Análisis de patrones de navegación de los estudiantes dentro del Campus UOC

Aplicaciones Web para trabajo colaborativo

Proyecto Fin de Carrera. Curso 2011/12

Memoria

Consultor: Fatos Xhafa

Autor: Antonio López Martínez

Control de versiones

Versión	Fecha	Descripción
1.0	04/06/2012	Versión inicial del documento

ÍNDICE

1 Resumen

El **Proyecto Final de Carrera** (PFC), como asignatura de la Universitat Oberta de Catalunya (UOC), consiste en la realización de un **trabajo de síntesis** de los conocimientos adquiridos en otras asignaturas de la carrera.

El presente PFC se enmarca en el área de **Aplicaciones Web para trabajo colaborativo** y tiene por **objetivo especificar un proceso para analizar** los **patrones de navegación** (del inglés **user behaviour patterns** o **web navigational patterns**) de los estudiantes de la UOC **en el Campus Virtual**.

En la minería de datos web, suelen considerarse tres ámbitos de investigación: (1) minería de la información, (2) minería de la estructura de un sitio web, y (3) minería para determinar patrones de navegación [7]. El presente PFC se centra en este último, concretamente en el análisis de patrones de navegación.

En muchos casos, ciertas **acciones** que los usuarios realizan en un sitio web, son **registradas en** los **logs** del servidor; el estudio del comportamiento de los usuarios a través de lo registrado en estos logs, permite la identificación de patrones de navegación.

Concretamente, el **estudio** realizado en el marco de este PFC, tomará como referencia **datos reales de los logs del Campus Virtual** de la UOC y **posibilitará** el análisis de los **patrones de navegación** de los estudiantes.

Actualmente, los datos de los logs del Campus Virtual son **procesados** por una **aplicación** que implementa un **algoritmo de bi-clustering** y, en el alcance inicial del presente PFC se contemplaba su instalación y uso. Sin embargo, el que esta aplicación esté pensada para ser ejecutada en una infraestructura informática de alto rendimiento, y dadas las limitaciones en el tiempo disponible para realizar un PFC de estas características, hicieron que **se desaconsejara su uso**.

Debido a ello, el **alcance final** del presente PFC se ha visto modificado para contemplar la **construcción** de una **aplicación Java** que permita realizar el preprocesamiento de los ficheros de log del Campus Virtual de la UOC, **generando un fichero con estructura y contenido adecuados para** ser procesado por **una aplicación de minería de datos**. En el caso del presente PFC, la aplicación seleccionada ha sido **Weka** que, gracias a los algoritmos que incluye, posibilita la obtención de los patrones de navegación buscados.

2 Plan de trabajo

El alcance inicial del PFC según el plan docente, contemplaba la realización del procesamiento de los ficheros de log del Campus Virtual mediante una **aplicación** que implementa un **algoritmo de bi-clustering**. Sin embargo, el que esta aplicación esté pensada para ser ejecutada en una infraestructura informática de alto rendimiento, y dadas las limitaciones en el tiempo disponible para realizar un PFC de estas características, hicieron que **se desaconsejara su uso** y fuera necesario valorar alternativas.

Debido a lo anterior, el plan de trabajo que se incluye a continuación no es el previsto inicialmente que fue realizado durante la PEC1 del proyecto, sino la adecuación final del mismo contemplando los cambios que han ido apareciendo a lo largo de la ejecución del proyecto.

2.1 Descripción del proyecto

2.1.1 Objetivos del proyecto

El presente proyecto tiene por **objetivo** principal **especificar un proceso para analizar los patrones de navegación** de los estudiantes de la UOC en el Campus Virtual **a partir de los datos reales contenidos en los logs**.

Una vez logrado el objetivo principal, en base a los patrones de navegación analizados, se extraerá un conjunto de posibles mejoras para el Campus Virtual y para los ficheros de log generados por el servidor.

2.1.2 Resultados del proyecto

A la finalización del proyecto se espera **haber especificado un proceso y haberlo aplicado para determinar y analizar de los patrones de navegación** más habituales **de los estudiantes de la UOC**.

La evolución del proyecto y el grado de consecución del objetivo principal podrá valorarse a través de un conjunto de **resultados intermedios esperados** y de un conjunto de documentos **entregables** que reflejen la progresión y el conocimiento que se va obteniendo:

1. **Entregable 1. Documento plan de trabajo**: en él se especificará el plan de trabajo que gobernará la ejecución del proyecto.
2. Estudio de técnicas de análisis de patrones de navegación, algoritmos y selección de la aplicación para el procesamiento de los log: proceso cuyo resultado esperado es la **adquisición del conocimiento** necesario para alcanzar plenamente los objetivos del proyecto.
3. **Entregable 2. Documento de especificación y análisis**: en él se detallarán las **conclusiones del análisis** realizado **sobre los ficheros de log** del Campus Virtual de la UOC tomados como muestra, **y se especificará** el **proceso que se seguirá para lograr los objetivos** del proyecto.
4. **Entregable 3. Aplicación para el preprocesamiento de los log del Campus Virtual**: este entregable consistirá en una aplicación construida como parte del alcance del presente proyecto, que posibilitará el tratamiento inicial de los ficheros de log del Campus Virtual para que puedan ser interpretados por la aplicación de minería de datos seleccionada.

5. **Procesamiento** de los logs proporcionados por el Consultor. Proceso cuyo resultado esperado es la determinación y análisis de los patrones de navegación de los usuarios del Campus Virtual.
6. Determinar posibles mejoras en el Campus Virtual y en los ficheros de log en base a los patrones de navegación analizados.
7. **Entregable 4. Memoria del PFC**: exposición detallada del trabajo realizado en el PFC y de los resultados alcanzados.
8. **Entregable 5. Presentación virtual**: documento de presentación del PFC sintetizando de forma clara y concisa el trabajo realizado y los resultados alcanzados.

2.1.3 Análisis de riesgos

Durante la elaboración del plan del proyecto, se han identificado los siguientes riesgos:

Disponibilidad del responsable de la ejecución	
Identificador	RI-01
Descripción	Debido a la necesidad de compaginar la ejecución del proyecto con su actividad laboral, el grado de disponibilidad de la persona responsable de la ejecución del proyecto podría no ser el esperado.
Impacto	Plazos de desarrollo
Probabilidad	Alta
Acciones	1. Realizar una planificación adecuada del proyecto y un plan de trabajo que contemple esta situación. 2. Realizar un seguimiento continuo del proyecto de tal forma que se detecten rápidamente desviaciones en los plazos de desarrollo. 3. Determinar con claridad el alcance del proyecto que posibilite la consecución del objetivo principal fijado.

Periodos de inactividad	
Identificador	RI-02
Descripción	El responsable de ejecución del proyecto es una única persona, por lo que podría haber paradas debido a enfermedad, trabajo o problemas familiares.
Impacto	Plazos de desarrollo
Probabilidad	Baja
Acciones	1. Realizar esfuerzos adicionales para recuperar las horas perdidas.

Pérdida de información	
Identificador	RI-03
Descripción	El dispositivo donde se están construyendo y almacenando los entregables del proyecto podría resultar dañado y provocar la pérdida de información.

Impacto	Plazos de desarrollo
Probabilidad	Baja
Acciones	1. Realizar copias de seguridad al finalizar cada jornada de trabajo en el proyecto. Dichas copias de seguridad deben estar guardadas en una ubicación externa al dispositivo en el que se está realizando el proyecto. 2. Disponer de un dispositivo alternativo en el cual continuar la realización del proyecto en caso de avería del principal.

2.2 Alcance del proyecto

Se considera dentro del alcance de este proyecto:

- Análisis de los ficheros de log del Campus Virtual de la UOC.
- Estudio de técnicas de análisis de patrones de navegación y algoritmos y aplicaciones de minería de datos.
- Especificación de un proceso para el análisis de patrones de navegación.
- Construcción de una aplicación para el tratamiento inicial de los ficheros de log del Campus Virtual de la UOC.
- Procesamiento de los logs del Campus Virtual, determinación y análisis de patrones de navegación más habituales de los estudiantes.
- Determinación de un conjunto de posibles mejoras para el Campus Virtual, para los ficheros de log y para el propio proceso realizado.

2.2.1 Análisis de los ficheros de log del Campus Virtual

Con la finalidad de poder disponer de datos reales, el Consultor proporcionará un conjunto de ficheros de log de muestra que será utilizados para la ejecución del proyecto; por cuestiones de seguridad y privacidad de la información, este conjunto de logs será muy limitado. **A partir de los ficheros de log del Campus Virtual** enviados por el Consultor, **se realizará un análisis de su estructura**.

2.2.2 Adquisición del conocimiento

El proyecto contemplará una necesaria fase de adquisición del conocimiento consistente en:

- Estudio de técnicas de análisis de patrones de navegación.
- Determinación de los algoritmos a aplicar.
- Localización de una aplicación de minería de datos que posibilite la obtención y análisis de los patrones buscados.

Este proceso de adquisición del conocimiento se basará fundamentalmente en la localización y lectura de información en Internet dando preferencia a los artículos científicos procedentes de fuentes fiables. Como resultado de este proceso de adquisición del conocimiento, se recopilarán un conjunto de referencias documentales que serán incluidas en la memoria final del proyecto.

2.2.3 Especificación del proceso para en análisis de patrones de navegación

Para poder analizar patrones de navegación de los estudiantes del Campus Virtual de la UOC, es importante **disponer de un proceso normalizado que permita realizar** dicho **análisis**; como parte del alcance del presente PFC, se realizará la especificación de dicho método.

2.2.4 Construcción de una aplicación para el tratamiento de ficheros de log

Teniendo presente que la estructura de los ficheros de log del Campus Virtual no podrá ser interpretada directamente por la aplicación de minería de datos seleccionada, como parte del alcance del proyecto se contempla la construcción de una aplicación informática que, implementando parte del proceso especificado, sea capaz de realizar un tratamiento previo de estos ficheros y generar como salida un fichero en un formato adecuado.

2.2.5 Procesamiento de los ficheros de log

El procesamiento de los ficheros de log se realizará mediante las dos aplicaciones informáticas anteriormente indicadas y **aplicando el proceso especificado**.

1. Por una parte, los ficheros de log iniciales serán tratados por la aplicación de preprocesamiento de tal forma que se obtenga un fichero de salida capaz de ser interpretado por la aplicación de minería de datos.
2. Por último, el fichero obtenido será procesado por la aplicación de minería de datos a través del conjunto de algoritmos seleccionado.

Aplicando el conjunto de algoritmos de minería de datos seleccionados, se obtendrán los patrones de navegación que serán analizados para sacar conclusiones.

2.2.6 Determinación de mejoras

En base a los resultados obtenidos, se valorarán **posibles mejoras** para el Campus Virtual y los ficheros de log generados por el servidor.

Dentro de este conjunto de mejoras, se considerarán también aquellas que se determinen como interesantes para la mejora del propio proceso ejecutado como parte del presente PFC.

2.3 Organización del proyecto

2.3.1 Relación de actividades

2.3.1.1 Lanzamiento del proyecto

Esta actividad comenzó el miércoles 29 de febrero de 2012 con el inicio del segundo semestre del curso 2011/12 en la UOC y los primeros contactos con el consultor, y finalizó el domingo 4 de marzo de 2012 con la aprobación de la propuesta del presente proyecto.

2.3.1.2 Planificación del proyecto y análisis de requisitos

Esta actividad tendrá como resultado la **elaboración del plan del proyecto**, considerado como el primer entregable del proyecto.

La ejecución de esta actividad se dividirá en las siguientes subactividades:

1. Definición y planificación del proyecto
2. Análisis de requisitos

Duración estimada: entre el 5 de marzo de 2012 y el 26 de marzo de 2012

DEFINICIÓN Y PLANIFICACIÓN DEL PROYECTO

Esta actividad dará comienzo con la realización de la definición y planificación del proyecto y posibilitará la producción de la primera versión del plan del proyecto. En esta primera versión del documento serán incluidos los requisitos conocidos con anterioridad a la ejecución de la siguiente subactividad.

ANÁLISIS DE REQUISITOS

Una vez realizada la definición y planificación del proyecto, se contactará con el Consultor solicitándole la información necesaria que posibilite el análisis de los requisitos del proyecto. La ejecución de esta subactividad producirá una versión ampliada del presente documento.

2.3.1.3 Desarrollo del proyecto

La ejecución de esta actividad se dividirá en las siguientes subactividades que transcurrirán, inicialmente, en paralelo:

1. Adquisición del conocimiento
2. Desarrollo

Duración estimada: entre el 27 de marzo de 2012 y el 28 de abril de 2012.

ADQUISICIÓN DEL CONOCIMIENTO

Con la ejecución de esta subactividad se pretende alcanzar el conocimiento necesario que posibilite el correcto desarrollo del proyecto y la consecución de los objetivos definidos.

DESARROLLO

Esta subactividad consistirá en la realización de la especificación y análisis del proyecto de forma detallada, tomando como referencia los requisitos determinados durante la actividad anterior.

Durante esta subactividad se realizará lo siguiente:

- Análisis detallado de la estructura de los logs del Campus Virtual
- Especificación del proceso para el tratamiento de los logs del Campus Virtual y el análisis de patrones de navegación.
- Selección de la aplicación de minería de datos.
- Realización de una prueba piloto que permita valorar lo realizado hasta la fecha.

  A imagen y semejanza de lo que se realizará finalmente, la prueba piloto consistirá en el procesamiento de ficheros con la aplicación de minería de datos, determinación y análisis de los

patrones de navegación. La prueba piloto se caracterizará por que el fichero de log utilizado estará construido a mano, será de tamaño reducido y sus contenidos estarán previamente adecuados.

Llegados a este punto se construirá el documento de especificación y análisis (segundo entregable) y se continuará realizando lo siguiente:

- Construcción de la aplicación para el tratamiento inicial de los ficheros de log (tercer entregable).
- Obtención de mejoras para el Campus Virtual y los logs del servidor.

2.3.1.4 Entrega final

Esta actividad posibilitará la **construcción de la memoria del PFC** (cuarto entregable) **y** del **documento de presentación** (quinto entregable) y con su ejecución concluirá el trabajo iniciado durante la actividad de desarrollo del proyecto.

A lo largo de esta actividad se determinarán los patrones de navegación de los estudiantes del Campus Virtual y se realizará un análisis de los mismos aplicando el proceso especificado.

Duración estimada: entre el 29 de abril de 2012 y el 4 de junio de 2012.

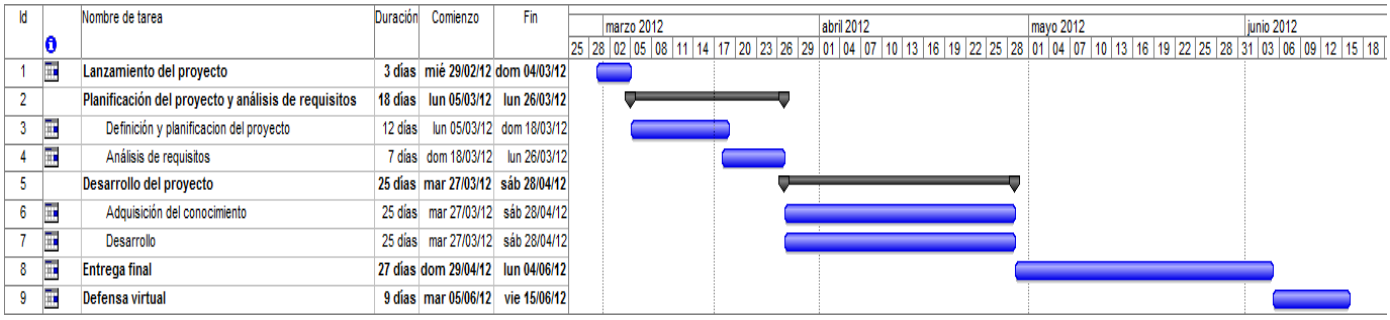
2.3.1.5 Defensa virtual

Con esta actividad se dará por finalizada la ejecución del PFC realizando la defensa virtual del mismo.

Duración estimada: entre el 5 de junio de 2012 y el 15 de junio de 2012.

2.3.2 Calendario de trabajo

El calendario de trabajo puede observarse a través del siguiente diagrama de Gantt:



Id	Nombre de tarea	Duración	Comienzo	Fin
1	**Lanzamiento del proyecto**	**3 días**	**mié 29/02/12**	**dom 04/03/12**
2	**Planificación del proyecto y análisis de requisitos**	**18 días**	**lun 05/03/12**	**lun 26/03/12**
3	Definición y planificacion del proyecto	12 días	lun 05/03/12	dom 18/03/12
4	Análisis de requisitos	7 días	dom 18/03/12	lun 26/03/12
5	**Desarrollo del proyecto**	**25 días**	**mar 27/03/12**	**sáb 28/04/12**
6	Adquisición del conocimiento	25 días	mar 27/03/12	sáb 28/04/12
7	Desarrollo	25 días	mar 27/03/12	sáb 28/04/12
8	**Entrega final**	**27 días**	**dom 29/04/12**	**lun 04/06/12**
9	**Defensa virtual**	**9 días**	**mar 05/06/12**	**vie 15/06/12**

2.3.3 Hitos principales y entregables

Fecha	Descripción del hito
04/03/2012	Formalización de la solicitud de propuesta del PFC.
18/03/2012	Finalización de la primera versión del documento de plan de proyecto.

26/03/2012	Segunda versión del documento de plan de proyecto incluyendo el análisis inicial de requisitos. Entregables: • Entregable 1. Plan de proyecto
28/04/2012	Finalización de la fase de desarrollo del proyecto. Se ha debido adquirir el conocimiento necesario para cumplir los objetivos del proyecto. Se ha debido realizar el análisis completo del proyecto, la especificación del proceso y la prueba piloto que posibilite la obtención de los primeros resultados. Entregables: • Entregable 2. Documento de especificación y análisis.
04/06/2012	Finalización de la memoria del PFC y del documento de presentación. Entregables: • Entregable 3. Aplicación de preprocesamiento de logs (código fuente y ejecutables) • Entregable 4. Memoria del PFC • Entregable 5. Documento de presentación
15/06/2012	Realización de la defensa virtual del PFC en una fecha anterior o igual a la indicada.

2.3.4 Equipo de trabajo

El equipo de trabajo se compone de dos personas:

- El responsable y autor del PFC (Antonio López).
- El Consultor (Fatos Xhafa) que proporcionará el material necesario para lograr el cumplimiento de los objetivos definidos, realizará el seguimiento del proyecto y validará lo realizado.

2.3.5 Definición de roles

Debido a la naturaleza académica del proyecto, Antonio López asumirá los roles necesarios (jefe de proyecto, analista...) necesarios para la buena marcha y correcta ejecución del proyecto.

El Consultor tendrá el rol de cliente por tratarse de un 'emisor' de requisitos y ser responsable de la aceptación final de lo realizado. A la par, el Consultor asumirá responsabilidades de control y seguimiento del proyecto.

