



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica

Universitat Politècnica de València

# **Técnicas de minería de datos para la detección de la visibilidad web de empresas españolas**

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Muñoz Edo, Gloria Amparo

**Tutores:** Calduch Losa, Ángeles

Orduña Malea, Enrique

2015 / 2016



## Agradecimientos

Agradezco a mis tutores, Ángeles Calduch y Enrique Orduña, su apoyo, interés y orientación, a lo largo del desarrollo de este proyecto y el hacerme sentir como una más del equipo.

# Resumen

---

La extracción de datos de la Web y su posterior análisis, le sirve a las empresas para la toma de decisiones a nivel estratégico. Pero para poder analizar esos datos, es necesario estructurarlos previamente en una base de datos. Cuando se trabaja con grandes cantidades de datos, estas tareas se hacen tediosas y repetitivas, por lo que se convierte en imprescindible la automatización de estos procedimientos.

En este trabajo, se pretende establecer las bases tecnológicas para automatizar los procesos de estructuración de los datos Web, importación de éstos a una base de datos y posterior extracción de los mismos para que puedan ser analizados y visualizados. Para ello, se aplican las técnicas diseñadas a un caso de estudio concreto, como son las empresas españolas del sector vinícola. Estas técnicas consisten en la implementación de una serie de *scripts* en el lenguaje de programación *Python* (para realizar las tareas de estructuración e importación de los datos a una base de datos) y de otra serie de *scripts* en *R* (para la parte final de prueba del análisis y visualización).

Se ha logrado automatizar gran parte de todo el proceso, queda pendiente de resolver, en el origen de la extracción de los datos, el problema de la existencia de caracteres mal codificados, imposibles de recodificar de manera automatizada tras su exportación a ficheros.

**Palabras clave:** Webmetría; Indicadores web; Bases de datos; Motores de búsqueda; *Web data mining*; Automatización de procesos.

# Abstract

---

Data extraction and its subsequent analysis from the Web serve companies to decision-taking activities at strategic level. However, in order to correctly analyze these data, it is compelled to previously structure them in a database. When it comes to work with large amounts of data, these procedures turn tedious and repetitive, especially when data come from different sources and different formats, preventing their direct analysis.

This work intends to establish the technological foundations to automate some web data structure processes by importing them into a database to be analyzed and visualized automatically through exporting routines. To do this, the designed techniques are applied to the case study of those Spanish companies belonging to the wine market. These techniques consist on the one hand of the implementation of a set of scripts written in *Python* (responsible for the structure and data import from the native files from to a database), and on the other hand a set of scripts written in *R* (responsible for the analysis and visualization of the results).

Much of the whole process has been automated. However, the problem about characters wrongly codified in the data extraction starting process remains unresolved, since these characters could not been recodified automatically after exporting them into files.

**Keywords:** Webometrics; Web indicators; Databases; Web data mining; Processes automating.



# Índice general

---

<b>1. Introducción .....</b>	<b>1</b>
1.1. Marco contextual .....	2
1.1.1. Minería Web .....	2
1.1.2. Cibermetría y Webmetría.....	6
1.2. Problemática a resolver.....	9
1.3. Objetivos .....	11
1.4. Estructura del trabajo .....	12
<b>2. Metodología .....</b>	<b>14</b>
2.1. Material de trabajo .....	14
2.2. Software utilizado .....	14
2.3. Lenguajes de programación y librerías utilizadas.....	14
2.4. Obtención de las muestras .....	15
2.4.1. Muestra 1: Menciones Web .....	18
2.4.2. Muestra 2: Impacto Web.....	19
2.5. Creación de un almacén para los datos .....	25
2.5.1. MySQL .....	25
2.5.2. MAMP .....	27
2.5.3. MySQL Workbench.....	28
2.5.4. Python .....	31
2.6. Análisis de los datos.....	32
2.6.1. R.....	33
2.6.2. RStudio .....	34
2.6.3. Prueba de análisis.....	36

<b>3. Resultados</b> .....	<b>38</b>
3.1. Creación de un almacén para los datos .....	38
3.1.1. Ficheros maestros.....	38
3.1.2. Muestra 1 .....	40
3.1.3. Muestra 2 .....	49
3.1.4. Síntesis del proceso.....	53
3.2. Análisis de los datos con R y RStudio .....	55
3.2.1. Citation Flow y Trust Flow.....	55
3.2.2. External Back Links.....	56
3.2.3. Gráficas .....	57
<b>4. Discusión</b> .....	<b>61</b>
<b>5. Conclusiones</b> .....	<b>65</b>
<b>6. Bibliografía</b> .....	<b>68</b>

---

## Anexos

Anexo I - Scripts en Python .....	72
Anexo II - Scripts en R .....	96
Anexo III - Gráficas obtenidas con R .....	104

---

# Índice de figuras

---

Figura 1.1.	Crecimiento de la WWW en el segundo trimestre de 2016.....	2
Figura 1.2.	Instantánea global indicadores estadísticos digitales de relevancia, Ene/2016 ..	3
Figura 1.3.	Pronóstico de tráfico IP mundial mensual previsto para 2015 - 2020.....	4
Figura 1.4.	Mapa conceptual de la clasificación de minería Web Fuente: J. C. Dürsteler .....	5
Figura 1.5.	Fases de la minería Web .....	6
Figura 1.6.	Relación entre Bibliometría, Informetría, Cienciometría, Cibermetría y Web ....	7
Figura 2.1.	Esquema del proceso de obtención de las muestras .....	15
Figura 2.2.	Detalle del contenido del fichero master_spa de empresas españolas.....	17
Figura 2.3.	Detalle del contenido del fichero master_int de empresas internacionales.....	17
Figura 2.4.	Detalle del contenido del fichero de hits, en formato CSV.....	19
Figura 2.5.	Ejemplo de visualización de la herramienta Site Explorer de Majestic .....	21
Figura 2.6.	Detalle del fichero vino160404 en formato CSV, Abril de 2016 .....	23
Figura 2.7.	La "fuerza" de cada página se transmite a las demás mediante los enlaces.....	24
Figura 2.8.	Esquema gráfico del proceso de creación de la base de datos .....	25
Figura 2.9.	Sitio web de descarga del servidor MySQL .....	26
Figura 2.10.	Proceso de instalación de MySQL en una máquina Mac OS X.....	26
Figura 2.11.	Sitio web de descarga de la aplicación MAMP .....	27
Figura 2.12.	Proceso de instalación de la aplicación MAMP en una máquina Mac OS X.....	28
Figura 2.13.	Interfaz gráfica de aplicación MAMP y detalle de configuración de puert .....	28
Figura 2.14.	Sitio web de descarga de la aplicación MySQL Workbench .....	29
Figura 2.15.	Detalle de la instalación de MySQL Workbench en una máquina Mac OS X .....	29
Figura 2.16.	Pantalla inicial de la aplicación MySQL Workbench. ....	30
Figura 2.17.	Configuración de la nueva conexión .....	30
Figura 2.18.	Interfaz gráfica de trabajo de MySQL Workbench. ....	31
Figura 2.19.	Sitio web para la descarga del paquete de instalación de Python 3.5.2.....	32
Figura 2.20.	Sitio web para la descarga de R .....	33
Figura 2.21.	Detalle instalación de R en una máquina Mac OS X.....	34
Figura 2.22.	Sitio web para la descarga de RStudio .....	34
Figura 2.23.	Detalle descarga del instalador de RStudio para Mac OS X .....	35
Figura 2.24.	Interfaz de trabajo de RStudio para máquina Mac OS X .....	35
Figura 2.25.	Esquema proceso de prueba del análisis con R. ....	36
Figura 3.1.	Detalle del fichero master_Spa y de la fórmula de las claves UID.....	39
Figura 3.2.	Detalle del fichero master_int y de la fórmula de las claves UID .....	40
Figura 3.3.	Fichero master_int con las columnas añadidas de las claves UID nacionales...40	
Figura 3.4.	Fichero de hits con los datos separados en tres columnas .....	41
Figura 3.5.	Fichero de hits con las dos columnas de las claves UID añadidas.....	42
Figura 3.6.	Fichero master_int relleno con el número de nombramientos .....	43
Figura 3.7.	Detalle de la ejecución del script de llenado del fichero master_int.....	43
Figura 3.8.	Detalle proceso importación a MySQL Workbench del fichero master_int .....	44
Figura 3.9.	Detalle proceso importación a MySQL Workbench del fichero master_int, ventana de configuración de los ajustes de importación. ....	45
Figura 3.10.	Detalle proceso importación a MySQL Workbench del fichero master_int, proceso de importación terminado con éxito.....	45
Figura 3.11.	Detalle proceso importación a MySQL Workbench del fichero master_int, resultado de la importación.....	46
Figura 3.12.	Vista en MySQL Workbench de tabla correspondiente al fichero master_spa ..	46
Figura 3.13.	Vista en MySQL Workbench de tabla correspondiente al fichero master_int....	46
Figura 3.14.	Detalle de configuración de la clave primaria en la tabla vino_master_spa.....	47
Figura 3.15.	Pantalla que muestra en lenguaje SQL los cambios que se van a aplicar.....	47



Figura 3.16.	Detalle del error de ejecución del script 7, así como de su ejecución exitosa.....	48
Figura 3.17.	Detalle en MySQL Workbench de la tabla correspondiente al fichero de hits ....	49
Figura 3.18.	Detalle configuración de las claves primarias de la tabla correspondiente al fichero de hits .....	49
Figura 3.19.	Detalle de la ejecución del script 10.....	50
Figura 3.20.	Detalle de configuración de las relaciones entre las tablas .....	51
Figura 3.21.	Ventana de verificación de los cambios realizados .....	51
Figura 3.22.	Detalle de la relación entre las claves primarias de las tablas.....	52
Figura 3.23.	Esquema de la síntesis del proceso para el fichero de hits (Muestra 1).....	54
Figura 3.24.	Esquema de la síntesis del proceso para los ficheros de Majestic (Muestra 2)...	54
Figura 3.25.	Gráfica dispersión ratio Citation Flow/Trust Flow de tabla vino160404 .....	55
Figura 3.26.	Detalle de la ejecución del script 1 .....	56
Figura 3.27.	Esquema del proceso del script 1 .....	56
Figura 3.28.	Gráfica series temporales de los indicadores EBL de la tabla vino160404. ....	57
Figura 3.29.	Esquema del proceso del script 2.....	57
Figura 3.30.	Detalle ejecución script 3. Menú principal.....	58
Figura 3.31.	Resultado de la ejecución del script 3 para el indicador CF.....	58
Figura 3.32.	Resultado de la ejecución del script 3 para el indicador TTF .....	59
Figura 3.33.	Detalle ampliado de los diagramas de barras para los indicadores TTF.....	60
Figura 3.34.	Esquema del proceso del script 3.....	60
Figura 4.1.	Topten del ranking de gestores de bases de datos, Agosto 2016.....	61

---

# 1. Introducción

---

Hoy en día, nuestra sociedad genera ingentes cantidades de información digital, que junto con el aumento de la necesidad en la capacidad de almacenamiento de las bases de datos, han hecho que todo tipo de entidades, ya sean particulares, organizaciones gubernamentales o empresas privadas, tengan la posibilidad de disponer de una gran cantidad y variedad de datos relativos a su actividad diaria (Belinchón Monjas, 2011).

Los datos Web generados por una empresa reflejan: a) qué actividades realiza la empresa y cómo las realiza; y b) qué impacto tienen esos contenidos en otros usuarios y empresas. Tanto el tamaño como el impacto de estos contenidos (medidos tanto seccional como longitudinalmente) son de un indudable valor en los procesos de toma de decisiones estratégicas, si son analizados y contextualizados de forma adecuada (Blog sobre Business Intelligence, 2016).

Sin embargo, mucha de la información que se recoge en las bases de datos no se encuentra estructurada, por lo que resulta muy difícil de explotar desde el punto de vista estadístico, proceso que es imprescindible para que estos datos puedan ser empleados en las tomas de decisiones empresariales. Para poder utilizarla se necesita un proceso de tratamiento y análisis exhaustivo de los datos recogidos, a este proceso se le conoce como *minería de datos* o *data mining* y se define como "el proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos" (Witten & Frank, 2005).

Se podría decir que la *minería de datos* se enfrenta a dos retos importantes (Hasperué, 2014):

1. Trabajar con grandes cantidades de datos provenientes de sistemas de información, con los problemas que ello conlleva como ruido, datos no disponibles, intratables o volátiles, diferentes fuentes y formatos, etc.
2. Usar las técnicas más adecuadas para analizar esos datos y extraer de ellos nuevos conocimientos e información útil.

La *minería de datos* es tan sólo una etapa de un proceso más amplio, conocido como *extracción de conocimiento a partir de datos* (KDD, del inglés *Knowledge Discovery from Databases*). Este proceso tiene varias fases que incorporan diversas técnicas de campos como estadística, aprendizaje automático, bases de datos y gestión de la información (Hernández Orallo, Ramírez Quintana, & Ferri Ramírez, 2004).

El primer paso en el proceso de *extracción de conocimiento a partir de datos* es conocer y recopilar los datos con los que se va a trabajar, para ello se pueden utilizar técnicas de *minería de datos* y también otras técnicas relacionadas con otras áreas de conocimiento como son (Hasperué, 2014):

- La *minería de texto*: "Proceso que descubre información útil que no está presente explícitamente en ninguno de los documentos objeto de análisis y que



surge cuando se estudian adecuadamente y se relacionan dichos documentos" (Xu, Kurz, Piskorski, & Shmeier, 2002).

- La *minería stream*: "Cualquier operación realizada para extraer y analizar textos procedentes de distintas fuentes externas con el objetivo de obtener inteligencia, ... descubrimiento de información y conocimiento que anteriormente no se conocía, a partir de corpus textuales" (Sullivan, 2001).
- La *minería Web*: "El proceso global de descubrir información o conocimiento potencialmente útil y previamente desconocido a partir de datos de la Web" (Etzioni, 1996).

En lo que concierne a este trabajo, se va a centrar en esta última área de conocimiento, la *minería Web* o *Web mining*.

## 1.1. Marco contextual

### 1.1.1. Minería Web

Las técnicas de *minería Web* difieren significativamente de las técnicas utilizadas por la *minería de datos* propiamente dicha (Molina Félix, 2002), ya que los datos hay que extraerlos de un repositorio de gran tamaño, como es la *World Wide Web* (la Web, a partir de ahora), donde se encuentran documentos que contienen datos de tipos muy diversos (texto, imágenes, contenido multimedia, etc.), son por tanto datos no estructurados<sup>1</sup> o semiestructurados<sup>2</sup>, a diferencia de los datos estructurados<sup>3</sup> que contienen las bases de datos, por lo que su extracción, tratamiento y análisis no es trivial.

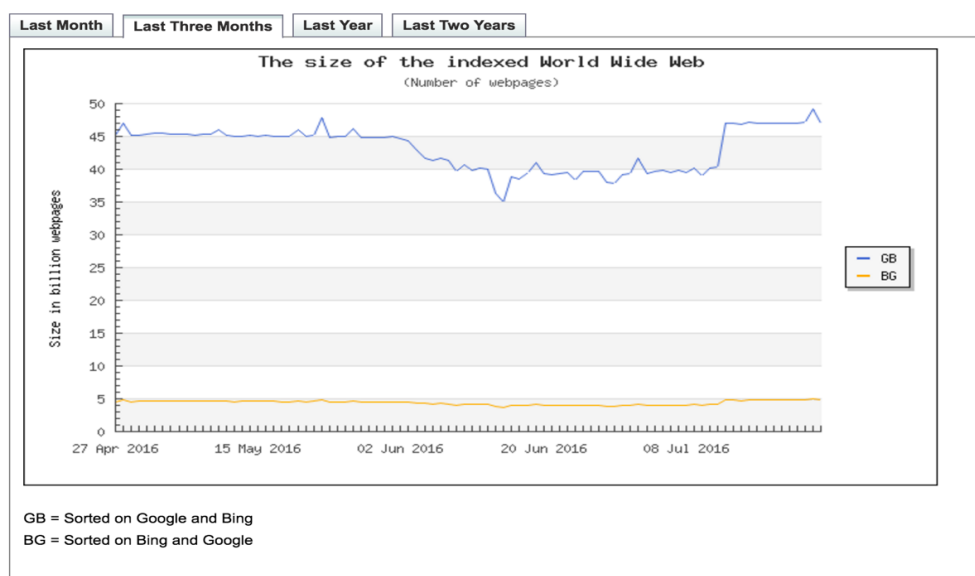


Figura 1.1. Crecimiento de la WWW en el segundo trimestre de 2016.  
Fuente: <http://www.worldwidewebsite.com/>

<sup>1</sup> Datos que no representan ningún esquema para la información que contienen.

<sup>2</sup> Datos que no representan un esquema rígido, ya que es implícito y está contenido en los propios datos.

<sup>3</sup> Datos que representan un esquema rígido, bien definido y diferenciado.

Respecto al tamaño de la Web, es necesario dar algunos datos referentes a su crecimiento, para tener una visión más amplia acerca del futuro y del motivo que hace urgente el organizar esta información y optimizar su acceso y tratamiento. La Web consta con cerca de 4,76 billones de páginas (datos estimados a fecha de 26 de Julio de 2016, medido en función del número de páginas web indizadas en el motor de búsqueda *Google*) (Kunder, 2016), además crece exponencialmente, ya que se crean cerca de 1,5 millones de páginas diariamente (véase Figura 1.1).

Actualmente, más de 3.419 billones de personas usan internet, lo que supone algo más del 46% de la población mundial, según los datos de *We Are Social* para Enero de 2016 (We Are Social, 2015).

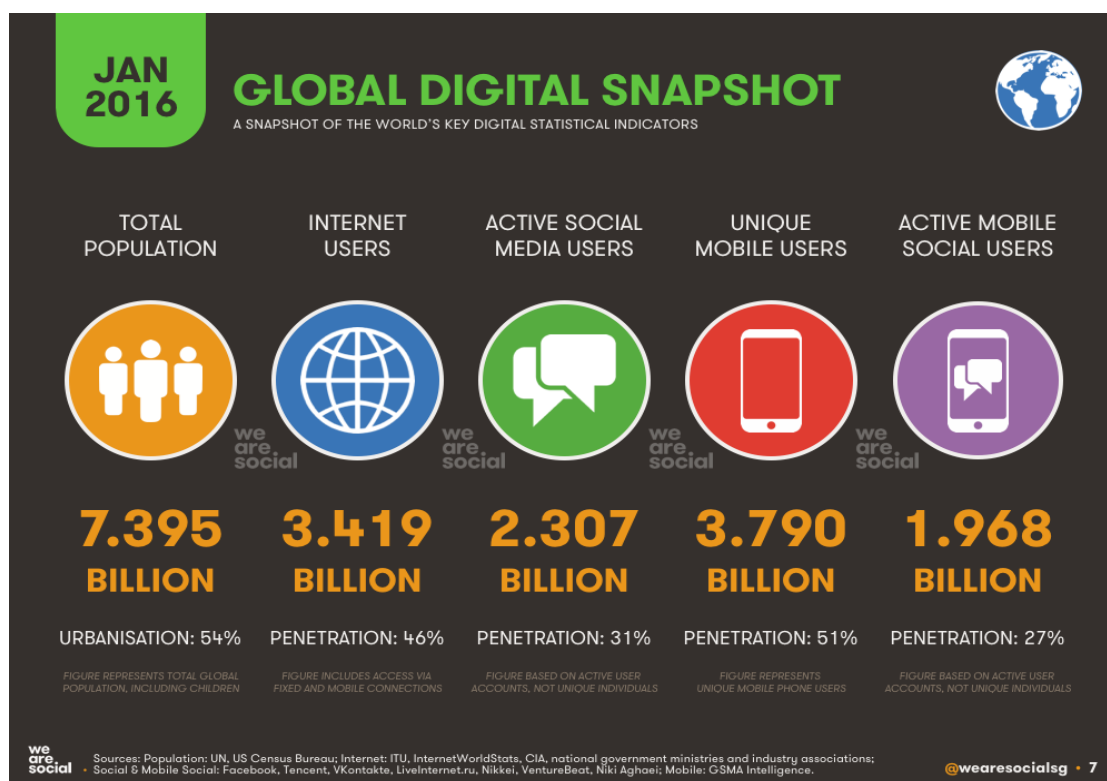


Figura 1.2. Instantánea global de indicadores estadísticos digitales de relevancia, Enero de 2016.  
Fuente: <http://wearesocial.com>

El alcance de la Web arroja números difíciles de comprender con una simple lectura, dado su orden de magnitud. Por ejemplo, según datos de *Cisco Systems* (2016), el tráfico IP mundial en 2015 se sitúa en 72,5 Exabytes (1EB =  $10^{17}$  Bytes) por mes y, según sus previsiones, se triplicará para el año 2020, llegando a 194,4 EB mensuales (véase Figura 1.3).

IP Traffic, 2015-2020							
	2015	2016	2017	2018	2019	2020	CAGR 2015-2020
<b>By Type (PB per Month)</b>							
Fixed Internet	49,494	60,160	73,300	89,012	108,102	130,758	21%
Managed IP	19,342	22,378	25,303	28,155	30,750	33,052	11%
Mobile data	3685	6180	9931	14,934	21,708	30,564	53%
<b>By Segment (PB per Month)</b>							
Consumer	58,539	72,320	89,306	109,371	133,521	162,209	23%
Business	13,982	16,399	19,227	22,729	27,040	32,165	18%
<b>By Geography (PB per Month)</b>							
Asia Pacific	24,827	30,147	36,957	45,357	55,523	67,850	22%
North America	24,759	30,317	36,526	43,482	50,838	59,088	19%
Western Europe	11,299	13,631	16,408	19,535	23,536	27,960	20%
Central and Eastern Europe	5205	6434	8116	10,298	13,375	17,020	27%
Latin America	4500	5491	6705	8050	9625	11,591	21%
Middle East and Africa	1930	2698	3822	5380	7663	10,865	41%
<b>Total (PB per Month)</b>							
Total IP traffic	72,521	88,719	108,533	132,101	160,561	194,374	22%

## Definitions

**Consumer** - Includes fixed IP traffic generated by households, university populations, and Internet cafés

**Business** - Includes fixed IP WAN or Internet traffic generated by businesses and governments

**Mobile** - Includes mobile data and Internet traffic generated by handsets, notebook cards, and mobile broadband gateways

**Internet** - Denotes all IP traffic that crosses an Internet backbone

**Managed IP** - Includes corporate IP WAN traffic and IP transport of TV and VoD

Figura 1.3. Pronóstico de tráfico IP mundial mensual previsto para 2015 - 2020.

Fuente: <http://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf>

Por consiguiente, con toda esa enorme cantidad de información en línea, la Web es un área fértil para la investigación de *minería de datos* y obviamente de *minería Web*, la cual en el ámbito del acceso, de la recuperación y de la organización de información, tiene amplios campos de aplicación en la Web (Reyes & Lobaina, 2007), como por ejemplo:

- Los motores de búsqueda.
- El comercio electrónico.
- El diseño Web.
- El posicionamiento Web.
- La seguridad.

La *minería Web* se subdivide en áreas que abarcan el contenido del sitio, la estructura de navegación y el comportamiento de los usuarios. La explotación de la información que se encuentra en la Web, se puede realizar por tanto desde diferentes puntos de vista (Liu, 2007):

- *Minería Web de contenido*: Trata de analizar el contenido que se puede encontrar o extraer de la Web, este punto de vista se enfoca principalmente en la extracción de conocimiento sobre el contenido de documentos.
- *Minería de uso Web*. Tiene como principal objetivo extraer el tráfico y patrones de uso de la Web por parte de los usuarios. Al realizar una navegación por la Web, los usuarios dejan huellas digitales (direcciones IP, cookies, navegadores, páginas web visitadas, etc.). Esas huellas digitales se almacenan en los ficheros *log* (ficheros de registros de sucesos, eventos o transacciones) de los servidores web.
- *Minería Web de estructura*: Intenta descubrir el modelo subyacente de las estructuras de los enlaces del web, el cual se basa en la topología de los

hiperenlaces. Este modelo puede ser usado para categorizar las páginas web y es útil para generar información, como la calidad de una página web o la relación entre diferentes páginas web (De Gyves Camacho, 2009). Es decir, pretende revelar la estructura real de un sitio web a través de la recogida de datos referentes a su estructura y, principalmente, a su conectividad. Tiene en cuenta dos tipos de enlaces: estáticos y dinámicos. El presente trabajo se enmarca en este último campo, el de la *minería Web de estructura*.

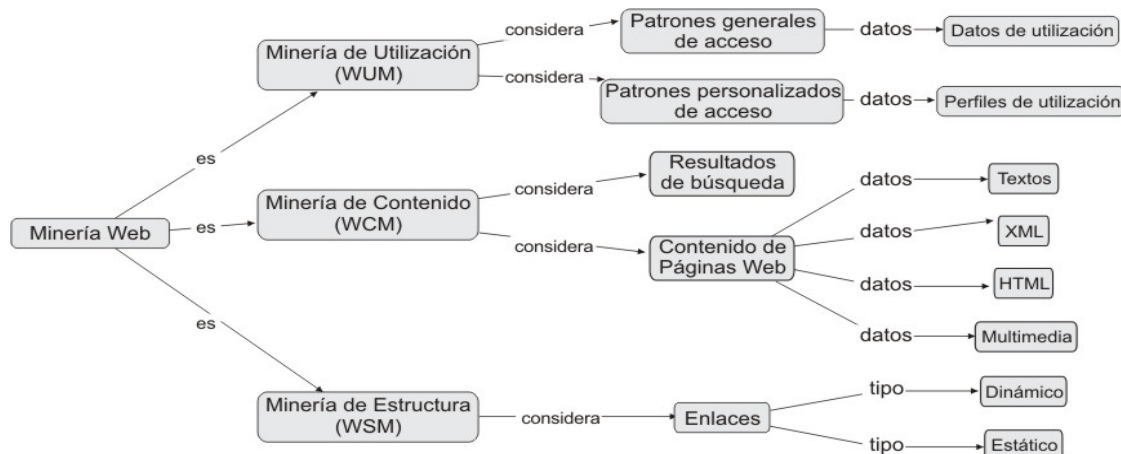


Figura 1.4. Mapa conceptual de la clasificación de la minería Web  
Fuente: Juan Carlos Dürsteler (Dürsteler, 2002).

El procesamiento de los ficheros *log*, que se generan automáticamente en los servidores, también produce información muy valiosa, como por ejemplo cómo mejorar la estructura de un sitio web, con objeto de crear una navegación más efectiva y un acceso más eficiente. Algunas herramientas de *minería Web* analizan y procesan esos ficheros de datos para producir información significativa, como puede ser la navegación de un cliente al hacer una compra en línea. La información de uso de la Web capta actividades del usuario en línea y descubre una gran variedad de patrones de conducta diferentes que recogen el comportamiento de los usuarios (Ortega & Aguillo, 2013).

Para poder procesar los datos y transformarlos en información útil, se llevan a cabo una serie de etapas dentro del proceso global de la *minería Web* (Hernández Orallo, Ramírez Quintana, & Ferri Ramírez, 2004):

- *Descubrimiento de las fuentes o recolección de datos*: Consiste en la recuperación automática de la información relevante (documentos y servicios en la Web) para su posterior procesamiento. El objetivo de esta etapa es recuperar automáticamente los documentos más relevantes, indexándolos para optimizar la búsqueda.
- *Selección y preprocesado de la información*: Una vez recuperados los documentos, se ordenan, limpian y preparan para la próxima etapa, utilizando herramientas para obtener esa información de forma automática. Se eliminan los datos erróneos o incompletos.
- *Generalización o descubrimiento de patrones*: Descubrir patrones generales desde los sitios web individuales, así como desde múltiples sitios. Existen múltiples técnicas aplicables al descubrimiento de patrones, como el

agrupamiento y clasificación o como el establecimiento de reglas de asociación y el hallazgo de secuencias frecuentes.

- *Análisis de patrones*: Comprende la interpretación, validación y visualización de los patrones minados.



Figura 1.5. Fases de la minería Web.

Fuente: [http://www.bvs.sld.cu/revistas/aci/vol16\\_4\\_07/aci111007.html](http://www.bvs.sld.cu/revistas/aci/vol16_4_07/aci111007.html)

En general, las técnicas de *minería Web* se han desarrollado para poder medir las masivas cantidades de datos e información que se distribuyen por la Web. Algunas de las disciplinas que, de forma parcial, han abordado el estudio y aplicación de estas técnicas son la Cibermetría (del inglés *Cybermetrics*) y la recuperación de información (*information retrieval*). Más específicamente, la *minería de datos* se ha aplicado a los archivos de transacciones registradas en los servidores web, con la intención de analizar las sesiones que se realizan. La *minería de uso Web* permite individualizar a cada usuario y reconstruir el recorrido completo de su visita (Ortega & Aguillo, 2009).

Otro aspecto importante en el ámbito de las Tecnologías de la Información es el uso de indicadores métricos, que al ser analizados permitan comprender, controlar y predecir hasta cierto grado un fenómeno en particular. Uno de esos indicadores es el análisis del consumo de información en la Web (Vicente Cuervo & López Menéndez, 2008), que puede ser una importante herramienta cibernétrica, para conocer no sólo las visitas y visitantes que recibe una página web, sino también los patrones de comportamiento, lo que puede ayudar a diseñar su estructura y contenidos.

Llegados a este punto, se va a continuar ahondando en lo que consiste la Cibermetría y por extensión la Webmetría (del inglés *Webometrics*).

### 1.1.2. Cibermetría y Webmetría

Dada la importancia y extensión que ha adquirido la Cibermetría en los últimos tiempos, se hace necesario conocer algunas definiciones de conceptos relacionados con la Bibliometría aplicada a la Web o Internet, es decir, la Cibermetría. Este área de conocimiento está en pleno estudio y desarrollo, por lo que existe una variedad de términos bastante notable en muy corto espacio de tiempo, según se va avanzando en las investigaciones, ya que la nomenclatura todavía no está estandarizada (Arroyo, Ortega, Pareja, Prieto, & Aguillo, 2005).

La relación de la Cibermetría con otras ciencias afines, como Bibliometría, Informetría o Cienciometría, es de cierto solapamiento en algunas áreas, ya que comparten un mismo enfoque y un mismo ámbito, enmarcado por la Informetría y que es el de los estudios cuantitativos de la información (Arroyo, Ortega, Pareja, Prieto, & Aguillo, 2005). La Cibermetría se engloba dentro de la Bibliometría, únicamente cuando los contenidos web que se analizan son contenidos académicos o científicos.

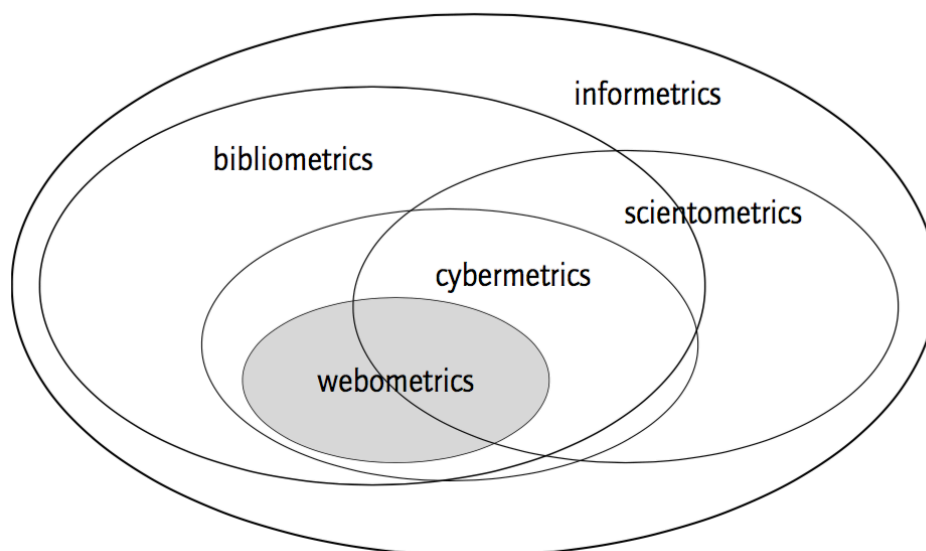


Figura 1.6. Relación entre Bibliometría, Informetría, Cienciometría, Cibermetría y Webmetría.  
Fuente: Adaptado a partir de Björneborn e Ingwersen (2004)

El origen de la Cibermetría se podría situar a mediados de los años 90, con la publicación de la revista *Cybermetrics* en 1997. En sus inicios tenía un marcado carácter descriptivo, sus principales objetivos se dirigían hacia aspectos como el estudio de la evolución del tamaño de la Web o la descripción de los motores de búsqueda (Arroyo, Ortega, Pareja, Prieto, & Aguillo, 2005).

Se propusieron varios términos para designar la nueva disciplina, como señala Björneborn (Björneborn & Ingwersen, 2004): *Netometry*, *Webometry*, *Internetrometry*; aunque finalmente se adoptaron dos, que, sin llegar a serlo, a menudo se emplean como sinónimos: *Cybermetrics* (que toma su nombre de la revista electrónica *Cybermetrics*) y *Webometrics* (Almind & Ingwersen, 1997). Para su traducción al español se hizo la adaptación directa del inglés, dando así lugar a Cibermetría y Webmetría respectivamente.

Para distinguir ambos conceptos, hay que fijarse en su ámbito de actuación y en las definiciones que propone Björneborn (Björneborn & Ingwersen, 2004). Según él, la Cibermetría puede ser entendida como "el estudio de los aspectos cuantitativos de la construcción y uso de los recursos de información, estructuras y tecnologías de internet, desde perspectivas bibliométricas e informétricas", mientras que la Webmetría puede definirse exactamente de la misma forma, con la salvedad de que se refiere solo a una parte de internet, es decir, a una parte de la Web. Esto significa que la Cibermetría acoge todo el espectro de análisis de la Web y la Webmetría selecciona una parte de ella, una sección o localización muy concreta.

Una de las aplicaciones directas de la Cibermetría (Orduña-Malea & Aguillo, 2014) es su utilización en la elaboración de informes de mercado, es decir, en el análisis de un sector industrial a partir de la información presente en la Web y del uso y consumo de ésta por parte de los usuarios. Pero su aplicación en sectores de actividad industrial es aún muy escasa, ya que una empresa no suele enlazar a otras del mismo sector, lo que dificulta los análisis de enlaces y menciones en el mundo comercial.



También se ven limitadas la precisión de los indicadores de tamaño y audiencia web, ya que, debido a temas de propiedad industrial, en ocasiones es prácticamente imposible acceder a ciertos contenidos.

Dado el gran auge que está experimentando el mercado *online*, la aplicación de indicadores cibernéticos para caracterizar estas actividades resulta de vital importancia, ya que gracias a la Cibermetría se pueden capturar los movimientos de este mercado de forma mucho más precisa que con técnicas generales de *minería de datos*, que pese a ofrecer información valiosa y difícilmente obtenible mediante otros mecanismos o procesos, presentan ciertas limitaciones y problemas (Fader, 2007):

- Están basadas fundamentalmente en datos de uso y consumo de información (la huella digital que dejan los usuarios), pero no tienen en cuenta otros parámetros, procesos e indicadores, tales como la creación e impacto de los contenidos, infraestructuras y servicios en Internet.
- La mayoría de las soluciones informáticas existentes tienen precios muy elevados y su adecuación a las dinámicas de los sitios de redes sociales es cuestionable.
- La difusión y marketing de estos productos se centra excesivamente en aspectos técnicos, que les son totalmente desconocidos al usuario.
- Suelen ser aplicaciones comerciales, y por consiguiente la metodología que utilizan en el tratamiento y procesamiento de la información no es pública.
- Las metodologías de análisis cuantitativas y métricas relacionadas con el *big data*<sup>4</sup> (Wikipedia, 2016) no se encuentran generalmente científicamente validadas, ni se relacionan con otras disciplinas métricas afines necesarias para comprender estos fenómenos, como por ejemplo la Informetría y las ciencias del conocimiento.

Respecto a los indicadores cibernéticos, mencionar que existen varios y de muy variadas tipologías para estudiar el contenido de la Web, a continuación se mencionan los más significativos y viables (Berrocal, Figuerola, & Zazo, 2004):

- **TAMAÑO:** El tamaño documental de un dominio o sede es el número total de páginas bajo dicho dominio. La unidad de trabajo es una dirección URL (del inglés *Uniform Resource Locator*, Localizador de Recursos Uniforme) que se puede obtener de cualquiera de los motores de búsqueda. El principal problema suele ser identificar subdominios o unidades de menor rango, cuando el volumen de páginas es muy elevado. Una variante de la medida de tamaño es centrarse en un formato determinado de documentos. Existen delimitadores específicos para extraer los llamados ficheros ricos, es decir, documentos en formatos tan populares como Adobe Acrobat (PDF), MS Word (DOC o RTF) o MS PowerPoint (PPT) entre otros.
- **VISIBILIDAD:** Uno de los indicadores más importantes, ya que mide el número total de enlaces externos diferentes recibidos por un dominio. Se pueden utilizar distintos buscadores. Puesto que solo se miden enlaces entrantes externos, hay

---

<sup>4</sup> Big data, macrodatos o datos masivos es un concepto que hace referencia al almacenamiento de grandes cantidades de datos y a los procedimientos usados para encontrar patrones repetitivos dentro de esos datos.

que eliminar los enlaces de navegación, es decir los que provienen del mismo dominio. La sintaxis obliga a solicitar el número de enlaces y a a excluir los del dominio propio. Una manera de analizar la visibilidad en detalle es clasificar el origen de los enlaces. Esto se puede realizar filtrando por dominio. Una alternativa interesante, pero más limitada a la medición de la visibilidad, es la obtención del rango de página (*PageRank*), que en *Google* combina el número de enlaces con el peso (importancia) de las páginas que lo originan.

- MENCION. El número de veces que un término o frase aparece mencionado en un motor de búsqueda. En Cibermetría académica se suele utilizar para localizar nombres de personas, de instituciones, el título de un artículo o un libro o simplemente una temática. El problema de esta aproximación son los lexemas, los nombres pueden aparecer con múltiples variantes y/o en diferentes idiomas.
- POPULARIDAD: Número y distribución de las visitas recibidas en un plazo determinado. Se utilizan conjuntamente sesiones y usuarios distintos. Resulta adecuado recurrir a listados ordenados según el número de visitas a través de un sistema prefijado.
- IMPACTO. Es fundamentalmente una variante del anterior, que resulta de dividir el número total de enlaces externos diferentes recibidos en un dominio, por su tamaño expresado en número de páginas. En la actualidad se utilizan otras alternativas, como el *Webometrics Rank* que se calcula combinando visibilidad y tamaño, no en valores absolutos, sino como posiciones de un listado global. Un ejemplo de aplicación de este sistema es el Ranking Mundial de Universidades basado en indicadores Web<sup>5</sup>.

## 1.2. Problemática a resolver

El presente trabajo forma parte de un proyecto de investigación de mayor envergadura del *Grupo Trademetrics*<sup>6</sup>, financiado por el Ministerio de Economía y Competitividad (MEC), con el título "*Propuesta metodológica para el análisis cibernético de productos, marcas, personas y empresas del mercado online español*" (CSO2013-46138-P).

Con el referido proyecto de investigación, se pretende identificar (Ruipérez & Malea, 2015):

- qué medir: catálogo de indicadores de naturaleza cibernética que permitan la medición de las dimensiones y categorías identificadas a partir del estudio de la literatura científica y profesional existente.
- dónde medir: establecimiento de una serie de criterios de selección de fuentes.
- cómo medir: método directo y el obtenido a través del API (del inglés *Application Programming Interface*) de la fuente, en el caso de que éste sea accesible.

---

<sup>5</sup> Véase para más información <http://www.webometrics.info/en>.

<sup>6</sup> Grupo de investigación orientado a la aplicación de indicadores web en la presencia e impacto de las empresas y marcas en la Web (<http://trademetrics.upv.es>).

Para tal fin, se recopilaron una serie de empresas (URL) del sector vino, que van a formar el objeto de estudio a modo de ejemplo. A partir de ahí, se midieron obteniendo, para cada URL, todos los indicadores cibernéticos de todas las fuentes establecidas en la fase previa.

Llegados a este punto, se necesita estructurar de alguna manera los datos brutos obtenidos y almacenarlos en una base de datos, para que posteriormente puedan ser analizados estadísticamente, con el propósito de conocer sus propiedades y relaciones. Se hace indispensable aquí, en la medida de lo posible, la automatización de todos estos procesos.

Otra finalidad del proyecto de investigación de *Trademetrics*, es estudiar diversos modelos predictivos (tanto lineales como paramétricos), con el fin de determinar la posibilidad de predecir futuras tendencias a partir de datos actuales con cierta precisión.

Con todo ello se alcanzará el resultado de la investigación, es decir, un modelo final optimizado de análisis cibernético de las empresas del sector vino, que se pueda hacer extensible al análisis de productos, marcas, personas y otras empresas del mercado *online* español.

