UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
ESCUELA DE POST GRADO
MAESTRÍA EN CIENCIAS MENCIÓN INFORMÁTICA

TESIS

MINERÍA DE USO WEB PARA IDENTIFICAR PREFERENCIAS DE NAVEGACIÓN EN LAS PÁGINAS WEB DE LA UNSAAC

PARA OBTENER EL GRADO ACADÉMICO DE:
MAESTRO EN CIENCIAS MENCIÓN INFORMÁTICA

PRESENTADO POR:
BR. WILLIAN ZAMALLOA PARO
ASESOR:
MGT. NILA ZONIA ACURIO USCA

2019

INDICE GENERAL

INDICE DE TABLAS

INDICE DE FIGURAS

RESUMEN

Anticiparse a lo que desea ver un usuario de una página web hoy en día es muy importante, para hacer tangible esta característica se podrían realizar periódicamente desde encuestas hasta cuestionarios complejos de realizar y complicados de consolidar, en el caso de ser esto posible habría un margen de error. Por consiguiente, para el presente proyecto se tiene como objetivo determinar las preferencias de navegación de los usuarios de las páginas web de la UNSAAC en base a los log del servidor web del dominio www.unsaac.edu.pe.

Un log es un archivo que almacena toda la interacción de los usuarios o personas que consultan información en una página web la cual guarda todas estas interacciones sin sesgo alguno.

Los log utilizados en el presente proyecto fueron facilitados por la RCU (Red de Comunicaciones UNSAAC) los que corresponden al año 2017, el contenido de estos archivos en crudo no están estructurados, además no cuentan con algún tipo de pre procesamiento o procesamiento alguno.

El objetivo del proyecto fue determinar las preferencias de navegación para lo cual se pasaron por dos fases, se realizó la preparación de datos y el descubrimiento de patrones, donde se realizó el pre procesamiento de datos mediante un programa de mi autoría se pasaron los log a una base de datos, en esta instancia se realizó una limpieza previa excluyendo a los log accedidos desde la ip 127.0.0.1, seguidamente se creó una base de datos de transacciones donde se realizó la siguiente limpieza en este caso excluyendo logs que contenían archivos de hojas de estilo, javascript, imágenes, bots de google e iconos, seguidamente se realizó la minería de uso web donde se aplicó la minería de reglas de asociación y finalmente la obtención y análisis de patrones.

Se seleccionaron muestras significativas en forma aleatoria simple debido a que la cantidad total de logs ascendieron a 94949716 aproximadamente y el tiempo computacional para procesarlos era alto, se tomó el 3.33% haciendo que la muestra fuera de 3797989, para obtener dicha muestra se utilizó un programa que pueda dar lectura a archivos de más de 1 GB, se procedió a dividir cada uno de los 40 archivos log tomando en cuenta que la navegación del ultimo usuario en el log no se pierda, de allí que el porcentaje de la muestra no sea exacto, pero luego de la limpieza y estructuración se obtuvo 32994 transacciones donde están contenidas las preferencias de navegación.

Finalmente se obtuvieron 11 reglas de asociación las cuales representan las preferencias de navegación de los usuarios, con un 90% de confianza a las cuales se les realizaron métricas de validación para descartar que dichas preferencias se deban al azar, también se obtuvo los patrones y las secuencias en que se visitan las distintas páginas de la UNSAAC y el orden en el que lo hacen, con estos resultados se demostró que mediante el proceso de minería de uso web se puede estructurar logs, también se pudo identificar las preferencias de navegación de los usuarios en función a los accesos realizados a la página web de la UNSAAC y demostrar que en archivos cercanos al big data hay preferencias ocultas y valiosas para la institución.

PALABRAS CLAVE

Minería de datos, Minería web, Minería de uso web, Log, reglas de asociación, preferencias, apriori.

ABSTRACT

Anticipating what a user of a web page wants to see, nowadays it is very important, to make this feature tangible they could be carried out periodically from surveys to questionnaires, complex to perform and complicated to consolidate, in the case of being this possible there would be A margin of error. Therefore, the objective of this project is to determine the browsing preferences of the users of the UNSAAC web pages, based on the log of the web server of the domain www.unsaac.edu.pe.

A log is a file that stores all the interaction of users or people who consult information on a web page, which saves all these interactions without any bias.

The log used in this project were provided by the RCU (UNSAAC Communications Network), which correspond to the year 2017, the content of these raw files, are not structured, also do not have any type of preprocessing or any processing.

The objective of the project was to determine the navigation preferences for which they went through two phases, data preparation and pattern discovery were carried out, where the pre-processing of data was carried out where the logs were passed through a program of my own. to a database, in this instance a previous cleaning was carried out excluding the log accessed since ip 127.0.0.1, then a transaction database was created where the following cleaning was performed in this case excluding logs containing files of style sheets, javascript, images, google bots and icons, then the web use mining where the association rules mining was applied and finally the obtaining and analysis of patterns.

Significant samples were selected in simple random form, because the total amount of logs amounted to approximately 94949716, and the computational time to process them was high, 3.33% was taken making the sample out of 3797989, to obtain said sample a program that can read files larger than 1 GB, each of the 40 log files was divided taking into account that the navigation of the last user in the log is not lost, hence the percentage of the sample does not It is accurate, but after cleaning and structuring, 32994 transactions were obtained where navigation preferences are contained.

Finally, 11 association rules were obtained which represent the user's browsing preferences, with a 90% confidence to which validation metrics were made to rule out that these preferences are due to chance, also the patterns and the sequences in which the different pages of the UNSAAC are visited and the order in which they do it, with these results it was demonstrated that through the process of web use mining, logs can be structured, user navigation preferences could also be identified based on the accesses made to the UNSAAC website and demonstrate that in archives close to big data there are hidden and valuable preferences for the institution.

KEYWORDS
Data mining, Web mining, Web use mining, Log, association rules, preferences, apriori.

AGRADECIMIENTOS

A mis padres y a toda mi familia por su apoyo incondicional, por su presencia, concejos y recomendaciones, quienes no dejaron que pierda de vista mis objetivos.

A mi esposa Patricia y en especial a mi hija Alejandra, por su comprensión, porque me impulsaron siempre a seguir adelante, y mejorar profesionalmente cada día.

A cada uno de los catedráticos de la Maestría en Ciencias Mención Informática de la Escuela de Posgrado, por sus enseñanzas, a mi asesora Mg, Nila Zonia Acurio Usca, por su orientación y apoyo.

A la Universidad Nacional de San Antonio Abab del Cusco, por permitirme formar parte de su excelente plana de profesionales, me llena de mucho orgullo pertenecer a esta prestigiosa casa de estudios.

INTRODUCCION

Hoy en día las páginas web deberían anticiparse a lo que buscan o quieren las personas, lo que hace pensar en la forma como se podría realizar esto. El hallazgo de preferencias de navegación en páginas web, permite realizar dicha tarea, existen varias formas de realizar esto pero eso depende mucho de los datos que se utilizan y de las técnicas que se puedan usar, lo que no se evidencio en otras investigaciones, es el hecho de que el hallazgo de dichas preferencias dependen de factores como, el formato de los log, el tamaño de la data que se analizara, de cada log como unidad de análisis dentro del archivo; además que dichas preferencias están ocultas y no son visibles en una inspección rápida o simple o una más rigurosa, debido a sus grandes proporciones.

Se añadió el hecho que si se realiza dicha tarea, podría darse una recomendación a la persona que este navegando en una u otra página, anticipándonos y sugiriendo la siguiente página que vera, esto dio lugar a que se plantee la búsqueda de preferencias de navegación y para realizar esto se utilizaran las reglas de asociación; se eligió dicha opción por las similitudes que hay con una canasta de compras, aplicada al comercio electrónico y la forma como una página web podría en forma análoga, ser un producto de la canachrsta de compras, que el usuario desea llevar.

En el capítulo 1, se da a conocer la descripción del problema, el objetivo que se desea alcanzar, la justificación respectiva y las razones porque se desarrolla esta investigación.

En el capítulo 2, se engloba el marco teórico, el estado del arte, y conceptos relacionados al desarrollo de este proyecto, se puede evidenciar también la lectura de trabajos de investigación y libros, que se usan como antecedentes del proyecto, se define la minería de uso web y su proceso.

En el capítulo 3, se definen la hipótesis, la identificación de variables, indicadores y como se operan las variables.

En el capítulo 4, se describe el diseño de la investigación, unidad de análisis, las técnicas de recolección de datos, se analiza e interpreta la información, se define también el alcance del proceso de minería de uso web sus dos fases y sus respectivas etapas.

En el capítulo 5, se presentan los resultados, analizando e interpretando cada uno de ellos, se realiza la prueba de hipótesis, se detalla los procesos aplicados a los log estructurados desde su origen en crudo, hasta la obtención de las reglas de asociación, que demuestran que hay preferencias ocultas en dichos archivos.

Finalmente se redactaron las conclusiones y recomendaciones de la presente investigación y se incluyeron los anexos correspondientes.

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 SITUACION PROBLEMÁTICA

En la actualidad, buscar información sobre algún tema de interés es algo que ya no se realiza como antes, la búsqueda de dicha información así como muchas actividades humanas ha ido cambiando conforme avanzaba la tecnología si antes se utilizaban directorios telefónicos, revistas, libros, etc. ahora todo eso ha cambiado se usa el internet para realizar todas estas tareas de modo que se satisface la necesidad de informarse en un solo espacio, esto gracias a todas estas herramientas tecnológicas existentes en la actualidad.

Hoy en día existen muchas formas de acceder a esta información entre ellas se encuentran las páginas web, que vienen a ser documentos digitales los cuales contienen textos, imágenes y videos, con una temática muy variada donde las personas pueden encontrar información de: ciencias, salud, tecnología, educación, solo por mencionar algunos ejemplos; pero la gran variedad de dichas páginas con toda su información en muchos casos se centralizan en páginas web las cuales responden a personas o instituciones, donde cada una ellas es la que administra el contenido de las mismas.

Las páginas web son el primer contacto que tienen las personas con las instituciones o empresas, donde estos últimos informan mediante artículos, publicaciones, etc. información relativa a los servicios, productos o descripciones de dichas instituciones, según corresponda, dichas páginas web son almacenadas en servidores web.

La Universidad Nacional San Antonio Abad del Cusco (UNSAAC), tiene una página web con el dominio http://www.unsaac.edu.pe/, la cual viene funcionando desde el 19 de diciembre de 1996, y que en el transcurso del tiempo se ha venido actualizando y modificando su apariencia, algunas de las actividades principales de la RCU (RED DE COMUNICACIONES UNSAAC) son, la elaboración y actualización de páginas web de todas las dependencias de la UNSAAC y publicación de las principales actividades de la universidad (RCU, 2018).

Actualmente en la página web de la UNSAAC se publica información de los distintos vicerrectorados como son el académico y el de investigación, además se realizan publicaciones de manera casi diaria en cuanto se refiere a eventos como congresos, cursos de capacitación, convocatorias, anuncios y alberga también en sus subdominios, páginas de las distintas carreras profesionales y áreas administrativas. La principal ventaja que tiene la página web de la UNSAAC es que permite informar de distintos sucesos que acontecen y mencionados más arriba, sin la necesidad de visitarla en físico. Pero en muchos casos no se sabe que recursos son o no consultados y aprovechados por los usuarios de dichas páginas.

Hay muchas formas en las cuales podríamos obtener un feedbak, como realizar una encuesta o cuestionario a un grupo de los usuarios de la página web, en la cual nos dirían que secuencia de páginas vistan o cual es la que más buscan, que si bien es cierto nos darían una idea de lo que buscamos, sin embargo se tiene el problema de que previamente se tendría que saber cuáles son las páginas de la UNSAAC, tener la certeza de que esas páginas siguen vigentes o que nuevas

páginas se agregaron, lo que conllevaría a una encuesta o cuestionario demasiado elaborada y compleja, pues se le tendría que dar alternativas de posibles secuencias, o que ellos indiquen sus propias secuencias en el caso de que no se halle en las alternativas, y en muchos casos probablemente sea subjetivo y sobre todo se tendría que realizar periódicamente, lo cual sería una actividad tediosa, la de consolidar dichos resultados y realizar un análisis posterior del mismo.

De igual manera, no se sabe cuál o cuáles son las páginas más consultadas o como van variando, solo se tiene una opinión o apreciación de cuál sería o podría ser la página o páginas más visitadas, cada página web existe de manera independiente pues no se sabe si hay alguna relación o coincidencia entre ellas, al momento de ser visitadas.

Frente a esa situación, se centrará en otro tipo de análisis, orientado más a la minería de datos, el cual permite realizar dicha tarea pues está pensado en el análisis de datos estructurados en distintos formatos, de tal forma que el conocimiento extraído, brinde un beneficio a la organización, por su exactitud y flexibilidad al ser la minería de datos un método más confiable.

Al revisar y consultar varios trabajos como son los de (Malviya & Agrawal, 2015), (Lin & WenZheng, 2015) o (Sukumar, Robert, & Yuvaraj, 2016), solo por mencionar algunos, se tomó la decisión que, el análisis se realizaría utilizando los logs de la página almacenados en los servidores de la institución, los cuales registran el comportamiento real que tuvo un usuario al momento de interactuar con la página web, y no deja lugar a interpretaciones sesgadas o polarizadas, también es importante mencionar que dichos logs no presentan la información en forma estructurada, lo que impide cualquier tipo de visualización de características similares o agrupaciones que nos permitirían entender mejor las interacciones.

En el presente trabajo de investigación, se propone el uso en específico, de la minería de uso web, que a su vez es parte de la minería web y esta a su vez parte de la minería de datos, se eligió dicha opción, pues la minería de uso web se centra en el análisis de logs, todo esto con el objetivo de encontrar características similares en función a los preferencias que tienen los usuarios, al navegar en las páginas web de la UNSAAC, además servirá a manera de feedback al RCU, de cómo es que navegan los usuarios en dichas páginas web.

1.2 FORMULACION DEL PROBLEMA

¿Mediante la minería de uso web, se puede identificar las preferencias de navegación de los usuarios en las páginas web de la UNSAAC?

1.3 JUSTIFICACION DE LA INVESTIGACION

Esta investigación planteada, fue motivada por ciertas características observables, que tienen las páginas web con trafico considerable, como son las avocadas al comercio electrónico, solo por mencionar un ejemplo, estos servicios en particular son las sugerencias que dan dichas paginas cuando uno navega en ellas y como diferencian entre las distintas personas que están en ellas, recomendando cosas similares o distintas y acertando en su mayoría de veces con las sugerencias que dan.

La característica de anticiparse a la página que un usuario desea ver, cada vez es más común en la web, entonces se pretende trasladar los hechos mencionados anteriormente, pero a una página web que no se aboque al comercio electrónico, sino por el contrario que brinde un servicio de información, a sus usuarios, de tal forma que el resultado de esta investigación sea de utilidad para la UNSAAC, donde se resalte la importancia de conocer la relación que hay entre las páginas web y el impacto que tendría el hecho de anticiparse a la preferencia de una u otra página, que un usuario tiene sobre las demás, por esta razón este trabajo resulta un gran aporte para la institución, pues se tendrá directamente un diagnóstico de todas las interacciones de los usuarios, esto permitirá observar, cuan relevantes son, también el tiempo que demora obtener las preferencias y las dificultades que se tienen al momento de estructurarlas, para de esta forma tener una idea clara de cuál es el estado real de la página, datos que son valiosos para la RCU y que se pueden utilizar en futuros mantenimientos, restructuraciones de dichas páginas o en configuraciones más convenientes en sus servidores, pues actualmente no se tiene información precisa de que se está registrando realmente en los log del servidor web.

Conocer con anticipación y sugerir que página es la siguiente que prefiere ver un usuario, solo sería posible, siempre y cuando un usuario indique cual o cuales son las páginas de su interés, e inclusive el orden en el que las visitarían, pero para que se obtenga dicha información, el usuario tendría que conocer las páginas que hay actualmente y también si se añadió o retiro alguna página, con lo que dicha información obtenida estaría sujeta a cierto sesgo, además se tendría que agrupar a usuarios con las mismas respuestas y clasificar las páginas que ellos visitan y establecer la secuencia de como navegan lo que a simple vista será una tarea complicada y demasiado laboriosa.

Lo que se plantea como solución a este problema, es obtener un modelo de reglas de asociación la cual es una técnica de la minería de datos, que muestran las preferencias por una página u otra, con un sesgo mínimo, que no depende de si se añadió o retiro una página y sobre todo que se obtendrán a partir de todas las páginas que visito un determinado usuario o varios usuarios, permitiendo agruparlos por sus preferencias, estableciendo su secuencia de navegación, de tal forma que se sabe de antemano cual es la siguiente página que el usuario prefiere ver, por lo que las reglas de asociación podrán determinar dichas preferencias.

1.4 OBJETIVOS DE LA INVESTIGACION

1.4.1 OBJETIVO GENERAL

Aplicar la minería de uso web para identificar las preferencias de navegación en las páginas web de la UNSAAC.

1.4.2 OBJETIVOS ESPECÍFICOS

- ✓ Construir una estructura de datos que incluya las acciones realizadas por los usuarios a partir de los archivos log, para su análisis respectivo, usando el proceso de minería de uso web.
- ✓ Identificar las preferencias de navegación a partir del acceso que realizan los usuarios de páginas web de la UNSAAC, los cuales se registran en el log.
- ✓ Demostrar que en archivos cercanos al big data, hay preferencias ocultas y valiosas, para este caso en particular.

CAPITULO II

MARCO TEORICO CONCEPTUAL

2.1 BASES TEORICAS FILOSOFICAS

2.1.1 MINERIA DE DATOS

Para (Hernández Orallo, Ramiréz Quintana, & Ferri Ramiréz, 2004, pág. 3), la minería de datos (DM) por sus siglas en inglés, no aparece por el desarrollo de tecnologías esencialmente diferentes a las anteriores, sino que se crea, en realidad, por la aparición de nuevas necesidades y, especialmente, por el reconocimiento de un nuevo potencial: el valor, hasta ahora generalmente infrautilizado, de la gran cantidad de datos almacenados informáticamente en los sistemas de información de instituciones, empresas, gobiernos y particulares. Los datos pasan de ser una "materia prima" que hay que explotar para obtener el verdadero "producto elaborado", el conocimiento; un conocimiento que ha de ser especialmente valioso para la ayuda en la toma de decisiones sobre el ámbito en el que se han recopilado o extraído los datos. Es cierto que, la estadística es la primera ciencia que considera los datos como su materia prima, pero las nuevas necesidades y, en particular, las nuevas características de los datos (en volumen y tipología) hacen que las disciplinas que integran lo que se conoce como "minería de datos" sean numerosas y heterogéneas.

Para (Han & Kamber, 2006) la minería de datos son técnicas que son aplicadas para realizar asociaciones, estimaciones, clasificación, y segmentación. El objetivo de la minería de datos es analizar datos para obtener patrones útiles y convertirlos a estructuras más comprensibles para complementar las aplicaciones.

De la misma forma (Han, Kamber, & Pei, 2012, pág. 8) amplían la definición diciendo que la minería de datos es, el proceso de descubrir patrones interesantes y por ende conocimiento, de grandes cantidades de datos. Las fuentes de datos pueden incluir bases de datos, DataWarhouse, la Web, otros repositorios de información, o datos que fluyen en sistemas dinámicos.

Por otro lado (Leskovec, Rajaraman, & Jeff, 2014, pág. 1) sostiene que la minería de datos es descubrir modelos de datos.

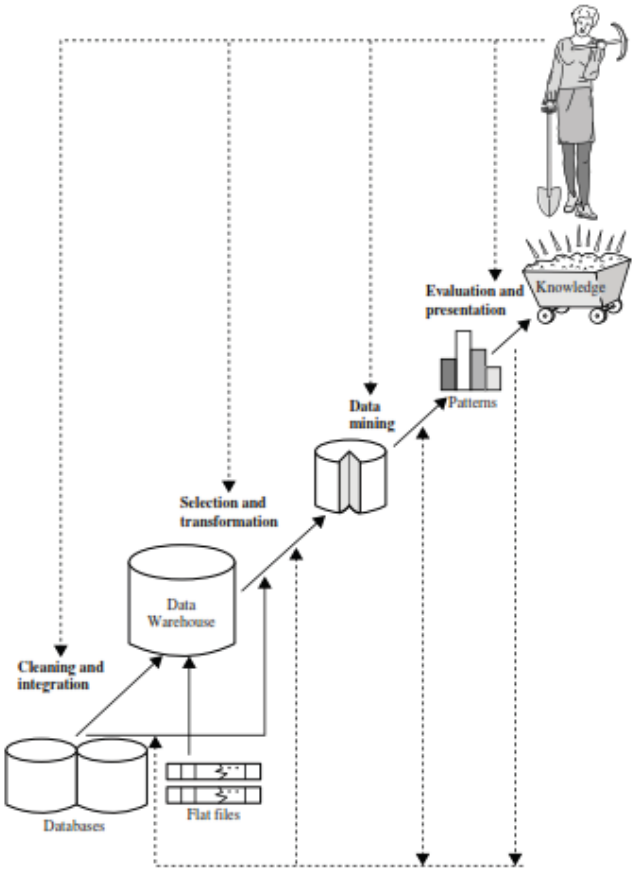

Figura 1. Data mining como un paso en el descubrimiento del conocimiento (Han & Kamber, 2006, pág. 7)

2.1.2 MINERIA DE DATOS COMPLEJO

Algunas clasificaciones por ser recientes encajarían dentro de este ítem, al respecto varios autores en sus textos realizan clasificaciones como se indica a continuación.

Para (Hernández Orallo, Ramiréz Quintana, & Ferri Ramiréz, 2004) :

- Minería de datos espaciales.
- Minería de datos temporales.
- Minería de datos multimedia.
- Minería web.
    - Minería del contenido de la web.
    - Minería de la estructura de la web.
    - Minería del uso de la web
- Sistemas de minería de web y textos.

Para (Han, Kamber, & Pei, 2012), las definen como tendencias y fronteras de investigación en data mining:

- Minería de secuencia de datos.
    - Series de tiempo.
    - Secuencia de símbolos.
    - Secuencias biológicas.
- Minería de datos en la estadística.
- Vistas en las profundidades de la minería de datos.
- Minería de datos en audio y multimedia.
- Minería de datos en análisis de datos financieros.
- Minería de datos en los retails e industrias de telecomunicaciones.
- Minería de datos en ciencias e ingeniería.
- Minería de datos para detectar y prevenir intrusiones.
- Minería de datos para sistemas de recomendación.

En cambio (Leskovec, Rajaraman, & Jeff, 2014), no realizan clasificaciones, pero abordan todo lo siguiente:

- Minería de datos en data streams.
- Análisis de links.
- Anuncios en la web.
- Sistemas de recomendación.
- Minería en redes sociales usando grafos, solo por mencionar algunos, pues varios ítems mencionados en el texto, encajan en las clasificaciones realizadas por los otros autores.

2.1.3 MINERIA WEB

(Hernández Orallo, Ramiréz Quintana, & Ferri Ramiréz, 2004) Que textualmente dice lo siguiente Etzioni [Etzioni 1996] definió la minería web como el uso de técnicas de minería de datos para descubrir y extraer información automáticamente desde el *Word Wide Web*.

(Sangeetha & Suresh, 2014) El termino minería web fue mencionado por Etzioni (1996) para denotar el uso de la minería de datos para automatizar y descubrir documentos de la web y servicios, extraer información de recursos de la web. La minería web es la aplicación de técnicas para extraer patrones interesantes, potencialmente útiles y todos estos implícitos en la información de los datos web.

(Han, Kamber, & Pei, 2012) La minería web es la aplicación de técnicas de minera de datos para descubrir patrones, estructuras y conocimiento de la web. De acuerdo al análisis objetivo, la minería web puede ser organizada en 3 áreas.

(Jha & Jaiswal, 2016) Definen la minería web como: la utilización de información mediante estrategias de minería para concentrar aprendizaje de la información de la Web. Donde se le pone énfasis en la variedad de información identificada por ID de invitados o acciones separadas por robots.

(Neelima & Rodda, 2016) Sostiene que la minería web es una de las técnicas de la minería de datos, para extraer información útil basada en las necesidades del usuario.

Todas estas referencias clasifican la minería web en tres grandes categorías, estas son la minera de la estructura web, minería de uso web y minería de contenido web.

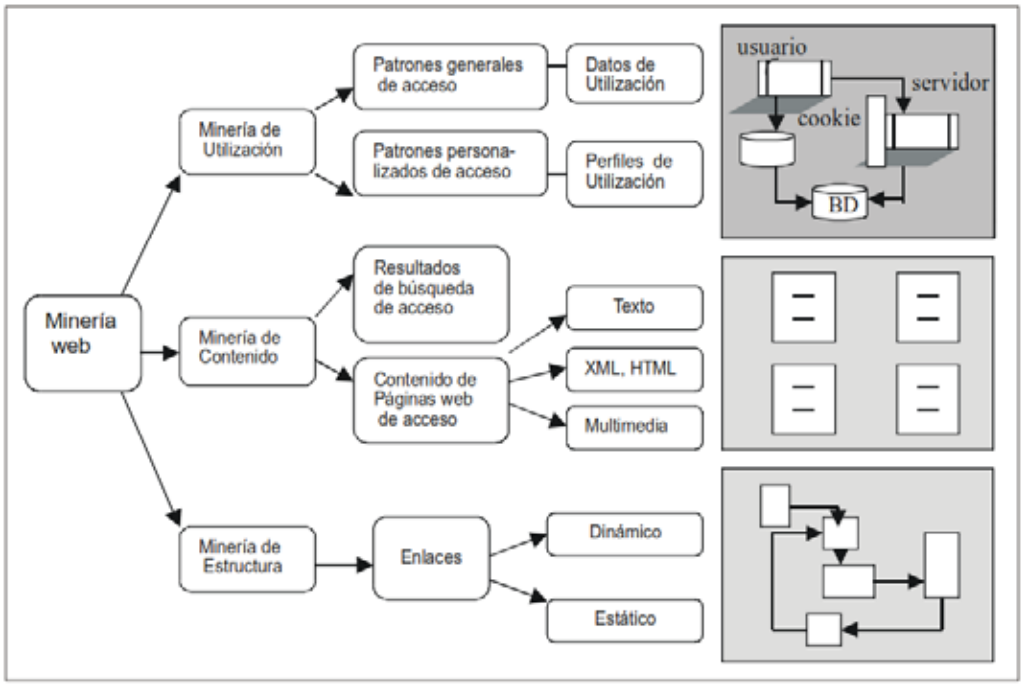

Figura 2. Categorías del web mining (UNMSM, 2005)

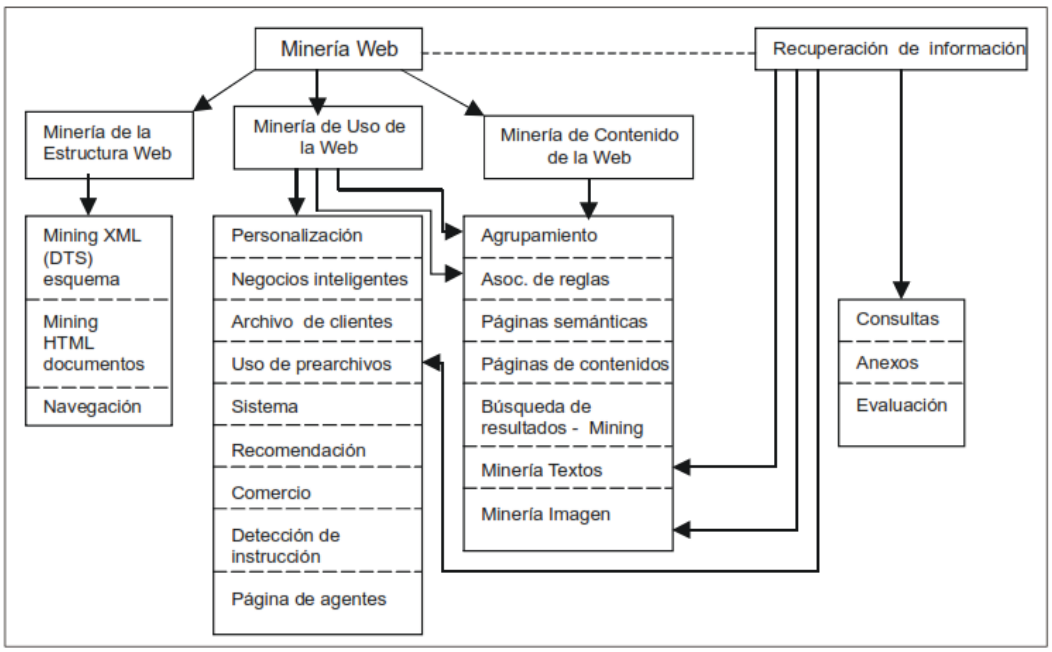

Figura 3. Taxonomia del web mining (UNMSM, 2005)

2.1.3.1 MINERIA DE CONTENIDO WEB

(Hernández Orallo, Ramiréz Quintana, & Ferri Ramiréz, 2004) La minería de contenidos web describe el descubrimiento de información útil desde los contenidos textuales y gráficos de los documentos web, y tiene sus orígenes en el procesamiento del lenguaje natural y en la recuperación de la información.

Analiza, por tanto, documentos más que los enlaces entre ellos. Los contenidos en la web han cambiado sustancialmente desde su origen. Al principio, Internet consistía en diferentes tipos de servicios y fuentes de datos: librerías digitales accesibles desde la web, las bases de datos de muchas empresas que ofrecen electrónicamente sus negocios y servicios, aplicaciones y sistemas que están siendo migrados a la web o emergen en este entorno.

De hecho, algunos de los datos en la web son ocultos ya que se generan dinámicamente o se obtienen como respuesta a preguntas cuyos datos residen en bases de datos privadas. Resumiendo, los contenidos en la web pueden ser de varios tipos: textual, imágenes, audio, video, meta-datos e hipervínculos y constan de datos no estructurados (textos), datos muy poco estructurados (como en los documentos HTML), datos semi-estructurados (como documentos XML) y datos más estructurados (como los contenidos en bases de datos generadas desde páginas HTML). Sin embargo, como la mayoría del contenido corresponde a texto no estructurado, por lo tanto, esta es el área más investigada.

(Han, Kamber, & Pei, 2012) Dice que, la minería de contenido web, analiza contenidos web como texto, datos multimedia, y datos estructurados (con páginas web o links entre páginas web). Esto se da para entender los contenidos de páginas web, proveyendo información basada en claves y escalas indexadas, con resolución de entidad / concepto, páginas web relevantes y como está en el ranking, resúmenes de contenidos de la página web, y otros valores relacionados a la búsqueda y análisis.

(Jha & Jaiswal, 2016) Mencionan que alude a la revelación de datos durante el cual las primeras cosas que se miden en acumulaciones tradicionales de registros de medios interactivos, para instancias de contenido, imágenes, audios que son unidades que usualmente tienen las páginas en línea.

2.1.3.2 MINERIA DE ESTRUCTURA WEB

(Jiang, K. Leung, & Pazdor, 2016) Menciona, que la minería de estructura web, es aquella que se enfoca en crear una estructura de grafos conectando diferentes páginas en la web mediante hyperlinks.

Al respecto (Han, Kamber, & Pei, 2012) dice que la minería de estructura web, es el proceso de usar grafos y teoría de minado de redes y métodos para analizar los nodos y conexiones estructuradas en la web. Se extraen patrones de hyperlinks, donde un hyperlink es una estructura compuesta que conecta páginas web. También puede minar documentos estructurados (por ejemplo, analizar la estructura tipo árbol de páginas descritas en HTML o etiquetas XML). Ambas clases de la minería de estructura web ayudan a entender los contenidos web y podrían coadyuvar a transformar contenidos web a estructuras conocidas.

(Hernández Orallo, Ramiréz Quintana, & Ferri Ramiréz, 2004) Indica que la minería de estructura web trata de descubrir el modelo subyacente a la estructura de los enlaces web y analiza, fundamentalmente, la topología de los hipervínculos (con o sin descripción de los enlaces). Este modelo se puede usar para categorizar páginas web y es útil para generar información como la similitud y relación entre diferentes sitios web, así como para detectar paginas principales y paginas concentradores (que apuntan a páginas principales), estudiar topologías, etc.

2.1.3.3 MINERIA DE USO WEB

Para (Han, Kamber, & Pei, 2012) la minería de uso web es el proceso de extraer información útil (por ejemplo, cliks de usuarios) de los log de servidores. Encontrar patrones relacionados de lo general a lo particular en función a grupos de usuarios, búsqueda de patrones para entender a usuarios, tendencias asociaciones; y predecir que paginas requieren los usuarios. Ayuda a mejorar la búsqueda de eficiencia y efectividad, esto para promover productos relacionados a diferentes grupos de usuarios en un determinado instante. Compañías de búsquedas web encaminan la minería de uso web a mejorar la calidad de servicio.

(Hernández Orallo, Ramiréz Quintana, & Ferri Ramiréz, 2004) Indican que la minería de uso web es el proceso de analizar la información sobre los accesos web. A diferencia de las minerías de contenido y de estructura que usan datos reales sobre la web, la minería de uso web, mina datos secundarios derivados de la interacción de los usuarios, estos datos incluyen los archivos de logs de acceso al servidor, logs del navegador, logs de los servidores proxy, perfiles de usuario, datos de registro, sesiones y transacciones del usuario, cookies, preguntas del usuario, pulsos del ratón y desplazamientos por las páginas y en general cualquier otro dato fruto de la interacción.

La web tiene actualmente un papel importante en el mundo de los negocios. Cada vez más las empresas diseñan sus propios sitios web, que se están convirtiendo en el primer punto del contacto entre empresas y clientes. Lo que está generando la necesidad de conocer como los usuarios interactúan con estos sitios web. La minería del uso web captura las actividades de los usuarios durante su conexión y extrae información puede ayudar a comprender las preferencias de navegación de los usuarios o a mejorar futuras páginas, adaptando las interfaces de los sitios web a los usuarios en forma individual o más personalizada.

Cuando los usuarios interactúan en un sitio web los datos que registran su comportamiento se almacenan en los logs de los servidores web. Estos pueden llegar a almacenar en un sitio web de tamaño medio, varios megabytes por día.

Existen dos aproximaciones principales para minar los patrones de navegación de los usuarios desde los datos log:

- Transformar los datos a notación tabular y aplicar técnicas estándar de minería de datos, como la reglas de asociación [Chen et al. 1998].
- Desarrollar ad-hoc (desarrollo personalizado, apropiado, adecuado) para trabajar directamente con los datos log [Spiliopoulou et al. 1999].

Por otra parte, las aplicaciones de la minería de uso pueden clasificarse en:

- Aprendizaje de patrones de navegación.

- Aprendizaje de perfiles de usuario para modelar interfaces adaptativas (personalización).

2.1.4 EL PROCESO DE LA MINERIA WEB

Para (Hernández Orallo, Ramiréz Quintana, & Ferri Ramiréz, 2004) la minería web puede descomponerse en las siguientes subtareas:

1. Descubrimiento de las fuentes: localizar los documentos y servicios en la web.
2. Selección y pre-procesado de la información: extraer automáticamente información específica desde las fuentes web descubiertas.
3. Generalización: descubrir patrones generales desde los sitios web individuales, así como múltiples sitios.
4. Análisis: validación y/o interpretación de los patrones minados.

Para (Liu, 2012), el proceso de la minería web estaría representado en la siguiente gráfica.