Así pues, la distribución de roles queda de la siguiente manera:

Rol	Persona asignada
Coordinador	Fatos Xhafa
Jefe de proyecto	Antonio López

Analista	Antonio López
Desarrollador	Antonio López
Responsable de pruebas	Antonio López

2.3.6 Mecanismos de control

Se considera conveniente establecer como mecanismos de control los siguientes:

- El seguimiento del cumplimiento de las fechas, tanto las fijadas por el Consultor como las definidas como hitos en el presente documento.
- El envío al Consultor de borradores de los entregables previstos, de tal forma que pueda realizar una validación previa de los mismos antes de las entregas definitivas.

2.4 Valoración económica

Por tratarse de un proyecto puramente académico, el coste del mismo se presupone nulo, no considerando los costes del equipo de trabajo. La no necesidad de asumir los costes de las licencias del software utilizado, también ha sido considerada a efectos de realizar la presente valoración económica.

3 Especificación y análisis

En este apartado se detallan las **conclusiones del análisis** realizado **sobre dos ficheros de log** del Campus Virtual de la UOC tomados como muestra, **y se describe** el **proceso que se seguirá** durante el desarrollo del proyecto **para lograr el objetivo final** fijado.

3.1 Problemas previos a resolver

Para lograr el objetivo del proyecto, los principales **problemas** detectados vienen dados por el **tamaño y contenido de los ficheros de log** del Campus Virtual de la UOC.

- El problema del **tamaño** viene dado a consecuencia de que se va a trabajar con ficheros que contienen millones de líneas y que por lo tanto requieren considerable capacidad de cálculo para ser leídos y procesados. Sirva como dato, que los dos ficheros de log proporcionados por el Consultor tienen un tamaño que ronda los 2 y 4 GB respectivamente y que estos ficheros únicamente registran la actividad de un día del Campus Virtual de la UOC.

  Ante la no disponibilidad de un entorno de procesamiento adecuado, el estudio tomará como **muestra** fragmentos de dichos ficheros de log.

- En el caso del **contenido**, si bien los ficheros de log siguen una estructura definida, esta estructura no permite extraer de forma directa los patrones de navegación cuyo análisis es el objeto del presente PFC; por ello será necesario realizar un **preprocesamiento** de dichos ficheros de log.

  Este problema podría mitigarse adecuando la salida de log del servidor del Campus Virtual de la UOC [16], pero dado que esto no es posible (ni aconsejable, puesto que su uso va más allá del análisis de patrones de navegación), se ha considerado dentro del alcance del presente PFC, la construcción de una aplicación que posibilite la realización de este preprocesamiento.

3.2 Estructura de los logs del Campus Virtual

Los ficheros de log del Campus Virtual de la UOC están formados por millones de líneas, donde cada una de ellas representa la operación efectuada por un determinado estudiante dentro del Campus Virtual. Entre otra información, estas líneas registran la dirección IP del dispositivo desde el que se ha accedido y no el nombre de usuario del estudiante, lo que asegura una mayor privacidad.

A modo de ejemplo, esta es una de las líneas (previa anonimización de la dirección IP) que forma parte de los ficheros log del Campus Virtual:

```
[13/Mar/2012:00:15:42 +0100] xxx.xxx.xxx.xxx "POST /tren/trenacc HTTP/1.1" 200 "https://cv.uoc.edu/tren/trenacc" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.11 (KHTML, like Gecko) Chrome/17.0.963.78 Safari/535.11" 14943 157
```

Cabe destacar que la línea de ejemplo está simplificada, generalmente las URL que aparecen en los ficheros de log, van acompañadas por numerosos parámetros.

Dado que el volumen de información diario que se registra en los logs puede superar el Gigabyte de tamaño, la UOC ha definido una política de rotación diaria para los logs, generándose un nuevo fichero de log cada día, lo que hace su tratamiento más eficiente.

ANÁLISIS DE DATOS

El hecho de que los ficheros de log sigan una estructura bien definida facilitará la aplicación de técnicas de minería de datos, permitiendo la extracción de perfiles y la realización de análisis de datos.

Por ejemplo, si fuera necesario se podría averiguar cuáles son las secciones con más y menos visitas del campus (a través de las URL registradas), cuales son la fechas o momentos de mayor y menor actividad (analizando un conjunto muestra de los logs diarios del Campus Virtual), la secuencia de acciones realizadas desde una determinada IP para llegar a su destino dentro del Campus Virtual...

Si bien es una ventaja para el tratamiento individual de los logs, la división de los mismos en forma de ficheros diarios, puede suponer una dificultad añadida, a la hora de realizar análisis trasversales de la información como, por ejemplo, la evolución en el número de accesos de los estudiantes a su expediente (que se prevé más elevado en fechas correspondientes a la finalización de un semestre).

3.2.1 Estructura de una línea

De lo anterior, puede concluirse que los ficheros de log del Campus Virtual de la UOC no siguen una estructura estandarizada (por ejemplo XML), pero si mantienen una estructura donde cada una de las líneas sigue este patrón:

```
[<fecha y hora registro>] <dirección IP> "<método petición> <ruta recurso solicitado> <protocolo>" <código respuesta> "<ruta origen>" "<navegador>" <n.º bytes obtenido> <p1>
```

Esta estructura es muy común en los ficheros de log generados por los servidores web más conocidos, por ejemplo Apache [16].

DESCRIPCIÓN DE LOS ELEMENTOS DE UNA LÍNEA

Cada una de las líneas representa la solicitud de acceso a un recurso realizada desde una determinada dirección IP y está formada por los siguientes elementos [16] separados por espacios:

- **fecha y hora registro**: Momento temporal en el que se ha registrado la operación. Su formato es el siguiente: `<día>/<mes>/<año>:<hora>:<minuto>:<segundo> <uso horario>`. Donde:
  - **día**: Día de registro de la operación en formato `dd`.
  - **mes**: Mes de registro de la operación en formato `mmm` (los meses se representan con tres letras; por ejemplo marzo se representa como 'Mar').
  - **año**: Año de registro de la operación en formato `aaaa` (cuatro dígitos de año).
  - **hora**: Hora de registro de la operación en formato `hh`.
  - **minuto**: Minuto de registro de la operación en formato `MM`.
  - **segundo**: Segundo de registro de la operación en formato `ss`.
  - **huso horario**: Modificador horario según el Tiempo Medio de Greenwich (GMT).
- **dirección IP**: Dirección IP del dispositivo desde el que se ha realizado la acción registrada en el log.

- **método petición**: Cada una de las líneas de los logs del Campus Virtual lleva asociado un tipo de operación identificado por el método de petición utilizado (POST, GET, HEAD...). Se hablará de los métodos de petición un poco más adelante en este mismo documento.
- **ruta recurso solicitado**: Ruta del Campus Virtual a la que se ha accedido, acompañada de los parámetros necesarios para realizar la operación demandada.
- **protocolo**: Protocolo utilizado para acceder al recurso. Por ejemplo: HTTP/1.1.
- **código respuesta**: Código de respuesta obtenido tras el acceso al recurso.
- **ruta origen**: Dirección desde la que se ha realizado la solicitud del recurso.
- **navegador**: Información identificativa del navegador desde el que se ha accedido al recurso.
- **n.º bytes obtenido**: total de bytes recibidos al acceder al recurso. Por ejemplo, en caso de haber accedido correctamente a un fichero, el valor que aquí figuraría sería el tamaño del mismo.
- **p1**: el log del Campus Virtual de la UOC dispone de un último parámetro que no ha conseguido identificarse. No obstante, dado que su relevancia es nula para el presente estudio, únicamente vamos a limitarnos a señalar su existencia.

3.2.2 Códigos de respuesta

Antes de continuar, es importante señalar que el protocolo considerado a efectos del presente estudio será únicamente HTTP; las líneas de los ficheros log que refieran a otros protocolos serán descartadas y no se considerarán a efectos de la determinación de patrones de navegación.

En el caso del protocolo HTTP, estos códigos de respuesta están normalizados como puede verse en [10, 17], por lo que es posible determinar cuándo se ha producido un error en el acceso al recurso o cuando este se ha realizado con éxito.

Estos códigos están formados por tres dígitos siendo el primero de ellos el que identifica el tipo de respuesta según los siguientes criterios:

- **1xx**: Informativo. La petición se recibe y sigue el proceso. Esta familia de respuestas indican una respuesta provisional.
- **2xx**: Éxito. La acción requerida por la petición ha sido recibida, entendida y aceptada.
- **3xx**: Redirección. Para completar la petición se han de tomar más acciones.
- **4xx**: Error del cliente. La petición no es sintácticamente correcta y no se puede llevar a cabo.
- **5xx**: Error del servidor. El servidor falla al atender la petición que aparentemente es correcta.

Por ser visibles desde el navegador, son especialmente conocidos los siguientes códigos:

Código	Descripción
400	Petición errónea.
401	Acceso no autorizado.
403	Acceso prohibido.
404	Recurso no encontrado.

3.2.3 Métodos de petición

Como se ha indicado anteriormente, en cada una de las líneas de los logs, se indica el método bajo el que se ha realizado la solicitud de una determinada operación. Los métodos detectados en los logs de muestra sobre los que se ha realizado el presente análisis son los siguientes:

- GET: Se utiliza para recuperar información, generalmente especificando parámetros en la URL, por ejemplo: `http://host.com/script.cgi?nombre1=valor1&nombre2=valor2`
- POST: Cuando una petición se realiza usando el método POST, los datos se adjuntan a la petición a modo de objeto. Una de las ventajas de POST sobre GET es que facilita el envío de un mayor volumen de datos al servidor y que los parámetros están ocultos (no se visualizan en la URL).
- HEAD: Es equivalente al método GET excepto que el servidor no devolverá contenido, sólo las cabeceras HTTP. Generalmente se utiliza para comprobar si un enlace es válido.
- OPTIONS: Permite al cliente conocer las opciones y requisitos asociados con un recurso o las capacidades del servidor.

Aunque el protocolo HTTP considera también los siguientes métodos, como puede verse en [10]:

- PUT: Permite guardar el contenido de la petición en el servidor bajo la URL de la petición.
- DELETE: Método utilizado para que el servidor borre el recurso indicado por la URL de la petición.
- TRACE: Se utiliza para determinar si existe el receptor del mensaje enviado y usar la información para hacer un diagnóstico.

3.3 Patrones de navegación a analizar

Cuando un usuario accede al Campus Virtual de la UOC, solicita el acceso a una serie de recursos (por lo general, ficheros o páginas web); por otra parte, el acceso al Campus Virtual implica el inicio de una sesión de trabajo por parte del usuario que ha accedido, dicha sesión de trabajo es identificada por una secuencia alfanumérica.

Desde el punto de vista de los ficheros de log del Campus Virtual, los usuarios son identificados por la dirección IP del dispositivo con el que han accedido y cada acceso de un usuario a un determinado recurso, queda reflejado en una línea en el fichero de log correspondiente al día en el que se ha realizado el acceso. Por otra parte, los recursos son identificados por su URL en el Campus Virtual, a la que se incorpora el identificador de sesión del usuario que ha accedido al recurso.

Por limitaciones de tiempo, inicialmente, el presente PFC se va a centrar en el análisis de dos patrones de navegación:

1. **Recursos comúnmente accedidos**

   Este patrón permitirá identificar a aquellos **recursos solicitados de manera más usual** por los usuarios del Campus Virtual de la UOC cuyas acciones se han registrado en los logs.

   Dada la pareja dirección IP e identificador de sesión, denominaremos **secuencia de navegación** al conjunto de recursos visitados por una determinada dirección IP durante una sesión de trabajo; es decir, que una secuencia de navegación estará formada por las URL a las que ha accedido un usuario desde que inicia sesión en el Campus Virtual hasta que finaliza su actividad. De esa forma, si un mismo usuario (identificado por la dirección IP del dispositivo desde el que ha accedido) iniciara una nueva sesión, el conjunto de recursos accedidos se consideraría parte de otra secuencia.

2. **Reglas de navegación más habituales**

   Este patrón permitirá identificar los **comportamientos más usuales de los usuarios** del Campus Virtual de la UOC.

   Por ejemplo, dado el conjunto de recursos (R1, R2, R3, R4) para los que se han registrado accesos en un fichero de log del Campus Virtual, se podría determinar que los usuarios que han accedido a R1 también lo han hecho a R2, y que los usuarios que han accedido a R3 también lo han hecho a R4.

3.4 Métodos aplicados

Para poder determinar los patrones de navegación indicados en el apartado anterior, se aplicarán los siguientes métodos de minería de datos:

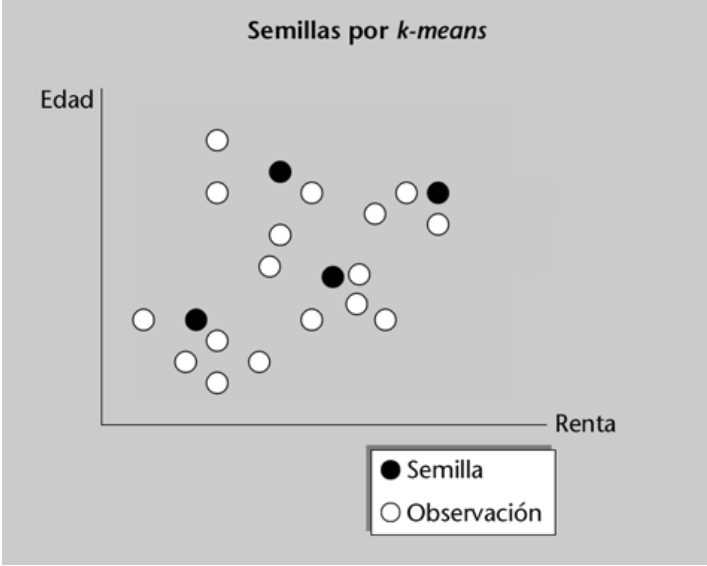
3.4.1 K-Means

K-Means (o método de los centroides) es un **método de agregación** (del inglés **clustering**) que, como tal, propone a partir de un conjunto de datos, la obtención de una **enumeración de grupos** (clusters) **de objetos con características similares**. Concretamente, el método K-Means, se basa en la obtención de un número *k* de grupos que es **fijado al principio del proceso**.

La teoría dice que el proceso comienza fijando un punto inicial del espacio (denominado semilla o, en inglés, seed) como centro del grupo potencial que se va a formar. Esta semilla puede ser bien uno de los objetos que forman parte del conjunto de datos inicial, bien una combinación de valores creada de forma artificial representando un resumen de las características de varios objetos. A partir de esto, se pueden extraer los pasos que sigue el método K-Means:

1. Seleccionar la semillas iniciales
2. Calcular los centros
3. Asignar objetos al grupo con centro más próximo
4. Recalcular los centros
5. Continuar hasta que no haya variación en los grupos

A modo de ejemplo, en la siguiente figura se pueden distinguir cuatro grupos formados alrededor de otras tantas semillas en un dominio definido sobre dos dimensiones (edad y renta):



Fuente: módulo 5 de los apuntes de la asignatura de Minería de Datos de la UOC

Una vez formados los grupos, se determina, para cada objeto a estudiar, que centroide tiene más cerca asignando dicho objeto al grupo representado por el centroide más próximo. A este proceso se le denomina agregación.

Como resultado de la agregación de objetos, es necesario volver a calcular el centro de cada grupo. Esto se hace obteniendo para cada dimensión, el valor medio de todos los objetos que forman parte del grupo tratado. Una vez calculados los nuevos centros, el proceso se inicia otra vez y se repite hasta que, en dos iteraciones consecutivas no se produzcan cambios en los centros (o se produzcan pocos).

APLICACIÓN EN EL PFC

En el caso del presente estudio, los **grupos** a considerar **serán** las **secuencias de navegación de los usuarios** del Campus Virtual de la UOC, de tal forma que se puedan determinar los recursos a los que se ha accedido con más frecuencia.

3.4.2 Apriori

El algoritmo Apriori se utiliza en minería de datos para **determinar las reglas de asociación en un conjunto de datos**. Este algoritmo se basa en el conocimiento previo (a priori) de los conjuntos de datos frecuentes. Un elemento es considerado frecuente si su frecuencia es mayor que el valor de confianza (confidence).

Es importante señalar que este algoritmo no soporta valores numéricos por lo que, en función de las características del conjunto de datos a procesar, es posible que se requiera realizar un preprocesado del mismo, generalmente una discretización.

REGLAS DE ASOCIACIÓN

Una regla de asociación es una expresión de la forma $X => Y$, donde X e Y son conjuntos de elementos. Esta expresión debe interpretarse como: cuando se observa un conjunto de datos que contiene el elemento X, también suele aparece el elemento Y.

Por ejemplo, gracias a las reglas de asociación, podría concluirse que en un supermercado, los clientes que compran leche desnatada (X), también compran cereales con fibra (Y). No obstante, el ejemplo más famoso es el indicado en [21] basado en el comportamiento de los compradores en un supermercado:

Se descubrió que muchos hombres acaban comprando pañales por encargo de sus esposas. En la cadena de supermercados Wal-Mart, donde se descubrió este hecho, se adoptó la medida de colocar la cerveza junto a los pañales. De esta manera consiguió aumentar la venta de cerveza.

Las reglas de asociación se utilizan para descubrir hechos comunes dentro de un determinado conjunto de datos.

CONFIANZA Y SOPORTE

Dados los conjuntos de elementos *X* e *Y*, y la base de datos binaria *r*, la **confianza**, representada como *conf(X => Y, r)*, es la probabilidad condicional de que una línea elegida aleatoriamente dentro de r que coincida con *X*, también coincida con *Y*. Es decir, la confianza refiere a los casos que una regla predice correctamente.

Por otra parte, el **soporte** (o frecuencia) refiere a los casos que cubre una regla. Habitualmente es representado como *sop(X, r)* siendo *X* un conjunto de elementos y *r* una base de datos binaria. Por lo general, las reglas de asociación que interesan son las que tienen un valor de soporte muy alto.

APLICACIÓN EN EL PFC

En el caso del presente estudio, la aplicación de este algoritmo permitirá descubrir **reglas de asociación en los accesos a recursos** realizados por los usuarios del Campus Virtual de la UOC, registrados en los ficheros de log.

3.5 Aplicación para el procesamiento de los logs

3.5.1 Aplicación seleccionada

La aplicación seleccionada para el estudio es **Weka** [15], la cual proporciona un conjunto de algoritmos de minería de datos entre los que se encuentran aquellos seleccionados para la realización del estudio.

Esta aplicación es libremente descargable desde la siguiente dirección:

http://www.cs.waikato.ac.nz/ml/weka/index_downloading.html

CARACTERÍSTICAS

Weka es una aplicación desarrollada en Java que dispone de una colección de herramientas de visualización, preprocesado y algoritmos para análisis de datos y modelado predictivo, unidos a una interfaz gráfica de usuario para acceder fácilmente a sus funcionalidades.

Las principales características de esta herramienta son la siguientes:

- Se trata de software libre, disponible bajo licencia GNU.
- Es fácilmente portable a otras plataformas por estar completamente desarrollado en Java.
- Soporta diversas tareas de minerías de datos, principalmente: preprocesamiento de datos, clustering, clasificación, regresión, visualización, y selección.

- Posibilita la lectura de ficheros de datos en texto plano (CSV, ARFF…) o directamente de bases de datos relacionales mediante el API JDBC.
- Es extensible, por lo que pueden incorporarse nuevos algoritmos.