Con el propósito de cumplir con los objetivos y tareas especificados en el proyecto de investigación de *Trademetrics*, fue diseñado un plan de trabajo constituido por 11 fases:

1. Diseño y creación del sitio web del proyecto.
2. Diseño conceptual previo del modelo de análisis.
3. Diseño y validación de indicadores.
4. Recopilación de fuentes.
5. Decisión del método de extracción de datos: directa, mediante API.
6. Diseño de la toma de datos.
7. Medición.
  - a. Toma de datos y almacenamiento en el sistema diseñado en la fase 6.
  - b. Normalización y depuración de los datos brutos.
  - c. Obtención de datos elaborados a partir de los datos brutos extraídos en la fase 5.
8. Análisis de los datos.
  - a. Análisis estadístico descriptivo y correlacional de los datos.
  - b. Modelos predictivos.
9. Comprobación del diseño previo y, en función de los resultados, diseño final del modelo de análisis.
10. Diseño de una intranet para la consulta y visualización de los datos obtenidos.
  - a. Desarrollo de un sitio web en la intranet.
  - b. Volcado automático de la información de valor a la intranet para su consulta y utilización.
  - c. Diseño de las técnicas de visualización de resultados
11. Discusión de los resultados obtenidos para conocer el estado del comercio online de los sectores españoles del vino y de la moda.

El presente proyecto se engloba en una parte del desarrollo de las fases 6, 7.a, 7.b, 7.c, 8.a y 10.b. Con este trabajo se pretende abordar la resolución de la problemática de cómo automatizar el proceso de almacenamiento de los datos extraídos (que se encuentran en soporte físico con formato CSV) en una base de datos, la realización de consultas a la base de datos, el análisis de la información y de los indicadores más relevantes y la visualización de los resultados. Así mismo, se desea utilizar *R* para toda la parte interna del análisis y comprobar si podría servir para programar el proceso de automatización y visualización de resultados.

La toma de decisiones se basa a menudo en la información que se encuentra en la Web, existen muchas formas de medir esa información. En el estudio de *Trademetrics* se hacen uso de las técnicas de la Webmetría, la cual se enfrenta a datos semiestructurados, para la toma y análisis de estos datos se precisa de la automatización de ciertos procesos y esta labor no es trivial, puesto que existen fuentes muy diversas, con codificación distinta, que ofrecen los datos en formatos diferentes, por lo cual es necesario disponer de un sistema que lo automatice y que no solamente gane tiempo, sino que también gane precisión, independientemente de la fuente de la que se obtengan los datos.

Los productos de software que existen en la actualidad, enfocados a la toma de datos Web, no suplen las necesidades que tiene la Cibermetría, la cual precisa de herramientas propias para sus análisis. De la lectura del artículo "*Fuentes de enlaces web para análisis cibernéticos*" (Orduña Malea, 2012), se extrae la conclusión de que existe una falta de *software* que permita automatizar la toma y análisis de datos webmétricos, considerando las diferentes fuentes de las que provienen, esto justifica la creación de un sistema de estas características, ya que hay una problemática o dificultad técnica a la hora de medir de una manera homogénea y en un tiempo razonable todos estos datos. Tener un sistema que pueda automatizar esas tareas, supone un avance importante en la mejora de los análisis de los datos y una inestimable ayuda en la toma de decisiones empresariales.

### 1.3. Objetivos

El **objetivo general del proyecto de Trademetrics** consiste en desarrollar y validar (para su futura aplicación) un modelo de análisis cibernético orientado al análisis sectorial del mercado *online* español para su internacionalización, centrándose en la presencia, consumo e impacto Web de empresas, productos y marcas. Además, en dicho proyecto, se propone identificar los factores que influyen en la difusión web de productos, marcas y empresas por sectores y que condicionan el mercado. La finalidad práctica es implementar una herramienta para su integración en una página web, que sea capaz extraer datos de la Web, analizar esos datos en función de su naturaleza y durante el tiempo, capacitar a la herramienta para realizar los tipos de consultas que se le soliciten y aplicar modelos predictivos para la toma de decisiones.

El **objetivo general del presente trabajo** es establecer cuáles son las tecnologías o procedimientos más óptimos o adecuados para lograr automatizar las tareas que se requieren, para procesar de forma automática los datos Web recopilados de un conjunto de empresas, en ficheros y formatos diferentes, con el fin de importarlos a una base de datos en un entorno web, estructurando esos datos y dejándolos



preparados, para que puedan ser analizados estadísticamente y visualizados, probando el entorno de *R* dentro de este análisis final. Para ello, se han establecido una serie de tareas específicas, cuyo cumplimiento llevará a obtener el objetivo final.

#### **Tareas específicas:**

- Diseñar un conjunto de claves únicas de identificación para las empresas del caso de estudio, que sirvan para su indexación en una base de datos.
- Integrar dichas claves en los ficheros maestros y en cada uno de los ficheros procedentes de la extracción de datos Web, con la finalidad de poder realizar búsquedas indexadas.
- Recomendar un almacén o base de datos que albergue la información recopilada de fuentes externas.
- Automatizar el proceso de almacenaje de los datos en una base de datos.
- Intentar maximizar el uso del lenguaje de programación *R* durante todo el proceso y en su defecto, sugerir un lenguaje alternativo para intentar optimizar aquellas tareas que con *R* resulten más tediosas, en cuanto a implementación y ejecución se refiere.
- Utilizar *R* para realizar las consultas a la base de datos y extraer la información que se precise.
- Probar la realización con *R* de análisis de los campos más significativos.
- Visualizar gráficamente los análisis realizados.

## 1.4. Estructura del trabajo

Tras el presente capítulo introductorio, se pasa a un **Segundo Capítulo** en el que se explica la metodología utilizada para resolver el problema, los pasos seguidos para lograr cumplir los objetivos, las técnicas empleadas para la extracción de las muestras, así como una breve introducción a estas técnicas, el material de trabajo y el software utilizado, además de las librerías y lenguajes de programación que se han usado para implementar los diferentes *scripts*<sup>7</sup> (Wikipedia, 2016), empleados para automatizar las tareas descritas en los objetivos.

En el **Tercer Capítulo** se analiza la solución aplicada, así como los resultados obtenidos, todo ello con mayor profundidad, describiendo todos los pasos que se han seguido en los procesos de obtención de las muestras, creación de un almacén para los datos, acceso desde *R* a los datos almacenados y posterior análisis de ciertos indicadores que se han considerado de mayor relevancia.

En el **Cuarto Capítulo** se discute la interpretación de los resultados obtenidos a la luz de la problemática planteada en el punto 1.2, haciendo sugerencias de posibles mejoras que se podrían introducir.

Finalmente, en el **Quinto Capítulo** se encuentran las conclusiones sobre el trabajo y sugerencias de futuros estudios relacionados con la investigación.

---

<sup>7</sup> Archivo de órdenes o de procesamiento por lotes, programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.

Para dar un mejor entendimiento a este trabajo, se incluyen tres **anexos**: Los anexos I y II contienen el código de los *scripts* implementados en *Python* y en *R* respectivamente y el anexo III contiene las gráficas que se obtienen como resultado de la ejecución del *script* 3 de *R*.

## 2. Metodología

---

En este capítulo se describen los recursos empleados y los pasos seguidos para resolver la problemática inicial y alcanzar los objetivos propuestos.

### 2.1. Material de trabajo

- MacBook Pro (Retina 13 pulgadas, principios de 2015).
- Procesador: Intel Core i5 de doble núcleo a 2,7 GHz.
- Memoria RAM: 8GB 1867 MHz DDR3.
- Sistema operativo: OS X El Capitán, versión 10.11.5.

### 2.2. Software utilizado

- Hoja de cálculo **Excel 2016** de *Microsoft Office* versión para Mac.
- Hoja de cálculo de **Libre Office** versión 2.0.
- **MAMP** versión 3.5: Aplicación que permite montar un servidor local, de forma rápida y sencilla, disponiendo de *Apache*, *MySQL* y *PHP*.
- **MySQL Community Server** versión 5.7.14: Sistema de gestión de base de datos relacional y servidor *MySQL*.
- **MySQL Workbench** versión 6.3.7: Herramienta visual de diseño de bases de datos que integra desarrollo de software, administración, diseño, creación y mantenimiento para el sistema de base de datos *MySQL*.
- **RStudio** versión 0.99.903: Entorno de desarrollo integrado para el lenguaje de programación *R*, para manipulación de datos, computación estadística, cálculo y gráficos.
- **Sublime Text** versión 2.0.2: Editor de texto y código.

### 2.3. Lenguajes de programación y librerías utilizadas

- **Python:** versión 3.5.1, las librerías que se han utilizado son:
  - *CSV* versión 3.5: Este módulo implementa clases para leer y escribir en archivos CSV.
  - *MYSQL.CONNECTOR* versión 2.0: Permite el acceso a bases de datos *MySQL*.
  - *OpenPyXL* versión 2.3.5: Librería de *Python* para escribir y leer archivos XLSX/XLSM de Excel 2010.
  - *OS* versión 3.5: Este módulo proporciona una manera sencilla de utilizar la funcionalidad del sistema operativo.
  - *SYS* versión 3.5.1: Módulo que permite el acceso a variables y funciones utilizadas por el intérprete.

- *TIME* versión 3.5.1: Este módulo ofrece varias funciones relacionadas con el manejo de la hora del reloj.
  - *WARNINGS* versión 2.1: Módulo que permite el manejo de los mensajes de advertencia o *warnings*.
- **R:** versión 3.3.1, las librerías que se han utilizado son:
- *DBI* versión 0.4-1: Define una interfaz de base de datos para la comunicación entre *R* y los sistemas de gestión de bases de datos relacionales.
  - *E1071* versión 1.6-7: Funciones para el análisis de clases, transformada de Fourier, agrupamiento difuso (*fuzzy clustering*), máquinas de vectores de soporte (*support vector machines*), clasificador de Naïve Bayes, ...
  - *RANDOMCOLOR* versión 1.0.0: Dispone de métodos sencillos para generar distintos colores atractivos de forma aleatoria.
  - *RMySQL* versión 0.10.9: Implementa la interfaz *DBI* para *MySQL* y bases de datos *MariaDB*.

## 2.4. Obtención de las muestras

Los procesos de automatización que se han llevado a cabo parten de dos ficheros, que contienen un listado con distintas URL y datos asociados (los datos son de diferente naturaleza en cada muestra). El trabajo realizado pretende automatizar los procesos de estructuración e importación de los datos, desde esos ficheros a la base de datos. Seguidamente, se pasa a detallar las muestras de datos con el fin de aclarar su naturaleza y composición (véase Figura 2.1).

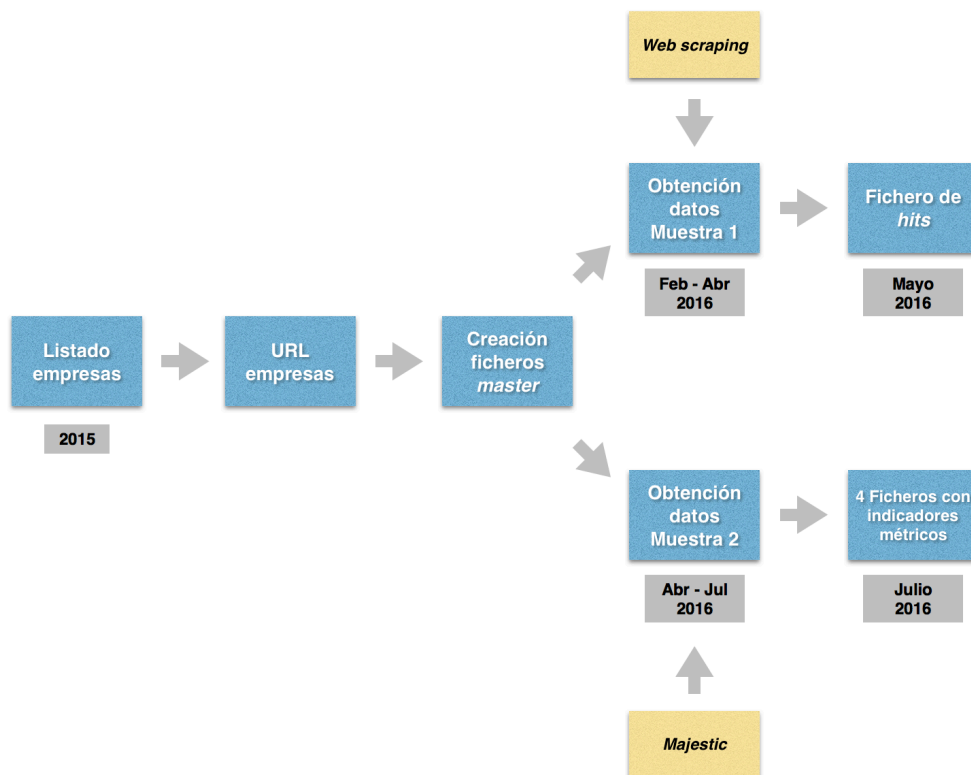


Figura 2.1. Esquema del proceso de obtención de las muestras.



Se dispone de dos listas maestras de empresas del sector vinícola, obtenidas en el año 2015, a través de varios procedimientos. Una de las listas se corresponde a 2.744 empresas españolas (bodegas); para su confección se utilizaron directorios oficiales (entre ellos *Camerdata*<sup>8</sup>), que se ampliaron con consultas directas en *Google* a través de palabras clave del sector. La otra lista contiene 331 empresas de renombre internacional (tiendas virtuales o blogs) especializadas en el sector vino. En su confección se utilizó *Google Ad Planner*<sup>9</sup>, para averiguar las principales palabras clave relacionadas con el vino. Después se fue incluyendo cada palabra en el buscador de *Google* y recogiendo manualmente las empresas que ocupaban las primeras posiciones, posteriormente se eliminaron los duplicados. Para finalizar, se obtuvieron todas las URL de las empresas de ambos listados.

Con los datos de estas empresas se confeccionaron dos ficheros maestros (ficheros *master*, a partir de ahora) que se guardaron en formato XLSX. Estos ficheros son los que se nos facilitan con los datos del caso de estudio, contienen las URL correspondientes a las páginas web y a diferentes recursos de redes sociales (*Facebook*, *Google+*, *LinkedIn*, etc.), de cada una de las empresas seleccionadas. Estas URL van a ser la unidad de trabajo para realizar la extracción de los datos de la Web.

Los ficheros *master* vienen en formato XLSX y los datos correspondientes al contenido de sus columnas son:

- **ID**: Número que sirve para enumerar la empresa dentro de todas las existentes.
- **EMPRESA**: Razón social de la empresa.
- **FUENTE**: Fuente o recurso de internet del que proviene la URL.
  - En el fichero de empresas internacionales, puede tomar los valores de: *Facebook*, *Google+*, *Instagram*, *LinkedIn*, *Pinterest*, *Site* (página web), *Twitter*, y *YouTube*.
  - En el fichero de empresas españolas, puede tomar los valores de: *Facebook*, *Instagram*, *LinkedIn*, *Site*, *Twitter*, y *YouTube*.
- **URL**: Identificador del recurso o fuente correspondiente.
- **TIPO** (sólo en el fichero de empresas internacionales): Identifica si la URL es una tienda virtual o un blog.
- **PAÍS** (sólo en el fichero de empresas internacionales): Identifica el país de origen, de entre tres posibles: Canadá, Estados Unidos o Reino Unido.

---

<sup>8</sup> Fichero de empresas españolas, <http://www.camerdata.es/index.php>.

<sup>9</sup> Véase para más información <https://adwords.google.com/KeywordPlanner?hl=es>.

ID	EMPRESA	FUENTE	URL1
1	1 1 + 1 = 3	INSTAGRAM	instagram.com/1mes1fan3
2	1 1 + 1 = 3	SITE	umesufan3.com
3	1 1 + 1 = 3	TWITTER	twitter.com/UmesUfan3
4	2 10 sentits	INSTAGRAM	instagram.com/10sentits
5	2 10 sentits	SITE	10sentits.cat
6	2 10 sentits	TWITTER	twitter.com/10Sentits
7	2 10 sentits	YOUTUBE	youtube.com/user/10sentits
8	3 15 albas	SITE	15albas.es
9	4 2amigos	FACEBOOK	facebook.com/2amigos-500500696793680
10	4 2amigos	SITE	2amigos.com
11	4 2amigos	TWITTER	twitter.com/2amigoswine
12	4 2amigos	YOUTUBE	youtube.com/user/2amigoscom
13	5 3 ases	FACEBOOK	facebook.com/TRES-ASES-VINO-147947088568166
14	5 3 ases	INSTAGRAM	instagram.com/bodega_3ases
15	5 3 ases	SITE	3asesvino.com
16	5 3 ases	TWITTER	twitter.com/3asesvino
17	5 3 ases	YOUTUBE	youtube.com/user/3AsesVino

Figura 2.2. Detalle del contenido del fichero master\_spa de empresas españolas.

ID	EMPRESA	FUENTE	URL2	TIPO	PAÍS
1	1 12x75.com	SITE	12x75.com	blog	Reino Unido
2	2 14 Hands Winery	FACEBOOK	facebook.com/14handswine	tienda virtual	Estados Unidos
3	2 14 Hands Winery	SITE	14hands.com	tienda virtual	Estados Unidos
4	2 14 Hands Winery	TWITTER	twitter.com/14handswine	tienda virtual	Estados Unidos
5	2 14 Hands Winery	YOUTUBE	youtube.com/user/14handswine	tienda virtual	Estados Unidos
6	3 50 <sup>th</sup> Parallel Estate	SITE	50thparallel.com	tienda virtual	Canadá
7	4 A Good Time with Wine	FACEBOOK	facebook.com/agoodtimewithwine	blog	Estados Unidos
8	4 A Good Time with Wine	SITE	agoodtimewithwine.com	blog	Estados Unidos
9	4 A Good Time with Wine	TWITTER	twitter.com/mmwine	blog	Estados Unidos
10	5 ABC Fine Wine & Spirits	FACEBOOK	facebook.com/ABCFineWineSpirits	tienda virtual	Estados Unidos
11	5 ABC Fine Wine & Spirits	LINKEDIN	linkedin.com/company/abc-fine-wine-&-spirits	tienda virtual	Estados Unidos
12	5 ABC Fine Wine & Spirits	PINTEREST	pinterest.com/abcfws	tienda virtual	Estados Unidos
13	5 ABC Fine Wine & Spirits	SITE	abcfws.com	tienda virtual	Estados Unidos
14	5 ABC Fine Wine & Spirits	TWITTER	twitter.com/abcwinecountry	tienda virtual	Estados Unidos
15	5 ABC Fine Wine & Spirits	YOUTUBE	youtube.com/user/ABCWineCountry	tienda virtual	Estados Unidos
16	6 Adam Puchta Winery	FACEBOOK	facebook.com/adampuchtawinery	tienda virtual	Estados Unidos
17	6 Adam Puchta Winery	GOOGLE+	plus.google.com/1104702769922459693241	tienda virtual	Estados Unidos
18	6 Adam Puchta Winery	PINTEREST	pinterest.com/adampuchtawine	tienda virtual	Estados Unidos
19	6 Adam Puchta Winery	SITE	adampuchtawine.com	tienda virtual	Estados Unidos
20	6 Adam Puchta Winery	TWITTER	twitter.com/AdamPuchtaWines	tienda virtual	Estados Unidos
21	7 Adnams Southwold	FACEBOOK	facebook.com/adnams	tienda virtual	Reino Unido
22	7 Adnams Southwold	GOOGLE+	plus.google.com/+adnams	tienda virtual	Reino Unido
23	7 Adnams Southwold	INSTAGRAM	instagram.com/adnams	tienda virtual	Reino Unido
24	7 Adnams Southwold	PINTEREST	pinterest.com/adnams	tienda virtual	Reino Unido

Figura 2.3. Detalle del contenido del fichero master\_int de empresas internacionales.

Por otro lado, también se dispone de unos ficheros con las métricas obtenidas por dos métodos distintos, que conforman las muestras 1 y 2 del caso de estudio. La muestra 1 recoge las menciones a las empresas internacionales y la muestra 2 refleja el impacto Web que tienen las empresas españolas. Estas muestras han sido proporcionadas por otro equipo de trabajo, perteneciente al proyecto de investigación de *Trademetrics*. Dicho grupo está realizando otro proyecto paralelo al actual, el cual se desarrolla sobre técnicas de extracción de datos de sitios web. A partir de estas muestras se trabaja sobre los ficheros para normalizarlos e importarlos a un almacén de datos, del que poder extraer las posteriores consultas y análisis, para que todo el proceso sea realizable de forma automatizada.

A continuación, se describe brevemente la metodología utilizada en la extracción de cada una de las muestras del caso de estudio y el detalle del contenido de cada muestra.

### 2.4.1. Muestra 1: Menciones Web

La extracción de datos para esta muestra se realiza a través de un programa implementado en *Python*, utilizando la técnica de *web scraping*<sup>10</sup> (Escuela de datos, 2016), obteniéndose, por el momento, un único fichero (fichero de *hits*, a partir de ahora) que contiene el número de nombramientos o menciones de la página web o *site* internacional en el *site* nacional. La extracción de los datos que contiene el fichero de *hits* se realizó durante los meses de Febrero, Marzo y Abril de 2016. Este fichero tiene 908.264 entradas, que corresponden a la combinación de los 331 *sites* de empresas internacionales, multiplicados por los 2.744 *sites* de empresas españolas.

En un futuro, está pensado obtener también el número de nombramientos de *sites* de empresas españolas en *sites* de empresas internacionales; también se desea obtener, en próximas etapas del proyecto de investigación, el número de nombramientos del resto de recursos web o URL almacenados en los ficheros *master*, con el objetivo de tener un mayor campo de visión a la hora de analizar el impacto Web que tienen las empresas españolas del sector vinícola.

#### 2.4.1.1. *Web scraping*

El *web scraping* (también llamado *screen scraping*, *web data extraction* o *web harvesting*) es una técnica empleada para extraer grandes cantidades de datos de sitios web (Neuman, 2013). Se trata de un programa que interactúa con los sitios web de la misma manera que un navegador web, pero en lugar de mostrar los datos servidos por el sitio web en la pantalla, el programa guarda esos datos extraídos en un archivo local o base de datos, en forma de tabla (hoja de cálculo).

Los datos que se encuentran en la mayoría de sitios web sólo se pueden ver utilizando un navegador web. Ejemplo de ello son los listados de directorios de páginas amarillas, las redes sociales, sitios de compras en línea, etc. (Guía de posicionamiento web, 2012). La mayoría de estos sitios web no ofrece la funcionalidad de poder exportar una copia de los datos que se muestran de forma estandarizada, a una base de datos u hoja de cálculo. Las únicas opciones posibles son entonces copiar y pegar los datos manualmente desde el navegador en un archivo local, o imprimir la página web o guardar la página en local. Éste es un trabajo muy tedioso que puede tomar muchas horas o incluso días para completarlo.

El *web scraping* es la técnica de automatización de este proceso, de modo que en lugar de copiar manualmente los datos de los sitios web, el programa de "raspado" (traducción literal de *scraping* al español) llevará a cabo la misma tarea en mucho menos tiempo.

---

<sup>10</sup> Traducido en español significa "escarbar o raspar la Web". *Web scraping* es una técnica que sirve para extraer información de páginas web de forma automatizada, a través de un programa que simula la navegación de un humano.

### 2.4.1.2. Detalle

El fichero de *hits* (nombramientos) viene en tres formatos diferentes: CSV, TXT y XLSX, conteniendo cada uno de estos ficheros la misma información. Este fichero tiene datos de tres campos distintos, que son:

- **URL1:** URL internacional nombrada.
- **URL2:** URL nacional en la que se efectúa el nombramiento.
- **HIT\_NUMBER:** Número de nombramientos de la URL1 en la URL2.

Pero esos datos no se encuentran distribuidos en las tres columnas que cabría esperar (ver Figura 2.4), vienen en dos columnas con la siguiente distribución:

- La columna *url* contiene los datos de la *url1* y de la *url2*, pero con el siguiente formato: *<página web internacional> site: "<página web nacional>"*. Por lo que precisa de un tratamiento previo a su integración en la base de datos de *MySQL*, ya que hay que separar la información en dos columnas, dicho tratamiento se explica detalladamente en el apartado 3.1.2.
- La columna *hit\_number* contiene el número de nombramientos de la *url1* en la *url2* y no precisa tratamiento previo.

	A	B	C	D	E	F	G	H	I
1	url	hit_number							
2	adnams.co.uk site:"4kilos.com"	0							
3	12x75.com site:"umesufan3.com"	0							
4	14hands.com site:"umesufan3.com"	0							
5	50thparallel.com site:"umesufan3.com"	0							
6	agoodtimewithwine.com site:"umesufan3.com"	0							
7	abcfws.com site:"umesufan3.com"	0							
8	adampuchtawine.com site:"umesufan3.com"	0							
9	adnams.co.uk site:"umesufan3.com"	0							
10	alpinewines.co.uk site:"umesufan3.com"	0							
11	anconaswine.com site:"umesufan3.com"	0							
12	angelsgatewinery.com site:"umesufan3.com"	0							
13	anthonyrosewine.com site:"umesufan3.com"	0							
14	anticanapavalley.com site:"umesufan3.com"	0							
15	argylewinery.com site:"umesufan3.com"	0							
16	artisanvineyards.com site:"umesufan3.com"	0							
17	artisanwinedepot.com site:"umesufan3.com"	0							
18	direct.asda.com site:"umesufan3.com"	0							
19	aubonclimat.com site:"umesufan3.com"	0							
20	northwest-wine.com site:"umesufan3.com"	0							

Figura 2.4. Detalle del contenido del fichero de hits, en formato CSV.

### 2.4.2. Muestra 2: Impacto Web

Para la obtención de esta muestra se ha utilizado la herramienta *Majestic*<sup>11</sup>, que mide, entre otros factores, la visibilidad de las webs a través del número de enlaces que recibe. Se mide, durante los meses de Abril, Mayo, Junio y Julio de 2016, el número de enlaces entrantes totales que tienen las páginas web españolas, para determinar su visibilidad (cuántos enlaces entrantes se reciben al mes, desde cuántos sitios, qué

<sup>11</sup> Véase <https://es.Majestic.com/>.

autoridad tienen en la web, etc.), recogiendo los datos en 4 ficheros CSV, uno por cada mes.

#### 2.4.2.1. *Majestic*

*Majestic* es una herramienta de posicionamiento web (SEO, del inglés *Search Engine Optimization*) que inspecciona páginas web y mapea internet con el fin de ofrecer la mayor base de datos comercial de enlaces del mundo. Permite detectar y analizar los enlaces entrantes a cualquier web. Además, pone al servicio de sus clientes todos los medios disponibles para ofrecer un servicio SEO muy útil. Esta herramienta es utilizada por expertos SEO y especialistas en medios online con el fin de generar *linkbuilding* (creación de enlaces), administración de reputación, desarrollo de tráfico en internet y análisis de la competencia, entre otras cosas (Quiles, 2013).

Dado que la información de los enlaces también constituye un componente de la clasificación en buscadores, el conocimiento del perfil de sus propios enlaces, así como de los sitios web de la competencia, puede facilitar un estudio racional del posicionamiento en buscadores. *Majestic* revisa constantemente páginas web y visita alrededor de mil millones de URL al día (Majestic-12 Ltd, 2016).

Sin embargo, no es una herramienta gratuita, es una API comercial y como tal se ofrece con varios paquetes de servicios a distintos precios. Para la extracción de datos Web, llevada a cabo en el proyecto de investigación de *Trademetrics*, se contrató un servicio intermedio, en el cual se puede hacer uso de una aplicación en la que se ingresa la URL y devuelve, en una tabla en HTML, todos los indicadores que se tengan contratados.

*Majestic* dispone de varias utilidades, entre ellas destacan (Romero & Díaz, 2015):

- *Site Explorer*: Permite explorar un dominio o una URL con todo detalle. Es posiblemente la herramienta más completa y más interesante, ya que aporta valiosa información sobre los subdominios, enlaces, redireccionamientos, imágenes, etc.
- *Bulk Backlink Checker*: Pretende ahorrar tiempo cuando sólo se quiere conocer el número de enlaces externos entrantes para varios dominios.
- *Neighbourhood Checker*: Analiza los servidores web que alojan dominios en una única dirección IP.
- *Clique Hunter*: Muestra los dominios que contienen enlaces entrantes a las webs de la competencia.
- *Comparator*: Compara hasta 5 dominios distintos.
- *Majestic Million*: Un *ranking* o clasificación de 1.000.000 de páginas, ordenadas por índices de *Majestic* y actualizado con frecuencia.

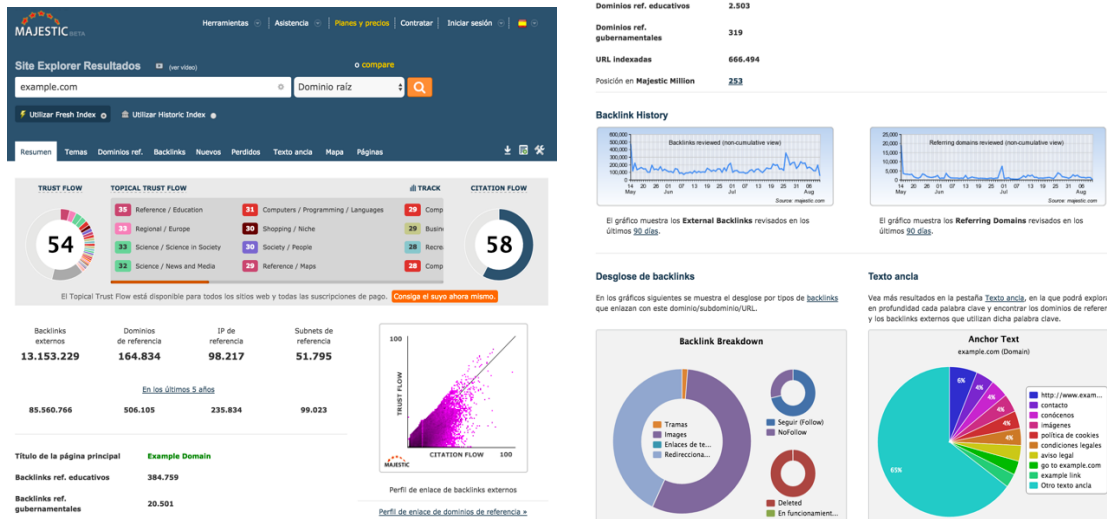


Figura 2.5. Ejemplo de visualización de la herramienta Site Explorer de Majestic.

Fuente: <https://es.majestic.com/reports/site-explorer?q=example.com&IndexDataSource=F>

### 2.4.2.2. Detalle

Los ficheros que componen la muestra 2 son cuatro y están en formato CSV, provienen de extracciones de datos Web hechas con *Majestic*. Estos ficheros tienen un contenido distinto a los obtenidos con *web scraping*, poseen 44 columnas (correspondientes a los parámetros que devuelve *Majestic* por defecto) y 2744 filas (correspondientes a las empresas españolas de vino sobre las que se ha efectuado el estudio). De los 44 parámetros facilitados por *Majestic*, tan solo se van a explicar 21, ya que son los considerados más relevantes para el proyecto de *Trademetrics*, dado que reflejan la actividad en la Web, el resto son parámetros técnicos internos, que no tienen interés para el caso de estudio y por lo tanto se obvian. Estos indicadores más relevantes, se corresponden a los siguientes términos del glosario de *Majestic* (Majestic-12 Ltd, 2016), en orden alfabético:

- **citationFlow:** (Tipo entero) Flujo de menciones, es un indicador entre 0 y 100 (en escala logarítmica como el *PageRank*) que sirve para medir la cantidad de enlaces entrantes. *Citation Flow* (CF) se utiliza junto con *Trust Flow* (TF) o flujo de confianza, juntos forman el algoritmo de *Flow Metrics* (indicadores de flujo) de *Majestic*. CF predice el grado de influencia de una URL en función de la cantidad de sitios enlazan con ella.
- **extBacklinks:** (Tipo Entero) Número de enlaces externos (*External Back Links*) que se están recibiendo en el sitio analizado. Un *backlink* o enlace de referencia es un enlace entrante que proviene de un sitio web o dominio de referencia distinto. En el sector de los servicios de posicionamiento en buscadores, a un *backlink* se le suele llamar enlace entrante.
- **extBacklinksEdu:** (Tipo Entero) Número de enlaces externos que vienen de dominios *.edu* (dominio para universidades). Se consideran enlaces externos de sitios de confianza.



- **extBacklinksGov:** (Tipo Entero) Número de enlaces externos que vienen de dominios *.gov* (entidades gubernamentales). Se consideran enlaces externos de sitios de confianza.
- **outDomainExternal:** (Tipo Entero) Número de dominios externos a los que se dirige al menos un enlace desde el dominio que se analiza.
- **outLinksExternal:** (Tipo Entero) Número de enlaces externos a los que se dirige al menos un enlace desde el sitio que se analiza.
- **outLinksInternal:** (Tipo Entero) Número de enlaces internos a los que se dirige al menos un enlace desde el sitio que se analiza (son como autoenlaces).
- **redirectFlag:** (Tipo Booleano) Es una marca o *flag* que indica si la URL de origen redireccionará la ubicación de destino de una página a otra.
- **redirectTo:** (Tipo Texto) Página web a la que redirecciona el sitio que se analiza.
- **refDomains:** (Tipo Entero) Dominio único de referencia, número total de sitios desde donde hay al menos un enlace entrante al sitio que se analiza, es decir, es un sitio web que tiene un *backlink* que apunta a una página o a un enlace con el sitio que se está analizando. Aquí no se cuentan el número de enlaces, sino los sitios desde donde viene algún enlace.
- **refDomainsEdu:** (Tipo Entero) Número de sitios desde donde hay al menos un enlace entrante al sitio que se está analizando y que sean de dominio *.edu*.
- **refDomainsGov:** (Tipo Entero) Número de sitios desde donde hay al menos un enlace entrante al sitio que se está analizando y que sean de dominio *.gov*.
- **status:** (Tipo Texto) *Found/NotFound* según el sitio web haya sido encontrado o no en las páginas webs indizadas por *Majestic*.
- **topicalTrustFlow\_topic:** (Tipo Carácter) El *Topical Trust Flow* (TTF) o flujo de confianza temático es una función de *Majestic* que sirve para categorizar los sitios web. *Majestic* ha categorizado la *World Wide Web* prácticamente en su totalidad, en casi 1.000 categorías o *topics* distintos, de modo que los usuarios puedan ver en qué sector tiene más influencia el sitio web. Gracias al TTF los usuarios pueden encontrar qué influye en determinadas categorías y determinar fácilmente si es necesario eliminar enlaces. El TTF facilita una serie de números, en una escala logarítmica de registro del 0 al 100 para cada categoría. El número indica la influencia relativa de una página web, subdominio o dominio raíz en un tema o categoría determinados. *Majestic* devuelve los tres *topics* o categorías que han obtenido mayor puntuación, etiquetándolos como *topicalTrustFlow\_topic\_0, 1 y 2*.
- **topicalTrustFlow\_value:** (Tipo Entero) Puntuación que asigna *Majestic* a la categoría correspondiente. Devuelve las tres puntuación de mayor valor, según su función de categorización, y las etiqueta como *topicalTrustFlow\_value\_0, 1 y 2*.
- **trustFlow:** (Tipo Entero) *Trust Flow* o Flujo de confianza, es un indicador de calidad establecido por *Majestic*, en una escala logarítmica del 0 al 100. Se enfoca más en la calidad de los enlaces, en vez de en la cantidad como el *citationFlow*. *Majestic* recopiló multitud de conjuntos de sitios de procedencia fiable mediante una revisión manual de la Web. Este proceso constituye la base de este indicador. Los sitios a los que enlacen sitios de procedencia fiable

obtendrán puntuaciones más altas, mientras que los sitios que tengan enlaces cuestionables obtendrán puntuaciones mucho más bajas.

- **web:** (Tipo Texto) Dirección de la página web que se está analizando.

Figura 2.6. Detalle del fichero vino160404 en formato CSV, Abril de 2016.

### 2.4.2.3. Indicadores métricos seleccionados

A continuación, se pasa a explicar con mayor detalle, el grupo de indicadores métricos de *Majestic*, que se han seleccionado para hacer la prueba con el entorno de R, cuya metodología se describe en el apartado 2.6.3 (página 36) y los resultados en el apartado 3.2 (página 55).

#### A) Citation Flow y Trust Flow

Como se ha introducido anteriormente, la métrica *Citation Flow* (CF) se centra en la **cantidad** de enlaces entrantes. La escala es logarítmica y va de 0 a 100. A algunos enlaces se les da más puntuación que a otros, dependiendo de si provienen o no de sitios que *Majestic* considera de confianza. El CF intenta predecir la "fortaleza" (Figura 2.7) de una página web frente al resto y lo hace analizando la cantidad de enlaces que apuntan hacia esa página, tanto externos como internos. Hay que recalcar que esta métrica no analiza la calidad de los enlaces, simplemente su cantidad, por esta razón no suele usarse de forma independiente, sino con otro indicador que ayuda a comprender si los enlaces son de "calidad", este indicador es el *Trust Flow* (TF), también de escala logarítmica de 0 a 100 (Marcilla, 2016).

Con el indicador TF, *Majestic* trata de cuantificar de alguna manera la **calidad** de un enlace, es decir, si proviene de una página web que no suele enlazar a páginas de dudosa reputación, si sus contenidos son de calidad, si a su vez es enlazada por otras páginas web consideradas de calidad, etc. Para construir el TF, *Majestic* revisa manualmente una enorme cantidad de páginas, que se supone son autoridades reconocidas y cuyos enlaces gozan de una buena reputación (Universidades, organismos gubernamentales, etc.).





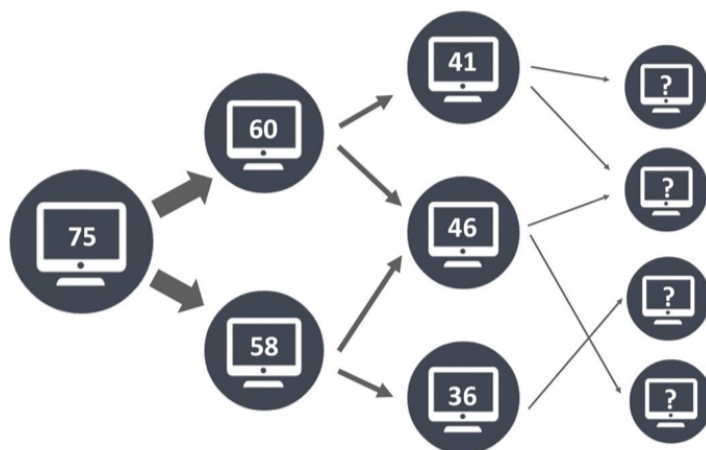


Figura 2.7. La "fuerza" de cada página se transmite a las demás mediante los enlaces.  
Fuente: <http://ninjaseo.es/metricas-seo/>

Un enlace será más confiable (tendrá mayor TF) cuánto más cerca esté de una fuente de confianza reconocida, es decir, cuantos menos enlaces se tengan que recorrer para llegar de ese sitio de confianza al de destino. Según esto, es más difícil tener un buen TF que un buen CF.

### B) External Back Links

Este indicador (EBL) mide la cantidad de enlaces entrantes que está recibiendo la página web que se está analizando. Dentro del EBL se encuentran otros dos indicadores, el *External Back Links Edu* (EBLE) que mide la cantidad de enlaces entrantes provenientes de instituciones de educación superior (Universidades) y el *External Back Links Gov* (EBLG) que contiene la misma medida pero sobre instituciones gubernamentales. Estos dos dominios son considerados de nivel superior, por lo que recibir un enlace de sitios con dominios *.edu* o *.gov*, significa que la página web está siendo enlazada por sitios de alta calidad y confiabilidad, lo que da mayor relevancia a dicha página (Marcilla, 2016).

### C) Topical Trust Flow

La mejor forma de juzgar cualquier enlace o página es utilizar varias métricas que se complementen y el mejor complemento para el ratio CF/TF es el *Topical Trust Flow* (TTF) de *Majestic*. Esta métrica ayuda a determinar la relevancia y autoridad de una URL dentro de su nicho concreto, gracias a que *Majestic* ha determinado su tema al haber categorizado prácticamente toda la Web en casi 1.000 categorías distintas o *topics* (Marcilla, 2016).

Como ya se ha comentado anteriormente, una URL tendrá un TF alto si tiene enlaces que provienen de otras webs confiables también con un alto TF. Pero el *Trust Flow* temático añade una nueva variable al TF normal: la relevancia temática. Si una URL tiene un alto TTF sobre cierta categoría, esto significa que recibe muchos enlaces desde páginas que tratan este tema, es decir, recibe enlaces relevantes para conseguir posicionarse por dicha temática, por lo que se puede deducir que es más fácil posicionarse en un tema determinado, cuando se tienen enlaces que provienen de páginas temáticamente relacionadas con la propia página.