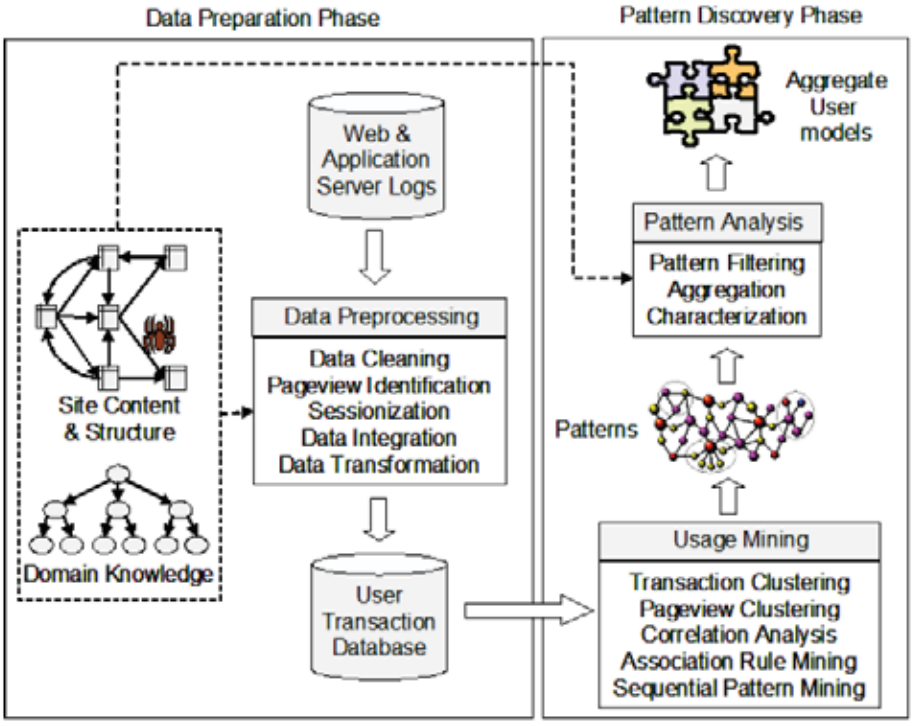

Figura 4. Proceso de la minería de uso web (Liu, 2012, pág. 528)

2.1.5 COLECCIÓN DE DATOS Y PRE PROCESAMIENTO

(Liu, 2012, pág. 528) Una importante tarea en cualquier aplicación de la minería de datos, es la creación de un buen conjunto de datos sobre el cual se pueda realizar minería y aplicar algoritmos estadísticos. Esta particularidad es importante en la minería de uso web para poder observar las relaciones y características que presentan las secuencias de links obtenidas de múltiples fuentes de datos.

El mismo autor también indica que, la preparación del proceso de datos en muchas ocasiones tiene un costo computacional elevado, además requiere del uso de algoritmos especiales y

heurísticas que generalmente no se emplean en otros dominios. Este proceso es crítico para la extracción exitosa de patrones de datos. El proceso podría incluir, pre procesamiento de los datos, integración de datos de diversas fuentes y transformación e integración de estos datos, en una forma adecuada, para realizar operaciones específicas de minería de datos. Generalmente se le conoce a este proceso como preparación de datos.

Muchas de las investigaciones y prácticas en el uso de preparación de datos se han enfocado en el pre procesamiento y la integración de estas fuentes de datos para realizar diferentes tipos de análisis. El uso de la preparación de datos presenta un número de cambios únicos los cuales pueden ser analizados por una variedad de algoritmos y técnicas heurísticas para las tareas de pre procesamiento como la fusión y limpieza de datos, identificación de sesiones de usuario, páginas vistas. La correcta aplicación de las técnicas de la minería de datos, a la minería de uso web, es dependiente de la correcta aplicación de las tareas de pre procesamiento

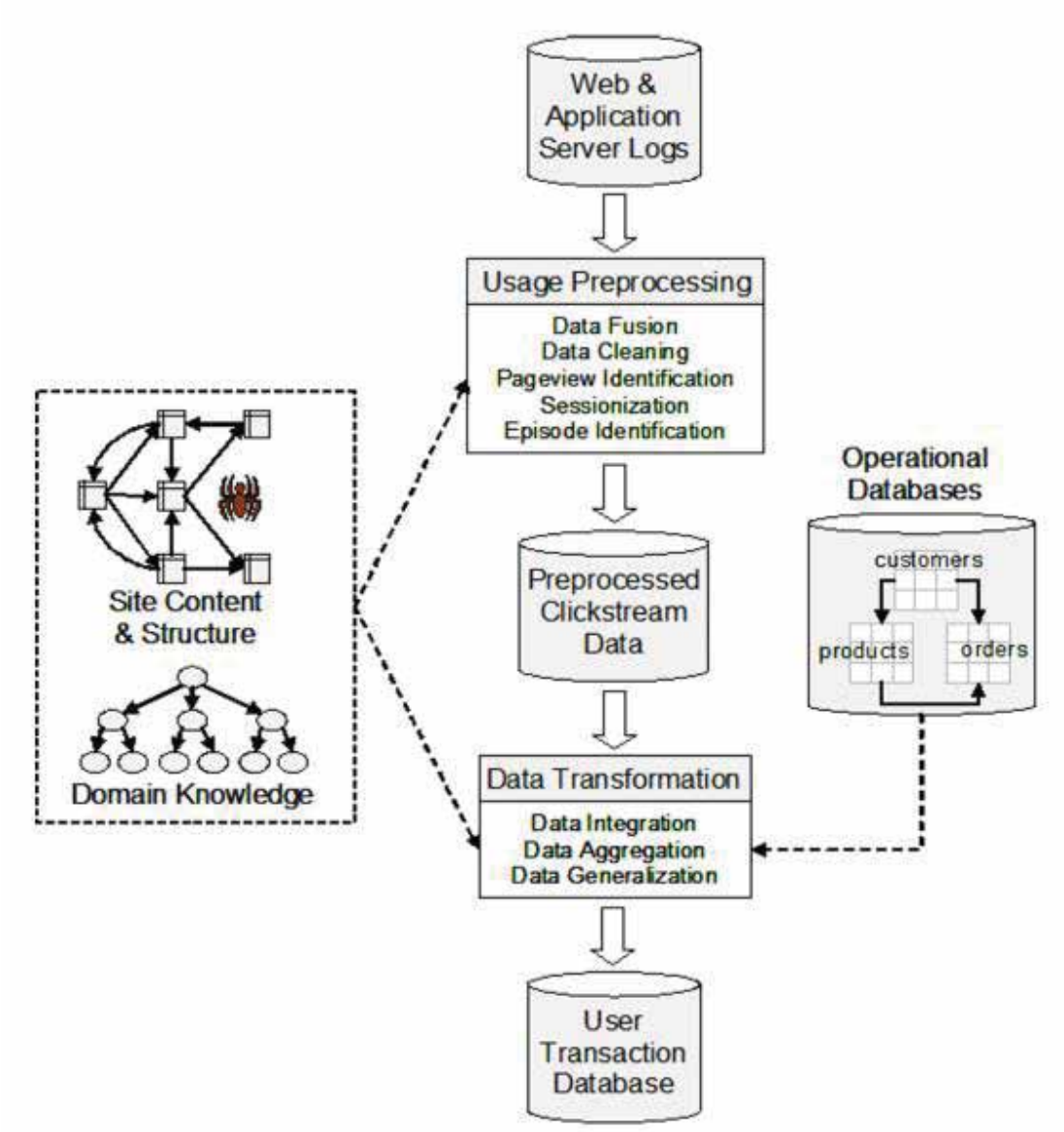

Figura 5. Pasos en la preparacion de datos en la web usage mining (Liu, 2012, pág. 529)

2.1.6 TIPOS DE WEB SERVER LOG

(Sharma & Yadav) Los log de servidores web, en realidad son textos planos semi estructurados, y manejan diversos formatos entre los diferentes softwares de servidores que se tiene en el mercado, pero generalmente tienen los siguientes tipos de log:

- Access Log
- Error Log
- Referrer Log
- Transfer Log

Se detalla a continuación de acuerdo a (Castaño Diaz, 2008) los tipos de Access log.

2.1.6.1 NCSA (Common Log Format)

Los formatos de log NCSA están basados en NCSA HTTP y están ampliamente aceptados entre los vendedores de servidores HTTP. El formato ofrece un alto grado de configuración, usándose formato de texto. Existen dos tipos de formato, común y combinado:

Común, contiene solamente información básica de los accesos, del recurso solicitado y algunas partes de información, pero no contiene la referencia ni, el cliente.

Campos que componen el formato:

host rfc931 username date:time request statuscode bytes

Un ejemplo de registro usando este formato:

125.125.125.125 - dsmith [10/Oct/1999:21:15:05 +0500] "GET /index.html HTTP/1.0" 200 1043

Combinado,este formato es una extensión del NCSA Común, contiene la misma información qu e elanterior y además añade 2 campos adicionales: la referencia y el cliente

host rfc931 username date:time request statuscode bytes referrer user_agent

Un ejemplo de formato combinado:

125.125.125.125 - dsmith [10/Oct/1999:21:15:05 +0500] "GET /index.html HTTP/1.0" 200 1043 "http://www.ibm.com/" "Mozilla/4.05 [en] (WinNT; I)" "USERID=CustomerA;IM PID=01234"

2.1.6.2 W3C EXTENDED (USED BY MICROSOFT IIS 4.0 AND 5.0)

Este formato de ficheros log es utilizado en por Microsoft Internet Information Server. Los cam posestán separados por espacios en blanco, si alguno de ellos no se utiliza se registra el símbolo "-"
como marca para omitirse. Los campos lo forman un prefijo y un identificador, separados por " -".

Campos:

date time c-ip cs-username s-ip cs-method cs-uri-stem cs-uri-query sc-status sc-bytes cs-bytes time-taken cs-version cs(User-Agent) cs(Cookie) cs(Referrer)

Ejemplo:

1998-11-19 22:48:39 206.175.82.5 - 208.201.133.173 GET /global/images/navlineboards.gif - 200 540 324 157 HTTP/1.0 Mozilla/4.0+(compatible;+MSIE+4.01;+Windows+95) USERID=CustomerA;+IMPID=01234 http://yourturn.rollingstone.com/webx?98@@webx1.html

2.1.6.3 Apache log file

(The Apache Software Foundation, 2018) El registro de acceso al servidor registra todas las solicitudes procesadas por el servidor. La ubicación y el contenido del registro de acceso están controlados por la directiva CustomLog. La directiva LogFormat se puede utilizar para simplificar la selección de los contenidos de los registros. Esta sección describe cómo configurar el servidor para registrar información en el registro de acceso.

Por supuesto, almacenar la información en el registro de acceso es solo el comienzo de la administración del registro. El siguiente paso es analizar esta información para generar estadísticas útiles. El análisis de registros en general está más allá del alcance de este documento, y no es realmente parte del trabajo del servidor web en sí. Para obtener más información sobre este tema y para las aplicaciones que realizan análisis de registro, consulte Open Directory o Yahoo.

Varias versiones de httpd de Apache han utilizado otros módulos y directivas para controlar el registro de acceso, incluidos mod_log_referer, mod_log_agent y la TransferLogdirectiva. La CustomLog directiva ahora incluye la funcionalidad de todas las directivas más antiguas.

El formato del registro de acceso es altamente configurable. El formato se especifica utilizando una cadena de formato que se parece mucho a una cadena de formato printf (1) de estilo C. Algunos ejemplos se presentan en las siguientes secciones. Para obtener una lista completa de los posibles contenidos de la cadena de formato, consulte la documentación de mod_log_config .

Formato de registro común, una configuración típica para el registro de acceso puede verse de la siguiente manera.

- **LogFormat "%h %l %u %t \"%r\" %>s %b" common**
- **CustomLog logs/access_log common**

Esto define el apodo common y lo asocia con una cadena de formato de registro particular. La cadena de formato consiste en el porcentaje de directivas, cada una de las cuales indica al servidor que registre una determinada información. Los caracteres literales también pueden colocarse en la cadena de formato y se copiarán directamente en la salida de registro. El carácter de comillas (") se debe escapar colocando una barra invertida antes para evitar que se interprete como el final de la cadena de formato. La cadena de formato también puede contener los caracteres de control especiales " \n" para la nueva línea y " \t" para la pestaña.

La CustomLogdirectiva establece un nuevo archivo de registro usando el apodo definido. El nombre de archivo para el registro de acceso es relativo al ServerRoot a menos que comience con una barra inclinada.

La configuración anterior escribirá entradas de registro en un formato conocido como el Formato de registro común (CLF). Este formato estándar puede ser producido por muchos servidores web diferentes y leído por muchos programas de análisis de registros. Las entradas del archivo de registro producidas en CLF se verán así:

- **127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326**

Cada parte de esta entrada de registro se describe a continuación.

127.0.0.1(%h)

Esta es la dirección IP del cliente (host remoto) que realizó la solicitud al servidor. Si HostnameLookups está configurado en On, entonces el servidor intentará determinar el nombre de host y registrarlo en lugar de la dirección IP. Sin embargo, esta configuración no se recomienda, ya que puede ralentizar significativamente el servidor. En cambio, es mejor usar un post-procesador de registro como logresolve para determinar los nombres de host. La dirección IP informada aquí no es necesariamente la dirección de la máquina en la que el usuario está sentado. Si existe un servidor proxy entre el usuario y el servidor, esta dirección será la dirección del proxy, en lugar de la máquina de origen.

-(%l)

El "guión" en la salida indica que la información solicitada no está disponible. En este caso, la información que no está disponible es la identidad RFC 1413 del cliente determinada por identdla máquina del cliente. Esta información es muy poco confiable y casi nunca se debe usar, excepto en redes internas estrictamente controladas. El httpd de Apache ni siquiera intentará determinar esta información a menos que se establezca IdentityCheckOn .

frank(%u)

Este es el ID de usuario de la persona que solicita el documento según lo determinado por la autenticación HTTP. El mismo valor generalmente se proporciona a los scripts CGI en la REMOTE_USERvariable de entorno. Si el código de estado para la solicitud (ver abajo) es 401, entonces este valor no debería ser confiable porque el usuario aún no está autenticado. Si el documento no está protegido por contraseña, esta entrada será " -" igual que la anterior.

[10/Oct/2000:13:55:36 -0700] (%t)

La hora en que el servidor terminó de procesar la solicitud. El formato es:

[day/month/year:hour:minute:second zone]

day = 2*digit

month = 3*letter

year = 4*digit

hour = 2*digit

minute = 2*digit

second = 2*digit

zone = (`+' | `-') 4*digit

Es posible hacer que la hora se muestre en otro formato especificando %{format}ten la cadena de formato de registro, donde formatestá como en strftime(3)la biblioteca estándar de C.

"GET /apache_pb.gif HTTP/1.0" (\"%r\")

La línea de solicitud del cliente se da entre comillas dobles. La línea de solicitud contiene una gran cantidad de información útil. En primer lugar, el método utilizado por el cliente es GET. Segundo, el cliente solicitó el recurso /apache_pb.gif, y tercero, el cliente usó el protocolo HTTP/1.0. También es posible registrar una o más partes de la línea de solicitud de forma independiente. Por ejemplo, la cadena de formato " %m %U%q %H" registrará el método, la ruta, la cadena de consulta y el protocolo, dando como resultado exactamente el mismo resultado que " %r".

200(%>s)

Este es el código de estado que el servidor envía de vuelta al cliente. Esta información es muy valiosa, ya que revela si la solicitud dio como resultado una respuesta exitosa (códigos que comienzan en 2), una redirección (códigos que comienzan en 3), un error causado por el cliente (códigos que comienzan en 4) o un error en el servidor (códigos que comienzan en 5). La lista completa de posibles códigos de estado se puede encontrar en la especificación HTTP (RFC2616 sección 10).

2326(%b)

La última entrada indica el tamaño del objeto devuelto al cliente, sin incluir los encabezados de respuesta. Si no se devolvió contenido al cliente, este valor será " -". Para registrar " 0" sin contenido, use %Ben su lugar.

Formato de registro combinado, otra cadena de formato comúnmente utilizada se llama Formato de registro combinado. Se puede usar de la siguiente manera.

- **LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" combined**
- **CustomLog log/acces_log combined**

Este formato es exactamente el mismo que el formato de registro común, con la adición de dos campos más. Cada uno de los campos adicionales usa el percent-directive , donde el encabezado puede ser cualquier encabezado de solicitud HTTP. El registro de acceso con este formato se verá así:%{header}i

- **127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326 "http://www.example.com/start.html" "Mozilla/4.08 [en] (Win98; I ;Nav)"**

Los campos adicionales son:

- **"http://www.example.com/start.html" (\"%{Referer}i\")**

El encabezado de solicitud HTTP "Referer" (sic). Esto le da al sitio que el cliente informa haber sido referido. (Esta debería ser la página que enlaza o incluye /apache_pb.gif).

- **"Mozilla/4.08 [en] (Win98; I ;Nav)" (\"%{User-agent}i\")**

El encabezado de solicitud HTTP de User-Agent. Esta es la información de identificación que el navegador del cliente informa sobre sí mismo.

2.1.7 REGLAS DE ASOCIACION Y PATRONES SECUENCIALES

(Liu, 2012) Las reglas de asociación son una importante clase de regularidades en datos, la minería de reglas de asociación son una fundamental tarea de la minería de datos. Esta es la razón por la cual es un importante modelo inventado y extensamente estudiado por las bases de datos y la comunidad de minería de datos. El objetivo es encontrar todas las ocurrencias o asociaciones entre los ítems analizados. Una aplicación clásica de la minería de reglas de asociación es el del análisis de datos de una market basket (canasta de compras) que ayuda a descubrir que ítems que compran los clientes en un supermercado están asociados. Un ejemplo de regla de asociación es el siguiente: Queso → Cerveza [support=10%, confidence=80%], esta regla nos dice que el 10% de clientes que compra queso también compra cerveza el 80% de las veces.

Estos modelos de minería de hecho son muy generales y pueden ser usados en muchas aplicaciones. Por ejemplo, en el contexto de la web y los documentos de texto, pueden ser usadas para encontrar ocurrencias y relaciones entre palabras y patrones de uso web.

Pero la minería de reglas de asociación no considera la secuencia en el cual los ítems fueron comprados o llevados. La minería de patrones secuenciales toma en consideración esto. Un ejemplo de un patrón secuencial es que el "5% de clientes que compraron primero una cama luego compraron un colchón y luego compraron almohadas". Los ítems no son comprados en el mismo instante, pero si uno después de otro. Como los patrones son útiles en la minería de uso web para analizar la secuencia de clics en los servidores web. Son también útiles para encontrar lenguajes o patrones lingüísticos de textos de lenguaje natural.

2.1.7.1 CONCEPTOS BÁSICOS DE REGLAS DE ASOCIACIÓN

(Liu, 2012) Los problemas de la minería de reglas de asociación pueden ser definidos como sigue: Sea $I = \{i_1, i_2, \dots, i_m\}$ son el conjunto de ítems. Sea T = $(t_1, t_2, \dots, t_n)$, un set de transacciones (base de datos), donde cada transacción t_i es un conjunto de ítems tal que $t_i \subseteq I$, una regla de asociación es una implicación de la forma: X → Y, donde X ⊂ I, Y ⊂ I, y X ∩ Y = ∅. X (o Y) es un conjunto de ítems.

Ejemplo 1: Queremos analizar como los ítems comprados en un supermercado son relacionados unos con otros. I es el conjunto de ítems comprados en el supermercado. Una transacción es un simple conjunto de ítems comprados por un cliente. La transacción podría ser: {carne, pollo, queso}, lo que significa que el cliente tiene 3 ítems en su canasta de compra, carne,

pollo y queso. Una regla de asociación podría ser: $carne, pollo \rightarrow queso$, donde {carne, pollo} es X y {queso} es Y.

Una transacción $t_i \in T$ es decir cuenta con un conjunto de ítems X si X es un subconjunto de t_i. El soporte (support count) de X en T es el número de transacciones en T que contiene X. La relevancia de la regla esta definida por su soporte (support) y su confianza (confidence).

2.1.7.1.1 SUPPORT

El soporte de una regla, $X \rightarrow Y$, es el porcentaje de transacciones en T que cuentan con X ∪ Y, y pueden ser estimados como la probabilidad Pr(X ∪ Y). El soporte de las reglas de asociación se determina como la frecuencia con que la regla es aplicable a la transacción T. Sea n el número de transacciones en T. El soporte de la regla $X \rightarrow Y$, es calculada como sigue:

$$support = \frac{(X \cup Y).count}{n}$$

El soporte es una medida importante porque si es demasiado bajo, la regla podría ocurrir por casualidad. En un entorno empresarial, una regla que cubra solo algunos casos o transacciones podría no ser de mucha utilidad, pues no es muy conveniente hacer negocios o actuar en base a una regla poco rentable.

2.1.7.1.2 CONFIDENCE

La confianza de una regla, $X \rightarrow Y$, es el porcentaje de transacciones en T que cuentan con X y también cuentan con Y . Esto es la estimación de la probabilidad condicional, Pr(X | Y) es calculada como sigue:

$$confidence = \frac{(X \cup Y).count}{X.count}$$

La confianza determina la predictibilidad de la regla. Si la confianza de una regla es demasiado baja, no puede seguramente inferir o predecir $Y\ de\ X$. Una regla con baja predictibilidad en de uso limitado.

Objetivo. - dado un conjunto de transacciones T, el problema de minería de reglas de asociación es descubrir todas las reglas en T que tienen soporte y confianza mayor o igual al soporte mínimo (minsup) especificada por el usuario y confianza mínima (minconf).

Ejemplo (Liu, 2012, pág. 19): se tiene un conjunto de 7 transacciones. Cada transacción t_i es un conjunto de ítems comprados en una cesta en una tienda por un cliente. El conjunto I, es un conjunto de todos los ítems comprados en la tienda.

t_1: $carne, pollo, leche$
t_2: $carne, queso$
t_3: $queso, botas$
t_4: $carne, pollo, queso$
t_5: $carne, pollo, ropa, queso, leche$
t_6: $pollo, ropa, leche$
t_7: $pollo, leche, ropa$

Figura 6 .- Un ejemplo de transacciones

Tenemos las especificaciones del usuario, minsup=30% y minconf=80%, la siguiente regla de asociación, **pollo, ropa → leche** $[sup = 3/7\ , conf = 3/3]$ es válida pues su soporte es 42.86% (> 30%) y su confianza es 100% (>80%). La siguiente regla también es válida, donde el consecuente tiene 2 ítems: **ropa → leche, pollo** $[sup = 3/7\ , conf = 3/3]$, claramente muchas reglas de asociación pueden ser descubiertas.

Podemos apreciar que la presentación de los datos en la transacción es simple, la cantidad y el precio no son considerados en este modelo.

También podemos notar que un documento de texto o una oración en un simple documento de texto pueden ser tratados como una transacción sin considerar secuencias de palabras y numero de ocurrencias de cada palabra. Dado un conjunto de documentos o de oraciones podemos encontrar palabras relacionadas.

Un gran número de algoritmos de minería de reglas de asociación, han sido desarrollados, los cuales tienen diferentes minados eficientes. Dando como resultado conjunto de reglas, pero, todos ellos basados en la definición de reglas de asociación. Que dado un conjunto de transacciones de datos T, con soporte y confianza mínima, el conjunto de reglas de asociación existentes en T es determinado en forma única. Cualquier algoritmo debería encontrar el mismo conjunto de reglas más allá de su eficiencia computacional y requerimientos de memoria podrían ser diferentes el algoritmo mejor conocido es el de Apriori.

2.1.7.2 ALGORITMO DE APRIORI

(Liu, 2012) Este algoritmo trabaja en dos pasos:

1.- genera todo los conjuntos de ítems frecuentes: un conjunto de ítems frecuentes es un conjunto de ítems cuya transacción tiene un soporte por encima del minsup.

2.- generar todas las confianzas de las reglas de asociación de los conjuntos de ítems frecuentes: la confianza de una regla de asociación es una regla con confianza por encima de minconf.

Llamaremos al número de ítems en un conjunto, su tamaño, y a un conjunto de ítems de tamaño k, k-itemset. Por ejemplo, $\{pollo, ropa, leche\}$ es una frecuencia de 3-itemset como su soporte

es de $3/7$ (minsup=30%). Del conjunto de items, podemos generar las siguientes tres reglas de asociación (minconf=80%):

Regla 1: pollo, ropa → leche $[\boldsymbol{sup} = \boldsymbol{3/7}\ , \boldsymbol{conf} = \boldsymbol{3/3}]$

Regla 2: ropa, leche → pollo $[\boldsymbol{sup} = \boldsymbol{3/7}\ , \boldsymbol{conf} = \boldsymbol{3/3}]$

Regla 3: ropa → leche, pollo $[\boldsymbol{sup} = \boldsymbol{3/7}\ , \boldsymbol{conf} = \boldsymbol{3/3}]$

2.1.7.2.1 GENERACION DE CONJUNTO ITEMS FRECUENTES

El algoritmo Apriori se basa en la propiedad a priori o **downward closure**, para generar eficientemente todos los conjuntos de ítems frecuentes.

Propiedad downward closure: Si un conjunto de ítems tiene un soporte mínimo, entonces todos los sub conjuntos no vacíos de este conjunto también tienen soporte mínimo.

La idea es simple porque si una transacción contiene un conjunto de ítems X, entonces estos deberían contar con algún sub conjunto no vacío de X. Esta propiedad y el minsup threshold podan un gran número de conjuntos de ítems que no pueden ser presentes.

Para asegurarnos de la eficiencia de la generación de conjunto de ítems, el algoritmo asume que los items en I están ordenados en **orden lexicográfico** (orden total). El orden es utilizado por el algoritmo en cada conjunto. Una notación utilizada es la siguiente $\{w[1], w[2], \dots, w[k]\}$ para representar un k-itemset, w se compone de items $w[1], w[2], \dots, w[k]$ donde $w[1] < w[2] < \dots, w[k]$, teniendo de esta forma un orden total.

El algoritmo de Apriori para generar conjuntos de items frecuentes, se basa en la búsqueda inteligente por niveles (level-wise search). Esto genera todos los items frecuentes para realizar multiples pasadas por la data.

```
Algorithm Apriori(T)
1  C1 ← init-pass(T);
2  F1 ← {f|f ∈ C1, f.count/n ≥ minsup};
3  for(k = 2; Fk−1 ≠ ∅; k = k + 1) do
4     Ck ← candidate-gen(Fk−1 );
5     for each transaction t ∈ T do
6        for each cadidate c ∈ Ck do
7           if c is contained in t then
8              c.count + +;
9        endfor
10    endfor
11    Fk ← {c ∈ Ck | c.count/n ≥ minsup};
12 endfor
13 return  F ← Uk Fk ;
```

Figura 7.- Algoritmo Apriori para la generacion de items frecuentes (Liu, 2012, pág. 21)

En el primer paso, se cuenta los soportes de cada ítem (línea 1) y determina si cada uno de ellos es frecuente (línea 2). F_1 es el conjunto de 1-itemsets frecuentes. En cada sub secuencia pasada k, hay 3 pasos.

1. Comienza con el conjunto semilla de ítems F_{k-1} encontrados, que son frecuentes en las $(k-1)$ pasadas. Usa estas semillas para generar **conjunto de ítems candidatos** C_k (línea 4), los cuales son posibles conjuntos de ítems frecuentes. Esto es gracias al uso de la función **candidate-gen().**
2. Las transacciones de la base de datos son luego escaneadas y después el soporte actual de cada candidatos del conjunto de ítems $\boldsymbol{c}$ en $\boldsymbol{C_k}$ es contado (líneas 5-10). Notamos que no necesitamos cargar todos los datos en memoria antes del proceso. En lugar de perder tiempo con dicha tarea, solo una de las transacciones reside en memoria. Esto es una característica importante del algoritmo. Esto hace al algoritmo escalable para grandes conjuntos de datos, los cuales no pueden ser cargados en memoria.
3. En la última pasada, se determina cuál de los conjunto de ítems candidatos son frecuentes actualmente (línea 11).

La salida final del algoritmo es el conjunto $\boldsymbol{F}$ de todos ítems frecuentes (línea 13).

```
Function candidate-gen(F_{K-1})
1   C_k ← ∅;
2   forall f_1, f_2 ∈ F_{k-1}
3       with f_1 = {i_1, ..., i_{k-2}, i_{k-1}}
4       and f_2 = {i_1, ..., i_{k-2}, i'_{k-1}}
5       and i_{k-1} < i'_{k-1} do
6           c = {i_1, ..., i_{k-2}, i_{k-1}, i'_{k-1}}
7           C_k ← C_k ∪ {c};
8           for each (k-1)-subset s of c do
9               if (s ∉ F_{k-1}) then
10                  delete c from C_k ;
11          endfor
12  endfor
13  return C_k;
```

Figura 8.- Función Candidate-gen (Liu, 2012, pág. 21)

2.1.7.2.2 FUNCIÓN CANDIDATE-GEN

La función de generación de candidatos, consiste en dos pasos, el **join step** y el **pruning step**.

Join step (líneas 2-6): este paso une o junta dos $(\boldsymbol{k-1})-\boldsymbol{itemset}$ frecuentes para producir un posible candidato $\boldsymbol{c}$ (línea 6). Los dos itemset frecuentes $\boldsymbol{f_1}$ y $\boldsymbol{f_2}$ tienen exactamente los mismos ítems excepto el ultimo (líneas 3-5). $\boldsymbol{c}$ es añadido al conjunto de candidatos $\boldsymbol{C_k}$ (línea 7).

Pruning step (líneas 8-11): un candidato $\boldsymbol{c}$ del **join step**, podría no ser un candidato final. Este paso determina si todos los $\boldsymbol{k-1}$ sub conjuntos de $\boldsymbol{c}$ están en $\boldsymbol{F_{k-1}}$. Si alguno de ellos no está

en F_{k-1}, c no puede ser frecuente de acuerdo a la propiedad ***downward closure,*** y este es eliminado de en C_k.

Ilustraremos mediante un ejemplo, cómo trabaja la función.

(Liu, 2012, pág. 22) **Ejemplo 3**: sea el conjunto de ítems frecuentes de 3 niveles

$$F_3 = \{\{1,2,3\},\{1,2,4\},\{1,3,4\},\{1,3,5\},\{2,3,4\}\}$$

Por simplicidad se utilizará números para representar a los ítems. El ***join step*** (el cual genera candidatos de nivel 4) producirá dos candidatos del conjunto de ítems, $\{1,2,3,4\}$ y $\{1,3,4,5\}$. $\{1,2,3,4\}$ es generado por la unión del primer y segundo conjunto de ítems en F_3 como su primer y segundo ítems son los mismos respectivamente. $\{1,3,4,5\}$ es generado por la unión de $\{1,3,4\}$ y $\{1,3,5\}$. Después del ***pruning step***, tenemos solo: $C_4 = \{\{1,2,3,4\}\}$, porque $\{1,4,5\}$ no está en F_3 y así $\{1,3,4,5\}$ no puede ser frecuente.

(Liu, 2012, pág. 22) **Ejemplo 4**: veamos ejemplo completo del algoritmo Apriori, basado en transacciones (Figura 6), usaremos un minsup=30%.

$$F_1 = \{\{carne\}:4,\{queso\}:4,\{pollo\}:5,\{ropa\}:3,\{leche\}:4\}$$

Nota. - el número después de cada conjunto de ítems frecuentes es la cantidad de soporte del conjunto de ítems, el número de transacciones. Un minsuport es 3 porque el soporte 3/7 es más del 30%, donde 7 es el número total de transacciones.

$$C_2 = \left\{\begin{matrix}\{carne,queso\},\{carne,pollo\},\{carne,ropa\},\{carne,leche\},\{queso,pollo\},\\ \{queso,ropa\},\\ \{queso,leche\},\{pollo,ropa\},\{pollo,leche\},\{ropa,leche\}\end{matrix}\right\}$$

$$F_2 = \left\{\begin{matrix}\{carne,pollo\}:3,\{carne,queso\}:3,\\ \{pollo,ropa\}:3,\{pollo,leche\}:4,\{ropa,leche\}:3\end{matrix}\right\}$$

$$C_3 = \{\{pollo,ropa,leche\}\}$$

Nota.- $\{carne,queso,leche\}$ es también producido en la línea 6 (Figura 8 **)** pero $\{queso,leche\}$ no están en F_2 por consiguiente el conjunto de ítems $\{carne,queso,pollo\}$ no está incluido en C_3.

$$F_3 = \{\{pollo,ropa,leche\}:3\}$$

Finalmente, algunas observaciones sobre el algoritmo de Apriori son las siguientes:

- Teóricamente, es un algoritmo exponencial. Donde el número de ítems en I es m. El espacio de todos los ítems es $O(2^m)$ porque cada ítem podría o no estar en un conjunto de ítems. Sin embargo, el algoritmo de minería explota la poca densidad (**sparseness**) de los datos y su gran valor de soporte para hacer posible y eficiente la minería. El

sparseness de los datos en el contexto del análisis de la canasta de compras significa que almacenan las ventas de muchos ítems, pero cada compra solo lleva pocos de esos ítems.

- El algoritmo puede escalar a un gran conjunto de datos, así como no cargar todos ellos en memoria. Esto solo se corre $\boldsymbol{k}$ veces, donde $\boldsymbol{k}$ es el tamaño del conjunto de ítems más grande. En la práctica, $\boldsymbol{k}$ es algunas veces pequeño (< 10). Esta propiedad es muy importante pues muchos conjuntos de datos del mundo real son tan grandes que no se pueden cargar en memoria principal.
- El algoritmo se basa en búsqueda por niveles. Tiene la flexibilidad para detenerse en cualquier nivel. Esto en muy útil en la práctica pues muchas aplicaciones donde los conjuntos de ítems frecuentes son muy grandes o las reglas son demasiadas grandes para su uso.
- Como se mencionó, una transacción $\boldsymbol{T}$, un minsup y minconf son dados, el conjunto de ítems frecuentes que pueden ser encontrados son únicamente determinados en $\boldsymbol{T}$. Cualquier algoritmo debería encontrar el mismo conjunto de ítems frecuentes. Esta propiedad sobre la minería de reglas de asociación no es la que se espera como resultado por muchas otras tareas de la minería de datos, por ejemplo, clasificación o clustering, para los cuales diferentes algoritmos pueden producir muchos y muy variados resultados.
- El principal problema con la minería de reglas de asociación es que a veces producen un gran número de conjuntos de ítems (y también reglas), decenas de cientos o más, los cuales dificultan su análisis y de esta forma no son tan útiles. Esto es llamado ***interestingness problem.***

Una eficiente implementación de el algoritmo de Apriori, incluyen sofisticadas estructuras de datos y técnicas de programación.

2.1.7.3 GENERACION DE REGLAS DE ASOCIACION

En muchas aplicaciones, los conjuntos de ítems frecuentes son útiles y suficientes. Entonces, no necesitamos generar reglas de asociación. En aplicaciones donde las reglas son las deseadas, se usarán conjuntos de ítems frecuentes para generar todas las reglas de asociación.

Comparado con la generación del conjunto de ítems frecuentes, las reglas de asociación son relativamente más simple. Para generar reglas de todos los ítems frecuentes $\boldsymbol{f}$, usamos todos los subconjuntos de $\boldsymbol{f}$. Para cada subconjunto $\boldsymbol{\alpha}$, vemos la regla de la siguiente forma

$$(\boldsymbol{f}-\boldsymbol{\alpha}) \rightarrow \boldsymbol{\alpha}, \qquad \boldsymbol{if}$$

$$\boldsymbol{confidence} = \frac{\boldsymbol{f}.\boldsymbol{count}}{\boldsymbol{(f-\alpha).count}} \geq \boldsymbol{minconf},$$

donde $\boldsymbol{f}.\boldsymbol{count}$ (o $(\boldsymbol{f}-\boldsymbol{\alpha}).\boldsymbol{count}$) es la cantidad de soporte de $\boldsymbol{f}.\boldsymbol{count}$ (o $(\boldsymbol{f}-\boldsymbol{\alpha}).\boldsymbol{count}$). El soporte de las reglas es $\boldsymbol{f}.\boldsymbol{count}/\boldsymbol{n}$, donde $\boldsymbol{n}$ es el número de transacciones en el conjunto $\boldsymbol{T}$. Todos los soportes cuentan necesariamente con la confianza calculada y disponible porque si $\boldsymbol{f}$ es frecuente, entonces alguno de sus subconjuntos no vacíos es también frecuente y su soporte cuenta con un registro en el proceso de minería.