ALGORITMOS

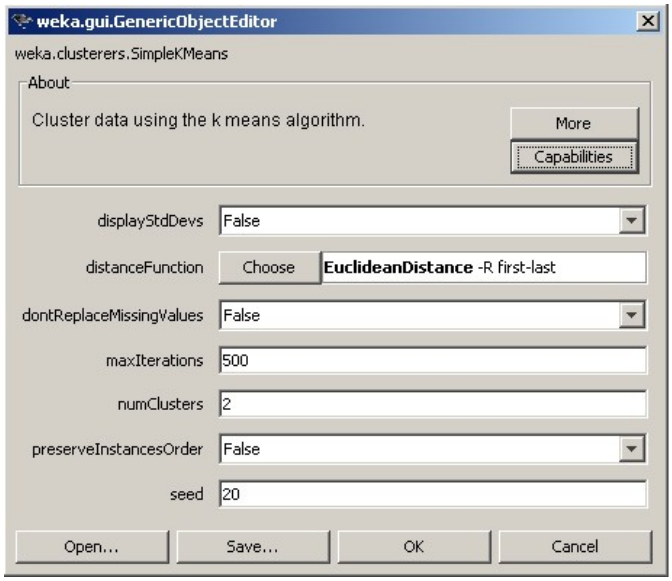
Retomando lo indicado en el apartado anterior, los métodos seleccionados para el presente estudio han sido K-Means y Apriori; a continuación se detallará el enfoque de ambos desde el punto de vista de la aplicación Weka.

3.5.2 K-Means

Weka implementa el método K-Means a través de un algoritmo denominado SimpleKMeans implementado en la clase `weka.clusterers.SimpleKMeans`. La ejecución de esta algoritmo es parametrizable a través de una serie de opciones donde las más relevantes son las siguientes:

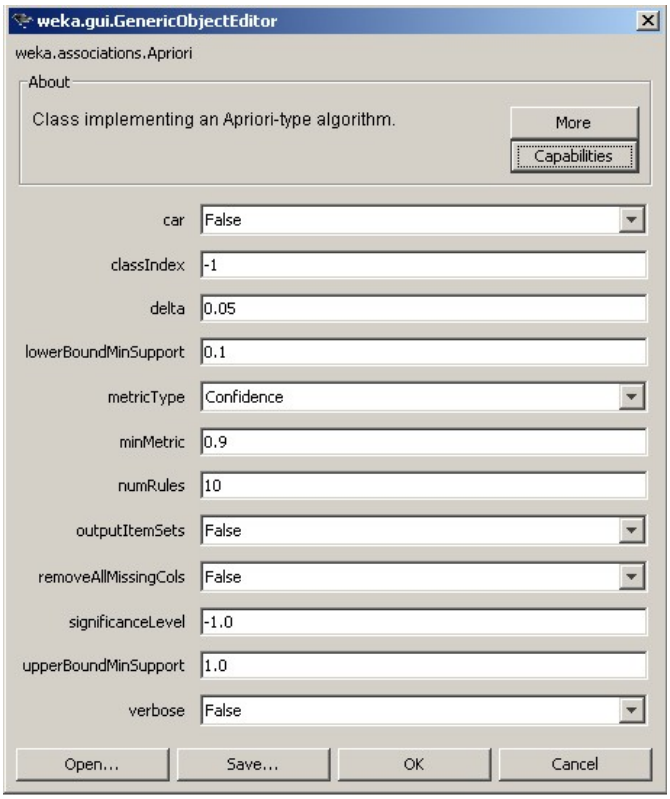
Parámetro	Descripción
numClusters	Total de agrupaciones (clusters) a obtener.
seed	Valor de la semilla a partir de la cual se generará el número aleatorio que inicializará los centros de los clusters..

La pantalla de configuración del algoritmo en Weka tiene un aspecto similar al siguiente:



3.5.3 Apriori

Weka implementa el algoritmo de asociación Apriori a través de la clase `weka.associations.Apriori`. La ejecución de esta algoritmo es parametrizable a través de una serie de opciones que pueden visualizarse en la siguiente imagen:



4 Especificación del proceso de análisis de patrones de navegación

Recuérdese que el objetivo de este PFC es la especificación de un proceso para el análisis de patrones de navegación de los estudiantes del Campus Virtual de la UOC a partir de los ficheros de log generados por el servidor; a lo largo de este apartado, se especificarán las actividades que se realizarán para lograr dicho objetivo.

En un primer nivel de detalle, este proceso puede representarse a través del siguiente **diagrama** de flujo:

4.1.1 Primera actividad. Determinación del conjunto de datos a tratar

La ejecución de esta primera actividad del proceso, permitirá **resolver el primero de los dos problemas planteados** en el apartado 3.1 de esta memoria (del segundo se ocupa la siguiente actividad del proceso), concretamente:

- **Tamaño de los ficheros de log**

  Hablamos de ficheros que contienen miles de líneas y que por lo tanto requieren considerable capacidad de cálculo para ser leídos y procesados. Sirva como dato, que los dos ficheros de log proporcionados por el Consultor tienen un tamaño que ronda los 2 y 4 GB respectivamente y que estos ficheros únicamente registran la actividad de un día del Campus Virtual de la UOC.

Para resolver este problema, el estudio tomará como muestra una parte de uno de los dos ficheros de log diarios proporcionados por el Consultor; no obstante, es importante señalar que el proceso especificado es extrapolable a cualquier conjunto de datos independientemente de su tamaño.

4.1.2 Segunda actividad. Procesamiento inicial de los ficheros de log

Para poder analizar los patrones de navegación, los logs deben procesados aplicando algoritmos de minería de datos, pero antes es necesario que sean preprocesados [11, 18] con una doble intención:

- Eliminar la información que no interesa para el estudio.
- Estructurar la información de una forma que pueda ser interpretada por la aplicación de minería de datos seleccionada.

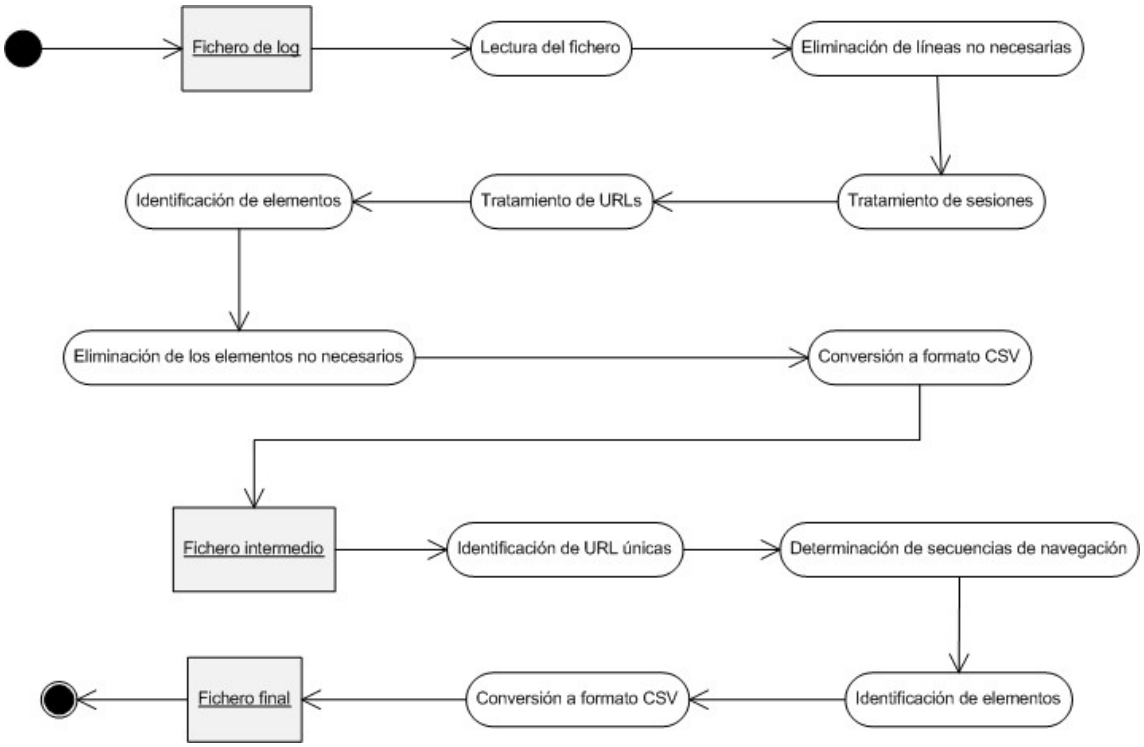
4.1.2.1 Descripción del flujo de trabajo

La ejecución de esta parte del proceso producirá como salida **dos ficheros**:

- A partir del fichero de log original, un **fichero intermedio** que conservará la estructura del fichero log original; es decir, que únicamente habrá sido tratado para eliminar información superflua y normalizar los contenidos del fichero de log original.

  Se ha considerado necesario mantener un fichero intermedio con el fin de poder tomarlo como punto de partida en caso de querer realizar otros estudios futuros, ajenos al alcance de este PFC.
- A partir del fichero intermedio, un **fichero preparado para ser interpretado y procesado por la aplicación de minería de datos**. Este fichero no respetará la estructura del fichero de log original sino que estará convenientemente reestructurado para que puedan aplicarse los algoritmos seleccionados para el presente estudio.

Para lograr esto, se ejecutará el siguiente flujo de trabajo:



OBTENCIÓN DEL FICHERO INTERMEDIO

1. **Lectura del fichero de log**

   A partir del fichero de log inicial, se identificarán y separarán los elementos de cada una de sus líneas (*ver apartado 3.2.1*) teniendo en cuenta que, dada la estructura de los ficheros log del Campus Virtual de la UOC, la identificación no será directa.

   A la hora de realizar la identificación hay que tener presente que pueden existir **espacios que no separen elementos del fichero de log** sino que formen parte de los mismos. Para ver más claro lo que aquí se indica, veamos la siguiente línea de un fichero de log:

   ```
   [13/Mar/2012:00:15:42@+0100] xxx.xxx.xxx.xxx "POST /tren/trenacc
   HTTP/1.1" 200 "https://cv.uoc.edu/tren/trenacc"
   "Mozilla/5.0@(Windows@NT@6.1;@WOW64)@AppleWebKit/535.11@(KHTML,@like@Gec
   ko)@Chrome/17.0.963.78@Safari/535.11" 14943 157
   ```

   En el ejemplo anterior pueden verse varios caracteres `@`, estos caracteres se han incluido manualmente para distinguir visualmente aquellos espacios que forman parte de un elemento; estos espacios no deben sustituirse por comas sino respetarse.

2. **Eliminación de líneas no necesarias**

   Los **ficheros** de log están **formados por miles de líneas**, pero **no todas ellas van a interesar para el análisis de patrones de navegación**. Con el fin de reducir el volumen de datos a procesar, durante este paso del flujo de trabajo **se eliminarán las líneas que**:

   a. Registren operaciones no realizadas con el **protocolo HTTP**.

   b. Refieran a accesos a:

      - **Ficheros de imagen** (JPG, GIF, PNG y BMP)
      - **Iconos** (ICO)
      - Ficheros **JavaScript** (JS)
      - **Hojas de estilo** CSS
      - Películas **Flash** (SWF)
      - Ficheros con extensión 'text'.

   c. No refieran a operaciones realizadas por estudiantes del Campus Virtual, por tratarse generalmente de operaciones realizadas por **sistemas automatizados**. Estas líneas pueden distinguirse porque el elemento 'navegador' toma como valor un guión.

   d. Para realizar el procesamiento de los logs, interesará determinar los accesos realizados durante una sesión de trabajo. Los identificadores de sesión pueden distinguirse por ir precedidos por los caracteres `?s=` o `&s=` en función de si el parámetro `s` es el primero o no respectivamente en una determinada URL.

      Aunque lo deseable sería descartar todas aquellas **líneas** en las que el elemento 'ruta recurso solicitado' o 'ruta origen' **no incluya una referencia al identificador de sesión**, el análisis inicial de los ficheros de log ha desaconsejado hacer esto ya que implicaría la pérdida de excesivas líneas. En cualquier caso, no se ha eliminado la referencia a esta actividad para tenerla presente en futuros estudios.

   e. Incluyan un **código de respuesta que señale un error** en el acceso al recurso, esto es, todas aquellas líneas cuyo código de respuesta no empiece por 2 (rango 200 a 299).

   f. Refieran a una **dirección IP para la cual se han detectado valores anómalos**.

3. **Tratamiento de sesiones**

   En el momento de realizar el procesamiento de los logs, será muy importante distinguir las sesiones de trabajo de los usuarios cuyas operaciones se han registrado en los log. Por ello, en este paso del preprocesamiento se extraerá a una nueva columna el identificador de sesión que figura en el elemento 'ruta recurso solicitado'. Recordar que el identificador de sesión puede distinguirse por ir precedido por los caracteres `?s=` o `&s=` en función de si el parámetro `s` es el primero o no en una determinada URL.

4. **Tratamiento de URLs**

   En este paso del preprocesamiento, se eliminarán de las URL registradas en el log todos los parámetros que acompañan al recurso accedido; es decir, la secuencia de caracteres que sigue al carácter '?' (incluyendo a este último). En este paso del flujo de trabajo, también hay que tener presentes otros casos menos numerosos, como cuando en vez de el carácter '?' es el carácter ';' el que actúa de separador.

5. **Eliminación de los elementos no necesarios**

   A efectos del estudio a realizar, de todos los elementos que forman parte de una línea y que se han descrito anteriormente en este documento, únicamente se considerarán los siguientes:

   - dirección IP
   - ruta recurso solicitado
   - sesión (elemento creado en el paso anterior)

   El resto de elementos serán eliminados en este paso del preprocesamiento del fichero.

   Sirva como aclaración que, aunque en estudios más amplios podrían resultar de gran interés, dado el alcance del presente estudio y de que los ficheros de log que se van a tratar son diarios, no se va a considerar el elemento que refiere a la fecha y hora de registro.

6. **Identificación de elementos**

   Este paso del preprocesamiento del fichero consistirá en la inclusión de una cabecera que identifique cada uno de los elementos que forman una línea del fichero log. Dados los elementos que se han mantenido en el paso anterior, los valores de la cabecera serán los siguientes:

   - IP
   - RECURSO
   - SESION

7. **Conversión a formato CSV** (Comma Separated Values)

   Como se ha indicado anteriormente, cada una de las líneas de los ficheros de log está formada por elementos separados por espacios; este formato no es soportado por Weka que, entre otros formatos, necesita que se le proporcionen ficheros en formato CSV (formado por elementos separados por comas).

8. **Creación del fichero intermedio**

   Llegados a este punto del flujo de trabajo, se generará el fichero intermedio que contendrá únicamente la información de interés para el estudio.

   Por ejemplo, un posible fichero podría ser el siguiente:

```
IP,RECURSO,SESION
111.111.111.111,uoc/recurso1,1111111111a
```

```
111.111.111.111,uoc/recurso2,1111111111a
222.222.222.222,uoc/recurso1,2222222222a
333.333.333.333,uoc/recurso1,3333333333a
333.333.333.333,uoc/recurso3,3333333333a
444.444.444.444,uoc/recurso3,4444444444a
555.555.555.555,uoc/recurso1,5555555555a
555.555.555.555,uoc/recurso2,5555555555a
555.555.555.555,uoc/recurso3,5555555555a
111.111.111.111,uoc/recurso1,1111111111b
```

OBTENCIÓN DEL FICHERO FINAL

Para poder analizar los patrones fijados como parte del alcance del presente PFC, **el fichero a procesar** por la aplicación de minería de datos **debe tener una estructura muy concreta**. A diferencia del fichero de log original, **cada una de las líneas** no corresponderá con el registro del acceso a un recurso realizado por un usuario, sino que **representará** la **secuencia de accesos realizados por** dicho **usuario a lo largo de una sesión de trabajo**. Para obtener este nuevo fichero, será necesario realizar lo siguiente:

9. **Identificación de URL únicas**

   Durante este paso se obtendrá el conjunto de URL únicas que figuran en la columna 'RECURSO' del fichero intermedio obtenido anteriormente. Dichas direcciones corresponderán a cada una de las columnas del nuevo fichero.

10. **Determinación de secuencias de navegación**

    Evidentemente, no todos los usuarios habrán accedido a todas las URL identificadas en el paso anterior, por lo que llegados a este punto habrá que determinar a cuales de ellas ha accedido cada usuario.

    Dada cada una de las columnas del nuevo fichero (recordar que cada columna corresponde a una URL), en cada una de las líneas se darán valores 'T' (true) o 'F' (false), según la dirección IP que figura en dicha línea haya accedido o no a la URL correspondiente [12].

11. **Identificación de elementos**

    Este paso de la obtención del fichero final, consistirá en la inclusión de una cabecera que identifique cada uno de los elementos que forman una línea mismo.

    Tal y como se ha indicado anteriormente, cada una de las direcciones únicas recopiladas corresponderá con una columna del nuevo fichero, pero también será necesario tener presente una columna adicional que corresponderá con la dirección IP del usuario que ha realizado la secuencia de accesos registrada.

    Así pues, dadas las N direcciones únicas identificadas en el paso anterior, la apariencia de la cabecera será similar a la siguiente:

    - IP
    - RECURSO_1
    - RECURSO _2
    - (...)

- RECURSO _N

12. **Conversión a formato CSV** (Comma Separated Values)

    Al igual que se ha realizado para el fichero intermedio, el fichero final deberá tener formato CSV (formado por elementos separados por comas) por lo que será necesario realizar una conversión previa.

13. **Creación del fichero final**

    Llegados a este punto del flujo de trabajo, se generará el fichero final en formato y tendrá una apariencia similar a la del siguiente ejemplo:

```
IP,uoc/recurso1,uoc/recurso2,uoc/recurso3
111.111.111.111,T,T,F
222.222.222.222,T,F,F
333.333.333.333,T,F,T
444.444.444.444,F,F,T
555.555.555.555,T,T,T
111.111.111.111,T,F,F
```

    Puede observarse que se distingue entre la secuencia de acciones realizada por la dirección IP 111.111.111.111 durante una sesión inicial de trabajo, de la secuencia de acciones realizada por la misma dirección IP en una sesión de trabajo diferente.

4.1.2.2 Implementación del flujo de trabajo

La aplicación de minería de datos seleccionada (Weka) dispone de herramientas para el preprocesado de conjuntos de datos, pero no implementa los mecanismos necesarios para el presente estudio. Por ello, el preprocesamiento descrito en el apartado anterior se realizará con una **aplicación desarrollada en Java como parte del presente PFC**, que recibirá como entrada uno de los ficheros log del Campus Virtual y dará como salidas los dos ficheros indicados en el apartado anterior.

Esta aplicación será denominada **Web Access Logs Preprocessing Tool** (WALPO) y estará construida en forma de cliente tradicional de escritorio utilizando la librería Swing de Java.

En el **apartado 5** de este documento se incluye la **especificación inicial** de esta aplicación.

