UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

AGENTE AUTÓNOMO ADAPTABLE PARA LA CONSTRUCCIÓN DE SUGERENCIAS SOBRE EL USO DE LA PLATAFORMA MOODLE

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADA EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A:

NOMBRE DEL ALUMNO :
GRISEL ANGÉLICA PÉREZ QUEZADA

DIRECTOR DE TESIS:
DR. GUSTAVO DE LA CRUZ MARTÍNEZ

2015

Ciudad Universitaria, D. F.

Agradecimientos

A mi familia por su incondicional apoyo.

A mi asesor de tesis, que me ha brindado sus conocimientos, apoyo, comprensión y sobre todo paciencia.

Índice General

Introducción

Generalmente entre los usuarios hay un desconocimiento de todo el potencial que ofrece un software a través de su interfaz, si a esto le sumamos que varias de las interfaces no son intuitivas, tenemos como resultado que un alto porcentaje de los usuarios subutiliza estas interfaces. Esto se debe a que la mayoría de las interfaces se construyen teniendo en mente un usuario modelo; por consiguiente, el software no tiene forma de presentar la información de manera más personalizada, ya que el problema radica en que no todos los usuarios piensan, aprenden y resuelven las cosas de la misma manera, sabiendo de ante mano que las personas desarrollan a través de sus vivencias cierta individualidad.

Es por esta razón que es necesario observar la actividad de la interacción del usuario a través de sus sesiones de trabajo y tareas que vaya resolviendo sobre la interfaz para conocer más sobre las necesidades específicas de cada usuario. Este seguimiento tiene sus bases en el área de interacción humano-computadora, la cual se describirá en el capítulo 1, que tiene como propósito el estudio del intercambio de información entre un usuario y una interfaz, para entender cómo los usuarios resuelven las tareas y así crear sistemas que lo apoyen de manera más específica.

Dentro del área del diseño de interfaces de usuario existen varios tipos. El tipo de interfaz que estudiaremos en este trabajo de tesis es la interfaz adaptativa la cual permite tener esa flexibilidad ante las necesidades del usuario, con el propósito de mejorar la forma en que los usuarios interactúan con la interfaz.

Existen diferentes modelos que permiten estudiar esta interacción para el caso de la web; en este contexto, a esta área se le conoce como *personalización de la web*. Las propuestas de esta área muestran que es posible adaptar un sistema web a las necesidades de cada usuario a partir de la observación de este último. En este trabajo se utilizará principalmente el trabajo de dos autores (Brusilovsky, 2007) y (Mobasher, 2007), que han desarrollado estudios importantes en el área. Estos dos modelos son la base de este trabajo, ya que muestran diferentes formas de analizar los perfiles de usuario, sin embargo, son muy generales y deben completarse con otras teorías propuestas sobre el modelado del usuario a partir del seguimiento de su interacción con una interfaz.

De manera más específica se analiza la interacción con una interfaz web, donde se lleva a cabo un análisis de los datos almacenados en las bitácoras de acceso de los servidores web, para encontrar frecuencias en la navegación de cada usuario. Una herramienta de gran utilidad para el análisis de esta información es la minería de datos, que se encarga de la extracción de información a partir del descubrimiento de patrones, tendencias y relaciones no explícitas que son difíciles de visualizar con técnicas de análisis tradicionales. Dentro de la minería existen varias ramas, una de ellas es la minería de la web, encargada de analizar

la información que se produce sobre la web. Dentro de esta rama se desprende la minería del uso de la web, donde se hace uso de las técnicas de minería de datos para el descubrimiento de patrones sobre la navegación web.

La minería de datos cuenta con una gran variedad de algoritmos estadísticos para encontrar patrones de uso en la navegación. Los patrones son una parte interesante de esta tesis porque sin ellos no es posible identificar los intereses de cada usuario. Como se describe más adelante en este trabajo, observando los patrones de navegación podemos detectar si existen dudas sobre cómo está organizado un sitio o alguna sección donde los usuarios no pueden resolver una tarea, inclusive si tienen dudas sobre su funcionamiento, pero en especial se puede identificar qué usuarios sí realizan tareas específicas y cómo las resuelven. Este tipo de observaciones son utilizadas para determinar la clasificación de los usuarios en cuanto a sus necesidades, cómo se mueven por la interfaz, qué les interesa, entre otras cosas.

Para el descubrimiento y análisis de patrones, en este trabajo se utiliza la herramienta llamada Weka que tiene implementados diferentes algoritmos como One-Rule o k-means, por mencionar algunos. El objetivo de este trabajo no es modificar los algoritmos, sino usarlos como una herramienta para detectar los patrones de navegación y así construir un clasificador basado en reglas, que un agente usará para generar las sugerencias personalizadas para los usuarios.

La tarea esencial en una interfaz adaptativa es la predicción de los intereses y necesidades que puede tener un usuario, buscando siempre encontrar una sugerencia consistente y útil. Es por esta razón que se decidió desarrollar un agente que tenga las características de ser autónomo y adaptable lo cual le permitirá captar los cambios en su entorno y evaluar las condiciones para determinar cuál es la sugerencia más adecuada para el usuario.

El objetivo principal es mostrar que es posible generar sugerencias sobre la plataforma Moodle. Sin embargo, sólo se analizó la interfaz para el rol de alumno, sobre los roles restantes como profesor y administrador se debe hacer un análisis diferente, porque el modelo de aprendizaje aplicado al agente es diferente para cada uno de los roles.

Una característica importante de la solución presentada en este trabajo, es que se diseñó una estrategia para capturar la interacción del usuario con una mínima modificación al código del sitio que se observa. Se decidió analizar la interacción de los usuarios de una plataforma de aprendizaje basado en web, llamada Moodle. Después de analizar el comportamiento de los usuarios de esta plataforma, se observó que los alumnos no encontraban con facilidad los recursos que son de utilidad para resolver una tarea del curso, así que se generó una serie de sugerencias para que los alumnos pudieran encontrar más fácilmente los recursos relacionados con sus tareas, exámenes y proyectos, con el objetivo de orientarlos a través de esta aplicación.

Esta IUA (interfaz de usuario adaptativa) será capaz de procesar información del usuario para generar sugerencias sin que el usuario lo pida explícitamente, el sistema tendrá que generar este conocimiento de acuerdo con la actividad de los usuarios. La sugerencia desplegará un recuadro en un lugar específico de la página para que el usuario pueda verlo, donde mostrará un conjunto de recursos que están relacionados a esa actividad.