Esta exhaustiva generación de reglas es estratégica, pero, ineficiente. Para diseñar un algoritmo eficiente, se observa que el soporte de $\boldsymbol{f}$ debajo de la confianza no cambia como cambia $\boldsymbol{\alpha}$. Esto se debe a que para mantener una regla $(\boldsymbol{f}-\boldsymbol{\alpha}) \rightarrow \boldsymbol{\alpha}$, todas las reglas de la forma $(\boldsymbol{f}-\boldsymbol{\alpha}_{\boldsymbol{sub}}) \rightarrow$

α_{sub} **,** deberían también mantenerse, donde α_{sub} es un subconjunto no vacío de α, porque el soporte de $(f - \alpha_{sub})$ debería ser menor o igual al soporte de $(f - \alpha)$**.** Por ejemplo dado un conjunto de ítems $\{A, B, C, D\}$**,** si la regla $\{A, B \rightarrow C, D\}$ se mantiene**,** entonces las reglas $\{A, B, D \rightarrow C\}$ y $\{A, B, C \rightarrow D\}$ deberían también mantenerse.

Dado un conjunto de ítems frecuentes f , si una regla con consecuente α se mantiene**,** entonces estas reglas con sus consecuentes son subconjuntos de α**.** Esto es similar a la ***downward closure property*** donde si un conjunto de ítems es frecuente, entonces también lo son sus subconjuntos.

```
Algorithm genRules(F)
1   for each frecuent k-itemset f_k in F , k ≥ 2 do
2      output every 1 − item consequent rule of f_k with confidence
                ≥ minconf and support ← f_k .count/n
3      H_1 ← {concequents of all 1
                − item consequent rules derived from f_k above};
4      ap-genRules(f_k , H_1 );
5   endfor

Procedure ap-genRules(f_k , H_m )
1   if (k > m + 1)AND ( H_m ≠ ∅ ) then;
2      H_{m+1} ← candidate-gen(H_m );
3      for each h_m in H_{m+1} do
4         conf ← f_k .count/(f_k − h_{m+1} ) .count;
5         if (conf > minconf) then
6            output the rule (f_k − h_{m+1} ) → h_{m+1} with confidence
                = conf and support = f_k .count/n ;
7         else
8            delete h_{m+1} from H_{m+1} ;
9      endfor
10     ap-genRules(f_k , H_{m+1} );
11  endif;
```

Figura 9.- Algoritmo de generación de reglas de asociación (Liu, 2012, pág. 23)

Dicho esto, de los conjuntos de ítems frecuentes f**,** primero generamos todas las reglas con un ítem en el consecuente. Luego usamos los consecuentes de estas reglas y la función **(Figura 8.- Función Candidate-gen)** para generar todos los posibles consecuentes con dos ítems que pueden aparecer en una regla. Un algoritmo usado para representar esta idea en el de la (***Figura 9.- Algoritmo de generación de reglas de asociación***). Notamos que todas las reglas que tienen un solo consecuente son los generados en primer lugar, en la línea 2 de la función (**Figura 9.- Algoritmo de generación de reglas de asociación genRules()**.

(Liu, 2012, pág. 25) **Ejemplo 4.-** basado en transacciones (Figura 6), usaremos un minsup=30% y minconf = 80%. El conjunto de ítems frecuentes es como sigue (ver ejemplo 3)

$$F_1 = \{\{carne\}: 4, \{queso\}: 4, \{pollo\}: 5, \{ropa\}: 3, \{leche\}: 4\}$$

$$F_2 = \left\{ \begin{matrix} \{carne, pollo\}: 3, \{carne, queso\}: 3, \\ \{pollo, ropa\}: 3, \{pollo, leche\}: 4, \{ropa, leche\}: 3 \end{matrix} \right\}$$

$$F_3 = \{\{pollo, ropa, leche\}: 3\}$$

Nosotros usamos solo el conjunto de ítems de F_3, para generar reglas (generamos las reglas de cada conjunto de ítems de F_2 de la misma forma). El conjunto de ítems en F_3 generan las siguientes posibles reglas con un consecuente.

Regla 1: pollo, ropa → leche $[sup = 3/7\ , conf = 3/3]$

Regla 2: pollo, leche → ropa $[sup = 3/7\ , conf = 3/4]$

Regla 3: ropa, leche → pollo $[sup = 3/7\ , conf = 3/3]$

Debido al requerimiento de minconf, solo la Regla 1 y Regla 3 son generadas a la salida de la línea 2 del algoritmo de **genRules().** Así, $H_1 = \{\{pollo\}, \{leche\}\}$**.** La función **ap-genRules()** es llamada entonces. En la línea 2 de **ap-genRules()** producimos $H_2 = \{\{pollo\}, \{leche\}\}$**.** La siguiente regla es generada entonces:

Regla 4: ropa → leche, pollo $[sup = 3/7\ , conf = 3/3]$

De esta forma, tres reglas de asociación son generadas del conjunto de ítems frecuentes $\{pollo, ropa, leche\}$ en F_3, denominadas **Regla 1, Regla 3** y **Regla 4**.

2.1.7.4 FORMATOS DE DATOS PARA LA MINERIA DE REGLAS DE ASOCIACION

(Liu, 2012) Hasta aquí, nosotros hemos usado solo transacciones de datos para la minería de reglas de asociación. Los conjuntos de datos de la canasta de compras son presentados naturalmente en este formato. Los documentos de textos pueden ser presentados como las transacciones de datos. Cada documento es una transacción, y cada palabra distinta es un ítem. Las palabras duplicadas son removidas.

Pero, también puede ser representada en tablas relacionales. Solo necesitamos convertir las transacciones a tablas, los cuales son bastante sencillos si cada atributo en la tabla toma valores **categóricos**. Simplificando el cambio a un par **clave-valor**.

Attribute1	Attribute2	Attribute3
a	a	x

b	n	y

$$t_1: (Attribute1, a), (Attribute2, a), (Attribute3, x)$$

$$t_2: (Attribute1, b), (Attribute2, n), (Attribute3, y)$$

Figura 10.- De tabla de datos a transaccion de datos. (Liu, 2012, pág. 27)

Ejemplo 5: la tabla de datos en la **Figura 10.- De tabla** de datos a transaccion de datos.pueden ser convertidos en una transacción de datos en la **Figura 10 (B)**. Cada par atributo-valor es considerado como un **ítem**.

Usando solamente valores no es suficiente en la forma de la transacción porque diferentes atributos podrían tener los mismos valores. Por ejemplo, sin incluir los nombres de los atributos, el valor **a** de los **atributo1** y **atributo2** no se distinguen. Después de la conversión, **Figura 10 (B)** pueden ser usados para la minería.

Si un atributo toma valores numéricos, esto puede volverse complejo. Se requiere primero discretizar su rango de valores en intervalos, y tratar cada intervalo como un valor categórico. Por ejemplo, los valores de los atributos están en el rango de 1-100. Podríamos dividirlos en 5 intervalos del mismo tamaño, 1-20, 21-40, 41 -60, 61-80, 81-10. Cada intervalo es luego tratado como un valor categórico. La discretizacion puede ser hecha manualmente basado en conocimiento experto o automático.

2.2 MARCO CONCEPTUAL

2.2.1 BASES DE DATOS

Colección o depósito de datos integrados, con redundancia controlada y con una estructura que refleje las interrelaciones y restricciones existentes en el mundo real; los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de éstas, y su definición y descripción, únicas para cada tipo de datos, han de estar almacenadas junto con los mismos. Los procedimientos de actualización y recuperación, comunes y bien determinados, habrán de ser capaces de conservar la integridad, seguridad y confidencialidad del conjunto de los datos (Llanos Ferrais, 2007, pág. 272).

2.2.2 Cookies

(Whalen, 2018) Una galleta, galleta informática o cookie es una pequeña información enviada por un sitio web y almacenada en el navegador del usuario, de manera que el sitio web puede consultar la actividad previa del usuario.

Sus principales funciones son:

- Llevar el control de usuarios: cuando un usuario introduce su nombre de usuario y contraseña, se almacena una galleta para que no tenga que estar introduciéndolas para cada página del servidor. Sin embargo, una galleta no identifica a una persona, sino a una combinación de computadora de la clase de computación-navegador-usuario.
- Conseguir información sobre los hábitos de navegación del usuario, e intentos de spyware (programas espía), por parte de agencias de publicidad y otros. Esto puede causar problemas de privacidad y es una de las razones por la que las cookies tienen sus detractores.

2.2.3 DATA STREAMS

Secuencia de códigos digitales de señales o paquetes de datos usados para transmitir y recibir información o que están en proceso de ser transmitidas (Ha, 2019).

2.2.4 HIPERVÍNCULOS

Un hipervínculo (también llamado enlace, vínculo, o hiperenlace) es un elemento de un documento electrónico que hace referencia a otro recurso, como por ejemplo otro documento o un punto específico del mismo o de otro documento. Combinado con una red de datos y un protocolo de acceso, un hipervínculo permite acceder al recurso referenciado en diferentes formas, como visitarlo con un agente de navegación, mostrarlo como parte del documento referenciador o guardarlo localmente (Ryte, 2018).

2.2.5 HTML

HTML, sigla en inglés de HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su

invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado (W3C, 2018).

2.2.6 LOG FILES

(W3C Working Draft WD-logfile-960323, 2018)En informática, se usa el término log, historial de log o registro a la grabación secuencial en un archivo o en una base de datos de todos los acontecimientos (eventos o acciones) que afectan a un proceso particular (aplicación, actividad de una red informática, etc.). De esta forma constituye una evidencia del comportamiento del sistema.

Por derivación, el proceso de generación del log se le suele llamar guardar, registrar o loguear (un neologismo del inglés logging) y al proceso o sistema que realiza la grabación en el log se le suele llamar logger o registrador.

Generalmente los acontecimientos vienen anotados con:

- El momento exacto o data (fecha, hora, minuto, segundo) en el que ocurrió lo que permite analizar paso a paso la actividad.
- Una o más categorizaciones del acontecimiento registrado. Es frecuente usar categorías distintas para distinguir la importancia del acontecimiento estableciendo distintos niveles de registro los cuales suelen ser: depuración, información, advertencia y error.

2.2.7 MODELOS DE DATOS

(Llanos Ferrais, 2007) Hacen referencia a la forma de los datos los cuales pueden ser estructurados, semi-estructurados o desestructurados.

Un modelo de datos es un lenguaje orientado a hablar de una Base de Datos. Típicamente un modelo de datos permite describir:

- Las estructuras de datos de la base: El tipo de los datos que hay en la base y la forma en que se relacionan.
- Las restricciones de integridad: Un conjunto de condiciones que deben cumplir los datos para reflejar la realidad deseada.
- Operaciones de manipulación de los datos: típicamente, operaciones de agregado, borrado, modificación y recuperación de los datos de la base.

Otro enfoque es pensar que un modelo de datos permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí.

2.2.8 PÁGINAS WEB

Una página web, o página electrónica, página digital, o ciberpágina es un documento o información electrónica capaz de contener texto, sonido, vídeo, programas, enlaces, imágenes, y muchas otras cosas, adaptada para la llamada World Wide Web (WWW) y que puede ser accedida mediante un navegador web. Esta información se encuentra generalmente en formato HTML o XHTML, y puede proporcionar acceso a otras páginas web mediante enlaces de hipertexto. Frecuentemente también incluyen otros recursos como pueden ser hojas de estilo en cascada, guiones (scripts), imágenes digitales, entre otros.

Las páginas web pueden estar almacenadas en un equipo local o en un servidor web remoto. El servidor web puede restringir el acceso únicamente a redes privadas, por ejemplo, en una intranet corporativa, o puede publicar las páginas en la World Wide Web. El acceso a las páginas web es realizado mediante una transferencia desde servidores, utilizando el protocolo de transferencia de hipertexto (W3C, 2018).

2.2.9 PÁGINAS WEB ESTÁTICAS

En el caso de las páginas estáticas, al acceder el usuario, el servidor descarga simplemente un simple fichero con un contenido codificado en HTML que se visualiza a continuación en su navegador. Un proceso muy similar a la descarga de cualquier fichero, por ejemplo, un documento PDF, se destaca el hecho de que no se almacena la información en una base de datos.

El principal problema de estas páginas es que no permiten la interacción con el usuario, equivalente a una colección de documentos invariables, como un libro, en la web, y que para realizar alguna modificación se tendría que realizar en el mismo código HTML.

2.2.10 PÁGINAS WEB DINÁMICAS

Las páginas dinámicas que se generan al momento de la visualización. No son un simple documento HTML, sino que se están creadas en algún lenguaje interpretado. El ejemplo más popular es PHP, el lenguaje en el que están programadas aplicaciones muy populares como WordPress o MediaWiki, el software en el que está implementado la propia Wikipedia.

Esto permite la creación de aplicaciones muy complejas. Un ejemplo típico serían las tiendas online como Amazon, Mercado Libre, Linio, etc.

Aquí la web interactúa con el usuario y es necesario que componga las páginas de manera dinámica. Por ejemplo: cuando un usuario busca determinados productos, la aplicación realiza una consulta a su base de datos, obtiene los resultados y compone con ellos "sobre la marcha" el HTML que corresponde a la lista de los productos. Una vez compuesto dinámicamente el HTML de la página entera, se devuelve al navegador exactamente igual que si hubiese sido una página HTML estática. S e destaca el uso de una base de datos donde se almacena dicha información consultada o requerida.

2.2.11 ROBOTS

(Robots.txt, 2018) Robots Web (también conocidos como Web Wanderers, Crawlers, o Spiders), son programas que atraviesan la Web automáticamente. Los motores de búsqueda como Google los usan para indexar el contenido web, los spammers los utilizan para buscar direcciones de correo electrónico y tienen muchos otros usos.

2.2.12 SERVIDORES WEB

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente. El código recibido por el cliente es renderizado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo.

Generalmente se usa el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador (EasyPHP, 2018). (Network Working Group, 2018)

2.2.13 URI

(Wayback Machine, 2018) Un identificador de recursos uniforme o URI —del inglés uniform resource identifier— es una cadena de caracteres que identifica los recursos de una red de forma unívoca. La diferencia respecto a un localizador de recursos uniforme (URL) es que estos últimos hacen referencia a recursos que, de forma general, pueden variar en el tiempo.

Normalmente estos recursos son accesibles en una red o sistema. Los URI pueden ser localizador de recursos uniforme (URL), uniform resource name (URN), o ambos.

Un URI consta de las siguientes partes:

- Esquema: nombre que se refiere a una especificación para asignar los identificadores, e.g. urn:, tag:, cid:. En algunos casos también identifica el protocolo de acceso al recurso, por ejemplo http:, mailto:, ftp:, etc.
- Autoridad: elemento jerárquico que identifica la autoridad de nombres (por ejemplo //www.example.com).
- Ruta: Información usualmente organizada en forma jerárquica, que identifica al recurso en el ámbito del esquema URI y la autoridad de nombres (e.g. /domains/example).
- Consulta: Información con estructura no jerárquica (usualmente pares "clave=valor") que identifica al recurso en el ámbito del esquema URI y la autoridad de nombres. El comienzo de este componente se indica mediante el carácter '?'.
- Fragmento: Permite identificar una parte del recurso principal, o vista de una representación del mismo. El comienzo de este componente se indica mediante el carácter '#'.

Aunque se acostumbra llamar URL a todas las direcciones web, URI es un identificador más completo y por eso es recomendado su uso en lugar de la expresión URL.

Un URI se diferencia de un URL en que permite incluir en la dirección una subdirección, determinada por el “fragmento”.

2.2.14 URL

(Network Working Group, 2018) Un Localizador Uniforme de Recursos (LUR, más conocido por la sigla URL, del inglés Uniform Resource Locator) es un identificador de recursos uniforme (Uniform Resource Identifier, URI) cuyos recursos referidos pueden cambiar, esto es, la dirección puede apuntar a recursos variables en el tiempo.1 Están formados por una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que designa recursos en una red, como Internet.

Los LUR fueron una innovación en la historia de Internet. Fueron usadas por primera vez por Tim Berners-Lee en 1991, para permitir a los autores de documentos establecer hiperenlaces en la World Wide Web (WWW). Desde 1994, en los estándares de Internet, el concepto de LRU ha sido incorporado dentro del más general de URI, pero el término URL todavía se utiliza ampliamente.

Aunque nunca fueron mencionadas como tal en ningún estándar, mucha gente cree que las iniciales LRU significan universal -en lugar de 'uniform'- resource locator (localizador universal de recursos). Esta se debe a que en 1990 era así, pero al unirse las normas "Functional Recommendations for Internet Resource Locators" (RFC 1736) y "Functional Requirements for Uniform Resource Names" (RFC 1737) pasó a denominarse "Identificador Uniforme de Recursos" (RFC 2396). Sin embargo, la letra "U" en URL siempre ha significado "uniforme".

El LRU es una cadena de caracteres con la cual se asigna una dirección única a cada uno de los recursos de información disponibles en Internet. Existe un URL único para cada página de cada uno de los documentos de la WWW, para todos los elementos de Gopher y todos los grupos de debate Usenet, y así sucesivamente.

El LRU de un recurso de información es su dirección en Internet, la cual permite que el navegador web la encuentre y la muestre de forma adecuada. Por ello, el URL combina el nombre de la computadora que proporciona la información, el directorio donde se encuentra, el nombre del archivo, y el protocolo a usar para recuperar los datos para que no se pierda alguna información sobre dicho, factor que se emplea para el trabajo.

Se puede entender que una URI = URL + URN.

Un HTTP URL combina en una dirección simple los cuatro elementos básicos de información necesarios para recuperar un recurso desde cualquier parte en Internet:

- El protocolo que se usa para comunicar, o enviar datos.
- El anfitrión (servidor o host) con el que se comunica.
- El puerto de red en el servidor para conectarse.
- La ruta al recurso en el servidor (por ejemplo, su nombre de archivo).

Un URL típico puede ser del tipo:

- http://es.wikipedia.org:80/wiki/Special:Search?search=tren&go=Go

Donde:

- http es el protocolo.
- es.wikipedia.org es el anfitrión.
- 80 es el número de puerto de red en el servidor (siendo 80 el valor por omisión para el protocolo HTTP, esta porción puede ser omitida por completo).
- /wiki/Special:Search es la ruta de recurso.
- ?search=tren&go=Go es la cadena de búsqueda (parte opcional).

2.2.15 WORD WIDE WEB

(W3C, 2018) En informática, la World Wide Web (WWW) o red informática mundial es un sistema de distribución de documentos de hipertexto o hipermedios interconectados y accesibles vía Internet. Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener textos, imágenes, vídeos u otros contenidos multimedia, y navega a través de esas páginas usando hiperenlaces.

2.3 ANTECEDENTES DE LA INVESTIGACIÓN

Cabe destacar que la presente investigación referida al tema nace de la tendencia que hay de estas tecnologías de resolver problemas, que los métodos tradicionales no pueden, aunque en el ámbito nacional ya hay experiencias sobre la aplicación de estos conceptos, es en el ámbito internacional donde más se ve su desarrollo.

2.3.1 DATA MINING AND ANALYSIS IN DEPTH CASE STUDY OF QAFQAZ UNIVERSITY HTTP SERVER LOG ANALYSIS

Paper de (Adamov, 2014) presentado en la **"Application of Information and Communication Technologies (AICT), 2014 IEEE 8th International Conference on",** donde concluye manifestando lo siguiente.

Que es importante tomar en cuenta el rápido crecimiento y la gran competencia que hay en el mercado del comercio electrónico y servicios en línea, la eficiencia y la conveniencia son los factores claves para el éxito.

Comentario. - lo que resalto de este trabajo es el hecho de que da a entender cuál es el formato de log comúnmente encontrado en la configuración de servidores web Apache, la razón por la cual elige R (información y librerías abundantes), para poder realizar el pre procesamiento de logs, y para realizar el filtrado de la información utilizó expresiones regulares, pues en los archivos logs hay muchos registros innecesarios, que necesitan ser filtrados.

Además, está el hecho de que, manifiesta que trabajo con 42 millones de log, que pesan aproximadamente 4.2 GB, pero no específica si trataron con un solo archivo o fueron varios, tampoco cuáles fueron las características del hardware para poder realizar todo el proceso, también el hecho de que, como una alternativa a R, está el comando egrep de Linux, el cual también puede trabajar con expresiones regulares.

2.3.2 A REVIEW STUDY OF SERVER LOG FORMATS FOR EFFICIENT WEB MINING

Paper de (Sharma & Yadav, A review study of server log formats for efficient web mining, 2015) presentado en la **"Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on"**, que en las conclusiones indica lo siguiente:

Esta publicación, explica que es la minería web actualmente, cuales son las categorías y en cuál de estas puede estar definido, indica sobre los tipos de datos. Describe que es un log de servidor y los diferentes formatos disponibles. Principalmente provee la comparativa de estudio entre varios tipos de formatos de logs, para tomar en cuenta para una eficiente minería web.

Comentario. - resalto de este trabajo, el hecho de que nos da una visión más amplia de los tipos y formatos de archivos log, hace énfasis en el contenido de estos, describiendo los tipos de objetos web que podríamos encontrar al interior de estos archivos. Además de indicar la categoría de los log, muestra la estructura interna de los log inclusive por tipo de servidor web, como son Apache (Common Log Format) o IIS (Internet Information Server).

2.3.3 A STUDY ON WEB USAGE MINING: THEORY AND APPLICATIONS

Paper de (Malviya & Agrawal, 2015) publicado en la **"Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference on"**, que en las conclusiones indica lo siguiente:

El modelo de minería de uso web, es un tipo de minería de log de servidores web, la minería de uso web actúa de forma importante en la realización de la usabilidad del diseño de un sitio web, el incremento de la relación con los clientes.

También presenta un soporte para proveer una personalización de servicios, diseño de un sitio web y la toma de decisiones en los negocios, etc. Concluimos también con que se puede resolver problemas como encontrar información deseada, relacionada, aprendizaje, conocimiento, recomendaciones o personalización de datos, y recientes investigaciones de la minería de uso web están en este campo. Se compara la minería de uso web con la minería de datos y los almacenes de datos en algunos factores, además observamos que la minería de uso web es muy esencial en la actualidad para el mundo web y muchos usos en el futuro en cuanto a la privacidad del usuario.

Comentario. - lo que resalto de este trabajo, es el hecho de que, en parte señala las dificultades que hay en el descubrimiento de información, en grandes cantidades de datos, realiza también una comparación de varios autores y su contribución en el campo de la minería de uso web y muestra las técnicas que usaron estos, así como el algoritmo que aplicaron. Dentro de las aplicaciones de la minería de uso web, resaltan: el descubrimiento del tráfico web, para generar nuevas políticas concernientes al servidor web, que coadyuvaran a la satisfacción del usuario, otra aplicación es la restructuración de los sitios web, en cuanto a composición de contenidos y la distribución de los enlaces en la página, además también menciona dos aplicaciones más como son la personalización de sitios web y el soporte al diseño de los sitios web.

2.3.4 REVIEW ON MODERN DATA PREPROCESSING TECHNIQUES IN WEB USAGE MINING (WUM)

Paper de (Sukumar, Robert, & Yuvaraj, 2016) publicado en la **"Computation System and Information Technology for Sustainable Solutions (CSITSS), International Conference on"**, que en la conclusión indica lo siguiente:

Esta publicación ha presentado muchos puntos de tareas de pre procesamiento, que son necesarios para mejorar la minería de uso web. Este trabajo también presento un trabajo experimental con resultados de los log de servidores web del Colegio de Artes Gubernamentales de Coimbatore, realizando experimentos en procesamiento de datos, heurísticas y técnicas aplicadas a limpiar los log en crudo.

Luego de esto los datos limpios, son pasados a una fase denominada identificación de usuarios, en esta fase se utiliza un algoritmo que identifica distintos tipos de usuario que acceden al sitio web. Hallando también el número de sesiones de sesiones de usuario.

Finalmente, los resultados del pre procesamiento del archivo de las sesiones de usuario, son utilizados para identificar usuarios únicos, número total de accesos, visitas, promedio por día, accesos fallados y respuestas exitosas.

El pre procesamiento de datos ha sido usado para analizar los log de servidores web, ambos estudios teóricos y experimentales muestran la efectividad de aplicar heurísticas basadas en pre procesamiento.

Pero también se encontraron algunas dificultades en la colección de datos, las métricas para la identificación de usuarios, la identificación de sesiones y aplicar los resultados al descubrimiento de patrones. Son algunos de los trabajos enfocados a estas áreas, pero aún hay más que deben ser exploradas.

Comentario. - este trabajo hace énfasis en el pre procesamiento de datos, para la limpieza de los log, construye su propio algoritmo para limpieza de datos, identificación de usuarios y sesiones, además incluye por separado un algoritmo para limpiar registros dejados por los robots y arañas de google.

En los resultados indica que trabajo con 22613 logs, que corresponden aproximadamente a 15 días del funcionamiento del servidor y detalla lo ocurrido día por día del día 1 al 12, mediante tablas indicando lo encontrado.

2.3.5 COMPARATIVE ANALYSIS OF WEB-MINING APPROACHES FOR EFFICIENT MINING OF SERVER LOG FORMATS

Paper de (Sharma, Bohra, & Yadav, 2016) publicado en la **"Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2016 5th International Conference on",** que en las conclusiones indica lo siguiente:

La minería web, utiliza la minería en los sistemas de información para mostrar reportes en la World Wide Web. El procedimiento general de la minería web, incorpora extracción de datos de la World Wide Web, de la información minada o que está en el mismo sitio web, en esta investigación fueron usadas dos técnicas, el Apriori y FP Growth basados en el tiempo de ejecución y la generación de patrones.

Comentario. - lo que resalto de esta publicación es, la construcción de un programa en Java, donde propone una arquitectura, en la cual implementa dos algoritmos como son el Apriori y FP Growth, dicha herramienta realiza, la limpieza de datos, clustering y filtrado de datos. Aunque no indica con que tamaño de logs ha utilizado, ni la cantidad procesada, muestra un cuadro comparativo donde muestra que el algoritmo Apriori es mucho mejor que FP Growth en tiempo de ejecución y que ocurre lo contrario cuando se quiere generar mejores patrones, en este caso FP Growth es mejor.

2.3.6 PREDICTING USER BEHAVIOR THROUGH SESSIONS USING THE WEB LOG MINING

Paper de (Neelima & Rodda, 2016) publicado en la **"Advances in Human Machine Interaction (HMI), 2016 International Conference on"**, que en el abstract indica lo siguiente:

La minería de uso web es una de las áreas emergentes de la investigación y un sub dominio importante de la minería de datos, debemos tomar en cuenta que todas sus técnicas y son importantes a tomar en cuenta, al momento de realizar el pre procesamiento en cuanto a eficiencia y efectividad.

Esta publicación trata de centrarse en las áreas del pre procesamiento, limpieza de datos, identificación de usuarios e identificación de sesiones. Una vez que los datos estuvieron pre procesados se pudo aplicar técnicas de minería de datos, como clustering, asociación y clasificación, para aplicaciones de la minería de uso web como son el comercio electrónico, aprendizaje electrónico, personalización, etc.

La minería de log web, es una de las recientes áreas de investigación en la minería de datos. También es un aspecto importante hoy en día, pues la cantidad de datos se va incrementando continuamente.

Los archivos log, se analizaron para poder facilitarlos al administrador del sitio web, para que pueda tomar en consideración, para la configuración del ancho de banda y la capacidad del servidor da la organización. Para analizar estos logs, es posible descubrir varias clases de conocimiento, el cual puede aplicar para analizar el comportamiento de los usuarios.

La propuesta de este sistema es usada para analizar, las sesiones de usuario, obtenidas de las sesiones de los usuarios en diferentes intervalos de tiempo. Y esto es utilizado para configurar el servidor y ajustar la configuración del sitio web, y es de mucha utilidad, para el administrador.

Comentario.- en este trabajo se hace énfasis en la descripción, formatos y tipos de archivos logs, además describe una metodología que consiste en la limpieza identificación de usuarios e identificación de sesiones, realiza la implementación de sus propios algoritmos para realizar la metodología que propone y en sus resultados experimentales, la limpieza y estructuración son almacenadas en una bases de datos MySql Server, donde muestra el resultado obtenido en graficas tipo circular, apreciándose únicamente la identificación de usuarios y sesiones, aparentemente generadas en Excel , la cantidad de los logs tratados es de 1546 y no indica a que entidad le pertenece.

2.3.7 EXTRACCIÓN DE PATRONES SEMÁNTICAMENTE DISTINTOS A PARTIR DE LOS DATOS ALMACENADOS EN LA PLATAFORMA PAIDEIDA.

Tesis de maestría de (FLORES LAFOSSE, 2016) donde, Paideia es una plataforma educativa basada en Moodle. Actualmente se utiliza en cursos de pregrado, cursos virtuales y programas/diplomas especiales.

Cuyo objetivo principal fue: Explotar los datos de un entorno virtual de aprendizaje a fin de caracterizar el comportamiento de los individuos a partir de las acciones realizadas por los usuarios en la plataforma Paideia, que tuvo como hipótesis: Se puede extraer tres tipos de patrones semánticamente diferentes a partir de los datos asociados a los logs de Paideia y que en sus conclusiones mencionan que: podemos afirmar que es posible extraer patrones semánticamente distintos de los datos de los logs de una plataforma educativa.

Comentario.- se resalta que, no hay pre procesamiento de datos, ni un procedimiento de limpieza, pues la plataforma moodle tiene una tabla donde almacena en una base de datos todos los logs, y como un usuario tiene que loguearse, por ende la identificación de estos se almacenan allí, utilizo también Weka que es un programa hecho en java para realizar minería de datos, y generar reglas de asociación, y otro software denominado SPMF, que es una implementación realizada en Java el cual utiliza para mostrar el conjunto de ítems.

CAPITULO III

METODOLOGÍA

3.1 TIPO Y DISEÑO DE INVESTIGACIÓN

Según (Hernandez Sampieri, 2014), el tipo y diseño de investigación pertenecen a las definiciones que se tienen a continuación.

Tipo: Investigación descriptiva, porque se pretende especificar las características de los log y sus componentes, para lo cual se recogió información al respecto de las acciones que se dan en torno a la interacción con una página web, en este caso, de la UNSAAC.

Diseño: corresponde a estudios de caso (casos de estudio), porque se analiza las acciones de los usuarios en las páginas web de la UNSAAC, como una unidad de análisis, donde identificamos sus límites y características en su contexto. Cumple además con la mayor cantidad de ítems de la secuencia general del estudio de caso, aplicado a ciencias e ingeniería, el cual se compone de: planteamiento del problema, definición del caso, seleccionar el caso y el sitio en contexto, construir el marco teórico base del estudio, recolectar datos o evidencia necesaria, construir una base de datos para poder adicionar, cruzar y comparar información proveniente de las distintas fuentes, procesar y analizar datos de la base de datos y reportar resultados.

Por su finalidad el diseño comprendería a un estudio de caso de tipo intrínseco, pues se pretende conocer más sobre la singularidad de este caso en particular.

3.2 UNIDAD DE ANALISIS.

Lo que se observó, fueron los log alojados en los servidores de la Red de Comunicaciones UNSAAC (RCU), correspondiente al año 2017 y realizar el proceso de análisis, que comprenden a los recursos (páginas de error, archivos con extensión .css, archivos con extensión .js, imágenes, bots de google, archivos con extensión .pdf) regularmente accedidos por los usuarios que interactúan con la página web de la UNSAAC.

3.3 POBLACIÓN DE ESTUDIO

Para realizar el análisis, se efectuará con los usuarios de las páginas web de la Universidad Nacional de San Antonio Abab del Cusco. Específicamente los que comprenden al dominio www.unsaac.edu.pe, en cifras aproximadamente a, ***94,949,716*** de log correspondiente únicamente al año 2017, los cuales fueron facilitados por la RCU.

3.4 SELECCIÓN DE LA MUESTRA.

Se ha tomado como caso de estudio, logs de servidores web de la Universidad Nacional de San Antonio Abab del Cusco, de los cuales se tiene la información del año 2017.

Nro. archivos logs	*41*
Peso	**24.24 GB**
Numero de logs	**94,949,716.00**

Tabla 1.- Datos en crudo

Algo muy importante que se observó antes de realizar dicho análisis fue el hecho que, si bien los servidores web tienen la capacidad de registrar la dirección ip y el usuario que accede, para este caso un usuario no necesita loguearse para ver el contenido de las páginas web de la UNSAAC, así que solo se tiene la dirección ip, como identificador único de cada transacción.

Entonces se debe informar también que esto genera un problema más de exactitud, pues al utilizar un proveedor de servicios de internet como claro, movistar o similares, hay una posibilidad de que dichas empresas limiten el acceso a internet utilizando tecnología NAT 3 (Egevang K., 2018), esto, significa que muchos usuarios comparten la misma dirección ip, con lo que el análisis que se realiza estaría condicionado a esta situación descrita; se realizó una prueba simple, que corrobora lo afirmado.

3.5 TAMAÑO DE MUESTRA

A continuación, se muestra los resultados del procesamiento de dichos datos, para lo cual se creó un programa, el cual se construyó exclusivamente para realizar las tareas antes mencionadas como se muestra en el Anexo 1 y el 3.33% del total de la población, que fueron elegidos en forma aleatoria, de cada uno de los archivos, el cual se muestra a continuación.

Nro. archivos logs	*41*
Peso	**1.21 GB**
Nro de logs	**3,797,989.00**

Tabla 2.- Logs estructurados y procesados

Para determinar el tamaño de la muestra se recurrió a la fórmula propuesta por Murray y Larry (2005), donde para la población de **94,949,716.00** con valores estándares para trabajos de investigación como desviación estándar de 0.5, confianza del 95% y limite aceptable de error del 5%, dio un tamaño de muestra de **385**, por lo que se descartó dicho valor al ser muy pequeño y no representar de manera significativa a la población, además considerando de que se trata de logs sin pre procesamiento ni limpieza por lo que esta cantidad se reduciría aún más.

Entonces se optó por el muestreo aleatorio simple, es decir se dividieron los archivos tomando en consideración el 3.33% de cada log, la razón por la cual no es un porcentaje exacto es porque los archivos log tienen una cantidad de registros log y al realizar la división no necesariamente coincidirá con la última acción de una ip determinada, haciendo esto último que no sea exacto.

3.6 TECNICAS DE RECOLECCION DE DATOS E INFORMACION

Para el presente proyecto se optó por el muestreo aleatorio simple, los cuales representan el 3.33% del total debido a que la cantidad de datos para ser tratados u observados, eran cercanos al big data y como antes mencionamos, requerían de un costo computacional muy alto, la forma como se seleccionaron los datos fue tomando simplemente el criterio de seleccionar el 3.33% del total de cada archivo log, dicha selección no afecta en absoluto el resultado final, pues se comprobaron con tres archivos al azar y el resultado fue similar, y que en algunos casos correspondía a 200,000 o 70,000 según la cantidad de logs, y se seleccionaron los n primeros en vista que los logs están ordenados cronológicamente y viendo que no se pierdan la información del ultimo usuario de la muestra para no alterar el análisis. La cantidad de datos recolectados que asciende a **3,797,989** logs (en crudo) y que posteriormente pasaron a ser **32,994** logs (listos para el análisis), es considerable pues otros análisis se realizaron de entre 500 a 4000 logs (en crudo) aproximadamente.

3.7 ANALISIS E INTERPRETACION DE LA INFORMACION

Se detallará la forma como se utilizó el proceso de minería de uso web. El cual se representará en la siguiente gráfica de la figura 11.