## 2.5. Creación de un almacén para los datos

Los almacenes de datos son bases de datos, que se usan principalmente para el posterior análisis de los mismos y extracción de conclusiones que desarrollen una inteligencia de negocio. Funcionan como un depósito de datos históricos de distintas fuentes, que luego se convierten en información de valor para las empresas, ya que permiten, entre otras cosas, establecer comparaciones periódicas, obtener tendencias de comportamiento de páginas web, etc. Contar con un almacén de datos permite también, que los datos se estructuren de tal forma que se puedan hacer consultas analíticas y estadísticas más complejas (Padrón Torres, 2006).

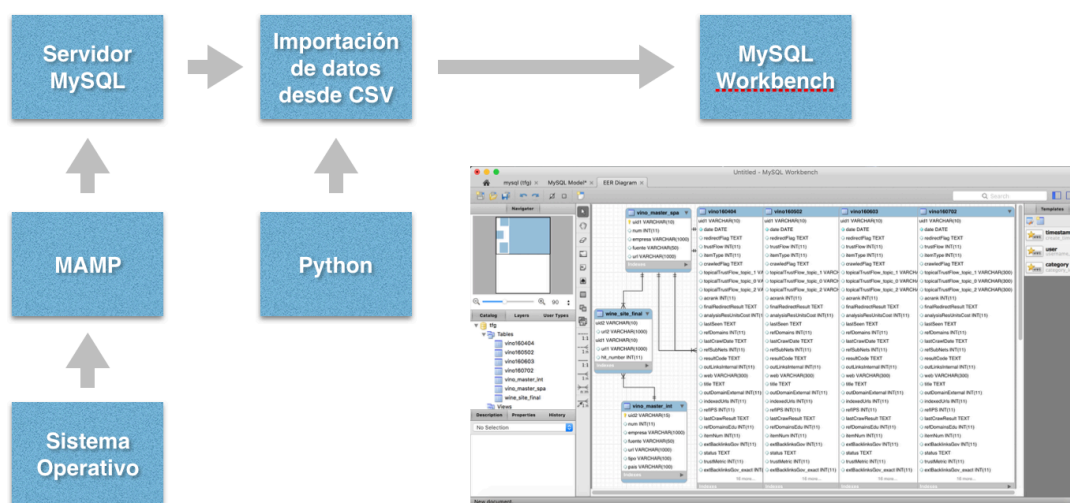


Figura 2.8. Esquema gráfico del proceso de creación de la base de datos.

Como almacén para los datos extraídos de la Web, se hace uso de la base de datos y servidor de *MySQL*, manejada con la interfaz gráfica de *MySQL Workbench*. Para iniciar el servidor se utiliza la aplicación *MAMP*. Como lenguaje de programación para implementar los *scripts*, que se precisan para la automatización del proceso de importación de los datos, desde los ficheros CSV a la base de datos *MySQL*, se emplea *Python*. Esta parte del trabajo se realizó durante el segundo cuatrimestre de 2016.

### 2.5.1. MySQL

*MySQL* es un sistema de gestión de bases de datos relacional (RDBMS), multihilo y multiusuario, con más de seis millones de instalaciones, basado en un lenguaje de consulta estructurado (SQL, del inglés *Structured Query Language*) (Santillán, Ginestà, & Mora, 2007). Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. *MySQL*, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información. *MySQL* fue escrito en *C* y *C++* y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como *PHP*, *Perl*, *Java* y *Python* y su integración en distintos sistemas operativos.

También es muy destacable, la condición de *open source* (código abierto, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente su desarrollo y continuas actualizaciones y hace de *MySQL* una de las herramientas más utilizadas por los programadores orientados a Internet.

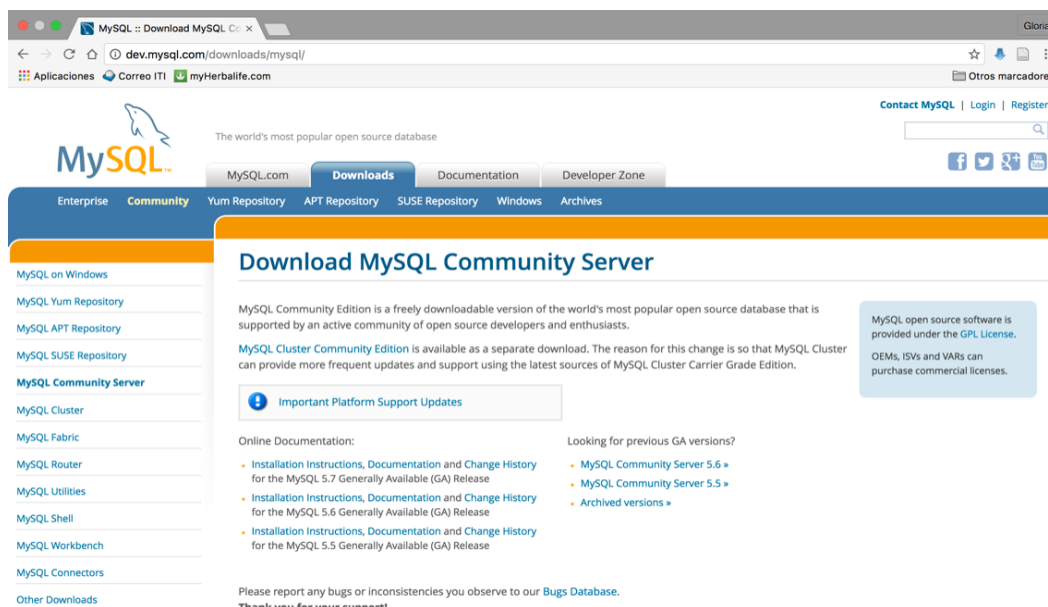


Figura 2.9. Sitio web de descarga del servidor MySQL.  
<http://dev.mysql.com/downloads/mysql/>

*MySQL* no viene instalado por defecto en las máquinas Mac OS X, así que se procede a su descarga de la página web referenciada en la Figura 2.9, seleccionando para ello la descarga del paquete "Mac OS X 10.11 (x86, 64-bit), DMG Archive". Se ejecuta y se procede con su instalación, siguiendo unos sencillos pasos que va indicando el asistente (Figura 2.10).

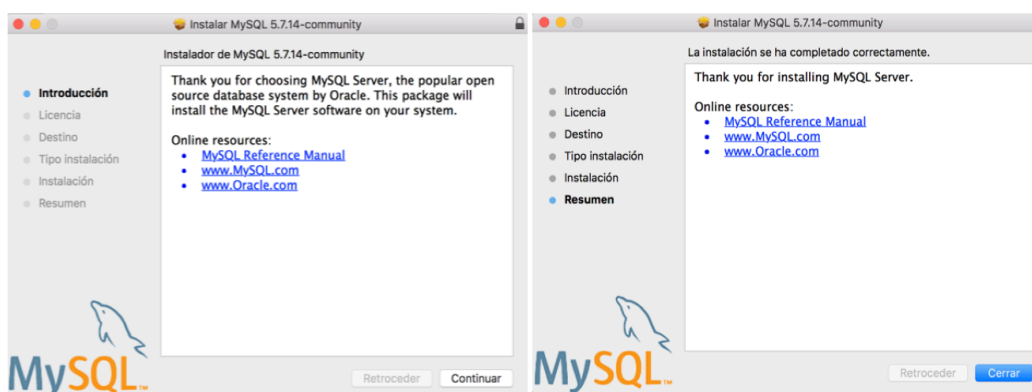


Figura 2.10. Proceso de instalación de MySQL en una máquina Mac OS X.

Hay que tener en cuenta que, dado que *MySQL* es una base de datos con arquitectura cliente-servidor, necesita un servidor que escuche en el puerto 3306 TCP (*Transmission Control Protocol*) por defecto, para que atienda las consultas y un cliente que permita conectar con el servidor.

Como servidor se ha utilizado un servidor local virtualizado de *MySQL* y para su implantación se ha usado la aplicación *MAMP*<sup>12</sup>. Como cliente se ha utilizado el cliente de consola de la propia *MySQL*.

## 2.5.2. MAMP

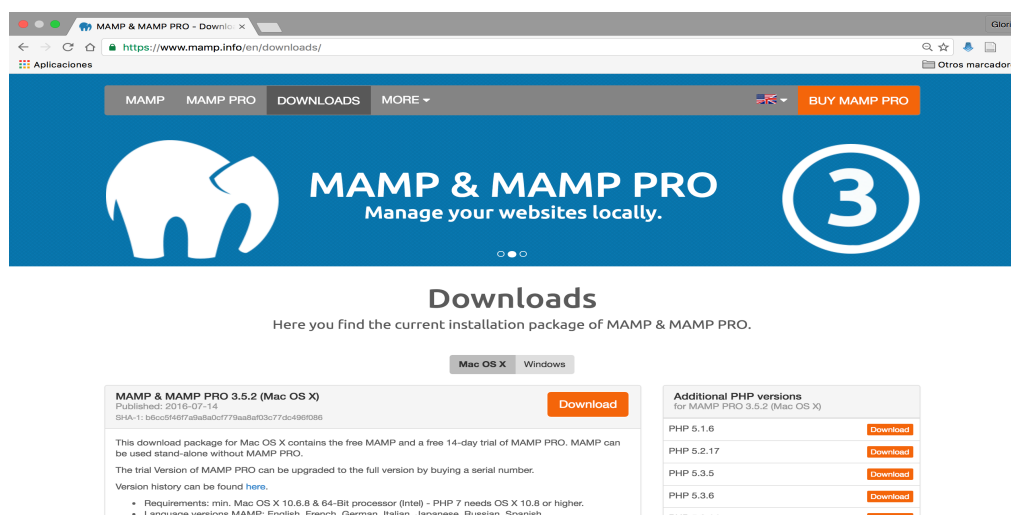


Figura 2.11. Sitio web de descarga de la aplicación MAMP.  
<https://www.mamp.info/en/downloads/>

El término *MAMP* (*Mac-Apache-MySQL- PHP/Python/PERL*) hace referencia al sistema creado por la conjunción de esas aplicaciones libres (de código abierto), con el sistema operativo *Macintosh*. Este grupo de aplicaciones generalmente es usado para crear servidores web (ALEGSA, 2016).

*MAMP* provee a los desarrolladores con los cuatro elementos necesarios para un servidor web: un sistema operativo (*Macintosh*), un manejador de base de datos (*MySQL*), un software para servidor web (*Apache* y *MySQL*) y un software de programación *script* web (*PHP*, *Python* o *PERL*).

Dispone de dos productos, uno de ellos es una solución gratuita *MAMP* y el otro es la solución profesional *MAMP PRO*, con muchos más servicios y utilidades y por supuesto de pago. Para el desarrollo del presente trabajo se usa la aplicación gratuita, que sirve para iniciar el servidor *MySQL* en local de una forma rápida y sencilla. Se descarga la aplicación de la página referenciada en la Figura 2.11.

Para instalarla se ejecuta el archivo de extensión *.pkg* y se siguen los pasos que van guiando a través de las ventanas consecutivas.

<sup>12</sup> Véase <https://www.mamp.info/en/>.

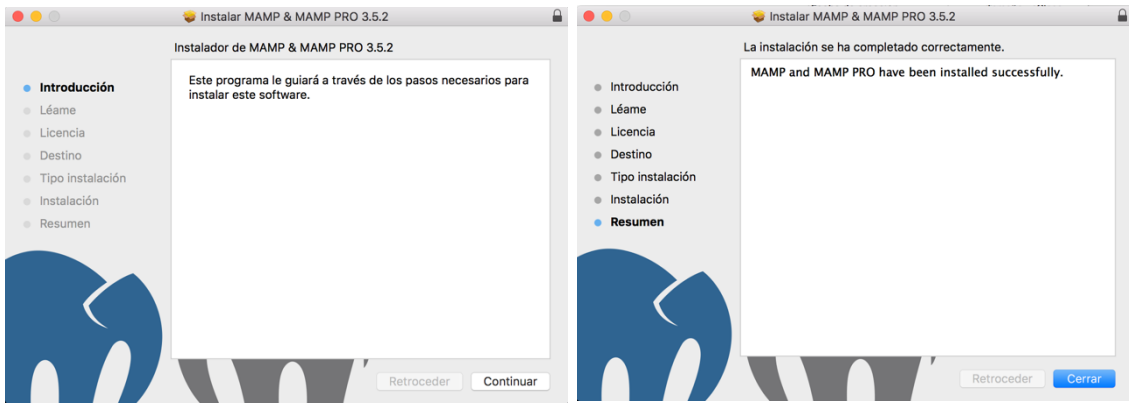


Figura 2.12. Proceso de instalación de la aplicación MAMP en una máquina Mac OS X.

Después de su instalación, dentro de la opción "Preferencias", se va a la etiqueta "Puertos" y se configuran los puertos según se indica en la imagen derecha de la Figura 2.13, dejando el resto de opciones de todos los apartados tal y como vienen por defecto, pues lo único que interesa es la configuración del puerto de conexión del servidor *MySQL*.

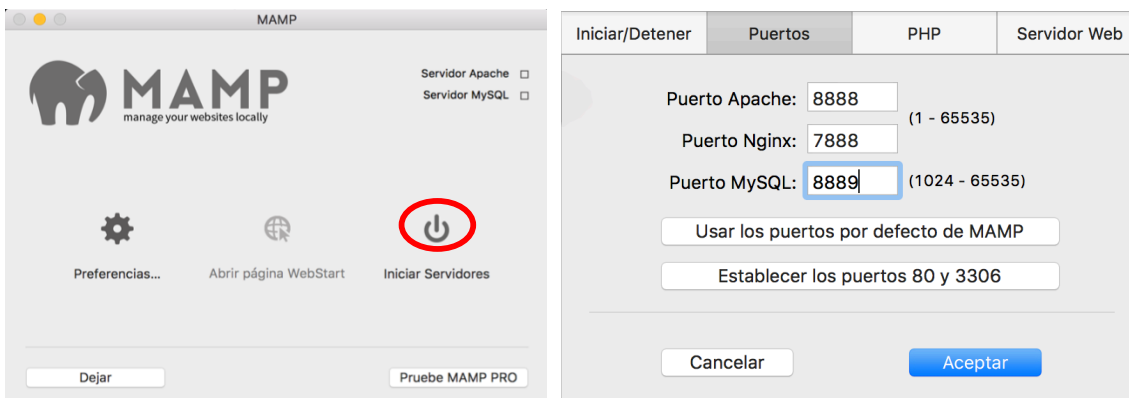


Figura 2.13. Interfaz gráfica de la aplicación MAMP y detalle de la configuración de puertos.

Se inicia el servidor *MySQL* accionando el interruptor de "Iniciar Servidores", que se puede apreciar en la imagen izquierda de la Figura 2.13. Se necesita también una interfaz para manejar la base de datos de forma amigable, para ello se recurre a *MySQL Workbench*.

### 2.5.3. MySQL Workbench

*MySQL Workbench* es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración, diseño, creación y mantenimiento para el sistema de base de datos *MySQL* (MySQL, 2016). También cabe destacar, que igual que *MySQL*, es una herramienta *open source* y por lo tanto gratuita.

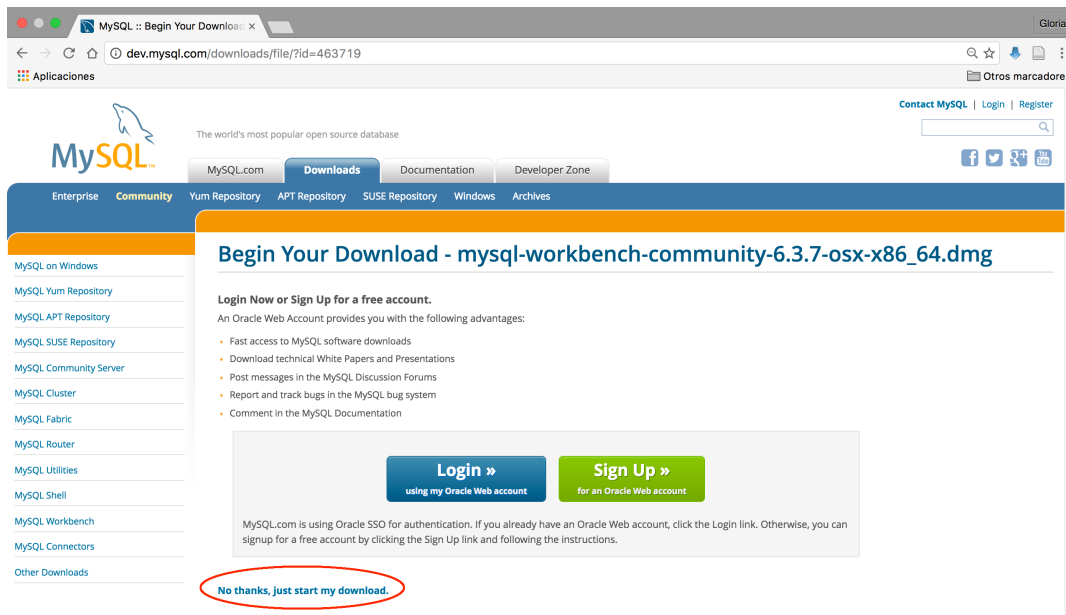


Figura 2.14. Sitio web de descarga de la aplicación MySQL Workbench.  
<https://www.mysql.com/products/workbench/>

Para instalarla se ejecuta el archivo `.dmg` que se descarga de la página web de referencia en la Figura 2.14. En la ventana que se abre a continuación, se arrastra el icono de *MySQL Workbench* al icono "Aplicaciones", según la Figura 2.15. y con eso ya está instalado. Ahora ya se puede iniciar *MySQL Workbench* desde la carpeta "Aplicaciones".



Figura 2.15. Detalle de la instalación de MySQL Workbench en una máquina Mac OS X.

Cuando se inicia la aplicación se ve la pantalla de la Figura 2.16. Seguidamente, se crea una nueva conexión haciendo clic sobre el icono del signo más y se rellenan los campos de la ventana que se abre, tal y como muestra la imagen de la Figura 2.17.

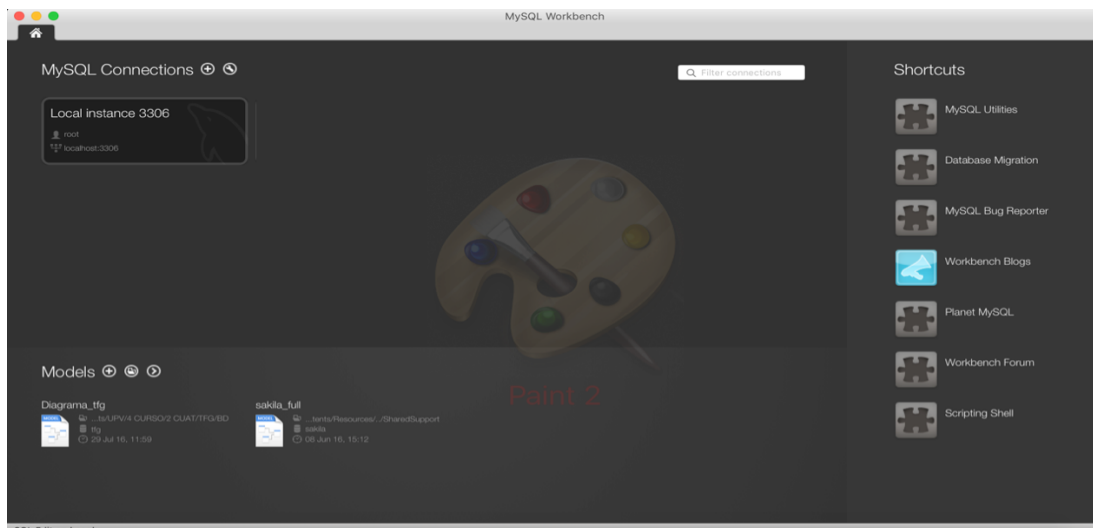


Figura 2.16. Pantalla inicial de la aplicación MySQL Workbench.

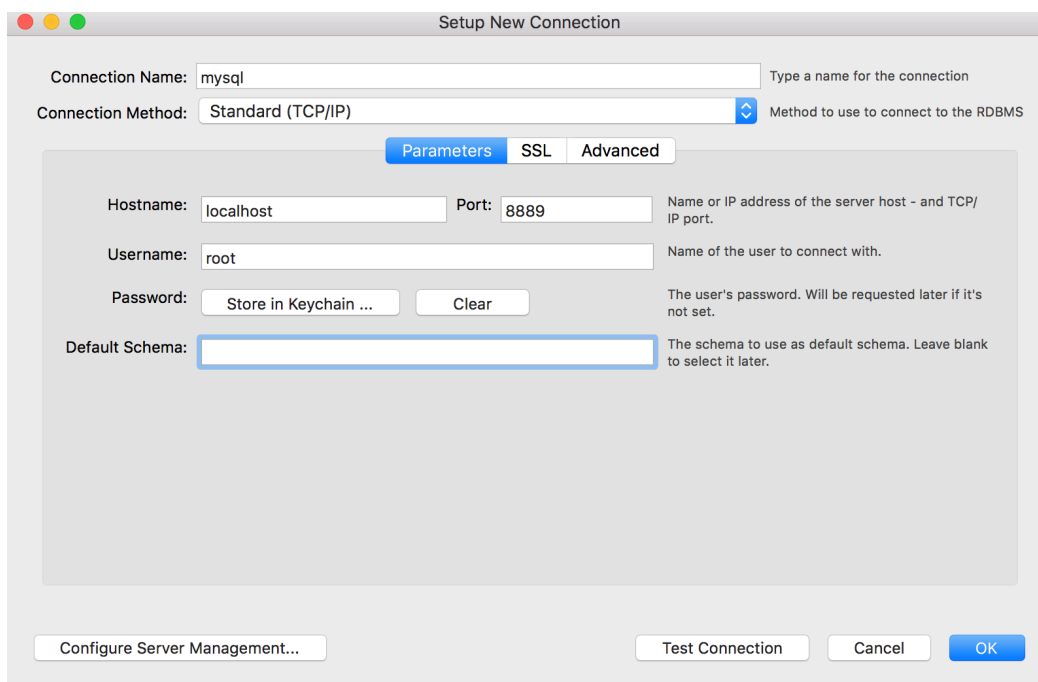


Figura 2.17. Configuración de la nueva conexión.

Después de darle a "OK", aparece en la pantalla inicial de *MySQL Workbench*, otra conexión llamada "mysql" junto al enlace de la conexión "Local instance 3306". Al seleccionarla se hace la conexión al servidor *MySQL* virtualizado en local, en el puerto 8889 TCP y se abre la interfaz de trabajo de la base de datos (Figura 2.18).

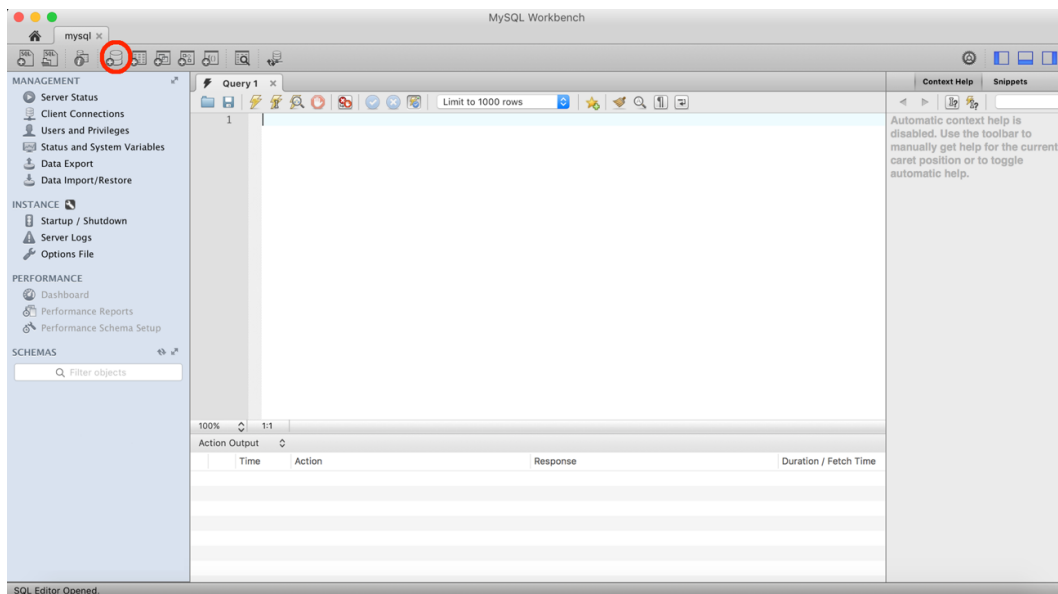


Figura 2.18. Interfaz gráfica de trabajo de MySQL Workbench.

Al hacer clic sobre la imagen rodeada por el círculo rojo en la figura anterior, se abre la ventana de creación de un nuevo esquema de base de datos, donde hay que rellenar el nombre de la base de datos ("tfg") y la codificación de los caracteres de las tablas, en nuestro caso se selecciona "*utf8 – default collation*". Quedando así preparada la base de datos para la importación de las tablas, desde los archivos CSV, a través de los *scripts* implementados en *Python*.

## 2.5.4. Python

*Python* es un lenguaje de programación de alto nivel e interpretado (orientado a la realización de *scripts*), cuya filosofía hace hincapié en una sintaxis que favorece un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y programación funcional. Usa tipado dinámico (no necesita declaración de tipos de variables), es multiplataforma y posee una licencia de código abierto (*Python Software Foundation License*<sup>13</sup>) (Duque, 2010).

*Python* destaca por su sencillez, legibilidad y precisión de sintaxis, ya que en pocas líneas permite programar algoritmos bastante complejos, cualidades altamente valoradas por los desarrolladores. Sin embargo, al ser un lenguaje interpretado es más lento en su ejecución frente a los lenguajes compilados, lo que no lo hace adecuado para la programación de bajo nivel o para aplicaciones en las que el rendimiento sea crítico. No obstante, se encuentra en el puesto número cuatro de la clasificación (*ranking*) de lenguajes más utilizados, según *RedMonk*<sup>14</sup>, por delante de lenguajes compilados como *C*, *C++* y *C#*. De todos modos, esto no es un problema importante para el trabajo en el campo de la analítica web, menos aún si se tiene en cuenta que va a ahorrar mucho tiempo de programación.

<sup>13</sup> Véase para más información <https://docs.python.org/3/license.html>.

<sup>14</sup> Véase para más información <http://redmonk.com/sogrady/2016/07/20/language-rankings-6-16/>.



También hay que decir, que cuenta con una comunidad muy activa, capaz de aportar tutoriales y respuestas a gran variedad de problemas y que dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de cadenas de caracteres, números, archivos, etc., además de disponer de infinidad de librerías que se pueden importar en los programas.

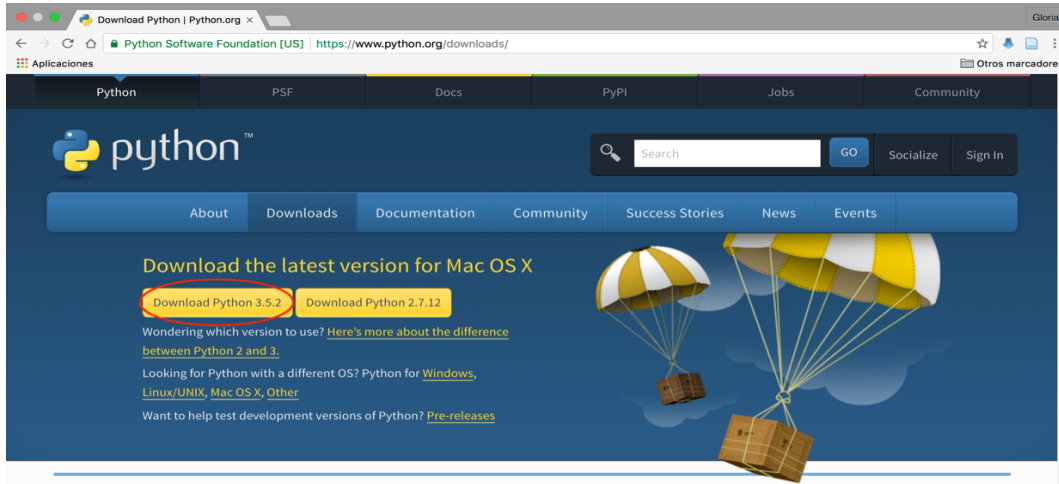


Figura 2.19. Sitio web para la descarga del paquete de instalación de Python 3.5.2.  
<https://www.python.org/downloads/>

Para instalar *Python* en una máquina Mac OS X, hay que descargar el fichero de extensión *.pkg* de la página web de referencia de la Figura 2.19, después se ejecuta el archivo descargado y se siguen los sencillos pasos que se van indicando, en las ventanas sucesivas del asistente.

En cuanto a las librerías que se van a necesitar en nuestros *scripts*, únicamente es preciso realizar una instalación previa de las siguientes:

- *mysql.connector*: Se descarga el fichero *.dmg* de la siguiente página web <https://dev.mysql.com/downloads/connector/python/>. Para instalarlo, se ejecuta y se siguen los pasos que va indicando el asistente.
- *OpenPyXL*: Para su instalación se ejecuta en el terminal la orden `pip install openpyxl`<sup>15</sup>.

Para el resto de librerías, basta tan sólo con hacer el *import* correspondiente.

## 2.6. Análisis de los datos

Para la extracción de los datos almacenados en la base de datos *MySQL*, así como para la posterior prueba de realización de análisis estadísticos sobre dichos datos, se ha seleccionado el lenguaje de programación *R* y su herramienta de interfaz gráfica *RStudio*. Esta parte del trabajo se realizó durante el segundo cuatrimestre de 2016.

---

<sup>15</sup> Véase para más información <http://openpyxl.readthedocs.io/en/default/>.

## 2.6.1. R

R es un sistema para análisis estadísticos y gráficos creado por Ross Ihaka y Robert Gentleman (Ihaka & Gentleman, 1996). R tiene una naturaleza doble de programa y lenguaje de programación orientado a objetos y es considerado como un dialecto del lenguaje S creado por los *Laboratorios AT&T Bell*. S está disponible como el programa *S-PLUS* comercializado por *Insightful*<sup>16</sup>.

R se distribuye gratuitamente bajo los términos de la *GNU General Public Licence*<sup>17</sup>; su desarrollo y distribución son llevados a cabo por varios estadísticos conocidos como el *Grupo Nuclear de Desarrollo de R*.

Los archivos necesarios para instalar R, ya sea desde las fuentes o binarios pre-compilados, se distribuyen desde el sitio de internet *Comprehensive R Archive Network* (CRAN)<sup>18</sup> junto con las instrucciones de instalación. Al entrar en la página web para la descarga de R, se selecciona la descarga correspondiente a la máquina Mac OS X (ver Figura 2.20) y posteriormente el paquete "R-3.3.1.pkg".

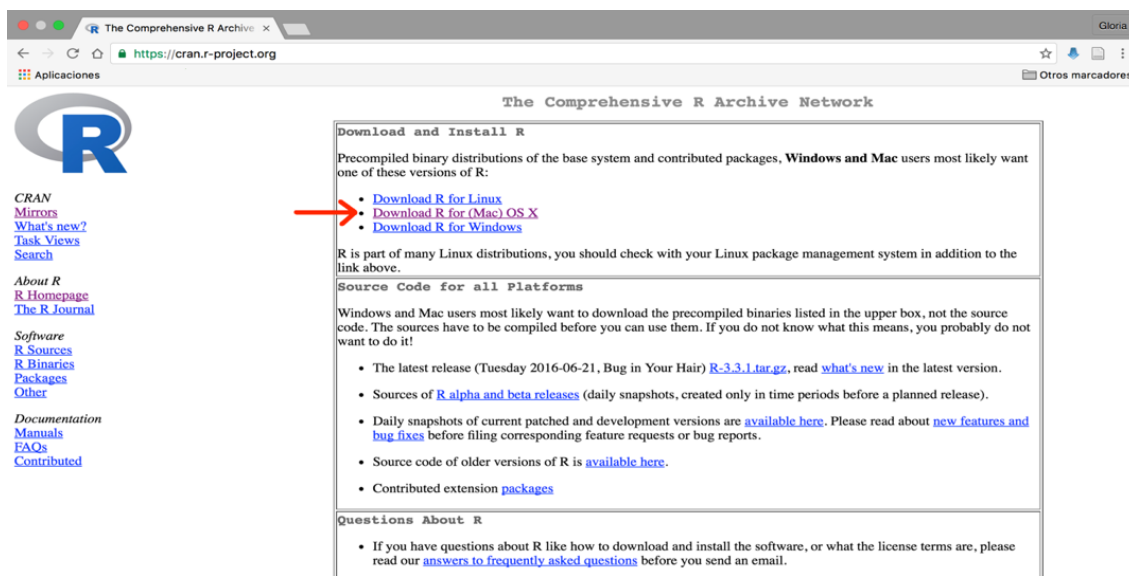


Figura 2.20. Sitio web para la descarga de R.  
<https://cran.r-project.org/>

Se ejecuta el archivo de extensión *.pkg* y se siguen los pasos a través de las ventanas de la instalación (Figura 2.21).

<sup>16</sup> Véase para más información <http://www.insightful.com/products/splus/default.html>.

<sup>17</sup> Véase para más información <http://www.gnu.org/>.

<sup>18</sup> <http://cran.r-project.org/>.

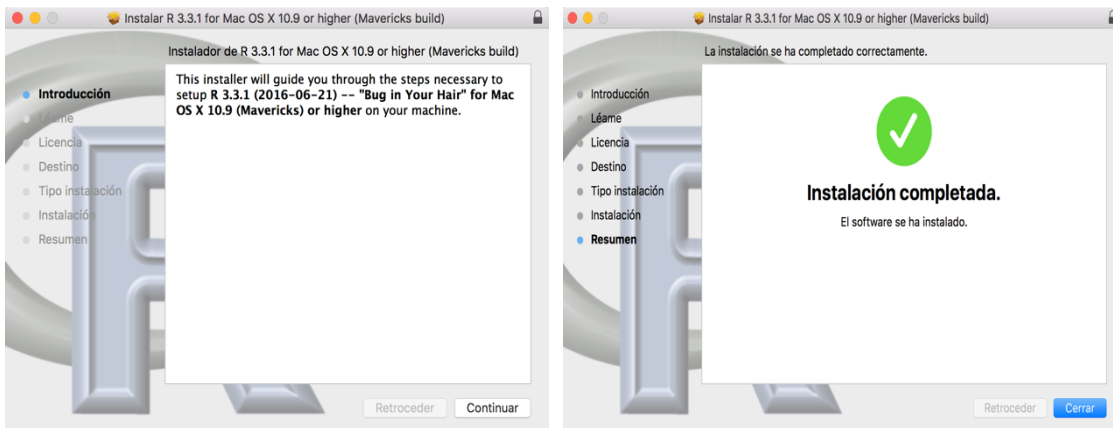


Figura 2.21. Detalle instalación de R en una máquina Mac OS X.

## 2.6.2. RStudio

R es un entorno de desarrollo integrado (IDE) para R. Es software libre con licencia GPLv3<sup>19</sup> y se puede ejecutar sobre distintas plataformas o incluso desde la web usando *RStudio Server*. Incluye una consola, editor de sintaxis que apoya la ejecución de código, así como herramientas para el trazado, la depuración y la gestión del espacio de trabajo.

A continuación, se procede a la descarga de *RStudio* de la página referenciada en la figura siguiente Figura 2.22.

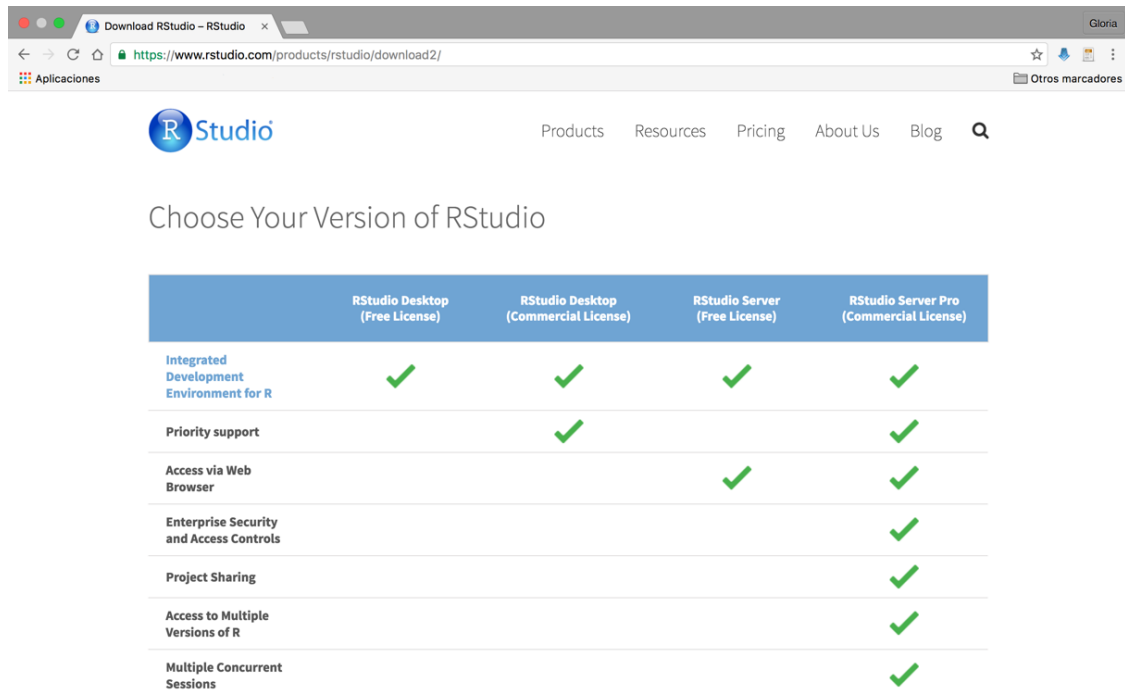


Figura 2.22. Sitio web para la descarga de RStudio.  
<https://www.rstudio.com/products/rstudio/download2/>

<sup>19</sup> Véase <https://www.gnu.org/licenses/quick-guide-gplv3.html>.

Se selecciona para ello la opción de *RStudio Desktop* y de entre los instaladores que se muestran, se escoge "RStudio 0.99.903 - Mac OS X 10.6+ (64-bit)", ver Figura 2.23.

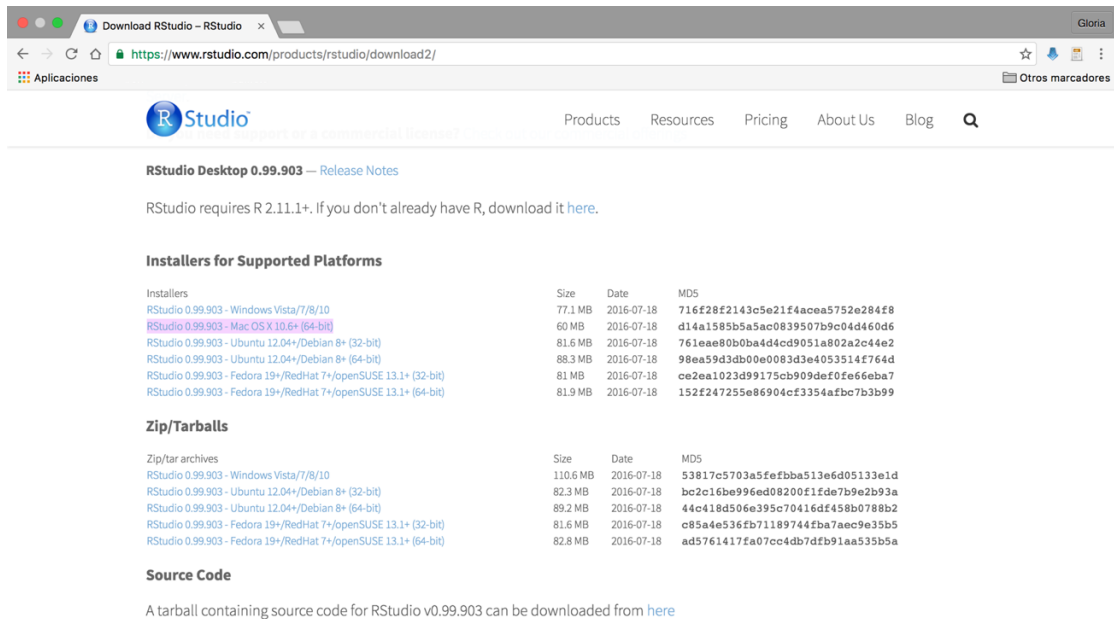


Figura 2.23. Detalle descarga del instalador de RStudio para Mac OS X.