Para llegar a este resultado debemos pasar por varias etapas, primero se debe mostrar cómo obtener los datos de los usuarios sin que se llegue a interrumpir de alguna manera la interacción del usuario con la interfaz o llegue a ser intrusivo.

La obtención de información del usuario sobre una plataforma web, se realiza convencionalmente mediante formularios que contienen preguntas acerca del funcionamiento de la aplicación, intereses personales, entre otros. El objetivo de una IUA es tomar información del usuario de manera más directa y sin que los usuarios estén enterados de esta acción. Con el propósito de no interferir en sus decisiones o influir en ellas.

Para ello es necesario procesar los datos debidamente para encontrar información útil donde se logre identificar cuáles son las problemáticas más recurrentes y en base al análisis se puedan definir las reglas para la construcción del agente. El agente genera una sugerencia por medio de la información que se ha procesado, con el objetivo de que sea de utilidad para el usuario.

La propuesta que aquí se presenta está dividida en 6 capítulos, con el fin de introducir los elementos y teorías que necesitamos para la construcción de nuestro motor de sugerencias personalizadas.

En el primer capítulo se exponen las propiedades que debe tener una interfaz de usuario y las diferencias que hay entre una interfaz de usuario tradicional y una interfaz de usuario adaptativa, además se describen dos propuestas para la obtención de la navegación de sistemas web.

El segundo capítulo presenta un panorama general de la minería de datos, sus diferentes áreas, técnicas y métodos con los que procesa la información para realizar su análisis. En particular, se describe la minería de la web que se subdivide en minería del uso, contenido y estructura de la web. En este trabajo utilizaremos la minería del uso de la web, que se encarga de analizar la información producida en la web, por ejemplo, la información de las bitácoras de uso de los servidores web para encontrar patrones en la navegación.

En el tercer capítulo se plantea la propuesta de este trabajo para realizar el seguimiento del usuario, la cual consiste en obtener el flujo de todos los eventos generados por el usuario sobre la interfaz que posteriormente guardaremos en una tabla de una base de datos, para un análisis que permita encontrar patrones en la navegación del usuario.

A través del cuarto capítulo se define lo que es un agente y sus características. Además se explica el proceso de construcción para una interfaz de usuario adaptativa y por último, se describe cómo se implementó el agente autónomo adaptable sobre la plataforma Moodle.

En el quinto capítulo se muestra cómo funciona la plataforma Moodle con el agente autónomo adaptable ya integrado, se analiza su funcionamiento y sus posibles fallas.

En el sexto capítulo se presentan las conclusiones de esta tesis y el trabajo a futuro de este proyecto.

Es importante resaltar el hecho de que sólo queremos mostrar que es posible construir una IUA mediante este modelo y no que resolverá todos los problemas que el usuario tenga sobre la interfaz, para esto es necesario una mayor entrada de datos, un análisis más riguroso sobre el procesamiento de los datos y probar con algún otro algoritmo que obtenga patrones si los resultados de éste y el que expusimos aquí llevan a un mejor resultado.

Capítulo 1

Interfaz de usuario adaptativa.

En los últimos años ha crecido el interés por el uso de las nuevas tecnologías para apoyar y automatizar muchas actividades que realizamos en la vida cotidiana, a tal grado que se han vuelto una necesidad para los seres humanos.

Una interfaz es un dispositivo capaz de transformar señales generadas por éste en señales comprensibles para algún otro dispositivo. Por ejemplo: el ratón es un instrumento que sirve para extender los movimientos de nuestra mano y lleva señales a la pantalla en forma de cursor.

En este capítulo sólo se presentarán los tipos de interfaz que se suelen encontrar a través de la web y su clasificación. Estas interfaces se encuentran en casi todos los sitios y su construcción lleva un proceso de análisis con respecto a las actividades que el usuario debe resolver: su aprendizaje, preferencias, intereses y otras más.

Las interfaces de usuario tienen un problema muy particular debido a que los usuarios tienen individualidad, no realizan las tareas de la misma manera, se ha visto que esto provoca que muchos usuarios no puedan generar el proceso de aprendizaje correcto sobre la interfaz que utilizan, se ha pensado que esto es debido a que se presenta el mismo contenido de la interfaz a todos los usuarios, ya que la construcción de una interfaz se utiliza un usuario modelo, el cual tiene características específicas como el saber usar una computadora o haber utilizado una interfaz parecida anteriormente.

Para modificar esta perspectiva hacemos uso de otro tipo de interfaz que es conocida como interfaz de usuario adaptativa (IUA), a través de estos capítulos se describirá el proceso de construcción, sus características y objetivos que puede cumplir.

La IUA es una interfaz especial, ya que para su construcción se necesita del procesamiento de datos que vienen de la interacción del usuario con la interfaz, con el objetivo de presentar a cada usuario información que sea de su interés.

Para su construcción son necesarias algunas teorías relacionadas con el aprendizaje del usuario y la interacción de éste con alguna interfaz, las teorías que se han utilizado son conocidas como hipermedia adaptativa y personalización de la web. Estas teorías han desarrollado varias metodologías para obtener información por medio de los elementos de

la interfaz con los que interactúa el usuario, por ejemplo: con vínculos, botones, campos de texto, entre otros.

1.1. Interfaz de Usuario.

Una interfaz de usuario (IU) es un dispositivo que permite la interacción y entendimiento de dos sistemas que no hablan el mismo lenguaje. En este caso solo se estudiaran las interfaces de usuario sobre la web, por ejemplo: un sitio web nos permite realizar tareas referentes a un problema en particular realizándose una interacción entre el usuario y la computadora. Las interfaces de usuario para la web contienen un conjunto de elementos gráficos donde el usuario interactúa con éstos por medio de: formularios, vínculos, botones, entre otros. Estos elementos son visualizados en el navegador, que es el medio que permite al usuario realizar tareas sobre la aplicación para la web. La interfaz de usuario es el resultado de un proceso de diseño que se enfoca principalmente en detectar las funcionalidades del sistema y quién es el usuario final.

En el proceso de diseño de una interfaz se sigue una metodología de desarrollo que permite entender la actividad del usuario. Por ejemplo: el diseño centrado en el usuario (Schneiderman , 2006), toma en cuenta las características de éste, es decir, a qué grupo de personas va dirigido y qué es lo que se quiere resolver, esto nos dará una idea más clara sobre las tareas más relevantes. En cualquier caso, existe el riesgo de que algunas de las tareas del usuario no sean tan fáciles de encontrar.