3.7.1 ALCANCE DEL PROCESO DE LA MINERIA DE USO WEB

El proceso de minería de uso web, comprende dos fases, según (Liu, 2012, pág. 528), la fase de preparación de datos y la de descubrimiento de patrones (preferencias), el cumplimiento en su totalidad de cada una de estas fases depende en gran medida del formato en el que se registran los logs y del contenido de estos, por lo que se da el caso, que algunas etapas no estén presentes durante el proceso, debido a que todo depende de la heurística que se aplique al contexto dado.

En el estudio de caso, de los log de la UNSAAC, en cuanto al formato y contenido, en la primera fase se centrara en las etapas de, limpieza de datos, identificación de páginas vistas, integración y transformación de datos, no se verá la etapa de (sessionization) que es la etapa de segmentación del registro de actividades de cada usuario, puesto que no se cuenta con cuentas de usuario a nivel de la página y con ello caracterizar tiempos de permanencia en cada una de las páginas en función a la fecha y horas de acceso de cada página. En cuanto a la segunda fase se centrará en las etapas de minería de reglas de asociación, análisis correlacional, minería de patrones secuenciales, no se verá el clustering de transacciones, ni el clustering de páginas vistas, pues dependen de la etapa de segmentacion. También se verá el análisis y filtrado de los patrones de navegación.

Para la segunda fase, se utiliza el algoritmo Apriori, aunque como menciona (Liu, 2012, pág. 23) existen muchos otros algoritmos implementados basados en Apriori como FP-growth, MS-Apriori, CAR-Apriori, GSP, MS-GSP, etc. que se diferencian porque son más eficientes, utilizan estructuras de datos y técnicas muchos más sofisticadas y rápidas, pero que nos darán los mismos resultados a nivel de conjunto de ítems frecuentes y por ende las mismas reglas de asociación, esta fue básicamente la razón por la que se determinó el uso del algoritmo Apriori, ya que a nivel de resultados, no habrá ninguna diferencia.

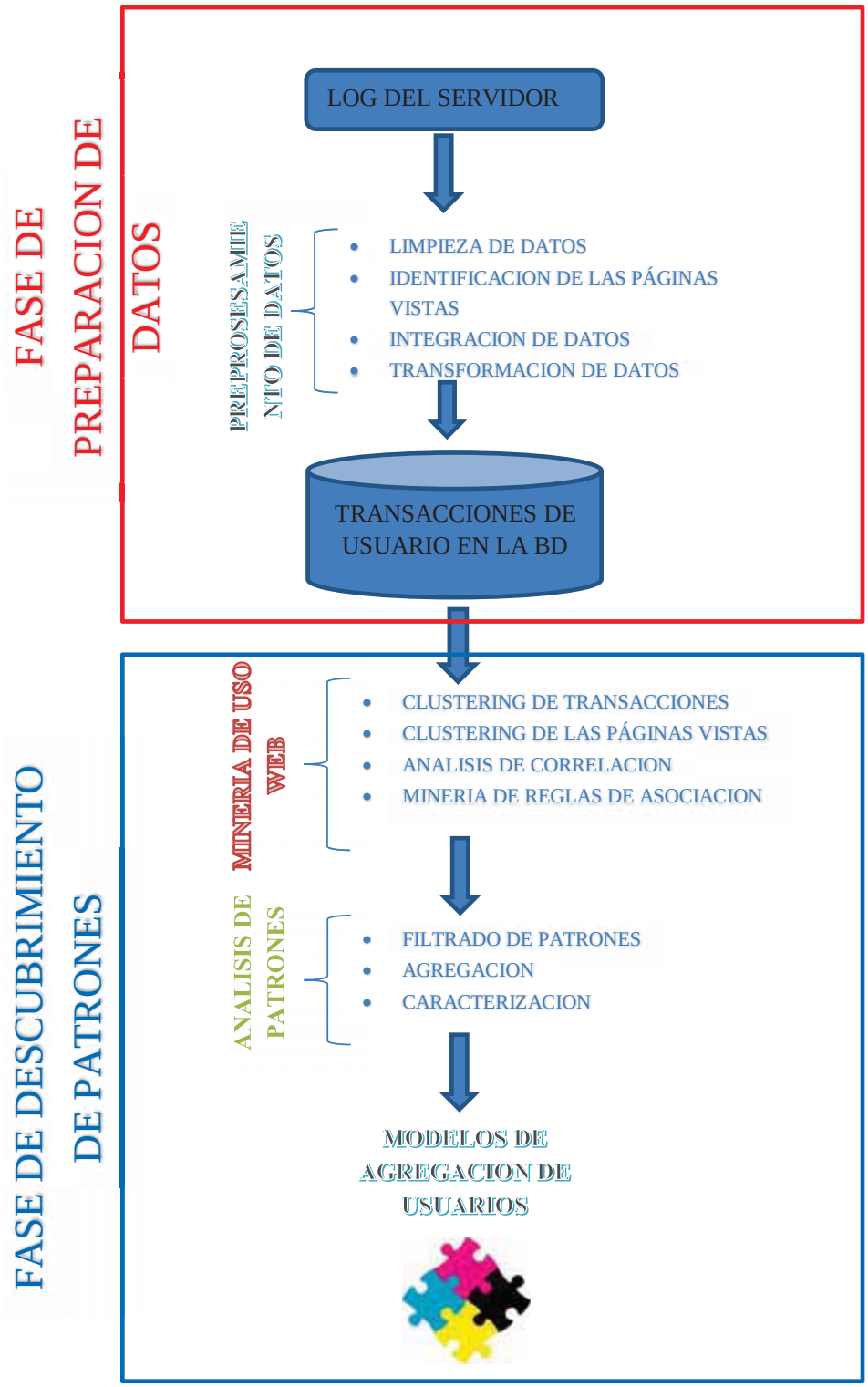

Figura 11.- proceso de limpieza y estructuración de los log del servidor web de la UNSAAC.

3.7.2 DELIMITACION

El presente proyecto se delimita de la siguiente manera, para la realización de la minería de uso web se tomará en cuenta únicamente los logs del servidor correspondientes al dominio www.unsaac.edu.pe de los cuales solo se tratarán **3,797,989** de los ***94,949,716*** de log debido a que el tiempo computacional invertido para realizar la limpieza y estructuración es muy alto, además se desconoce qué porcentaje del total será útil realmente al análisis, el algoritmo que utilizaremos es Apriori aunque hay algoritmos basados en este que son mucho más rápidos según detallan comparaciones en los antecedentes, pero que arrojan los mismos resultados en cuanto a reglas de asociación se determinó que no sería necesario usar uno diferente al propuesto en vista que lo que nos interesa realmente es el resultado y no la rapidez con la que se obtiene, también dentro de la muestra seleccionada que corresponde al 2017 al momento de realizar una inspección preliminar se observó que no existen logs correspondientes a los meses de marzo, abril y mayo, se desconoce la causa de dicha ausencia pues al consultar con los encargados de la RCU nadie dio razón del motivo mostrando extrañeza por dicha falta, su supuso que la falta de logs probablemente se deba a un ataque de hacking.

3.7.3 FASE DE PREPARACION DE DATOS

3.7.3.1 LOG DEL SERVIDOR WEB DE LA UNSAAC

A continuación, se detalla los log facilitados por la Red de Comunicaciones UNSAAC.

LOGS UNSAAC 2017

Nombre del archivo	***Peso en KB***	***Peso en MB***	***Nro de logs***	***Peso en GB***
access.log.44	825,034	805.70	2,606,200	0.79
access.log.43	1,023,421	999.43	3,459,717	0.98
access.log.42	978,885	955.94	3,015,802	0.93
access.log.41	518,300	506.15	1,820,931	0.49
access.log.40	386,613	377.55	1,621,281	0.37
access.log.39	452,137	441.54	1,898,738	0.43
access.log.38	270,202	263.87	1,144,015	0.26
access.log.37	6	0.01	26	0.00
access.log.36	94	0.09	515	0.00
access.log.35	55	0.05	385	0.00
access.log.34	130	0.13	813	0.00
access.log.33	690,526	674.34	2,529,437	0.66
access.log.32	663,797	648.24	2,550,405	0.63
access.log.31	594,460	580.53	2,886,880	0.57
access.log.30	554,435	541.44	2,114,428	0.53
access.log.29	815,175	796.07	3,035,521	0.78
access.log.28	622,703	608.11	2,383,560	0.59

access.log.27	501,501	489.75	1,928,210	0.48
access.log.26	484,492	473.14	1,853,272	0.46
access.log.25	603,037	588.90	2,294,889	0.58
access.log.24	1,407,474	1374.49	5,315,580	1.34
access.log.23	771,368	753.29	2,890,068	0.74
access.log.22	738,915	721.60	2,804,552	0.70
access.log.21	620,450	605.91	2,361,210	0.59
access.log.20	585,963	572.23	2,212,084	0.56
access.log.19	586,511	572.76	2,020,200	0.56
access.log.18	709,223	692.60	2,636,506	0.68
access.log.17	743,272	725.85	2,813,200	0.71
access.log.16	455,739	445.06	1,728,272	0.43
access.log.15	610,127	595.83	2,350,229	0.58
access.log.14	470,271	459.25	1,794,567	0.45
access.log.13	529,811	517.39	2,012,152	0.51
access.log.12	524,886	512.58	2,010,311	0.50
access.log.11	780,761	762.46	2,949,795	0.74
access.log.10	858,164	838.05	3,206,010	0.82
access.log.9	780,221	761.93	2,937,957	0.74
access.log.8	1,627,621	1589.47	5,847,330	1.55
access.log.7	751,795	734.17	2,765,619	0.72
access.log.6	617,588	603.11	2,331,062	0.59
access.log.5	787,347	768.89	3,015,178	0.75
access.log.4	471,472	460.42	1,802,809	0.45
41	***25,413,982***	***24818.34***	***94,949,716***	***24.24***
Total archivos	***Total en Kb***	***Total en Mb***	***Total Logs***	***Total en GB***

Tabla 3.-Log del 2017 del dominio www.unsaac.edu.pe

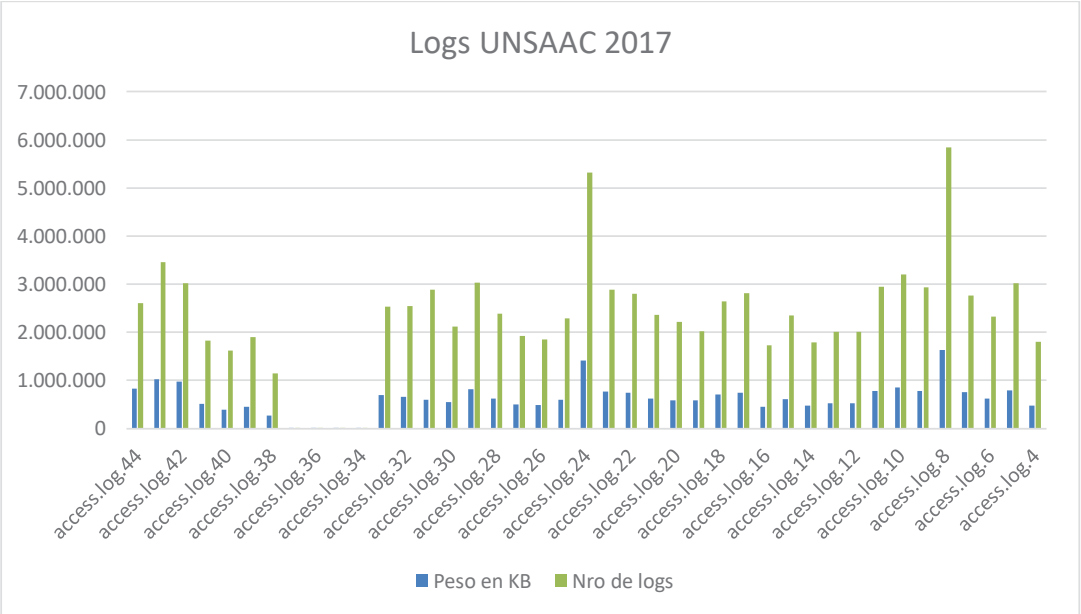

Figura 12.- logs en crudo.

Como se puede apreciar en la tabla 2, se tiene 24.24 GB distribuidos en 41 archivos, que en su totalidad contienen a 94 949 716 logs, de estos archivos se pudo apreciar que, de los más representativos podríamos mencionar a los más significativos, que el archivo más pesado es de 1.34 GB con 5,847,330 logs y el más liviano es 0.37 GB con 1,621,281 de logs; los archivos con pesos por encima de los 600 MB (prácticamente todos). No pudieron ser abiertos con ningún procesador de texto tradicional, justamente por contener información cercana al big data (denominación que le dan los programas, a archivos que no pueden ser abiertos o visualizados con procesadores de texto tradicional).

A demás, una vez que se pudo abrir los archivos se procedió a realizar la extracción del 3.33% del total de logs de cada archivo, y se procedió a renombrarlos, como en el caso del archivo con nombre Access.log.04 a Access.log.04.5, y de esta manera con todos los archivos, como se puede apreciar uno de los log en crudo en la figura 13, antes de realizar el análisis.

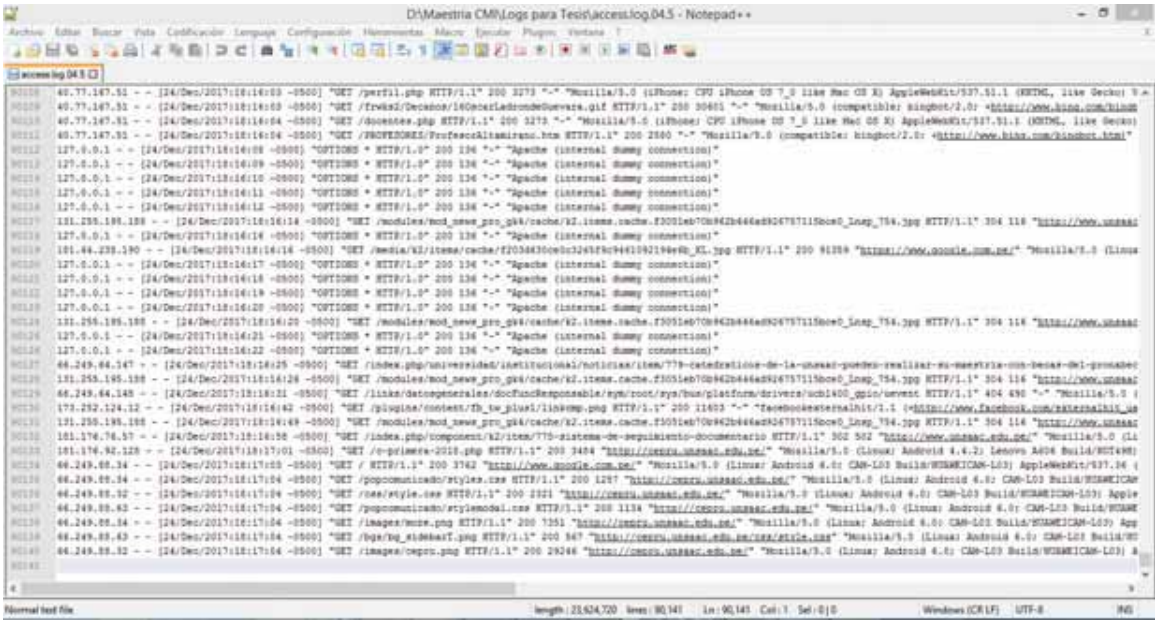

Figura 13.- parte del archivo access.log.04.5 en crudo, con mas de 90,000 lineas

3.7.3.2 FASE DE PREPROCESAMIENTO DE DATOS

3.7.3.2.1 LIMPIEZA DE DATOS

Como se puede apreciar en los datos en crudo, cada línea contiene: la ip del usuario, fecha y hora de la consulta, método POST o GET, la URL del recurso consultado, versión del protocolo de transferencia HTTP, código del estado de respuesta y numero de bytes de la respuesta HTTP.

Debido a la enorme cantidad de datos y realizando pruebas con archivos log representativos, de más o menos de entre 30 MB y 40 MB, en una computadora con procesador Intel Core i5-5200 CPU 2.20 GHz, con 4 GB de memoria RAM en S.O. Windows 8.1 de 64bits, también que al

limpiar y estructurar 10 000 log se demora un promedio de 10 minutos aproximadamente y 160 000 log los realiza en 2 horas y 40 minutos aproximadamente, esto llevaría a la enorme tarea de procesar los más de 94 millones de logs, lo que demandaría un tiempo de 1 582.50 horas aproximadamente, que en días seria aproximadamente 66 días de 24 horas, lo cual demandaría un costo computacional altísimo, por lo que se realizara el análisis tomando en cuenta muestras de cada uno de estos log en un aproximado de 3.33% de cada uno de los archivos que contienen los log, señalando que en estudios parecidos se tomaron un poco más de 500 log, por lo que el análisis de los log de este estudio será mucho más preciso.

Para que se realice dicha tarea se utilizó Emeditor (Emurasoft Inc., 2018) el que facilito bastante la división de los archivos a unos más manipulables, además se descartaron los archivos log, access.log.34, access.log.35, access.log.36, access.log.37, pues prácticamente, no se encontró información alguna en ellos, se desconoce la razón por la cual no hay registrado log alguno en estos archivos.

LOGS UNSAAC 2017 EN CRUDO				*3.33% DE CADA LOG EN CRUDO*		
Nombre del archivo	***Peso en KB***	***Peso en MB***	***Nro de logs***	***Nombre del archivo***	***Peso en MB***	***3.33% de nro.de logs***
access.log.44	825,034	805.70	2,606,200	access.log.44.5	26.83	86786
access.log.43	1,023,421	999.43	3,459,717	access.log.43.5	33.28	115209
access.log.42	978,885	955.94	3,015,802	access.log.42.5	31.83	100426
access.log.41	518,300	506.15	1,820,931	access.log.41.5	16.85	60637
access.log.40	386,613	377.55	1,621,281	access.log.40.5	12.57	53989
access.log.39	452,137	441.54	1,898,738	access.log.39.5	14.70	63228
access.log.38	270,202	263.87	1,144,015	access.log.38.5	8.79	38096
access.log.37	6	0.01	26	access.log.37.5	0.00	1
access.log.36	94	0.09	515	access.log.36.5	0.00	17
access.log.35	55	0.05	385	access.log.35.5	0.00	13
access.log.34	130	0.13	813	access.log.34.5	0.00	27
access.log.33	690,526	674.34	2,529,437	access.log.33.5	22.46	84230
access.log.32	663,797	648.24	2,550,405	access.log.32.5	21.59	84928
access.log.31	594,460	580.53	2,886,880	access.log.31.5	19.33	96133
access.log.30	554,435	541.44	2,114,428	access.log.30.5	18.03	70410
access.log.29	815,175	796.07	3,035,521	access.log.29.5	26.51	101083
access.log.28	622,703	608.11	2,383,560	access.log.28.5	20.25	79373
access.log.27	501,501	489.75	1,928,210	access.log.27.5	16.31	64209
access.log.26	484,492	473.14	1,853,272	access.log.26.5	15.76	61714
access.log.25	603,037	588.90	2,294,889	access.log.25.5	19.61	76420
access.log.24	1,407,474	1374.49	5,315,580	access.log.24.5	45.77	177009
access.log.23	771,368	753.29	2,890,068	access.log.23.5	25.08	96239
access.log.22	738,915	721.60	2,804,552	access.log.22.5	24.03	93392
access.log.21	620,450	605.91	2,361,210	access.log.21.5	20.18	78628
access.log.20	585,963	572.23	2,212,084	access.log.20.5	19.06	73662
access.log.19	586,511	572.76	2,020,200	access.log.19.5	19.07	67273
access.log.18	709,223	692.60	2,636,506	access.log.18.5	23.06	87796
access.log.17	743,272	725.85	2,813,200	access.log.17.5	24.17	93680

access.log.16	455,739	445.06	1,728,272	access.log.16.5	14.82	57551
access.log.15	610,127	595.83	2,350,229	access.log.15.5	19.84	78263
access.log.14	470,271	459.25	1,794,567	access.log.14.5	15.29	59759
access.log.13	529,811	517.39	2,012,152	access.log.13.5	17.23	67005
access.log.12	524,886	512.58	2,010,311	access.log.12.5	17.07	66943
access.log.11	780,761	762.46	2,949,795	access.log.11.5	25.39	98228
access.log.10	858,164	838.05	3,206,010	access.log.10.5	27.91	106760
access.log.9	780,221	761.93	2,937,957	access.log.9.5	25.37	97834
access.log.8	1,627,621	1589.47	5,847,330	access.log.8.5	52.93	194716
access.log.7	751,795	734.17	2,765,619	access.log.7.5	24.45	92095
access.log.6	617,588	603.11	2,331,062	access.log.6.5	20.08	77624
access.log.5	787,347	768.89	3,015,178	access.log.5.5	25.60	100405
access.log.4	471,472	460.42	1,802,809	access.log.4.5	15.33	60034
41	***25,413,982***	***24818.34***	***94,949,716***	41	***826.45***	***3161826***
Total archivos	***Total en Kb***	***Total en Mb***	***Total Logs***	***Total archivos***	***Total en Mb***	***Total Logs (3.33%)***

Figura 14.- 3.33% del total de logs.

Se desarrolló un programa en PHP (se puede ver los anexos) y procedimientos almacenados en MySQL, cuya finalidad fue realizar el proceso de carga de archivos, lectura, selección y llamado de los procedimientos almacenados para realizar la estructuración y la integración de los log, en una base de datos, para posteriormente puedan ser exportados para será analizados con R.

Se optó por utilizar un programa propio, puesto que el uso de las expresiones regulares que se detallaron en los antecedentes, al momento de probarlas con los archivos log, no dieron buenos resultados, esto debido a la configuración propia de cada formato de log, además se tuvo inconvenientes con la carga de los archivos que al ser pesados no podían ser cargados en el programa, por lo que se tuvo que elegir PHP y además configurar un servidor web Apache, para que permitiera cargar archivos de 500 MB por ejemplo y de esta forma realizar el proceso mencionado.

Luego de realizar el proceso de filtrado, limpieza y estructurado, el cual consistió en eliminar archivos con extensiones, .css, .js, .jpg, .rar, direcciones de protocolos de internet local como el 127.0.0.1, bots de google, páginas con código de error 404 etc. solo por mencionar algunos, pues no aportan nada en absoluto al análisis, aproximadamente demoro 39 horas distribuidas en 7 días, pues la computadora que se usó, no es de uso exclusivo para el procesamiento de datos del presente proyecto. En dicho proceso, se construyó una aplicación web en PHP, la cual realizo la lectura de archivos y pre procesamiento, que luego se insertó en una base de datos en MYSQL, donde se terminó de realizar el procesamiento y la limpieza, utilizando para ello procedimientos almacenados y funciones, que terminaron en la estructuración, finalmente se convirtió la base de datos al formato .CSV, para poderlo analizar con R y RStudio.

A continuación, se presenta la información obtenida y que luego fue tratada.

Grupos de recursos más accedidos 2017	
Recurso	***Cantidad***

/modules/	409272
/templates/	317745
/media/	200110
/bgs/	134664
/investigacion/	20875
/convocatorias/	17639
/popcomunicado/	14056
/index.php/	9197
/lib/	8260
/storage/	5822
/Imagenes/	5364
/pestania/	4261
/popresolucion/	2896
/pics/	2335
/académico	1866
/wp-content/	1844
/links/	1532
/laescuela/	1462
/banner/	1238
/psocial/	1210
/Scripts/	1203
/tesispostgrado/	1191
/maestrias/	1121
/javascripts/	618
/slider/	610
/wp-admin/	596
/slidenob/	455
/../images/	417
/js/	368
/Index_files/	347
/c-primera-2018.php	344
/administrator/	340
/c-ordinario-2016-II.php	333
/servicios/	293
/c-ordinario-2017-II.php	285
/inversion.php	260
/investiga/	251
/universidad/	242
/oficinas/	225
/stylesheets/	224
/menu/	206
/wp-includes/	197
/c-intensivo-2017.php	159
/c-ordinario-2017-I.php	143
/res/	122
/c-ordinario-2015-II.php	110

/PROFESORES/	102
/style/	102
/admision.php	74
/aestatutaria/	67
/otros/2doconcursoAscensoPAdm2016.php	66
/informacion.php	52
/JovenesObra/	43
/c-intensivo-2018.php	33
/Classic-1/	28
/vrin/	28
/temporales/	24
/lightbox/	16
/AccesosGenerales/	15
/academico.php	11
/cirugiabucal.html	6
/institucional/	6
/autoridades/	5
/c-ordinario-2012-I.php	5
/administrativo.php	3
/admision.htm	1
/c-ordinario-2016-I.php	1

Tabla 4.- Recursos accedidos durante el 2017

Como se puede apreciar en la tabla 5, muchos de los recursos no cuentan con nombres adecuados.

RECURSO	*HOST*	*CANTIDAD*
/templates/gk_musicstatehttps://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js	http://www.UNSAAC.edu.pe/	22245
/favicon.ico	http://cepru.UNSAAC.edu.pe/	6985
/lib/_class.noobSlide.packed.js	http://www.UNSAAC.edu.pe/	2674
/style.css	http://www.UNSAAC.edu.pe/	2388
/lib/mootools-1.2-core.js	http://www.UNSAAC.edu.pe/	2377
/bgs/down.png	http://postgrado.UNSAAC.edu.pe/	1066
/bgs/left.png	http://postgrado.UNSAAC.edu.pe/	1052
/templates/gk_musicstatehttps://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js	http://www.UNSAAC.edu.pe/index.php/academico/pre-grado/calendarios-academicos	953
/favicon.ico	http://postgrado.UNSAAC.edu.pe/	597
/templates/gk_simplicity/images/style3/typography/bullet-square2.png	http://www.UNSAAC.edu.pe/templates/gk_simplicity/css/style3.css	593
/styler.js	http://museoinka.UNSAAC.edu.pe/	490

/favicon.ico	http://cepru.UNSAAC.edu.pe/c-ordinario-2017-II.php	418
/rasberry.css	http://museoinka.UNSAAC.edu.pe/	407
/templates/gk_musicstatehttps://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js	http://www.UNSAAC.edu.pe/index.php	355
/favicon.ico	http://cepru.UNSAAC.edu.pe/c-ordinario-2017-I.php	323
/favicon.ico	http://cepru.UNSAAC.edu.pe/c-primera-2018.php	304
/templates/gk_musicstatehttps://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js	http://www.UNSAAC.edu.pe/index.php/servicios	265
/tesispostgrado/images/move-top.png	http://postgrado.UNSAAC.edu.pe/tesispostgrado/css/style.css	264
/templates/gk_musicstatehttps://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js	http://www.UNSAAC.edu.pe/index.php/universidad/institucional	234

Tabla 5.- Parte de la relacion de recursos con error 404

También se resalta este inconveniente, que hay muchos recursos, que actualmente ya no están en el servidor, pero los llamados a dichos recursos siguen persistiendo, hallamos en la muestra que se analizó, 2400 recursos únicos con error 404. Esta es una de las razones porque el análisis toma tiempo, pues se trata de información innecesaria.

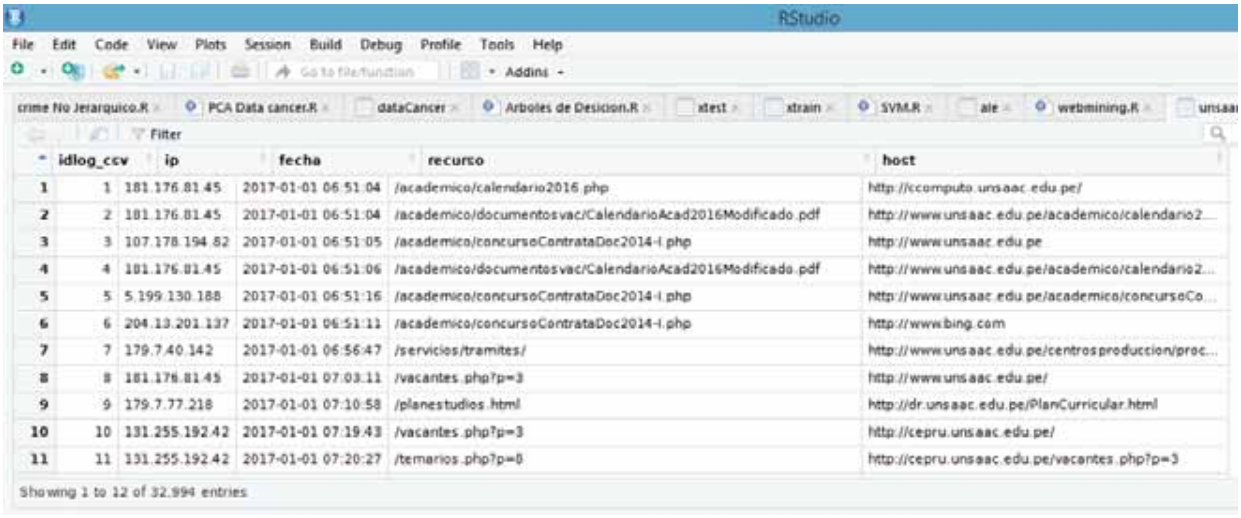

Figura 15.- logs estructurados en RStudio

De la información obtenida podríamos interpretar lo siguiente:

- Los log son tipos de datos no estructurados, lo cual dificulta tremendamente cualquier tipo de análisis tradicional que se pretenda hacer sobre ellos.
- Los log registran información importante como la dirección ip que solicito un recurso en específico, la fecha y hora en que lo hizo, y si este existe o no en el servidor, pero también

registra recursos que no se solicitaron directamente como las hojas de estilo o los javascript solo por mencionar algunos ejemplos.

- Se obtuvo información de más de 50,000 recursos que presentaban el código de error 404, es decir que aun los siguen solicitando sin estos ya encontrarse en el servidor.
- Del código ejecutado, tenemos 7074 usuarios únicos y 2005 recursos solicitados, y a partir de esto, realizaremos el análisis respectivo.

3.7.3.2.2 INTEGRACION Y TRANSFORMACION DE DATOS

En esta parte se tienen los 41 archivos logs en una sola base de datos, donde luego de haber pasado previamente, la limpieza y estructuración correspondiente, destacamos el hecho de que se obtuvo 32994 logs listos para ser analizados.

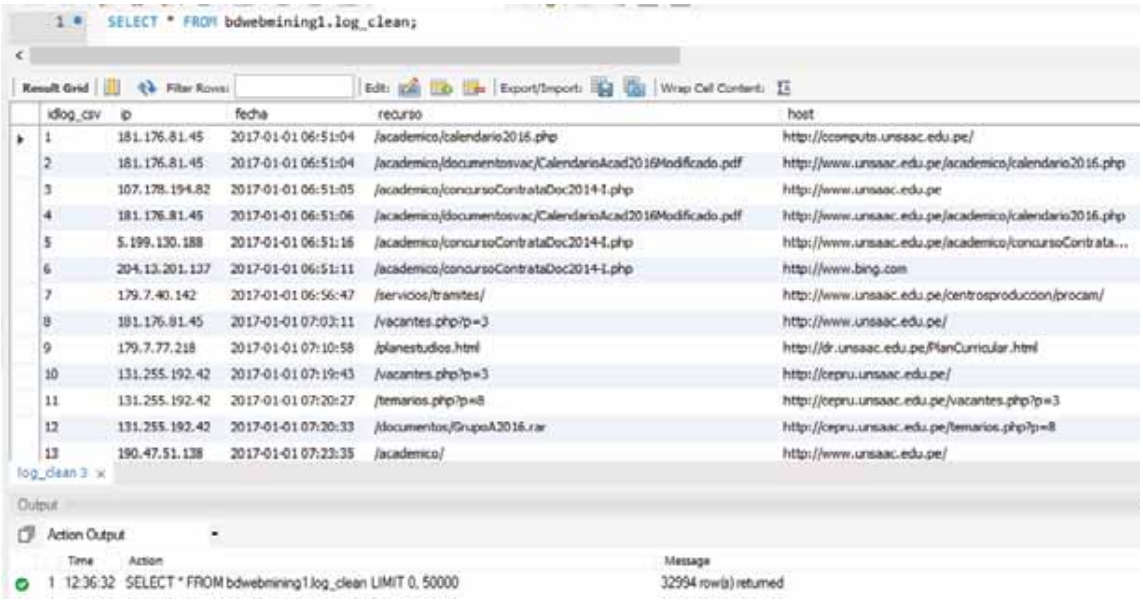

Figura 16.- logs limpios y estructurados.

A continuación, se muestran los datos con los cuales realizaremos la minería de datos, como se puede apreciar analizaremos 32994 registro de log.

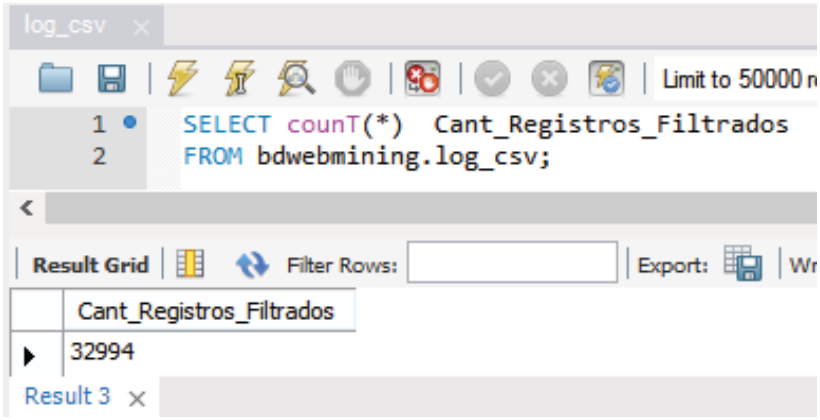

Figura 17.- Cantidad de log filtrados para la mineria de datos

3.7.3.2.3 TRANSACCIONES DE USUARIO APARTIR DE LA BASE DE DATOS

Una vez estructurados los log, se procedió a obtener las transacciones, para lo cual se utilizó R y RStudio (RStudio, 2018), se destaca el hecho de que se realizó una conversión de los datos estructurados en MySQL a un archivo en formato CSV (valor separado por comas), para de esta forma poderlos tratar con R, a continuación, mediante los siguientes scripts, se detalla la obtención de las transacciones.

```
#************************************************
#************************************************
#****** MINERIA DE USO WEB UNSAAC 2017 ***********
#************************************************
#************************************************
#*** AUTOR: WILLIAN ZAMALLOA PARO *****************
#************************************************
# IMPORTACIÓN DIRECTA DE LOS DATOS A UN OBJETO TIPO TRANSACTION
#
=====================================================================
=
library(tidyverse)
library(magrittr)
library(arules)
library("arulesViz")
transacciones <- read.transactions(file = "UNSAAC_LOG_2017_completo_2019.csv",
                                   format = "single",
                                   sep = ";",
                                   cols = c("ip", "recurso"),
                                   rm.duplicates = TRUE)
# MATRIZ DE TRANSACCIONES
transacciones
# VISUALIZAMOS LAS 5 PRIMERAS TRANSACCIONES
inspect(transacciones[1:5])
```

Estas líneas de código, realizan la lectura de los log, desde el archivo denominado **"UNSAAC_LOG_2017_completo_2019.csv",** además configuramos en formato simple, especificamos que los valores están separados por **";",** determinamos que trabajaremos con las columnas **"ip", "recurso",** y removemos los duplicados si los hubiese. Seguidamente visualizamos la cantidad de filas y columnas que posee la matriz de transacciones y luego las 5 primeras transacciones.

```
> transacciones
transactions in sparse format with
 7074 transactions (rows) and
 2005 items (columns)
> # VISUALIZAMOS LAS 5 PRIMERAS TRANSACCIONES
> inspect(transacciones[1:5])
    items                                                  transactionID
[1] {/index.php/component/k2/item/256-centro-de-computo}   10.10.128.162
[2] {/index.php/component/k2/item/258-instituto-de-idiomas} 10.10.130.130
[3] {/informacion.php,
     /nivoslider/index.html}                               100.36.106.99
[4] {/component/users/?view=registration,
     /index.php/component/users/?view=registration,
     /index.php?option=com_users&view=registration,
     /option=com_user?view=register}                       104.206.96.107
[5] {/component/users/?view=registration,
     /index.php/component/users/?view=registration,
     /index.php?option=com_users&view=registration,
     /option=com_user?view=register}                       104.223.31.75
>
```

Figura 18.- transacciones de la base de datos.