Y se procede a su instalación ejecutando el archivo de extensión .dmg que se ha descargado. En la ventana que se abre a continuación, tan solo hay que desplazar el icono de *RStudio* al icono de "Aplicaciones" y ya estará instalada la herramienta. Al ejecutar *RStudio* se ve una pantalla similar a la de la Figura 2.24, donde se muestran las distintas áreas de trabajo.

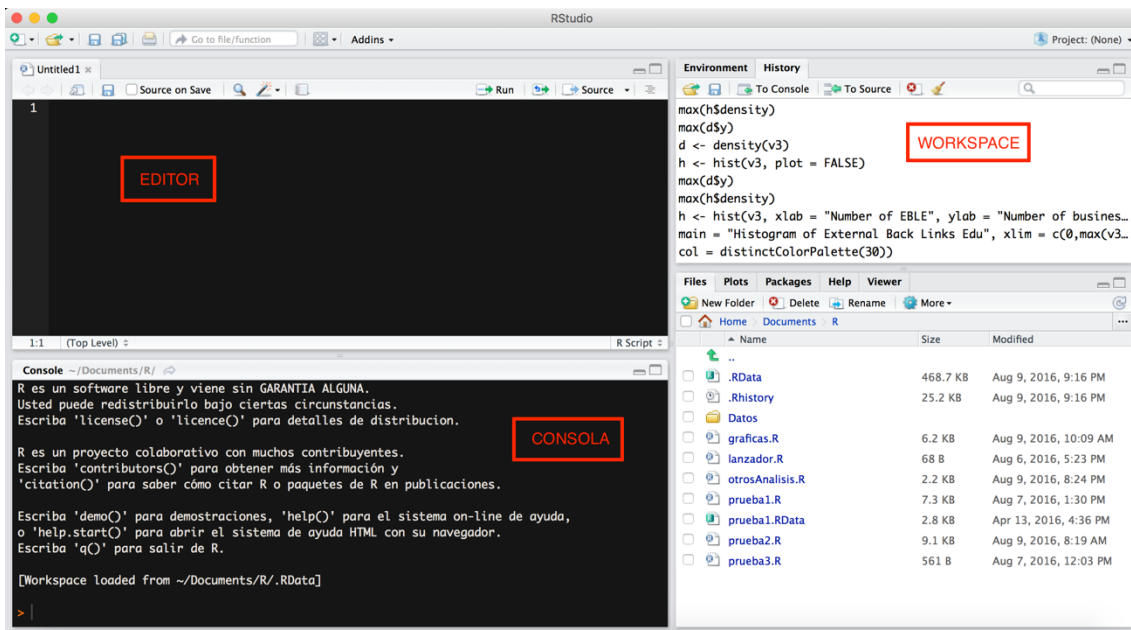


Figura 2.24. Interfaz de trabajo de RStudio para máquina Mac OS X.

Tan sólo queda ya por instalar las librerías que se van a utilizar en los *scripts*. Para ello, se inicia la aplicación de *RStudio* y en la consola se escriben y se ejecutan las siguientes instrucciones:

- `install.packages(DBI)`, cuando haya terminado la instalación se continúa con el resto de paquetes.
- `install.packages(RMySQL)`
- `install.packages(randomcoloR)`
- `install.packages(e1071)`

### 2.6.3. Prueba de análisis

Para probar el entorno de *R*, se hace uso de varios *scripts* (véase Anexo II) en el lenguaje de programación *R* que sirven para analizar y visualizar un pequeño grupo de indicadores de *Majestic*, los cuales se han considerado de mayor importancia frente al resto, en cuanto al punto de vista del análisis de la potencia o visibilidad del sitio web se refiere. Estos indicadores son: *Citation Flow*, *Trust Flow*, *External Back Links (Edu y Gov)* y *Topical Trust Flow*.

Antes de ejecutar los *scripts*, hay que iniciar el servidor de *MySQL* con la aplicación *MAMP*. Estos *scripts* se diseñan de forma que, en líneas generales, cargan las librerías descritas en el apartado 2.3 (página 14), realizan la conexión con la base de datos de *MySQL*, envían una consulta en lenguaje *SQL*, recogen el resultado de dicha consulta en forma de tablas (*data frame*) y, haciendo uso de las librerías, analizan los datos y trazan varias gráficas, que se visualizan en pantalla o se guardan en ficheros PDF.

Con el motivo de unificar criterios, se ha seleccionado como caso de prueba la tabla correspondiente al fichero de *Majestic* del mes de Abril (*vino160404.csv*).

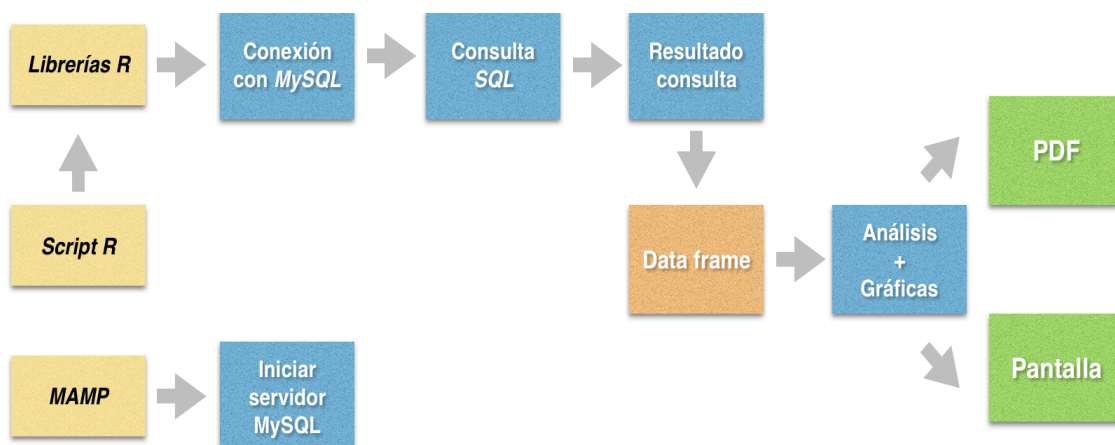


Figura 2.25. Esquema proceso de prueba del análisis con *R*.

**NOTA:** Se recuerda que la extracción de datos mediante *web scraping* (Muestra 1) se realizó durante los meses de Febrero, Marzo y Abril de 2016, recogiendo todos los datos en un único fichero (fichero de *hits*). La extracción de datos con *Majestic* (Muestra 2) se realizó durante los meses de Abril, Mayo, Junio y Julio de 2016, recogiendo los datos en 4 ficheros CSV, uno por cada mes. El tratamiento de los ficheros con los datos recogidos, la creación de la base de datos y las pruebas de análisis desde *R*, se realizaron durante el segundo cuatrimestre de 2016.

## 3. Resultados

---

En este capítulo, se va a analizar más detenidamente la solución que se ha adoptado para lograr el objetivo de automatizar la captura y almacenamiento estructurado de los datos incluidos en cada una de las dos muestras de datos detalladas en el capítulo anterior. Se describe detalladamente todo el proceso que se ha llevado a cabo, haciendo uso de las metodologías descritas en el capítulo 2, para conseguir el cumplimiento de los objetivos descritos en el apartado 1.3.

### 3.1. Creación de un almacén para los datos

En este apartado se describen detalladamente los pasos seguidos en el tratamiento previo y estructuración de los ficheros que nos fueron suministrados, así como el proceso que se lleva a cabo para crear la base de datos que los almacena y el procedimiento que se sigue para su importación a dicha base de datos. Todos los *scripts* implementados, tanto para el tratamiento de los ficheros como para su importación a la base de datos, se realizan en el lenguaje de programación *Python*, como se ha comentado en el capítulo de metodología.

#### 3.1.1. Ficheros maestros

Como primera tarea, se decide diseñar unas claves UID (*Unique Identifier*) que identifiquen de manera inequívoca y única a cada una de las empresas, relacionándolas, en el caso de las españolas, con cada una de sus fuentes y, además en el caso de las internacionales, con el país de origen y con el tipo de página web (tienda virtual o blog).

En un primer momento, se piensa en un diseño para las empresas españolas, que incluya 4 dígitos con el número de la empresa + las letras "SP" + 1 letra para la fuente, pero se descarta, dado que va a haber búsquedas, por ejemplo, en las que se quiera encontrar sólo las empresas que sean españolas o sólo las que sean de Canadá, con lo que la búsqueda sería más rápida y efectiva si los primeros caracteres de la clave identificaran el país.

Finalmente se decide adoptar el siguiente diseño de claves UID:

- Para las empresas españolas: "SPA" + 4 dígitos que identifican el número de orden de la empresa + 1 letra para identificar la fuente (*Facebook*: F, *Instagram*: I, *LinkedIn*: L, *Site*: S, *Twitter*: T, *YouTube*: Y). Total 8 caracteres para el campo de clave UID, con lo que salen 7.332 identificadores para las empresas españolas.  
Ejemplo: La clave UID para la empresa "10 sentits", que tiene el número de orden 2, con la fuente de *Twitter*, quedaría así: SPA0002T.
- Para las empresas internacionales: 2 letras para indicar el país de procedencia (*Canada*: CA, *United States*: US, *United Kingdom*: UK) + 1 letra para identificar el tipo de *site* (Blog: B, Tienda Virtual: V) + 4 dígitos que identifican el número de orden de la empresa + 1 letra para la

fuente (las mismas que las españolas y además *Google+*: G, *Pinterest*: P). En total también son 8 caracteres para el este campo de clave UID, con lo que salen 1.046 identificadores para las empresas internacionales. Ejemplo: La clave UID para la empresa "Adam Puchta Winery", que tiene asignado el número 6, con la fuente de *Google+*, cuyo *site* es una tienda virtual y que es originaria de Estados Unidos, quedaría así: UEB0006G.

Se añade una columna en cada fichero master, que contendrá las claves UID, editando para ello el fichero XLSX desde EXCEL. Para introducir las claves diseñadas se utiliza la herramienta de diseño de funciones de EXCEL, introduciendo la fórmula en la primera celda y copiándola en el resto de filas inferiores. La fórmula diseñada para las claves UID de las empresas españolas es la siguiente:

```
= "SPA" & SI(LARGO($B2)=1; "000"; SI(LARGO($B2)=2; "00"; SI(LARGO($B2)=3; "0"; ""))) & $B2 & SI($D2="FACEBOOK"; "F"; SI($D2="INSTAGRAM"; "I"; SI($D2="LINKEDIN"; "L"; SI($D2="SITE"; "S"; SI($D2="TWITTER"; "T"; "Y")))))
```

	A	B	C	D	E	F	G	H
1	uid1	num	empresa	fuente	url1			
2	SPA0001I	1 1 + 1 = 3		INSTAGRAM	instagram.com/1mes1fan3			
3	SPA0001S	1 1 + 1 = 3		SITE	umesufan3.com			
4	SPA0001T	1 1 + 1 = 3		TWITTER	twitter.com/UmesUfan3			
5	SPA0002I	2 10 sentits		INSTAGRAM	instagram.com/10sentits			
6	SPA0002S	2 10 sentits		SITE	10sentits.cat			
7	SPA0002T	2 10 sentits		TWITTER	twitter.com/10Sentits			
8	SPA0002Y	2 10 sentits		YOUTUBE	youtube.com/user/10sentits			
9	SPA0003S	3 15 albas		SITE	15albas.es			
10	SPA0004F	4 2amigos		FACEBOOK	facebook.com/2amigos-500500696793680			
11	SPA0004S	4 2amigos		SITE	2amigos.com			
12	SPA0004T	4 2amigos		TWITTER	twitter.com/2amigoswine			
13	SPA0004Y	4 2amigos		YOUTUBE	youtube.com/user/2amigoscom			
14	SPA0005F	5 3 ases		FACEBOOK	facebook.com/TRES-ASES-VINO-147947088568166			
15	SPA0005I	5 3 ases		INSTAGRAM	instagram.com/bodega_3ases			
16	SPA0005S	5 3 ases		SITE	3asesvino.com			
17	SPA0005T	5 3 ases		TWITTER	twitter.com/3asesvino			
18	SPA0005Y	5 3 ases		YOUTUBE	youtube.com/user/3AsesVino			
19	SPA0006F	6 4 Kilos vinicola		FACEBOOK	facebook.com/4KilosVinicola			
20	SPA0006L	6 4 Kilos vinicola		LINKEDIN	linkedin.com/in/4kilos-vinicola-b1875357			
21	SPA0006S	6 4 Kilos vinicola		SITE	4kilos.com			

Figura 3.1. Detalle del fichero master\_Spa y de la fórmula de las claves UID

La fórmula diseñada para las claves UID de las empresas internacionales es la siguiente:

```
= SI($G2="Canadá"; "CA"; SI($G2="Estados Unidos"; "US"; "UK")) & SI($F2="blog"; "B"; "V") & SI(LARGO($B2)=1; "000"; SI(LARGO($B2)=2; "00"; SI(LARGO($B2)=3; "0"; ""))) & $B2 & SI($D2="FACEBOOK"; "F"; SI($D2="GOOGLE+"; "G"; SI($D2="INSTAGRAM"; "I"; SI($D2="LINKEDIN"; "L"; SI($D2="PINTEREST"; "P"; SI($D2="SITE"; "S"; SI($D2="TWITTER"; "T"; "Y"))))))))
```





	A	B	C	D	E	F	G	H	I
1	uid2	num	empresa	fuente	url2	tipo	pais		
2	UKB0001S	1	12x75.com	SITE	12x75.com	blog	Reino Unido		
3	USV0002F	2	14 Hands Winery	FACEBOOK	facebook.com/14handswine	tienda virtual	Estados Unidos		
4	USV0002S	2	14 Hands Winery	SITE	14hands.com	tienda virtual	Estados Unidos		
5	USV0002T	2	14 Hands Winery	TWITTER	twitter.com/14handswine	tienda virtual	Estados Unidos		
6	USV0002Y	2	14 Hands Winery	YOUTUBE	youtube.com/user/14handswine	tienda virtual	Estados Unidos		
7	CAV0003S	3	50th Parallel Estate	SITE	50thparallel.com	tienda virtual	Canadá		
8	USB0004F	4	A Good Time with Wine	FACEBOOK	facebook.com/agoodtimewithwine	blog	Estados Unidos		
9	USB0004S	4	A Good Time with Wine	SITE	agoodtimewithwine.com	blog	Estados Unidos		
10	USB0004T	4	A Good Time with Wine	TWITTER	twitter.com/mmwine	blog	Estados Unidos		
11	USV0005F	5	ABC Fine Wine & Spirits	FACEBOOK	facebook.com/ABCFineWineSpirits	tienda virtual	Estados Unidos		
12	USV0005L	5	ABC Fine Wine & Spirits	LINKEDIN	linkedin.com/company/abc-fine-wine-&spirits	tienda virtual	Estados Unidos		
13	USV0005P	5	ABC Fine Wine & Spirits	PINTEREST	pinterest.com/abcfws	tienda virtual	Estados Unidos		
14	USV0005S	5	ABC Fine Wine & Spirits	SITE	abcfws.com	tienda virtual	Estados Unidos		
15	USV0005T	5	ABC Fine Wine & Spirits	TWITTER	twitter.com/abcwinecountry	tienda virtual	Estados Unidos		
16	USV0005Y	5	ABC Fine Wine & Spirits	YOUTUBE	youtube.com/user/ABCWineCountry	tienda virtual	Estados Unidos		
17	USV0006F	6	Adam Puchta Winery	FACEBOOK	facebook.com/adampuchtawinery	tienda virtual	Estados Unidos		
18	USV0006G	6	Adam Puchta Winery	GOOGLE+	plus.google.com/104702769922459693241	tienda virtual	Estados Unidos		
19	USV0006P	6	Adam Puchta Winery	PINTEREST	pinterest.com/adampuchtawine	tienda virtual	Estados Unidos		
20	USV0006S	6	Adam Puchta Winery	SITE	adampuchtawine.com	tienda virtual	Estados Unidos		
21	USV0006T	6	Adam Puchta Winery	TWITTER	twitter.com/AdamPuchtaWines	tienda virtual	Estados Unidos		
22	UKV0007F	7	Adnams Southwold	FACEBOOK	facebook.com/adnams	tienda virtual	Reino Unido		
23	UKV0007G	7	Adnams Southwold	GOOGLE+	plus.google.com/+adnams	tienda virtual	Reino Unido		
24	UKV0007I	7	Adnams Southwold	INSTAGRAM	instagram.com/adnams	tienda virtual	Reino Unido		
25	UKV0007P	7	Adnams Southwold	PINTEREST	pinterest.com/adnams	tienda virtual	Reino Unido		

Figura 3.2. Detalle del fichero master\_int y de la fórmula de las claves UID.

### 3.1.2. Muestra 1

Inicialmente se toma como base de datos las propias tablas de EXCEL, correspondientes a los ficheros *master* en formato XLSX, modificándolas para que puedan albergar también la cantidad de nombramientos que contiene el fichero de *hits* y añadiendo una pestaña para cada recogida mensual de datos.

Se realiza un emparejamiento de todas las empresas internacionales con todas las nacionales y así se deja el fichero *master* preparado para luego rellenar las celdas correspondientes con el número de nombramientos. Para ello, se modifica el fichero *master* de empresas internacionales desde EXCEL, añadiendo tantas columnas como claves UID hay en el fichero *master* de empresas españolas y quedando la tabla de EXCEL ya preparada (según muestra la Figura 3.3) para insertarle el número de *hits*.

	A	B	C	D	E	F	G	H	I	J	K
1	uid2	num	empresa	fuente	url2	tipo	pais	SPA0001I	SPA0001S	SPA0001T	SPA0002I
2	UKB0001S	1	12x75.com	SITE	12x75.com	blog	Reino Unido				
3	USV0002F	2	14 Hands Winery	FACEBOOK	facebook.com/14handswine	tienda virtual	Estados Unidos				
4	USV0002S	2	14 Hands Winery	SITE	14hands.com	tienda virtual	Estados Unidos				
5	USV0002T	2	14 Hands Winery	TWITTER	twitter.com/14handswine	tienda virtual	Estados Unidos				
6	USV0002Y	2	14 Hands Winery	YOUTUBE	youtube.com/user/14handswine	tienda virtual	Estados Unidos				
7	CAV0003S	3	50th Parallel Estate	SITE	50thparallel.com	tienda virtual	Canadá				
8	USB0004F	4	A Good Time with Wine	FACEBOOK	facebook.com/agoodtimewithwine	blog	Estados Unidos				
9	USB0004S	4	A Good Time with Wine	SITE	agoodtimewithwine.com	blog	Estados Unidos				
10	USB0004T	4	A Good Time with Wine	TWITTER	twitter.com/mmwine	blog	Estados Unidos				
11	USV0005F	5	ABC Fine Wine & Spirits	FACEBOOK	facebook.com/ABCFineWineSpirits	tienda virtual	Estados Unidos				
12	USV0005L	5	ABC Fine Wine & Spirits	LINKEDIN	linkedin.com/company/abc-fine-wine-&spirits	tienda virtual	Estados Unidos				
13	USV0005P	5	ABC Fine Wine & Spirits	PINTEREST	pinterest.com/abcfws	tienda virtual	Estados Unidos				
14	USV0005S	5	ABC Fine Wine & Spirits	SITE	abcfws.com	tienda virtual	Estados Unidos				
15	USV0005T	5	ABC Fine Wine & Spirits	TWITTER	twitter.com/abcwinecountry	tienda virtual	Estados Unidos				
16	USV0005Y	5	ABC Fine Wine & Spirits	YOUTUBE	youtube.com/user/ABCWineCountry	tienda virtual	Estados Unidos				
17	USV0006F	6	Adam Puchta Winery	FACEBOOK	facebook.com/adampuchtawinery	tienda virtual	Estados Unidos				
18	USV0006G	6	Adam Puchta Winery	GOOGLE+	plus.google.com/104702769922459693241	tienda virtual	Estados Unidos				
19	USV0006P	6	Adam Puchta Winery	PINTEREST	pinterest.com/adampuchtawine	tienda virtual	Estados Unidos				
20	USV0006S	6	Adam Puchta Winery	SITE	adampuchtawine.com	tienda virtual	Estados Unidos				
21	USV0006T	6	Adam Puchta Winery	TWITTER	twitter.com/AdamPuchtaWines	tienda virtual	Estados Unidos				
22	UKV0007F	7	Adnams Southwold	FACEBOOK	facebook.com/adnams	tienda virtual	Reino Unido				
23	UKV0007G	7	Adnams Southwold	GOOGLE+	plus.google.com/+adnams	tienda virtual	Reino Unido				

Figura 3.3. Fichero master\_int con las columnas añadidas de las claves UID nacionales.

Para rellenar el fichero *master* con el número de nombramientos, como primera opción, se considera la programación de un *script*, que realice la inserción del número de nombramientos en dicha tabla, leyéndolos previamente del fichero de *hits*. Obviamente, este procedimiento va a ser altamente costoso, en cuanto a tiempo de ejecución, ya que los procesos de lectura y escritura en ficheros se demoran mucho. Sin embargo, se decide completar su implementación con el fin de testar su funcionamiento. Se utiliza para su implementación la librería *OpenPyXL*. Pero previamente a la lectura e inserción del número de nombramientos, es necesario tratar el fichero de *hits*, puesto que vienen todos los datos de las URL juntos en una sola columna y hay que dividirlo en tres columnas, para poder leerlos desde el *script*. Esta operación se ejecuta en un primer momento desde EXCEL, utilizando las herramientas que ofrece el gestor de hojas de cálculo, quedando el fichero de *hits* según se muestra en la Figura 3.4. Posteriormente, dada la necesidad de automatizar los procesos, se confecciona un *script* (véase Anexo I, *script* 6) que normaliza el fichero CSV, se describe más adelante.

	A	B	C	D	E	F	G	H	I	J
1	url mencionada	site en el que se menciona	hit_number							
2	12x75.com	10dabril.com	0							
3	12x75.com	10sentits.cat	0							
4	12x75.com	15albas.es	0							
5	12x75.com	2859.es.all.biz	0							
6	12x75.com	2amigos.com	0							
7	12x75.com	3asesvino.com	0							
8	12x75.com	40gradosnorte.com	0							
9	12x75.com	4kilos.com	0							
10	12x75.com	7magnifics.com	0							
11	12x75.com	a2vinoycultura.com	0							
12	12x75.com	aalto.es	0							
13	12x75.com	abadal.net	0							
14	12x75.com	abadia-retuerta.com	0							
15	12x75.com	abadiadeacon.com	0							
16	12x75.com	abadiadearibayos.es	0							
17	12x75.com	abadialaarroyada.es	0							
18	12x75.com	abarando.com	0							
19	12x75.com	abeica.com	0							
20	12x75.com	abiotxakolina.com	0							
21	12x75.com	acontia.es	0							
22	12x75.com	acoroa.com	0							
23	12x75.com	actualidad.campante.com	0							

Figura 3.4. Fichero de *hits* con los datos separados en tres columnas.

Se comienza por implementar un primer *script* (véase Anexo I, pág. 72), que lleva a cabo la inicialización a 0, en el fichero *master*, de las celdas que van a contener el número de nombramientos. Posteriormente, se implementa el *script* 2 (véase Anexo I, pág. 73), que va a recorrer el fichero *master*, creando un diccionario de claves-valor (*{key: value}*), en el que la *key* va a ser la URL y el *value* va a ser la clave UID correspondiente, después va a rellenar en el fichero de *hits* dos columnas, que se han insertado previamente desde EXCEL, con las claves UID correspondientes a cada URL, así se podrán realizar posteriores búsquedas en el fichero *master* por coincidencias de clave UID.

Pero al ejecutar este último *script* surgen problemas de codificación de formatos, ya que hay un carácter especial que no se reconoce en la lectura del fichero de *hits*. Tras varias pruebas intentando recodificar el fichero, con distintos formatos de codificación (*Unicode*, *UTF-8*, *ASCII*, etc.) y no conseguir ningún resultado positivo, se



decide efectuar una búsqueda manual del carácter no reconocido, que resulta ser el símbolo "Đ", que se corresponde con la letra "ñ". Se procede a sustituir dicho carácter desde EXCEL, con la herramienta de "Búsqueda y reemplazo", después de esto el *script* 2 se ejecuta sin problemas, insertando las dos columnas con las claves UID correspondientes a cada URL en el fichero de *hits* (ver Figura 3.5).

	A	B	C	D	E	F	G	H	I
1	ud1	url mencionada	ud2	site en el que se menciona	hit_number				
2	UKB0001S	12x75.com	SPA1929S	10dabril.com	0				
3	UKB0001S	12x75.com	SPA0002S	10sentits.cat	0				
4	UKB0001S	12x75.com	SPA0003S	15albas.es	0				
5	UKB0001S	12x75.com	SPA2129S	2859.es.all.biz	0				
6	UKB0001S	12x75.com	SPA0004S	2amigos.com	0				
7	UKB0001S	12x75.com	SPA0005S	3asesvino.com	0				
8	UKB0001S	12x75.com	SPA0365S	40gradosnorte.com	0				
9	UKB0001S	12x75.com	SPA0006S	4kilos.com	0				
10	UKB0001S	12x75.com	SPA0007S	7magnificos.com	0				
11	UKB0001S	12x75.com	SPA0011S	a2vinoycultura.com	0				
12	UKB0001S	12x75.com	SPA0368S	aalto.es	0				
13	UKB0001S	12x75.com	SPA0013S	abadal.net	0				
14	UKB0001S	12x75.com	SPA0017S	abadia-retuerta.com	0				
15	UKB0001S	12x75.com	SPA0014S	abadiadeacon.com	0				
16	UKB0001S	12x75.com	SPA0015S	abadialaarroyada.es	0				
17	UKB0001S	12x75.com	SPA0369S	abadialaarroyada.es	0				
18	UKB0001S	12x75.com	SPA2631S	abarando.com	0				
19	UKB0001S	12x75.com	SPA0018S	abeica.com	0				
20	UKB0001S	12x75.com	SPA0020S	abiotxakolina.com	0				
21	UKB0001S	12x75.com	SPA0592S	acontia.es	0				
22	UKB0001S	12x75.com	SPA0009S	acoroa.com	0				
23	UKB0001S	12x75.com	SPA1593S	actualidad.campante.com	0				

Figura 3.5. Fichero de hits con las dos columnas de las claves UID añadidas.

Se decide mejorar y ampliar el código del *script* 2 en el *script* 3 (véase Anexo I, pág. 75), para ello se crea también un diccionario de claves-valor que va a almacenar una tupla con las dos claves UID como *key* del diccionario y el *value* será el número de nombramientos ( $\{(UID1, UID2): hit\_number\}$ ). Este segundo diccionario se va a llenar con el número de nombramientos conforme se recorre el fichero de *hits* y al mismo tiempo se van a ir sustituyendo las URL por sus correspondientes claves UID, en vez de insertar dos nuevas columnas, con lo que se reduce considerablemente el tiempo de ejecución. Después se recorren las filas y columnas del fichero *master*, llenando las celdas cuyas claves UID hacen *matching*<sup>20</sup> (Villate, 2005) con las *keys* del diccionario, con el *value* asociado, si no hacen *matching* se pone un 0. De esta forma se puede suprimir el proceso de inicialización a 0 de las celdas del fichero *master*, por lo que el *script* de inicialización pierde su utilidad y queda rechazado. El resultado final de este proceso se ve reflejado en la Figura 3.6 y la ejecución del *script* en la Figura 3.7.

<sup>20</sup> Matching: concordancia, coincidencia, emparejamiento.

uid	num	empresa	fuente	url	tipo	pais	SPA0001I	SPA0001S	SPA0001T	SPA0002I	SPA0002S	SPA0002T
UK80001S	1	12x75.com	SITE	12x75.com	blog	Reino Unido	0	0	0	0	0	0
USV0002F	2	14 Hands Winery	FACEBOOK	facebook.com/14handswine	tienda virtual	Estados Unidos	0	0	0	0	0	0
USV0002S	2	14 Hands Winery	SITE	14hands.com	tienda virtual	Estados Unidos	0	0	0	0	0	0
USV0002T	2	14 Hands Winery	TWITTER	twitter.com/14handswine	tienda virtual	Estados Unidos	0	0	0	0	0	0
USV0002Y	2	14 Hands Winery	YOUTUBE	youtube.com/user/14handswine	tienda virtual	Estados Unidos	0	0	0	0	0	0
CAV0003S	3	50th Parallel Estate	SITE	50thparallel.com	tienda virtual	Canadá	0	0	0	0	0	0
US80004F	4	A Good Time with Wine	FACEBOOK	facebook.com/agoodtimewithwine	blog	Estados Unidos	0	0	0	0	0	0
US80004S	4	A Good Time with Wine	SITE	agoodtimewithwine.com	blog	Estados Unidos	0	0	0	0	0	0
US80004T	4	A Good Time with Wine	TWITTER	twitter.com/mmwine	blog	Estados Unidos	0	0	0	0	0	0
USV0005F	5	ABC Fine Wine & Spirits	FACEBOOK	facebook.com/ABCFineWineSpirits	tienda virtual	Estados Unidos	0	0	0	0	0	0
USV0005S	5	ABC Fine Wine & Spirits	LINKEDIN	linkedin.com/company/bc-fine-wine-&-spirits	tienda virtual	Estados Unidos	0	0	0	0	0	0
USV0005P	5	ABC Fine Wine & Spirits	PINTEREST	pinterest.com/abcwines	tienda virtual	Estados Unidos	0	0	0	0	0	0
USV0005S	5	ABC Fine Wine & Spirits	SITE	abcwines.com	tienda virtual	Estados Unidos	0	0	0	0	0	0
USV0005T	5	ABC Fine Wine & Spirits	TWITTER	twitter.com/abcwinecountry	tienda virtual	Estados Unidos	0	0	0	0	0	0
USV0005Y	5	ABC Fine Wine & Spirits	YOUTUBE	youtube.com/user/ABCWineCountry	tienda virtual	Estados Unidos	0	0	0	0	0	0
USV0006F	6	Adam Puchta Winery	FACEBOOK	facebook.com/adampuchtawinery	tienda virtual	Estados Unidos	0	0	0	0	0	0
USV0006G	6	Adam Puchta Winery	GOOGLE+	plus.google.com/104702769922459693241	tienda virtual	Estados Unidos	0	0	0	0	0	0
USV0006P	6	Adam Puchta Winery	PINTEREST	pinterest.com/adampuchtawine	tienda virtual	Estados Unidos	0	0	0	0	0	0
USV0006S	6	Adam Puchta Winery	SITE	adampuchtawine.com	tienda virtual	Estados Unidos	0	0	0	0	0	0
USV0006T	6	Adam Puchta Winery	TWITTER	twitter.com/AdamPuchtaWines	tienda virtual	Estados Unidos	0	0	0	0	0	0
UKV0007F	7	Adnams Southwold	FACEBOOK	facebook.com/adnams	tienda virtual	Reino Unido	0	0	0	0	0	0
UKV0007G	7	Adnams Southwold	GOOGLE+	plus.google.com/+adnams	tienda virtual	Reino Unido	0	0	0	0	0	0
UKV0007I	7	Adnams Southwold	INSTAGRAM	instagram.com/adnams	tienda virtual	Reino Unido	0	0	0	0	0	0

Figura 3.6. Fichero master\_int relleno con el número de nombramientos.

```

~/Documents/UPV/4 CURS0/2 CUAT/TFG/BD/2 R- sobrescribiendo cols urls - openpyxl -- bash
gloriemunozedo@Gloria:~/Documents/UPV/4 CURS0/2 CUAT/TFG/BD/2 R- sobrescribiendo cols urls - openpyxl $ python vino_int.py
Cargando datos de ficheros excel ...
V2.VINO_master_int.xlsx --- OK
V2.VINO_master_Spa.xlsx --- OK
Wine_A_site-siteFinal.xlsx --- OK
Llenando los diccionarios de claves ...
Diccionario claves UID int --- OK
Diccionario claves UID spa --- OK
Añadiendo las claves UID al fichero Wine_A_site-siteFinal.xlsx ...
OK
Guardando cambios en un nuevo fichero wine_site_final.xlsx ...
OK
Llenando el fichero V2.VINO_master_int.xlsx con los hits extraídos ...
OK
Guardando cambios en fichero V2.VINO_master_int.xlsx ...
OK
Tiempo transcurrido 552.64

```

Figura 3.7. Detalle de la ejecución del script de llenado del fichero master\_int.

A pesar de que todas estas modificaciones han mejorado el tiempo de ejecución, que ha pasado de 13 minutos y 11 segundos a 9 minutos y 21 segundos (como se puede apreciar en la imagen anterior), se considera que es un tiempo de ejecución excesivamente alto, en vistas a integrar el script como un módulo, en un *applet*<sup>21</sup> (Wikipedia, 2016) de ejecución en tiempo real. Por tal motivo se decide descartar esta primera opción de implementación, consistente en que las tablas de EXCEL alberguen la base de datos, y se pasa a considerar otra opción totalmente distinta, que es migrar los archivos EXCEL a una base de datos. Como posibles opciones de bases de datos se barajan *MongoDB*, *PostgreSQL*, *Oracle* y *MySQL*. Se decide utilizar *MySQL* por su gran rapidez en búsquedas e indexación de campos de texto, simplicidad, disponibilidad en gran cantidad de plataformas y sistemas y su gran capacidad de integración en librerías de cualquier lenguaje de programación.

<sup>21</sup> Componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web.

Para conectar con el servidor de *MySQL* se utiliza la aplicación *MAMP*, y como gestora de la base de datos se utiliza la aplicación *MySQL Workbench* (ver proceso de instalación y configuración en el apartado 2.5). Antes de la ejecución de cualquier *script*, hay que iniciar el servidor de *MySQL* con la aplicación *MAMP*.

Previamente a la importación de los ficheros *master* a la base de datos, hay que tratar los formatos para evitar futuros problemas de codificación, por esto se pasa a formato CSV los ficheros *master*, que venían en formato XLSX, utilizando para ello el programa *Libre Office*; se decide hacer así debido a que este programa, en las configuraciones de guardado, permite elegir la codificación *UTF-8*.

Posteriormente, se importan los ficheros *master* en formato CSV a *MySQL*, utilizando la aplicación *MySQL Workbench* e identificando el campo de clave UID como clave única e indexada. A continuación se describe el proceso de importación del fichero *master\_int* a la citada aplicación, siendo el proceso de importación del fichero *master\_spa* similar a éste.

En la interfaz de la aplicación *MySQL Workbench*, se aprecia en la parte inferior izquierda la etiqueta "*Tables*", al hacer clic con el botón derecho del ratón sobre ella, se abre el menú contextual. Hay que seleccionar la opción "*Table Data Import Wizard*" (ver Figura 3.8). En la siguiente ventana que se abre, se busca con el explorador el fichero en formato CSV a importar y se clicca sobre el botón "*Next*".

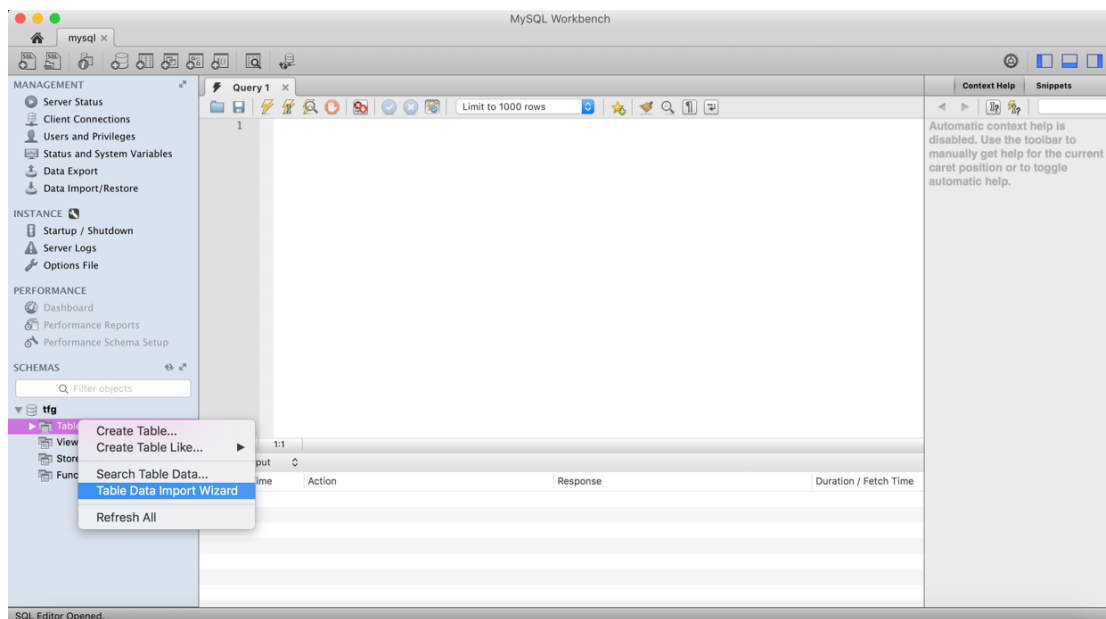


Figura 3.8. Detalle del proceso de importación a *MySQL Workbench* del fichero *master\_int*.

En la siguiente ventana se selecciona la opción "*Create new table*" y se le da nombre a la nueva tabla. En el siguiente paso (Figura 3.9), el asistente muestra el formato de codificación que se va a utilizar, los campos que ha reconocido junto con su tipo y una muestra de cómo va a quedar la tabla. Hay que comprobar que esté todo correcto antes de clicar en "*Next*".

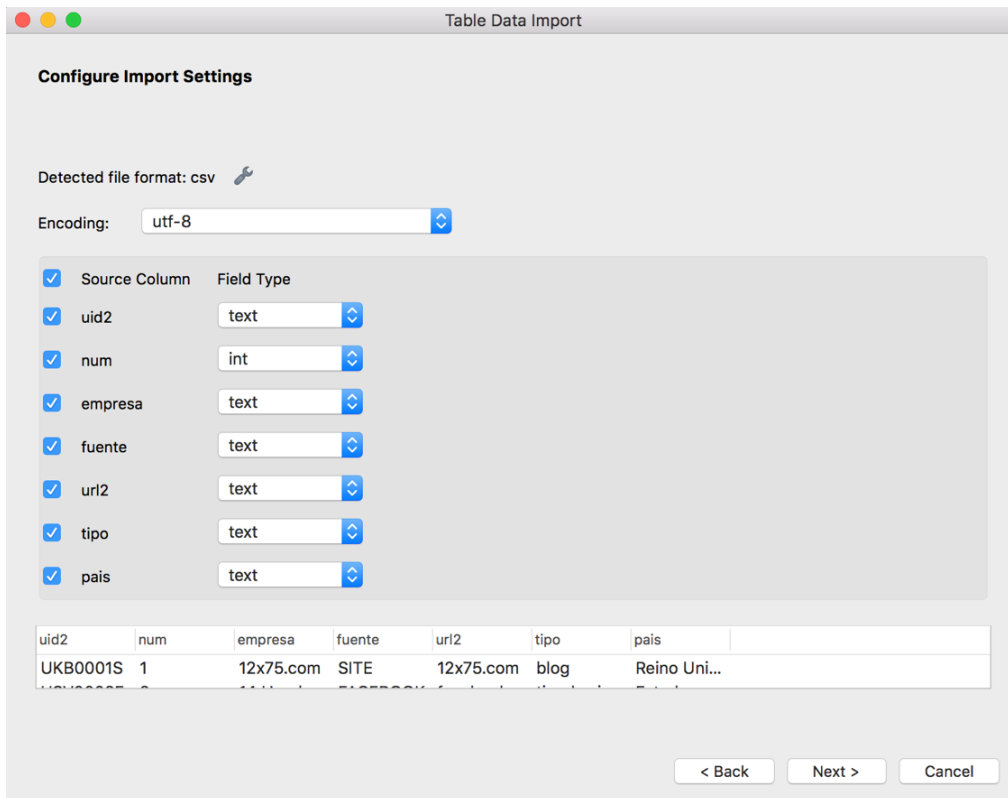


Figura 3.9. Detalle del proceso de importación a MySQL Workbench del fichero master\_int, ventana de configuración de los ajustes de importación.

El asistente comunica que se va a proceder con la importación del fichero, se clicca en "Next", a continuación el asistente informa que el proceso ha terminado con éxito (Figura 3.10) y, en la siguiente ventana, muestra un resumen de los datos de la importación (Figura 3.11), en la que dice que ha importado un fichero CSV, que ha tardado 2,53 segundos en realizar la importación, que ha creado una nueva tabla (*vino\_master\_int*) y que ha importado a ésta 1.046 registros.