4.1.3 Tercera actividad. Procesamiento del fichero obtenido

El último paso a realizar, previo al análisis de los patrones de navegación, consiste en el procesamiento del fichero obtenido tras la ejecución de la aplicación de preprocesamiento (WALPO). Como se ha indicado previamente, la aplicación seleccionada para ello ha sido Weka y, como parte de la misma, los siguientes algoritmos:

- SimpleKMeans
- Apriori

Durante la explicación de ambos algoritmos en el apartado 3.5, se ha podido ver cómo, a través de Weka, se pueden realizar diferentes parametrizaciones de los mismos, con el fin de adecuar los resultados obtenidos. Esta fase del proceso contemplará la realización de variaciones de dichos parámetros con el fin de obtener diversos resultados de la ejecución del mismo algoritmo, que puedan ser contrastados.

Los **resultados** obtenidos de la ejecución de los algoritmos SimpleKMeans y Apriori, permitirán determinar los recursos comúnmente accedidos y las reglas de navegación más habituales respectivamente, es decir, los **patrones de navegación** previstos e indicados en el apartado 3.3 del presente documento.

4.1.4 Cuarta actividad. Análisis de resultados

Finalmente, este último paso del proceso, posibilitará el **análisis** de los resultados producidos por la ejecución de Weka y por lo tanto el análisis **de los patrones de navegación obtenidos**, objetivo principal del presente proyecto.

Con la finalidad de probar la viabilidad del proceso especificado, antes de iniciar el trabajo con datos reales cuyos resultados serán indicados en la memoria final del PFC, **se ha realizado una 'prueba piloto'** a partir de un fichero de prueba. La explicación del desarrollo de la prueba piloto y los resultados obtenidos se han detallado a continuación, en el **apartado 6** del presente documento.

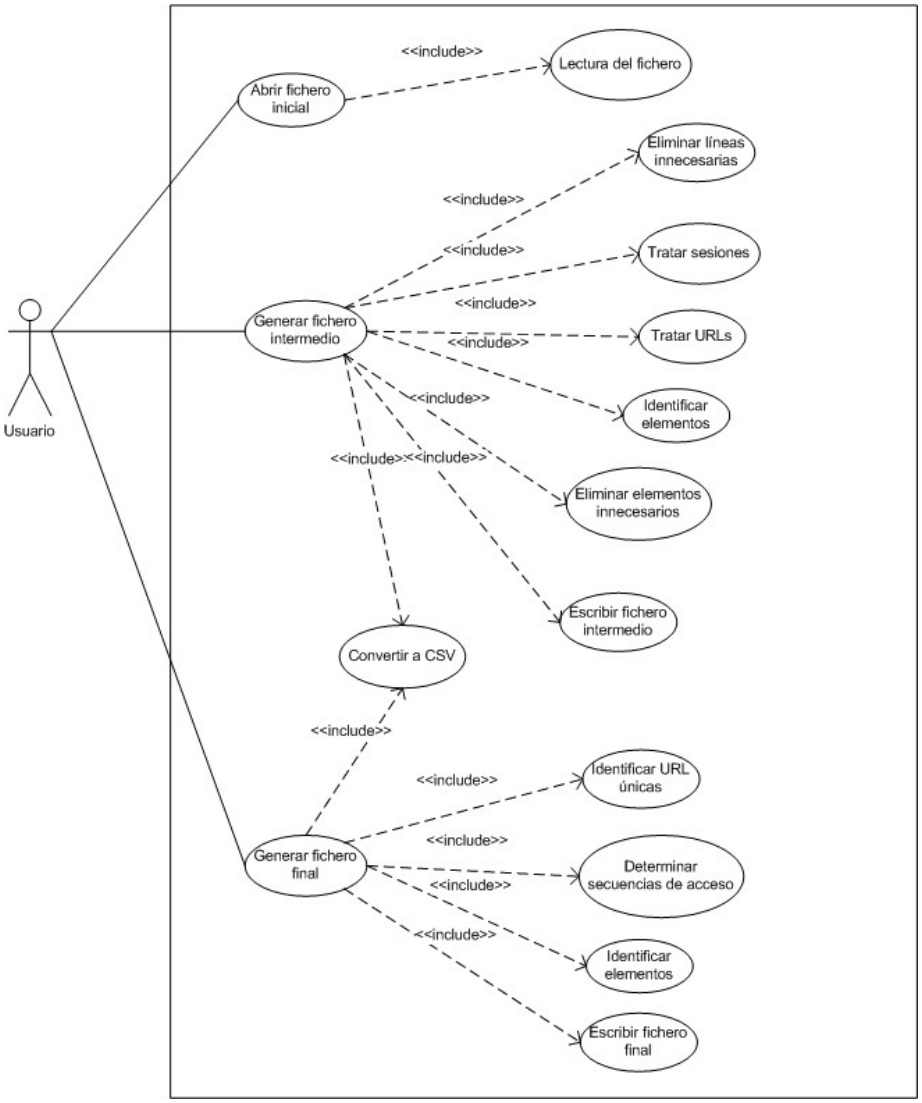
5 Estudio funcional de la aplicación

Como se ha indicado anteriormente, el flujo de trabajo que posibilitará la obtención de un fichero que pueda ser procesado por la aplicación de minería de datos, se implementará con una aplicación desarrollada en Java; esta aplicación será denominada Web Access Logs Preprocessing Tool (WALPO). En el presente apartado se realizará la especificación inicial de esta aplicación.

5.1 Casos de Uso

Dada su intencionalidad, la aplicación a construir únicamente considerará un actor que denominaremos 'Usuario', este actor será el responsable de la ejecución de los Casos de Uso principales.

El diagrama de Casos de uso de la aplicación es el que se presenta a continuación:



5.1.1 Especificación textual

Como habrá podido observarse, el diagrama anterior coincide casi exactamente con el flujo de trabajo descrito con anterioridad en el presente documento por lo que podría decirse que cada Caso de Uso corresponde a uno los pasos del proceso. En este apartado, se detallará la especificación de cada uno de los Casos de Uso representados en dicho diagrama.

Debe tenerse presente que en las tablas que se presentarán a continuación se refiere constantemente a 'elementos' o 'líneas' que 'no tienen valor para el estudio'; para facilitar la lectura de dichas tablas, y dado que los criterios que hacen que algo 'no tenga valor para el estudio' se han detallado en profundidad en el flujo de trabajo representado anteriormente, se ha creído conveniente no referir en detalle a dichos criterios.

Abrir fichero inicial	
Identificador	CU-001
Descripción	Apertura de un fichero de log del Campus Virtual de la UOC.
Precondiciones	El fichero debe tener un formato válido y extensión .log.
Secuencia normal	1. El actor demanda la apertura de un fichero 2. El sistema muestra una ventana para posibilitar la búsqueda del fichero 3. El actor localiza el fichero y solicita su apertura 4. El sistema verifica la validez del fichero 5. El sistema recorre el fichero y almacena su contenido en memoria: ejecución de CU-002
Secuencia alternativa	1. Si el fichero no tiene un formato válido, este no se abrirá notificando al actor la incidencia.
Postcondiciones	Se dispondrá del contenido del fichero de log almacenado en memoria.

Lectura del fichero inicial	
Identificador	CU-002
Descripción	Recorrer el fichero de log abierto y tratar sus líneas.
Precondiciones	Se ha debido desencadenar el proceso mediante la ejecución de CU-001.
Secuencia normal	1. Determinar la línea a tratar. 2. Procesar la línea separando sus elementos y almacenándolos en una estructura normalizada en memoria. 3. Repetir desde el paso 1 hasta alcanzar la última línea del fichero.
Secuencia alternativa	1. En caso de error en alguno de los pasos del flujo de trabajo, se notificará al actor la incidencia. 2. Si alguna de las líneas tiene un formato no válido, esta queda descartada y no se almacena en memoria.
Postcondiciones	Se dispondrá de una estructura normalizada almacenada en memoria con el

	contenido del fichero de log.

Generar fichero intermedio	
Identificador	CU-003
Descripción	Proceso de obtención del fichero intermedio recorriendo cada una de las líneas del fichero original almacenadas en una estructura normalizada en memoria y realizando su tratamiento.
Precondiciones	Se ha debido ejecutar correctamente CU-002 y por tanto debe existir una estructura normalizada en memoria con el contenido del fichero de log inicial.
Secuencia normal	1. Determinar la línea a tratar 2. Eliminación de líneas innecesarias: ejecución de CU004 3. Tratamiento de sesiones: ejecución de CU005 4. Tratamiento de URLs: ejecución de CU006 5. Eliminación de elementos innecesarios: ejecución de CU007 6. Repetir desde el paso 1 hasta alcanzar y procesar la última línea del fichero 7. Identificación de elementos: ejecución de CU008 8. Conversión a formato CSV: ejecución de CU009 9. Escritura del fichero intermedio: ejecución de CU010
Secuencia alternativa	1. En caso de error en alguno de los pasos del flujo de trabajo, se notificará al actor la incidencia.
Postcondiciones	Se dispondrá de un fichero intermedio con estructura y contenidos normalizados.

Eliminar líneas innecesarias	
Identificador	CU-004
Descripción	Eliminación de todas aquellas líneas que no tienen valor para el estudio.
Precondiciones	1. Se ha debido desencadenar el proceso mediante la ejecución de CU-003 2. Debe haberse seleccionado la línea a tratar
Secuencia normal	1. Recorrer cada uno de los elementos de la línea verificando si el elemento tratado cumple las condiciones necesarias para mantenerlo. a. En caso de encontrar un elemento que no cumple las condiciones necesarias (*ver apartado 4.1.2.1*), descartar la línea completa y devolver el control a CU-003 evitando la ejecución de todos los pasos hasta el 6 b. En caso de que todos los elementos cumplan las condiciones

	necesarias, mantener la línea
Secuencia alternativa	1. En caso de error en alguno de los pasos del flujo de trabajo, se notificará al actor la incidencia.

Tratar sesiones	
Identificador	CU-005
Descripción	Determinar el identificador de sesión que aparece en cada una de las líneas tratadas.
Precondiciones	1. Se ha ejecutado correctamente CU-004 2. Debe haberse seleccionado la línea a tratar
Secuencia normal	1. Buscar en el elemento 'ruta recurso solicitado' el identificador de sesión. En caso de no encontrarlo, buscarlo en el elemento 'ruta origen' 2. Crear un nuevo elemento en memoria que contenga el identificador de sesión
Secuencia alternativa	1. En caso de error en alguno de los pasos del flujo de trabajo, se notificará al actor la incidencia.

Tratar URLs	
Identificador	CU-006
Descripción	Dados los elementos que contienen URLs, tratarlos para eliminar la información no relevante para el estudio.
Precondiciones	1. Se ha ejecutado correctamente CU-005 2. Debe haberse seleccionado la línea a tratar
Secuencia normal	1. Buscar en el elemento 'ruta recurso solicitado' 2. Tratar el contenido del elemento según las reglas definidas en el flujo de trabajo
Secuencia alternativa	1. En caso de error en alguno de los pasos del flujo de trabajo, se notificará al actor la incidencia.

Eliminar elementos innecesarios	
Identificador	CU-007
Descripción	Para cada línea, eliminación de todos aquellos elementos que no tienen valor para el estudio.
Precondiciones	1. Se ha ejecutado correctamente CU-006 2. Debe haberse seleccionado la línea a tratar
Secuencia normal	1. Determinar el elemento a tratar 2. Eliminar el elemento si pertenece al conjunto de los considerados innecesarios

	3. Repetir el paso 1 hasta alcanzar y procesar el último elemento de la línea
Secuencia alternativa	1. En caso de error en alguno de los pasos del flujo de trabajo, se notificará al actor la incidencia.

Identificar elementos	
Identificador	CU-008
Descripción	Identificar los elementos que aparecerán en el fichero intermedio definiendo una cabecera identificativa.
Precondiciones	Se ha ejecutado correctamente la primera parte de CU-003
Secuencia normal	1. Generar la cabecera incluyendo la denominación de cada columna y separando dichas columnas por comas (recordar que el fichero intermedio tendrá formato CSV)
Secuencia alternativa	1. En caso de error en alguno de los pasos del flujo de trabajo, se notificará al actor la incidencia.

Convertir a CSV	
Identificador	CU-009
Descripción	Sustitución, para cada línea del fichero, de espacios por comas.
Precondiciones	Se ha finalizado correctamente el tratamiento del contenido del fichero correspondiente (intermedio o final).
Secuencia normal	1. Determinar la línea a tratar 2. Construir una estructura separando con comas los elementos que deban figurar en el fichero a generar 3. Repetir desde el paso 1 hasta alcanzar la última línea
Secuencia alternativa	1. En caso de error en alguno de los pasos del flujo de trabajo, se notificará al actor la incidencia.

Escribir fichero intermedio	
Identificador	CU-010
Descripción	Escritura en el fichero intermedio.
Precondiciones	Se ha ejecutado correctamente CU-009
Secuencia normal	1. Escribir en el fichero intermedio los valores especificados
Secuencia alternativa	1. En caso de error en alguno de los pasos del flujo de trabajo, se notificará al actor la incidencia.

Generar fichero final	
Identificador	CU-011

Descripción	Proceso de obtención del fichero final recorriendo cada una de las líneas del fichero intermedio y realizando su tratamiento.
Precondiciones	Se ha debido generar correctamente el fichero intermedio.
Secuencia normal	1. Abrir el fichero intermedio 2. Identificar las URL únicas: ejecución de CU012 3. Determinar las secuencias de navegación: ejecución de CU13 4. Identificación de elementos: ejecución de CU014 5. Conversión a formato CSV: ejecución de CU009 6. Escritura del fichero intermedio: ejecución de CU015
Secuencia alternativa	1. En caso de error en alguno de los pasos del flujo de trabajo, se notificará al actor la incidencia.
Postcondiciones	Se dispondrá del fichero final listo para ser procesado por la aplicación de minería de datos aplicando los algoritmos seleccionados.

Identificar URL únicas	
Identificador	CU-012
Descripción	Dadas las URL presentes en el fichero intermedio, recorrerlo para almacenar en memoria todas ellas una única vez (sin repeticiones).
Precondiciones	Se ha debido desencadenar el proceso mediante la ejecución de CU-011
Secuencia normal	1. Determinar la línea a tratar 2. Determinar si la línea contiene una URL ya tratada. Si la URL no ha sido tratada, almacenarla. 3. Repetir desde el paso 1 hasta alcanzar la última línea del fichero.
Secuencia alternativa	1. En caso de error en alguno de los pasos del flujo de trabajo, se notificará al actor la incidencia.
Postcondiciones	Se dispondrá en memoria de una lista con la URL únicas encontradas.

Determinar secuencias de navegación	
Identificador	CU-013
Descripción	Identificar las URL a las que ha accedido un usuario durante una sesión de trabajo.
Precondiciones	Se ha ejecutado correctamente CU-012
Secuencia normal	1. Determinar la línea a tratar 2. Identificar dirección IP y sesión. Si la pareja es nueva, inicializar una nueva estructura en memoria que representará a una línea en el fichero final. Esta estructura tendrá tantas posiciones como direcciones únicas hayan sido identificadas en CU-012 más una posición adicional para almacenar la dirección IP

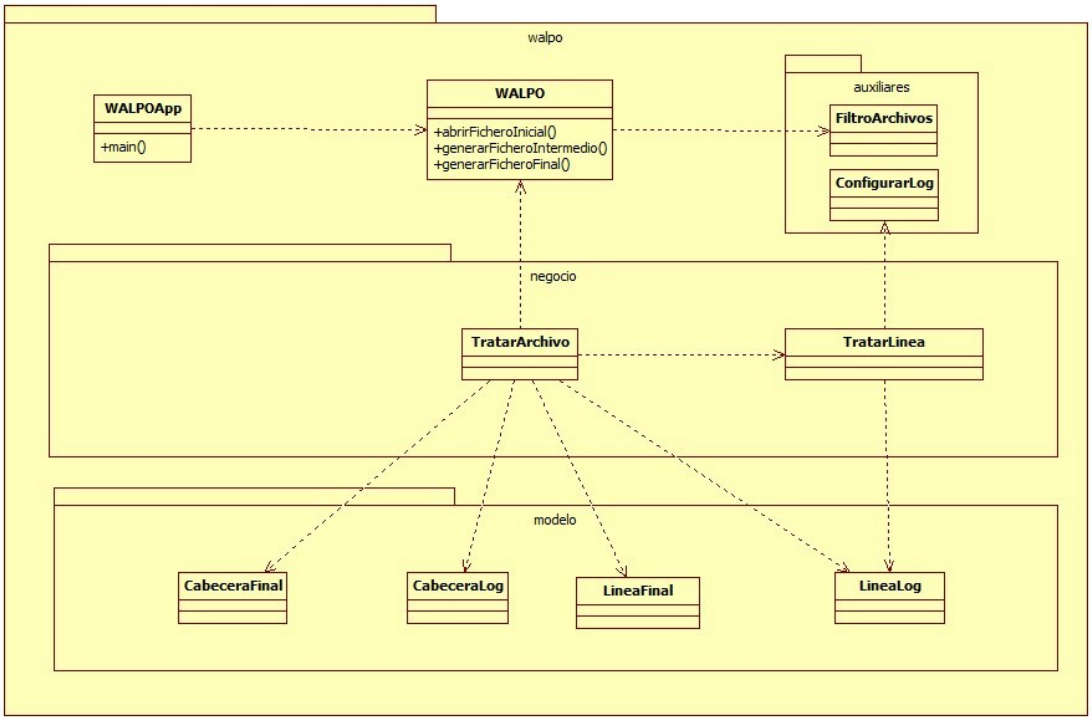
	3. Recorrer la lista de direcciones únicas generada en CU-012 comparando cada una de ellas con la URL indicada en la línea tratada. Si hay correspondencia dar valor 'T' (true) y si se llega al final sin correspondencia, dar valor 'F' (false) en la posición correspondiente a dicha URL 4. Repetir desde el paso 1 hasta alcanzar la última línea del fichero intermedio
Secuencia alternativa	1. En caso de error en alguno de los pasos del flujo de trabajo, se notificará al actor la incidencia.

Identificar elementos	
Identificador	CU-014
Descripción	Identificar los elementos del fichero final ubicando una cabecera identificativa al principio del mismo.
Precondiciones	Se ha ejecutado correctamente CU-013
Secuencia normal	1. Incluir el valor para la columna que identificará la dirección IP 2. Recorrer la lista de direcciones únicas generada en CU-012 incluyéndolas en la cabecera separadas por comas (recordar que el fichero intermedio tendrá formato CSV)
Secuencia alternativa	1. En caso de error en alguno de los pasos del flujo de trabajo, se notificará al actor la incidencia.

Escribir fichero final	
Identificador	CU-015
Descripción	Escritura en el fichero intermedio.
Precondiciones	Se ha ejecutado correctamente CU-009
Secuencia normal	1. Escribir en el fichero final los valores especificados
Secuencia alternativa	1. En caso de error en alguno de los pasos del flujo de trabajo, se notificará al actor la incidencia.