Como se puede apreciar en la figura 18 se tiene una matriz de 7074 filas y 2005 columnas, además los identificadores de las transacciones pasaron a ser las direcciones ip (filas) y los recursos las columnas. Y de esta manera se tiene 2005 usuarios únicos y 7074 recursos únicos.

3.7.3.3 FASE DE DESCUBRIMIENTO DE PREFERENCIAS

3.7.3.3.1 MINERIA DE REGLAS DE ASOCIACION

Se aplicó el algoritmo de Apriori, se determinó que un itemset seria frecuente si aparecía al menos en un mínimo de 75 transacciones. Por lo que el soporte vendrá determinado por este valor, que vendría a ser 0.01 aproximadamente. Además, se determina que el nivel de confianza debe ser 90% o superior. Se puede apreciar que se obtuvieron 11 reglas de asociación.

```
> inspect(sort(x = reglas, decreasing = TRUE, by = "confidence"))
     lhs                                                                      rhs                                             support    confidence lift     count
[1]  {/images/academico/CalendarioModidicado2017I-IIRes310-2017-UNSAAC.pdf,
      /index.php/academico/pre-grado/calendarios-academicos}              => {/images/academico/CronogramaAcademico2017A.pdf} 0.01187447 1.0000000 16.00452  84
[2]  {/images/academico/CalendarioModidicado2017I-IIRes310-2017-UNSAAC.pdf} => {/images/academico/CronogramaAcademico2017A.pdf} 0.02403167 0.9714286 15.54725 170
[3]  {/images/academico/Recalendarizacion2017IIcu323.pdf}                => {/images/academico/CronogramaAcademico2017A.pdf} 0.01865988 0.9565217 15.30868 132
[4]  {/index.php/academico/pre-grado/calendarios-academicos}              => {/images/academico/CronogramaAcademico2017A.pdf} 0.03053435 0.9513419 15.22898 216
[5]  {/asignaturas.php,
      /ciclos.php,
      /nosotros.php,
      /presentacion.php}                                                  => {/infraestructura.php}                            0.01031948 0.9480519 38.54322  73
[6]  {/images/academico/Recalendarizacion2017IIcu323.pdf,
      /index.php/academico/pre-grado/calendarios-academicos}              => {/images/academico/CronogramaAcademico2017A.pdf} 0.01003675 0.9466667 15.15095  71
[7]  {/ciclos.php,
      /infraestructura.php,
      /nosotros.php}                                                      => {/presentacion.php}                               0.01229856 0.9255319 27.62537  87
[8]  {/asignaturas.php,
      /ciclos.php,
      /infraestructura.php,
      /nosotros.php}                                                      => {/presentacion.php}                               0.01031948 0.9240506 27.58116  73
[9]  {/asignaturas.php,
      /nosotros.php,
      /presentacion.php}                                                  => {/infraestructura.php}                            0.01201583 0.9139785 37.15795  85
[10] {/infraestructura.php,
      /nosotros.php}                                                      => {/presentacion.php}                               0.01512581 0.9067797 27.06565 107
[11] {/asignaturas.php,
      /ciclos.php,
      /infraestructura.php,
      /presentacion.php}                                                  => {/nosotros.php}                                   0.01031948 0.9012346 18.69599  73
```

Figura 19.- Reglas generadas

3.7.3.3.2 ANALISIS DE LAS PREFERENCIAS DE NAVECACION

Con las reglas de asociación, se determina que en la regla 11 y la regla 9, solo por mencionar un ejemplo se puede apreciar que los que vieron la página presentacion.php después vieron infraestructura.php o nosotros.php.

Y los patrones se generan a partir de estas reglas, pues al tener la certeza de una probabilidad alta es que se establece un patrón el cual puede formarse entre dos a mas páginas web, como se detallara más adelante.

CAPITULO IV

RESULTADOS Y DISCUSION

4.1 ANALISIS, INTERPRETACION Y DISCUSION DE RESULTADOS

Comenzaremos visualizando el tamaño de las transacciones y la distribución que tiene cada una de estas. Al ejecutar el script, obtenemos el siguiente resultado.

```
#TAMAÑO DE LAS TRANSACCIONES
taman <- size(transacciones)
summary(taman)

#DISTRIBUCION DE LOS CUANTILES
quantile(taman, probs = seq(0,1,0.1))
```

```
> #TAMAÑO DE LAS TRANSACCIONES
> taman <- size(transacciones)
> summary(taman)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   2.000   2.866   3.000 160.000
>
> #DISTRIBUCION DE LOS CUANTILES
> quantile(taman, probs = seq(0,1,0.1))
  0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
   1    1    1    1    1    2    2    3    4    6  160
>
```

Figura 20.- distribucion de los recursos en las transacciones

Se puede apreciar en la figura 20 que la gran mayoría de usuarios visita entre 2 y 4 páginas web, además el 90% de ellos visita como máximo 6 páginas web.

Lo siguiente que se realizo fue un análisis que consiste en identificar cuáles son los items más frecuentes (aquellos que tienen mayor soporte) dentro del conjunto de todas las transacciones de las páginas web. Utilizando la función itemFrequency() se extrajo esta información del conjunto de transacciones que contienen las páginas web.

```
#ITEMS FRECUENTES, SE MUESTRAN LOS 5 PRIMEROS
frecuencia_items <- itemFrequency(x = transacciones, type = "relative")
frecuencia_items %>% sort(decreasing = TRUE) %>% head(5)
```

PAGINA WEB	FRECUENCIA
/index.php/component/k2/item/254-admision	0.11860334
/vacantes.php?p=3	0.11620017
/index.php/component/k2/item/256-centro-de-computo	0.10785977
/index.php	0.06502686
/temarios.php?p=8	0.06304778

Tabla 6.- Los 5 primeros items mas frecuentes.

En la tabla 7 se aprecia la cantidad de veces que aparece cada una de estas páginas en las transacciones.

```
#NUMERO DE TRANSACCIONES EN LAS QUE APARECE CADA ITEM (SE MUESTRAN 5)
frecuencia_items <- itemFrequency(x = transacciones, type = "absolute")
frecuencia_items %>% sort(decreasing = TRUE) %>% head(5)
```

PAGINA WEB	NRO. DE OCURRENCIAS
/index.php/component/k2/item/254-admision	839
/vacantes.php?p=3	822
/index.php/component/k2/item/256-centro-de-computo	763
/index.php	460
/temarios.php?p=8	446

Tabla 7.- Numero de transacciones en las que aparece una pagina web

En la tabla 8, se observa las ocurrencias de cada página web, en este caso es importante estudiar cómo se distribuye el soporte de las paginas individuales en un conjunto de transacciones antes identificar conjuntos de páginas frecuentes o crear reglas de asociación, ya que, dependiendo del caso, tendrá sentido emplear un límite de soporte u otro.

Para este caso el número de páginas es grande, prácticamente todas las paginas son raras, por lo que los soportes son muy bajos. Una vez establecidos la frecuencia de cada una de las páginas procedemos a aplicar el algoritmo de Apriori, el cual nos ayudara a identificar tanto conjunto de páginas frecuentes como reglas de asociación que superen un determinado soporte y confianza.

```
#APRIORI
#CALCULAMOS EL SOPORTE
soporte <- 75 / dim(transacciones)[1]
#GENERAMOS EL CONJUNTO DE ITEMS
itemsets <- apriori(data = transacciones,
                    parameter = list(support = soporte,
                                     minlen = 1,
                                     maxlen = 20,
                                     target = "frequent itemset"))

#LOS VISUALIZAMOS
summary(itemsets)
```

Se procede a extraer el conjunto de páginas, donde se incluye también a aquellos que están conformados por una sola página, y que hayan sido visitados al menos 75 veces, número que varía en función de la cantidad de transacciones, se tomó en que cuenta la mayor cantidad de visualizaciones esta entre 1 y 100, y de todos los valores con los que se probó, a partir de esta cantidad (75), cuyo soporte es aproximadamente 0.01, dio como resultado un número suficiente de conjunto de páginas y reglas de asociación, que permitió encontrar mejores posibilidades de análisis.

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen            target   ext
         NA    0.1    1 none FALSE            TRUE       5    0.01      1     20 frequent itemsets FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 70

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[2005 item(s), 7074 transaction(s)] done [0.00s].
sorting and recoding items ... [45 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 done [0.00s].
writing ... [135 set(s)] done [0.00s].
creating S4 object  ... done [0.02s].
>
```

Figura 21.- ejecucion del algoritmo Apriori

De la figura 21, se puede apreciar que se tiene, 135 conjuntos de páginas frecuentes, que superan el soporte mínimo de 0.01.

```
> summary(itemsets)
set of 135 itemsets

most frequent items:
      /ciclos.php /vacantes.php?p=3     /nosotros.php  /asignaturas.php /presentacion.php           (Other)
               30                 27                26                22                22               142

element (itemset/transaction) length distribution:sizes
 1  2  3  4  5
45 54 29  6  1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   2.000   1.993   3.000   5.000

summary of quality measures:
    support           count
 Min.   :0.01004   Min.   : 71.0
 1st Qu.:0.01145   1st Qu.: 81.0
 Median :0.01442   Median :102.0
 Mean   :0.02069   Mean   :146.3
 3rd Qu.:0.02078   3rd Qu.:147.0
 Max.   :0.11860   Max.   :839.0

includes transaction ID lists: FALSE

mining info:
          data ntransactions support confidence
 transacciones          7074    0.01          1
>
```

Figura 22.- conjuntos de paginas

De la figura 22, se puede apreciar que la mayoría de conjuntos de páginas frecuentes (54), están conformados por 2 páginas.

```
# REGLAS DE ASOCIACION
# CALCULAMOS EL SOPORTE
soporte <- 75 / dim(transacciones)[1]
# GENERAMOS LAS REGLAS
reglas <- apriori(data = transacciones,
                  parameter = list(support = soporte,
                                   confidence = 0.90,
                                   # Se especifica que se creen reglas
                                   target = "rules"))
# VISUALIZMOS DETALLES DE LAS REGLAS
summary(reglas)
```

Se comenzará analizando las páginas web que los usuarios navegan secuencialmente (en este caso se estructuraron en función a la fecha y hora en la que acceden), entonces para ello analizaremos las variables recurso e ip, los cuales ayudaran a ver en este caso, que ip (usuario), visito que recurso esto es lo que muestra la ejecución del código en R.

Ahora se procederá a realizar la creación de las reglas de asociación las cuales permitirán analizar las preferencias de navegación. Utilizaremos el paquete arules de R, el cual permitirá ver la secuencia de clicks como transacciones, el número de veces que los 7074 recursos son accedidos por los usuarios además de la frecuencia relativa con la que los recursos son requeridos por los 2005 usuarios, de tal forma que se pueda observar el conjunto de ítems frecuentes. Para de esta manera poder graficarlos y apreciarlos mejor. Además, se estableció, que la confianza (confidence) para poder tener la certeza de que una regla sea relevante, es del 90% o superior.

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target   ext
        0.9    0.1    1 none FALSE            TRUE       5    0.01      1     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 70

set item appearances ...[0 item(s)] done [0.01s].
set transactions ...[2005 item(s), 7074 transaction(s)] done [0.02s].
sorting and recoding items ... [45 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 done [0.00s].
writing ... [11 rule(s)] done [0.02s].
creating S4 object  ... done [0.03s].
```

Figura 23.- Generacion de reglas de asociacion

En la figura 23, se puede apreciar que, se han generado 11 reglas, que cumplen con una confianza del 90% o superior.

```
> summary(reglas)
set of 11 rules

rule length distribution (lhs + rhs):sizes
2 3 4 5
3 3 2 3

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   2.500   3.000   3.455   4.500   5.000

summary of quality measures:
    support         confidence          lift            count
 Min.   :0.01004   Min.   :0.9012   Min.   :15.15   Min.   : 71.0
 1st Qu.:0.01032   1st Qu.:0.9190   1st Qu.:15.43   1st Qu.: 73.0
 Median :0.01202   Median :0.9467   Median :18.70   Median : 85.0
 Mean   :0.01505   Mean   :0.9405   Mean   :23.08   Mean   :106.5
 3rd Qu.:0.01689   3rd Qu.:0.9540   3rd Qu.:27.60   3rd Qu.:119.5
 Max.   :0.03053   Max.   :1.0000   Max.   :38.54   Max.   :216.0

mining info:
         data ntransactions support confidence
 transacciones          7074    0.01        0.9
> |
```

Figura 24.- detalle de las 11 reglas generadas.

Como se puede apreciar en la Figura 24, la mayoría de las 11 reglas está formada, por 2, 3 ó 5 páginas en la parte izquierda (antecedente) de la regla, que las confianzas varían entre 90% y 95%, además una de ellas tiene una confianza del 100%, y que el soporte máximo es de 0.03 aproximadamente.

4.2 VALIDACIÓN DE LAS REGLAS DE ASOCIACIÓN

Efectivamente, las razones por la cuales, se estructuraron los archivos log, es porque este conjunto, es una base de datos, pero no una, estructurada, razón por la cual, se realizó el proceso de estructuración, según (Olaya, 2014, pág. 206) es importante por las siguientes razones, que afectan directamente a los datos:

- Mayor independencia. Los datos son independientes de las aplicaciones que los usan, así como de los usuarios.
- Mayor disponibilidad. Se facilita el acceso a los datos desde contextos, aplicaciones y medios distintos, haciéndolos útiles para un mayor número de usuarios.
- Mayor seguridad (protección de los datos). Por ejemplo, resulta más fácil replicar una base de datos para mantener una copia de seguridad que hacerlo con un conjunto de ficheros almacenados de forma no estructurada. Además, al estar centralizado el acceso a los datos, existe una verdadera sincronización de todo el trabajo que se haya podido hacer sobre estos (modificaciones), con lo que esa copia de seguridad servirá a todos los usuarios.

Esto tiene una consecuencia directa sobre los resultados que se obtienen de la explotación de la base de datos, presentándose al respecto ventajas como, por ejemplo:

- Mayor coherencia. La mayor calidad de los datos que se deriva de su mejor gestión deriva en mayor calidad de los resultados.
- Mayor eficiencia. Facilitando el acceso a los datos y haciendo más sencilla su explotación, la obtención de resultados es más eficiente.
- Mayor valor informativo. Resulta más sencillo extraer la información que los datos contienen, ya que uno de los cometidos de la base de datos es aumentar el valor de estos como fuente de información.

Por último, los usuarios de la base de datos también obtienen ventajas al trabajar con estas, entre los que cabe citar:

- Mayor facilidad y sencillez de acceso. El usuario de la base de datos se debe preocupar únicamente de usar los datos, disponiendo para ello de las herramientas adecuadas y de una estructura sólida sobre la que apoyarse.
- Facilidad para reutilización de datos. Esto es, facilidad para compartir.

En este caso se evidencia pues se construyeron programas a nivel de aplicación y base de datos, que facilitaron la estructuración, como se hace constar en los anexos, aunque tomaron un tiempo considerable al final se logró el objetivo, que se buscaba, pues los log quedaron listos para poder crear las reglas y obtener las preferencias de navegación.

A continuación, validamos que las preferencias de los usuarios tienen una confianza mayor o igual al 90 %. Para lo cual se realizará una inspección.

```
inspect(sort(x = reglas, decreasing = TRUE, by = "confidence"))
```

# REGLA	SUPPORT	CONFIDENCE	CERTEZA DE LA REGLA
1	**0.01187447**	**1.0000000**	**100%**
2	**0.02403167**	**0.9714286**	**97.14%**
3	**0.01865988**	**0.9565217**	**95.65%**
4	**0.03053435**	**0.9515419**	**95.15%**
5	**0.01031948**	**0.9480519**	**94.81%**
6	**0.01003675**	**0.9466667**	**94.67%**
7	**0.01229856**	**0.9255319**	**92.55%**
8	**0.01031948**	**0.9240506**	**92.41%**
9	**0.01201583**	**0.9139785**	**91.40%**
10	**0.01512581**	**0.9067797**	**90.68%**
11	**0.01031948**	**0.9012346**	**90.12%**

Tabla 8.- Soporte y confianza de las reglas de asociacion

Como podemos observar en la tabla 9, la confianza de la regla es más del 90% y se han obtenido 11 reglas, que cumplen con esta exactitud. Además para que se pueda cuantificar la calidad de las reglas, y que de esta forma reflejen relaciones reales entre páginas, se calculara otras métricas, de esta forma se evaluaran las reglas mediante: lift, coverage y fisher exact test, para lo cual usaremos la función interestMeasure(), que evaluara las reglas creadas por la el algoritmo Apriori.

```
metricas <- interestMeasure(reglas, measure = c("coverage", "fishersExactTest"),
                            transactions = transacciones)
métricas
```

COVERAGE	FISHERS EXACT TEST
0.01950806	**7.923330e-159**
0.02473848	**6.597614e-212**
0.03208934	**3.527721e-269**
0.01060221	**1.149170e-82**
0.01187447	**2.480459e-105**
0.01668080	**9.792180e-156**
0.01314673	**3.569656e-137**
0.01328810	**9.373928e-127**
0.01145038	**6.134997e-90**
0.01116766	**2.997388e-105**
0.01088493	**1.192349e-119**

Tabla 9.- Calculo del Coverage y Fishers Exsact Test

En la tabla 10, se observa el calculó del coverage (cobertura), que viene a ser el soporte (support), de la parte izquierda de la regla, la cual es la frecuencia con la que el antecedente aparece en el conjunto de las transacciones. Por otro lado, el valor del fishers exact test, viene a ser el valor asociado a la probabilidad de observar una regla solo por casualidad o por azar.

De la tabla se puede apreciar que, el valor de coverage, es muy aproximado a los del soporte previamente calculado, lo que nos indica que las reglas son válidas, también el valor del fishers exact test, como se puede apreciar es muy bajo, indicándonos que la generación de las reglas no está dispuesta por azar, siendo el valor más alto **1.149170 x 10^{-82}**.

Se vio por conveniente utilizar estos valores, para validar las reglas con las métricas ya generadas, incluyendo también una métrica más, que vendría a ser el lift, para lo cual se evaluara dicho valor.

```
#MEDICION DE CALIDAD DE LAS 11 REGLAS CON LAS METRICAS
quality(reglas) <- cbind(quality(reglas), metricas)
#ORDENAMOS EL RESULTADO EN FUNCION A LA CONFIANZA
df_reglas %>%  arrange(desc(confidence))
```

```
      support confidence     lift count   coverage fishersExactTest
1  0.01187447  1.0000000 16.00452    84 0.01187447    2.480459e-105
2  0.02403167  0.9714286 15.54725   170 0.02473848    6.597614e-212
3  0.01865988  0.9565217 15.30868   132 0.01950806    7.923330e-159
4  0.03053435  0.9515419 15.22898   216 0.03208934    3.527721e-269
5  0.01031948  0.9480519 38.54322    73 0.01088493    1.192349e-119
6  0.01003675  0.9466667 15.15095    71 0.01060221     1.149170e-82
7  0.01229856  0.9255319 27.62537    87 0.01328810    9.373928e-127
8  0.01031948  0.9240506 27.58116    73 0.01116766    2.997388e-105
9  0.01201583  0.9139785 37.15795    85 0.01314673    3.569656e-137
10 0.01512581  0.9067797 27.06565   107 0.01668080    9.792180e-156
11 0.01031948  0.9012346 18.69599    73 0.01145038     6.134997e-90
```

Figura 25.- Calidad de las 11 reglas en funcion a 3 metricas.

Como se aprecia en la figura 25 al aplicar las métricas no hay variaciones significativas a los resultados ya obtenidos, y el lift como se puede observar es alto, cabe destacar que el lift compara la frecuencia observada de una regla, con la frecuencia esperada simplemente por azar, es decir si la regla no existe realmente, pues cuanto más se aleje de 1 el valor del lift, mas evidencia de que dicha regla no se debe a un factor aleatorio, lo que se evidencia en este caso, es que el patrón de la regla, es real.

Dicho esto, se comprueba la validez de la hipótesis planteada, pues al estructurarse los log, del servidor web, se obtuvieron 11 reglas de asociación, que muestran las preferencias de navegación de los usuarios de las páginas web de la UNSAAC, dichas reglas obtenidas tienen una certeza superior al 90%, de las cuales se verifico la calidad de estas mediante métricas, con lo que se pudo establecer que son válidas, valiosas y de calidad.

4.3 PRESENTACION DE RESULTADOS

Para la presentación de resultados, se utilizó RStudio y R (RStudio, 2018), software de código abierto orientado a labores estadísticas e ideal para minería de datos y análisis de datos. Muy utilizado y que tiene la característica de realizar todo por línea de comandos.

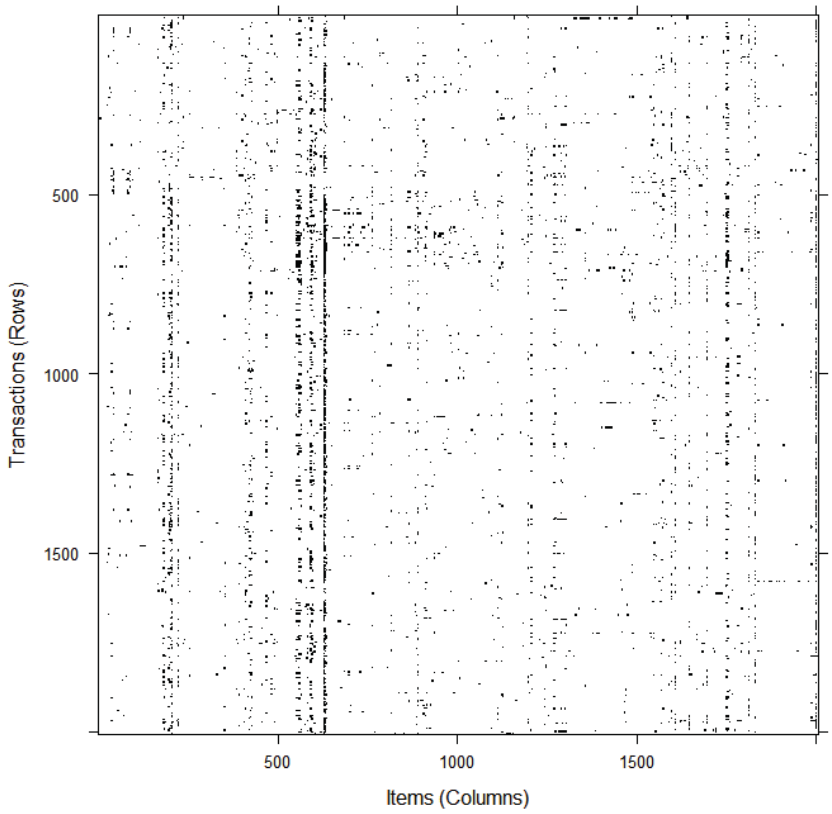

Figura 26.- matriz dispersa

En la figura 26 se pude apreciar la matriz dispersa, la cual como se observa tiene una distribución no estructurada, donde los puntos negros son todos los valores diferentes de cero, es decir las páginas que aparecen en cada una de las transacciones.

Como ya se mencionó anteriormente utilizaremos reglas de asociación para la generación de dichas reglas, por lo cual se utilizó el algoritmo de Apriori, que trae R, en el paquete arules, al cual se le dio configuro con un soporte mínimo del 0.01 y una confianza mínima del 90%.

	lhs		rhs	support	confidence	lift	count
[1]	{/images/academico/CalendarioModidicado2017I-IIRes310-2017-UNSAAC.pdf, /index.php/academico/pre-grado/calendarios-academicos}	=>	{/images/academico/CronogramaAcademico2017A.pdf}	0.01187447	1.0000000	16.00452	84
[2]	{/images/academico/CalendarioModidicado2017I-IIRes310-2017-UNSAAC.pdf}	=>	{/images/academico/CronogramaAcademico2017A.pdf}	0.02403167	0.9714286	15.54725	170
[3]	{/images/academico/Recalendarizacion2017IIcu323.pdf}	=>	{/images/academico/CronogramaAcademico2017A.pdf}	0.01865988	0.9565217	15.30868	132
[4]	{/index.php/academico/pre-grado/calendarios-academicos}	=>	{/images/academico/CronogramaAcademico2017A.pdf}	0.03053435	0.9515419	15.22898	216
[5]	{/asignaturas.php, /ciclos.php, /nosotros.php, /presentacion.php}	=>	{/infraestructura.php}	0.01031948	0.9480519	38.54322	73
[6]	{/images/academico/Recalendarizacion2017IIcu323.pdf, /index.php/academico/pre-grado/calendarios-academicos}	=>	{/images/academico/CronogramaAcademico2017A.pdf}	0.01003675	0.9466667	15.15095	71
[7]	{/ciclos.php, /infraestructura.php, /nosotros.php}	=>	{/presentacion.php}	0.01229856	0.9255319	27.62537	87
[8]	{/asignaturas.php, /ciclos.php, /infraestructura.php, /nosotros.php}	=>	{/presentacion.php}	0.01031948	0.9240506	27.58116	73
[9]	{/asignaturas.php, /nosotros.php, /presentacion.php}	=>	{/infraestructura.php}	0.01201583	0.9139785	37.15795	85
[10]	{/infraestructura.php, /nosotros.php}	=>	{/presentacion.php}	0.01512581	0.9067797	27.06565	107
[11]	{/asignaturas.php, /ciclos.php, /infraestructura.php, /presentacion.php}	=>	{/nosotros.php}	0.01031948	0.9012346	18.69599	73

Figura 27.- Reglas de asociación con una confianza mayor al 90%

#	LHS	=>	RHS	SUPPORT	CONFIDENCE	LIFT
1	**{/images/academico/CalendarioModidicado2017I-IIRes310-2017-UNSAAC.pdf, /index.php/academico/pre-grado/calendarios-academicos}**	=>	**{/images/academico/CronogramaAcademico2017A.pdf}**	**0.011874**	**1.000000**	**16.00452**
2	**{/images/academico/CalendarioModidicado2017I-IIRes310-2017-UNSAAC.pdf}**	=>	**{/images/academico/CronogramaAcademico2017A.pdf}**	**0.024032**	**0.971429**	**15.54725**
3	**{/images/academico/Recalendarizacion2017IIcu323.pdf}**	=>	**{/images/academico/CronogramaAcademico2017A.pdf}**	**0.01866**	**0.956522**	**15.30868**
4	**{/index.php/academico/pre-grado/calendarios-academicos}**	=>	**{/images/academico/CronogramaAcademico2017A.pdf}**	**0.030534**	**0.951542**	**15.22898**
5	**{/asignaturas.php, /ciclos.php, /nosotros.php, /presentacion.php}**	=>	**{/infraestructura.php}**	**0.010319**	**0.948052**	**38.54322**
6	**{/images/academico/Recalendarizacion2017IIcu323.pdf, /index.php/academico/pre-grado/calendarios-academicos}**	=>	**{/images/academico/CronogramaAcademico2017A.pdf}**	**0.010037**	**0.946667**	**15.15095**
7	**{/ciclos.php, /infraestructura.php, /nosotros.php}**	=>	**{/presentacion.php}**	**0.012299**	**0.925532**	**27.62537**
8	**{/asignaturas.php, /ciclos.php, /infraestructura.php, /nosotros.php}**	=>	**{/presentacion.php}**	**0.010319**	**0.924051**	**27.58116**
9	**{/asignaturas.php, /nosotros.php, /presentacion.php}**	=>	**{/infraestructura.php}**	**0.012016**	**0.913979**	**37.15795**
10	**{/infraestructura.php, /nosotros.php}**	=>	**{/presentacion.php}**	**0.015126**	**0.90678**	**27.06565**
11	**{/asignaturas.php, /ciclos.php, /infraestructura.php, /presentacion.php}**	=>	**{/nosotros.php}**	**0.010319**	**0.901235**	**18.69599**

Tabla 10.- Reglas de asociacion en formato tabla

Una vez realizado esto podemos observar tanto en la tabla11 como en la figura 27, las 11 reglas de asociación con su antecedente y su respectivo consecuente que cumplen con lo solicitado en el paso anterior, como por ejemplo, la regla $/images/academico/CalendarioModidicado2017I-IIRes310-2017-UNSAAC.pdf \rightarrow /images/academico/CronogramaAcademico2017A.pdf$, tiene un soporte de 0.02, un lift 15.54 el cual es mayor a 1 y una confidencialidad de 97.14%, lo que la hace muy confiable.

Ahora generamos los gráficos que nos permitirán visualizar las preferencias de navegación, generados a partir de las reglas de asociación.

```
#GRAFICAMOS LAS REGLAS
plot(reglas) #scatter

#GRAFICA A DOS VALORES
plot(reglas, method = "two-key plot") #

# GRAFICO DE REGLAS CON MAYOR CONFIANZA, GRAFO INTERACTIVO
plot(reglas, method = "graph", control = list(type="items"), engine="htmlwidget",
    igraphLayout = "layout_in_circle")

#PATRONES DE NAVEGACION EN COORDENADAS PARALELAS
plot(reglas, method = "paracoord",alpha=.20, reorder=TRUE)
```

Ahora se puede, apreciar todas las reglas de asociación con una confianza mayor al 90%, del mismo modo apreciamos que el valor de lift es mayor a 1, que indica que ese conjunto aparece una cantidad de veces superior a lo esperado bajo condiciones de independencia.

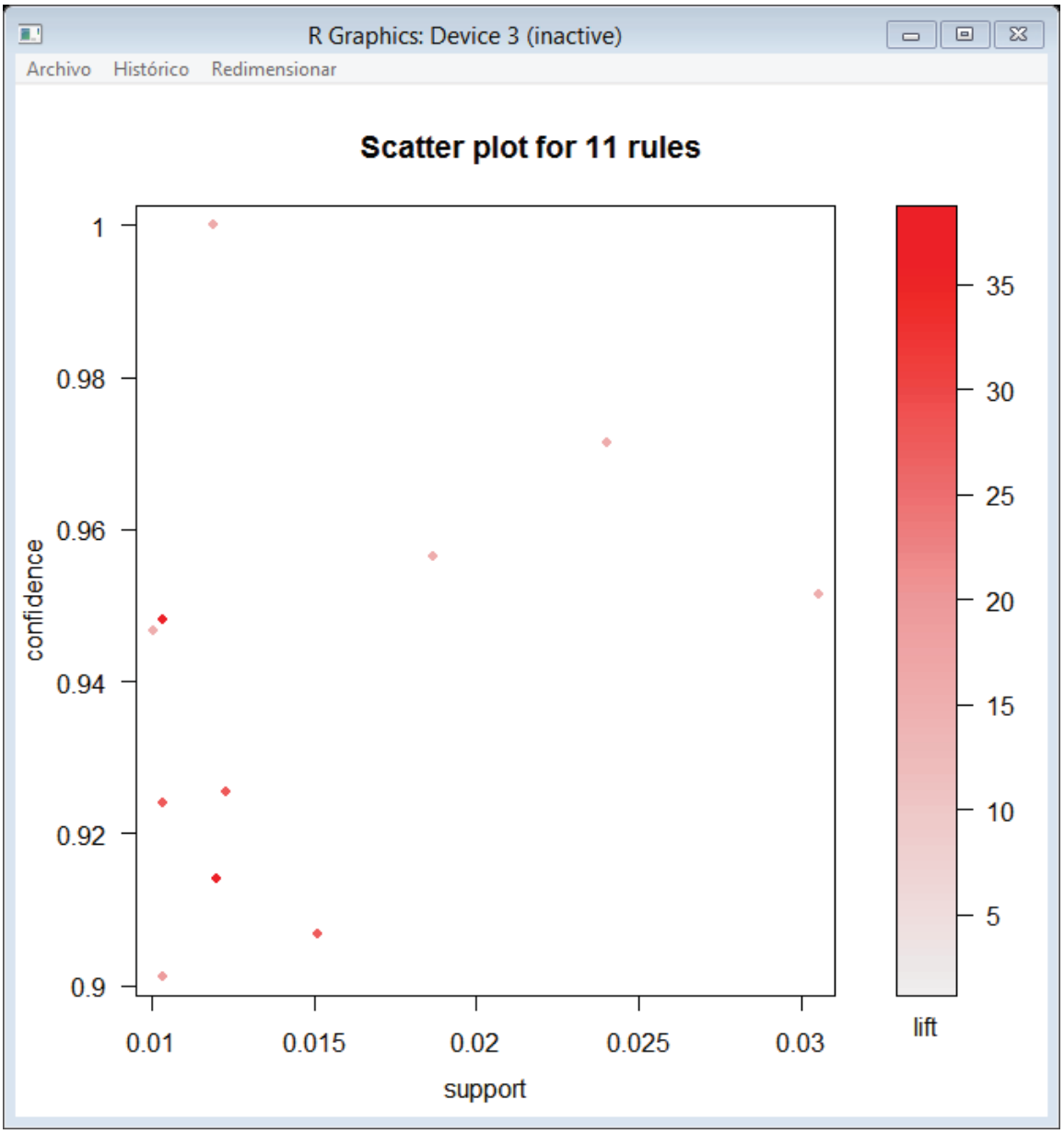

Figura 28.- Gráfico de dispersión de las reglas de asociación

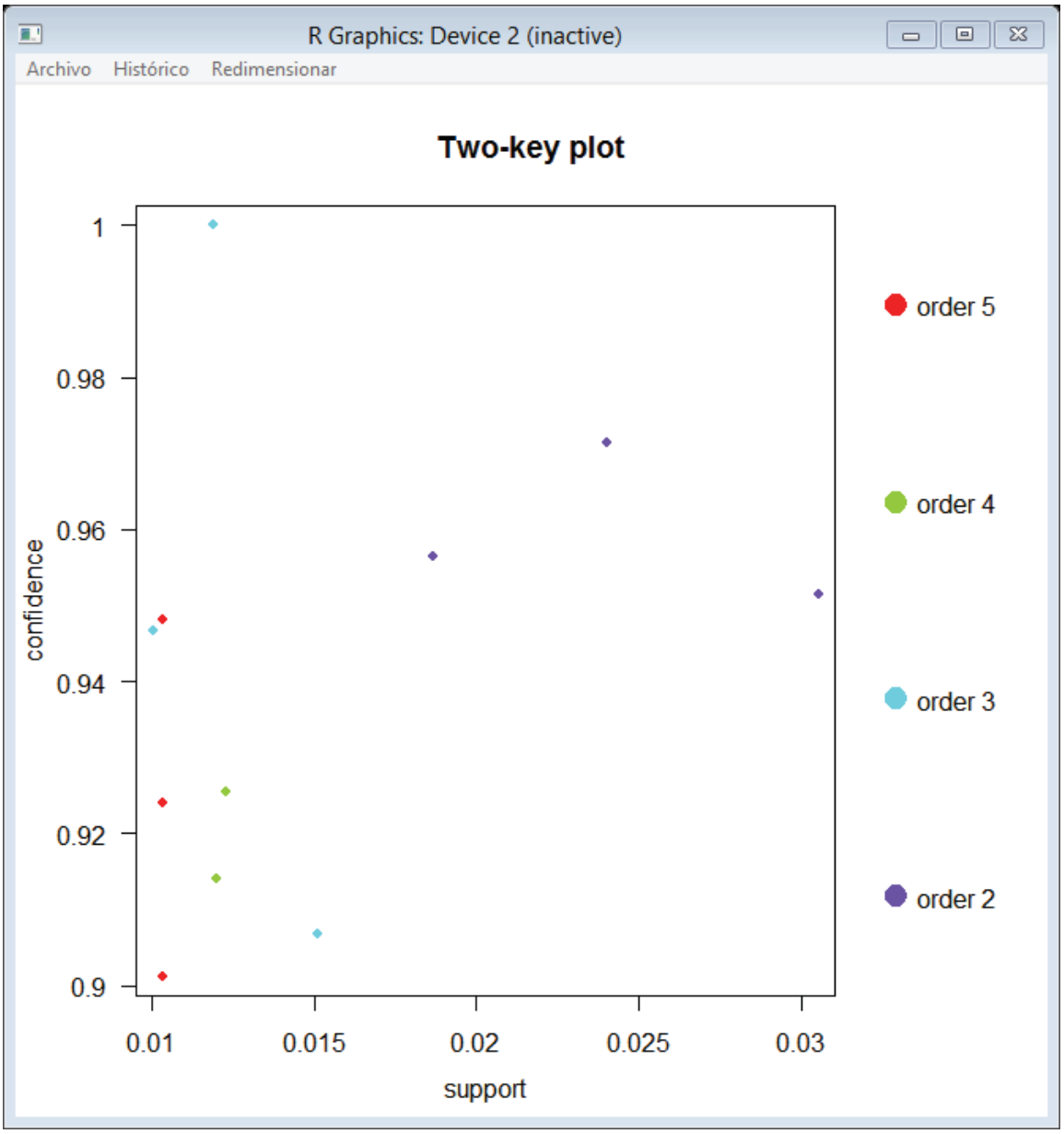

Figura 29.- gráfico de dos claves

En ambas figuras en la 28 y 29 tanto en la de dispersión y en la de dos valores respectivamente, se observa que los métodos de visualización dibujan un diagrama de dispersión bidimensional con diferentes medidas de interés (parámetros) en los ejes y una tercera medida (sombreado) está representada por el color de los puntos. Hay un valor especial para el sombreado llamado "orden" que produce un gráfico de dos claves donde el color de los puntos representa la longitud (orden) de la regla, donde por ejemplo el orden 3 significa que tenemos dos páginas en el antecedente y una página en el consecuente, los cuales se pueden apreciar en la gráfica.