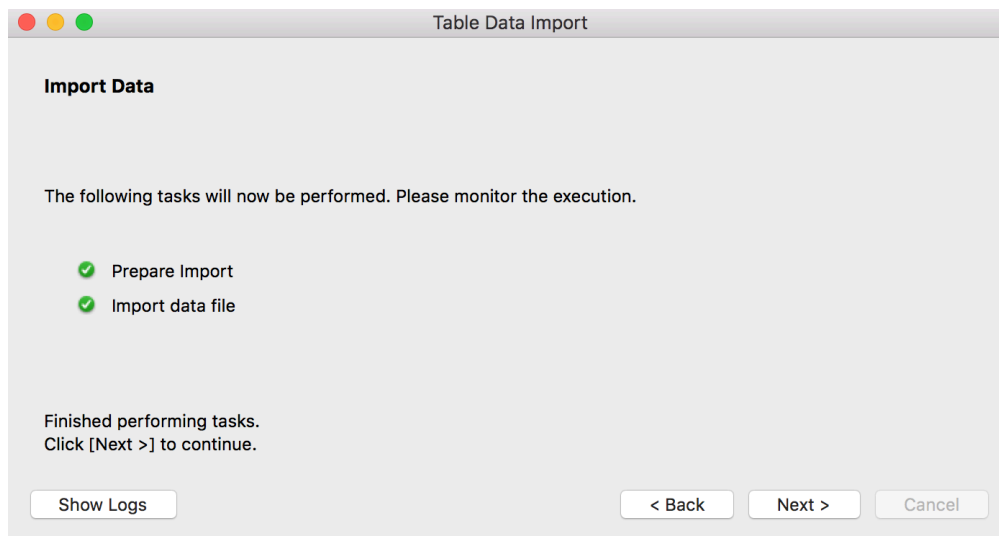


Figura 3.10. Detalle del proceso de importación a MySQL Workbench del fichero master\_int, proceso de importación terminado con éxito.

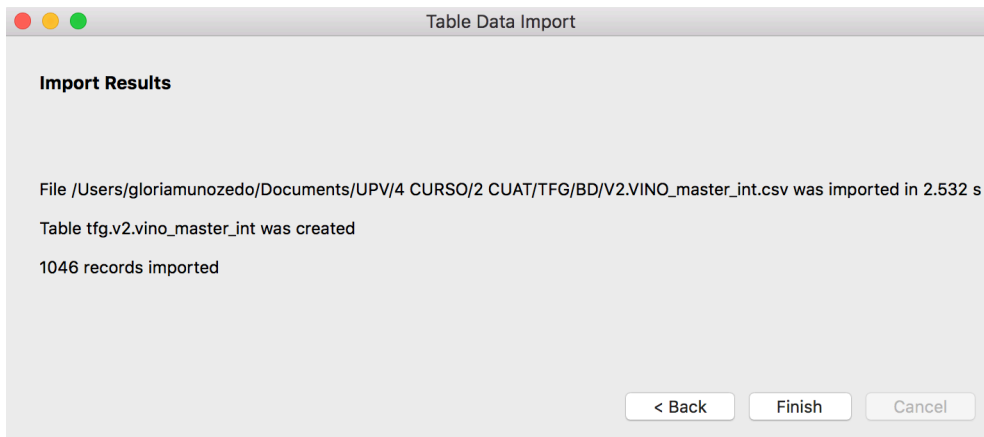


Figura 3.11. Detalle del proceso de importación a MySQL Workbench del fichero master\_int, resultado de la importación.

Así quedan ya importadas las dos tablas de los ficheros master a la aplicación MySQL Workbench, de las que se puede ver el detalle en la Figura 3.12 y en la Figura 3.13.

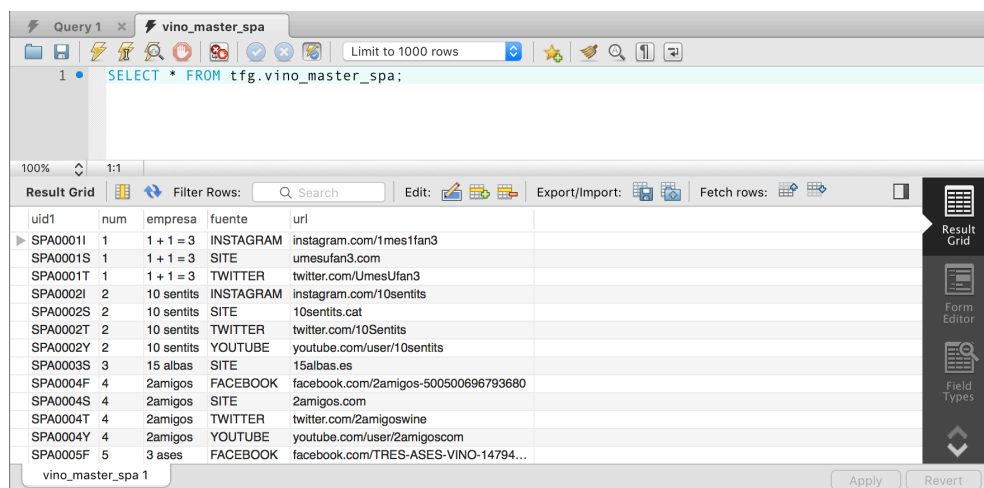


Figura 3.12. Vista en MySQL Workbench de la tabla correspondiente al fichero master\_spa.

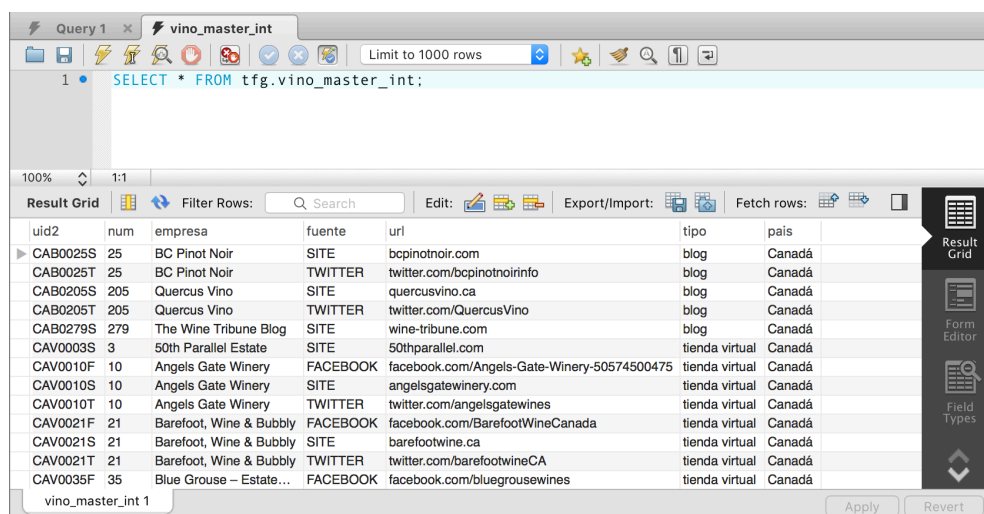
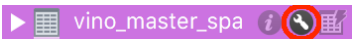


Figura 3.13. Vista en MySQL Workbench de la tabla correspondiente al fichero master\_int.

Para terminar la importación de estas dos tablas, solo queda marcar como claves primarias los campos *uid1* y *uid2*, operación que se realiza desde la aplicación *MySQL Workbench*. Para ello, se selecciona la tabla *vino\_master\_spa* en el panel izquierdo y se clic sobre la opción de configuración (la llave inglesa de la imagen) . A continuación se abre una ventana en la que se pueden modificar las condiciones de los campos, en ella se marca para la clave *uid1* la opción "PK" (*Primary Key*), automáticamente se marcará la opción "NN" (*Not Null*) (ver Figura 3.14), después se hace clic sobre el botón "Apply", al hacer esto se abre otra ventana (Figura 3.15) en la que se muestra en lenguaje *SQL*, la acción que se va a realizar, se vuelve a clicar en "Apply" y ya está aplicado el cambio. Se realiza el mismo procedimiento para la tabla *vino\_master\_int*. Con esto ya quedan las dos tablas perfectamente estructuradas en la base de datos.

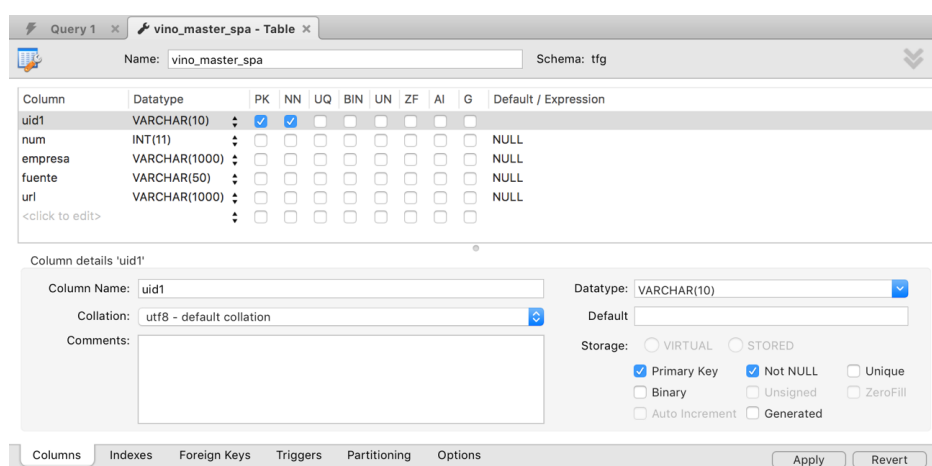


Figura 3.14. Detalle de configuración de la clave primaria en la tabla *vino\_master\_spa*.

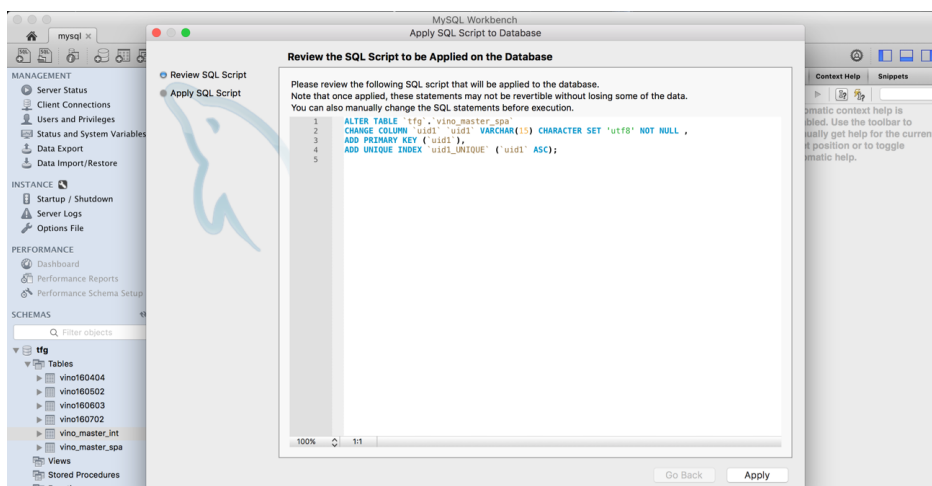


Figura 3.15. Pantalla que muestra en lenguaje *SQL* los cambios que se van a aplicar.

Dado que la finalidad de este trabajo es automatizar el máximo de tareas posibles, se decide a continuación implementar el *script 4* (véase Anexo I, pág. 78), para mapear el fichero de *hits* del formato CSV original (en el que vienen los datos de las URL en una única columna) al formato CSV normalizado (columnas separadas por comas y sustitución de caracteres especiales mal codificados) y así poder importarlo a



*MySQL* sin los citados problemas de codificación, para ello se utiliza la librería *csv* de *Python*. Después de normalizar el fichero de *hits*, se importa a *MySQL*, en un primer momento, utilizando también la aplicación de *MySQL Workbench*, siguiendo los pasos descritos anteriormente, posteriormente este proceso se automatiza en los *script* 6 y 13, detallados en los siguientes párrafos.

El siguiente paso es añadir las dos columnas con las claves *uid1* y *uid2* a la tabla de *hits*, para poder tenerla también indexada por claves primarias en *MySQL*. Para ello, se implementa el *script* 5 (véase Anexo I, pág. 79), que crea una tabla con el nombre del archivo, importa los datos del CSV a la base de datos, inserta las dos columnas y las rellena leyendo las claves de la tabla *master\_spa* y haciendo *matching* con las URL para insertarlas. El proceso de ejecución del *script* 5 se aborta a los siete minutos de su inicio, pues se considera que el tiempo es excesivo y se rechaza. También se elimina la tabla *wine\_site\_final* (correspondiente al fichero de *hits*) de la base de datos, haciendo uso de la opción "*Drop Table ...*" desde *MySQL Workbench*, para repetir el proceso.

Seguidamente, se implementa el *script* 6 (véase Anexo I, pág. 81), que es una ampliación del *script* 4. Este *script* conecta con la base de datos (haciendo uso de la librería *mysql.connector* de *Python*), obtiene las claves de las tablas *master*, crea un diccionario (clave UID, valor URL), normaliza el fichero original de *hits* en formato CSV, corrigiendo los caracteres mal codificados y separándolos en tres columnas, posteriormente inserta las dos columnas con las claves UID, haciendo *matching* con el diccionario de claves que se había creado y modificando posteriormente el fichero CSV con las correcciones. Para importar el fichero CSV a la base de datos, se modifica el *script* 5 en el 7 (véase Anexo I, pág. 83), el cual crea una tabla con el nombre del fichero CSV, pero esta vez con 5 columnas, conecta con la base de datos, importa el fichero de *hits* a *MySQL* y añade la condición de claves primarias a los campos *uid1* y *uid2*.

Al hacer este último proceso, se detecta otra anomalía (ver Figura 3.16) en el fichero original de *hits*: La clave *uid2*: UKV0007S correspondiente a la URL "adnams.co.uk" y la clave *uid1*: SPA0006S correspondiente a la URL "4kilos.com", tienen tres entradas idénticas; se procede a la eliminación de dos de dichas filas del fichero de *hits*. Se vuelve a ejecutar el *script*, esta vez con éxito, comprobando que por este método hay una mejora notable en cuanto a tiempo de ejecución, siendo el total, entre la normalización del CSV y su importación a *MySQL*, de tan solo 10 segundos.

```

gloriamunozedo@Gloria:~/Documents/UPV/4 CURSO/2 CUAT/TFG/BD/8 - csv scrapy normal con claves a mysql $ python vino_hits.py
Obteniendo las claves de los tablas master de MySQL ...
OK
Leyendo el fichero csv original ...add_key = ("ALTER TABLE " + csv + " CHANGE COLUMN 'uid1' 'uid1' VARCHAR(10) CHARA
Normalizando csv ...
OK
cursor.execute(add_key)
Table wine_site_final created OK
Data imported OK
Something went wrong: 1062 (23000): Duplicate entry 'UKV0007S,a' for key 'PRIMARY'
Tiempo transcurrido 9.52
gloriamunozedo@Gloria:~/Documents/UPV/4 CURSO/2 CUAT/TFG/BD/8 - csv scrapy normal con claves a mysql $ python vino_hits.py
Obteniendo las claves de los tablas master de MySQL ...
OK
Leyendo el fichero csv original ...time_fin = time()
Normalizando csv ...time_total = time_fin - time_ini
OK
Tiempo transcurrido 5.40
gloriamunozedo@Gloria:~/Documents/UPV/4 CURSO/2 CUAT/TFG/BD/8 - csv scrapy normal con claves a mysql $ python csvTmysql2.py
Table wine_site_final created OK
Data imported OK
The elapsed time for table wine_site_final was 4.25 seconds

```

Figura 3.16. Detalle del error de ejecución del *script* 7, así como de su ejecución exitosa.

De esta manera, queda también disponible en base de datos de *MySQL*, el fichero de *hits* (Figura 3.17) proveniente del *web scraping*, cuya tabla ya tiene añadida la condición de claves primarias para los campos *uid1* y *uid2*, condición que se puede comprobar entrando en el cuadro de configuración de la tabla en la aplicación de *MySQL Workbench* (Figura 3.18).

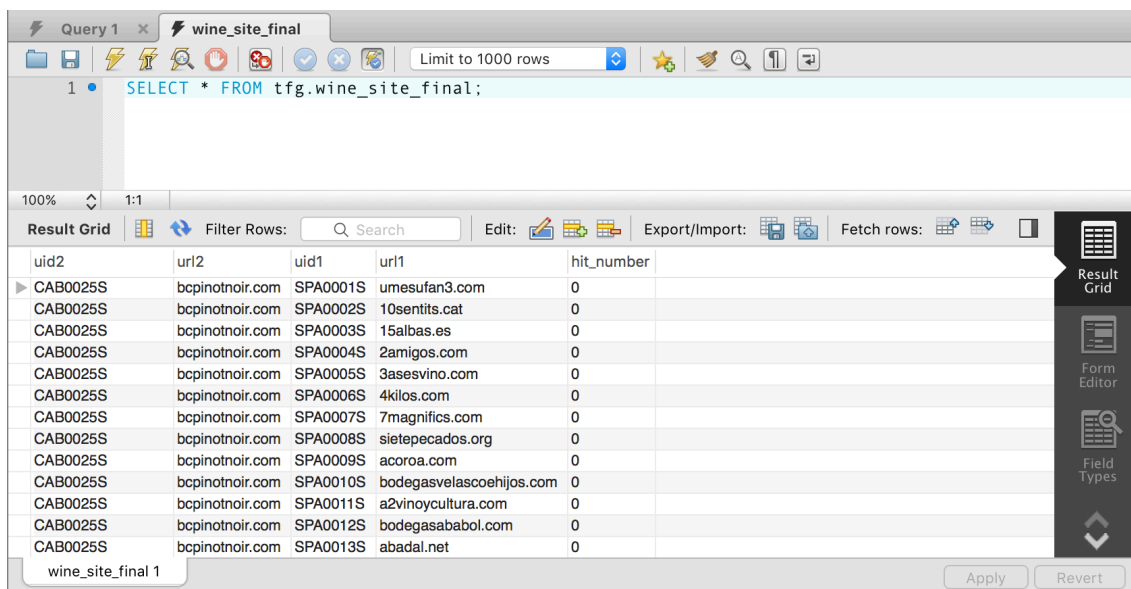


Figura 3.17. Detalle en *MySQL Workbench* de la tabla correspondiente al fichero de *hits*.

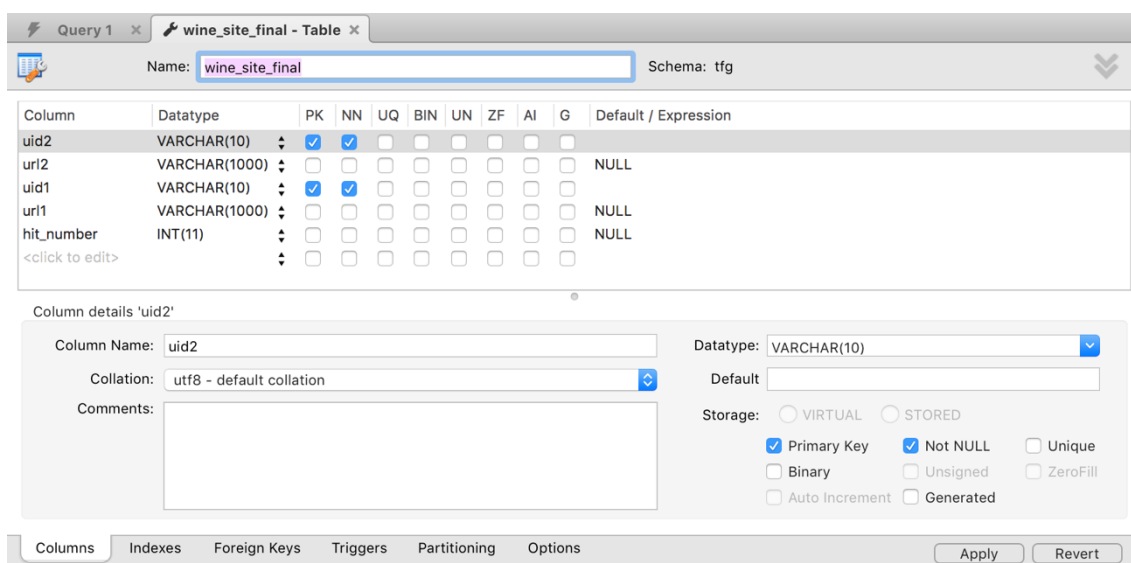


Figura 3.18. Detalle de configuración de las claves primarias de la tabla correspondiente al fichero de *hits*.

### 3.1.3. Muestra 2

Hay que recordar que esta muestra se compone de cuatro ficheros en formato CSV, con las extracciones tomadas durante cuatro meses y que los ficheros contienen 44 indicadores de *Majestic*, correspondientes a las 2.744 empresas españolas de vino.

Estos ficheros también han precisado de un tratamiento de recodificación, previo a su importación a la base de datos, puesto que también se han encontrado caracteres especiales mal codificados, como las combinaciones de caracteres "ï¿½" y "Ã±", ambas se corresponden con la letra "ñ" y el carácter "Ã" que se corresponde con la letra "à". Para sustituir estos caracteres por sus correspondientes bien codificados, se ha implementado el *script* 8 (véase Anexo I, pág. 85). Este *script* se importa como módulo en el *script* 9 (véase Anexo I, pág. 86), el cual realiza la normalización en bloque de todos los ficheros CSV del directorio.

Para importar los ficheros de *Majestic* a *MySQL* se decide implementar los *scripts* 10 y 11 (véase Anexo I, págs. 87 y 90), que los gestionan en bloque, de forma que se accede al directorio en el que se encuentran, se conecta con *MySQL*, se crea una tabla por cada fichero CSV con el nombre del fichero, se importan todos los datos a cada tabla correspondiente, se añade una columna tipo *DATE* en cada tabla, en la que se incluye la fecha en la que se realizó la extracción de los datos con *Majestic* (que se coge del nombre del fichero) y también se añade el campo *uid1* correspondiente a cada empresa, asignándole a este último la condición de clave primaria. El campo *DATE* se añade con el objetivo de poder hacer análisis longitudinales y analizar datos filtrados por fecha. Al añadir los campos *DATE* y *uid1* las tablas provenientes de los ficheros de *Majestic* se quedan con 46 columnas.

Al ejecutar este último *script* aparece otra anomalía en los ficheros: las entradas de las URL "srubios.com" y "cigarralsantamaria.com", en el fichero *master* en formato XLSX de las empresas españolas, tienen un espacio después del ".com", por lo que da un *KeyError* (error de coincidencia de claves) al intentar hacer el *matching* con las consultas de *MySQL*. Se corrigen las entradas en el fichero *master*, se vuelve a importar a *MySQL* y el *script* se ejecuta ya sin problemas, importando todas las tablas de *Majestic* a *MySQL Workbench*.

```
gloriamunozedo@priv2g57-247:~/Documents/UPV/4 CURSO/2 CUAT/TFG/BD/6 - csv majestic con claves a mysql $ python vino_sql.py
Table vino160404 created OK
Data imported OK
Inserting date and key columns ...
  elapsed time for table vino160404 was 11.37 seconds
-----
Table vino160502 created OK
Data imported OK
Inserting date and key columns ...
  elapsed time for table vino160502 was 11.02 seconds
-----
Table vino160603 created OK
Data imported OK
Inserting date and key columns ...
  elapsed time for table vino160603 was 10.47 seconds
-----
Table vino160702 created OK
Data imported OK
Inserting date and key columns ...
  elapsed time for table vino160702 was 10.64 seconds
-----
TOTAL elapsed time was 43.50 seconds
gloriamunozedo@priv2g57-247:~/Documents/UPV/4 CURSO/2 CUAT/TFG/BD/6 - csv majestic con claves a mysql $
```

Figura 3.19. Detalle de la ejecución del *script* 10.

En el detalle de ejecución de la Figura 3.19, se puede apreciar que el tiempo medio de importación de cada tabla es de 10 segundos, lo que se puede considerar aceptable.

Con esta última operación quedan importadas todas las tablas de *Majestic* a la base de datos, con su clave primaria bien definida. Queda por establecer las relaciones entre las claves de todas las tablas. Este proceso se realiza, en un primer momento, desde la aplicación *MySQL Workbench* (posteriormente se automatiza en el *script 12*, descrito en la página siguiente). Para ello, se selecciona la primera tabla a modificar *vino164040* y se hace clic en la opción de configuración (icono de la llave inglesa), en la ventana que se abre se va a la pestaña de "*Foreign Keys*" y se establecen las opciones como se indican en la Figura 3.20, después se clic en "*Apply*" y se abrirá otra ventana de confirmación de los cambios (Figura 3.21), se vuelve a clicar en "*Apply*".

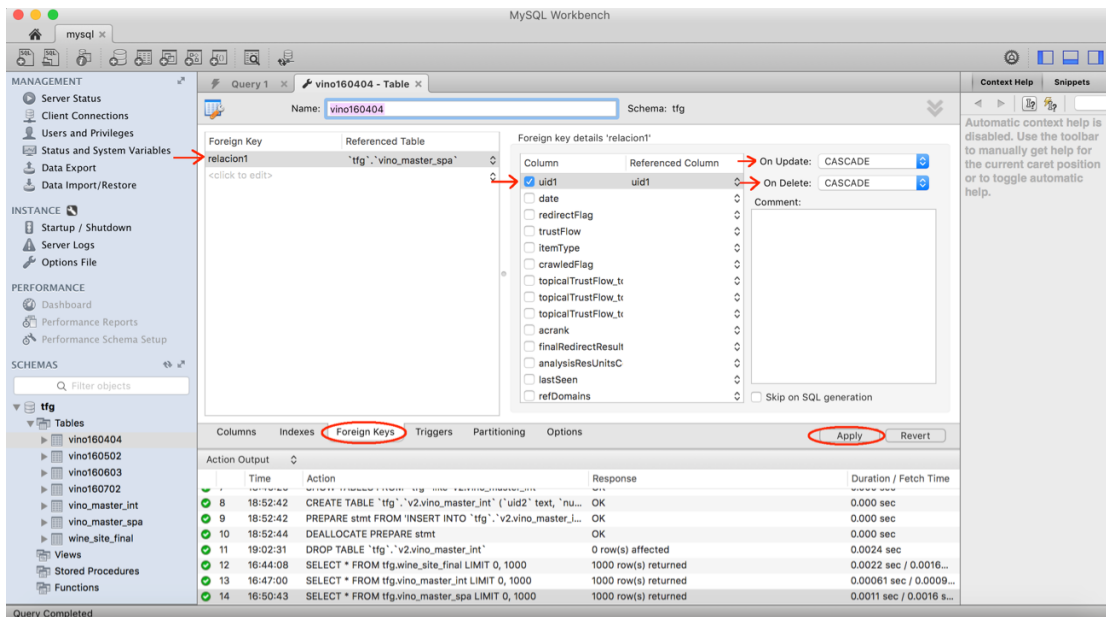


Figura 3.20. Detalle de configuración de las relaciones entre las tablas.

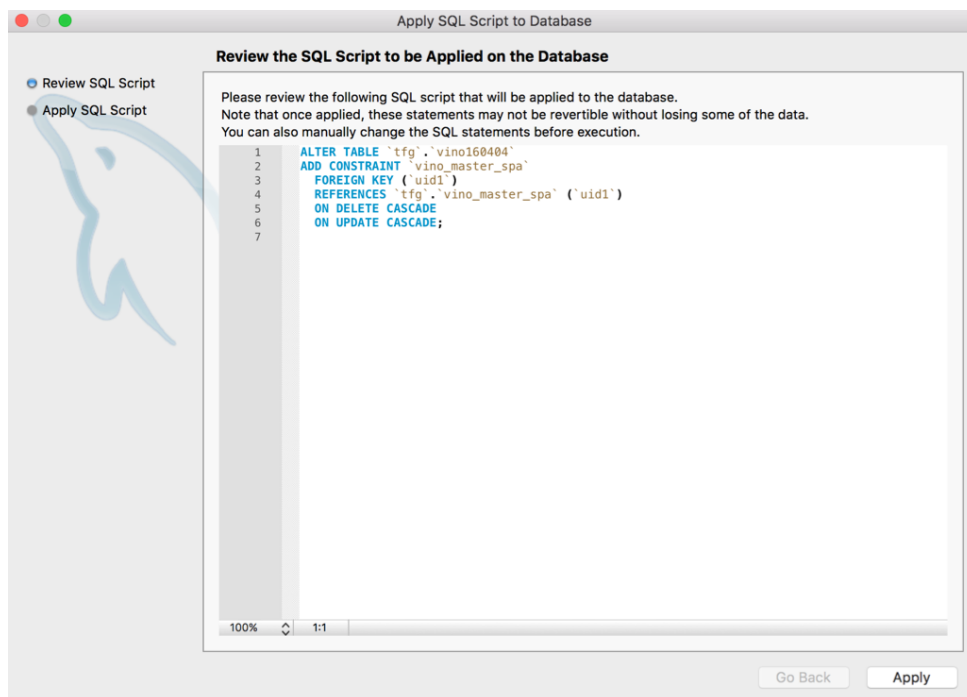


Figura 3.21. Ventana de verificación de los cambios realizados.

Con la finalidad de automatizar al máximo todos los procesos, se implementa el *script* 12 a partir del 10, añadiéndole las sentencias en *SQL*, que insertan las relaciones entre las tablas de los ficheros de *Majestic* con las tablas *master*; para la normalización previa del *CSV* original se utilizan los *script* 8 y 9, y para importar los ficheros a *MySQL* se utiliza el *script* 11. También se modifica el *script* 7 en el 13, que realiza la importación del fichero de *hits*, añadiendo la relación entre las claves primarias de esta tabla y las tablas *master* de las empresas; se utiliza el *script* 6 para la normalización previa del fichero *CSV* original. Así queda por tanto, finalizado el proceso de creación del almacén para los datos, creación de las tablas, importación de los ficheros y creación de las relaciones entre las tablas. La relación final entre todas las tablas se muestra en la Figura 3.22.

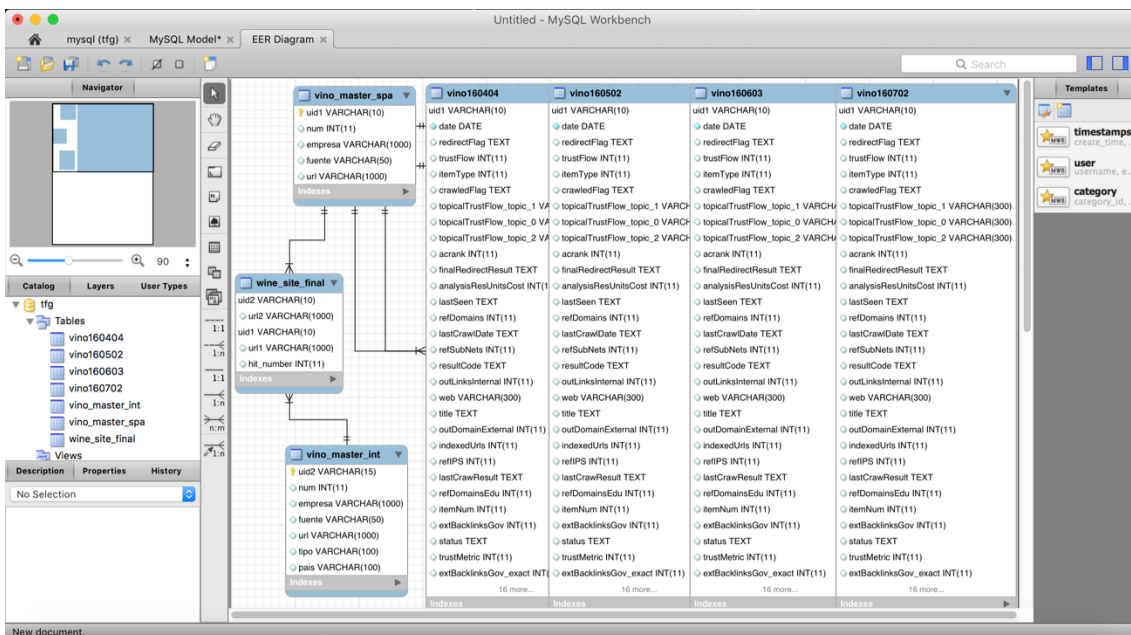


Figura 3.22. Detalle de la relación entre las claves primarias de las tablas.

### 3.1.4. Síntesis del proceso

Para un mejor entendimiento del proceso seguido, se detalla una síntesis en forma de tabla de todos los *scripts* implementados.

#### 3.1.4.1. Muestra 1

En la Tabla 1 se describen los *scripts* 1 a 7 y 13, utilizados para la normalización, estructuración e importación de los ficheros obtenidos mediante *web scraping*, incluyendo las librerías usadas para su desarrollo, así como su estado en el diseño (solución adoptada, reemplazado o rechazado). Por su parte, en la Figura 3.23 se ilustra el proceso de interrelación de estos *scripts* en el proceso.

Script	Estado	Librerías	Descripción
1	Rechazado	OpenPyXL Warnings	Inicializa a 0 las celdas del fichero <i>master_int</i>
2	Reemplazado por el 3	OpenPyXL Warnings	Rellena dos columnas adicionales del fichero de <i>hits</i> , insertadas previamente con EXCEL, con las claves UID que lee de los ficheros <i>master</i> , haciendo <i>matching</i> con las URL correspondientes del fichero de <i>hits</i> , después rellena el fichero <i>master_int</i> con el número de nombramientos que lee del fichero de <i>hits</i> .
3	Rechazado	OpenPyXL Time Warnings	Lee los ficheros <i>master</i> , crea un diccionario (URL, UID), lee el fichero de <i>hits</i> y al mismo tiempo que va sustituyendo cada URL por su clave UID correspondiente, llena otro diccionario con las claves ( <i>uid2</i> , <i>uid1</i> , <i>hit_number</i> ) y su nº de <i>hits</i> . Luego en el fichero de <i>hits</i> rellena el valor que se corresponde con la coordenada ( <i>uid2</i> , <i>uid1</i> ) del diccionario.
4	Reemplazado por el 6	CSV SYS	Transforma el fichero CSV de <i>hits</i> original en un fichero CSV de tres columnas, separadas por comas y sin caracteres mal codificados.
5	Rechazado	CSV Mysql.connector Time	Conecta con <i>MySQL</i> , crea una nueva tabla en la base de datos <i>tfg</i> , con el nombre del fichero de <i>hits</i> en formato CSV que se le pasa como argumento e importa los datos desde el CSV a la base de datos.
6	Solución adoptada	CSV Mysql.connector SYS Time	Conecta con <i>MySQL</i> , extrae las claves UID de las tablas <i>master</i> , crea un diccionario de claves ( <i>uid</i> , <i>url</i> ) para cada tabla, lee el fichero CSV de <i>hits</i> y crea otro CSV normalizado separado por comas con 5 columnas, en las que inserta el contenido del fichero de <i>hits</i> y las claves UID del diccionario, que hacen <i>matching</i> con las URL del fichero de <i>hits</i> .
7	Reemplazado por el 13	CSV Mysql.connector Time	Conecta con la base de datos, crea una nueva tabla, con el nombre del archivo de <i>hits</i> en CSV que se le pasa como argumento e importa los datos desde el CSV a la base de datos, luego añade condición de claves primarias a <i>uid1</i> y <i>uid2</i> .
13	Solución adoptada	CSV Mysql.connector SYS Time	Realiza el mismo proceso que el <i>script</i> 7 y además añade la relación con las tablas <i>master</i> . Se importa como módulo en el <i>script</i> 6.

Tabla 1. Síntesis de *scripts* diseñados para automatizar la normalización, estructuración e importación de los datos del fichero de *hits* (Muestra 1) a la base de datos.



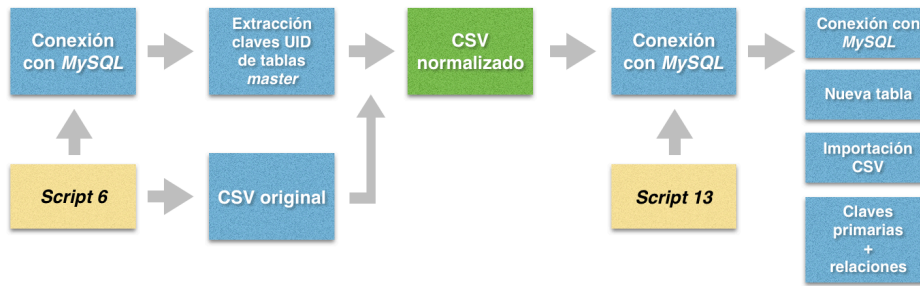


Figura 3.23. Esquema de la síntesis del proceso para el fichero de hits (Muestra 1).

### 3.1.4.2. Muestra 2

En la Tabla 2 se describen los scripts 8 a 12, utilizados para la normalización, estructuración e importación de los ficheros obtenidos de *Majestic*, incluyendo las librerías usadas para su desarrollo así como su estado en el diseño. Se incluye la Figura 3.24 en la que se ilustra el proceso de interrelación de estos scripts en el proceso.

Script	Estado	Librerías	Descripción
8	Solución adoptada	CSV SYS	Coge un fichero CSV de <i>Majestic</i> y lo transforma en un CSV normalizado separado por comas, sustituyendo los caracteres mal codificados por los que corresponda. Se importa como módulo en el <i>script</i> 9.
9	Solución adoptada	OS	Normaliza en bloque todos los ficheros CSV de <i>Majestic</i> , de un directorio dado.
10	Reemplazado por el 12	CSV Mysql.connector Time	Conecta con la base de datos, crea una nueva tabla con el nombre del fichero CSV de <i>Majestic</i> que se le pasa como argumento e importa los datos desde el CSV a la base de datos, luego inserta la columna DATE y la columna UID1 y añade la condición de clave primaria a UID1. Se importa como módulo en el <i>script</i> 11.
11	Solución adoptada	OS Time	Realiza la importación en bloque a <i>MySQL</i> todos los CSV de <i>Majestic</i> , desde un directorio dado.
12	Solución adoptada	CSV Mysql.connector Time	Realiza el mismo proceso que el <i>script</i> 10 y además añade las relaciones con la tabla <i>master_spa</i> a través de la clave primaria <i>uid1</i> . Se importa como módulo en el <i>script</i> 11.

Tabla 2. Síntesis de scripts diseñados para automatizar la normalización, estructuración e importación de los datos de los ficheros de *Majestic* a la base de datos.

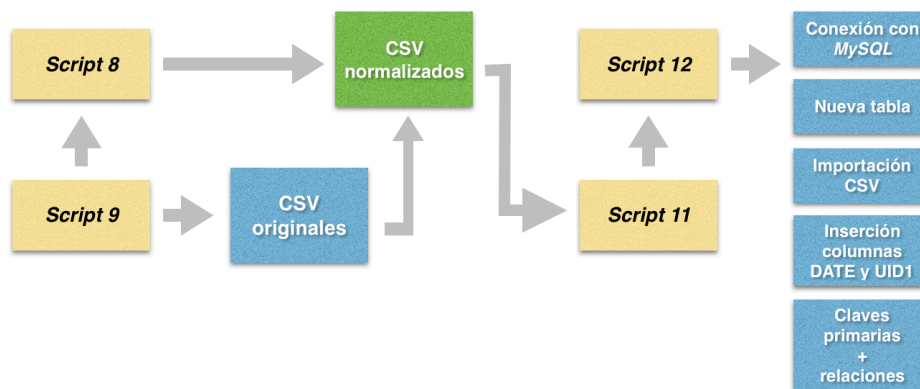


Figura 3.24. Esquema de la síntesis del proceso para los ficheros de *Majestic* (Muestra 2).

## 3.2. Análisis de los datos con R y RStudio

Es imposible, en este apartado, ahondar en detalles acerca de todas las posibilidades ofrecidas por *R* para realizar análisis estadísticos y gráficas, entre otros motivos, porque no forma parte de la finalidad de este trabajo. El propósito de este apartado es proporcionar una visión de cómo se puede incorporar *R* en los procesos de automatización, junto con el resto de tecnologías que aquí se utilizan. Para ello, se realiza un breve estudio estadístico descriptivo de una tabla de ejemplo de la Muestra 2 (en concreto la tabla del fichero *vino160404.csv*), como se ha indicado en el apartado 2.6.3 (página 36).

### 3.2.1. Citation Flow y Trust Flow

En base a las definiciones que se han hecho de estas métricas (véase apartado 2.4.2.3, página 23), un análisis interesante es compararlas realizando una correlación entre ambas (Figura 3.25), la cual permite establecer rápidamente la calidad general de las páginas web de las empresas españolas del sector vino, el objetivo sería acercarse lo máximo posible a 1, es decir, a la línea roja que divide la gráfica.