5.2 Diagrama de clases

A continuación se presenta el diagrama de clases de aplicación incluyendo los métodos más relevantes:



La descripción de cada una de las clases es la siguiente:

Clase	Descripción
WALPOApp	Clase principal que posibilita ejecutar la aplicación.
WALPO	Formulario que implementa la capa de presentación de la aplicación y los eventos que permitirán interactuar con la misma.
FiltroArchivos	Clase que implementa la lógica para determinar qué tipos de archivos de log son admitidos por la aplicación.
ConfigurarLog	Clase para fijar el nivel tanto del log indicado como de la consola. Útil durante la fase de desarrollo puesto que deshabilitar la salida de log reduce el tiempo de ejecución de la aplicación al realizarse menos registros.
TratarArchivo	Clase para la generación y tratamiento de los archivos intermedio y final.
TratarLinea	Clase que contiene la lógica para el tratamiento de las líneas del fichero de log inicial y las reglas de negocio que posibilitan comprobar su validez.
LineaLog	Clase para la persistencia de una línea del fichero de log inicial. Cada uno de sus atributos corresponde con los elementos que forman una línea.
CabeceraLog	Clase para la persistencia de la cabecera identificativa del fichero intermedio.

LineaFinal	Clase para la persistencia de una línea del fichero final.
CabeceraFinal	Clase para la persistencia de la cabecera identificativa del fichero final.

5.3 Implementación de la aplicación

5.3.1 Entorno de desarrollo

El entorno de desarrollo utilizado para la implementación de la aplicación ha sido NetBeans 6.9 junto con el JDK de Java en su versión 1.6.

5.3.2 Expresiones regulares

El principal punto crítico para la implementación de la aplicación es la necesidad de tratar líneas con estructura homogénea pero contenido heterogéneo una forma rápida y sencilla.

Al referir a 'contenido heterogéneo' se quiere decir que una determinada una línea puede contener caracteres que perturben y dificulten la identificación y separación de los elementos que la componen así como realizar la interpretación del contenido de cada elemento con el fin de evaluarlo tal y como se ha descrito en apartados anteriores de este documento.

Existen dos formas de afrontar el problema:

- Mediante un uso intensivo de código Java que tenga presente la diversidad de contenidos que se pueden dar.
- Mediante expresiones regulares [26].

Aunque su aprendizaje puede llegar a suponer algún problema, **el uso de expresiones regulares permite simplificar enormemente la implementación** de una aplicación.

A modo de ejemplo, veamos el algoritmo (con una pequeña parte en Java) que posibilita la identificación y separación de elementos de una línea de log gracias a la expresión regular `formatoLinea`:

```
public getLineaLog(String linea){

      String formatoLinea = "\\[([\\w:/]+\\s[+\\-]\\d{4})\\]* ([\\d.]+) \"(.+)
      (/.+) (.+)\" (\\d{3}) \"(.+)\" \"(.+)\" (\\d+) (\\d+)";

      Pattern p = Pattern.compile(formatoLinea);
      Matcher matcher = p.matcher(linea);

      //En caso de que la línea no tenga el formato esperado, se descarta
      if (!matcher.matches) { <Mostrar mensaje de error> }

      //Crear el objeto que almacenará la línea
```

```
        <Crear objeto con atributos para almacenar cada elemento de la línea>
}
```

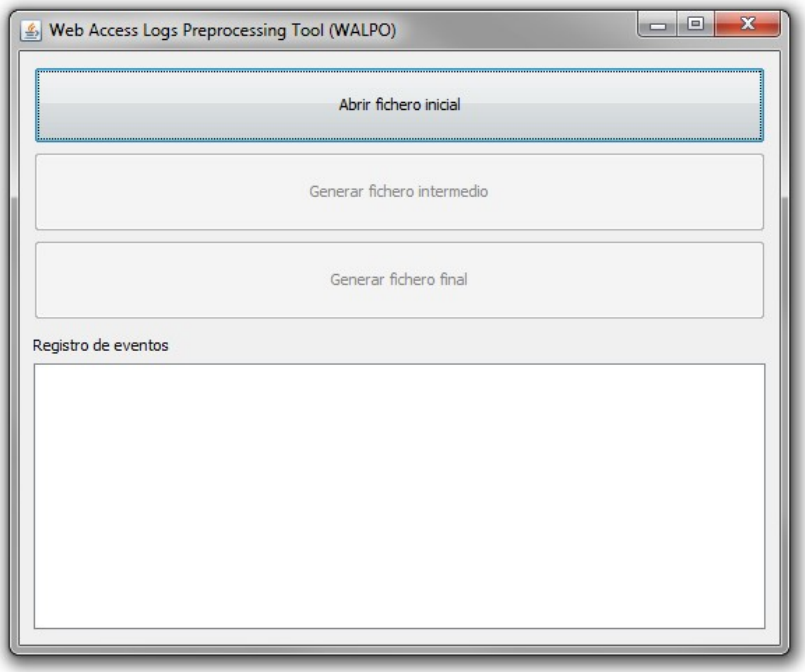
La expresión regular `formatoLinea` posibilita separar en diferentes atributos de una clase Java, cada uno de los 10 elementos que estructuran las líneas de los ficheros de log del Campus Virtual de la UOC:

```
[<fecha y hora registro>] <dirección IP> "<método petición> <ruta recurso solicitado> <protocolo>" <código respuesta> "<ruta origen>" "<navegador>" <n.º bytes obtenido> <p1>
```

Como puede observarse, con unas pocas líneas de código se ha resuelto un problema que de otra forma hubiera implicado la construcción de un método mucho más complejo.

5.3.3 Interfaz de usuario

La aplicación que posibilitará el preprocesamiento de los logs será una aplicación de 'escritorio' construida con la biblioteca Swing de Java, cuya apariencia será similar a la siguiente:



Como puede observarse, la interfaz de usuario prevista es extremadamente sencilla y dispondrá de los siguientes controles:

Control	Tipo	Descripción
Abrir fichero inicial	Botón	Abrirá una ventana de diálogo que posibilitará la localización y apertura de un fichero de log del Campus Virtual de la UOC. Una vez abierto el fichero inicial, almacenará su contenido en memoria para ser procesado.

Generar fichero intermedio	Botón	Se habilitará una vez abierto el fichero inicial y ejecutará los pasos del flujo de trabajo descrito anteriormente, generando el fichero intermedio.
Generar fichero final	Botón	Se habilitará una vez generado el fichero intermedio y ejecutará los pasos del flujo de trabajo descrito anteriormente que posibilitarán la obtención del fichero final a procesar por la aplicación de minería de datos.
Registro de eventos	Área de texto	Informará del estado del proceso y de los resultados que se van produciendo durante la ejecución del flujo de trabajo.

5.4 Instalación de la aplicación

Para realizar la instalación de la aplicación bastará con disponer de una máquina virtual de Java 6 o superior instalada en el sistema y ejecutar el archivo WALPO.jar entregado junto con la presente memoria.

Dado el tamaño de los ficheros a tratar, el proceso que ejecuta la aplicación es muy exigente en cuanto a la cantidad de memoria del sistema necesaria para ejecutarlo; por ello, se recomienda fijar los parámetros de memoria de la máquina virtual de Java con al menos los siguientes valores: `-Xmx800m -Xms256m`.

Para ejecutar la aplicación de forma que tengan efecto los parámetros indicados, debe utilizarse el siguiente comando:

```
java -jar WALPO.jar -Xmx800m -Xms256m
```

Para facilitar la ejecución de la aplicación con los parámetros indicados, se entrega el fichero WALPO.bat. En caso de que el fichero de log a procesar requiera una cantidad mayor de memoria, será suficiente con modificar los parámetros `-Xmx` y `-Xms` editando el fichero bat.

CONFIGURACIÓN DE LA MEMORIA DE LA MÁQUINA VIRTUAL DE JAVA

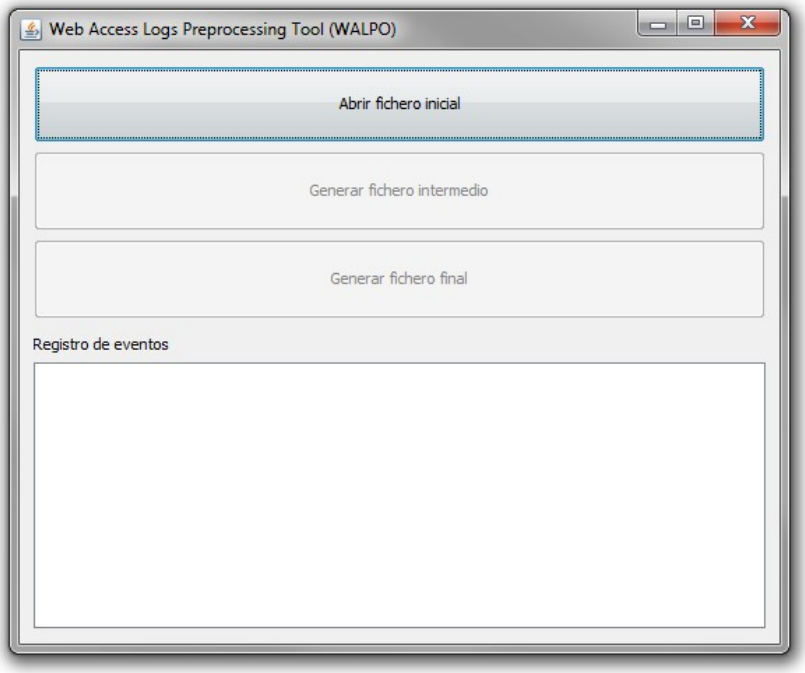
La memoria de la máquina virtual de Java puede configurarse a través de dos parámetros [29]:

- **Xms**: Indica el tamaño mínimo del heap que ha de reservar la máquina virtual.
- **Xmx**: Indica el tamaño máximo del heap.

Ambos parámetros son de vital importancia para el correcto funcionamiento de WALPO.

5.5 Manual de uso de la aplicación

La aplicación posibilita la generación de dos ficheros (uno intermedio y otro final) a partir de un fichero de log del Campus Virtual de la UOC.



El uso de la aplicación es secuencial, por lo que los botones de la misma se irán habilitando progresivamente.

En primer lugar será necesario pulsar el botón 'Abrir fichero inicial' lo que hará que se muestre una ventana para localizar un fichero de log del Campus Virtual de la UOC con extensión .log. En caso de que el fichero localizado no disponga de esta extensión, será necesario renombrarlo.

Una vez que haya sido procesado correctamente el fichero de log inicial, se habilitará el botón 'Generar fichero intermedio' el cual posibilitará generar el correspondiente fichero.

Por último, una vez el fichero intermedio se haya generado correctamente, se habilitará el botón 'Generar fichero final' para generar dicho fichero.

Tanto el fichero intermedio como el fichero final, se almacenarán en la misma ubicación que el fichero de log inicial y podrán identificarse por la terminación '_INTERMEDIO' y '_FINAL' respectivamente.

Los resultados de cada una de las acciones ejecutadas, podrán visualizarse en el registro de eventos ubicado en la parte inferior de la interfaz de usuario.

6 Prueba piloto

Como se ha indicado anteriormente, la finalidad de esta prueba piloto es verificar la viabilidad del proceso descrito antes de iniciar el trabajo con datos reales. En este apartado se detalla el desarrollo de la prueba que se realizó durante la fase de especificación y análisis del proyecto y los resultados obtenidos.

6.1 Preparación inicial

Para la realización de la prueba, se ha partido de un fichero con el mismo formato que el de los ficheros generados por la aplicación WALPO; el contenido del fichero es el siguiente:

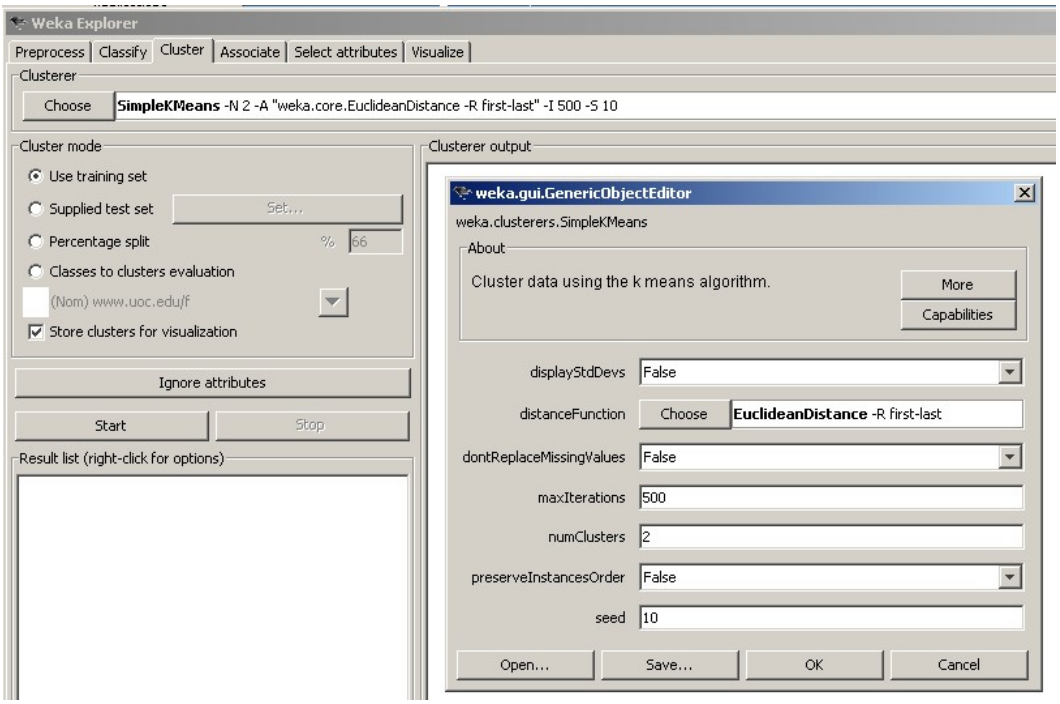
```
IP,www.uoc.edu/a,www.uoc.edu/b,www.uoc.edu/c,www.uoc.edu/d,www.uoc.edu/e,www.uoc.edu/f
111.111.111.111,T,T,T,F,F,F
222.222.222.222,F,T,F,T,T,F
333.333.333.333,F,T,T,F,F,T
444.444.444.444,F,F,F,F,T,F
555.555.555.555,T,F,T,F,T,T
111.111.111.111,T,F,T,F,F,T
```

Recordar que cada línea del fichero representa lo que se ha denominado' secuencia de navegación'; es decir, el conjunto de recursos visitados (o no), por un usuario (identificado por una dirección IP) durante una sesión de trabajo; así pues, puede verse que la dirección IP 111.111.111.111 ha realizado dos sesiones de trabajo. Puede observarse también, que los nombres de los recursos no son reales, sino que se han adecuado para que la prueba piloto sea más clara.

Este fichero se ha cargado en Weka mediante el módulo 'Weka Explorer' y sobre él, se ha realizado la ejecución de los dos algoritmos.

6.2 SimpleKMeans

En primer lugar, se ha ejecutado el algoritmo de agregación (clustering) SimpleKMenans, conservando los parámetros por defecto fijados por Weka:



Por último, antes de ejecutar el algoritmo, se descartará el atributo 'IP' por no aportar valor a los resultados que se buscan; para ello se utilizará la funcionalidad ofrecida a través del botón 'Ignore atributes', seleccionando el atributo correspondiente.

Es importante señalar que, dados estos parámetros, únicamente se generarán dos grupos (clusters).

La ejecución de este algoritmo ha producido los siguientes resultados:

```
kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 7.0
Missing values globally replaced with mean/mode

Cluster centroids:
                           Cluster#
Attribute        Full Data         0         1
                     (6)       (2)       (4)
=================================================
www.uoc.edu/a            T         F         T
www.uoc.edu/b            T         T         T
www.uoc.edu/c            T         F         T
www.uoc.edu/d            F         F         F
www.uoc.edu/e            F         T         F
www.uoc.edu/f            F         F         T


Clustered Instances

0       2 ( 33%)
1       4 ( 67%)
```

Si recordamos lo indicado anteriormente, con la ejecución del algoritmo SimpleKMeans se pretendía determinar los recursos comúnmente accedidos, veamos si se ha logrado mediante la interpretación de los resultados obtenidos.

ANÁLISIS DE RESULTADOS

Dada la tabla de resultados de la imagen anterior, donde 'T' indica que se ha accedido a un recurso y 'F' que no se ha producido tal acceso, la columna 'Full Data' representa la secuencia de navegación más habitual dado el total de secuencias que aparecen en el fichero procesado.

Antes de continuar, es importante señalar que la 'secuencia de navegación más habitual' no es la más repetida (es más, no tiene porqué existir como tal en el fichero procesado), sino que está formada por los valores más repetidos dado cada recurso. Para entenderlo mejor, veamos el resultado obtenido para dicha columna:

Recurso	Valor
www.uoc.edu/a	T
www.uoc.edu/b	T
www.uoc.edu/c	T
www.uoc.edu/d	F
www.uoc.edu/e	F
www.uoc.edu/f	F

Esto quiere decir, que dado el contenido del fichero procesado, en cada sesión de trabajo, los usuarios del Campus Virtual de la UOC suelen acceder comúnmente a los recursos: www.uoc.edu/a, www.uoc.edu/b y www.uoc.edu/c, pero no a los otros tres recursos que figuran en la tabla anterior.

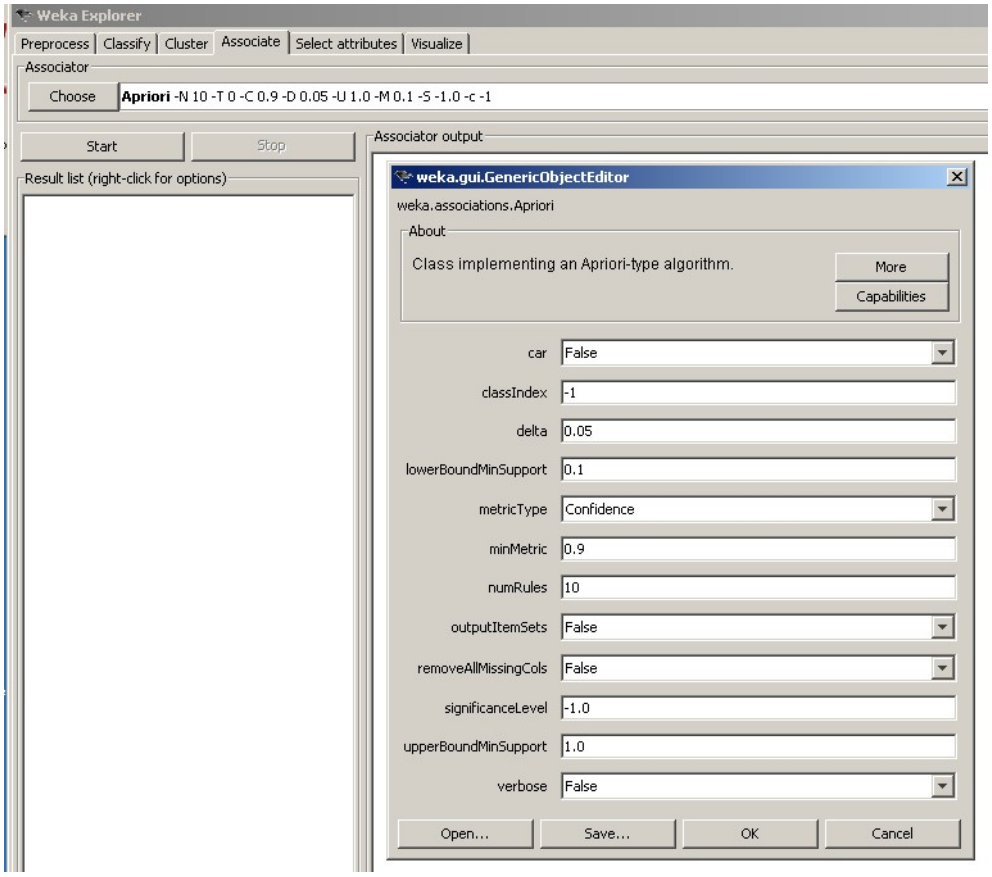
Continuando con la interpretación de los resultados obtenidos, en la imagen anterior pueden verse, por una parte, los dos grupos (clusters) generados y, por otra, lo que ha sido denominado 'Clustered Instances'. Los valores que aparecen etiquetados como 'Clustered Instances', representan las asignaciones de cada una de las secuencias de navegación que aparecen en el fichero a los grupos generados.