Figura 30.- Matriz de las reglas de asociacion

En la figura 30 se puede ver la matriz de influencia, la cual muestra la relación entre LHS (left-hand-sides) o parte izquierda de la regla y RHS (right-hand-sides) parte derecha de la regla. Y como agrupan todas las acciones relativas a las paginas, y los círculos muestran la densidad que corresponde a las páginas más visitadas. Visualización basada en matrices agrupadas por clusters, donde los antecedentes (columnas) en la matriz se agrupan. Los grupos están representados por los elementos más interesantes (el ratio de soporte más alto en el grupo para soportar a todas las reglas) en el

grupo. Los globos en la matriz se utilizan para representar con qué consecuente se conectan los antecedentes. Ahora podemos revisar las relaciones que hay entre cada recurso tal y como se aprecia en la figura 31 la representación de los grafos de las reglas de asociación.

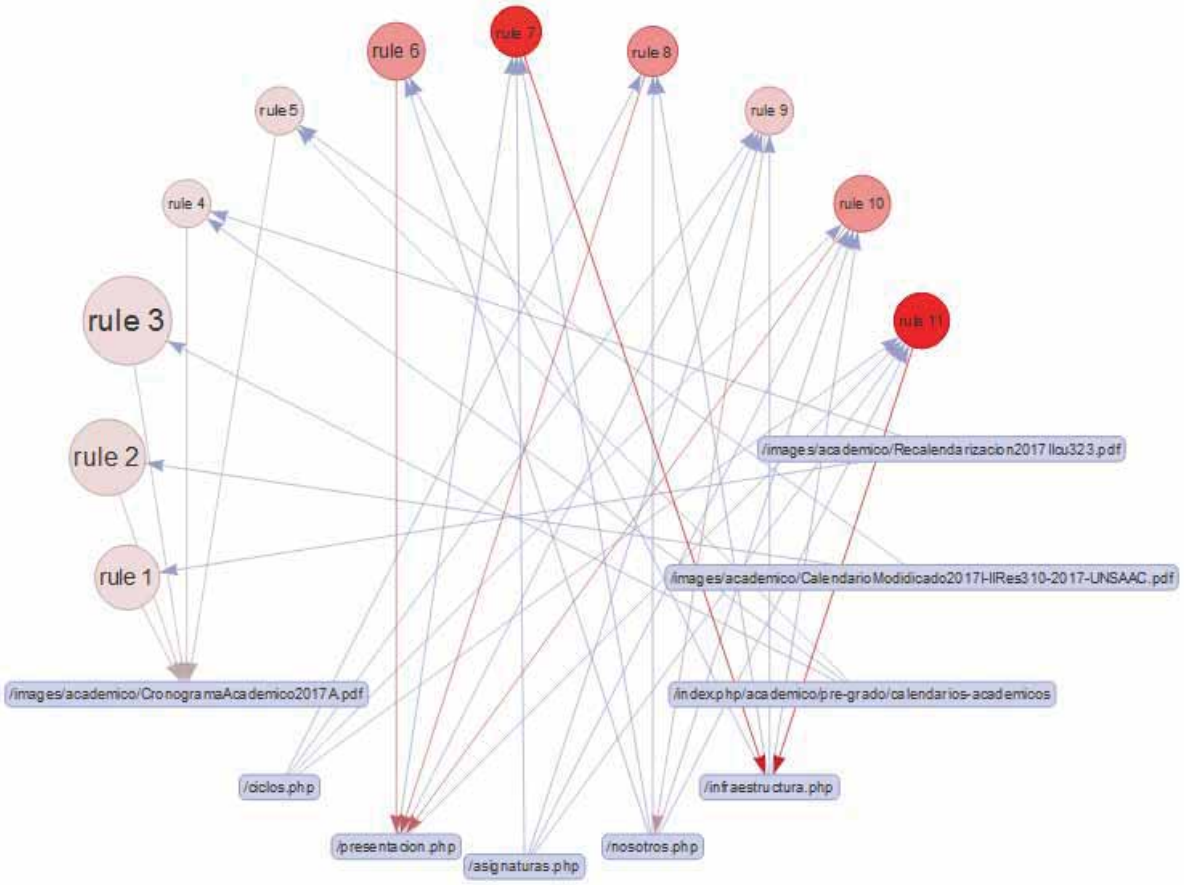

Figura 31-. Grafo de las reglas de asociacion.

