoAndHIV.html. [Último acceso: 2015].

[13] R. A. E. «Diccionario de la lengua española (22.a ed.).,» 2012. [En línea]. Available:

http://www.rae.es/rae.html.

[14] Wikipedia, «Infrutescencia - Wikipedia, la enciclopedia libre,» 9 4 2014. [En línea]. Available: https://es.wikipedia.org/wiki/Infrutescencia. [Último acceso: 2015].

[15] E. Mooi y M. Sarstedt, «A Concise Guide to Market Research,» de *The Process, Data, and Methods Using IBM SPSS Statistics*, Amsterdam, Springer Berlin Heidelberg, 2011, pp. 237-284.

[16] C. A. Núñez-Colín, J. E. Rodríguez-Pérez, R. Nieto-Ángel y A. F. Barrientos-Priego, «Construcción de dendrogramas de taxonomía numérica mediante el coeficiente de distancia χ2: una revisión,» *Chapingo,* pp. 229-237, 2004.

[17] D. Hand, F. Daly, A. D. Lunn, K. J. McConway y E. Ostrowski, «A Handbook of Small Data Sets,» Londres, Chapman & Hall, 1994.

[18] I. O. S. Improved Outcomes Software, «Distance Metrics Overview,» [En línea]. Available: http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Distance_Metrics_Overview.htm. [Último acceso: 2015].

[19] H. Gupta y R. Srivastava, «k-means Based Document Clustering with Automatic “k” Selection and Cluster Refinement,» *International Journal of Computer Science and Mobile Applications,* vol. 2, nº 5, pp. 7-13, 5 5 2014.

[20] L. Kaufman y P. J. Rousseeuw, «Clustering by means of medoids,» *Statistical Data Analysis Based on the L1-Norm and Related Methods,* 1987.

[21] L. Kaufman y P. J. Rousseeuw, «Finding groups in data: An introduction to cluster analysis,» *J. Wiley &, Sons,* 1990.

[22] The R Foundation, «R Proyect,» [En línea]. Available: https://www.r-project.org/. [Último acceso: 2015].

[23] RStudio Inc., «R Studio,» [En línea]. Available: https://www.rstudio.com/. [Último acceso: 2015].

[24] R Statistics, «Top 100 R packages for 2013 (Jan-May)!,» [En línea]. Available: http://www.r-statistics.com/2013/06/top-100-r-packages-for-2013-jan-may/. [Último acceso: 2015].

[25] W3C, World Wide Web Consortium, «On SGML and HTML,» 2012. [En línea]. Available: http://www.w3.org/TR/html4/intro/sgmltut.html.

[26] W3C, World Wide Web Consortium, «CSS current work & how to participate,» [En línea].

Available: http://www.w3.org/Style/CSS/current-work. [Último acceso: 08 2015].

[27] S. F. Winsor, «Inkscape Stuff,» 14 12 2008. [En línea]. Available: http://inkscapestuff.org/content/show.php/svg+tomato?content=95025. [Último acceso: 2015].

[28] ECMA-International, «ECMA-International - Standards@Internet Speed,» [En línea]. Available: http://www.ecma-international.org/. [Último acceso: 2015].

[29] ECMA International, «Standard ECMA-262,» 6 2015. [En línea]. Available: http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf.

[30] G. Fink y I. Flatow, «Pro Single Page Application Development: Using Backbone.js and ASP.NET,» 2014, p. 4.

[31] Google Inc., «Google Fusion Tables,» [En línea]. [Último acceso: 2015].

[32] P. Aldhous, «Rains come to Texas but drought could last for years,» *New Scientist,* 2011.

[33] CartoDB, «CartoDB is the easiest way to map and analyze your location data,» [En línea]. Available: https://cartodb.com. [Último acceso: 2015].

[34] NYCHenge, «NYCHenge,» [En línea]. [Último acceso: 2015].

[35] J. Branigan, «Visualizing data with CartoDB,» Azavea Atlas , 28 11 2012. [En línea]. Available: http://www.azavea.com/blogs/atlas/2012/11/visualizing-data-with-cartodb/. [Último acceso: 2015].

[36] Tableau Software, «tableau,» [En línea]. Available: http://www.tableau.com/. [Último acceso: 2015].

[37] IBM, «Many Eyes,» [En línea]. Available: http://www-01.ibm.com/software/analytics/many-eyes/.

[38] Google Inc., «Google Chart Tools,» [En línea]. Available: https://developers.google.com/chart/?hl=es. [Último acceso: 2015].