Mediante pruebas de usabilidad es posible detectar cuando el usuario tiene dificultades para entender cómo se usa la interfaz, si esto sucede, el usuario optará por no volver a usarla. Por esta razón es indispensable el análisis sobre la forma en que se realiza la interacción del usuario con la aplicación.

Existe una rama de estudio que se le conoce como interacción humano-computadora (IHC), en la que participan pedagogos, psicólogos, diseñadores gráficos, educadores, redactores de informes, expertos en factores humanos o en ergonomía, arquitectos de la información, antropólogos y sociólogos. Es una disciplina relacionada con el diseño, evaluación e implementación de sistemas de cómputo interactivos para uso humano, así como el estudio de los fenómenos que ocurren a su alrededor (Hewett, 1996).

El problema principal cuando se diseña una interfaz es que la información que se presenta es la misma para todos los usuarios, por ejemplo: en una tienda en línea se presentan ofertas de todas las categorías, pero si al usuario sólo le interesa ver las ofertas de la categoría de electrónica, el contenido no será de interés y tardará más tiempo en encontrar las ofertas de su agrado. En cambio, si se le presentaran los objetos que son de su interés desde un inicio, no tendría que buscarlos y sería más cómodo para los usuarios.

Para solucionar algunos de los problemas que se presentaron anteriormente, se han desarrollado nuevas líneas de trabajo que se enfocan en identificar y atender las necesidades específicas de los diferentes usuarios de un mismo sistema de cómputo.

1.2 Interfaz de usuario adaptativa.

El diseño de una interfaz de usuario adaptativo (IUA) es muy similar al proceso de diseño de una interfaz de usuario, pero con la diferencia de que se añaden funciones para obtener información del usuario y un agente que tiene las propiedades de ser autónomo y adaptable. Su objetivo es identificar las características de un usuario a través del conocimiento generado del análisis de la información obtenida de la observación de sus sesiones y sugerir información o adecuar la manera en que se soluciona una tarea de acuerdo a sus necesidades específicas.

Para un buen diseño de una interfaz de usuario (Mandel, 1997) menciona tres reglas de oro:

1. Dar control al usuario.
2. Reducir la carga de memoria del usuario.
3. Construir un interfaz consistente.

Muchas de las estrategias utilizadas para la implementación de la interfaz de usuario se desarrollan en ciclos iterativos, tratando de complementar en cada ciclo las 4 tareas básicas:

- Análisis y modelado de usuarios, tareas y el ambiente.
- Diseño de la interfaz.
- Construcción de la interfaz.
- Validación de la interfaz.

En cuanto a una IUA es necesario incorporar la información del usuario obtenida justamente después de que se tiene una interfaz de usuario terminada, donde se hace uso de ciertos mecanismos para obtener datos del usuario de manera más específica y no tan general. Además se utiliza un agente que construye las sugerencias y se retroalimenta de los datos que produce el usuario a través de sus sesiones. En general, estos puntos serían las principales distinciones entre una IU e IUA, considerando que al incorporar estos mecanismos no modifican la percepción que el usuario tiene del sistema, ya que no se genera ningún cambio en la interfaz de usuario y solo se añaden algunos elementos al HTML donde se muestra una sugerencia que es generada por el agente, de tal forma que no altera la estructura de la interfaz.

Las ventajas que existen sobre una IUA es que muestran sugerencias con respecto a la agrupación de preferencias, es decir, a un grupo de usuarios con intereses similares les mostrará recursos que son de su interés, ya que utiliza la información del usuario capturando sus hábitos, preferencias y especialmente la forma en que el usuario realiza la tarea para que el agente aprenda en base a la experiencia del usuario. Ésta entidad tiene la tarea de analizar y comprender cómo es que se realizan las tareas.

Por otro lado, existen problemas en el procesamiento de los datos, es posible que los datos que se han introducido en las tablas de la base de datos contengan valores nulos, vacíos o que simplemente no aporten información. En este caso, el manejo de valores nulos y vacíos es el mismo, al no aportar información útil sobre el usuario se omiten, ya que lo importante es capturar la interacción con el sistema. Sin embargo, esto puede ser distinto en otros casos, los valores nulos o vacíos dificultan el análisis por que no permite la aplicación de las técnicas existentes para descubrir información relevante. Se puede tener control sobre los datos nulos o vacíos si se sabe de dónde proviene el error y tratar de completarlos o descartarlos, para ello se aplican técnicas de evaluación de valores perdidos.

Para realizar un mejor estudio de este tipo de interfaces en (Ross, 2000) se propone la siguiente clasificación:

Adaptación directa de la interfaz: Este tipo de interfaz cuenta con elementos que intervienen de forma directa en la interacción con el usuario y tienen como tarea principal la predicción de la actividad del usuario. Dentro de esta categoría se observan dos tipos de comportamiento: las interfaces que filtran información (interfaces informativas) y aquellas que sugieren información (interfaces generadoras).

Interfaces intermediarias: Este tipo de interfaces hacen sugerencias, corrigen ideas equivocadas y en general guían al usuario (Maes, 1994). La interfaz cuenta con agentes autónomos que realizan diversas tareas de apoyo, sin embargo, el usuario mantendrá el control total de la interfaz. En la mayoría de éstas, el agente es visible para el usuario. A este tipo de interacción se le conoce como manejo indirecto (Kay, 1990).

Interfaces de agente: La principal característica de los agentes autónomos es que brinda un apoyo pro-activo al usuario. El agente típicamente asistirá mediante sugerencias, estas sugerencias pueden ser rechazadas y se cuida que no sean invasivas. Esta propuesta se apoya en la capacidad de las computadoras para solucionar problemas que se consideran complejos de resolver con métodos tradicionales, así como actividades repetitivas o tediosas, para dejar al usuario final la toma de decisiones que no pueden ser automatizadas.

1.3. Interfaces adaptativas para la web.

Para el caso de la navegación web, encontramos dos propuestas que estudian el problema de la adaptación de una interfaz web: *la personalización de la web y la hipermedia adaptativa*, ambas pretenden generar respuestas de acuerdo a las necesidades de cada usuario y resolver el problema de mostrar el mismo contenido para todos los usuarios.

1.3.1 Personalización de la web.

La personalización de la web tiene como objetivo proveer a los usuarios lo que necesitan o sugerir algo que le sea de utilidad sin preguntar al usuario explícitamente. Un ejemplo de esto se puede observar en el sitio de Amazon, el cual muestra productos que pueden interesarnos sin preguntarnos explícitamente por nuestras preferencias. Se basa en el filtrado colaborativo, es decir, toma los datos de los usuarios que tienen gustos similares así que es muy posible que a los usuarios con preferencias semejantes les gusten los mismos libros, películas, por mencionar algunos, por lo tanto, podemos sugerir productos en base a estas similitudes. Todo esto tiene sus bases en las teorías que a continuación se mencionan.