Así pues, dos de las seis secuencias de navegación (el 33 %) han sido asociadas al grupo '0', mientras que las otras cuatro (el 67 %) han sido asociadas al grupo '1'. Esto quiere decir, que los usuarios del Campus Virtual, en cada una de sus sesiones de trabajo, suelen acceder habitualmente a www.uoc.edu/a, www.uoc.edu/b, www.uoc.edu/c y www.uoc.edu/f, pero no al resto de recursos.

Por último, señalar que si se contrasta lo obtenido en la columna 'Full Data' y en la sección 'Clustered Instances' de los resultados, debería haber habido coincidencia en la secuencia de navegación más habitual. Esta situación se ha dado por la uniformidad de valores 'T' y 'F' en el recurso www.uoc.edu/f. Los resultados pueden mejorarse para tratar estas situaciones de 'empate', mediante la variación de los parámetros por defecto que define Weka para el algoritmo (por ejemplo incrementando el valor del parámetro 'numClusters').

6.3 Apriori

En segundo lugar, se ha ejecutado el algoritmo de asociación Apriori conservando, también en este caso,los parámetros por defecto fijados por Weka:



La ejecución de este algoritmo ha producido los siguientes resultados:

```
Apriori
=======

Minimum support: 0.55 (3 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 9

Generated sets of large itemsets:

Size of set of large itemsets L(1): 10

Size of set of large itemsets L(2): 8

Size of set of large itemsets L(3): 3

Best rules found:

 1. www.uoc.edu/c=T 4 ==> www.uoc.edu/d=F 4    conf:(1)
 2. www.uoc.edu/a=T 3 ==> www.uoc.edu/c=T 3    conf:(1)
 3. www.uoc.edu/a=T 3 ==> www.uoc.edu/d=F 3    conf:(1)
 4. www.uoc.edu/b=F 3 ==> www.uoc.edu/d=F 3    conf:(1)
 5. www.uoc.edu/e=F 3 ==> www.uoc.edu/c=T 3    conf:(1)
 6. www.uoc.edu/f=T 3 ==> www.uoc.edu/c=T 3    conf:(1)
 7. www.uoc.edu/e=F 3 ==> www.uoc.edu/d=F 3    conf:(1)
 8. www.uoc.edu/f=T 3 ==> www.uoc.edu/d=F 3    conf:(1)
 9. www.uoc.edu/a=T www.uoc.edu/d=F 3 ==> www.uoc.edu/c=T 3    conf:(1)
10. www.uoc.edu/a=T www.uoc.edu/c=T 3 ==> www.uoc.edu/d=F 3    conf:(1)
```

Si recordamos lo indicado anteriormente, con la ejecución del algoritmo Apriori se pretendían determinar las reglas de navegación más habituales, veamos si se ha logrado mediante la interpretación de los resultados obtenidos.

ANÁLISIS DE RESULTADOS

En la imagen anterior, que representa los resultados obtenidos de ejecutar el algoritmo Apriori, pueden observarse diez reglas de asociación con un valor '1' de confianza (conf). Esto quiere decir, que esas diez reglas, siempre se cumplen dado el conjunto de secuencias de navegación procesadas.

Por otra parte, puede observarse un valor numérico al lado de cada extremo de la regla; este valor numérico representa el total de secuencias de acceso del fichero procesado que cumplen dicha regla.

Por lo tanto, analizando cada una de las reglas de asociación obtenidas, puede concluirse que todos aquellos estudiantes del Campus Virtual de la UOC que en su sesión de trabajo:

1. Han accedido a www.uoc.edu/c, no han accedido a www.uoc.edu/d
2. Han accedido a www.uoc.edu/a, también han accedido a www.uoc.edu/c
3. Han accedido a www.uoc.edu/a, no han accedido a www.uoc.edu/d
4. No han accedido a www.uoc.edu/b, tampoco lo han hecho a www.uoc.edu/d
5. No han accedido a www.uoc.edu/e, sí que lo han hecho a www.uoc.edu/c
6. Han accedido a www.uoc.edu/f, también han accedido a www.uoc.edu/c
7. No han accedido a www.uoc.edu/e, tampoco lo han hecho a www.uoc.edu/d
8. No han accedido a www.uoc.edu/f, tampoco lo han hecho a www.uoc.edu/d
9. Han accedido a www.uoc.edu/a pero no a www.uoc.edu/d, han acabado accediendo a www.uoc.edu/c

10. Han accedido a www.uoc.edu/a y también a www.uoc.edu/c, no han accedido a www.uoc.edu/d

6.4 Conclusiones de la prueba

A través de los resultados obtenidos con la ejecución de la prueba, **se ha podido asegurar la viabilidad de ejecutar**, con datos reales**, el proceso especificado en el apartado 4 de este documento**, y de aplicar los dos algoritmos de minería de datos seleccionados.

Por lo tanto, tal y como estaba previsto en la planificación inicial, el siguiente paso en la ejecución del presente PFC, será aplicar el proceso especificado sobre un extracto (recordar la problemática del tamaño de los ficheros indicada al inicio del documento) de uno de los ficheros de log del Campus Virtual de la UOC proporcionados por el Consultor. El análisis de los patrones de navegación obtenidos, posibilitará determinar el comportamiento de los estudiantes del Campus Virtual de la UOC durante el intervalo de tiempo que corresponda a lo registrado en el fichero tomado como muestra..

7 Análisis de patrones de navegación

Antes de continuar, conviene recordar en qué punto nos encontramos **señalando lo realizado previamente**:

1. Análisis inicial del problema a resolver
2. Especificación del proceso para el análisis de patrones de navegación
3. Determinación de los patrones de navegación a analizar
4. Determinación de los algoritmos de minería de datos a aplicar
5. Selección de la aplicación de minería de datos a utilizar
6. Análisis, diseño e implementación de la aplicación para el preprocesado de los ficheros de log del Campus Virtual
7. Realización de la prueba piloto

Una vez finalizada con éxito la prueba piloto, se concluyó que **la preparación inicial realizada era adecuada para ejecutar el proceso especificado** con los ficheros de log reales del Campus Virtual de la UOC; **este apartado mostrará los resultados de la ejecución de dicho proceso**.

Recordar que **el proceso está formado por cuatro actividades** de primer nivel y diversas subactividades:

1. Determinación del conjunto de datos a tratar.
2. Procesamiento inicial del fichero de log del Campus Virtual de la UCO con la aplicación WALPO.
3. Procesamiento del fichero producido por WALPO con Weka, la aplicación de minería de datos seleccionada.
4. Análisis de los resultados obtenidos.

A continuación, se describe el resultado de la ejecución de cada una de las cuatro actividades del proceso; en el caso de la cuarta actividad, el análisis de los resultados se ha incluido en el apartado correspondiente a cada uno de los algoritmos de minería de datos que se han aplicado.

7.1 Determinación del conjunto de datos a tratar

Antes de proceder a su tratamiento, se determinó el conjunto de datos utilizado para el estudio, concretamente, se seleccionó el **conjunto de registros generados el día 9 de marzo de 2012 entre las 08:38 y las 09:12 horas**. Este conjunto de datos se extrajo del fichero de log *campusF5.log-20120309* proporcionado por el Consultor.

El conjunto de datos seleccionado se almacenó en un nuevo fichero denominado extracto_*campusF5.log-20120309.log*.

7.1.1 Problemas acontecidos

Debido a su tamaño, a la hora de abrir y tratar los ficheros de log del Campus Virtual, **se presentaron diversos problemas al utilizar los editores de texto más habituales**: el Bloc de Notas se bloqueaba,

WordPad presentaba serios problemas de rendimiento que hacían imposible tratar el fichero y Microsoft Word y Notepad++ no ha permitido su apertura; por ello, se buscó un editor de texto optimizado para tratar ficheros de gran tamaño: Large Text File Viewer (conocido también como LTF Viewer) [27]; sin embargo, se encontró una limitación que imposibilitaba realizar una copia de datos por encima de 16 MB.

7.1.2 Solución aplicada

Dada la imposibilidad de contar con un editor de texto que posibilitara extraer de forma ágil un determinado conjunto de datos de un fichero de log completo, **se optó por utilizar una herramienta que cortara el fichero de log en partes tratables**; la herramienta seleccionada fue GSplit [28].

Con la herramienta GSplit, se troceó el fichero de log en ocho fragmentos de 250 MB de tamaño; de estos ocho fragmentos se seleccionó el que contenía registros generados entre las 08:38 y las 09:12 horas.

Este rango horario, almacenado en un fichero de 250 MB de tamaño puede dar una idea del elevado volumen de datos almacenado en los ficheros de log del Campus Virtual y la problemática que su tamaño ha supuesto.

7.2 Procesamiento inicial del fichero de log

Una vez determinado el conjunto de datos a estudiar, se procedió a tratar el fichero extracto_*campusF5.log-20120309.log* con la aplicación WALPO obteniendo el fichero intermedio y el fichero final esperados:

- *extracto_campusF5.log-20120309_INTERMEDIO.csv*
- *extracto_campusF5.log-20120309_FINAL.csv*

PARAMETRIZACIÓN DE LA APLICACIÓN

Para la generación de ambos ficheros, WALPO se parametrizó de la siguiente manera:

Configuración inicial de WALPO	
Filtrado por número de accesos a recursos	Habilitado
Número de accesos mínimo al recurso	5
Número de accesos máximo al recurso	1.500
Número de accesos mínimo realizado por IP	5

Los parámetros anteriores corresponden a constantes definidas en el código fuente de la aplicación y entran en funcionamiento cuando se habilita el filtrado por número de accesos a recursos.

Es necesario delimitar el número máximo de accesos para eliminar el 'ruido' existente en los ficheros de log del Campus Virtual; denominamos 'ruido' a aquellos recursos con un número de accesos excesivamente alto para el conjunto de datos seleccionado. Los recursos descartados con este parámetro son aquellos que son accedidos de forma transparente al usuario (sin haberlo demandado expresamente) durante la navegación natural por el Campus Virtual (por ejemplo los recursos accedidos para realizar el acceso al Campus Virtual...) y que no aportan valor para analizar los patrones de navegación.

Aunque debe intentar evitarse, es necesario delimitar el número mínimo de accesos para optimizar el consumo de memoria de la aplicación y posibilitar el tratamiento por parte de Weka. En caso de tener que recurrir a este parámetro, hay que procurar que su valor sea lo más próximo a 0 posible.

Por último, las direcciones IP que no realizaron al menos el número de accesos fijado como parámetro, no se consideraron para el estudio.

Tras realizar pruebas con diferentes combinaciones de valores, se considera que los seleccionados son lo suficientemente equilibrados como para lograr descartar accesos sin valor para el estudio evitando la pérdida de información relevante; a la par, son valores adecuados para evitar problemas por volumen de datos durante la ejecución de los algoritmos con Weka.

7.2.1 Problemas acontecidos

Tras tratar por primera vez el fichero con el conjunto de datos seleccionado y realizar una revisión de los contenidos de los dos ficheros generados (intermedio y final), se detectaron dos direcciones IP (por privacidad no se va a referir a ellas en la presente memoria) con valores anómalos.

La anomalía detectada consistía en que dichas direcciones IP tenían un número de apariciones anormalmente alto en el fichero final, concretamente un total de 1952 y 1965 identificadores de sesión distintos; es decir, que en el intervalo de tiempo analizado, habían iniciado sesión en el Campus Virtual el número de veces indicado.

7.2.2 Solución aplicada

La detección de esta anomalía implicó la realización de una adecuación en el proceso inicialmente implementado en la aplicación WALPO para implementar un filtrado de direcciones IP y así descartarlas en los ficheros intermedio y final.

El proceso implementado finalmente en WALPO tras realizar esta adecuación es el que se ha especificado en la presente memoria.

7.2.3 Resultados obtenidos

Una vez mejorado el proceso inicialmente implementado en WALPO, se procedió a tratar de nuevo el fichero extracto_*campusF5.log-20120309.log* con la aplicación WALPO obteniendo los ficheros intermedio y el fichero final.

REGISTRO DE EVENTOS

Durante el tratamiento de los ficheros de log, WALPO produjo el siguiente registro de eventos:

```
Abriendo el fichero inicial extracto_campusF5.log-20120309.log...
Fichero inicial extracto_campusF5.log-20120309.log abierto correctamente.
Procesando el fichero inicialextracto_campusF5.log-20120309.log...
Fichero inicial extracto_campusF5.log-20120309.log procesado correctamente.
------------------------------------------------------
Generando el fichero intermedio...
```

```
Eliminando líneas no necesarias de extracto_campusF5.log-20120309.log...
Eliminación de líneas no necesarias realizada correctamente.
Obteniendo los identificadores de sesión...
Identificadores de sesión obtenidos correctamente.
Procesando las URL de los recursos solicitados...
URL de los recursos solicitados procesadas correctamente.
Transformando extracto_campusF5.log-20120309.log a formato CSV...
extracto_campusF5.log-20120309.log transformado a formato CSV correctamente.
El fichero intermedio se ha generado correctamente conteniendo 126753 líneas.
Se han descartado 372885 líneas de un total de 499638 existentes en el fichero
inicial.
-------------------------------------------------------
-------------------------------------------------------
Generando el fichero final...
Determinando el conjunto de recursos solicitados...
Recursos solicitados determinados correctamente.
Determinando las secuencias de navegación...
Secuencias de navegación determinadas correctamente.
Realizando la transformación a formato CSV...
Transformación a formato CSV realizada correctamente.
El fichero final se ha generado correctamente.
-------------------------------------------------------
```

FICHEROS DE ENTRADA Y SALIDA DE LA APLICACIÓN

Fichero de log inicial	
Tamaño del fichero	250 MB
Número de líneas	499.638
Número de elementos (columnas)	10

Fichero intermedio	
Tamaño del fichero	11,4 MB
Número de líneas	126.753
Número de elementos (columnas)	3

Observaciones	El 74,6 % de las líneas del fichero de log inicial fueron descartadas por WALPO durante la generación del fichero intermedio por no reunir los requisitos necesarios para el estudio (ver la descripción del flujo de trabajo en el apartado 4.1.2.1).

Fichero final	
Tamaño del fichero	1,11 MB
Número de líneas	1.384
Número de recursos únicos accedidos	411
Número de direcciones IP únicas	1.032
Observaciones	En una simple observación del contenido del fichero, pueden detectarse direcciones IP a las que se les han asignado diversos identificadores de sesión, lo que ha producido que estén representadas por varias líneas en el fichero final.

Recordar que si bien el fichero intermedio y el fichero de log inicial son comparables en cuanto a estructura, el fichero final tiene una estructura completamente diferente, similar a la del siguiente ejemplo:

```
IP,www.uoc.edu/a,www.uoc.edu/b,www.uoc.edu/c,www.uoc.edu/d,www.uoc.edu/e,www.uoc.edu/f
111.111.111.111,T,T,T,F,F,F
222.222.222.222,F,T,F,T,T,F
333.333.333.333,F,T,T,F,F,T
444.444.444.444,F,F,F,F,T,F
555.555.555.555,T,F,T,F,T,T
111.111.111.111,T,F,T,F,F,T
```

Recordar también que **cada línea del fichero final representa** lo que se ha denominado**' secuencia de navegación'**; es decir, el conjunto de recursos visitados (o no), por un usuario (identificado por una dirección IP) durante una sesión de trabajo; en el caso del ejemplo anterior, puede verse que la dirección IP 111.111.111.111 ha realizado dos sesiones de trabajo.

7.3 Procesamiento con Weka

El fichero final generado con WALPO se cargó en Weka utilizando el módulo 'Weka Explorer' y sobre él, se realizó la ejecución de los algoritmos considerados para el estudio.

7.3.1 Preparación inicial

Debido nuevamente a la problemática del tamaño de los ficheros a tratar, Weka fue incapaz de abrir el fichero final generado por WALPO. El problema fue causado por la cantidad de memoria reservada por

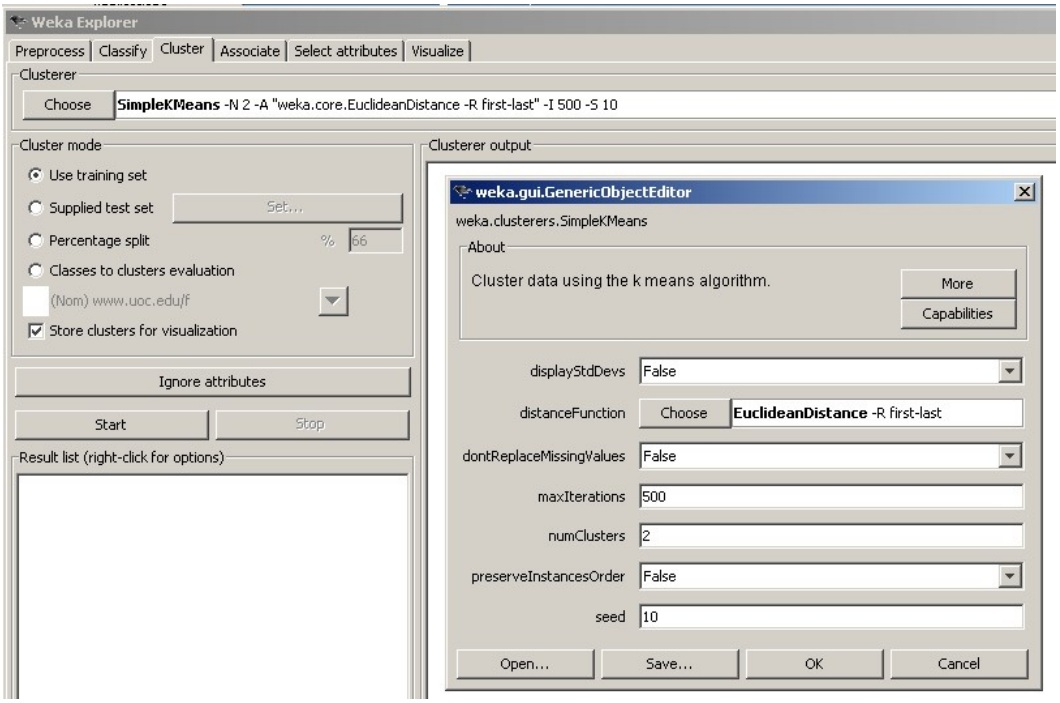
defecto para la aplicación y se solucionó aumentando el valor del parámetro `maxheap` en el fichero de configuración *RunWeka.ini* de la aplicación.