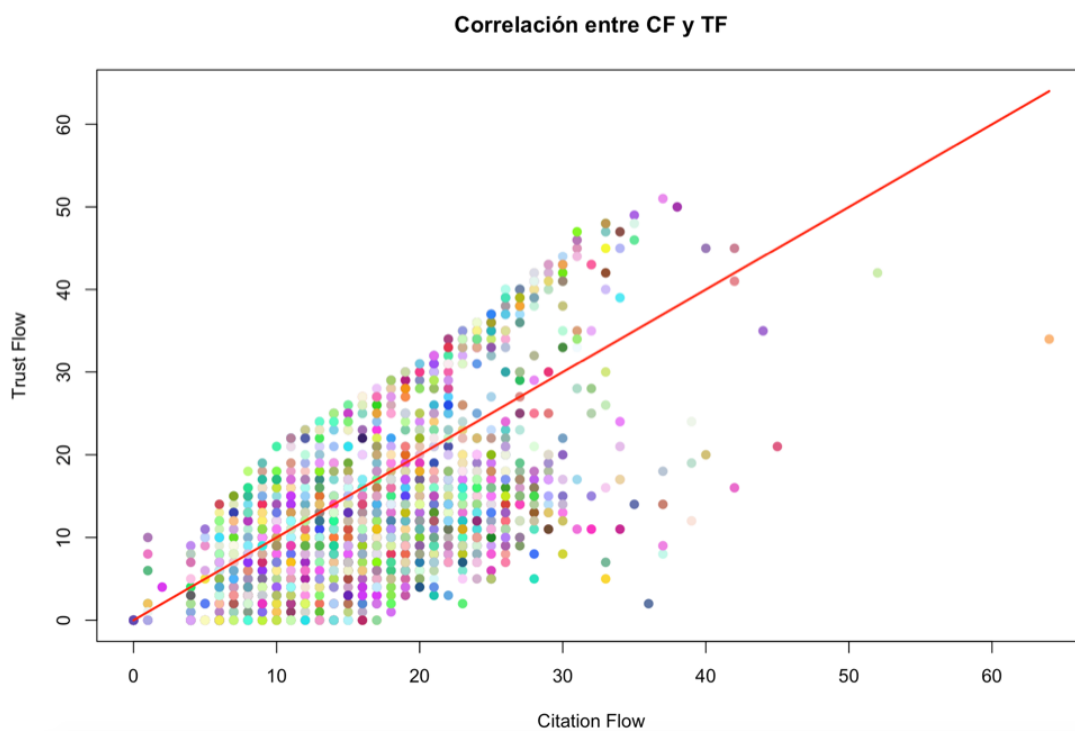


Figura 3.25. Gráfica de dispersión ratio Citation Flow/Trust Flow de la tabla *vino160404*.

Para confeccionar la gráfica de la figura anterior se ha ejecutado el *script 1* (véase Anexo II, pág. 96), el cual carga las librerías que se van a necesitar, conecta con la base de datos de *MySQL*, efectúa dos consultas en *SQL* para extraer el CF y el TF junto con el nombre de sus respectivas empresas (que saca de la tabla *master\_spa*, gracias a tener configurado como clave primaria el campo *uid1*), calcula el ratio CF/TF, dibuja la gráfica de dispersión del ratio y calcula los coeficiente de covarianza y de correlación, que muestra por pantalla.



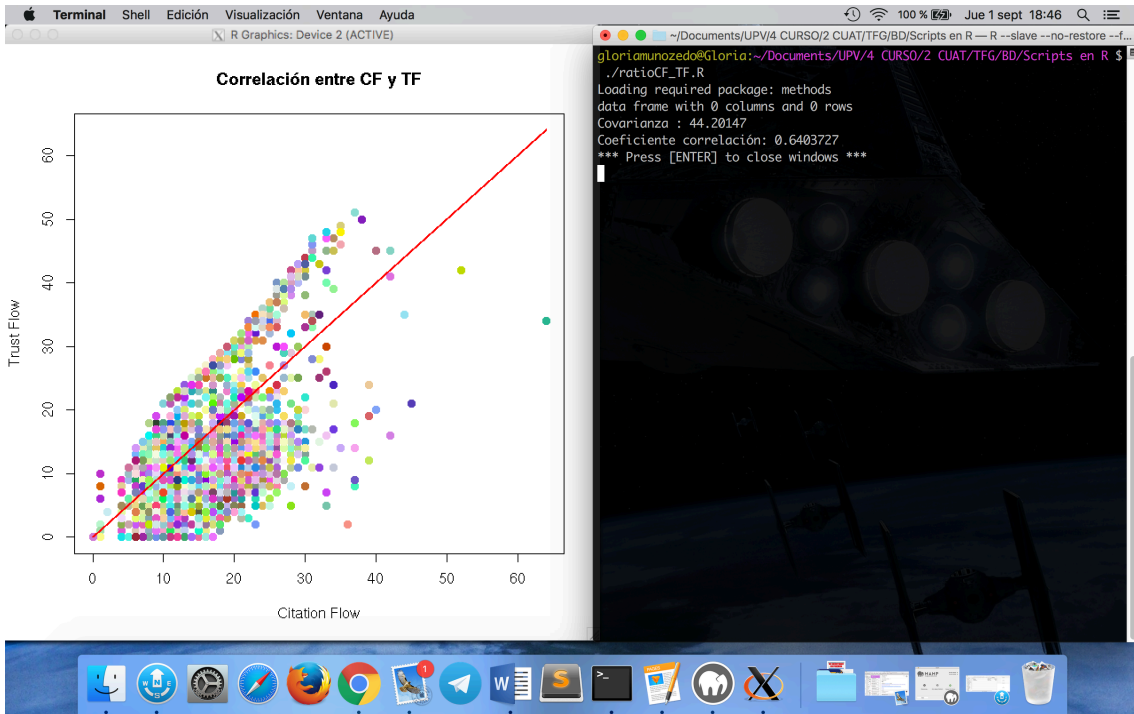


Figura 3.26. Detalle de la ejecución del script 1.

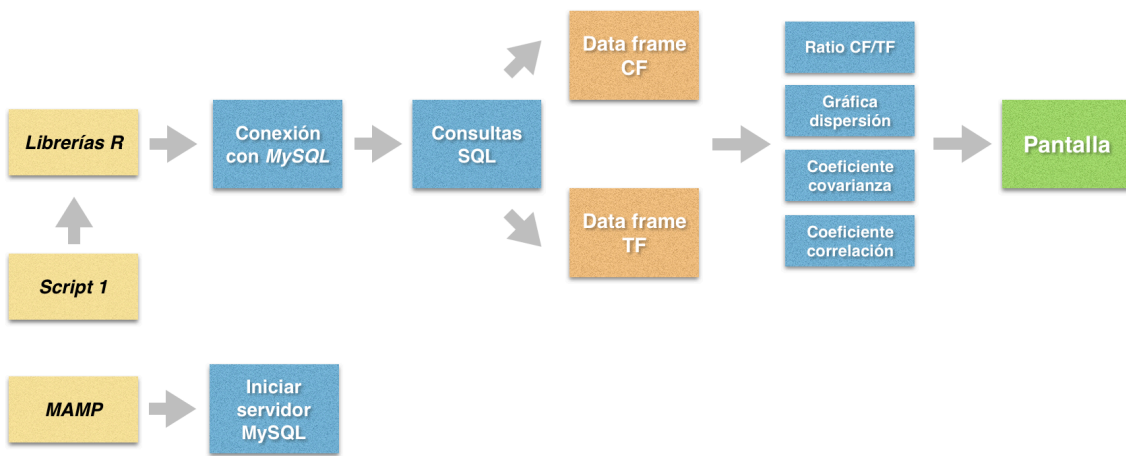


Figura 3.27. Esquema del proceso del script 1.

### 3.2.2. External Back Links

Para el ejemplo de análisis del grupo de indicadores EBL, se utiliza la misma tabla del ejemplo del apartado anterior (*vino160404*) y se implementa el *script 2* (véase Anexo II, pág. 98). Al ejecutarlo se obtiene la gráfica de la Figura 3.28.

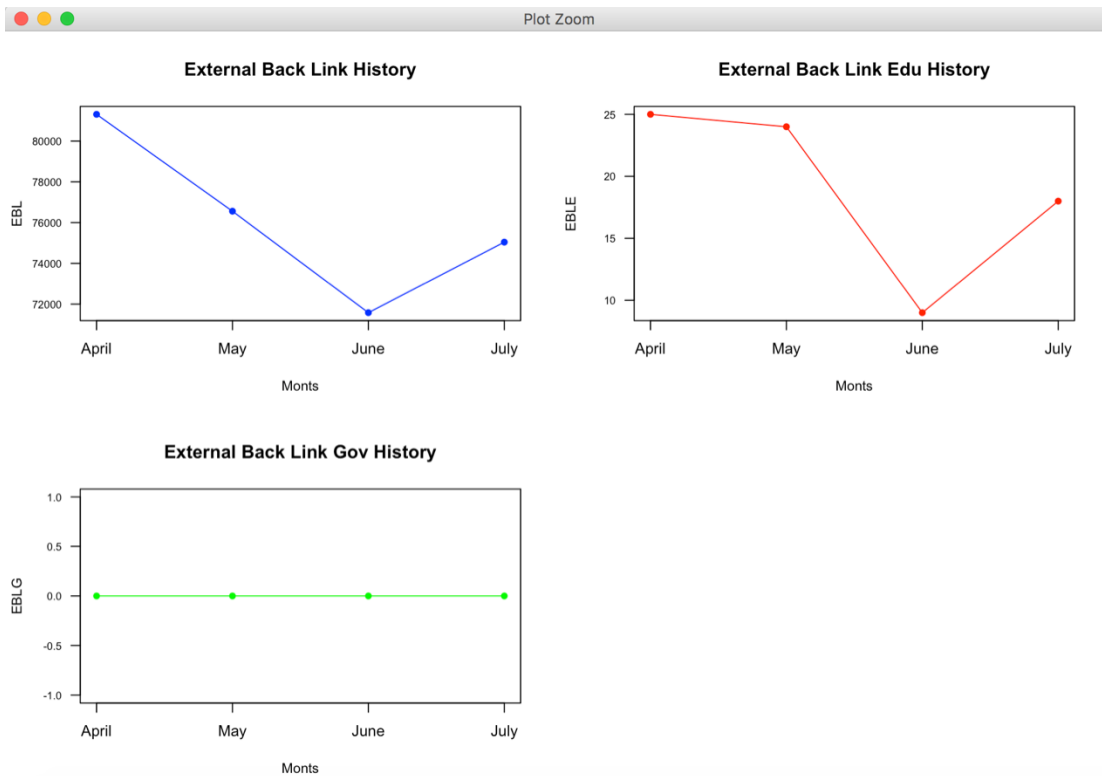


Figura 3.28. Gráfica series temporales de los indicadores EBL de la tabla vino160404.

Este *script* extrae datos de las cuatro tablas de indicadores de *Majestic* que se encuentran en la base de datos, a través de una consulta que se envía desde *R* en lenguaje *SQL*, y dibuja en la misma ventana tres gráficas de los indicadores EBL, EBLE y EBLG, las cuales permiten observar una comparativa de series temporales de cuatro meses (Figura 3.28). En el momento en el que se disponga de más datos se podrá efectuar un análisis más completo, incluso se podrán realizar predicciones de ciertos indicadores.

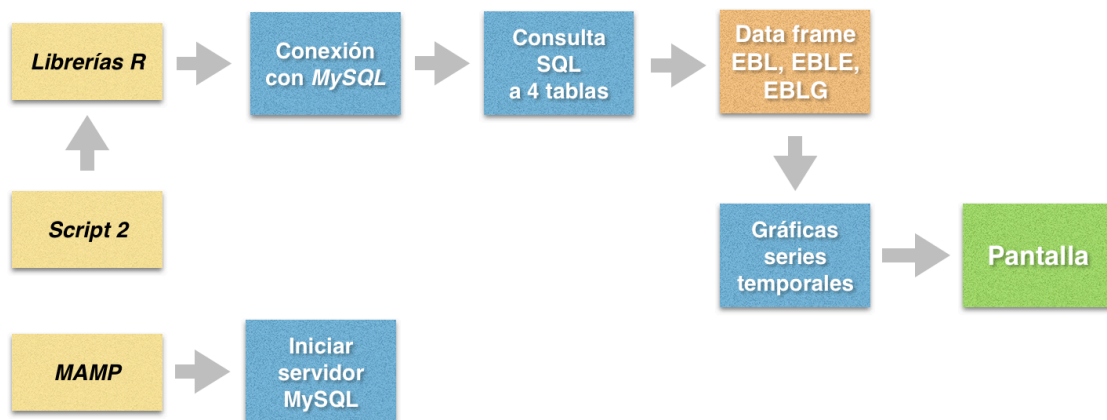


Figura 3.29. Esquema del proceso del script 2.

### 3.2.3. Gráficas

Con la finalidad de automatizar la obtención de varias gráficas en *R*, se ha implementado el *script* 3 (véase Anexo II, pág. 100). Al ejecutarlo, se visualiza en

pantalla el menú principal (Figura 3.30), en el que se puede seleccionar el indicador del cuál se quiere visualizar las gráficas. Después de seleccionar el indicador se muestra un segundo menú en el que se puede seleccionar la forma de visualización de los gráficos: por pantalla o guardarlos en local en formato PDF.

```

~/Documents/R — R --slave --no-restore --file=./graficas.R
gloriamunozedo@Gloria:~/Documents/R $ ./graficas.R
Loading required package: methods

-----
Select indicator to analyze :
-----

1 - Citation Flow
2 - External Back Links
3 - External Back Links Edu
4 - External Back Links Gov
5 - Topical Trust Flow
6 - Trust Flow

0 - Exit

> 1

-----
Select mode :
-----

1 - Display on screen
2 - Save in pdf

> 1

```

Figura 3.30. Detalle ejecución script 3. Menú principal.

Para mejor entendimiento se muestra un ejemplo, usando una vez más los datos de la tabla *vin0160404*; en el primer menú se selecciona "1 – Citation Flow" y en el segundo menú se selecciona "1 – Display on screen" y se obtiene el resultado que se muestra en la Figura 3.31.

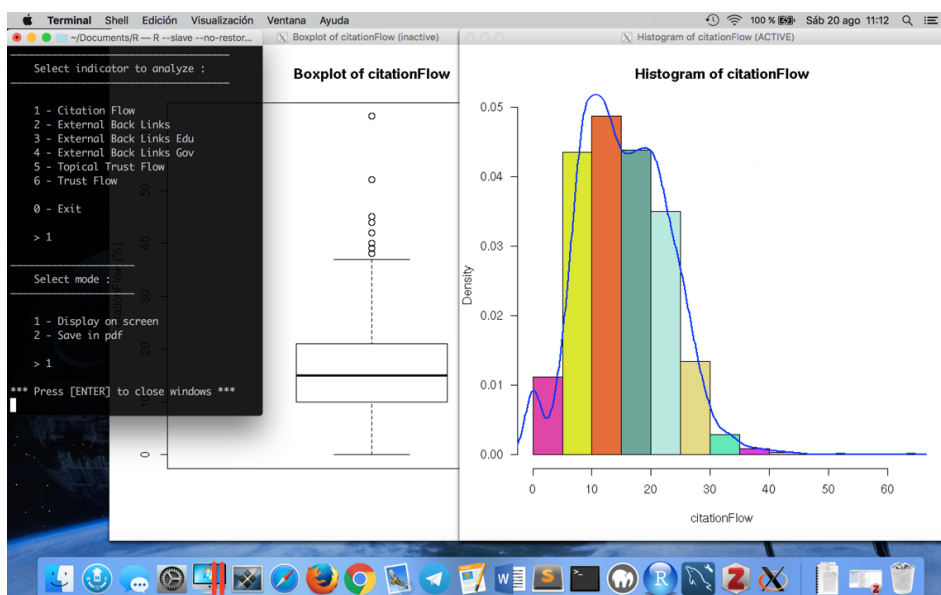


Figura 3.31. Resultado de la ejecución del script 3 para el indicador CF.

Para los indicadores CF, EBL, EBLE, EBLG y TF se ha decidido implementar la visualización de las gráficas del diagrama de caja (*boxplot*) y del histograma con la curva de densidad. Para el indicador TTF se ha decidido implementar la visualización

del diagrama de barras (*barplot*), mostrando 3 gráficas en la misma ventana (Figura 3.32), una para cada indicador de TTF que proporciona *Majestic*. En cada gráfica se han mostrado los 10 primeros *topics* correspondientes a cada indicador, en los que se han clasificado a las empresas españolas de la tabla de ejemplo de la muestra 2.

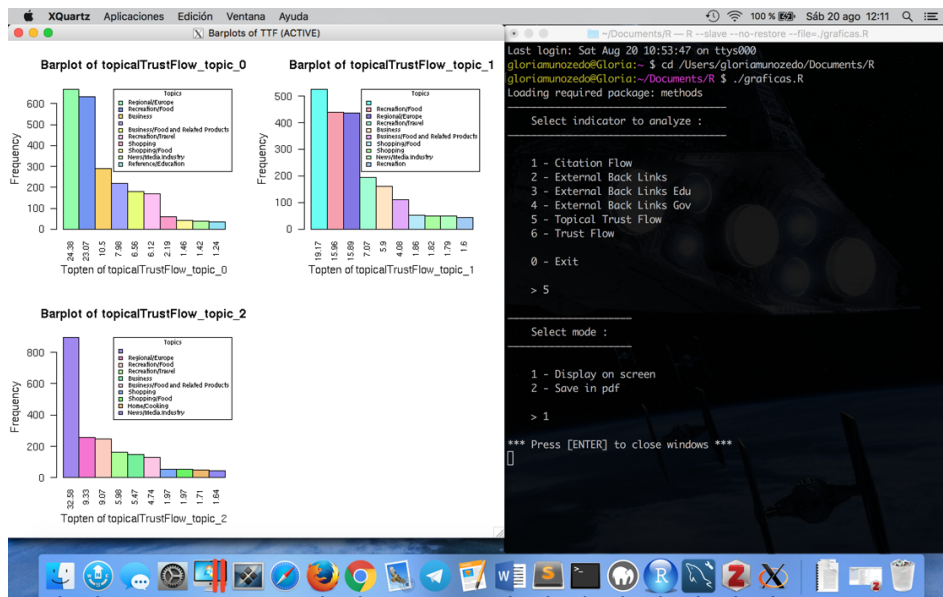


Figura 3.32. Resultado de la ejecución del script 3 para el indicador TTF.

Una vez que se han mostrado las gráficas solicitadas, la ejecución queda pausada hasta que se presione la tecla [ENTER]. Tras su pulsación, el programa retorna al menú principal. Para terminar la ejecución hay que seleccionar la opción "o – Exit".

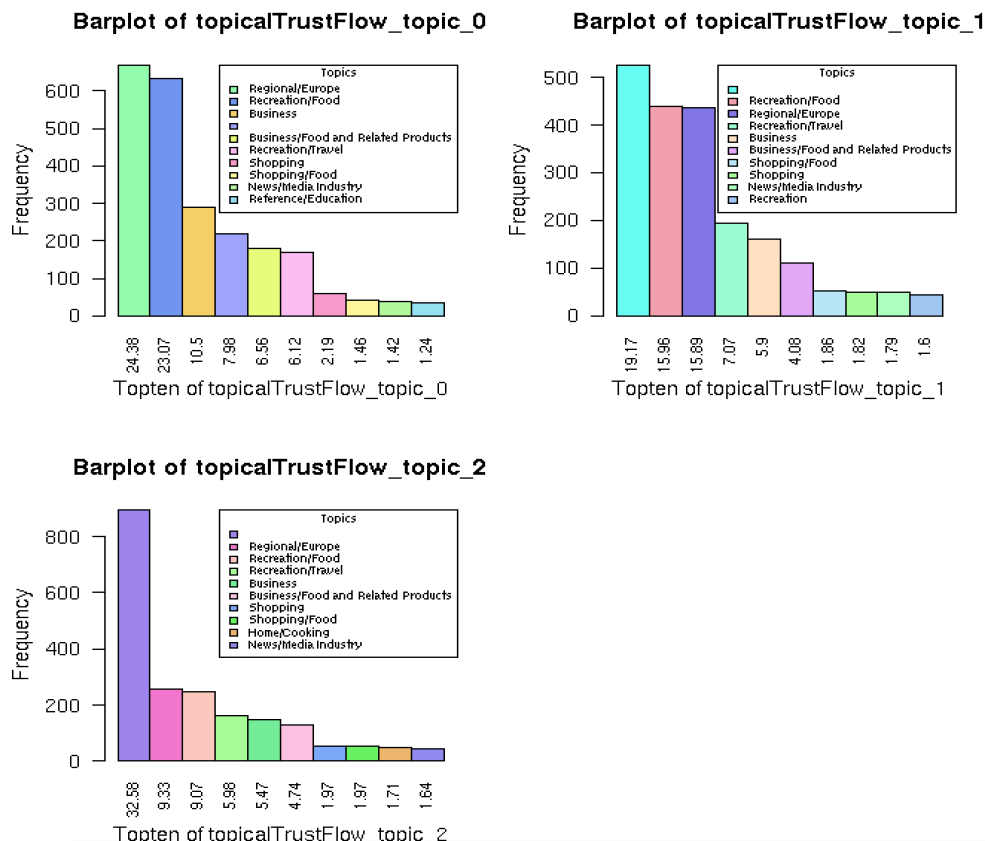


Figura 3.33. Detalle ampliado de los diagramas de barras para los indicadores TTF.

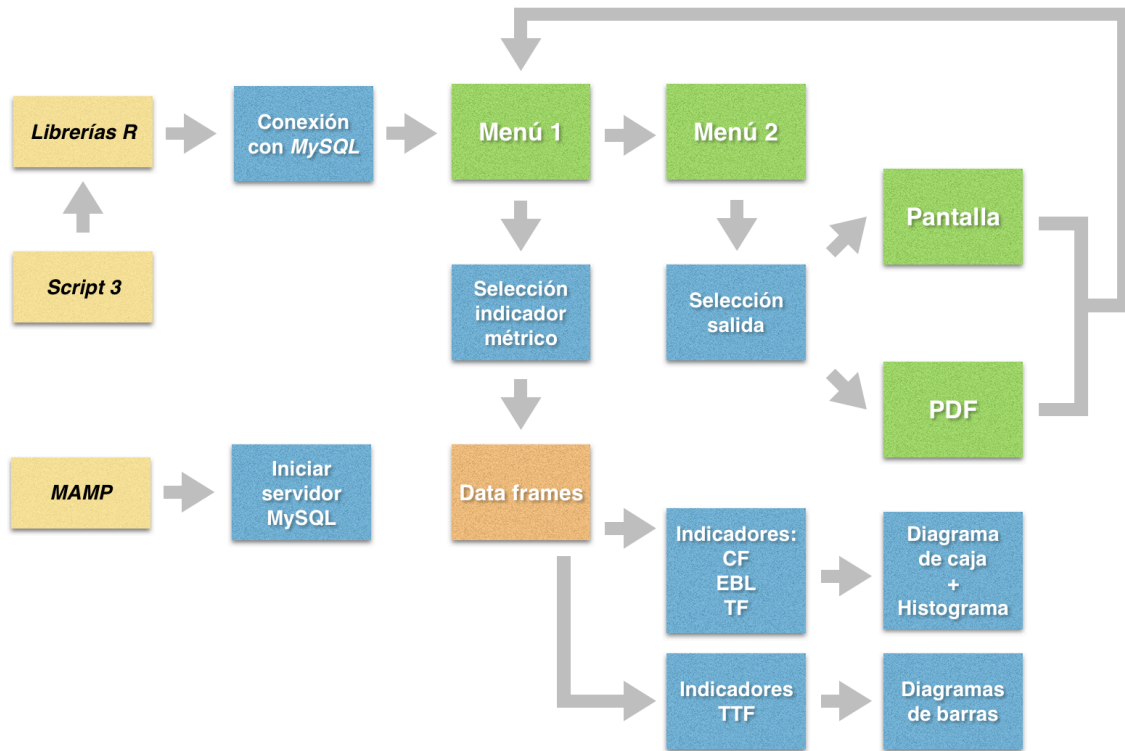


Figura 3.34. Esquema del proceso del script 3.

Como se ha podido comprobar, las posibilidades de realización de análisis con *R* son muy amplias y variadas. En este trabajo se han tomado un pequeño grupo de indicadores a modo de prueba, ya que se trata solamente de un trabajo exploratorio para ver las posibilidades técnicas. Para análisis más exhaustivos sería necesario definir qué indicadores se desean ver en cada estudio, qué análisis estadísticos se precisan y concretar las características y propiedades que se quieren analizar. Cada estudio tendrá unos requerimientos concretos, que como se ha demostrado, son ampliamente configurables. Con estos ejemplos se han comprobado las posibilidades de explotación gráfica que tiene *R*, una vez se tienen los datos originales almacenados en una base de datos.

## 4. Discusión

Durante el desarrollo del presente trabajo, han surgido varios temas de discusión sobre las soluciones adoptadas para llevar a cabo la realización de los objetivos y tareas específicas planteados en el apartado 1.3 y que pasan a detallarse seguidamente.

En primer lugar, el diseño que se utiliza para la clave primaria (véase apartado 3.1.1, pág. 38) consistente en una cadena de texto de ocho caracteres, podría resultar recomendable aumentarla a nueve caracteres, ya que hay empresas que tienen varias cuentas de *Facebook* o de *Twitter*, así los dos últimos caracteres indicarían la fuente y el número de orden dentro de la fuente.

Otra consideración sobre el diseño de la clave primaria que se plantea, sería la de utilizar directamente las URL de cada empresa; esto solucionaría el problema de cambio de fuente de extracción de los datos (distinta de *Majestic* o de *web scraping*), o el cambio de empresas para el caso de estudio (empresas españolas del sector vino); pudiéndose utilizar así para distintas fuentes de extracción de los datos, así como para diferentes sectores empresariales. Pero a su vez esta decisión plantea otros problemas, ya que las URL suelen ser demasiado largas para ser consideradas unas buenas claves primarias, a este problema hay que añadir los caracteres especiales que aparecen mal codificados en dichas URL.

En cuanto a la integración de claves UID en los ficheros *master*, se realiza haciendo uso de la hoja de cálculo EXCEL y, dado que lo que se pretende es automatizar las tareas al máximo, se propone mejorar la realización de esta tarea implementando un *script* en *Python*, que al igual que se hace con el fichero de *hits* y los ficheros de *Majestic*, introduzca una columna en las tablas, una vez que estén ya importadas a *MySQL*, con las claves correspondientes a cada empresa, generadas también en ese mismo *script*.

Referente al almacén de datos, se decide utilizar *MySQL* por estar considerado de entre 309 sistemas analizados, el mejor gestor de base de datos de código libre hasta la fecha (Agosto 2016), según el *ranking* de *DB-Engines* que se puede ver en la siguiente figura.

309 systems in ranking, August 2016

Rank			DBMS	Database Model	Score		
Aug 2016	Jul 2016	Aug 2015			Aug 2016	Jul 2016	Aug 2015
1.	1.	1.	Oracle	Relational DBMS	1427.72	-13.81	-25.30
2.	2.	2.	MySQL	Relational DBMS	1357.03	-6.25	+65.00
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1205.04	+12.16	+96.39
4.	4.	4.	MongoDB	Document store	318.49	+3.49	+23.84
5.	5.	5.	PostgreSQL	Relational DBMS	315.25	+4.10	+33.39
6.	6.	6.	DB2	Relational DBMS	185.89	+0.81	-15.35
7.	7.	8.	Cassandra	Wide column store	130.24	-0.47	+16.24
8.	8.	7.	Microsoft Access	Relational DBMS	124.05	-0.85	-20.15
9.	9.	9.	SQLite	Relational DBMS	109.86	+1.32	+4.04
10.	10.	10.	Redis	Key-value store	107.32	-0.71	+8.51

Figura 4.1. Top ten del ranking de gestores de bases de datos, Agosto 2016.  
Fuente: <http://db-engines.com/en/ranking>

Además de ser gratuito, *MySQL* es multiplataforma, multihilo, multiusuario, escalable, robusto, soporta gran cantidad de lenguajes de programación, posee una interfaz gráfica muy amigable que también es gratuita (*MySQL Workbench*), es el más utilizado de todos los gestores de bases de datos, con más de 10 millones de instalaciones<sup>22</sup>, posee gran cantidad de recursos en la Web, es fácil de aprender, de configurar y de instalar. Sin embargo, dado que se pretende manejar en un futuro grandes cantidades de datos, con muchas tablas y varias bases de datos a la vez, sería recomendable migrar a *Oracle* (puesto número 1 en el ranking), en el caso de que *MySQL* se quedara corto de recursos.

Por otra parte, también dentro del apartado de conexión con el servidor de *MySQL*, para su inicialización se está utilizando la aplicación *MAMP*, por resultar una forma sencilla y rápida de iniciar los servidores. Queda pendiente aquí investigar librerías en *Python* y en *R*, a través de las cuales se pueda realizar la inicialización de manera automatizada, incorporándola en los *scripts*.

En lo concerniente a la automatización del proceso de almacenaje de los datos, decir que es donde se encuentra el mayor problema de realización. Tanto en la importación de los datos del fichero de *hits*, como en la de los ficheros de *Majestic*, la problemática subyacente es la misma: la existencia de caracteres especiales mal codificados. Los motivos por los que estos ficheros contienen caracteres "raros" son variados y dependen de diversos factores (Cummings & Texin, 2011):

- Del editor con el que se haya hecho la página web. Si, por ejemplo, el archivo original estaba escrito en *ISO-8859-1* y se edita en *UTF-8*, se verán los caracteres especiales mal codificados. Si se guarda ese archivo tal y como está, se estará corrompiendo la codificación original (se guardará mal, en *UTF-8*) y viceversa.
- De la configuración de codificación que tenga el servidor en el que esté alojada la página web.
- De si hay un archivo oculto en el directorio raíz que sirva la página web (por ejemplo *.htaccess* de *Apache*).
- De si se ha especificado o no la codificación en las etiquetas *META* de la sección *HEAD* del *HTML*, o en la cabecera del *PHP*. Ejemplo:

```
<HEAD>
```

```
<meta name = "tipo_contenido" content = "text/html;"
```

- De la codificación elegida en la base de datos que contiene las tablas del *HTML*.
- Por último y no menos importante, depende de la codificación con la que se crean y guardan los archivos en destino.

Las codificaciones más frecuentes son *ISO-8859-1* y *UTF-8*. El problema de que se generen caracteres raros, reside en la mezcla y utilización de ambas codificaciones al

---

<sup>22</sup> Fuente <https://basededatosunounivia.wordpress.com/2015/03/13/oracle-vs-mysql-vs-sql-server-una-comparacion-entre-los-sistemas-gestores-de-bases-de-datos-relacionales-mas-populares/>.

mismo tiempo. Si por ejemplo se usa *ISO-8859-1* para la página web, el resto de puntos que se han comentado anteriormente deben estar en *ISO*. No puede haber mezcla de algunas partes en *ISO* y otras en *UTF-8*, porque entonces es cuando se produce la incongruencia, con el resultado de la generación de caracteres mal codificados en la extracción de datos Web (Gestiweb Integración De Soluciones Web S.L., 2012).

Lamentablemente, éste es uno de los grandes problemas de los desarrolladores web y se convierte en algo tremendamente problemático, sino se hace bien desde el principio. Cuando se realiza el *web scraping* es importante conocer la codificación o *charset* de la página web y la de su servidor, si se obvia este proceso se obtiene una extracción de datos corruptos, pero ésta no es una tarea trivial. Se podrían utilizar ciertas técnicas para detectar de la codificación, como por ejemplo extraer el *charset* de la cabecera del *PHP*, o el que está definido en la etiqueta *META*, hay varias librerías de *Python* que se podrían utilizar para este fin como *html5lib*<sup>23</sup>, *urllib2*<sup>24</sup> o *request*<sup>25</sup> entre otras, o hacer uso de un detector de codificación como por ejemplo *Beautiful Soup*<sup>26</sup> o *chardet*<sup>27</sup>, pero estas técnicas tampoco garantizan un éxito absoluto.

Comentar también acerca del proceso de importación de ficheros a la base de datos, que el campo fecha que se añade a cada tabla de los ficheros de *Majestic*, se extrae del nombre del fichero CSV; esto no es muy conveniente, puesto que si se cambia la forma de nombrado de los ficheros ya no se podrá sacar la fecha del nombre de estos. El procedimiento más adecuado sería que cuando se toman los datos de *Majestic*, se añada un campo (columna) en cada fichero, con la fecha en la que se realizó la extracción.

Con respecto a la intención de utilizar el lenguaje de programación R tanto para el proceso de automatización, como para el de análisis, comentar que no resulta recomendable. *R* tiene muchas cualidades como son:

- Es un *software* libre, de código abierto, multiplataforma y gratuito.
- Tiene una comunidad académica detrás que provee una muy buena documentación en línea.
- Continuamente aparecen nuevos paquetes gratuitos que expanden su capacidad para estimar o solucionar diferentes problemas
- Proporciona una plataforma inigualable para la programación de nuevos métodos estadísticos de una manera fácil y sencilla.
- Contiene rutinas estadísticas avanzadas no disponibles en otros paquetes.
- Tiene capacidades gráficas que permiten realizar gráficos detallados y de gran atractivo.
- Posee una interfaz gráfica muy amigable (*RStudio*).
- Se está convirtiendo en un estándar en la sociedad científica, por hacer figuras de calidad de publicación, además de poder exportarse a diferentes formatos incluidos PDF.

---

<sup>23</sup> Véase para más información <https://github.com/html5lib/html5lib-python>.

<sup>24</sup> Véase para más información <https://docs.python.org/2/library/urllib2.html>.

<sup>25</sup> Véase para más información <http://docs.python-requests.org/en/latest/user/quickstart/>.

<sup>26</sup> Véase para más información <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>.

<sup>27</sup> Véase para más información <https://pypi.python.org/pypi/chardet>.





- Sorprende que se encuentre en el puesto 12 del *ranking* (Junio 2016) de los lenguajes de programación más utilizados<sup>28</sup> (según *RedMonk*), dado que *R* no es un lenguaje genérico.

Sin embargo, como desventajas se pueden citar que es un lenguaje lento y difícil de programar, para quienes no estén familiarizados con entornos de programación, y que es exclusivamente un lenguaje orientado a programación estadística, por lo que no sirve para el tratamiento previo que precisan los ficheros CSV, ni tampoco para importar los datos a la base de datos, de una manera rápida y sencilla.

Por estas razones, se recomienda el uso de *R* exclusivamente para la parte de análisis estadístico, que incluye la extracción de datos a través de consultas en *SQL* a la base de datos, los análisis estadísticos descriptivos que se deseen realizar y la visualización de las gráficas. Para las tareas previas de automatización del tratamiento de los ficheros y su importación a *MySQL*, se recomienda el uso de *Python*, principalmente por su sencillez y rapidez de desarrollo, sobretodo en la implementación de *scripts*, además de ser de libre distribución, es multiparadigma (permite programación en varios estilos: orientada a objetos, estructural y funcional), tiene un gran soporte de integración con otros lenguajes y herramientas, posee infinidad de bibliotecas y soporta varias bases de datos. Como desventaja se puede decir, que al tratarse de un programa interpretado puede que sea algo más lento en ejecución que los compilados, pero dado que se usa para automatizar los procesos previos al análisis de los datos, no se considera que esta desventaja sea relevante.

Acerca de utilizar *R* para efectuar la consulta y extracción de datos de la base de datos, realizar análisis de indicadores métricos relevantes y visualizar gráficamente esos análisis, queda sobradamente demostrado que *R* y *RStudio* son herramientas muy potentes con las que se pueden obtener resultados muy buenos.

Por último, comentar que el sistema que se ha diseñado es independiente del sector (vino, moda, etc.) y del tipo de URL (*site*, *Facebook*, *Twitter*, etc.), por lo que sería posible extrapolar los resultados obtenidos a estos entornos, simplemente reconfigurando el formato de clave UID; sin embargo es dependiente de la fuente de extracción de los datos, ya que si los datos no provienen de *Majestic* no tendrán la misma estructura, y también es dependiente del tipo de fichero, puesto que el tratamiento de normalización (eliminación de caracteres mal codificados) e importación a *MySQL*, se ha diseñado considerando que los ficheros vienen en formato CSV.

---

<sup>28</sup> Véase para más información <http://redmonk.com/sogrady/2016/07/20/language-rankings-6-16/>.

## 5. Conclusiones

---

El principal objetivo de este proyecto, es desarrollar las bases para la automatización de la importación de datos extraídos de páginas web, a un almacén de datos y de la posterior extracción de datos de ese almacén para su análisis y visualización.

El proceso de automatización de la importación de los datos se ha podido realizar en parte, debido al problema de la aparición de caracteres mal codificados, esto ha hecho que fuera necesario un tratamiento previo de los ficheros. Dado que los caracteres no se han podido recodificar automáticamente, por tratarse de una acumulación de codificaciones erróneas imposibles de identificar, ha sido necesaria la identificación manual de cada carácter problemático, para su posterior tratamiento mediante un *script* en *Python*, que efectúa el reemplazamiento por el carácter correcto. Sin embargo, este procedimiento no es el óptimo, puesto que en nuevas muestras podrían aparecer otros caracteres mal codificados que aquí no se han contemplado. Lo ideal sería corregir la codificación en origen, es decir, en la fuente de la extracción, pero este problema es de difícil solución, ya que las páginas web de las que se obtienen los datos son de muy diversa procedencia y cada una presenta los datos de una manera distinta.

El resto del proceso de importación de las tablas a la base de datos, inserción de la clave primaria, definición de las relaciones entre las tablas y el de extracción, análisis de ciertos indicadores y visualización de los datos, se ha efectuado con éxito, utilizando *R* para la fase final de análisis y *Python* para las fases anteriores. Es preciso resaltar que queda aún mucho trabajo por hacer en el apartado de análisis, ya que realizando una selección y un preprocesado de la información se podrían aislar los datos extremos o anómalos (*outliers*) y así se podría efectuar un análisis estadístico más preciso. También sería interesante que ese análisis fuera más exhaustivo, teniendo en cuenta la naturaleza de los datos, ya que abriría paso a poder estudiar la existencia de patrones y realizar predicciones de ciertos indicadores que resultaran más relevantes, en vistas a la toma de decisiones empresariales a nivel estratégico. Pero estos temas ya son materia para otro proyecto.

También queda como sugerencia, en el momento en el que se disponga de un conjunto de muestras más amplio y recogido durante más tiempo, intentar combinar los datos del *web scraping* con los indicadores de *Majestic*, a fin de ver si las empresas que más menciones reciben son las que más enlaces reciben, la relación que existe entre el número de menciones y que la página web sea buena, o que el producto sea bueno pero la página sea mala, si existe alguna relación entre la calidad de la página web medida con los indicadores de *Majestic* y las menciones recibidas, ver como evolucionan el número de menciones y los indicadores métricos durante el tiempo, etc.

Para finalizar, decir que también existen otros objetivos subyacentes en la realización de este trabajo final de grado, los cuales son adquirir y mejorar ciertas



competencias generales y específicas<sup>29</sup>, que los alumnos de Grado en Ingeniería Informática, deben poseer al finalizar la titulación. En mi caso, este trabajo me ha ayudado a mejorar y adquirir las siguientes competencias:

G01. (G) Poseer y comprender conocimientos en su área de estudio que parten de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también aspectos que implican conocimientos procedentes de la vanguardia de dicho campo de estudio.

G02. (G) Saber aplicar sus conocimientos a su trabajo o vocación de una forma profesional y poseer las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

G03. (G) Desarrollar las habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

G04. (G) Razonar de manera abstracta, analítica y crítica, sabiendo elaborar y defender argumentos en su área de estudio y campo profesional.

G05. (G) Aprender de manera autónoma nuevos conocimientos y técnicas adecuados para la concepción, el desarrollo, la evaluación o la explotación de sistemas informáticos.

G06. (G) Localizar información relevante desde diferentes fuentes e investigar las novedades tecnológicas en su ámbito de trabajo y en áreas afines.

G07. (G) Comunicar de modo efectivo, a público especializado y no especializado, tanto por escrito como oralmente, conocimientos, procedimientos, informes y documentación técnica, resultados e ideas relacionadas con las TIC y, concretamente con la Informática, evaluando su impacto socioeconómico.

G09. (G) Saber describir las competencias y perfiles propios de su profesión.

G10. (G) Disponer de las habilidades sociales necesarias para el ejercicio adecuado de su profesión.

G12. (G) Capacidad de integrarse y trabajar eficientemente en equipos unidisciplinares así como de colaborar en un entorno multidisciplinar.

BO1. (E) Capacidad para la resolución de los problemas matemáticos que puedan plantearse en la ingeniería. Aptitud para aplicar los conocimientos sobre: álgebra lineal; cálculo diferencial e integral; métodos numéricos; algorítmica numérica; estadística y optimización.

BO4. (E) Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

---

<sup>29</sup> Véase [http://www.inf.upv.es/gradoII/Memoria\\_grado\\_II.pdf](http://www.inf.upv.es/gradoII/Memoria_grado_II.pdf)

CO1. (E) Capacidad para tener un conocimiento profundo de los principios fundamentales y modelos de la computación y saberlos aplicar para interpretar, seleccionar, valorar, modelar, y crear nuevos conceptos, teorías, usos y desarrollos tecnológicos relacionados con la informática.

CO3. (E) Capacidad para evaluar la complejidad computacional de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.

ES3. (E) Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.

ES4. (E) Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.

RO6. (E) Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.