(Mobasher, 2000) propone tres formas en las que podemos personalizar un sistema, las cuales son:

- **Sistemas basados en reglas:** Estos sistemas permiten a los administradores del sitio web especificar reglas basadas en demografía, psicografía y características personales de los usuarios, para adaptar el contenido mostrado al usuario cuyo perfil cumpla con una o más condiciones de las reglas. Las reglas son obtenidas de la interacción del usuario con el sistema y los datos son utilizados para determinar sus intereses. Generalmente estos sistemas generan reglas de decisión de forma manual o automática y así recomiendan elementos a los usuarios.

- **Sistemas de filtrado de contenido:** Las descripciones del contenido se representan por un conjunto de características o atributos con respecto a un elemento, en este caso un perfil de usuario está representado por la descripción de los objetos en los que ha tenido interés previo. Un ejemplo de esto es en las tiendas en línea donde se puede identificar cuáles son los artículos más visitados o más vendidos y esta información se puede usar para recomendar los productos que estén más relacionados con el perfil del usuario. Para encontrar los intereses sobre un objeto o artículo en particular, se utilizan métodos probabilísticos como la clasificación bayesiana.

- **Sistemas de filtrado colaborativo:** Es muy similar a las técnicas mencionadas antes aunque con más potencial. Tradicionalmente, la técnica principal es el método de

clasificación vecino más cercano (KNN). Compara el perfil de usuario con los perfiles que tienen guardados previamente en la base de datos de otros usuarios con el fin de encontrar los k usuarios que tienen gustos o intereses similares. Sin embargo, estas técnicas de filtrado colaborativo tienen varias limitaciones, la más importante es su falta de escalabilidad. El KNN requiere que la fase de formación de vecinos sea un proceso en línea, es decir, la fase de modelado se realiza en tiempo real. Mientras exista un aumento de los elementos y usuarios, este enfoque no podrá generar en un tiempo aceptable la construcción de recomendaciones o contenido dinámico durante la interacción con el usuario. Otra limitación de KNN es la naturaleza del conjunto de datos, a medida que el número de elementos aumenta, la base también crece por lo que la densidad del registro del usuario se reducirá.

1.3.2 Hipermedia adaptativa.

Por otra parte (Brusilovsky, 1998) propone la idea de adaptación de la web o hipermedia adaptativa, la cual consiste en que un sistema debe adaptarse a las metas, intereses y conocimiento individual de cada usuario, creando un modelo de usuario individual. Sugiere el uso de tecnologías y mecanismos que permiten adecuar el comportamiento del sistema a las necesidades del usuario.

Las tecnologías que se presentaran son diferentes, pero tienen el mismo propósito que es el de adaptar los elementos de la interfaz para presentar los enlaces a tareas específicas, conocimientos y preferencias de cada usuario. De acuerdo con (Brusilovsky, 2007) las tecnologías pueden agruparse de la siguiente forma:

Guía directa: Sugiere contenido basado en las características como metas, conocimientos y otros parámetros que son representados en el modelo de usuario. En la interfaz la guía directa puede enfatizar los vínculos de posible interés en la página o bien, generar los vínculos recomendados.

Ordenar vínculos: Esta propuesta se enfoca en darle un nivel de prioridad a todos los vínculos de una página en particular, el nivel de prioridad estará en función de las características que estén representadas en el modelo de usuario y un criterio de valoración.

Ocultar vínculos: Con esta tecnología se busca restringir el espacio de navegación ocultando, eliminando o deshabilitando los vínculos a páginas irrelevantes mediante el cambio de color.

Explicar vínculos: La idea básica de esta tecnología es complementar los vínculos con una explicación en forma de nota o comentario, lo que permite al usuario conocer más sobre los

nodos asociados a los vínculos. Se muestran notas o comentarios en los vínculos para que el usuario tenga una mejor idea del contenido al que desea acceder.

Generar vínculos: Propone la creación de nuevos vínculos en el sistema y se clasifican en tres tipos:

a) Nuevos vínculos descubiertos entre los documentos y agregados de forma permanente al sistema.

b) Vínculos generados de acuerdo a la similitud entre vínculos visitados previamente.

c) Recomendación de vínculos de acuerdo al contexto actual para el modelo de usuario.

Para este proyecto se hará uso de los mecanismos propuestos por Brusilovsky. Un mecanismo es la forma en que se completa y se presenta la información, por ejemplo, en el caso de una interfaz web se utilizan vínculos para hacer notar al usuario la existencia de otras secciones, pero también es posible poner una imagen relacionada con el contenido de esa sección. Ciertamente esto no implica que sea la mejor opción, pero es otra manera de mostrar información al usuario. A continuación se enlistan algunos mecanismos válidos:

Mecanismos de adaptación simple: No utilizan algoritmos de adaptación avanzados. Los ejemplos típicos son:

- ***Mecanismos basados en la historia:*** Llevan el control de cuantas veces se accede a un vínculo y se presenta visualmente un historial de la actividad.

- ***Mecanismos basados en eventos:*** Consiste en capturar el evento del vínculo visitado y se produce un cambio de color para su distinción.

- ***Mecanismos basados en el progreso:*** Extiende el principio de los mecanismos basados en historia, explora a un nivel más detallado la interacción del usuario con las páginas y presenta gráficamente su progreso.

Mecanismos basados en contenido: Se utiliza para recomendar la siguiente página de acuerdo al contexto de la página donde se sitúa. Para encontrar qué página va a sugerir utiliza vectores que contienen palabras clave, después las compara con el perfil de usuario y lo actualiza.

Mecanismos sociales: Este mecanismo es introducido por (Dourish y Chalmers, 1994), se basa en el concepto de la navegación social, es decir, se forman grupos de personas que hacen las mismas actividades que otros y se clasifica en dos tipos: la navegación social directa o indirecta. En la navegación social directa la interacción entre los usuarios se da de forma explícita entre unos y otros en un espacio informativo. En la navegación social indirecta los nuevos usuarios se guían por lo que otros han hecho, en este caso intervienen 2 tipos de sistemas: *los ambientes de historial enriquecido*, que brindan apoyo para la navegación y agregan acciones individuales y se puede visualizar lo que otros hacen; por otra parte, están *los sistemas de filtrado colaborativo* que consisten en hacer recomendaciones con las preferencias o recomendaciones de los demás usuarios.