[39] CTO Insights, «Top Chart Libraries,» 26 7 2011. [En línea]. Available: https://ctoinsights.files.wordpress.com/2011/07/google-chart-tool.png. [Último acceso: 2015].

[40] FusionCharts, «FusionCharts,» [En línea]. Available: http://www.fusioncharts.com/. [Último acceso: 2015].

[41] C. Saternos, «R-Bloggers,» 12 7 2010. [En línea]. Available: http://www.r-bloggers.com/hierarchical-visualizations-in-r-and-the-javascript-infovis-toolkit/.

[42] JavaScript Infovis Toolkit, «JavaScript Infovis Toolkit,» [En línea]. Available: http://philogb.github.io/jit/index.html. [Último acceso: 2015].

[43] M. Bostock, «D3js,» Data-Driven Documents. [En línea]. [Último acceso: 2015].

[44] GitHub, «Atom,» [En línea]. Available: https://atom.io/.

[45] GitHub, «GitHub - Atom,» [En línea]. Available: https://github.com/atom.

[46] S. Nourashrafeddin, E. Milios y D. V. Arnold, «An Ensemble Approach for Text Document Clustering,» *Proceedings of the 2014 ACM symposium on Document engineering,* pp. 107-116 , 2014.

[47] A. Hotho, A. Nurnberger y G. PaaB, «A Brief Survey of Text Mining,» Mayo 2005.

[48] Google Inc, «Google Code StopWords,» 2015. [En línea]. Available: http://code.google.com/p/stop-words/source/browse/trunk/stop-words/stop-words/stop-words-spanish.txt?r=3. [Último acceso: 2015].

[49] RSNA, ACR, «RadiologyInfo.org,» [En línea]. Available: http://www.radiologyinfo.org/sp/info.cfm?pg=safety-contrast. [Último acceso: 2015].

[50] Robert I. Kabacoff, Ph.D., «Quick-R,» [En línea]. Available: http://www.statmethods.net/advstats/cluster.html.

[51] B. J. Frey y D. Dueck, «Clustering by Passing Messages Clustering by Passing Messages,» *Science,* 2007.

[52] U. Bodenhofer, A. Kothmeier y S. Hochreiter, «Institute of Bioinformatics, Johannes Kepler University Linz,» 2011. [En línea]. Available: http://www.bioinf.jku.at/software/apcluster/APCluster-Manual.pdf.

[53] M. Bostock, «BiPartite,» 18 5 2015. [En línea]. Available: http://bl.ocks.org/. [Último acceso: 2015].

[54] M. Bostock, «Circle Packing,» 12 11 2013. [En línea]. Available: http://bl.ocks.org/mbostock/4063530.

[55] M. Bostock, «Zoomable Circle Packing,» 22 11 2013. [En línea]. Available: http://bl.ocks.org/mbostock/7607535.

[56] R. S. Coutinho, «dndTree,» 6 10 2013. [En línea]. Available: https://github.com/RodrigoSC/dndTree.

[57] Y. Zhao, «R and Data Mining: Examples and Case Studies,» 2013.

[58] F. M. Kwale, «An Efficient Text Clustering Framework,» *International Journal of Computer Applications,* vol. 79, nº 8, 2013.

7. LISTADO DE SIGLAS, ABREVIATURAS Y ACRÓNIMOS

A

AJAX *Asynchronous JavaScript And XML*

API *Application Programming Interface*

C

CIV Contraste IntraVenoso

CSS *Cascading Style Sheets*

CSV *Comma-Separated Values*

G

GNU *Gnu Not UNIX*

GPL *General Public License*

GPS *Global Positioning System*

H

HTML *HyperText Markup Language*

HTML5 Versión 5 del lenguaje HTML

J

JSON

K

KD *Knowledge Discovery*

KDD *Knowledge Discovery in Databases*

KDT *Knowledge Discovery in Text*

M

MIT *Massachusetts Institute of Technology*

N

NLP *Natural Processing Language*

R

RIA *Rich Internet Applications*

RM Resonancia Magnética

RMN Resonancia Magnética Nuclear

RSNA *Radiological Society of North America*

S

SGML *Standard Generalized Markup Language*

SIG Sistema de Información Geográfica

SSD *Solid State Drive*

SVG *Scalable Vector Graphics*

T

TAC Tomografía Axial Computarizada

TDM *Term Document Matrix*

W

W3C *World Wide Web Consortium*

X

XML *eXtensible Markup Language*