R12. (E) Conocimiento y aplicación de las características, funcionalidades y estructura de las bases de datos, que permitan su adecuado uso, y el diseño y el análisis e implementación de aplicaciones basadas en ellos.

TG1. (E) Presentación y defensa ante un tribunal universitario de un ejercicio original a realizar individualmente, consistente en un proyecto en el ámbito de las tecnologías específicas de la Ingeniería en Informática de naturaleza profesional en el que se sinteticen e integren las competencias adquiridas en las enseñanzas.

## 6. Bibliografía

---

ALEGSA. (2016). *Diccionario de informática y tecnología*. Recuperado el 3 de Agosto de 2016, de <http://www.alegsa.com.ar>

Almind, T. C., & Ingwersen, P. (1997). Infometric Analyses on the World Wide Web: Methodological approaches to 'Webometrics'. *Journal of Documentation* , 53.

Arroyo, N., Ortega, J. L., Pareja, V., Prieto, J. A., & Aguillo, I. (2005). Cibermetría. Estado de la cuestión. En Digital.CSIC (Ed.), *9as Jornadas Españolas de Documentación, FESABID 2005*. Madrid.

Belinchón Monjas, Y. (2011). Minería de datos. *Departamento de Ingeniería Telemática. Escuela Politécnica Superior Universidad Carlos III de Madrid* .

Berrocal, J. L., Figuerola, C. G., & Zazo, A. F. (2004). *Cibermetría: nuevas técnicas de estudio aplicables al Web*. Gijón: TREA.

Björneborn, L., & Ingwersen, P. (2004). Toward a basic framework for webometrics. *Journal of the American Society for Information Science and Technology* , 55 (14), 1216-1227.

Blog sobre Bussiness Intelligence. (2016). *Data mining y minería web: aplicaciones de la minería de datos*. Recuperado el 20 de Julio de 2016, de Lantares Solutions: <http://www.lantares.com/blog/data-mining-y-mineria-web-aplicaciones-de-la-mineria-de-datos>

Cisco. (2016). *White paper: Cisco VNI Forecast and Methodology, 2015-2020*. Recuperado el 26 de Julio de 2016, de <http://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf>

Cummings, C. R., & Texin, T. (2011). *UTF-8 Encoding Debugging Chart*. Recuperado el 15 de Agosto de 2016, de I18nQA.com: <http://www.i18nqa.com/debug/utf8-debug.html>

Dürsteler, J. C. (2002). *Visualización de información: Una visita guiada*. Barcelona: Ediciones Gestión 2000.

De Gyves Camacho, F. M. (2009). *Web Mining: Fundamentos Básicos*. Informe Técnico, Universidad de Salamanca.

Duque, R. G. (2010). *Python para todos*. (Autoedición, Ed.) Recuperado el 5 de Agosto de 2016, de MundoGeek: <http://mundogeek.net/tutorial-python/>

Escuela de datos. (Abril de 2016). Recuperado el 20 de Julio de 2016, de Introducción a la extracción de datos de sitios web: scraping: <http://es.schoolofdata.org/introduccion-a-la-extraccion-de-datos-de-sitios-web-scraping/>

- Etzioni, O. (1996). The world wide web: Quagmire or gold mine. *Communications of the ACM* , 39 (11), 65-68.
- Fader, P. (13 de Junio de 2007). What Data Mining Can and Can't Do. *The Voice of the CIO Community*. (A. Alter, Entrevistador)
- Gestiweb Integración De Soluciones Web S.L. (2012). *Problemas html acentos y eñes: charset UTF-8 / ISO-8859-1*. Recuperado el 15 de Agosto de 2016, de Gestiweb: <https://www.gestiweb.com/?q=content/problemas-html-acentos-y-e%C3%B1es-charset-utf-8-iso-8859-1>
- González De Dios, J., & Aleixandre Benavent, R. (2007). Evaluación de la investigación en Biomedicina y Ciencias de la Salud: indicadores bibliométricos y cibernéricos. *Boletín de la Sociedad de Pediatría de Asturias, Cantabria, Castilla y León* , 47 (200).
- Guía de posicionamiento web. (2012). *Ayuntamiento de Plasencia*. Recuperado el 5 de Agosto de 2016, de <https://empresa.plasencia.es/imagenes/noticias/Guia%20posicionamiento%20web.pdf>
- Hasperué, W. (2014). Extracción de conocimiento en grandes bases de datos utilizando estrategias adaptativas. La Plata, Argentina: Universidad de La Plata.
- Hernández Orallo, J., Ramírez Quintana, M. J., & Ferri Ramírez, C. (2004). *Introducción a la Minería de Datos*. Pearson Prentice Hall.
- Ihaka, R., & Gentleman, R. (1996). R: a language for data analysis and graphics. *Journal of Computational and Graphical Statistics* (5), págs. 299–314.
- Kunder, M. d. (9 de Febrero de 2016). *The World Wide Web Size*. Recuperado el 26 de Julio de 2016, de <http://www.worldwidewebsite.com/>
- Liu, B. (2007). *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Springer Science & Business Media.
- Majestic-12 Ltd. (2016). *Majestic*. Recuperado el 22 de Julio de 2016, de Glosario: <https://es.majestic.com/support/glossary>
- Marcilla, J. (23 de Abril de 2016). *Métricas para hacer un buen SEO en 2016*. Recuperado el 14 de Agosto de 2016, de NinjaSEO: <http://ninjaseo.es/metricas-seo/>
- Molina Félix, L. C. (2002). Data mining: torturando a los datos hasta que confiesen. *Universitat Oberta de Catalunya*.
- MySQL. (2016). *Manual de MySQL Workbench de la distribución*. Recuperado el 1 de Agosto de 2016, de <http://dev.mysql.com/doc/>
- Neuman, A. (22 de Noviembre de 2013). *Vía Digital*. Recuperado el 5 de Agosto de 2016, de ¿Qué es el Web Scraping o Screen Scraping y por qué nos debe importar?: <https://vidadigital.com.pa/que-es-el-web-scraping-o-screen-scraping-y-por-que-nos-debe-importar/>
- Orduña Malea, E. (2012). Fuentes de enlaces web para análisis cibernéricos. *Anuario ThinkEPI* , 6, 276-280.

- Orduña-Malea, E., & Aguillo, I. F. (2014). *Cibermetría. Midiendo el espacio red*. UOC.
- Ortega, J. L., & Aguillo, I. F. (2009). Minería del uso de webs. *EPI - El profesional de la información* , 18 (1), 20-26.
- Ortega, J. L., & Aguillo, I. F. (2013). Network visualisation as a way to the web usage analysis. *Aslib Proceedings: New Information Perspectives* , 65 (1), 40-53.
- Padrón Torres, L. (1 de Enero de 2006). *monografias.com*. Recuperado el 8 de Agosto de 2016, de <http://www.monografias.com/trabajos31/almacenes-datos/almacenes-datos.shtml>
- Quiles, H. (8 de Marzo de 2013). *Majestic SEO, una herramienta de marketing online imprescindible*. Recuperado el 7 de Agosto de 2016, de TreceBits: <http://www.trecebits.com/2013/03/08/majestic-seo-una-herramienta-de-marketing-online-imprescindible/>
- Reyes, S. C., & Lobaina, M. R. (2007). Minería Web: un recurso insoslayable para el profesional de la información. *Revista cubana de los profesionales de la información y de la comunicación* , 16 (4).
- Romero, D., & Díaz, B. (9 de Abril de 2015). *Blogger3.0*. Recuperado el 7 de Agosto de 2016, de Majestic SEO: destripa el SEO de tu web paso a paso: [http://blogger3cero.com/majestic-seo/#Historic\\_Index\\_y\\_Fresh\\_Index](http://blogger3cero.com/majestic-seo/#Historic_Index_y_Fresh_Index)
- Ruipérez, J. A., & Malea, E. O. (2015). *Métricas web para sectores económicos: el caso de la moda y el vino*. Trademetrics Research Group - Universidad Politécnica de Valencia, Barcelona.
- Santillán, L. A., Ginestà, M. G., & Mora, Ó. P. (2007). Bases de datos en MySQL. *Curso online* . (U. OpenCourseWare, Ed.) UOC.
- Sullivan, D. (2001). *Document warehousing and text mining*. Wiley Computer Publishing.
- Vicente Cuervo, M. R., & López Menéndez, A. J. (2008). Métricas e Indicadores de la Sociedad de la Información: panorámica de la situación actual. *Estadística Española* , 50 (168), 273-320.
- Villate, J. (2005). *Glosario de informática Inglés-Español*. Recuperado el 20 de Julio de 2016, de <http://quark.fe.up.pt/orca/pub-es/glosario.html>
- We Are Social. (2015). Recuperado el 26 de Julio de 2016, de <http://wearesocial.com/special-reports/digital-in-2016>
- Wikipedia. (Julio de 2016). Recuperado el 2 de Agosto de 2016, de [https://es.wikipedia.org/wiki/Big\\_data](https://es.wikipedia.org/wiki/Big_data)
- Wikipedia. (25 de Mayo de 2016). (Fundación Wikimedia, Inc.) Recuperado el 20 de Julio de 2016, de <https://es.wikipedia.org/wiki/Applet>
- Wikipedia. (17 de Junio de 2016). Recuperado el 23 de Julio de 2016, de <https://es.wikipedia.org/wiki/Script>

Witten, I. H., & Frank, E. (2005). *Data Mining. Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers.

Xu, F., Kurz, D., Piskorski, J., & Shmeier, S. (2002). Term Extraction and Mining of Term Relations from Unrestricted Texts in the Financial Domain. *Business Information Systems, Proceedings of BIS*.



# Anexo I - Scripts en Python

---

## Script 1 – Rechazado

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

### Script que inicializa a 0 las celdas del fichero master_int

import openpyxl
import warnings

# Para que ignore los warnings en la ejecución
warnings.simplefilter("ignore")

# Abre fichero Excel
excel_doc = openpyxl.load_workbook('V2.VINO_master_int.xlsx', data_only=True)

# Acceso a las hojas del libro
hojas = excel_doc.get_sheet_names()

# Acceso a las celdas de la hoja principal
hoja = excel_doc.get_sheet_by_name(hojas[0])

# Inicialización hits a 0
seleccion = hoja['H2':'JVG1047']
print ("Modificando celdas ...")
for filas in seleccion:
    for columna in filas:
        columna.value = 0

# Guardar cambios en el fichero
print ("Guardando fichero ...")
excel_doc.save('V2.VINO_master_int.xlsx')
```

## Script 2 – Reemplazado por el 3

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

### Script que rellena dos columnas adicionales del fichero de hits
### con las claves UID que lee de los ficheros master, después
### rellena el fichero master_int con el número de nombramientos

import openpyxl
import warnings

warnings.simplefilter("ignore")

# Leyendo ficheros de excel
print ("Leyendo ficheros de excel ...")
vino_master_int = openpyxl.load_workbook('V2.VINO_master_int.xlsx',
                                         data_only=True)
print ("    V2.VINO_master_int.xlsx --- OK")
vino_master_spa = openpyxl.load_workbook('V2.VINO_master_Spa.xlsx',
                                         data_only=True)
print ("    V2.VINO_master_Spa.xlsx --- OK")
vino_hits = openpyxl.load_workbook('Wine_A_site-siteFinal.xlsx',
                                   data_only=True)
print ("    Wine_A_site-siteFinal.xlsx --- OK")

# Acceso a las hojas de los libros
hojas_int = vino_master_int.get_sheet_names()
hojas_spa = vino_master_spa.get_sheet_names()
hojas_hits = vino_hits.get_sheet_names()

# Acceso a las celdas de la hoja principal
hoja_int = vino_master_int.get_sheet_by_name(hojas_int[0])
hoja_spa = vino_master_spa.get_sheet_by_name(hojas_spa[0])
hoja_hits = vino_hits.get_sheet_by_name(hojas_hits[0])

# Diccionarios de claves
key_int = {} # {url_int: UID}
key_spa = {} # {url_spa: UID}
key_hits = {} # {(UID_int, UID_spa): hit_number}

# Rellena los diccionarios de claves UID
def llenar_diccio( hoja ):
    diccio = {}
    ultFila = hoja.get_highest_row()
    fila = 2
    while ( fila <= ultFila ):
        clave = hoja.cell(row = fila, column = 5).value # key = URL
        valor = hoja.cell(row = fila, column = 1).value # value = UID
        diccio[ clave ] = valor
        fila += 1
    return diccio

print ("Llenando diccionarios de claves ...")
key_int = llenar_diccio( hoja_int )
print ("    Diccionario de claves: UID int --- OK")
```

```

key_spa = llenar_diccio( hoja_spa )
print ( "      Diccionario de claves: UID spa --- OK")

# Rellena las columnas de hoja_hits con las claves UID
ultFila = hoja_hits.get_highest_row()
fila = 2
print ("Añadiendo claves UID al fichero Wine_A_site-siteFinal.xlsx ...")
while ( fila <= ultFila ):
    # Lee la URL para guardar su clave UID
    url2 = hoja_hits.cell(row = fila, column = 2).value
    url1 = hoja_hits.cell(row = fila, column = 4).value
    # al mismo tiempo lee el hit_number
    hit_number = hoja_hits.cell(row = fila, column = 5).value
    # Escribe la UID en la hoja
    uid2 = key_int[ url2 ]
    uid1 = key_spa[ url1 ]
    hoja_hits.cell(row = fila, column = 1).value = uid2
    hoja_hits.cell(row = fila, column = 3).value = uid1
    # al mismo tiempo, rellena el diccionario key_hits
    key_hits[ (uid2, uid1) ] = hit_number
    fila += 1
print ("OK")

# Guarda los cambios en el fichero de hits
print ("Guardando cambios en Wine_A_site-siteFinal.xlsx ...")
vino_hits.save('Wine_A_site-siteFinal.xlsx')
print ("OK")

# Rellena el fichero V2.VINO_master_int.xlsx con el número de nombramientos
ultFila = hoja_int.get_highest_row()
fila = 2
ultCol = hoja_int.get_highest_column()
columna = 8
print ("Llenando el fichero V2.VINO_master_int.xlsx con los hits_number ...")
while ( fila <= ultFila ):
    uid_int = hoja_int.cell(row = fila, column = 1).value
    while ( columna <= ultCol ):
        uid_spa = hoja_int.cell(row = 1, column = columna).value
        if ( key_hits.has_key((uid_int, uid_spa)) ):
            hit_number = key_hits[ (uid_int, uid_spa) ]
            if ( hit_number != 0 ):
                hoja_int.cell(row = fila, column = columna).value =
                    hit_number
                columna += 1
        fila += 1
print ("OK")

# Guarda los cambios en el fichero master
print ("Guardando cambios en V2.VINO_master_int.xlsx ...")
vino_master_int.save('V2.VINO_master_int.xlsx')
print ("OK")

```

## Script 3 – Rechazado

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

"""
    Lee los ficheros master, guarda en un diccionario para cada fichero las
    urls y sus claves uid, lee el fichero de hits y al mismo tiempo va
    sustituyendo cada url por su uid correspondiente en los diccionarios y
    crea otro diccionario con las claves uid y su nº de hits, luego va al
    fichero de hits y la coordenada (uid2, uid1) la rellena con el valor del
    hit del diccionario
"""

import openpyxl
import warnings
from time import time

warnings.simplefilter("ignore")

time_ini = time()

# Carga ficheros de Excel
print ("Cargando datos de ficheros Excel ...")
vino_master_int = openpyxl.load_workbook('V2.VINO_master_int.xlsx',
                                         data_only=True)
print ("    V2.VINO_master_int.xlsx --- OK")
vino_master_spa = openpyxl.load_workbook('V2.VINO_master_Spa.xlsx',
                                         data_only=True)
print ("    V2.VINO_master_Spa.xlsx --- OK")
vino_hits = openpyxl.load_workbook('Wine_A_site-siteFinal.xlsx',
                                   data_only=True)
print ("    Wine_A_site-siteFinal.xlsx --- OK")

# Acceso a las hojas de los libros
hojas_int = vino_master_int.get_sheet_names()
hojas_spa = vino_master_spa.get_sheet_names()
hojas_hits = vino_hits.get_sheet_names()

# Acceso a las celdas de las hojas principales
hoja_int = vino_master_int.get_sheet_by_name(hojas_int[0])
hoja_spa = vino_master_spa.get_sheet_by_name(hojas_spa[0])
hoja_hits = vino_hits.get_sheet_by_name(hojas_hits[0])

# Diccionarios de claves
key_int = {} # {url_int: uid2}
key_spa = {} # {url_spa: uid1}
key_hits = {} # {(uid2, uid1): hit_number}

# Ultima fila de cada hoja
ultFila_int = hoja_int.get_highest_row()
ultFila_spa = hoja_spa.get_highest_row()
ultFila_hits = hoja_hits.get_highest_row()
```

```

# Llenado de los diccionarios de claves UID
def llenar_diccio( hoja, ultFila ):
    diccio = {}
    fila = 2
    while ( fila <= ultFila ):
        clave = hoja.cell(row = fila, column = 5).value # key = URL
        valor = hoja.cell(row = fila, column = 1).value # value = UID
        diccio[ clave ] = valor
        fila += 1
    return diccio

print ("Llenando los diccionarios de claves ...")
key_int = llenar_diccio( hoja_int, ultFila_int )
print ("    Diccionario claves UID int --- OK")
key_spa = llenar_diccio( hoja_spa, ultFila_spa )
print ("    Diccionario claves UID spa --- OK")

# Modificación columnas con URL de hoja_hits por las claves UID (int y spa)
fila = 2
print ("Añadiendo las claves UID al fichero Wine_A_site-siteFinal.xlsx ...")
while ( fila <= ultFila_hits ):
    # Leo las urls para sacar sus claves UID
    url2 = hoja_hits.cell(row = fila, column = 1).value
    url1 = hoja_hits.cell(row = fila, column = 2).value
    # al mismo tiempo leo el hit_number que les corresponde
    hit_number = hoja_hits.cell(row = fila, column = 3).value
    # Guardo en la hoja la clave UID asociada a la url leída,
    # sustituyendo las celdas de la URL por sus UIDs
    uid2 = key_int[ url2 ]
    uid1 = key_spa[ url1 ]
    hoja_hits.cell(row = fila, column = 1).value = uid2
    hoja_hits.cell(row = fila, column = 2).value = uid1
    # al mismo tiempo voy llenando el diccionario de claves y num_hits
    key_hits[ (uid2, uid1) ] = int( hit_number )
    fila += 1
print ("OK")

# Guardar cambios en el fichero de hits
print ("Guardando cambios en un nuevo fichero wine_site_final.xlsx ...")
vino_hits.save('wine_site_final.xlsx')
print ("OK")

# Llenado del fichero V2.VINO_master_int.xlsx con el numero de hits
fila = 2
ultCol = hoja_int.get_highest_column()
print ("Llenando el fichero V2.VINO_master_int.xlsx con los hits extraídos
...")
while ( fila <= ultFila_int ):
    columna = 8
    uid_int = hoja_int.cell(row = fila, column = 1).value
    while ( columna <= ultCol ):
        uid_spa = hoja_int.cell(row = 1, column = columna).value
        #if ( key_hits.has_key((uid_int, uid_spa)) ):# en python 2
        if (uid_int, uid_spa) in key_hits: # en python 3
            hit_number = key_hits[ (uid_int, uid_spa) ]
            hoja_int.cell(row = fila, column = columna).value =
                hit_number
        else :
            hoja_int.cell(row = fila, column = columna).value = 0

```

```
        columna += 1
    fila += 1
print ("OK")

# Guardar cambios en el fichero vino_master
print ("Guardando cambios en fichero V2.VINO_master_int.xlsx ...")
vino_master_int.save('V2.VINO_master_int.xlsx')
print ("OK")

time_fin = time()
time_total = time_fin - time_ini

print ("Tiempo transcurrido {:.2f}".format(time_total))
```

## Script 4 – Reemplazado por el 6

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# Coge el fichero de hits en csv original y lo transforma en
# un csv normalizado separado por comas

import sys
import csv

def parseCSV( csv_in, csv_out ):

    table = [] # va a ser una lista de tuplas de 3 elementos (columnas)
    table.append( ('url2', 'url1', 'hit_number') ) # cabecera de la tabla

    print ("Leyendo el fichero csv original ...")
    # al abrir el archivo con "with" es mas rápido y además nos evitamos
    # el try y el close
    with open(csv_in, 'rU') as csvarchivo: # esto es lo mismo que
                                          csvarchivo = open(...)
        csvarchivo.readline() # para que se salte la cabecera
        entrada = csv.reader(csvarchivo, delimiter=';')
        print ("Normalizando csv ...")
        for reg in entrada: # Cada línea del fichero se muestra como una
                            # lista de (2) campos
            # necesito guardarlo todo como una lista de tuplas
            # elimino caracteres no deseados y troceo el string
            col = reg[0].replace('site:', '').replace(' ', '')
                    .split(' ')
            # monto la tabla
            table.append( (col[0], col[1], int(reg[1])) )
        csvsalida = open(csv_out, 'w')
        salida = csv.writer(csvsalida, delimiter=',')
        # Escribir todas las tuplas de una lista con writerows()
        salida.writerows(table)
        del salida
        csvsalida.close()
        print ("OK")

parseCSV( "Wine_A_site-siteFinal.csv", "wine_site_final.csv" )
```

## Script 5 – Rechazado

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# Crea una nueva tabla en la base de datos tfg, con el nombre del
# fichero de hits en formato csv que se le pasa como argumento
# y sube los datos desde el csv a la base de datos mysql

from time import time
import mysql.connector as mdb
import csv

def upcsv( path_file ):

    # Obtiene el nombre del archivo
    path = path_file.split('/')
    file_name = path[len(path)-1]
    csv = file_name[:-4].lower()

    CREATE_TABLE = ( "CREATE TABLE `tfg`.`" + "`" + csv + "` ("
                    + "`url2` text, "
                    + "`url1` text, "
                    + "`hit_number` int"
                    + ")" )

    try:

        time_ini = time()

        # Conexión a la base de datos
        cnx = mdb.connect(user='root', password='root',
                          host='localhost', port='8889', database='tfg')
        cursor = cnx.cursor()

        # Creación de la tabla
        cursor.fetchone()
        cursor.execute(CREATE_TABLE)

        print ("Creación tabla " + csv + " OK")

        # Importación de los datos
        load_data = "LOAD DATA INFILE '" + path_file + "' INTO TABLE " +
                    csv + " FIELDS TERMINATED BY ';' LINES TERMINATED
                    BY '\n' IGNORE 1 LINES"
        cursor.execute(load_data)
        cnx.commit()

        print ("Datos importados OK")

        print ("Insertando columna uid2 ...")
        # Insertando la columna uid2
        insert_column = "ALTER TABLE " + csv + " ADD `uid2` VARCHAR(10)
                        DEFAULT NULL FIRST"
```





```

cursor.execute(insert_column)
# Actualizando los datos de la columna uid2
update_column = ("UPDATE " + csv + " T "
                 "SET T.uid2 = (SELECT V.uid2 "
                 "FROM vino_master_int V "
                 "WHERE V.url = T.url2)")
cursor.execute(update_column)
cnx.commit()
print ("OK")

print ("Insertando columna uid1 ...")
# Insertando columna uid1
insert_column = "ALTER TABLE " + csv + " ADD `uid1` VARCHAR(10)
                DEFAULT NULL AFTER `url2`"
cursor.execute(insert_column)
# Actualizando los datos de la columna uid1
update_column = ("UPDATE " + csv + " T "
                 "SET T.uid1 = (SELECT V.uid1 "
                 "FROM vino_master_spa V "
                 "WHERE V.url = T.url1)")
cursor.execute(update_column)
cnx.commit()
print ("OK")

# Añade condición de clave primaria a uid1 y uid2
add_key = ("ALTER TABLE " + csv + " CHANGE COLUMN `uid2` `uid2`
           VARCHAR(10) CHARACTER SET 'utf8' NOT NULL , "
           "ADD PRIMARY KEY (`uid2`)")
cursor.execute(add_key)
cnx.commit()

add_key = ("ALTER TABLE " + csv + " CHANGE COLUMN `uid1` `uid1`
           VARCHAR(10) CHARACTER SET 'utf8' NOT NULL , "
           "ADD PRIMARY KEY (`uid1`)")
cursor.execute(add_key)
cnx.commit()
print ("Añadida condición de clave primaria a uid1 y uid2 OK")
# Cerrando conexiones
cursor.close()
cnx.close()

time_fin = time()
time_total = time_fin - time_ini
print("Tiempo transcurrido para la tabla " + csv + ": {0:.2f}
      segundos".format(time_total))
print ("-----")
print ("-----")

except mdb.Error as err:
    print ("Something went wrong: {}".format(err))

```

```

upcsv("/Users/gloriamunozedo/Documents/UPV/4 CURSO/2 CUAT/TFG/BD/7 - hits
      normalizado a mysql con 5 columnas/wine_site_final.csv")

```

## Script 6 – Solución adoptada

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# Conecta con mysql, extrae las claves UID de las tablas master,
# crea un diccionario de claves (uid, url) para cada tabla,
# lee el fichero CSV de hits y crea otro CSV normalizado separado por comas,
# con 5 columnas en las que inserta el contenido del fichero de hits
# y las claves UID del diccionario, que hacen matching con las URL
# del fichero de hits

import sys
import csv
from time import time
import mysql.connector as mdb
#import script_7 as do
import script_13 as do

key_int = {}
key_spa = {}

def getKeys():
    try:
        print ("Obteniendo las claves de las tablas master de MySQL
                ...")
        # Conexión con la base de datos
        cnx = mdb.connect(user='root', password='root',
                          host='localhost', port='8889', database='tfg')
        cursor = cnx.cursor()

        # Consultas para obtener las claves y diccionarios
        query_spa = "SELECT uid1, url FROM vino_master_spa"
        cursor.execute(query_spa)
        for (uid1, url) in cursor:
            key_spa[url] = uid1

        query_int = "SELECT uid2, url FROM vino_master_int"
        cursor.execute(query_int)
        for (uid2, url) in cursor:
            key_int[url] = uid2

        cursor.close()
        cnx.close()

        print ("OK")

    except mdb.Error as err:
        print ("Something went wrong: {}".format(err))

def normalCSV( csv_in, csv_out ):

    table = [] # va a ser una lista de tuplas de 5 elementos (las 5
               # columnas: uid2, url2, uid1, url1, hit_number)
```

```

table.append( ('uid2', 'url2', 'uid1', 'url1', 'hit_number') )

print ("Leyendo el fichero csv original ...")
with open(csv_in, 'rU') as csvarchivo:
    csvarchivo.readline() # para que se salte la cabecera
    entrada = csv.reader(csvarchivo, delimiter=',')
    print ("Normalizando csv ...")
    for reg in entrada: # Cada línea del fichero se muestra como un
                        # array de 1 campo
        # necesito guardarlo todo como una lista de tuplas
        # elimino caracteres no deseados y troceo el string
        col = reg[0].replace('.com "', '.com')
                .replace('site:"', '')
                .replace('""', '')
                .replace('Ø', 'ñ')
                .replace('; ', ' ').split()

        url2 = col[0]
        url1 = col[1]
        hit_number = int(col[2])
        # Busco el matching de las url con las claves uid
        uid2 = key_int[url2]
        uid1 = key_spa[url1]
        # monto la tabla
        table.append( (uid2, url2, uid1, url1, hit_number) )
csvsalida = open(csv_out, 'w')
salida = csv.writer(csvsalida, delimiter=',')
# Escribe todas las tuplas de la lista
salida.writerows(table)
del salida
csvsalida.close()
print ("OK")

time_ini = time()

getKeys()
path_in = "/Users/gloriamunozedo/Documents/UPV/4 CURSO/2
          CUAT/TFG/BD/originales/Wine_A_site-siteFinal.csv"
path_out = "/Users/gloriamunozedo/Documents/UPV/4 CURSO/2
           CUAT/TFG/BD/normalizados/wine_site_final.csv"
normalCSV(path_in, path_out)
do.upcsv(path_out)

time_fin = time()
time_total = time_fin - time_ini

print ("Tiempo TOTAL transcurrido {0:.2f}".format(time_total))

```

## Script 7 – Reemplazado por el 13

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# Crea una nueva tabla en la base de datos tfg, con el nombre del
# archivo csv que se le pasa como argumento y sube los datos desde
# el csv a la base de datos mysql, luego añade condición de claves
# primarias a uid1 y uid2

from time import time
import mysql.connector as mdb
import csv

def upcsv( path_file ):

    # Obtiene el nombre del archivo
    path = path_file.split('/')
    file_name = path[len(path)-1]
    csv = file_name[:-4].lower()

    CREATE_TABLE = ( "CREATE TABLE `tfg`." + "`" + csv + "` ("
                    + "`uid2` VARCHAR(10), "
                    + "`url2` VARCHAR(1000), "
                    + "`uid1` VARCHAR(10), "
                    + "`url1` VARCHAR(1000), "
                    + "`hit_number` int"
                    + ")" )

    try:

        time_ini = time()

        # Conexión a la base de datos
        cnx = mdb.connect(user='root', password='root',
                          host='localhost', port='8889', database='tfg')
        cursor = cnx.cursor()

        # Creación de la tabla
        cursor.fetchone()
        cursor.execute(CREATE_TABLE)

        print ("Creación tabla " + csv + " OK")

        # Importación de los datos
        load_data = "LOAD DATA INFILE '" + path_file + "' INTO TABLE " +
                    csv + " FIELDS TERMINATED BY ',' LINES TERMINATED
                    BY '\n' IGNORE 1 LINES"
        cursor.execute(load_data)
        cnx.commit()

        print ("Datos importados OK")

        # Añade condición de clave primaria a uid1 y uid2
        add_keys = ("ALTER TABLE " + csv + " "
```

```

        "CHANGE COLUMN `uid2` `uid2` VARCHAR(10) CHARACTER
            SET 'utf8' NOT NULL , "
        "CHANGE COLUMN `uid1` `uid1` VARCHAR(10) CHARACTER
            SET 'utf8' NOT NULL , "
        "ADD PRIMARY KEY (`uid2`, `uid1`)"
    cursor.execute(add_keys)
    cnx.commit()

    # Cerrando conexiones
    cursor.close()
    cnx.close()

    time_fin = time()
    time_total = time_fin - time_ini
    print("Tiempo transcurrido para la tabla " + csv + ": {0:.2f}
        segundos".format(time_total))
    print ("-----")
        -----")

except mdb.Error as err:
    print ("Something went wrong: {}".format(err))

upcsv("/Users/gloriamunozedo/Documents/UPV/4 CURSO/2 CUAT/TFG/BD/8 - csv hits
normal con claves a mysql/wine_site_final.csv")

```

## Script 8 – Solución adoptada

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# Coge el fichero CSV de Majestic y lo transforma en un csv normalizado
# separado por comas, sustituyendo los caracteres mal codificados
# por los que corresponda
# Se importa como módulo en el script 9

import sys
import csv

def cleanCSV( csv_in, csv_out ):

    table = []

    print ("Leyendo el fichero csv original ...")
    with open(csv_in, 'rU') as csvarchivo:
        entrada = csv.reader(csvarchivo, delimiter=',')

        print ("Normalizando csv ...")
        for fila in entrada: # cada reg es una lista de 44 campos
                                (columnas)

            linea = []
            for campo in fila:
                linea.append(campo.replace('ï¿½', 'ñ')
                    .replace('Ã±', 'ñ')
                    .replace('Ã ', 'à'))
            table.append(linea)

        csvsalida = open(csv_out, 'w')
        salida = csv.writer(csvsalida, delimiter=',')

        # Escribe todas las tuplas de una lista
        salida.writerows(table)
        del salida
        csvsalida.close()
        print ("OK")
```



## Script 9 – Solución adoptada

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# Normaliza todos los csv, de un directorio dado

import os
import script_8 as do

# Variable para la ruta al directorio
path_in = '/Users/gloriamunozedo/Documents/UPV/4 CURSO/2
          CUAT/TFG/BD/originales/majestic vino/'
path_out = '/Users/gloriamunozedo/Documents/UPV/4 CURSO/2
           CUAT/TFG/BD/normalizados/majestic vino/'

# Lista con todos los ficheros del directorio
aux = os.walk(path_in)

# Recorre todos los ficheros del directorio,
# invocando para cada uno el método cleanCSV del modulo clean_csv,
# que corrige los caracteres extraños del texto
for root, dirs, files in aux:
    for csv in files:
        do.cleanCSV( path_in + csv, path_out + csv )
```

## Script 10 – Reemplazado por el 12

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# Este script se importa como módulo en el script 11,
# crea una nueva tabla en la base de datos tfg, con el nombre del
# archivo csv que se le pasa como argumento y sube los datos desde
# el csv a la base de datos mysql, luego inserta la columna DATE
# y la columna UID1 y añade la condición de clave primaria a UID1

from time import time
import mysql.connector as mdb
import csv

def upcsv( path_file ):

    # Obtiene el nombre del fichero
    path = path_file.split('/')
    file_name = path[len(path)-1]
    table_name = file_name[:-4].lower()

    # Fecha con el formato YYYYMMDD, extraída del nombre del fichero
    date = "20" + table_name[-6:]

    CREATE_TABLE = ( "CREATE TABLE `tfg`." + "`" + table_name + "` ("
        "`redirectFlag` text, "
        "`trustFlow` int, "
        "`itemType` int, "
        "`crawledFlag` text, "
        "`topicalTrustFlow_topic_1` varchar(300), "
        "`topicalTrustFlow_topic_0` varchar(300), "
        "`topicalTrustFlow_topic_2` varchar(300), "
        "`acrank` int, "
        "`finalRedirectResult` text, "
        "`analysisResUnitsCost` int, "
        "`lastSeen` text, "
        "`refDomains` int, "
        "`lastCrawlDate` text, "
        "`refSubNets` int, "
        "`resultCode` text, "
        "`outLinksInternal` int, "
        "`web` varchar(300), "
        "`title` text, "
        "`outDomainExternal` int, "
        "`indexedUrls` int, "
        "`refIPS` int, "
        "`lastCrawlResult` text, "
        "`refDomainsEdu` int, "
        "`itemNum` int, "
        "`extBacklinksGov` int, "
        "`status` text, "
        "`trustMetric` int, "
        "`extBacklinksGov_exact` int, "
        "`downloadBacklinks` int, "
```



```

        "`getTopBackLinks` int, "
        "`refDomainsGov_exact` int, "
        "`extBacklinksEdu` int, "
        "`redirectTo` varchar(300), "
        "`downloadRef` int, "
        "`topicalTrustFlow_value_2` int, "
        "`topicalTrustFlow_value_1` int, "
        "`refDomainsGov` int, "
        "`url` varchar(300), "
        "`topicalTrustFlow_value_0` int, "
        "`refDomainsEdu_exact` int, "
        "`outLinksExternal` int, "
        "`extBacklinks` int, "
        "`extBacklinksEdu_exact` int, "
        "`citationFlow` int"
    )"
)

try:

    time_ini = time()

    # Conexión con la base de datos
    cnx = mdb.connect(user = 'root', password = 'root', host = 'localhost',
                      port = '8889', database = 'tfg')
    cursor = cnx.cursor()

    # Creación de la tabla
    cursor.fetchone()
    cursor.execute(CREATE_TABLE)

    print ("Creación tabla " + table_name + " OK")

    # Importación de los datos
    load_data = "LOAD DATA INFILE '" + path_file + "' INTO TABLE " +
                table_name + " FIELDS TERMINATED BY ','
                LINES TERMINATED BY '\n'
                IGNORE 1 LINES"
    cursor.execute(load_data)

    print ("Datos importados OK")

    print ("Insertando columnas fecha y clave ...")
    # Inserta la columna fecha delante de las demás
    insert_column = "ALTER TABLE " + table_name + " ADD `date` DATE NOT NULL
                    DEFAULT %s FIRST" %date
    cursor.execute(insert_column)
    cnx.commit()

    # Inserta columna uid1 delante de todas
    insert_column = "ALTER TABLE " + table_name + " ADD `uid1` VARCHAR(10)
                    DEFAULT NULL FIRST"
    cursor.execute(insert_column)
    # Actualiza columna uid1 con las claves que saca de la tabla master
    update_column = ("UPDATE " + table_name + " T "
                    "SET T.uid1 = (SELECT V.uid1 "
                    "FROM vino_master_spa V "
                    "WHERE V.url = T.web)")
    cursor.execute(update_column)

```

```

cnx.commit()
print ("OK")

# Añade la condición de clave primaria a la clave uid1
add_key = ("ALTER TABLE " + table_name + " CHANGE COLUMN `uid1` `uid1`
          VARCHAR(10) CHARACTER SET 'utf8' NOT NULL , "
          "ADD PRIMARY KEY (`uid1`)")
cursor.execute(add_key)
print("Añadida condición de clave primaria OK")

# Cerrando conexiones
cnx.commit()
cursor.close()
cnx.close()

time_fin = time()
time_total = time_fin - time_ini
print("Tiempo transcurrido para la tabla " + table_name + ": {0:.2f}
      segundos".format(time_total))
print ("-----
      -")

except mdb.Error as err:
    print ("Something went wrong: {}".format(err))

```

## Script 11 – Solución adoptada

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# Sube a MySQL todos los csv de Majestic desde un directorio dado

import os
#import script_10 as do
import script_12 as do
from time import time

time_ini = time()

# Variable para la ruta al directorio
path = '/Users/gloriamunozedo/Documents/UPV/4 CURSO/2
        CUAT/TFG/BD/normalizados/majestic vino/'

# Lista con todos los ficheros del directorio
aux = os.walk(path)

# Recorre todos los ficheros del directorio,
# invocando para cada uno el método upcsv del modulo csvTOMysql,
# que sube los datos de los csv a MySQL, creando una tabla por cada fichero
for root, dirs, files in aux:
    for csv in files:
        do.upcsv( path + csv )

time_fin = time()
time_total = time_fin - time_ini
print("Tiempo TOTAL transcurrido {0:.2f}".format(time_total))
```

## Script 12 – Solución adoptada

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# Este script se importa como módulo en el script_11,
# crea una nueva tabla en la base de datos tfg, con el nombre del
# archivo csv de Majestic que se le pasa como argumento y sube
# los datos desde el csv a la base de datos mysql, luego inserta
# la columna DATE y la columna UID1 y añade la condición de
# clave primaria a UID1. Por último añade las relaciones con
# la tabla master_spa a través de la clave primaria uid1

from time import time
import mysql.connector as mdb
import csv

def upcsv( path_file ):

    # Obtiene el nombre del fichero
    path = path_file.split('/')
    file_name = path[len(path)-1]
    table_name = file_name[:-4].lower()

    # Fecha con el formato YYYYMMDD, extraída del nombre del fichero
    date = "20" + table_name[-6:]

    CREATE_TABLE = ( "CREATE TABLE `tfg`.`" + "`" + table_name + "` ("
                    + "`redirectFlag` text, "
                    + "`trustFlow` int, "
                    + "`itemType` int, "
                    + "`crawledFlag` text, "
                    + "`topicalTrustFlow_topic_1` varchar(300), "
                    + "`topicalTrustFlow_topic_0` varchar(300), "
                    + "`topicalTrustFlow_topic_2` varchar(300), "
                    + "`acrank` int, "
                    + "`finalRedirectResult` text, "
                    + "`analysisResUnitsCost` int, "
                    + "`lastSeen` text, "
                    + "`refDomains` int, "
                    + "`lastCrawlDate` text, "
                    + "`refSubNets` int, "
                    + "`resultCode` text, "
                    + "`outLinksInternal` int, "
                    + "`web` varchar(300), "
                    + "`title` text, "
                    + "`outDomainExternal` int, "
                    + "`indexedUrls` int, "
                    + "`refIPS` int, "
                    + "`lastCrawResult` text, "
                    + "`refDomainsEdu` int, "
                    + "`itemNum` int, "
                    + "`extBacklinksGov` int, "
                    + "`status` text, "
                    + "`trustMetric` int, "
                    + "`extBacklinksGov_exact` int, "
```

```

        "`downloadBacklinks` int, "
        "`getTopBackLinks` int, "
        "`refDomainsGov_exact` int, "
        "`extBacklinksEdu` int, "
        "`redirectTo` varchar(300), "
        "`downloadRef` int, "
        "`topicalTrustFlow_value_2` int, "
        "`topicalTrustFlow_value_1` int, "
        "`refDomainsGov` int, "
        "`url` varchar(300), "
        "`topicalTrustFlow_value_0` int, "
        "`refDomainsEdu_exact` int, "
        "`outLinksExternal` int, "
        "`extBacklinks` int, "
        "`extBacklinksEdu_exact` int, "
        "`citationFlow` int"
    )"
)

try:

    time_ini = time()

    # Conexión con la base de datos
    cnx = mdb.connect(user='root', password='root',
                      host='localhost', port='8889', database='tfg')
    cursor = cnx.cursor()

    # Creación de la tabla
    cursor.fetchone()
    cursor.execute(CREATE_TABLE)

    print ("Creación tabla " + table_name + " OK")

    # Importación de los datos
    load_data = "LOAD DATA INFILE '" + path_file + "' INTO TABLE " +
                table_name + " FIELDS TERMINATED BY ',' LINES
                TERMINATED BY '\n' IGNORE 1 LINES"
    cursor.execute(load_data)

    print ("Datos importados OK")

    print ("Insertando columnas fecha y clave ...")
    # Inserta la columna fecha delante de las demás
    insert_column = "ALTER TABLE " + table_name + " ADD `date` DATE
                    NOT NULL DEFAULT %s FIRST" %date
    cursor.execute(insert_column)
    cnx.commit()

    # Inserta columna uid1 delante de todas
    insert_column = "ALTER TABLE " + table_name + " ADD `uid1`
                    VARCHAR(10) DEFAULT NULL FIRST"
    cursor.execute(insert_column)
    # Actualiza columna uid1 con las claves que saca de tabla master
    update_column = ("UPDATE " + table_name + " T "
                    "SET T.uid1 = (SELECT V.uid1 "
                    "FROM vino_master_spa V "
                    "WHERE V.url = T.web)")
    cursor.execute(update_column)