Mecanismos basados en índices: Al igual que los mecanismos basados en contenido, este tipo de mecanismos manejan información relacionada a cada página que será utilizada para guiar al usuario. En este caso, la información sobre los contenidos no se produce de forma automática, sino que es especificada de manera manual y se basa en una serie de conceptos bien definidos por lo que brinda mayor precisión.

Para el mecanismo basado en índices existen varias técnicas que los agrupan de la siguiente manera:

- **Clasificación de los enfoques de indexación:** Existen tres atributos importantes para distinguir los diferentes enfoques de indexación desde la perspectiva del soporte de navegación adaptativa: cardinalidad, fuerza expresiva y navegación.

    - ***Cardinalidad****:* Es un conjunto que se compone de vínculos que están relacionados con algún tema o concepto, hay esencialmente sólo dos diferentes casos:

        - ***Indexación de concepto simple (Categorización):*** Cada página se relaciona a un concepto de acceso externo, es decir, se relaciona con algún vínculo que procede fuera de la aplicación.

        - ***Indexación multi-concepto:*** Es más potente, pero hace que el sistema sea más complejo y requiere modelos externos más elaborados.

    - ***Fuerza expresiva***: Se refiere a la cantidad de información que se puede asociar en cada enlace entre un concepto y una página.

    - ***Navegación:*** Es importante cuando distinguimos entre los casos donde la relación entre un concepto y una página existen solo sobre un nivel

conceptual, así como también para los casos donde cada vínculo representa una URL o PATH que indica de donde proviene.

- **Hiperespacio basado en concepto:** Un hiperespacio en esta área es un conjunto de páginas y vínculos que se presentan de manera estructurada; esta estructura debe tener la propiedad de ser jerárquica. Es uno de los métodos más sencillos para organizar conexiones entre modelos externos y el hiperespacio.

- **Indexado por página:** Es el enfoque de diseño estándar para sistemas con indexación multi-concepto. Con este enfoque la página es indexada con varios modelos de concepto externo, es decir, se crean vínculos entre una página y cada concepto que describa a ésta.

Este es un ejemplo de una aplicación implementada con un mecanismo de indexación basada en Interbook (Brusilovsky, 1998). Es la primera aplicación que utilizó técnicas de adaptación de la web. Interbook consiste en una colección de libros que están ordenados jerárquicamente, utiliza un modelo de dominio que crea el perfil de usuario basado en conocimientos e indexa cada sección de los libros en los estantes.

1.4 Resumen.

En este capítulo hemos planteado las bases teóricas que necesitamos para poder construir una interfaz de usuario adaptativa, el resultado final dependerá directamente del tipo de aplicación con la que se trabaja, sin embargo, debemos tener en cuenta que este tipo de interfaz únicamente resuelve uno de los problemas mencionados anteriormente sobre una interfaz de usuario, pero no todos. Tal y como se ha visto, las tecnologías y los mecanismos nos apoyan para resolver ciertos problemas, además de proporcionar de manera más simple e intuitiva el uso de la interfaz al usuario.

Las dos teorías que se describieron en este capítulo, *personalización de la web e hipermedia adaptativa*, son los marcos teóricos que usaremos para la captura de la iteración de los usuarios con la interfaz y es posible encontrar gran variedad de elementos de los que podemos extraer información.

Capítulo 2

Minería del uso de la web.

La minería de datos se define como el proceso de extraer conocimiento útil y comprensible dentro de grandes cantidades de datos que se encuentran en distintos formatos (Witten & Frank, 2000). Para ello, la minería de datos utiliza métodos de disciplinas como: probabilidad, inteligencia artificial, redes neuronales, entre otras.

Uno de los retos más importantes para los sistemas de búsqueda de información es el tiempo de respuesta hacia los usuarios, se han explorado diferentes estrategias que permitan mejorar el tiempo de respuesta así como la precisión de los resultados. Una de las estrategias para mejorar la búsqueda de información es a través del uso de palabras clave encontradas en las páginas, adicionalmente, en internet se usan descriptores XML con información relacionada de forma explícita con las páginas para optimizar estas búsquedas.

Para la explotación de esta gran cantidad de datos las empresas hacen uso de las relaciones explícitas que están presentes en su negocio; por ejemplo, la información de los clientes se almacena en una base de datos y a través de consultas se obtienen reportes de ventas, de cobros y deudas. Pero existen relaciones no triviales entre los datos que se pueden identificar con un simple análisis y que podrían ser de importancia; por ejemplo, descubrir qué tipo de usuario es al que le conviene dar algún crédito con base en su historial de operaciones bancarias.

Para encontrar conocimiento no explícito en este tipo de datos se emplea la minería de datos. La minería de datos se apoya en técnicas estadísticas para encontrar las pistas que nos conducirán a revelar la información escondida, además se ha identificado su utilidad en la toma de decisiones, el descubrimiento de tendencias o desviaciones para poder crear estrategias comerciales o de ventas.

En específico para este trabajo se usará una rama llamada minería de la web, que se concentra en analizar la información disponible y generada alrededor de las actividades en la web.

2.1 Metodologías para el descubrimiento del conocimiento.

Existen tres metodologías utilizadas en el proceso de descubrimiento del conocimiento en las bases de datos, éstas son: KDD, CRISP-MD y SEMMA. Estas metodologías constan de una serie de fases.

- ***KDD:*** Se define como un proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y en última instancia, comprensibles a partir de los datos. La figura 1 muestra las fases de este proceso, las cuáles son: selección,

limpieza, transformación y proyección de los datos, después se aplican técnicas de minería de datos para la obtención de patrones, éstos se analizan, se evalúan y se visualizan. Por último los resultados obtenidos son la representación conocimiento encontrado. (Fayyad, 1996).