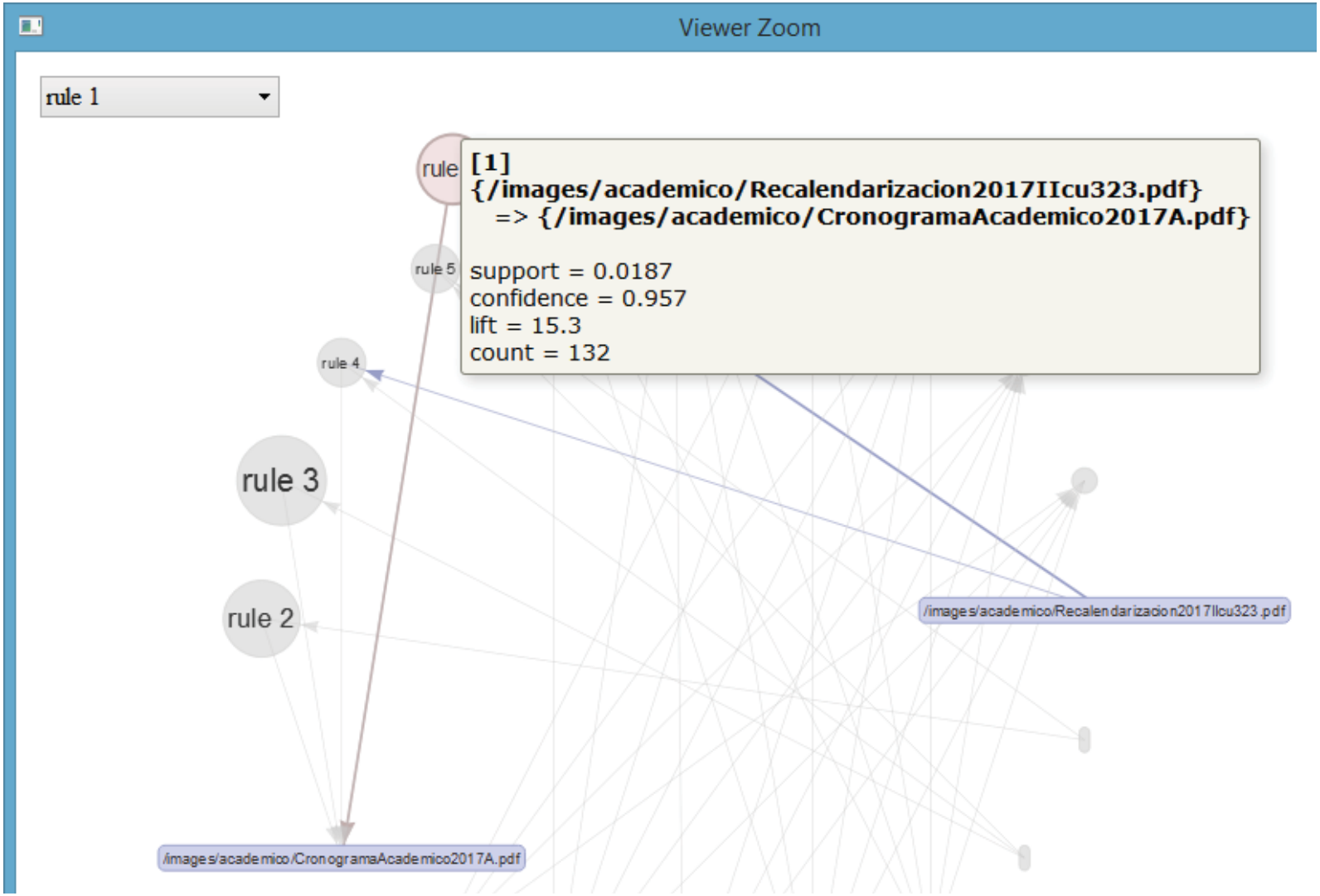

Figura 32.- Regla 1

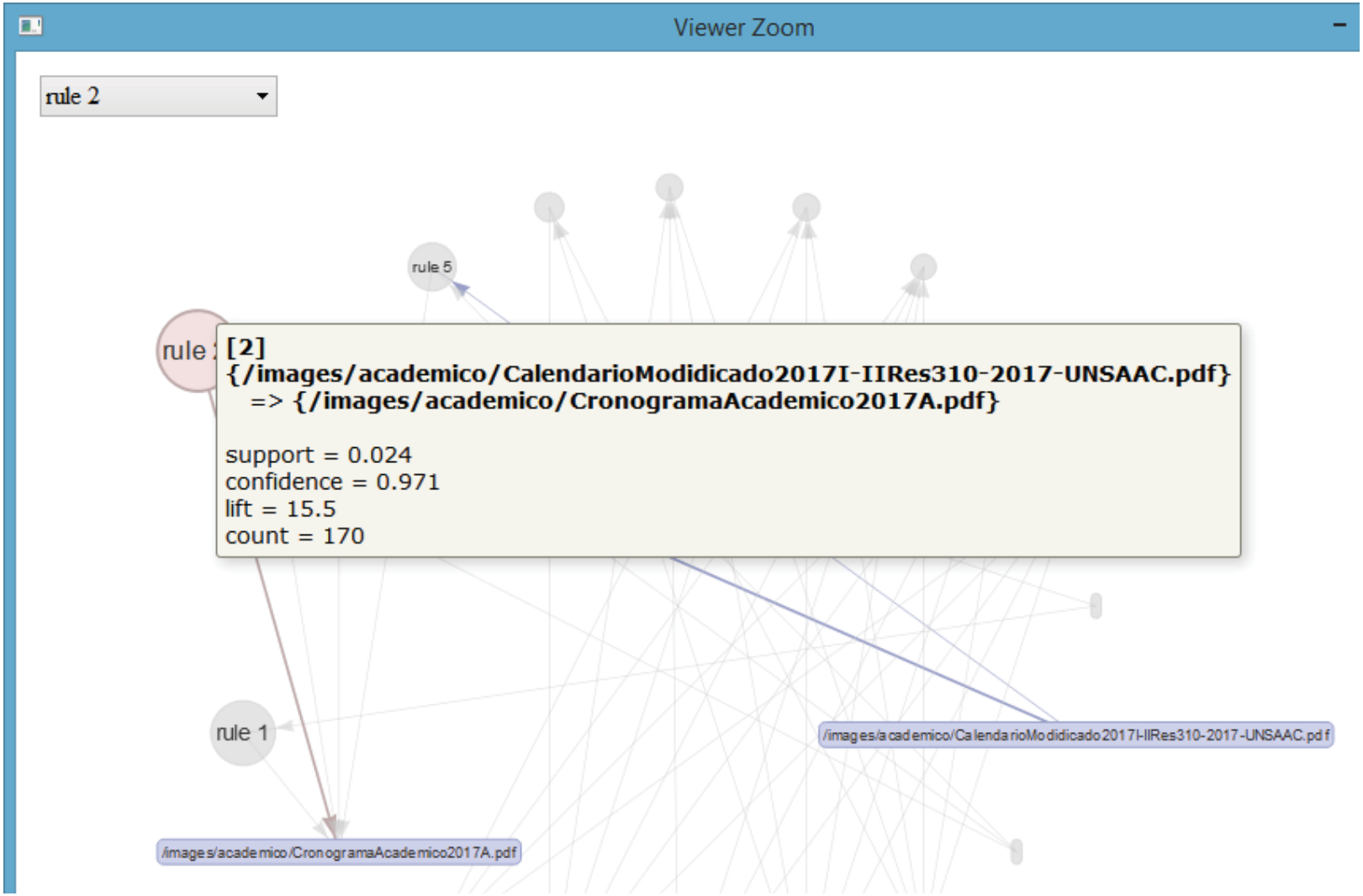

Figura 33.- Reglas 2

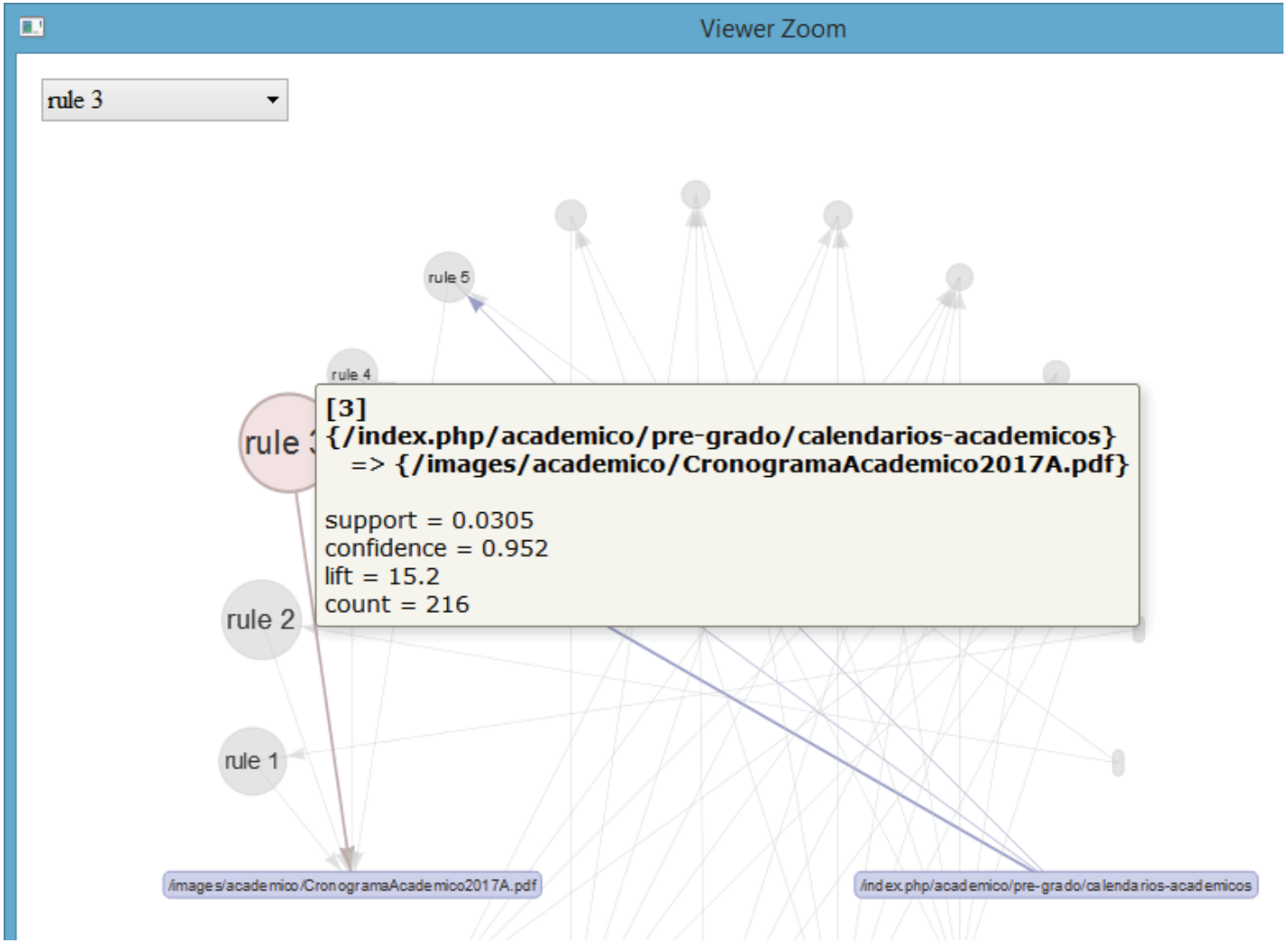

Figura 34.- Regla 3

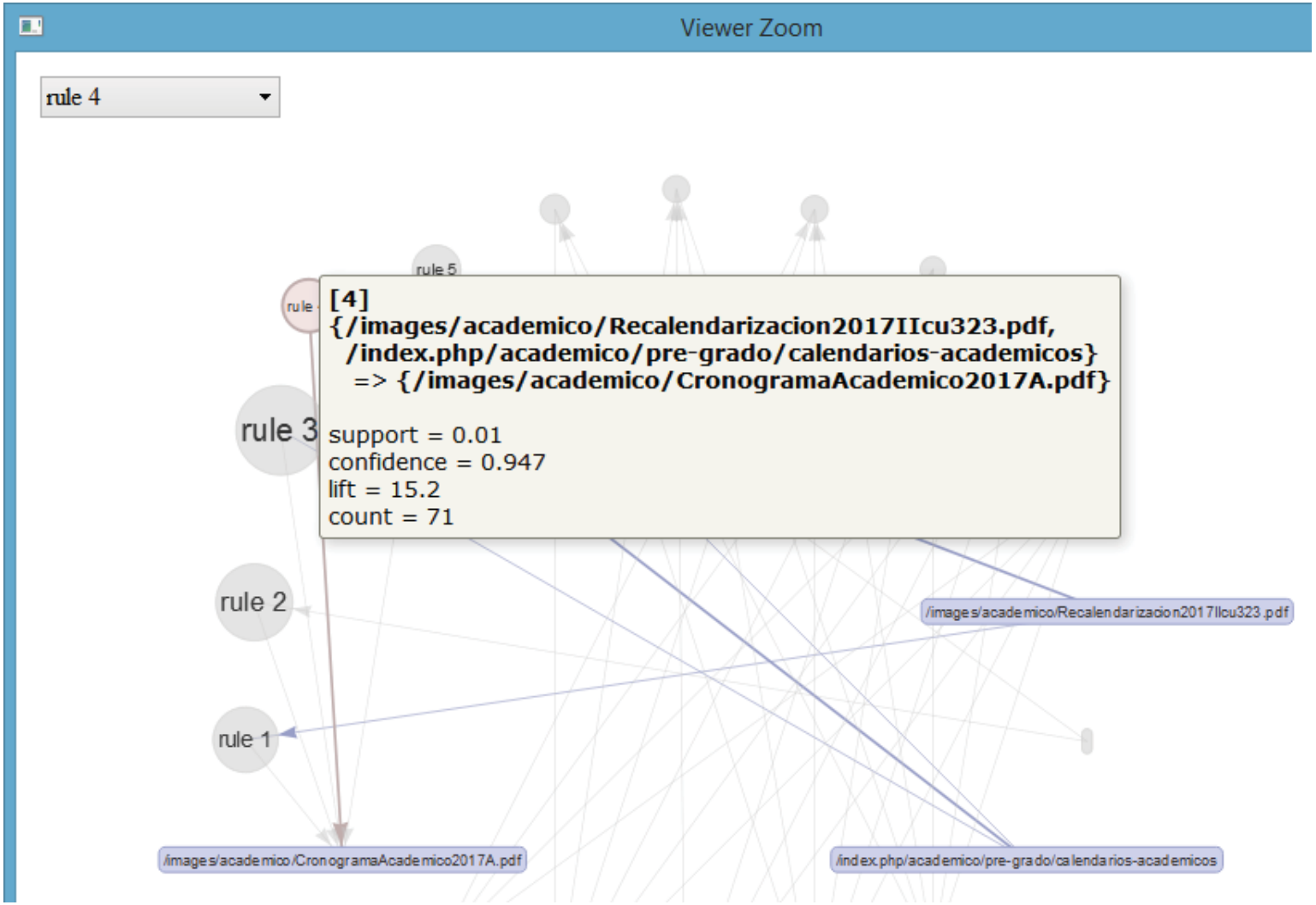

Figura 35.- Regla 4

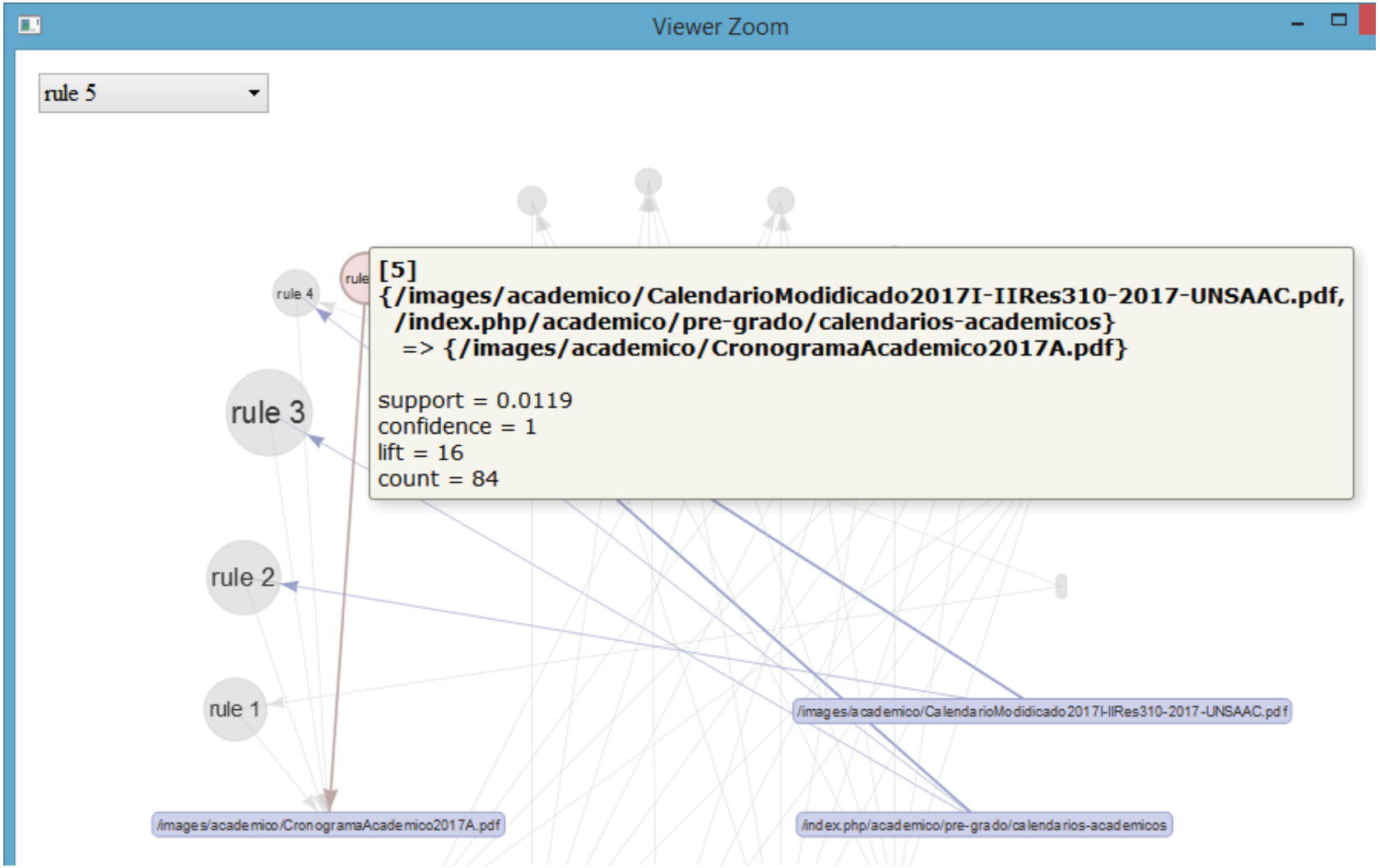

Figura 36.- Regla 5

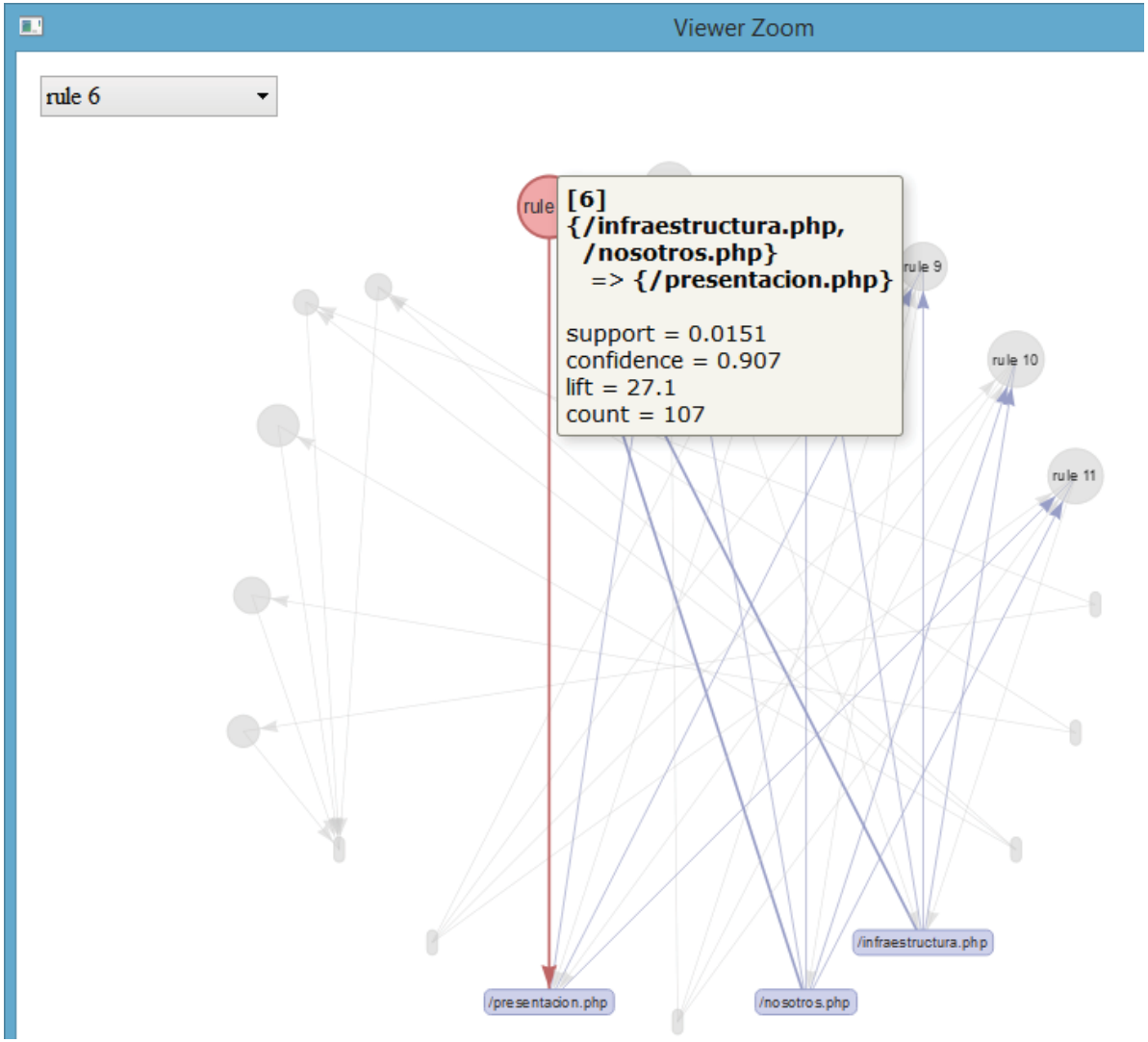

Figura 37.- Regla 6

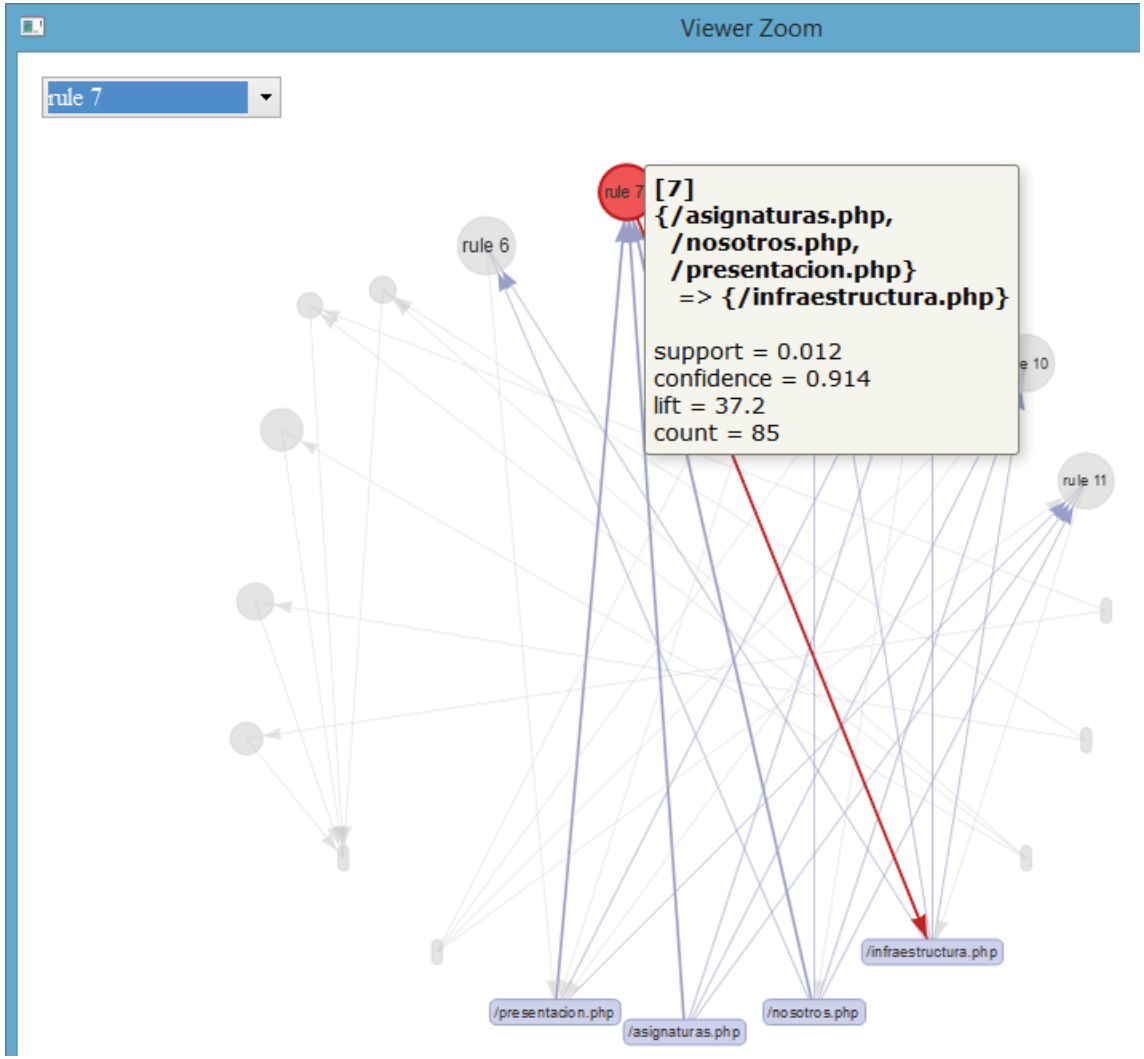

Figura 38.- Regla 7

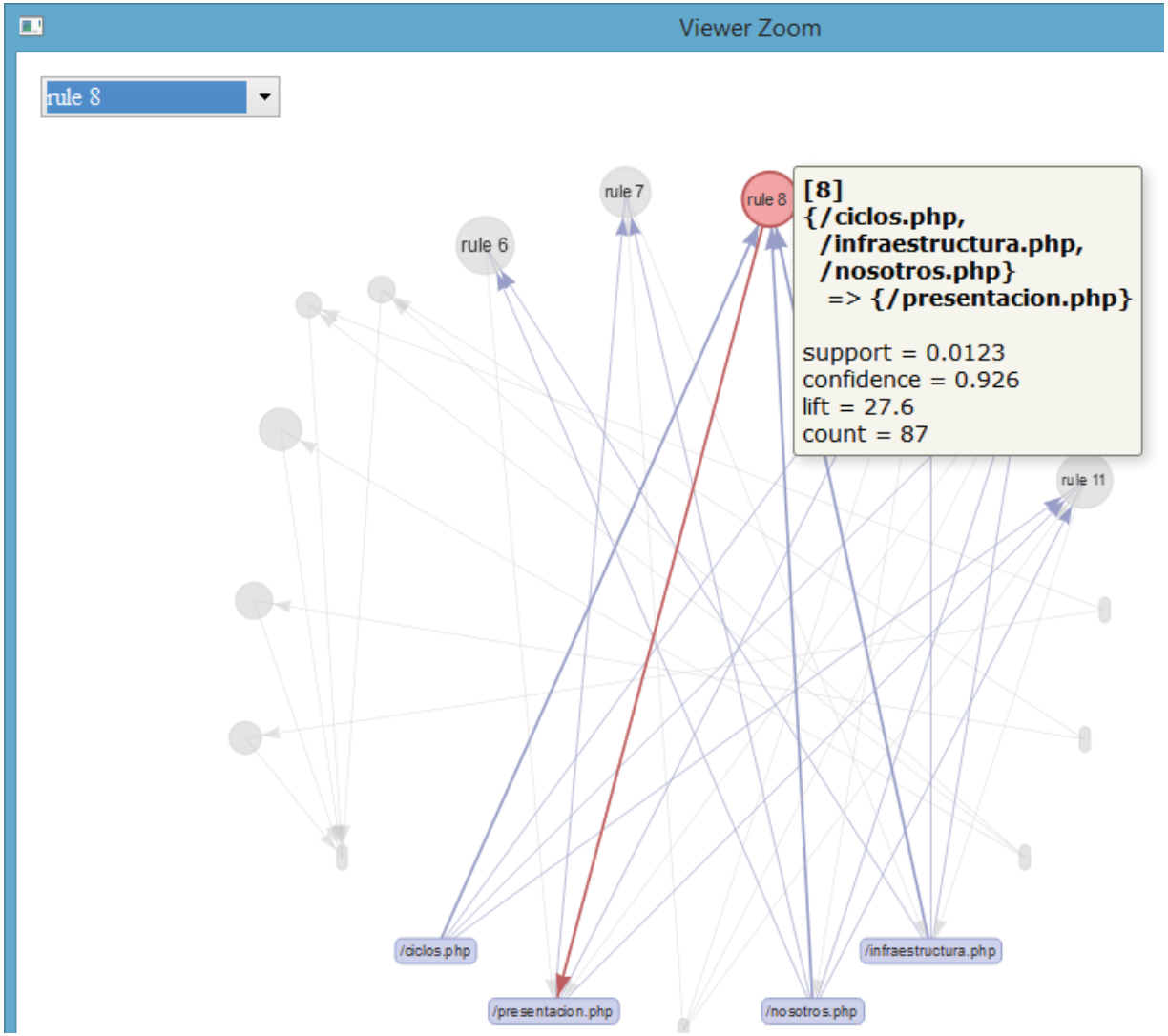

Figura 39.- Regla 8

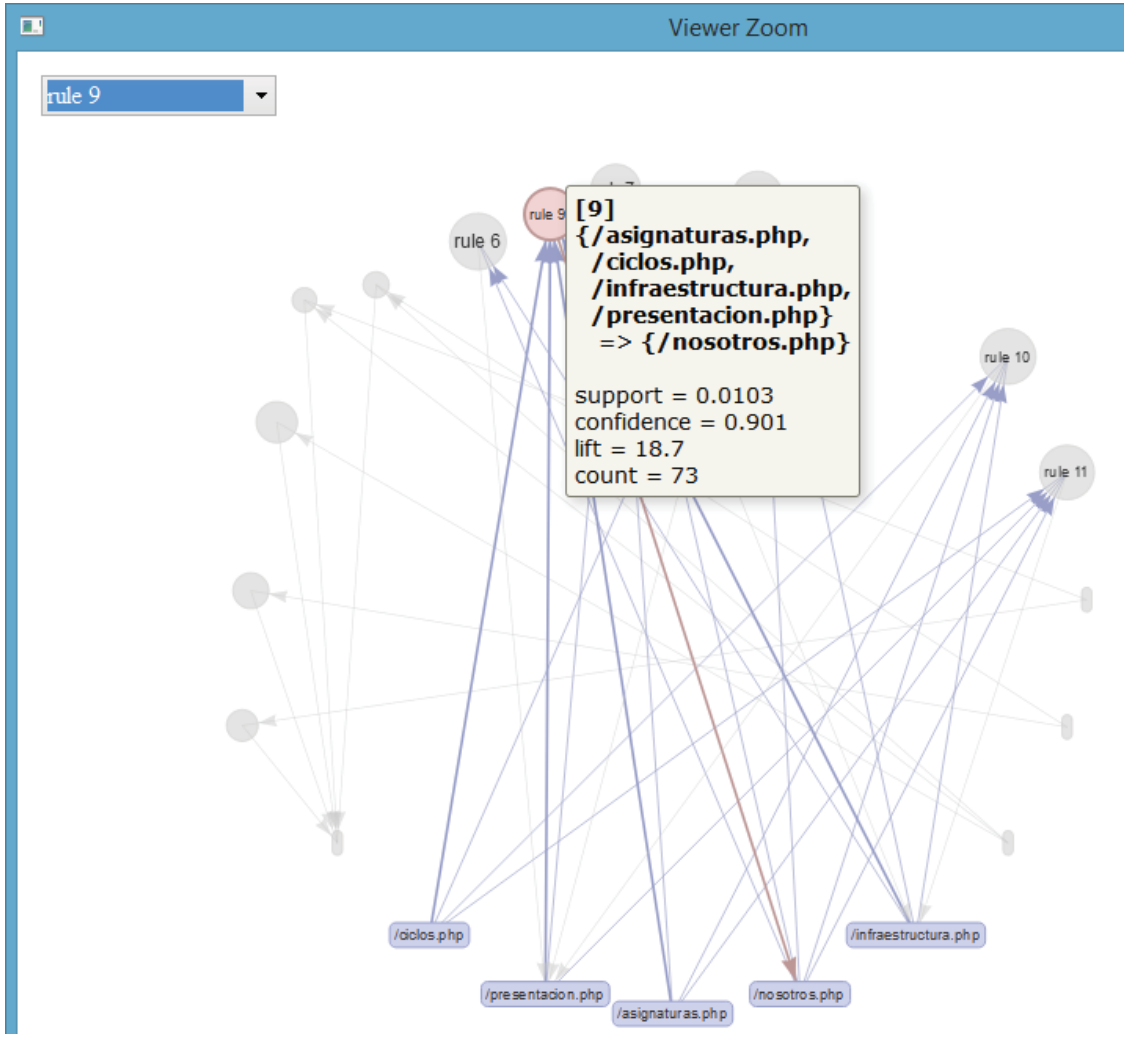

Figura 40.- Regla 9

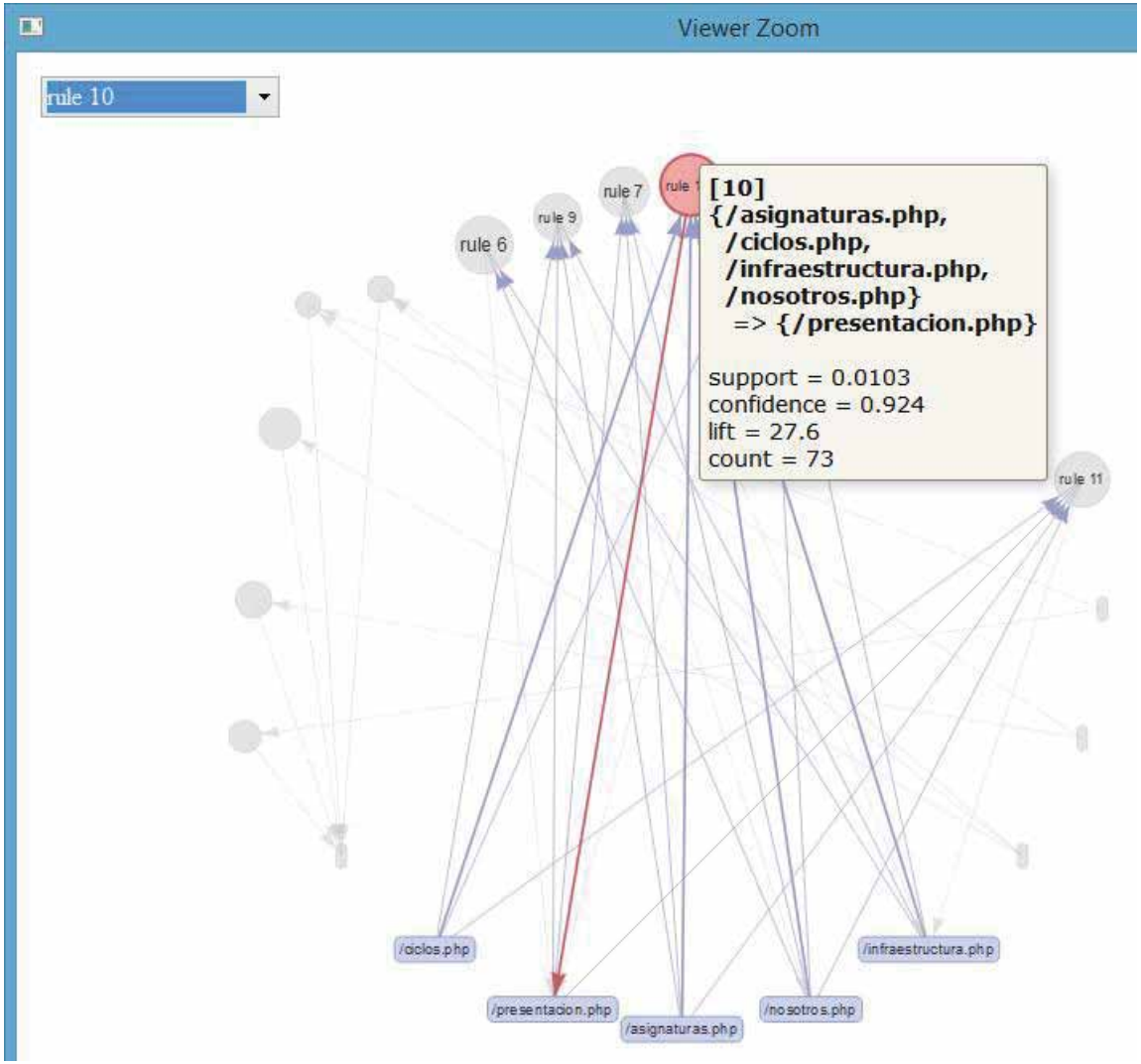

Figura 41.- Regla 10

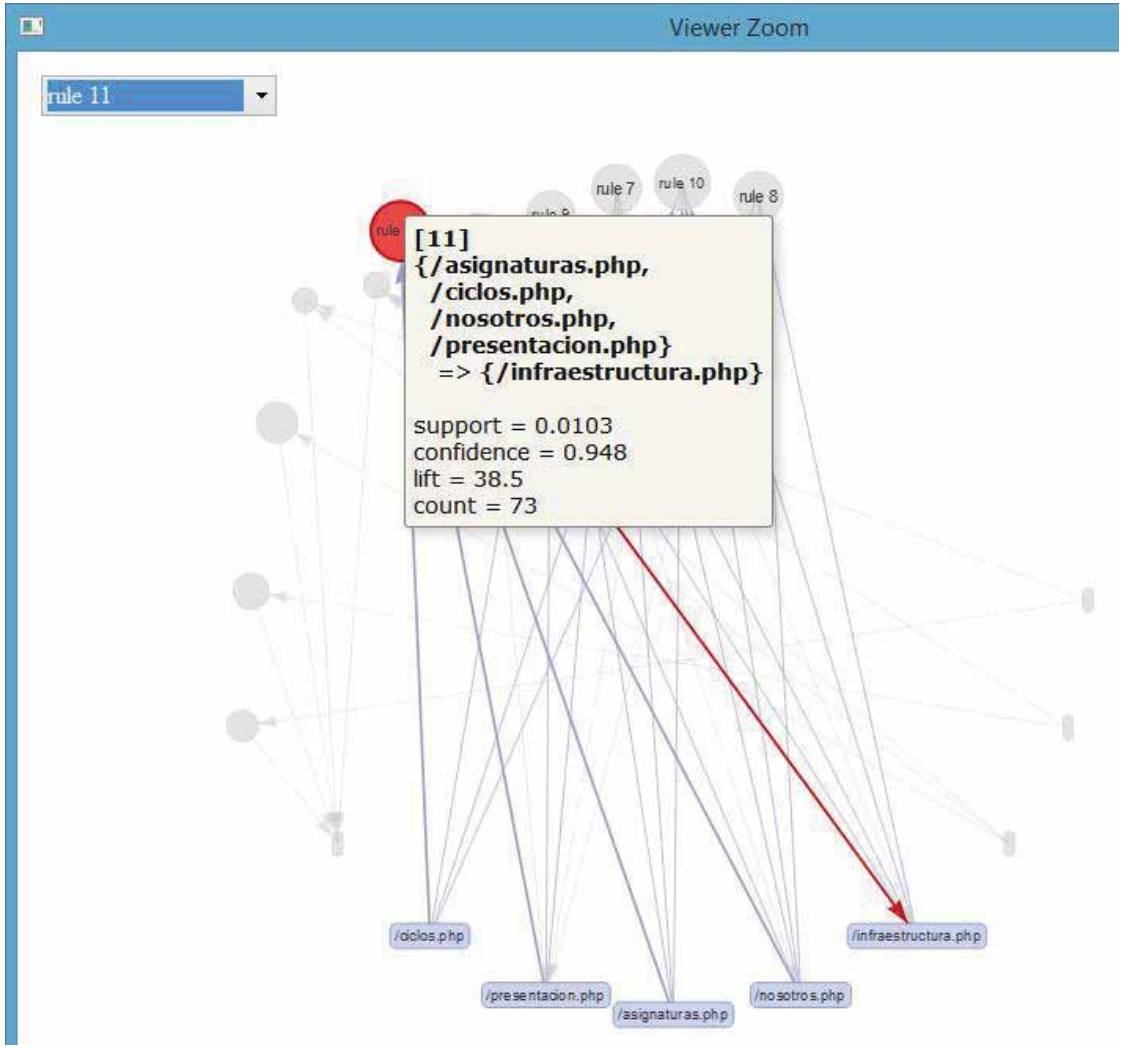

Figura 42.- Regla 11

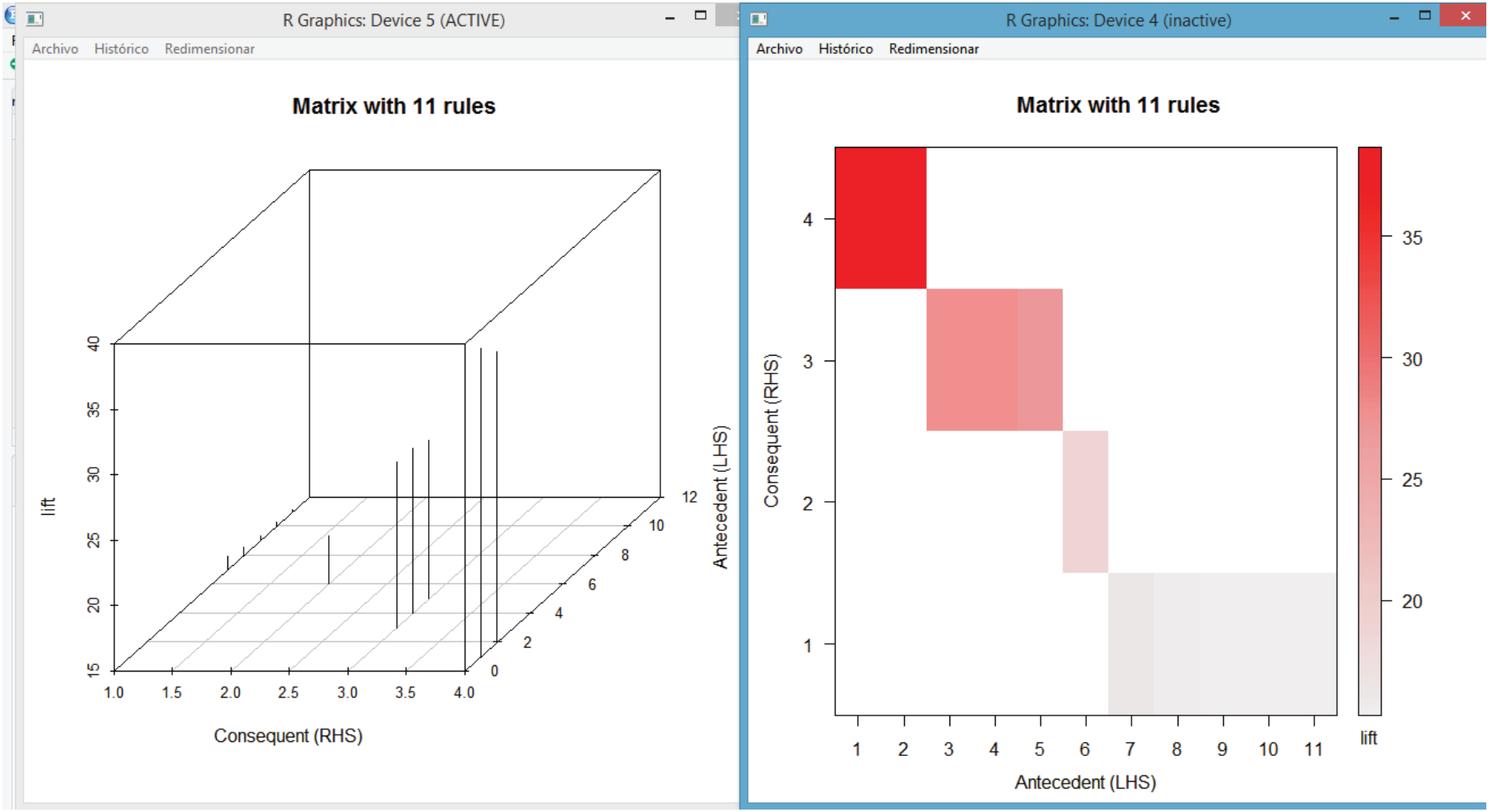

Figura 43.- Graficas en 2d y 3d de las 11 reglas.

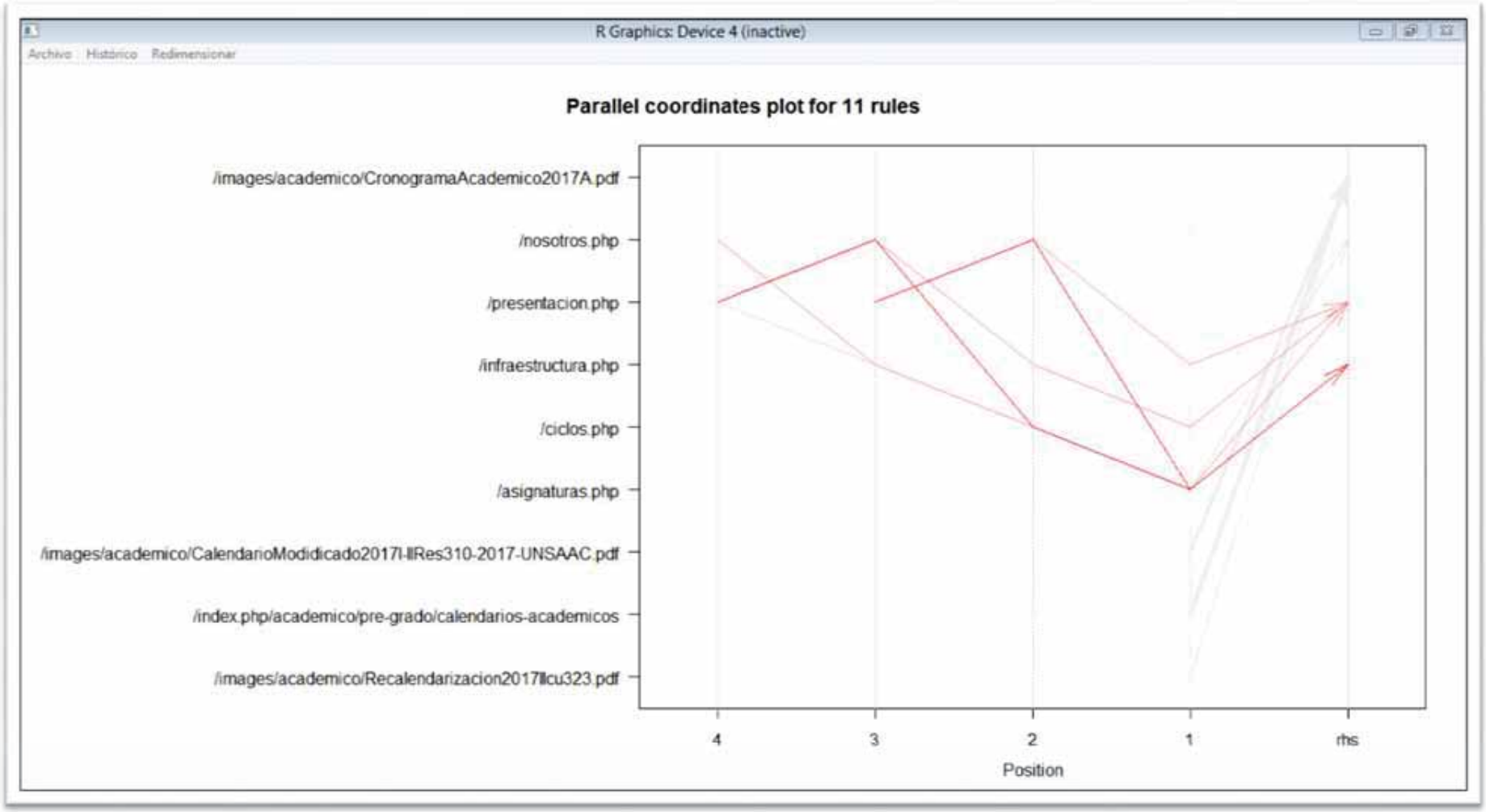

Figura 44.- Coordenadas paralelas para las reglas de asociacion.

Aquí se aprecia las preferencias que tienen los usuarios de las páginas web y claramente, esta representación nos muestra las reglas y los conjuntos de páginas como una secuencia.

Las coordenadas paralelas muestran los recursos en función a las reglas que están a la mano izquierda. Lo que evidencia básicamente es la comparación de las páginas y la relación que hay entre estas, y de esta forma se puede apreciar las preferencias y que estas siguen ciertos patrones o relaciones entre ellos. En este grafico se puede apreciar claramente cómo se compara varios recursos y como se relacionan entre sí, donde cada línea es una colección de puntos colocados en cada eje, que han sido conectados entre sí.

También se realizó un cuadro comparativo ejecutando el mismo procedimiento a los logs separados en grupos que especificamos a continuación.

- Cantidad de logs a analizar en los primeros 6 meses del año, en los últimos 6 meses del año y durante el año 2017.

	PRIMER SEMESTRE 2017	**SEGUNDO SEMESTRE 2017**	**ANUAL 2017**
NUMERO DE LOGS	8267	24727	32994
RANGO DE MESES DEL AÑO 2017	ENERO, FEBRERO, JUNIO	DE JULIO A DICIEMBRE	DE ENERO A DICIEMBRE, EXCEPTO MARZO, ABRIL Y MAYO

Tabla 11.- preferencias correspondiente al primer semestre

- Numero de reglas de asociación generadas por las preferencias de los usuarios en los primeros 6 meses del año, en los últimos 6 meses del año y durante el año 2017 para las que cumplan una confianza mayor o igual al 90%.

	PRIMER SEMESTRE 2017	**SEGUNDO SEMESTRE 2017**	**ANUAL 2017**
NUMERO DE REGLAS CON UN 90% DE CONFIANZA	3	10	11
RANGO DE MESES DEL AÑO 2017	ENERO, FEBRERO, JUNIO	DE JULIO A DICIEMBRE	DE ENERO A DICIEMBRE, EXCEPTO MARZO, ABRIL Y MAYO

Tabla 12 .- numero de reglas de asociacion por semestre y año

- Número de usuarios únicos, recursos únicos en los primeros 6 meses del año, últimos 6 meses del año y durante el año 2017.

	PRIMER SEMESTRE	SEGUNDO SEMESTRE	ANUAL
USUARIOS UNICOS	1796	5579	7074
RECURSOS UNICOS	958	1512	2005

Tabla 13.- usuarios unicos por semestre y año

El número de reglas tuvo variaciones para cada muestra semestral y anual, también se observa que, algunas de las reglas del primer semestre están contenidas en el segundo semestre y las del segundo en el anual, además se valida el método de minería de uso web utilizado, el cual genera resultados sobre las preferencias, semestrales y anual, dejando en claro que algunas preferencias se pueden mantener pero hay otras que aparecen y esto tiene sentido, pues el contenido que se muestra a nivel de la página web en algunos casos tiende a variar, al ser natural el hecho de que se añadan ciertas páginas o de que algunas ya no sean relevantes para los usuarios o se retiren y como las preferencias tienden a cambiar en un periodo corto, mediano o largo, en el tiempo, por lo que el proceso de obtención de las preferencias de navegación es una tarea constante.

	ENE	FEB	JUN	JUL	AGO	SET	OCT	NOV	DIC
NRO. LOGS	2625	1662	3980	4321	5061	3766	4574	4412	2593
NRO. USUARIOS	683	249	906	1109	1360	998	1106	1257	750
NRO. RECURSOS	530	269	516	207	476	504	564	577	378
REGLAS >= AL 90% DE CONFIANZA	4	14	12	7	118	38	46	16	126
REGLAS >= AL 80% DE CONFIANZA	6	18	14	12	350	51	73	26	237
REGLAS >= AL 70% DE CONFIANZA	13	27	21	23	556	64	111	38	275

Tabla 14.- logs, usuarios unicos, recursos unicos y reglas, por meses del 2017

Como se aprecia en la tabla se observa el hecho de que tanto el número de logs, recursos y usuarios varía en cada uno de los meses analizados, donde se muestra diferencias significativas, observándose también que el hecho de que haya más logs no significa que se tenga más recursos, pero sí que haya, más usuarios, por otro lado, también en cuanto a la confianza de las reglas, se obtuvieron reglas de asociación con certezas del 90%, 80% y 70%, en las cuales se observa variaciones muy grandes, desde la menor confianza a la mayor, esto es un indicativo de que podríamos mejorar el nivel de confianza mucho más aun, acercándonos al 99% de certeza puesto que observando las reglas generadas en algunos meses se vieron reglas con una confianza del 100% y no eran pocas sino más bien una cantidad considerable, la razón por la cual deberíamos evitar demasiadas reglas es por el mismo hecho de que al tener muchas preferencias (por ejemplo 50), un usuario normal no se daría la molestia de revisar esas 50 recomendaciones por lo que es más recomendable tener pocas reglas con un grado de confianza entre el 90% y 100% ya que de este modo se está garantizando la recomendación utilizando sus preferencias.

4.4 DISCUSIONES

(Adamov, 2014) en su trabajo DATA MINING AND ANALYSIS IN DEPTH CASE STUDY OF QAFQAZ UNIVERSITY HTTP SERVER LOG ANALYSIS, indica el formato del log con el que trabaja, hace uso del comando egrep de Linux conjuntamente con las expresiones regulares para realizar la estructuración y limpieza, además manifiesta que usaron 42 millones de logs y que pesaban 4.2 GB, aunque no menciona si es uno solo o varios sin detallar el procedimiento. Por otro lado, en el presente proyecto se tuvo una población de más de 90 millones cuyo peso es de más 24 GB no se pudo realizar la estructuración mediante el comando Linux porque el formato del log es diferente, por lo que se tuvo que construir un programa exclusivo para este formato y ambos trabajos se utilizaron R.

(Sharma & Yadav, 2015) en su trabajo A REVIEW STUDY OF SERVER LOG FORMATS FOR EFFICIENT WEB MINING, Describe los diferentes formatos disponibles de log realizando una comparativa de estudio entre los distintos tipos de formatos de logs. En el presente proyecto fue fundamental para la estructuración pues el formato que se encontró fue el Apache (Common Log Format) en su forma extendida y puso las directrices del programa que estructuro los log de la UNSAAC.

(Malviya & Agrawal, 2015) en su trabajo A STUDY ON WEB USAGE MINING: THEORY AND APPLICATIONS, lo que resalta de este trabajo es el hecho de que, en parte señala las dificultades que hay en el descubrimiento de información en grandes cantidades de datos. Dentro de las aplicaciones de la minería de uso web, resaltan: el descubrimiento del tráfico web, para generar nuevas políticas concernientes al servidor web, que coadyuvaran a la satisfacción del usuario, otra aplicación es la restructuración de los sitios web en cuanto a composición de contenidos y la distribución de los enlaces en la página, además también menciona dos aplicaciones más como son la personalización de sitios web y el soporte al diseño de los sitios web. En el presente proyecto se valida la dificultad que hay al descubrir información en grandes cantidades de datos y el hecho de

que a partir de este descubrimiento se creen políticas de configuración en el servidor además de una restructuración del sitio web.

(Sukumar, Robert, & Yuvaraj, 2016) en su trabajo REVIEW ON MODERN DATA PREPROCESSING TECHNIQUES IN WEB USAGE MINING (WUM), presento un trabajo experimental con resultados de los log de servidores web del Colegio de Artes Gubernamentales de Coimbatore, realizando experimentos en procesamiento de datos, heurísticas y técnicas aplicadas a limpiar los log en crudo, luego se realiza la identificación de usuarios, Finalmente, los resultados del pre procesamiento del archivo de las sesiones de usuario, son utilizados para identificar usuarios únicos, número total de accesos, visitas, promedio por día, accesos fallados y respuestas exitosas. En cuanto a los resultados indica que trabajo con 22613 logs, que corresponden aproximadamente a 15 días del funcionamiento del servidor y detalla lo ocurrido día por día del día 1 al 12, mediante tablas indicando lo encontrado. En el presente proyecto, se trabajaron con más de 3 millones de logs correspondientes al año 2017, también se emplearon heurísticas para limpiar los logs en crudo no se pudo identificar las sesiones de usuario debido a que el formato del logs solo contenían ips mas no usuarios.

(Sharma, Bohra, & Yadav, 2016) en su trabajo COMPARATIVE ANALYSIS OF WEB-MINING APPROACHES FOR EFFICIENT MINING OF SERVER LOG FORMATS, señala haber utilizado dos técnicas, el Apriori y FP Growth basados en el tiempo de ejecución y la generación de patrones.

Donde se realizó la construcción de un programa en Java en la cual implementa los dos algoritmos mencionados anteriormente, dicha herramienta realiza, la limpieza de datos, clustering y filtrado de datos. Aunque no indica con que tamaño de logs ha utilizado, ni la cantidad procesada, muestra un cuadro comparativo donde muestra que el algoritmo Apriori es mucho mejor que FP Growth en tiempo de ejecución y que ocurre lo contrario cuando se quiere generar mejores patrones, en este caso FP Growth es mejor. En el presente trabajo se limitó a utilizar el algoritmo Apriori únicamente en vista de que interesaban más el resultado de las preferencias de navegación y no tanto el tiempo en el que se ejecutaban, se obtuvieron patrones pero no realizamos una comparativa como en el antecedente descrito.

(Neelima & Rodda, 2016) en su trabajo PREDICTING USER BEHAVIOR THROUGH SESSIONS USING THE WEB LOG MINING, se centra en las áreas del pre procesamiento, limpieza de datos, identificación de usuarios e identificación de sesiones para luego aplicar técnicas como clustering, asociación y clasificación, la finalidad era que pueda tomar en consideración, para la configuración del ancho de banda y la capacidad del servidor da la organización. Donde se hace énfasis en la descripción, formatos y tipos de archivos logs, además describe una metodología que consiste en la limpieza identificación de usuarios e identificación de sesiones, realiza la implementación de sus propios algoritmos para realizar la metodología que propone y en sus resultados experimentales, la

limpieza y estructuración son almacenadas en una bases de datos MySql Server, donde muestra el resultado obtenido en graficas tipo circular, apreciándose únicamente la identificación de usuarios y sesiones, aparentemente generadas en Excel , la cantidad de los logs tratados es de 1546 y no indica a que entidad le pertenece.

En el presente proyecto guarda similitudes en cuanto al almacenamiento de los logs filtrados también se utilizó Mysql luego de realizar la limpieza y estructuración, la cantidad de logs utilizadas sigue siendo superior en este trabajo y la institución está identificada en este caso.

(FLORES LAFOSSE, 2016) en su trabajo EXTRACCIÓN DE PATRONES SEMÁNTICAMENTE DISTINTOS A PARTIR DE LOS DATOS ALMACENADOS EN LA PLATAFORMA PAIDEIDA donde, trabaja exclusivamente con logs de la plataforma Moodle, la cual tiene una tabla donde almacena en una base de datos todos los logs, y como un usuario tiene que loguearse, por ende la identificación de estos se almacenan allí, utilizo también Weka y otro software denominado SPMF, que es una implementación realizada en Java el cual utiliza para mostrar el conjunto de ítems. En el presente proyecto no se utilizó los software mencionados en vista que no se adecuaron al formato con el cual trabajan estos programas razón por la cual se realizó la implementación en R, obteniendo graficas de grafos similares a las representaciones de uno de esos software.

CONCLUSIONES

- ✓ Se logró construir una estructura de datos que incluye las acciones realizadas por los usuarios a partir de los archivos log, para su análisis respectivo, considerando que para alcanzar dicho objetivo, se utilizó el proceso de minería de uso web el cual consistió en, realizar la limpieza de datos, integración y transformación de datos, generación de las transacciones a partir de la base de datos, luego la minería de reglas de asociación y finalmente el análisis de preferencias de navegación, todo esto se logró gracias al programa construido para este propósito en las primeras fases del proceso y seguidamente para obtener las preferencias de navegación, se realizó utilizando para ello R.

- ✓ Se identificó las preferencias de navegación, donde se evidencia que hay relación entre los usuarios que acceden a las páginas web de la UNSAAC, para ello se crearon reglas de asociación las cuales responden a las actividades que realizan los usuarios, además se evidencio cual es la secuencia que siguen al navegar por las paginas, se sabe también cual es la siguiente página que verán, a partir de una página inicial determinada, esto con una certeza mayor al 90%, cabe resaltar el hecho de que dichas preferencias responden a interacciones reales entre las páginas y no se deben al azar, para verificar la validez y certeza de las preferencias, mediante las reglas de asociación se realizaron pruebas con métricas como, lift, coverage, fishersExactTest, las cuales como se detallaron, arrojaron indicadores válidos.

- ✓ Quedo demostrado que en archivos cercanos al big data, como es el de los log analizados para este caso en particular, existen preferencias ocultas y valiosas para la institución, estos log son archivos que requirieron de un programa especial para poder visualizarlos previamente y posteriormente procesarlos, pues los programas tradicionales no pudieron realizar dicha tarea, estos archivos tienen la característica de ser semi estructurados y en algunos casos no estructurados, los cuales ocultan información valiosa, como las preferencias de navegación de los usuarios de las páginas web de la UNSAAC, que se encuentran en estos log, pero por inspección simple era imposible ver esto.

RECOMENDACIONES Y TRABAJOS FUTUROS

- ✓ Antes de realizar el proceso de la minería datos, se recomienda verificar si la página web está bien estructurada o si al menos cuenta con una estructura propuesta o un mapa del sitio, esto facilitara la forma como se realice el análisis para poder obtener las preferencias de navegación. Y se recomienda cambiar la configuración de los tamaños de archivos log en el servidor web, para que sean manejables, puedan ser vistos usando un editor de texto normal, como el notepad, etc. ya que los análisis del contenido de los log, se dieron en más de seis intentos debido a que el tamaño de los archivos (mayores a 1 GB) es directamente proporcional a la dificultad de procesarlos y limpiarlos.

- ✓ En el caso de que se quiera trabajar con archivos cercanos al big data, como en este caso y se deseen visualizar, una buena opción que encontré fue EmEditor, que es capaz de abrir archivos de más de una giga de peso, aunque es software con licencia te permite probarlo por 30 días y fue de mucha utilidad al momento de crear el programa que realizo la tarea de limpieza de los log.

- ✓ Como trabajos futuros, se podría profundizar sobre las matrices de transacciones, ya que ellas solo muestran las páginas que visitaron los usuarios en varias sesiones, podríamos realizar un estudio sobre el contenido que le interesa a un determinado usuario, mediante ciertas palabras contenidas en la página web, ya que podríamos determinar la causa por la cual un determinado usuario permanece más tiempo en una página que en otra y de esta forma determinar que le interesa realmente o cuales son los contenidos más relevantes o que los atraen, de un sitio web.

- ✓ Otra alternativa como futuros trabajos de investigación seria la implementación de un sistema de recomendación o un sistema de recomendación basado en contenidos, que este centrado en la predicción y clasificación sobre la matriz de transacciones de los usuarios.

- ✓ Una sugerencia sobre lo visto en la presente investigación es que las páginas web deberían de tiparse, manejar etiquetas pues esto podría mejorar de manera tremenda el análisis, inclusive se podría aplicar machine learning o clustering.

- ✓ Para que se agilice el proceso de minería de uso web y obtener los resultados, se podría implementar un módulo en la página web que registre, el nombre de usuario o la dirección ip, la página que solicita un usuario, la fecha y hora de acceso, etc. (que no almacene archivos con extensión css, javascript, jpg, etc. pues no aportan nada al proceso de minería de uso web) y registrarlo directamente en una base de datos, para de esta forma evitar el proceso tedioso que significa realizar el pre procesamiento y generar las reglas mucho más rápido, pues como se observó de la muestra de 3161826 logs, luego de realizar la limpieza y la estructuración quedaron solo 32994 logs, que vendrían a ser aproximadamente el 1.04 % de dicha muestra lo que indica que el 98.96% de los log son innecesarios.

BIBLIOGRAFIA

Adamov, A. (2014). Data mining and analysis in depth. case study of Qafqaz University HTTP server log analysis. doi:10.1109/ICAICT.2014.7035947

Anitha, & Isakki. (7-9 de Enero de 2016). A survey on predicting user behavior based on web server log files in a web usage mining. Kovilpatti, India. doi:10.1109/ICCTIDE.2016.7725340

Castaño Diaz, V. J. (12 de 2008). Caracterizacion de servidores web en el ambito academico.