Tras diversas pruebas ejecutando varios algoritmos, el valor final fijado para el parámetro `maxheap` fue de 3512m (aproximadamente 3,5 GB de memoria), lo que puede dar un ejemplo de los requisitos hardware necesarios para realizar el análisis de este tipo de ficheros.

7.3.2 SimpleKMeans

7.3.2.1 Primera ejecución

En primer lugar, se ejecutó el algoritmo de agregación (clustering) SimpleKMenans, conservando los parámetros por defecto fijados por Weka:



Antes de ejecutar el algoritmo, se descartó el atributo 'IP' por no aportar valor a los resultados buscados; para ello se utilizó la funcionalidad ofrecida a través del botón 'Ignore atributes', seleccionando el atributo correspondiente.

Al ejecutar el algoritmo con los **parámetros por defecto** fijados por Weka **se generaron dos grupos (clusters)** con la siguiente distribución:

Resultados SimpleKMeans (parámetros por defecto)	
Número de líneas	1.384
Líneas en grupo 0	168 (12 %)
Líneas en grupo 1	1.216 (88 %)

Recordar que cada columna del fichero final procesado (descartando la primera por corresponder a la dirección IP), corresponde a un recurso accedido por alguna de las direcciones IP que figuran en el conjunto de datos analizado y que, para cada fila de dicho fichero (identificada por una dirección IP), cada recurso toma valor 'T' o 'F' según dicha dirección IP haya accedido o no al recurso durante una sesión de trabajo en el Campus Virtual. Teniendo esto presente, las direcciones accedidas en cada uno de los grupos es la son las siguientes:

Recursos accedidos en grupo 0
/tren/trenacc

Recursos accedidos en grupo 1
Ningún recurso accedido

Como puede observarse por lo resultados, resulta evidente que la ejecución sobre un elevado volumen de datos del algoritmo SimpleKMeans, considerando únicamente dos grupos, no aporta ningún valor al estudio; es por ello, que se realizó una nueva iteración.

7.3.2.2 Segunda ejecución

Debido al elevado número de datos, **ejecutar el algoritmo SimpleKMeans fijando** únicamente **dos grupos** (clusters) **produce resultados muy sesgados** y poco concluyentes (esto es muy apreciable a través de las tablas de resultados de la primera iteración), por ello, se realizó una segunda ejecución del algoritmo definiendo 10 grupos (parámetro `numClusters` de Weka con valor 10).

Los resultados obtenidos tras la nueva ejecución del algoritmo fueron los siguientes:

Resultados SimpleKMeans (numClusters = 10)	
Número de líneas	1.384
Líneas en grupo 0	40 (2,89 %)
Líneas en grupo 1	259 (18,71 %)
Líneas en grupo 2	133 (9,61 %)
Líneas en grupo 3	36 (2,60 %)
Líneas en grupo 4	429 (31,00 %)
Líneas en grupo 5	1 (0,07 %)
Líneas en grupo 6	117 (8,45 %)
Líneas en grupo 7	87 (6,29 %)
Líneas en grupo 8	227 (16,40 %)
Líneas en grupo 9	55 (3,87 %)

Dados los recursos que han sido marcados como accedidos (valor 'T') en alguno de los grupos y el conjunto total de grupos generados por el algoritmo, los resultados obtenidos fueron los siguientes:

Recurso / Grupo	0	1	2	3	4	5	6	7	8	9
/webapps/classroom/081_common/jsp/eventFS.jsp			T				T			
/WebMail/listMails.do				T						
/UOC2000/b/cgi-bin/ma_filter						T				
/cgi-bin/uocapp						T				
/UOC/a/jsstuff_mail.html									T	
/WebMail/resources/html/bodyHeight.html	T			T						
/webapps/classroom/081_common/jsp/iniciAula.jsp			T							
/UOC2000/b/extern_0.html						T				
/avis.html		T				T	T			T
/UOC2000/b/cgi-bin/ma_folders						T				
/UOC/a/ext_menu.html										T
/WebMail/readMailSecure.do	T			T						
/cgi-bin/avis						T	T			
/UOC2000/b/ext_menu.html						T				
/UOC/a/extern_0.html										T
/WebMail/resources/html/logobar.html				T						
/cgi-bin/ma_folders						T		T		
/cgi-bin/ma_buttons						T		T		
/cgi-bin/ma_mssgs						T		T		
/rb/inici/navigation/redir									T	
/UOC/js/banner.dat									T	
/webapps/classroom/081_common/jsp/event.jsp			T							
/UOC2000/b/cgi-bin/ma_buttons						T				
/cgi-bin/ma_filter						T				
/webapps/widgetsUOC/widgetsNovetatsExternesWithProviderServlet		T				T				T
/WebMail/sendMail.do	T			T						
/WebMail/readMail.do				T						
/webapps/widgetsUOC/widgetsRssServlet		T				T				
/UOC2000/b/cgi-bin/ma_mssgs						T				
/webapps/widgetsUOC/widgetsIcalServlet		T				T				

/WebMail/contacts.do	T			T						
/webapps/classroom/081_common/jsp/entrada.jsp			T				T			
/UOC/a/menu.htm									T	
/webapps/classroom/download.do			T							
Total accesos/grupo	4	4	5	7	0	16	4	3	4	4

Con la ejecución del algoritmo SimpleKMeans se pretendía determinar los recursos comúnmente accedidos, veamos lo logrado mediante la interpretación de los resultados obtenidos.

7.3.2.3 Análisis de resultados

Dadas la tablas de resultados presentadas en el apartado anterior, donde 'T' indica que se ha accedido a un recurso y 'F' que no se ha producido tal acceso, realizando el análisis de las mismas puede observarse lo siguiente:

1. El mayor grupo que se ha formado ha sido el grupo 4, que ha agrupado un total de 429 líneas, aunque por sí mismo no es el mayoritario.
2. Tras el grupo 4, los grupos 1 y 9 han sido los que más líneas han agrupado con 259 y 227 líneas respectivamente.
3. Se ha formado un grupo formado por una única línea.
4. El grupo 4, que ha sido el que más líneas ha agrupado ha tomado únicamente valores 'F'.
5. El grupo 5, que únicamente ha agrupado una línea, ha sido en el que más recursos han tomado valor 'T'.
6. El recurso que más valores 'T' ha tomado en el conjunto de los grupos ha sido /avis.html (4 valores), seguido de /webapps/widgetsUOC/widgetsNovetatsExternesWithProviderServlet (3 valores),

¿QUÉ CONCLUSIONES PODEMOS SACAR DE ESTO?

En primer lugar, que el grupo 4 haya sido el que más líneas ha agrupado y que a la par todos sus valores hayan sido 'F', no quiere decir que en el fichero final analizado existan líneas que no hayan realizado accesos a recursos sino que, dados los valores obtenidos registrados en dicho fichero, el valor 'F' es el mayoritario.

Los recursos que han tomado valor 'T' en alguno de los grupos son aquellos que han registrado más accesos en el fichero final (aunque no tienen por qué ser los más accedidos), quedando descartados los que han registrado menos accesos. Por ejemplo, en el caso de los dos recursos que más valores 'T' han tomado en el conjunto de grupos se tiene lo siguiente:

Recurso	Total accesos
/avis.html	454
/webapps/widgetsUOC/widgetsNovetatsExternesWithProviderServlet	277

Analizando el contenido del fichero final, los recursos que han sido considerados por el algoritmo han registrado como mínimo 42 accesos.

Si se observan únicamente los accesos realizados, podría decirse que los recursos accedidos más habitualmente son los representados en el grupo 5 pero, el hecho de que el número de líneas agrupadas haya sido despreciable (únicamente una), ha descartado esta posibilidad. Tampoco puede concluirse que la secuencia de navegación más habitual esté representada por el grupo 4, ya que el resultado está 'falseado' por el elevado número de valores 'F' presentes en el fichero final.

¿Cuáles son los recursos más habituales?; la respuesta es sencilla, aquellos que han sido accedidos más veces dado el conjunto de direcciones IP y sesiones del fichero final (recordar que una línea del fichero final queda determinada por esta pareja). Concretamente los siguientes:

Recurso	Total accesos
/avis.html	454
/webapps/classroom/081_common/jsp/entrada.jsp	406
/webapps/widgetsUOC/widgetsRssServlet	334
/webapps/widgetsUOC/widgetsIcalServlet	318
/rb/inici/navigation/redir	313
/webapps/classroom/081_common/jsp/eventFS.jsp	308
/webapps/widgetsUOC/widgetsNovetatsExternesWithProviderServlet	277
/UOC/a/jsstuff_mail.html	245
/cgi-bin/avis	245
/UOC/a/menu.htm	242

Entonces podríamos preguntarnos ¿porqué es necesario ejecutar un algoritmo de minería de datos si los recursos más habituales pueden obtenerse con un sencillo tratamiento del fichero final?; evidentemente, porque las técnicas de minería de datos permiten sacar conclusiones de importante valor añadido que no pueden extraerse directamente:

1. Los recursos accedidos por cada estudiante (identificado por su dirección IP y sesión de trabajo) son completamente heterogéneos, es decir, que existen muchas diferencias entre lo que realiza un estudiante u otro durante una sesión de trabajo; por otra parte, cada estudiante no accede a un elevado número de recursos durante una misma sesión de trabajo. Esto ha producido la aparición de un elevado número de valores 'F' en el fichero final.

   El hecho de que se haya generado un grupo con muchos valores 'T' pero que únicamente agrupa una línea (grupo 5) señala que es poco frecuente que un estudiante acceda a muchos recursos durante una sesión de trabajo en el Campus Virtual.

   El hecho de que se haya generado un grupo con todos sus valores a 'F' (grupo 4), permite también concluir que lo habitual es que cada estudiante no realice muchos accesos a recursos en cada sesión de trabajo (únicamente unos pocos recursos toman valor 'T' en cada fila del fichero final) pero también que el conjunto total de estudiantes accede a recursos muy diversos y de ahí las 411 columnas que aparecen en el fichero final.

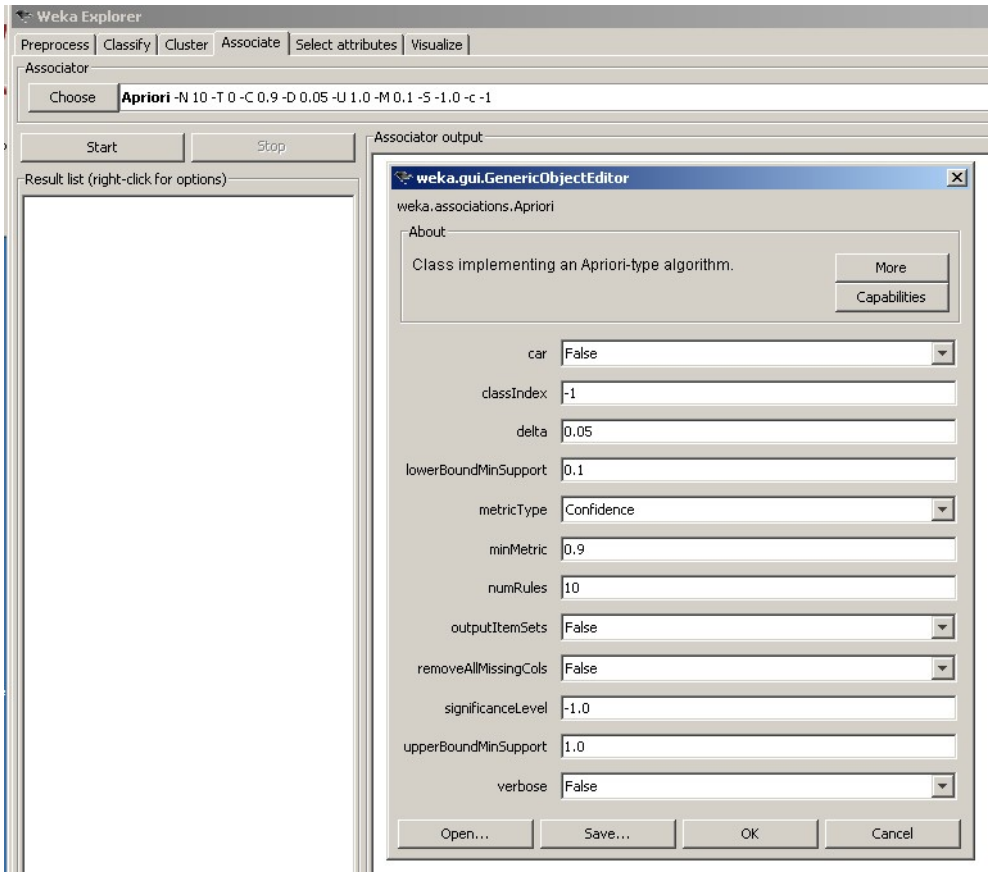
2. El análisis de los grupos 1 y 9 (El segundo y tercero que han agrupado más líneas), permite extraer que, independientemente del número total de accesos, los estudiantes del Campus Virtual suelen acceder a los recursos cuya URL es /avis.html y /webapps/widgetsUOC/widgetsNovetatsExternesWithProviderServlet; esto viene apoyado por el hecho de que son estos dos recursos los que han tomado más valores 'T' en el total de grupos.

En relación a los recursos más habituales, aunque no se conoce la equivalencia entre cada una de las URL y la parte del Campus Virtual de la UOC a la que corresponden, sí que es posible realizar una tentativa para intentar intuir dicha equivalencia a través de las URL de los recursos de la tabla anterior. Por ejemplo, se presupone que las URL que incluyen el término 'avís' refieren a algún tipo de notificación emitida en el Campus Virtual, del mismo modo que el término 'widget' podría referir a los módulos de la página de inicio del Campus Virtual, que 'classroom' podría referir a las aulas a las que tienen acceso los estudiantes y 'mail' al correo electrónico.

Quedarían descartadas /rb/inici/navigation/redir y /UOC/a/menu.htm por creer que refieren a algún tipo de redirección y a alguna carga del menú principal y, por tanto, considerarse 'ruido' que no ha sido filtrado por los algoritmos implementados.

7.3.3 Apriori

En segundo lugar, se ejecutó el algoritmo de asociación Apriori conservando los parámetros por defecto fijados por Weka:



Antes de ejecutar el algoritmo, se eliminó el atributo 'IP' por no aportar valor a los resultados buscados; para ello seleccionó dicho atributo en la pestaña 'Preprocess' y se utilizó la funcionalidad ofrecida a través del botón 'Remove'.

Al ejecutar este algoritmo sobre el fichero final, nuevamente apareció un problema derivado del elevado volumen de datos a tratar; en este caso, a pesar de haber reservado unos 3,5 GB de memoria para la ejecución de Weka, la aplicación, tras varios minutos ejecutando el algoritmo lanzó una excepción informando del agotamiento de la memoria 'heap' de Java. Dicho error se obtuvo en los sucesivos intentos realizados modificando el valor de los parámetros del algoritmo.

Tras realizar una indagación sobre las causas del problema, se llegó a la conclusión de que se trata de una cuestión conocida derivada de la forma en la que está implementado este algoritmo en Weka [30].

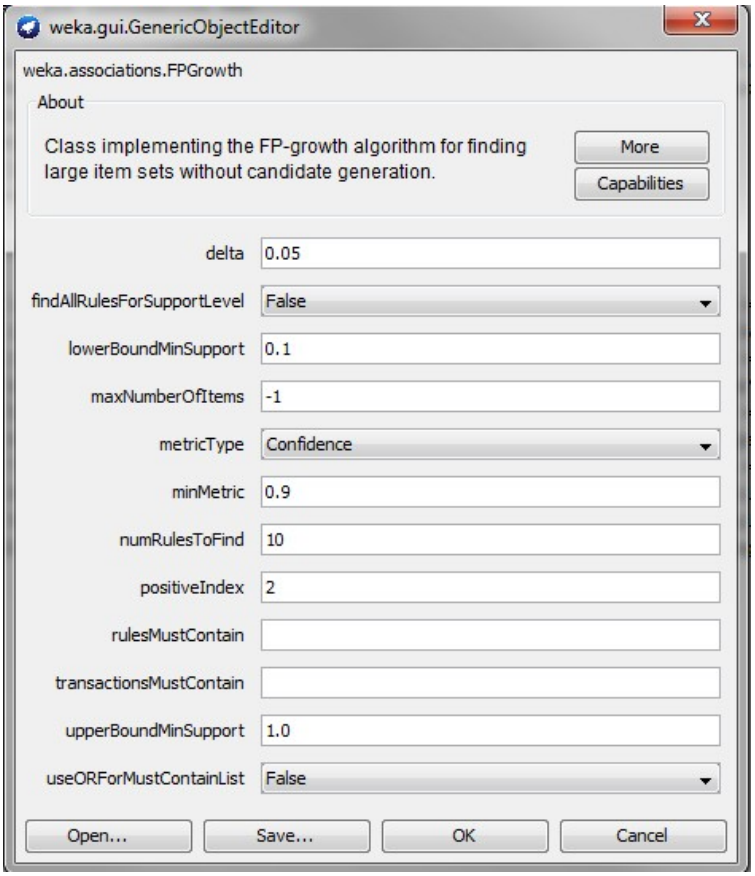
Debido a esto, fue necesario localizar un algoritmo alternativo que permitiera la obtención de reglas de asociación utilizando Weka; la respuesta llegó en [30], donde se encontraron referencias al algoritmo FPGrowth.

7.3.4 FPGrowth

FPGrowth es un algoritmo que se comporta de forma más óptima y rápida que APriori con grandes volúmenes de datos [31], puesto que únicamente realiza dos iteraciones. Este algoritmo se basa en el almacenamiento de la información de forma comprimida en una estructura de árbol denominada FP-tree y es aquí donde se consigue la mejora sobre APriori al reducir el número de iteraciones necesarias para procesar la información.

7.3.4.1 Primera ejecución

Tras los problemas producidos al intentar ejecutar el algoritmo APriori, se ejecutó en su lugar el algoritmo FPGrowth conservando los parámetros por defecto fijados por Weka:



Antes de ejecutar el algoritmo, se eliminó el atributo 'IP' tanto por no aportar valor a los resultados buscados como porque este algoritmo no soporta valores numéricos; para ello seleccionó dicho atributo en la pestaña 'Preprocess' y se utilizó la funcionalidad ofrecida a través del botón 'Remove'.