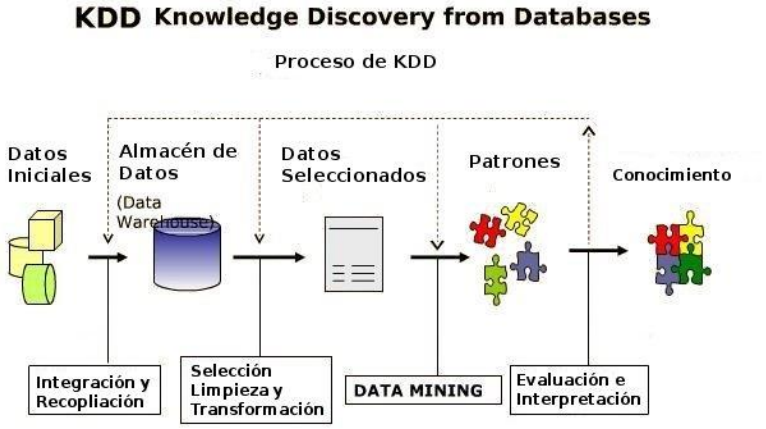

Figura 1: Proceso KDD

CRISP-MD: *(Cross-Industry Standard Process for Data Mining)* es una iniciativa financiada por la Comunidad Europea, se ha unido para desarrollar una plataforma para estandarizar el proceso de descubrimiento sobre la minería de datos. Éste proceso consta de seis fases que son:

1. ***Entendimiento del negocio:*** Inicia con el entendimiento de los objetivos y requerimientos del proyecto desde la perspectiva del negocio.
2. ***Entendimiento de los datos:*** Se obtienen los datos iniciales y los analiza teniendo presente los objetivos del negocio.
3. ***Preparación de los datos:*** Los datos se procesan para construir un conjunto de datos final.
4. ***Modelado:*** Se aplican las técnicas de minería que mejor se adapten al problema.
5. ***Evaluación:*** Se determina si el modelo construido satisface los objetivos del negocio.
6. ***Despliegue:*** Por último se aplican los resultados al modelo.

Algunas de las fases son bidireccionales, lo que significa que éstas permitirán revisar parcial o totalmente las fases anteriores. La figura 2 muestra éstas relaciones:

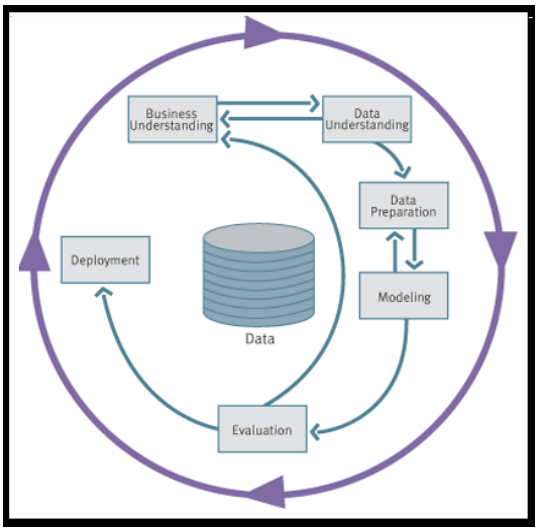

Figura 2: Proceso *CRISP-MD*

- ***SEMMA:*** Es un acrónimo a las cinco fases: *(Sample, Explore, Modify, Model, Assess).* La metodología es propuesta por SAS Institute Inc, la define como: "Proceso de selección, exploración y modelado de grandes cantidades de datos para descubrir patrones de negocios desconocidos". La figura 3 muestra el proceso.

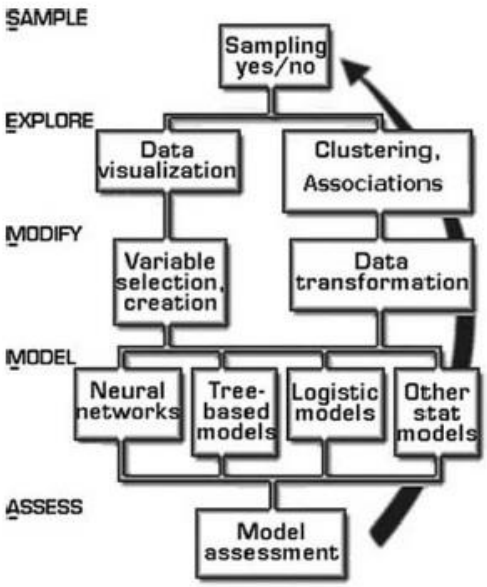

Figura 3: Proceso SEMMA

Para este proyecto utilizamos el siguiente modelo representado en la figura 4, propuesto por (Mobasher, 2007).

La meta del procesamiento de datos con este modelo es la transformación de flujo de clics a un conjunto de perfiles de usuario. Cada perfil contiene una secuencia de páginas visitadas en la duración de su sesión dentro de la aplicación, estos datos pueden ser usados como

entrada para una gran variedad de algoritmos de minería de datos, así como abstraídos y transformados.

La idea general se sustenta en que los datos deben llevar un pre-procesamiento para evitar problemas en fases posteriores. Es posible que datos incompletos, nulos o vacíos y erróneos, provoquen regresar a fases anteriores y sin resultado alguno, pero son necesarias varias iteraciones para extraer conocimiento de alta calidad.

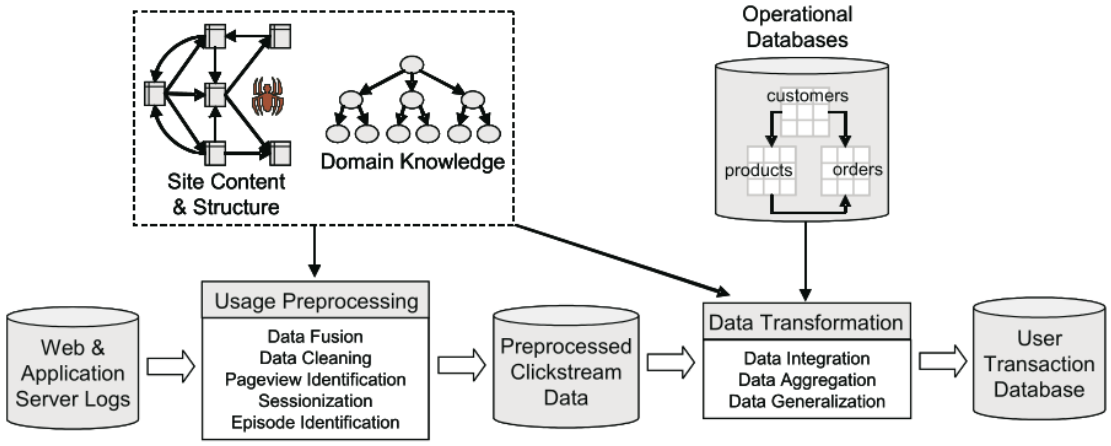

Figura 4: Procesamiento de datos (Mobasher)

2.2 Minería de datos.

Los tipos de datos a los que normalmente se les aplica la minería de datos pueden ser de dos tipos: estructurados o no estructurados. Los tipos de datos estructurados son los que están almacenados en las bases de datos relacionales, textuales y multimedia. Los tipos de datos no estructurados provienen de la web y de repositorios de documentos.