```

```

cnx.commit()

print ("OK")

# Añade la condición de clave primaria a la clave uid1
add_key = ("ALTER TABLE " + table_name + " CHANGE COLUMN `uid1`
          `uid1` VARCHAR(10) CHARACTER SET 'utf8' NOT NULL , "
          "ADD PRIMARY KEY (`uid1`)")
cursor.execute(add_key)
cnx.commit()
print("Añadida condición de clave primaria OK")

# Añade relación entre claves con la tabla master_spa
add_rel = ("ALTER TABLE " + table_name + " ADD FOREIGN KEY
          (`uid1`) "
          "REFERENCES `vino_master_spa` (`uid1`) "
          "ON DELETE CASCADE "
          "ON UPDATE CASCADE")
cursor.execute(add_rel)
cnx.commit()
print ("Añadida relación entre tablas OK")

# Cerrando conexiones
cursor.close()
cnx.close()

time_fin = time()
time_total = time_fin - time_ini
print("Tiempo transcurrido para la tabla " + table_name + ":
      {0:.2f} segundos".format(time_total))
print ("-----")

except mdb.Error as err:
    print ("Something went wrong: {}".format(err))

```



## Script 13 – Solución adoptada

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# Crea una nueva tabla en la base de datos tfg, con el nombre del
# archivo csv que se le pasa como argumento y sube los datos desde
# el csv a la base de datos mysql, luego añade condición de claves
# primarias a uid1 y uid2, por último añade la relación con las
# tablas master
# Se importa como módulo en el script 6

from time import time
import mysql.connector as mdb
import csv

def upcsv( path_file ):

    # Obtiene el nombre del archivo
    path = path_file.split('/')
    file_name = path[len(path)-1]
    csv = file_name[:-4].lower()

    CREATE_TABLE = ( "CREATE TABLE `tfg`." + "`" + csv + "` ("
                    + "`uid2` VARCHAR(10), "
                    + "`url2` VARCHAR(1000), "
                    + "`uid1` VARCHAR(10), "
                    + "`url1` VARCHAR(1000), "
                    + "`hit_number` int"
                    + ")" )

    try:

        time_ini = time()

        # Conexión a la base de datos
        cnx = mdb.connect(user='root', password='root',
                          host='localhost', port='8889', database='tfg')
        cursor = cnx.cursor()

        # Creación de la tabla
        cursor.fetchone()
        cursor.execute(CREATE_TABLE)

        print ("Creación tabla " + csv + " OK")

        # Importación de los datos
        load_data = "LOAD DATA INFILE '" + path_file + "' INTO TABLE " +
                    csv + " FIELDS TERMINATED BY ',' LINES TERMINATED
                    BY '\n' IGNORE 1 LINES"
        cursor.execute(load_data)
        cnx.commit()

        print ("Datos importados OK")
```

```

# Añade condición de clave primaria a uid1 y uid2
add_keys = ("ALTER TABLE " + csv + " "
           "CHANGE COLUMN `uid2` `uid2` VARCHAR(10) CHARACTER
           SET 'utf8' NOT NULL , "
           "CHANGE COLUMN `uid1` `uid1` VARCHAR(10) CHARACTER
           SET 'utf8' NOT NULL , "
           "ADD PRIMARY KEY (`uid2`, `uid1`)")
cursor.execute(add_keys)
cnx.commit()

# Añade relación entre claves con la tabla master_spa
add_rel = ("ALTER TABLE " + table_name +
           " ADD FOREIGN KEY (`uid1`) "
           "REFERENCES `vino_master_spa` (`uid1`) "
           "ON DELETE CASCADE "
           "ON UPDATE CASCADE")
cursor.execute(add_rel)
cnx.commit()

# Añade relación entre claves con la tabla master_int
add_rel = ("ALTER TABLE " + table_name +
           " ADD FOREIGN KEY (`uid2`) "
           "REFERENCES `vino_master_int` (`uid2`) "
           "ON DELETE CASCADE "
           "ON UPDATE CASCADE")
cursor.execute(add_rel)
cnx.commit()
print ("Añadida relación entre tablas OK")

# Cerrando conexiones
cursor.close()
cnx.close()

time_fin = time()
time_total = time_fin - time_ini
print("Tiempo transcurrido para la tabla " + csv + ": {0:.2f}
      segundos".format(time_total))
print ("-----")

except mdb.Error as err:
    print ("Something went wrong: {}".format(err))

```





# Anexo II - Scripts en R

---

## Script 1 – Ratio CF/TF

```
#!/usr/bin/env Rscript
# -*- coding: utf-8 -*-

# Suprime errores y warnings indeseados durante la ejecución
options(show.error.messages = FALSE)
options(warn = -1)

## ANALISIS DE INDICADORES CF/TF ##

# Carga de las librerías
library(DBI)          # Librería requerida para instalar RMySQL
library(RMySQL)      # Para trabajar con comandos de MySQL
library(randomcolor) # Colores para gráficas
library(e1071)       # Funciones para análisis estadísticos

###-----###
###          VARIABLE A MODIFICAR          ###
###-----###
### Consulta a MySQL para la tabla vino160404 ###

table = "vino160404"

###-----###

# Función que introduce una pausa hasta pulsar ENTER,
# y entonces se cierran todas los plots abiertos
readkey <- function()
{
  cat("*** Press [ENTER] to close windows ***\n")
  aux <- scan(file = "stdin", "", nlines = 1, quiet = TRUE)
  graphics.off()
}

# Conexión con la base de datos
con <- dbConnect(MySQL(), dbname = "tfg", user = "root", password = "root",
                 host = "127.0.0.1", port = 8889)
dbGetQuery(con, "SET NAMES 'utf8'") # Para que la codificación sea en utf8

# Consulta que extrae una tabla con el nombre de la empresa y su indicador CF
# Ordenada por CF descendente
query1.1 <- paste("SELECT M.empresa, V.citationFlow
                  FROM vino_master_spa M, ", table, " V
                  WHERE M.uid1 = V.uid1
                  ORDER BY V.citationFlow DESC", sep = "")
rows1.1 <- dbSendQuery(con, query1.1) # Envío la consulta
rows1.1 <- fetch(rows1.1, -1)       # Recojo la respuesta
```

```

# Ordenada por nombre de empresa
rows1.2 <- rows1.1[order(rows1.1[,1]), ]

# Consulta que extrae una tabla con el nombre de la empresa y su indicador TF
# Ordenada por TF descendente
query2.1 <- paste("SELECT M.empresa, V.trustFlow
                  FROM vino_master_spa M, ", table, " V
                  WHERE M.uid1 = V.uid1
                  ORDER BY V.trustFlow DESC", sep = "")
rows2.1 <- dbSendQuery(con, query2.1)
rows2.1 <- fetch(rows2.1, -1)
# Ordenada por nombre de empresa
rows2.2 <- rows2.1[order(rows2.1[,1]), ]

# Ranking por CF y TF
tableCor <- data.frame(rows1.1$empresa, rows1.1$citationFlow,
                      rows2.1$empresa, rows2.1$trustFlow)
View(tableCor)

# Ratio CF/TF
tableRatio <- data.frame(rows1.2$empresa, rows1.2$citationFlow,
                        rows2.2$trustFlow,
                        round(rows1.2$citationFlow/rows2.2$trustFlow,
                              digits = 2))
View(tableRatio)

# Gráfico de dispersión
x <- rows1.2$citationFlow
y <- rows2.2$trustFlow
if (max(range(x)) > max(range(y)))
{
  rango <- range(x)
} else
{
  rango <- range(y)
}
x11()
plot(x, y, xlim = rango, ylim = rango, pch = 19,
     main = "Correlación entre CF y TF", col = distinctColorPalette(2744),
     xlab = "Citation Flow", ylab = "Trust Flow")
lines(rango, rango, col = "red", lwd = 2)

# Covarianza y coeficiente de correlación entre CF y TF
cat("Covarianza : ")
cat(cov(x, y))
cat("\n")
cat("Correlación: ")
cat(cor(x, y))
cat("\n")

readkey()

```



## Script 2 – Series temporales EBL

```
#!/usr/bin/env Rscript
# -*- coding: utf-8 -*-

# Suprime errores y warnings indeseados durante la ejecución
options(show.error.messages = FALSE)
options(warn = -1)

# Carga de las librerías
library(DBI)          # Librería requerida para instalar RMySQL
library(RMySQL)      # Para trabajar con comandos de MySQL
library(randomcolor) # Colores para gráficas

### Consulta a MySQL para todas las tablas de Majestic ###
table <- c("vino160404", "vino160502", "vino160603", "vino160702")

# Conexión con la base de datos
con <- dbConnect(MySQL(), dbname = "tfg", user = "root", password = "root",
                 host = "127.0.0.1", port = 8889)
dbGetQuery(con, "SET NAMES 'utf8'") # Para que la codificación sea en utf8

# Función que introduce una pausa hasta pulsar ENTER,
# y entonces se cierran todas los plots abiertos
readkey <- function()
{
  cat("*** Press [ENTER] to close windows ***\n")
  aux <- scan(file = "stdin", "", nlines = 1, quiet = TRUE)
  graphics.off()
}

# Consulta que extrae los indicadores EBL, EBLE, EBLG
# de todas las tablas de Majestic que hay en MySQL
query <- paste("SELECT T1.extBacklinks, T1.extBacklinksEdu,T1.extBacklinksGov
FROM ", table[1], " T1
WHERE T1.uid1 = 'SPA1525S'
UNION ALL
SELECT T2.extBacklinks, T2.extBacklinksEdu,T2.extBacklinksGov
FROM ", table[2], " T2
WHERE T2.uid1 = 'SPA1525S'
UNION ALL
SELECT T3.extBacklinks, T3.extBacklinksEdu,T3.extBacklinksGov
FROM ", table[3], " T3
WHERE T3.uid1 = 'SPA1525S'
UNION ALL
SELECT T4.extBacklinks, T4.extBacklinksEdu,T4.extBacklinksGov
FROM ", table[4], " T4
WHERE T4.uid1 = 'SPA1525S'", sep = "")
rows <- dbSendQuery(con, query)
rows <- fetch(rows, -1)
# Construcción de la tabla de datos
tableEBL <- data.frame(rows)
row.names(tableEBL) <- table
View(tableEBL)
```

```

# Gráficas de las series temporales en una misma ventana
x11()
par(mfrow = c(2, 2), pch = 16)
# EBL
plot(tableEBL$extBacklinks, col = "blue", type = "p",
      xlab = "Monts", ylab = "EBL", main = "External Back Link History",
      las = 1, xaxt='n', cex.lab = 0.9, cex.axis = 0.7)
axis(1, at = 1:4, lab = c("April", "May", "June", "July"), las = 1)
lines(c(1,2,3,4), tableEBL$extBacklinks, col = "blue")
# EBLE
plot(tableEBL$extBacklinksEdu, col = "red", type = "p",
      xlab = "Monts", ylab = "EBLE", main = "External Back Link Edu History",
      las = 1, xaxt='n', cex.lab = 0.9, cex.axis = 0.7)
axis(1, at = 1:4, lab = c("April", "May", "June", "July"), las = 1)
lines(c(1,2,3,4), tableEBL$extBacklinksEdu, col = "red")
# EBLG
plot(tableEBL$extBacklinksGov, col = "green", type = "p",
      xlab = "Monts", ylab = "EBLG", main = "External Back Link Gov History",
      las = 1, xaxt='n', cex.lab = 0.9, cex.axis = 0.7)
axis(1, at = 1:4, lab = c("April", "May", "June", "July"), las = 1)
lines(c(1,2,3,4), tableEBL$extBacklinksGov, col = "green")

par(mfrow = c(1, 1), pch = 1)
readkey()

```



## Script 3 – Visualización de varias gráficas

```
#!/usr/bin/env Rscript
# -*- coding: utf-8 -*-

# Suprime errores y warnings indeseados durante la ejecución
options(show.error.messages = FALSE)
options(warn = -1)

# Carga de las librerías
library(DBI)          # Librería requerida para instalar RMySQL
library(RMySQL)      # Para trabajar con comandos de MySQL
library(randomcoloR) # Colores para gráficas

###-----###
###          VARIABLES A MODIFICAR          ###
###-----###
### Consulta a MySQL para la tabla vino160404 ###
table = "vino160404"
# Variable con la ruta donde se guardan las gráficas en pdf
path = "/Users/gloriamunozedo/Documents/R/"
###-----###

opt <- 1
while(opt) {

  cat("-----
  Select indicator to analyze :
  -----")

  1 - Citation Flow
  2 - External Back Links
  3 - External Back Links Edu
  4 - External Back Links Gov
  5 - Topical Trust Flow
  6 - Trust Flow

  0 - Exit

  > ")

  ask <- TRUE
  while (ask) {
    opt <- scan(file = "stdin", "", n = 1, quiet = TRUE)
    opt <<- as.numeric(opt)
    if (!is.na(opt) && opt >= 0 && opt <= 6)
    {
      ask <- FALSE
      cat("\n")
    }
    else cat("Valor no válido, selecciona una opción válida\n    > ")
  }
}
```

```

### Opción 0 - Exit ###
if (opt == 0)
{
  cat("\nGoodbye!\n")
  stop(domain = NA)
}

### Resto de opciones ###
switch(opt,
  { field <- "citationFlow" },
  { field <- "extBacklinks" },
  { field <- "extBacklinksEdu" },
  { field <- "extBacklinksGov" },
  { field <- "topicalTrustFlow" },
  { field <- "trustFlow" }
)

cat("-----
Select mode :
-----

1 - Display on screen
2 - Save in pdf

> ")

ask <- TRUE
while (ask) {
  opt2 <- scan(file = "stdin", "", n = 1, quiet = TRUE)
  opt2 <- as.numeric(opt2)
  if (!is.na(opt2) && opt2 >= 1 && opt2 <= 2)
  {
    ask <- FALSE
    cat("\n")
  }
  else cat("Valor no válido, selecciona una opción válida\n  > ")
}

# Conexión con la base de datos
con <- dbConnect(MySQL(), dbname = "tfg", user = "root", password = "root",
  host = "127.0.0.1", port = 8889)
dbGetQuery(con, "SET NAMES 'utf8'") # Para que codificación sea en utf8

# Función que introduce una pausa hasta pulsar ENTER,
# y entonces se cierran todas los plots abiertos
readkey <- function()
{
  cat("*** Press [ENTER] to close windows ***\n")
  aux <- scan(file = "stdin", "", nlines = 1, quiet = TRUE)
}

# Función para la consulta de 1 solo campo
query1 <- function (table, field) {
  query <- paste("SELECT ", field, " from ", table, sep = "")
  rows <- dbSendQuery(con, query) # Envío la consulta
  rows <- fetch(rows, -1) # Recojo la respuesta
}

```



```

    return(rows)
}

# Función para la consulta de 2 campos
query2 <- function (table, field1, field2) {
  query <- paste("SELECT ", field1, ", ", field2, " from ", table, sep = "")
  rows <- dbSendQuery(con, query) # Envío la consulta
  rows <- fetch(rows, -1)        # Recojo la respuesta
  return(rows)
}

# Función para dibujar el Diagrama de Caja
boxPlot <- function (table, field) {
  rows <- query1(table, field)
  options(bitmapType='cairo')
  graph <- "_boxp"
  if (opt2 == 1)
  {
    x11(title = paste("Boxplot of ", field, sep = ""))
  }
  boxplot(rows, main = paste("Boxplot of ", field, sep = ""), ylab =
    paste(field, " (%)", sep = ""))
}

# Función para dibujar el Histograma con la curva de densidad
histo <- function (table, field) {
  rows <- query1(table, field)
  v <- t(rows)
  d <- density(v)
  h <- hist(v, plot = FALSE)
  options(bitmapType='cairo')
  graph <- "_hist"
  if (opt2 == 1)
  {
    x11(title = paste("Histogram of ", field, sep = ""))
  }
  h <- hist(v, main = paste("Histogram of ", field, sep = ""),
    xlab = field, ylab = "Density", las = 1, prob = TRUE,
    xlim = c(0, max(v)), ylim = c(0, max(max(h$density))),
    col = distinctColorPalette(30))
  lines(d, col = "blue", lwd = 2)
}

# Función para dibujar el Diagrama de Barras del TTF
barPlot <- function (table, field1, field2) {
  rows <- query2(table, field1, field2)
  options(bitmapType='cairo')
  graph <- "_barp"
  # Ordeno por la columna de topics
  rows.ord <- rows[order(rows[,1]), ]
  # Saco los topics y sus frecuencias absolutas
  topics.freq <- table(rows.ord[1])
  topics.freq <- as.data.frame(topics.freq)
  # Me quedo con los 10 de mayor frecuencia

```

```

topics.freqord <- topics.freq[order(topics.freq[,2], decreasing = TRUE),
]
topten <- topics.freqord[1:10,]
# Tabla de frecuencias relativas del topten
ttf.table <- topten
ttf.table[3] <- 100*ttf.table[2]/sum(topics.freqord[,2])
# Diagrama de barras
barplot(t(topten[2]), main = paste("Barplot of ", field1, sep = ""),
        col = randomColor(count = 10, hue = c("random"),
        luminosity = c("light")), ylim = c(0, max(topten[2])),
        beside = TRUE, legend = topten$Var1, cex.names = 0.8,
        args.legend = list(title = "Topics", x = "topright",
        cex = 0.58), las = 2, names.arg = round(ttf.table$Freq.1, 2),
        space = rep(c(0), 10), axisnames = TRUE,
        xlab = paste("Topten of ", field1, sep = ""),
        ylab = "Frequency")
}

### MAIN ###
if (opt == 5)
{
  if (opt2 == 2) pdf(paste(table, field, ".pdf", sep = ""))
  if (opt2 == 1)
  {
    x11(title = "Barplots of TTF")
    par(mfrow = c(2, 2), pch = 16)
  }
  for (i in c(0, 1, 2)) {
    field1 <- paste(field, "_topic_", i, sep = "")
    field2 <- paste(field, "_value_", i, sep = "")
    barPlot(table, field1, field2)
  }
  par(mfrow = c(1, 1), pch = 1)
  if (opt2 == 1)
  {
    readkey()
    graphics.off()
  }
} else
{
  if (opt2 == 2) pdf(paste(table, field, ".pdf", sep = ""))
  boxPlot(table, field)
  histo(table, field)
  if (opt2 == 1)
  {
    readkey()
    graphics.off()
  }
}
}
}

```

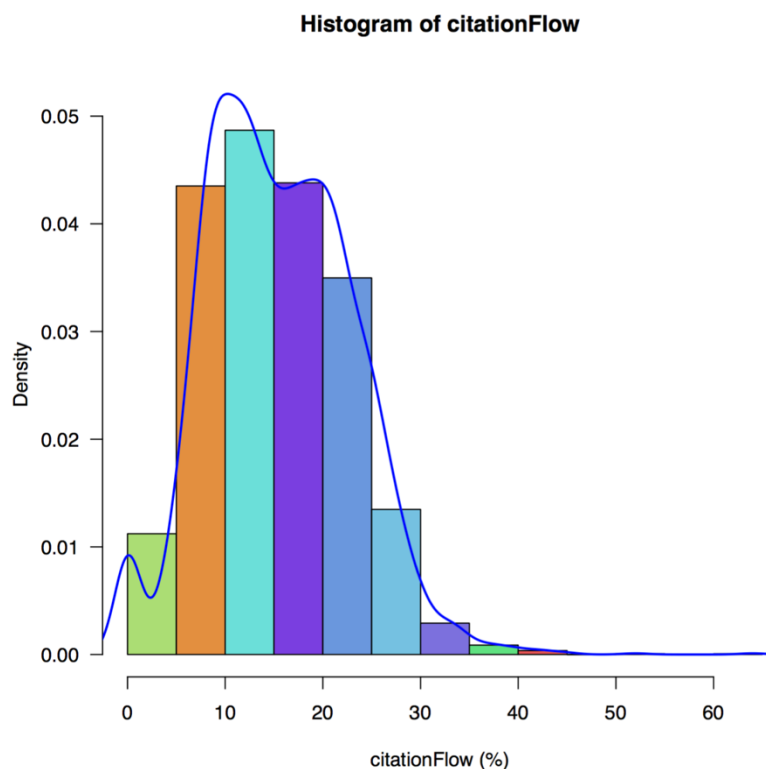
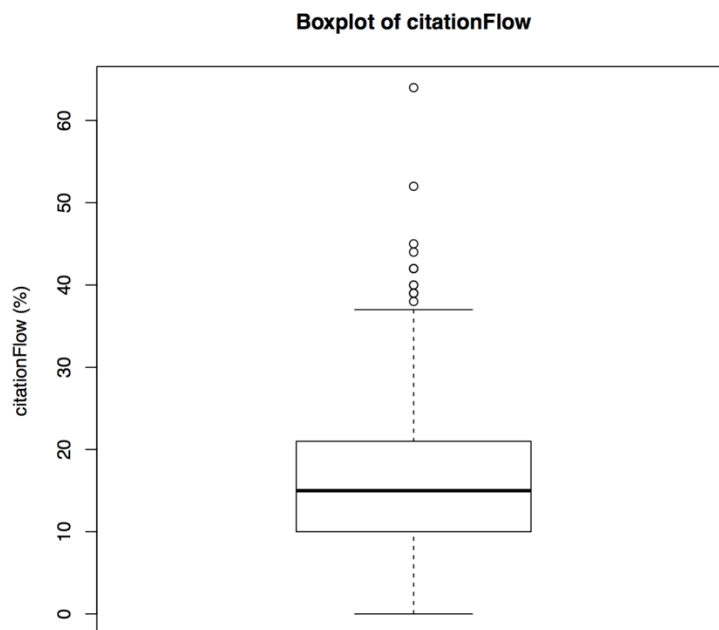




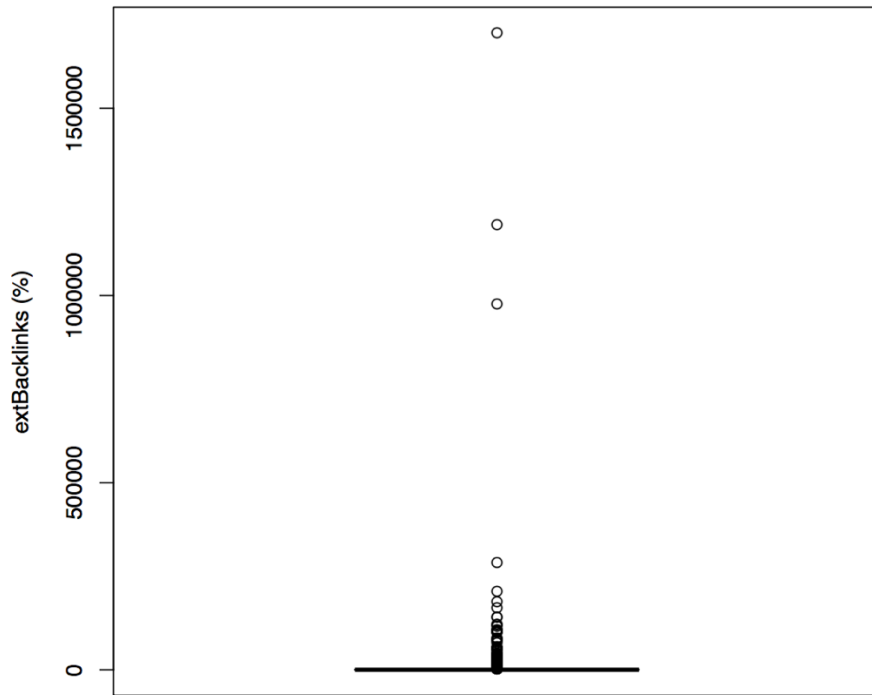
# Anexo III - Gráficas obtenidas con R

---

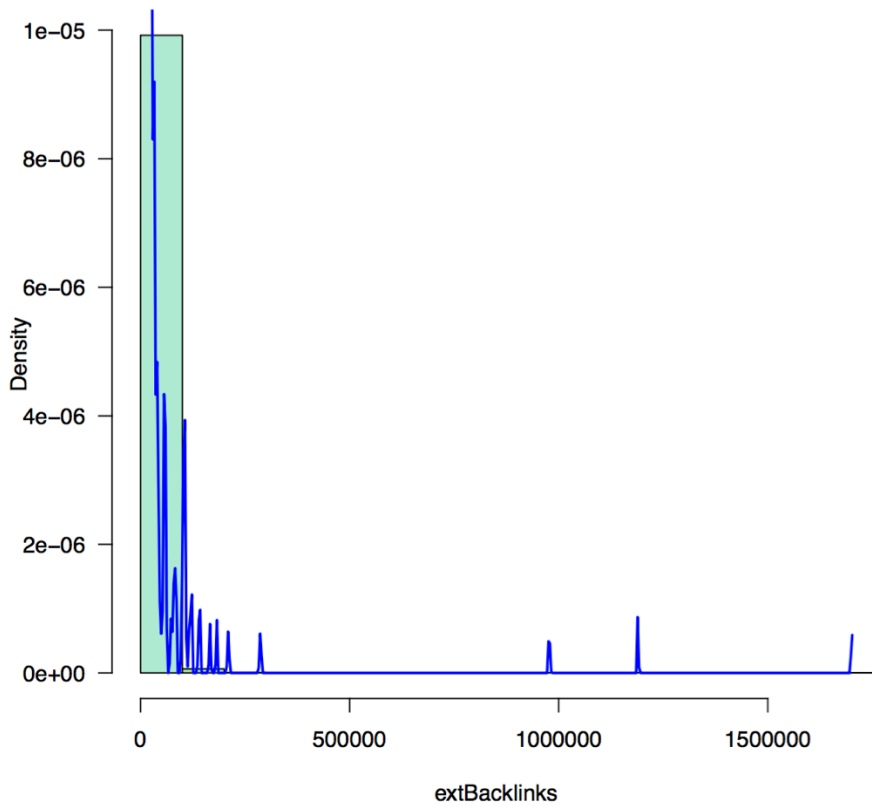
A continuación se adjuntan los gráficos que se obtienen en formato PDF, mediante la ejecución del *script* 3 del anexo II. Únicamente se incluyen los gráficos del ejemplo de la tabla *vino160404*, por no incrementar en exceso el contenido de la presente memoria.



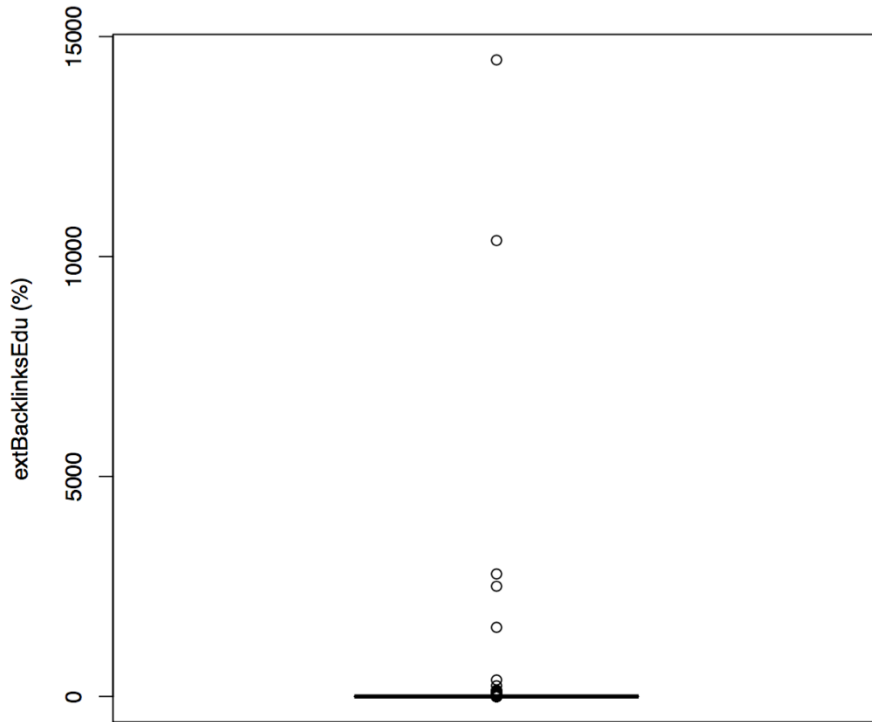
**Boxplot of extBacklinks**



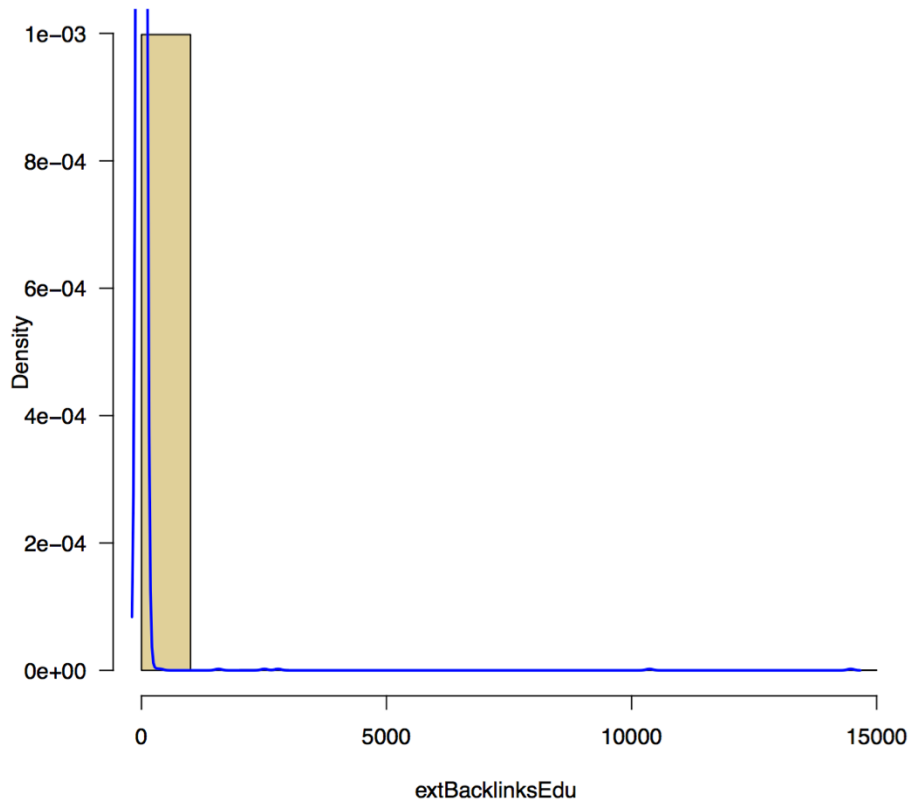
**Histogram of extBacklinks**



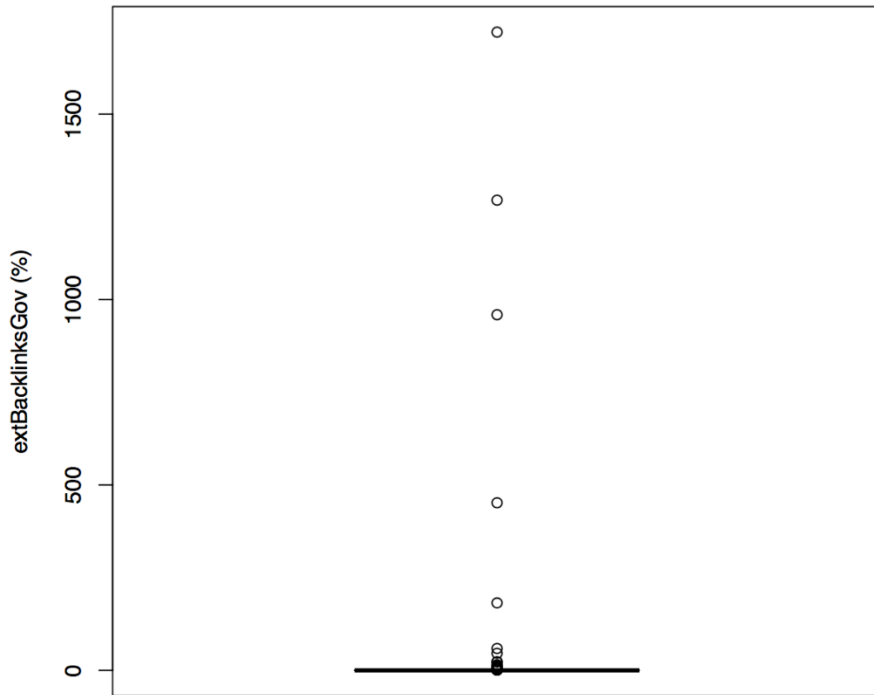
**Boxplot of extBacklinksEdu**



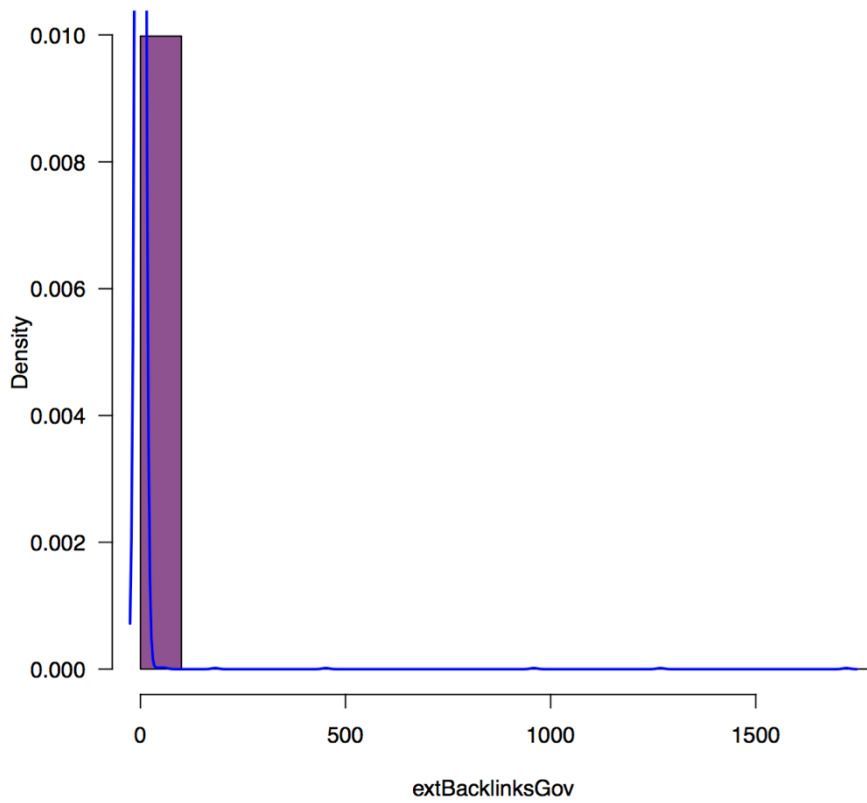
**Histogram of extBacklinksEdu**

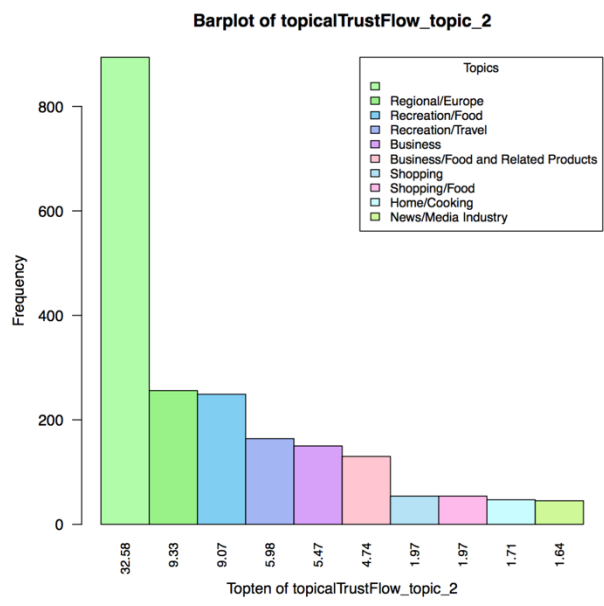
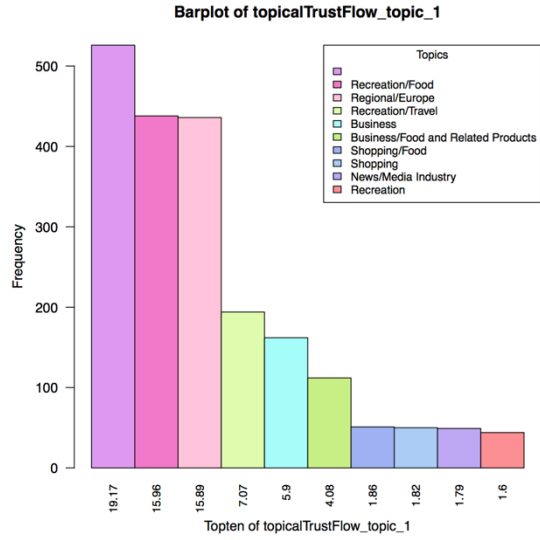
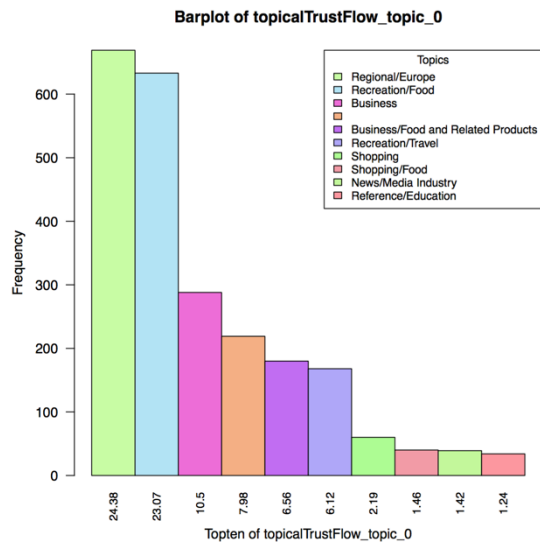


**Boxplot of extBacklinksGov**

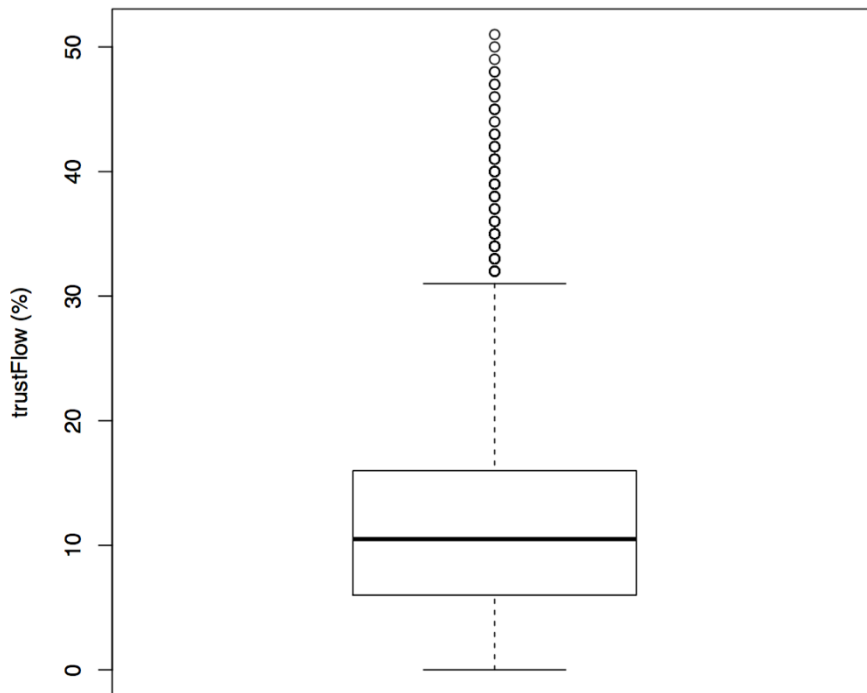


**Histogram of extBacklinksGov**





**Boxplot of trustFlow**



**Histogram of trustFlow**