La ejecución de este algoritmo produjo los siguientes resultados:

```
=== Run information ===

Scheme:       weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1
Relation:     extracto_campusF5.log-20120309_FINAL-weka.filters.unsupervised.attribute.Remove-R1
Instances:    1384
Attributes:   411
[list of attributes omitted]
=== Associator model (full training set) ===

FPGrowth found 57002 rules (displaying top 10)


1. [/webapps/classroom/img/pladocentdinamic/UOC-logo.svg=F]: 1379 ==>
[/webapps/classroom/dtd/xhtml1-transitional.dtd=F]: 1379   <conf:(1)> lift:(1)
lev:(0) conv:(4.98)

 2. [/webapps/classroom/dtd/xhtml1-transitional.dtd=F]: 1379 ==>
[/webapps/classroom/img/pladocentdinamic/UOC-logo.svg=F]: 1379   <conf:(1)>
lift:(1) lev:(0) conv:(4.98)
```

```
 3. [/webapps/classroom/img/pladocentdinamic/UOC-logo.svg=F]: 1379 ==>
[/webapps/classroom/dtd/xhtml-symbol.ent=F]: 1379   <conf:(1)> lift:(1)
lev:(0) conv:(4.98)

 4. [/webapps/classroom/dtd/xhtml-symbol.ent=F]: 1379 ==>
[/webapps/classroom/img/pladocentdinamic/UOC-logo.svg=F]: 1379   <conf:(1)>
lift:(1) lev:(0) conv:(4.98)

 5. [/webapps/classroom/img/pladocentdinamic/UOC-logo.svg=F]: 1379 ==>
[/webapps/classroom/dtd/xhtml-special.ent=F]: 1379   <conf:(1)> lift:(1)
lev:(0) conv:(4.98)

 6. [/webapps/classroom/dtd/xhtml-special.ent=F]: 1379 ==>
[/webapps/classroom/img/pladocentdinamic/UOC-logo.svg=F]: 1379   <conf:(1)>
lift:(1) lev:(0) conv:(4.98)

 7. [/webapps/classroom/img/pladocentdinamic/UOC-logo.svg=F]: 1379 ==>
[/webapps/classroom/dtd/xhtml-lat1.ent=F]: 1379   <conf:(1)> lift:(1) lev:(0)
conv:(4.98)

 8. [/webapps/classroom/dtd/xhtml-lat1.ent=F]: 1379 ==>
[/webapps/classroom/img/pladocentdinamic/UOC-logo.svg=F]: 1379   <conf:(1)>
lift:(1) lev:(0) conv:(4.98)

 9. [/app/phpBB3/totales.php=F]: 1376 ==>
[/webapps/classroom/img/pladocentdinamic/UOC-logo.svg=F]: 1376   <conf:(1)>
lift:(1) lev:(0) conv:(4.97)

10. [/app/microblog/totales.php=F]: 1376 ==>
[/webapps/classroom/img/pladocentdinamic/UOC-logo.svg=F]: 1376   <conf:(1)>
lift:(1) lev:(0) conv:(4.97)
```

Tras realizar una valoración inicial de los resultados obtenidos, estos se determinaron no satisfactorios dado que no posibilitaban extraer ningún tipo de conclusión. Se obtuvo un elevado número de reglas de asociación y ninguna de las analizadas, a pesar de su valor de confianza '1', aportó valor al estudio.

7.3.4.2 Segunda ejecución

Tras descartar los resultados obtenidos en la primera ejecución, se decidió realizar una segunda prueba modificando el valor por defecto de los siguientes parámetros:

Parámetros modificados	
findAllRulesForSupportLevel	True
maxNumberOfItems	2

Si bien el parámetro `findAllRulesForSupportLevel` se modificó con el objetivo de obtener todas las reglas posibles para el nivel de soporte fijado, el parámetro `maxNumberOfItems` se modificó para lograr el efecto contrario, restringir el número de reglas obtenidas a aquellas que estuvieran formadas como máximo por dos elementos.

Esta segunda ejecución del algoritmo produjo los siguientes resultados:

```
=== Run information ===

Scheme:       weka.associations.FPGrowth -P 2 -I 2 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S
Relation:     extracto_campusF5.log-20120309_FINAL-weka.filters.unsupervised.attribute.Remove-R1
Instances:    1384
Attributes:   411
[list of attributes omitted]
=== Associator model (full training set) ===

FPGrowth found 286 rules
```

En este caso se consiguió reducir el número de reglas y obtener resultados más satisfactorios. Aunque no todas las reglas de asociación que determinó el algoritmo aportaban valor al estudio, si que se consiguió extraer algunas interesantes:

1. `[/webapps/classroom/081_common/jsp/iniciAula.jsp=T] ==> [/webapps/classroom/081_common/jsp/entrada.jsp=T] conf:(0.97)`
2. `[/tren/trenacc=T] ==> [/tren/trenacc/web/GAT_EXP.PLANDOCENTE=F] conf:(0.98)`
3. `[/UOC/a/jsstuff_mail.html=T] ==> [/UOC/a/menu.htm=T] conf:(0.98)`
4. `[/UOC/a/jsstuff_mail.html=T] ==> [/tren/trenacc/web/GAT_EXP.PLANDOCENTE=F] conf:(0.99)`
5. `[/webapps/classroom/081_common/jsp/fitxa_calendari.jsp=T] ==> [/webapps/classroom/081_common/jsp/entrada.jsp=T] conf:(0.99)`
6. `[/webapps/classroom/student.do=T] ==> [/tren/trenacc/web/GAT_EXP.PLANDOCENTE=F] conf:(1)`
7. `[/webapps/widgetsUOC/widgetsRssServlet=T] ==> [/tren/trenacc/web/GAT_EXP.PLANDOCENTE=F] conf:(1)`
8. `[/webapps/widgetsUOC/widgetsIcalServlet=T] ==> [/tren/trenacc/web/GAT_EXP.PLANDOCENTE=F] conf:(1)`

Con la ejecución del algoritmo FPGrowth se pretendía determinar reglas de navegación a modo de tendencias más habituales, veamos lo logrado mediante la interpretación de los resultados obtenidos.

7.3.4.3 Análisis de resultados

Analizando cada una de las reglas de asociación obtenidas, puede concluirse que aquellos estudiantes del Campus Virtual de la UOC que en su sesión de trabajo:

1. Han accedido a /webapps/classroom/081_common/jsp/iniciAula.jsp, también han accedido a /webapps/classroom/081_common/jsp/entrada.jsp
2. Han accedido a /tren/trenacc, no han accedido a /tren/trenacc/web/GAT_EXP.PLANDOCENTE.

   Esta regla es especialmente interesante porque /tren/trenacc es la URL vinculada, entre otras zonas, a la consulta del expediente, datos personales, datos bancarios... Si se generalizara, podría afirmarse que los estudiantes que han accedido al Campus Virtual para realizar la consulta o gestión de estos datos, lo han hecho únicamente con esa intención.

3. Han accedido a /UOC/a/jsstuff_mail.html, también han pasado por /UOC/a/menu.htm
4. Han accedido a /UOC/a/jsstuff_mail.html, no han accedido a /tren/trenacc/web/GAT_EXP.PLANDOCENTE
5. Han accedido a /webapps/classroom/081_common/jsp/fitxa_calendari.jsp, han pasado por /webapps/classroom/081_common/jsp/entrada.jsp
6. Han accedido a /webapps/classroom/student.do, no han accedido a /tren/trenacc/web/GAT_EXP.PLANDOCENTE
7. Han accedido a /webapps/widgetsUOC/widgetsRssServlet, no han accedido a /tren/trenacc/web/GAT_EXP.PLANDOCENTE
8. Han accedido a /webapps/widgetsUOC/widgetsIcalServlet, no han accedido a /tren/trenacc/web/GAT_EXP.PLANDOCENTE

Aunque no se han incluido en las reglas de asociación seleccionadas, muchas de las reglas mostradas en Weka referían a/tren/trenacc/web/GAT_EXP.PLANDOCENTE con valor 'F'; de esto se puede extraer una última conclusión y es que los estudiantes que accedían a este recurso, no solían acceder a muchos otros durante su sesión de trabajo.

8 Conclusiones y futuras mejoras

8.1 Futuras mejoras

8.1.1 Mejoras del Campus Virtual de la UOC y los ficheros de log

Aún disponiendo de un proceso definido, como ha podido observarse, **analizar los patrones de navegación** de los estudiantes del Campus Virtual de la UOC a partir de los ficheros de log generados por el servidor **requiere un alto coste de recursos hardware**, **es una tarea compleja** que requiere múltiples tratamientos previos para eliminar el 'ruido' (registros que no aportan valor al análisis de patrones de navegación) de los ficheros **y no asegura un 100% de éxito**. Evidentemente, **no es un problema ni de la estructura del Campus Virtual ni de los ficheros de log** que cumplen perfectamente con su cometido.

Es por ello que **se propone la implantación de un sistema de log paralelo al actual**, especialmente diseñado para extraer conocimiento de la interacción de los usuarios en el Campus Virtual y facilitar el análisis de patrones de navegación. Los logs generados a través de este sistema paralelo **omitirían el 'ruido'** que aparece en los ficheros de log actuales (recordar que el tratamiento de los ficheros de log ha eliminado alrededor del 75% de los registros iniciales), **declararían de forma clara y explícita los recursos accedidos** (dado que las URL son difíciles de interpretar, estas podrían ir acompañadas por la denominación del recurso correspondiente) por cada estudiante y **dispondrían de una estructura normalizada** (por ejemplo XML). Disponer de un sistema paralelo posibilitaría incluso evitar el uso de ficheros de texto y **delegar la persistencia de la información en una base de datos relacional**.

Además de una previsible mejora del rendimiento en el acceso a la información, el uso de una base de datos relacional **facilitaría la recuperación de información** no sólo de un día concreto (recordar que los ficheros de log únicamente almacenan información de un día), sino de **intervalos de tiempo lo suficientemente prolongados** como para disponer de conjuntos de datos lo suficientemente amplios como **para aumentar la fiabilidad de los análisis**.

UN EJEMPLO

Existen múltiples formas de implementar este sistema de logs paralelo de forma no 'traumática' que varían en función de la plataforma de programación utilizada; por ejemplo, en el ámbito de la plataforma JEE existen los denominados filtros (en inglés filter) que actúan a modo de interceptores y no requieren modificar el código fuente de la aplicación a la que se incorporan. Los filtros se vinculan a clases Java (por ejemplo aquellas que son invocadas cuando un usuario realiza una determinada acción en la capa de presentación) y disparan la ejecución de otras clases; de esta forma, cuando un usuario demandara por ejemplo acceder al correo electrónico, se dispararía automáticamente el filtro y se registraría el evento correspondiente en el log.

Más concretamente, se podría declarar lo siguiente en el archivo *web.xml* de una aplicación web basada en JEE:

```
<filter>
```

```
      <filter-name>FiltroLog</filter-name>
      <filter-class>ClaseFiltroLog</filter-class>
</filter>
<filter-mapping>
      <filter-name>FiltroLog</filter-name>
      <url-pattern>ControladorCorreoElectronico</url-pattern>
</filter-mapping>
```

El código anterior desencadenaría la ejecución de la clase `ClaseFiltroLog` en el momento en que se detectara una llamada a la clase de tipo Servlet `ControladorCorreoElectronico`, todo ello sin necesidad de modificar dicho Servlet.

En el caso del Campus Virtual de la UOC, dependerá de las tecnologías con las que esté implementado para poder aplicar una solución equivalente.

8.1.2 Mejoras del proceso

Una vez conocidos los múltiples problemas derivados del gran volumen de datos a tratar, se cree que trabajar directamente con ficheros de texto no es la solución más adecuada sino que debería realizarse un volcado de la información en una base de datos.

Por lo tanto, la mejora en la implementación del proceso vendría dada en que los datos de los tres ficheros implicados en el mismo (fichero de log inicial, fichero intermedio y fichero final) quedarían almacenados en una base de datos relacional de tal forma que fueran tablas de dicha base de datos las que fueran accedidas por Weka, la aplicación de minería de datos, que contempla dicha posibilidad entre sus características funcionales.

Dado que el problema no pudo ser cuantitativamente valorado hasta que no se realizó la primera prueba con datos reales y que dicha prueba no pudo realizarse hasta haber especificado el proceso e implementado la aplicación de tratamiento de ficheros de log, no se dispuso del tiempo suficiente para implementar la mejora aquí indicada.

En relación a la aplicación, una de las mejoras contempladas es que la parametrización de la misma pudiera realizarse mediante ficheros de configuración externos, de tal forma que fuera fácilmente adaptable a diferentes estudios.

8.1.3 Mejoras de la aplicación WALPO

Si bien se ha logrado que la aplicación WALPO cumpla completamente con el objetivo fijado, como suele suceder en el mundo del software, es susceptible de mejora.

Una de las mejoras contempladas para la aplicación es dotarla de la capacidad de parametrizar las reglas de negocio (las expresiones regulares que posibilitan el tratamiento de los ficheros de log del Campus Virtual) y las constantes de configuración a través de ficheros externos. Tener la capacidad de configurar la aplicación a través de ficheros externos, posibilitaría su rápida adecuación(sin necesidad de recompilar la aplicación) para otro tipo de estudios o el tratamiento de ficheros de log con estructura diferente.

Por último, tal vez la mejora más importante es que la aplicación trabajara sobre una base de datos en vez de directamente con ficheros de texto; esto permitiría, por una parte, trabajar con históricos de datos

y, por otra parte, mitigar los problemas de memoria derivados de tener que tratar ficheros con un gran volumen de datos y mejorar el rendimiento.

8.2 Conclusiones

Aunque el objetivo marcado para este PFC era inicialmente el análisis de patrones de navegación de los estudiantes del Campus Virtual de la UOC, se cree que la verdadera aportación que se ha logrado con este proyecto es haber planteado un proceso normalizado (evidentemente susceptible de mejora) para realizar dicho análisis.

A nivel personal, la realización del presente PFC me ha permitido introducirme en un ámbito, el del análisis de patrones de navegación, en el que mis conocimientos iniciales no eran elevados y profundizar en otro, la minería de datos, en el que se tuvo contacto inicial en dos de las asignaturas cursadas en la UOC.

Los múltiples problemas detectados y resueltos al tener que tratar grandes volúmenes de datos han permitido conocer en profundidad un aspecto que no suele ser habitual, y me han permitido obtener conocimiento acerca de las limitaciones del software y la realización de optimizaciones del mismo.

Por otra parte, se cree que lo aprendido aportará un importante valor añadido en el desempeño de mi actividad laboral.

9 Bibliografía

[1] *Documento de especificación y análisis*. Alumno Julio Mateos. Disponible en el foro de la asignatura 'PFC-Aplicaciones web para trabajo colaborativo', en el Campus Virtual de la UOC.

[2] *Plan docente de la asignatura*. Disponible en el Campus Virtual de la UOC, asignatura 'PFC - Aplicaciones web para trabajo colaborativo'.

[3] *Apuntes de la asignatura 'Metodología y gestión de proyectos informáticos'*. Disponible en la sección 'Materiales' de la Secretaría del Campus Virtual de la UOC.

[4] *Apuntes de la asignatura 'Minería de datos'*. Disponible en la sección 'Materiales' de la Secretaría del Campus Virtual de la UOC.

[5] *Sistema de Análisis de Patrones de Navegación Usando Minería Web*. Disponible en:

http://www.dspace.espol.edu.ec/bitstream/123456789/5014/2/7998.pdf

[6] *Patrones de Navegación de Usuarios de un Campus Virtual*. Disponible en:

http://www.aipo.es/articulos/3/16.pdf

[7] *Data Mining of User Navigation Patterns*. Disponible en:

http://www.dcs.bbk.ac.uk/~mark/download/web_mining.pdf

[8] *Métodos de petición*. Disponible en:

http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node15.html

[9] *WebTaller - Aprender CGI. ¿Cuál es la diferencia entre GET y POST?* Disponible en:

http://www.webtaller.com/construccion/lenguajes/cgi/lessons/diferencia.php

[10] *HTTP ("Hyper Text Transfer Protocol")*. Disponible en:

http://www.cicei.com/ocon/gsi/tut_tcpip/3376c426.html#SECTION00733000000000000000

[11] *Data Mining of Web Access Logs From an Academic Web Site*. Vic Ciesielski and Anand Lalani. Diciembre 2003. Disponible en:

http://goanna.cs.rmit.edu.au/~vc/papers/his03-lalani.pdf

[12] *Data Mining of Web Access Logs*. Anand S. Lalani. Julio 2003. Disponible en:

http://goanna.cs.rmit.edu.au/~vc/papers/lalani-mbc.pdf

[13] *Visualizing and Discovering Web Navigational Patterns*. Jiyang Chen, Lisheng Sun, Osmar R. Zaïane, Randy Goebel. Disponible en:

http://webdb2004.cs.columbia.edu/papers/1-3.pdf

[14] Clustering navigation patterns on a website using a Sequence Alignment Method. Birgit Hay, Geert Wets and Koen Vanhoof. Disponible en:

http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.20.8354

[15] *Weka*. Wikipedia. Disponible en:

http://es.wikipedia.org/wiki/Weka_%28aprendizaje_autom%C3%A1tico%29

[16] *Apache Access Log Format*. Disponible en:

http://support.moonpoint.com/network/web/server/apache/log_format.php

[17] *Hypertext Transfer Protocol -- HTTP/1.1: Status Code Definitions*. Disponible en:

http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

[18] *A New Approach for Clustering of Navigation Patterns of Online Users*. Dipa Dixit; Jayant Gadge. Disponible en:

http://www.ijest.info/docs/IJEST10-02-06-26.pdf

[19] *K-Means*. Wikipedia. Disponible en:

http://en.wikipedia.org/wiki/K-means_clustering

[20] *Apriori algorithm*. Wikipedia. Disponible en:

http://en.wikipedia.org/wiki/Apriori_algorithm

[21] *Reglas de asociación*. Wikipedia. Disponible en:

http://en.wikipedia.org/wiki/Association_rule_learning

[22] *Minería de Datos y el Algoritmo APRIORI*. Dr. Claudio Meneses Villegas. Disponible en:

http://iii.informatica.edu.bo/index.php?option=com_content&view=article&id=98:mddyeaa&catid=43:revista2007&Itemid=60

[23] *Técnicas de análisis de datos en WEKA*. Disponible en:

http://isa.umh.es/asignaturas/crss/turorialWEKA.pdf

[24] *Aplicación de técnicas de aprendizaje automático para la identificación de patrones de interacción en una experiencia virtual de aprendizaje*. Priscila Valdiviezo Díaz. Disponible en:

http://reposital.cuaed.unam.mx:8080/jspui/bitstream/123456789/2702/1/priscila_valdiviezo_tecnicas_de_aprendizaje.pdf

[25] *Técnicas de Análisis de Datos: Aplicaciones prácticas utilizando Microsoft Excel y Weka*. José Manuel Molina López; Jesús García Herrero. Disponible en:

http://es.scribd.com/doc/58003065/Tecnicas-de-Analisis-de-Datos

[26] *Regular Expressions - User Guide*. Disponible en:

http://www.zytrax.com/tech/web/regex.htm

[27] *Open Large Text Files or Logs*. Disponible en:

http://cybernetnews.com/open-large-text-files-logs/

[28] *How to split a very large text or CSV file by a specific number of lines/rows*. Disponible en:

http://www.freewaregenius.com/2009/07/30/how-to-split-a-very-large-text-or-csv-file-by-a-specific-number-of-lines-rows/

[29] *10 points about Java Heap Space or Java Heap Memory*. Disponible en:

http://javarevisited.blogspot.com.es/2011/05/java-heap-space-memory-size-jvm.html

[30] *Scalability of WEKA using Apriori*. Disponible en:

http://forums.pentaho.com/archive/index.php/t-73920.html

[31] *Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach*. J. Han; J.Pei; Y. Yin. Mayo 2000. Disponible en:

http://www.cs.uiuc.edu/~hanj/pdf/dami04_fptree.pdf