Al analizar los datos para extraer el conocimiento, éste puede presentarse en forma de patrones, relaciones y reglas que construyen un modelo. Los modelos en la minería de datos pueden ser de dos tipos: descriptivos o predictivos. Los modelos predictivos pueden estimar el valor que tendrá alguna variable o campos en algún futuro, llamadas variables objetivo o dependientes, las variables independientes o predictivas se calculan a partir de otras variables o campos de la base de datos. Los modelos descriptivos identifican los patrones que pueden encontrarse en los datos que ayudan a la toma de decisiones.
En la minería de datos existen tareas y métodos, hay que especificar cuál es la diferencia entre estos dos conceptos: Una tarea en minería de datos es un tipo de problema; por ejemplo, "*clasificar las piezas de algún proveedor, en óptimas, defectuosas reparables y defectuosas irreparables*", es una tarea de clasificación. Para resolver las tareas se usan métodos: árboles de decisión o redes neuronales, entre otros.

2.3 Tareas de minería de datos.

En la minería de datos se pueden distinguir varios tipos de tareas y pueden considerarse como un tipo de problema a resolver por algún método de minería de datos. Cada tarea tiene ciertos requisitos y los resultados pueden variar con cada tipo de tarea. Una tarea formalmente se define como un conjunto $\boldsymbol{E}$ que contenga todos los posibles elementos de entrada, es decir, los registros contenidos en una base de datos. Los elementos de $\boldsymbol{E}$ se representan como un conjunto de valores para una serie de atributos (sean nominales o numéricos), i.e. Sea:

$$\boldsymbol{E = A_1 \times A_2 \times \dots A_n}$$

Entonces un elemento $\boldsymbol{e_i}$ es una tupla:

$$\boldsymbol{e_i = < a_1, a_2, \dots a_n > : a_i \in A_i}$$

De manera más detallada *(Hernández, Ramírez & Ferri, 2007)* proponen que las tareas de la minería de datos se dividen en *predictivas y descriptivas*:

1. **Predictivas:** Son problemas donde hay que predecir valores de algún conjunto de datos y dan como resultado una clase, categoría, valor numérico o un orden entre ellos. Dependiendo como es la correspondencia entre los datos y los valores de salida, podemos agruparlos de la siguiente manera:

   1.1. Clasificación (o discriminación): Empareja o asocia datos a grupos definidos, encuentra funciones que describen y distinguen las clases o conceptos para futuras predicciones.

   Los ejemplos son un conjunto de pares de elementos de dos conjuntos.

   $$\boldsymbol{\delta_i = \{< e_i , s_i > \mid e_i \in E \; y \; s_i \in S\}}$$

   Donde $\boldsymbol{S}$ es el conjunto de valores de salida y $\boldsymbol{E}$ los de entrada. Los elementos $\boldsymbol{e_i}$ al ir acompañados de un valor $\boldsymbol{s_i}$ se denominan conjunto de datos de etiquetado. La función $\boldsymbol{\lambda}$: $\boldsymbol{E \longrightarrow S}$ es una función clasificadora *i.e.* para cada valor de $\boldsymbol{E}$ debe tener un único valor en $\boldsymbol{S}$ por lo que la función es inyectiva. Como $\boldsymbol{S}$ es el conjunto de salida ésta contiene un conjunto de valores $\boldsymbol{c_1, c_2 \dots c_i}$ denominados clases. La función construida debe ser capaz de determinar la clase; es decir, trata de definir el mapeo entre $\boldsymbol{E}$ y $\boldsymbol{S}$ donde a cada $\boldsymbol{e_i}$ se le asigna una clase $\boldsymbol{c_i}$.

   1.2. *Clasificación suave:* Tiene la misma función que la técnica anterior, pero además contiene una función clasificadora que la acompaña obteniendo un valor más certero sobre la predicción. Sean los conjuntos $\boldsymbol{E}$ y $\boldsymbol{S}$ donde:

   $$\boldsymbol{\delta_i = \{< e_i , s_i > \mid e_i \in E \; y \; s_i \in S\}}$$

Además de la función $\boldsymbol{\lambda}: \boldsymbol{E} \longrightarrow \boldsymbol{S}$ se utiliza otra función $\boldsymbol{\theta}: \boldsymbol{E} \longrightarrow \mathfrak{R}$ que indica el grado de certeza de la predicción hecha por $\boldsymbol{\lambda}$. Este tipo de extensión permite realizar otras aplicaciones, como son los rankings de predicciones o la selección de los $\boldsymbol{n}$ mejores ejemplos.

1.3. ***Estimación de probabilidad de clasificación:*** Es la generalización de la clasificación suave, utiliza los mismos conjuntos $\boldsymbol{E}$, $\boldsymbol{S}$ donde:

$$\boldsymbol{\delta_i} = \{< e_i, s_i > | \boldsymbol{e_i} \in \boldsymbol{E} \boldsymbol{y} \boldsymbol{s_i} \in \boldsymbol{S}\}$$

La función de clasificación es distinta a la función suave, se trata de aprender exclusivamente $\boldsymbol{m}$ funciones $\boldsymbol{\theta_i}: \boldsymbol{E} \longrightarrow \mathfrak{R}$, donde $\boldsymbol{m}$ es el número de clases, es decir, cada función retorna para cada ejemplo $\boldsymbol{\theta}(\boldsymbol{m}) = \boldsymbol{p_i}$ un valor real. Cada uno de estos valores $\boldsymbol{p_i}$, significan el grado de certeza que un ejemplo sea de la clase i. Si además cumple que $\forall \boldsymbol{p_i}: \boldsymbol{0} \leq \boldsymbol{p_i} \leq \boldsymbol{1}$ y además $\boldsymbol{\Sigma p_i} = \boldsymbol{1}$, estas $\boldsymbol{p_i}$ representan la *probabilidad* de que un ejemplo sea de la clase i. El conjunto de funciones aprendidas se denominan *estimadoras de probabilidad*. Un *estimador* es una regla que expresa cómo calcular la *estimación*, basándose en la información de los ejemplos y se enuncia mediante una fórmula.