Castells, M. (2000). *The Information Age: Economy, Society and Culture. Volume I.* Madrid: Alianza Editorial, S. A.

EasyPHP. (20 de 12 de 2018). Obtenido de https://www.easyphp.org/easyphp-webserver.php

Egevang K. (11 de 09 de 2018). *The IP Network Address Translator (NAT)*. Obtenido de https://tools.ietf.org/html/rfc1631

Emurasoft Inc. (19 de 01 de 2018). *EmEditor*. Obtenido de https://www.emeditor.com/

FLORES LAFOSSE, N. (2016). EXTRACCION DE PATRONES SEMANTICAMENTE DISTINTOS A PARTIR DE LOS DATOS ALMACENADOS EN LA PLATAFORMA PAIDEIA. LIMA, PERU: PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ ESCUELA DE POST GRADO.

Ha, A. (05 de 04 de 2019). *Lotame pitches an 'unstacked' approach to selling data tools*. Obtenido de https://techcrunch.com/2019/04/05/lotame-unstacked/

Han, J., & Kamber, M. (2006). *Data mining: Concepts and Techniques.* (2nd ed.) Morgan Kaufman.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques.* USA: (3rd ed)ELSEVIER INC.

Hernández Orallo, J., Ramiréz Quintana, M. J., & Ferri Ramiréz, C. (2004). *Introducción a la Minería de Datos.* Madrid(España): PEARSON EDUCACION, S.A.

Hernandez Sampieri, R. (2014). *Metodologia de la investigacion 6ta edicion.* Mexico: McGraw-Hill.

Jha, P., & Jaiswal, A. (14-15 de Enero de 2016). Creating ontology for intelligent web by using web usage mining. Noida, India . doi:10.1109/CONFLUENCE.2016.7508133

Jiang, F., K. Leung, C., & Pazdor, A. G. (13-16 de Octubre de 2016). Web Page Recommendation Based on Bitwise Frequent Pattern Mining. Omaha, NE, USA. doi:10.1109/WI.2016.0111

Kabir, J. M. (1999). *La biblia del Sevidor Apache.* Madrid(España): ANAYA MULTIMEDIA.

Karna, N., Supriana, I., & Maulidevi, N. (2015). Social CRM using web mining for Indonesian academic institution. Bandung, Indonesia. doi:10.1109/ICITSI.2015.7437721

Khatri, R., & Gupta, D. (2015). An efficient periodic web content recommendation based on web usage mining. Kolkata, India . doi: 10.1109/ReTIS.2015.7232866

Leskovec, J., Rajaraman, A., & J. U. (2014). *Mining of Massive Datasets.* Cambridge University Press. Obtenido de http://www.mmds.org/

Lin, S., & WenZheng, X. (2015). E-Commerce Personalized Recommendation System Based on Web Mining Technology Design and Implementation. Halong Bay, Vietnam . doi: 10.1109/ICITBS.2015.93

Li-rui, W., & Zhi-fu, W. (2015). Implementation on E-Commerce Network Marketing Based on WEB Mining Technology. Nanchang, China . doi: 10.1109/ICMTMA.2015.141

Liu, B. (2012). *WEB DATA MINING Exploring Hyperlinks, Contents and Usage Data 2nd ed.* Berlin: Springer.

LIU, J., XIAO, L., & SHAO, X. (21-23 de Julio de 2016). Context-Dependency Relation Extraction Based on Web Mining. Harbin, China . doi:10.1109/IMCCC.2016.173

Llanos Ferrais, D. R. (2007). *FUNDAMENTOS DE INFORMATICA Y PROGRAMACION EN C.*
Malviya, B. K., & Agrawal, J. (2015). A Study on Web Usage Mining Theory and Applications. Gwalior, India . doi:10.1109/CSNT.2015.247
Mora, S. L. (2002). *Programacion de aplicaciones web: historia, principios basicos y clientes web.* España: Editorial Club Universitario.
Neelima, G., & Rodda, S. (3-5 de Marzo de 2016). Predicting user behavior through sessions using the web log mining. Doddaballapur, India . doi:10.1109/HMI.2016.7449167
Network Working Group. (15 de 12 de 2018). Obtenido de https://tools.ietf.org/html/rfc3986
Olaya, V. (2014). *Sistemas de informacion geaografica.* Obtenido de http://volaya.github.io/libro-sig/
Pinho Lucas, J. (10 de 2010). MÉTODOS DECLASIFICACIÓN BASADOSEN ASOCIACIÓN APLICADOS A SISTEMAS DE RECOMENDACIÓN. Salamanca.
Plaza Martin, V. (08 de 09 de 2015). Análisis de ficheros log de la WiFi-ULL usando técnicas de Big Data.
RCU. (19 de 07 de 2018). *Red de Comunicaciones UNSAAC*. Obtenido de http://rcu.UNSAAC.edu.pe/actividades.php?p=1
Ren, W., & Yan, J. (2015). An Improved CMAC Neural Network Model for Web Mining. Hangzhou, China . doi: 10.1109/ISCID.2015.61
Robots.txt. (23 de 05 de 2018). *The Web Robots Page*. Obtenido de https://www.robotstxt.org/
RStudio. (13 de 02 de 2018). *Open source and enterprise-ready professional software for data science - RStudio*. Obtenido de https://www.rstudio.com/
Ryte. (12 de 12 de 2018). *Ryte Wiki*. Obtenido de https://es.ryte.com/wiki/Hyperlink
Sangeetha, & Suresh, J. (2014). Page ranking algorithms used in Web Mining. Chennai, India . doi:10.1109/ICICES.2014.7033794
Sharma, P., & Yadav, S. (2015). A review study of server log formats for efficient web mining. Noida, India. doi:10.1109/ICGCIoT.2015.7380681
Sharma, P., Bohra, B., & Yadav, S. (7-9 de Setiembre de 2016). Comparative Analysis of Web-Mining Approaches for Efficient Mining of Server Log Formats . Noida, India. doi:10.1109/ICRITO.2016.7784949
Sukumar, P., Robert, L., & Yuvaraj, S. (6-8 de Octubre de 2016). Review on modern Data Preprocessing techniques in Web usage mining (WUM). Bangalore, India . doi:10.1109/CSITSS.2016.7779441
The Apache Software Foundation. (08 de 03 de 2018). *Log Files Apache HTTP Server*. Obtenido de https://httpd.apache.org/docs/1.3/logs.html
UNMSM. (2005). Revista de investigacion de sistemas de informacion. 7-13.
VILLALOBOS LUENGO, C. A. (2016). ANÁLISIS DE ARCHIVOS LOGS SEMI-ESTRUCTURADOS DE AMBIENTES WEB USANDO TECNOLOGÍAS BIG-DATA . CHILE.
W3C. (20 de 10 de 2018). Obtenido de https://www.w3.org/html/
W3C. (25 de 11 de 2018). *W3C Working Draft WD-logfile-960323*. Obtenido de https://www.w3.org/TR/WD-logfile
W3C. (19 de 10 de 2018). *WEB DESIGN AND APPLICATIONS*. Obtenido de https://www.w3.org/standards/webdesign/
Wayback Machine. (21 de 12 de 2018). Obtenido de https://web.archive.org/web/20190606202345/https://tools.ietf.org/html/rfc3986
Whalen, D. (15 de 10 de 2018). *The Unofficial Cookie FAQ*. Obtenido de http://www.cookiecentral.com/faq/#1.1

ANEXOS

```php
<?php
//set_time_limit(0);
require_once './ConexionBD.php';

function
insertar_log_beta($ip,$fecha,$gmt,$metodo,$recurso,$protocolo,$codigo,$mb,$host,$navegador
,$so,$archivo){
    try {

        $query="call
        spu_insertar_log_beta('$ip','$fecha','$gmt','$metodo','$recurso','$protocolo','$co
        digo','$mb','$host','$navegador','$so','$archivo')";
        $db = new ConexionBD();
        $db->ejecutarQuery($query);
        //echo "insertado $d, $h <br>";
    }catch (Exception $e) {
        throw $e;
    }
}

function insertar_log_file($archivo,$cant){
    try {
        $query="call spu_insertar_log_file('$archivo',$cant)";
        $db = new ConexionBD();
        $db->ejecutarQuery($query);
        //echo "insertado $d, $h <br>";
    }catch (Exception $e) {
        throw $e;
    }
}

function limpiar_log_file($archivo){
    try {
        $query="call spu_limpiar_log_beta('$archivo')";
        $db = new ConexionBD();
        $db->ejecutarQuery($query);
        //echo "insertado $d, $h <br>";
    }catch (Exception $e) {
        throw $e;
    }
}

//Determinamos si el archivo se subio o no a memoria
if (is_uploaded_file($_FILES['archivo']['tmp_name'])) {
    $nombreDirectorio = $_SERVER['DOCUMENT_ROOT'] . "/TesisWM/logs/";
    $nombreFichero = $_FILES['archivo']['name'];

    $nombreCompleto = $nombreDirectorio . $nombreFichero;
    //is_file – Indica si el nombre de fichero es un fichero normal
```

-1-

Figura 45.- Script que realiza la limpieza del log y lo estructura parcialmenteen una base de datos (parte 1)

```
    date_default_timezone_set("America/Bogota");
    echo "La hora de inicio es " . date("h:i:sa").'<br>';

    if (is_file($nombreCompleto)) {
//        $idUnico = time();
//        $nombreFichero = $idUnico . "-" . $nombreFichero;
        echo 'El archivo ya existe!';
    } else {
        move_uploaded_file($_FILES['archivo']['tmp_name'], $nombreDirectorio .
        $nombreFichero);
        print ("El fichero fue subido correctamente!!! <br>");
        // ------------------------------------------------------------------------
        $log = fopen($nombreDirectorio . $nombreFichero, "r");
        $i=0;
        $archivo= $nombreFichero;
        //Arreglos menores de 15 partes no las lee
        //Arreglos entre 23 y 24 elementos
        while (!feof($log)) {
            $linea = fgets($log);
            $arreglo = explode(" ", $linea);
            // echo "Linea $i : ".$linea ." CANT.".count($arreglo). "<br />";

            if(count($arreglo)> 16 && $arreglo[0] != "127.0.0.1" && $arreglo[10] !=
            '"https://www.google.com/"' && $arreglo[6] != '"-"' && $arreglo[10] != '"-"'){
                //
                $ip= $arreglo[0];
                $fecha= $arreglo[3];
                $gmt= $arreglo[4];
                $metodo= $arreglo[5];
                $recurso= $arreglo[6];
                $protocolo= $arreglo[7];
                $codigo= $arreglo[8];
                $mb= $arreglo[9];
                $host= $arreglo[10];
                $navegador= $arreglo[11];
                $so= $arreglo[12]." ".$arreglo[13]." ".$arreglo[14]." ".$arreglo[15]."
                ".$arreglo[16];


                insertar_log_beta($ip,$fecha,$gmt,$metodo,$recurso,$protocolo,$codigo,$mb,
                $host,$navegador,$so,$archivo);
//                print_r($arreglo);

            }


            $i++;
        }
        //
        echo "La hora de fin es " . date("h:i:sa").'<br>';
```

-2-

Figura 46.- Script que realiza la limpieza del log y lo estructura parcialmenteen una base de datos (parte 2)

```
        insertar_log_file($archivo,$i);
        //
        print ("Logs insertados Nro logs: $i <br>");
        fclose($log);
        print ("Archivo Log, cerrado!!! <br>");
        echo "La hora de ini limpieza es " . date("h:i:sa").'<br>';
        limpiar_log_file($archivo);
        echo "La hora de fin limpieza es " . date("h:i:sa").'<br>';
        print ("Archivo Log, depurado e insertado filtrado!!! <br>");
    }
} else {
    print ("No se ha podido subir el fichero <br>");
}

//echo '<HR>';
//echo '<p>*****************************************';
//echo 'CONTENIDO DE $_SERVER[PHP_SELF] :' . $_SERVER['PHP_SELF'] . '<br>';
//echo 'CONTENIDO DE $_SERVER[SERVER_NAME] :' . $_SERVER['SERVER_NAME'] . '<br>';
//echo 'CONTENIDO DE $_SERVER[DOCUMENT_ROOT] :' . $_SERVER['DOCUMENT_ROOT'] . '<br>';
```

-3-

Figura 47.- Script que realiza la limpieza del log y lo estructura parcialmenteen una base de datos (parte 3)

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `spu_limpiar_log_beta`(parchivo varchar(45))
BEGIN

-- declaramos variables
    declare cantidad int default 0;
    declare fecha_min datetime;
    declare fecha_max datetime;

    -- Limpiamos la data
    update log_beta -- tmp_id_logs_ok
    set fecha = fn_format_date(fecha), gmt = left(gmt,5),
        metodo = right(metodo,3), protocolo = left(protocolo,8),
        host=substring(host,2,length(host)-2),
        navegador=substring(navegador,2,length(navegador)-1);

    -- limpiamos los logs
    delete from bdwebmining.log_beta
    where recurso = '/' or host = '-' or host like '%google%' or host like '%Googlebot%';
    -- insertamos
    insert into log
    (idlog,ip,fecha,gmt,metodo,recurso,protocolo,estado,mb,host,navegador,so,archivo)
    select null,ip,fecha,gmt,metodo,recurso,protocolo,estado,mb,host,navegador,so,archivo
    from log_beta;
    -- actualizamos la tabla log_files
    select count(*) into cantidad
    from log
    where archivo=parchivo;

    select min(fecha) into fecha_min
    from log
    where archivo=parchivo;

    select max(fecha) into fecha_max
    from log
    where archivo=parchivo;

    -- actualizamos
    update log_files
        set nro_ok_logs=cantidad, begin_date=fecha_min, end_date=fecha_max
    where nombre=parchivo;

    -- truncate
    TRUNCATE TABLE log_beta;

    select 1 as 'Error', 'OK LOG INSERT' as 'Mensaje';


END
```

-1-

Figura 48.- procedimiento que realiza la limpieza definiva y la estructuracion final.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `spu_insertar_log_beta`(pip varchar(45),
pfecha varchar(45),
                                    pgmt varchar(10), pmetodo varchar(10), precurso
                                    varchar(500),
                                    pprotocolo varchar(10), pestado varchar(10),
                                    pmb varchar(10), phost varchar(300),
                                    pnavegador varchar(45), pso varchar(45),
                                    parchivo varchar(45))
BEGIN
    -- insertamos en tabla log_beta
    insert into log_beta

    values(null,pip,pfecha,pgmt,pmetodo,precurso,pprotocolo,pestado,pmb,phost,pnavegador,p
    so,parchivo);
    select 1 as 'Error', 'OK LOG_BETA' as 'Mensaje';
END
```

-1-

Figura 49.- procedimiento que inserta los log generados por el script

LOG 43.5 La hora de inicio es 11:41:06am El fichero fue subido correctamente!!! La hora de fin es 12:22:26pm Logs insertados Nro logs: 86493 Archivo Log, cerrado!!! La hora de ini limpieza es 12:22:26pm La hora de fin limpieza es 12:22:42pm Archivo Log, depurado e insertado filtrado!!!	LOG 42.5 La hora de inicio es 12:28:18pm El fichero fue subido correctamente!!! La hora de fin es 12:49:13pm Logs insertados Nro logs: 75395 Archivo Log, cerrado!!! La hora de ini limpieza es 12:49:13pm La hora de fin limpieza es 12:49:16pm Archivo Log, depurado e insertado filtrado!!!
LOG 41.5 La hora de inicio es 01:32:44pm El fichero fue subido correctamente!!! La hora de fin es 02:09:22pm Logs insertados Nro logs: 45523 Archivo Log, cerrado!!! La hora de ini limpieza es 02:09:22pm La hora de fin limpieza es 02:09:25pm Archivo Log, depurado e insertado filtrado!!!	LOG 40.5 La hora de inicio es 02:15:13pm El fichero fue subido correctamente!!! La hora de fin es 02:51:56pm Logs insertados Nro logs: 40532 Archivo Log, cerrado!!! La hora de ini limpieza es 02:51:56pm La hora de fin limpieza es 02:52:02pm Archivo Log, depurado e insertado filtrado!!!
LOG 39.5 La hora de inicio es 03:00:29pm El fichero fue subido correctamente!!! La hora de fin es 03:51:37pm Logs insertados Nro logs: 47468 Archivo Log, cerrado!!! La hora de ini limpieza es 03:51:38pm La hora de fin limpieza es 03:51:49pm Archivo Log, depurado e insertado filtrado!!!	LOG 38.5 La hora de inicio es 04:06:15pm El fichero fue subido correctamente!!! La hora de fin es 04:36:41pm Logs insertados Nro logs: 28600 Archivo Log, cerrado!!! La hora de ini limpieza es 04:36:41pm La hora de fin limpieza es 04:36:48pm Archivo Log, depurado e insertado filtrado!!!
LOG 33.5 La hora de inicio es 04:49:41pm El fichero fue subido correctamente!!! La hora de fin es 05:45:20pm Logs insertados Nro logs: 63236 Archivo Log, cerrado!!! La hora de ini limpieza es 05:45:20pm La hora de fin limpieza es 05:46:30pm Archivo Log, depurado e insertado filtrado!!!	LOG 32.5 La hora de inicio es 06:00:08pm El fichero fue subido correctamente!!! La hora de fin es 06:58:21pm Logs insertados Nro logs: 63760 Archivo Log, cerrado!!! La hora de ini limpieza es 06:58:21pm La hora de fin limpieza es 06:58:32pm Archivo Log, depurado e insertado filtrado!!!
LOG 31.5 La hora de inicio es 09:36:39am El fichero fue subido correctamente!!! La hora de fin es 10:42:10am Logs insertados Nro logs: 72172 Archivo Log, cerrado!!! La hora de ini limpieza es 10:42:10am La hora de fin limpieza es 10:42:28am	LOG 30.5 La hora de inicio es 11:18:29am El fichero fue subido correctamente!!! La hora de fin es 12:04:08pm Logs insertados Nro logs: 52861 Archivo Log, cerrado!!!

Archivo Log, depurado e insertado filtrado!!!	La hora de ini limpieza es 12:04:08pm La hora de fin limpieza es 12:04:17pm Archivo Log, depurado e insertado filtrado!!!
LOG 29.5 La hora de inicio es 02:42:45pm El fichero fue subido correctamente!!! La hora de fin es 03:52:38pm Logs insertados Nro logs: 75888 Archivo Log, cerrado!!! La hora de ini limpieza es 03:52:38pm La hora de fin limpieza es 03:52:55pm Archivo Log, depurado e insertado filtrado!!!	LOG 28.5 La hora de inicio es 03:58:03pm El fichero fue subido correctamente!!! La hora de fin es 04:56:04pm Logs insertados Nro logs: 59589 Archivo Log, cerrado!!! La hora de ini limpieza es 04:56:04pm La hora de fin limpieza es 04:56:21pm Archivo Log, depurado e insertado filtrado!!!
LOG 27.5 La hora de inicio es 05:42:32pm El fichero fue subido correctamente!!! La hora de fin es 06:23:25pm Logs insertados Nro logs: 48205 Archivo Log, cerrado!!! La hora de ini limpieza es 06:23:25pm La hora de fin limpieza es 06:23:42pm Archivo Log, depurado e insertado filtrado!!!	LOG 26.5 La hora de inicio es 08:16:03pm El fichero fue subido correctamente!!! La hora de fin es 08:56:56pm Logs insertados Nro logs: 46332 Archivo Log, cerrado!!! La hora de ini limpieza es 08:56:56pm La hora de fin limpieza es 08:57:12pm Archivo Log, depurado e insertado filtrado!!!
LOG 25.5 La hora de inicio es 09:25:30pm El fichero fue subido correctamente!!! La hora de fin es 10:24:11pm Logs insertados Nro logs: 57372 Archivo Log, cerrado!!! La hora de ini limpieza es 10:24:11pm La hora de fin limpieza es 10:24:29pm Archivo Log, depurado e insertado filtrado!!!	LOG 24.5 La hora de inicio es 08:09:38am El fichero fue subido correctamente!!! La hora de fin es 10:30:48am Logs insertados Nro logs: 132890 Archivo Log, cerrado!!! La hora de ini limpieza es 10:30:48am La hora de fin limpieza es 10:31:35am Archivo Log, depurado e insertado filtrado!!!
LOG 23.5 La hora de inicio es 10:45:39am El fichero fue subido correctamente!!! La hora de fin es 12:03:09pm Logs insertados Nro logs: 72252 Archivo Log, cerrado!!! La hora de ini limpieza es 12:03:09pm La hora de fin limpieza es 12:04:00pm Archivo Log, depurado e insertado filtrado!!!	LOG 22.5 La hora de inicio es 12:15:58pm El fichero fue subido correctamente!!! La hora de fin es 01:20:22pm Logs insertados Nro logs: 70114 Archivo Log, cerrado!!! La hora de ini limpieza es 01:20:22pm La hora de fin limpieza es 01:20:51pm Archivo Log, depurado e insertado filtrado!!!
LOG 21.5 La hora de inicio es 01:41:07pm El fichero fue subido correctamente!!!	LOG 20.5 La hora de inicio es 02:54:21pm El fichero fue subido correctamente!!!

La hora de fin es 02:44:28pm Logs insertados Nro logs: 59030 Archivo Log, cerrado!!! La hora de ini limpieza es 02:44:28pm La hora de fin limpieza es 02:44:55pm Archivo Log, depurado e insertado filtrado!!!	La hora de fin es 03:45:59pm Logs insertados Nro logs: 53302 Archivo Log, cerrado!!! La hora de ini limpieza es 03:45:59pm La hora de fin limpieza es 03:46:37pm Archivo Log, depurado e insertado filtrado!!!
LOG 19.5 La hora de inicio es 03:56:47pm El fichero fue subido correctamente!!! La hora de fin es 04:43:58pm Logs insertados Nro logs: 50505 Archivo Log, cerrado!!! La hora de ini limpieza es 04:43:58pm La hora de fin limpieza es 04:44:52pm Archivo Log, depurado e insertado filtrado!!!	LOG 18.5 La hora de inicio es 05:44:07pm El fichero fue subido correctamente!!! La hora de fin es 06:51:32pm Logs insertados Nro logs: 65913 Archivo Log, cerrado!!! La hora de ini limpieza es 06:51:32pm La hora de fin limpieza es 06:52:07pm Archivo Log, depurado e insertado filtrado!!!
LOG 17.5 La hora de inicio es 07:13:27pm El fichero fue subido correctamente!!! La hora de fin es 08:19:45pm Logs insertados Nro logs: 70330 Archivo Log, cerrado!!! La hora de ini limpieza es 08:19:45pm La hora de fin limpieza es 08:20:31pm Archivo Log, depurado e insertado filtrado!!!	LOG 16.5 La hora de inicio es 09:25:29pm El fichero fue subido correctamente!!! La hora de fin es 10:06:51pm Logs insertados Nro logs: 43207 Archivo Log, cerrado!!! La hora de ini limpieza es 10:06:51pm La hora de fin limpieza es 10:07:21pm Archivo Log, depurado e insertado filtrado!!!
LOG 15.5 La hora de inicio es 10:20:55pm El fichero fue subido correctamente!!! La hora de fin es 11:15:11pm Logs insertados Nro logs: 58756 Archivo Log, cerrado!!! La hora de ini limpieza es 11:15:11pm La hora de fin limpieza es 11:15:49pm Archivo Log, depurado e insertado filtrado!!!	LOG 14.5 La hora de inicio es 11:20:03pm El fichero fue subido correctamente!!! La hora de fin es 11:58:19pm Logs insertados Nro logs: 44864 Archivo Log, cerrado!!! La hora de ini limpieza es 11:58:19pm La hora de fin limpieza es 11:58:50pm Archivo Log, depurado e insertado filtrado!!!
LOG 13.5 La hora de inicio es 06:59:15am El fichero fue subido correctamente!!! La hora de fin es 07:47:30am Logs insertados Nro logs: 50304 Archivo Log, cerrado!!! La hora de ini limpieza es 07:47:30am La hora de fin limpieza es 07:48:05am Archivo Log, depurado e insertado filtrado!!!	LOG 12.5 La hora de inicio es 08:12:29am El fichero fue subido correctamente!!! La hora de fin es 08:58:27am Logs insertados Nro logs: 50258 Archivo Log, cerrado!!! La hora de ini limpieza es 08:58:27am La hora de fin limpieza es 08:59:03am Archivo Log, depurado e insertado filtrado!!!
LOG 11.5 La hora de inicio es 09:08:38am El fichero fue subido correctamente!!!	LOG 10.5 La hora de inicio es 03:36:42pm

La hora de fin es 10:18:17am Logs insertados Nro logs: 73745 Archivo Log, cerrado!!! La hora de ini limpieza es 10:18:18am La hora de fin limpieza es 10:19:00am Archivo Log, depurado e insertado filtrado!!!	El fichero fue subido correctamente!!! La hora de fin es 04:49:30pm Logs insertados Nro logs: 80150 Archivo Log, cerrado!!! La hora de ini limpieza es 04:49:30pm La hora de fin limpieza es 04:50:12pm Archivo Log, depurado e insertado filtrado!!!
LOG 09.5 La hora de inicio es 04:58:28pm El fichero fue subido correctamente!!! La hora de fin es 06:02:51pm Logs insertados Nro logs: 73449 Archivo Log, cerrado!!! La hora de ini limpieza es 06:02:51pm La hora de fin limpieza es 06:03:34pm Archivo Log, depurado e insertado filtrado!!!	LOG 08.5 La hora de inicio es 06:30:41pm El fichero fue subido correctamente!!! La hora de fin es 08:43:13pm Logs insertados Nro logs: 146183 Archivo Log, cerrado!!! La hora de ini limpieza es 08:43:14pm La hora de fin limpieza es 08:44:13pm Archivo Log, depurado e insertado filtrado!!!
LOG 07.5 La hora de inicio es 06:26:50am El fichero fue subido correctamente!!! La hora de fin es 07:22:21am Logs insertados Nro logs: 69140 Archivo Log, cerrado!!! La hora de ini limpieza es 07:22:21am La hora de fin limpieza es 07:23:10am Archivo Log, depurado e insertado filtrado!!!	LOG 06.5 La hora de inicio es 08:13:31am El fichero fue subido correctamente!!! La hora de fin es 08:51:58am Logs insertados Nro logs: 58277 Archivo Log, cerrado!!! La hora de ini limpieza es 08:51:58am La hora de fin limpieza es 08:52:43am Archivo Log, depurado e insertado filtrado!!!
LOG 05.5 La hora de inicio es 09:12:31am El fichero fue subido correctamente!!! La hora de fin es 10:06:42am Logs insertados Nro logs: 75379 Archivo Log, cerrado!!! La hora de ini limpieza es 10:06:42am La hora de fin limpieza es 10:07:29am Archivo Log, depurado e insertado filtrado!!!	LOG 04.5 La hora de inicio es 10:23:03am El fichero fue subido correctamente!!! La hora de fin es 10:53:22am Logs insertados Nro logs: 45070 Archivo Log, cerrado!!! La hora de ini limpieza es 10:53:22am La hora de fin limpieza es 10:54:21am Archivo Log, depurado e insertado filtrado!!!

Tabla 15.- Relacion de las respuestas del script, al estructurar los logs

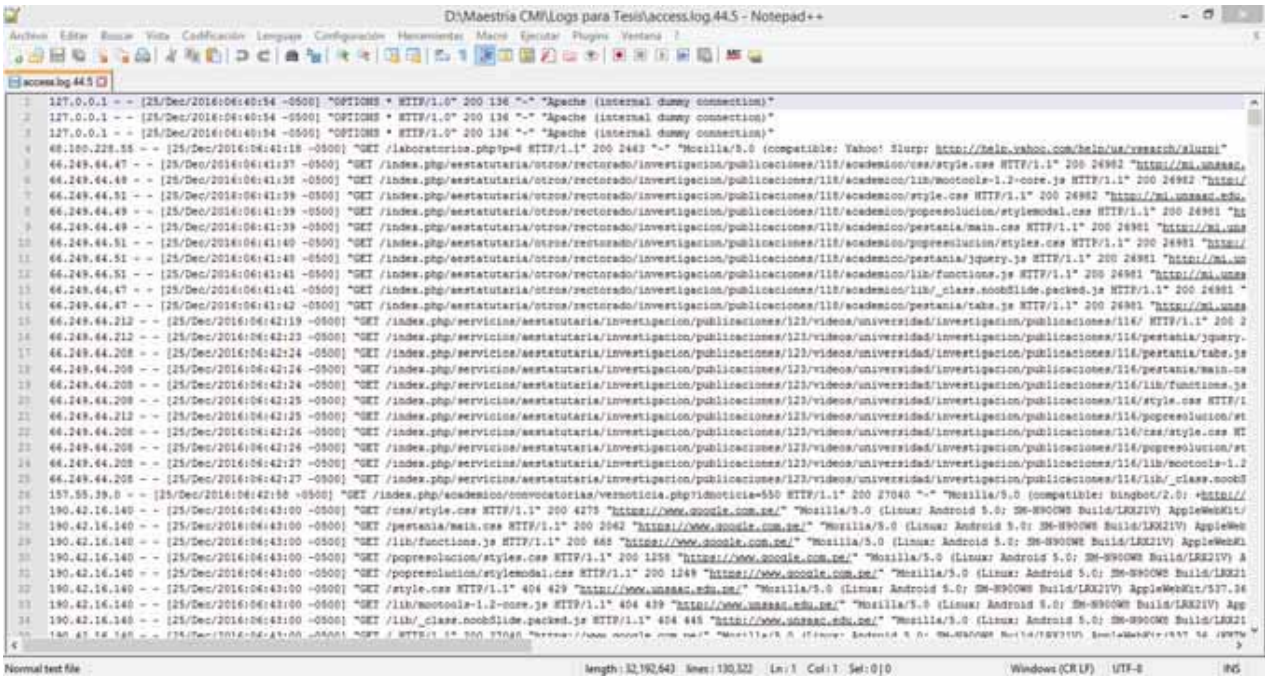

Figura 50 .- Access log 44.5(original)

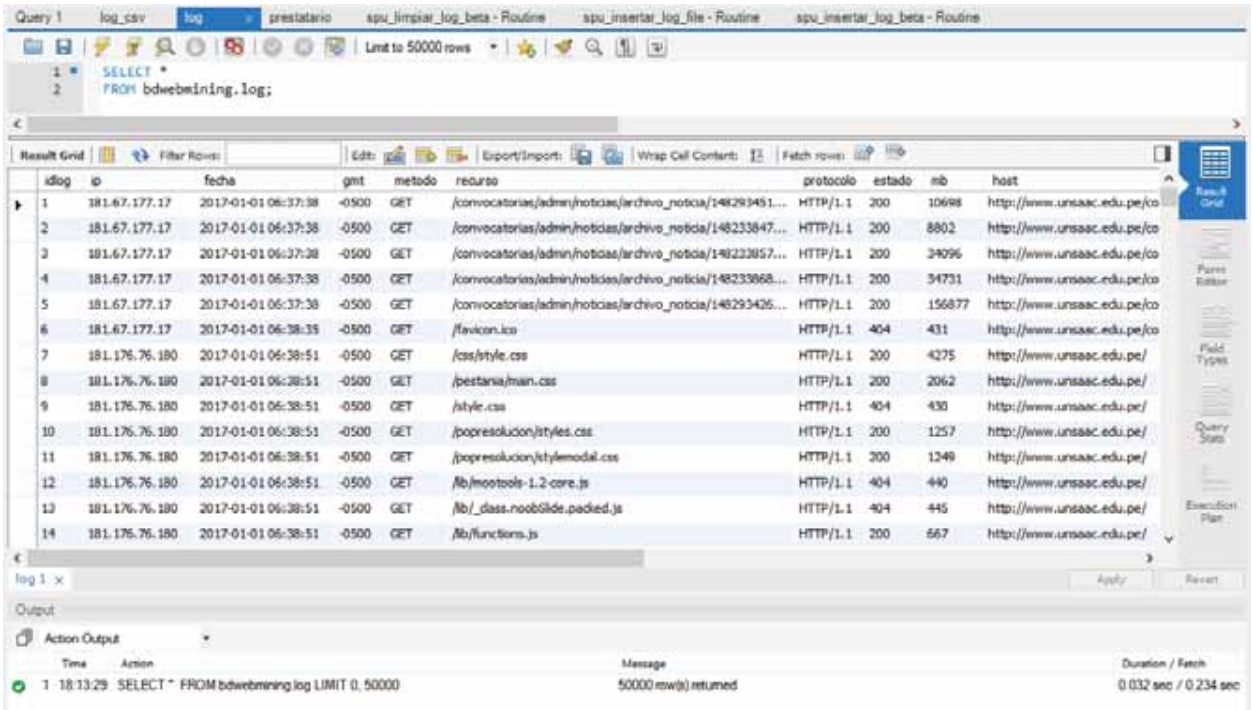

Figura 51.- Logs estructurados en la base de datos.

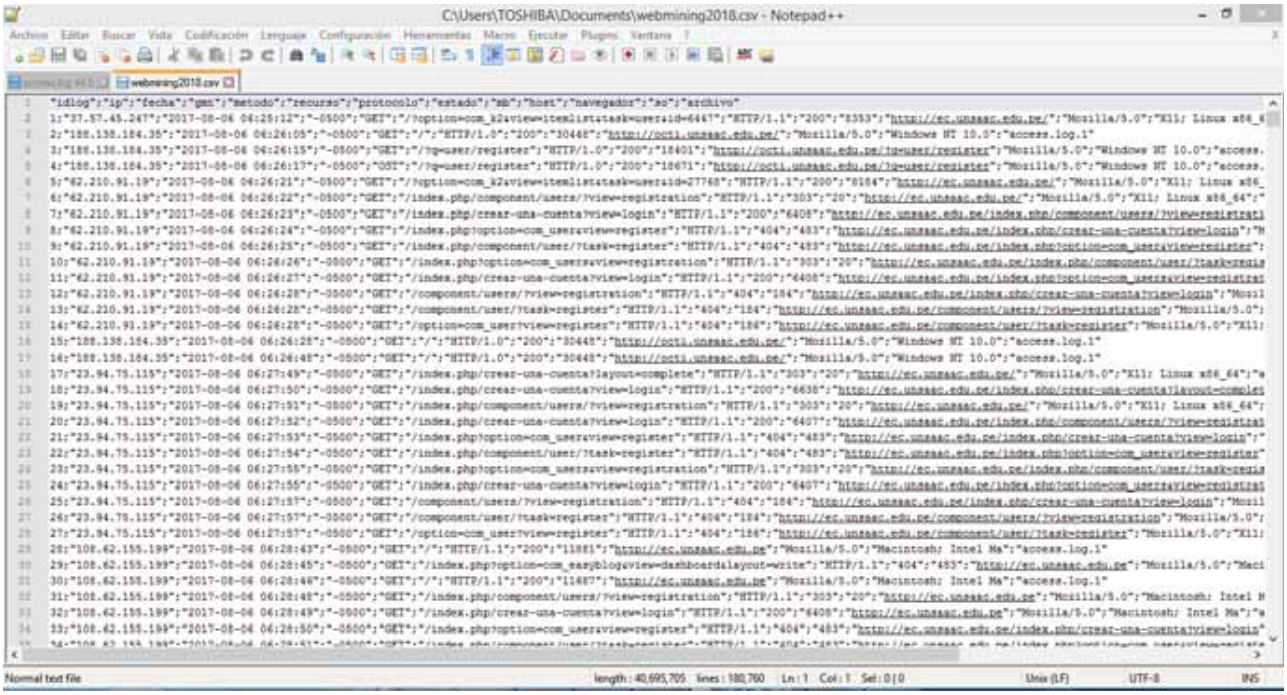

Figura 52.- Logs estructurados en formato CSV