1.4. ***Categorización:*** Las categorías son la clasificación más básica sobre objetos o conceptos y representan un conjunto de objetos con características específicas. Sean $\boldsymbol{E}$, $\boldsymbol{S}$ donde :

$$\boldsymbol{\delta_i} = \{< e_i, s_i > | \boldsymbol{e_i} \in \boldsymbol{E} \boldsymbol{y} \boldsymbol{s_i} \in \boldsymbol{S}\}$$

Los conjuntos antes mencionados pueden asignar varias clases y cada categoría va acompañada de su certeza con un estimador de probabilidad, en este caso la suma de las probabilidades puede ser mayor a 1, aunque en $\boldsymbol{\delta_i}$ se pueden tener varias etiquetas para el mismo ejemplo, el estimador de probabilidades de clasificación y el estimador de probabilidades de categorización son lo mismo. Para usarlo como clasificador debemos seleccionar la clase de mayor probabilidad y para utilizarlo como una categorización debemos seleccionar las $\boldsymbol{k}$ mejores categorías o las que superen cierto umbral.

1.5. ***Preferencias o priorización:*** El aprendizaje de las preferencias consiste en determinar a partir de 2 o más ejemplos un orden de preferencia. Cada ejemplo es una secuencia:

$$< \boldsymbol{e_i}, \boldsymbol{e_{i+1}}, \ldots \boldsymbol{e_n} >: \boldsymbol{e_i} \in \boldsymbol{E}, \text{ con } \boldsymbol{i} \geq 2$$

Donde el orden de la secuencia representa la predicción. Un conjunto de datos para este problema es un conjunto de secuencias:

$$\boldsymbol{\delta_j} = \{< \boldsymbol{e_1}, \boldsymbol{e_2}, \ldots \boldsymbol{e_i} >: \boldsymbol{e_i} \in \boldsymbol{E}\}$$

Donde el modelo aprendido sólo será capaz de decir cuál prefiere entre dos elementos (decide cuando $\boldsymbol{e_i}$ es mejor que $\boldsymbol{e_j}$), pero no ordenar más de 2 objetos.

1.6. ***Regresión:*** Ayuda a descubrir la dependencia del valor de un atributo con respecto a otros atributos. Sea $\boldsymbol{\delta}: \boldsymbol{E} \longrightarrow \boldsymbol{S}$ el conjunto de correspondencias entre los conjuntos $\boldsymbol{E}$ y $\boldsymbol{S}$, donde $\boldsymbol{S}$ es el conjunto de valores de salida y $\boldsymbol{E}$ el conjunto de valores de entrada. Al ir acompañados de un valor de $\boldsymbol{S}$, se denominan comúnmente ejemplos etiquetados por lo que $\boldsymbol{\delta}$ es un conjunto de datos etiquetado. El objetivo es aprender una función $\boldsymbol{\lambda}: \boldsymbol{E} \longrightarrow \boldsymbol{S}$ que represente la correspondencia existente entre los ejemplos, es decir, para cada valor de $\boldsymbol{E}$ tenemos un único valor en $\boldsymbol{S}$. La diferencia con respecto a la clasificación es que $\boldsymbol{S}$ es numérico y puede ser un entero o real.

2. Técnicas Descriptivas:

El objetivo no es predecir nuevos datos si no describir características comunes, correlaciones, asociaciones y otras relaciones entre los datos existentes. Los ejemplos en este caso no están etiquetados ni ordenados y se definen como:

$$\boldsymbol{\delta_i} = \{\boldsymbol{e_i} | \, \boldsymbol{e_i} \in \boldsymbol{E}\}$$

2.1. ***Agrupamiento (Clustering):*** El objetivo de esta tarea es obtener grupos o conjuntos entre los elementos de $\boldsymbol{\delta_i}$, de tal manera que los elementos asignados al mismo grupo sean *similares*. Lo importante del agrupamiento respecto a la clasificación es que son precisamente los grupos y la pertenencia a los grupos lo que se quiere determinar, y a priori no se sabe cuántos o cómo son los grupos. En algunos casos se puede proporcionar el número de grupos que se desea obtener. Otras veces este número se determina por el algoritmo de agrupamiento según las características de los datos. La función buscada es idéntica a la de clasificación $\boldsymbol{\lambda}: \boldsymbol{E} \longrightarrow \boldsymbol{S}$, con la diferencia de que los valores de $\boldsymbol{S}$ y sus miembros se crean o se calculan durante el proceso de aprendizaje.

2.2. ***Correlaciones y Factorizaciones:*** Se asocian sólo a atributos numéricos. Además permite observar cómo se relacionan las variables entre sí, por ejemplo, si alguna variable aumenta su valor es posible que alguna otra o más variables aumenten también su valor y al contrario si disminuye el valor, las que estén relacionadas disminuirán sus valores. Sea E el conjunto que contiene a los ejemplos:

$$\boldsymbol{E} = \boldsymbol{A_1} \times \boldsymbol{A_2} \times \ldots \boldsymbol{A_n}$$

El objetivo es determinar si dos o más atributos $\boldsymbol{A_i}$ y $\boldsymbol{A_j}$ están correlacionados linealmente o relacionados de cualquier otro modo. Este tipo de relaciones son bidireccionales o no orientadas. Para ver si existe algún tipo de orientación (causa/efecto) se pueden utilizar modelos de regresión sobre esos dos atributos.

Para determinar la correlación entre 2 variables se debe calcular un *coeficiente de correlación lineal* $\boldsymbol{r}$, que mide el grado de intensidad de esta posible relación entre las variables.

$$r = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i * \sum_{i=1}^{n} y_i}{\sqrt{\left[n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2\right]\left[n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2\right]}}$$

Dónde:

r = coeficiente de correlación

n = número de pares ordenados

x = variable independiente

y = variable dependiente

2.3. ***Reglas de asociación:*** El objetivo de estas reglas es encontrar asociaciones o correlaciones entre los elementos de las bases de datos. Dados los ejemplos del conjunto:

$$\boldsymbol{E = A_1 \times A_2 \times \ldots A_n}$$

Una regla de asociación se define generalmente de la siguiente forma, si $\boldsymbol{A_i = a}$ y $\boldsymbol{A_j = b}$ ^...^$\boldsymbol{A_k = h}$ entonces $\boldsymbol{A_r = u}$ ^ $\boldsymbol{A_s = v}$^....^$\boldsymbol{A_z = w}$, donde los atributos son nominales y las igualdades se definen utilizando algún valor de los posibles por cada atributo. Un ejemplo es el siguiente donde $\boldsymbol{A_i}$ es la i-esima compra:

Si $\boldsymbol{A_n}$*="aguacates" Y* $\boldsymbol{A_m}$*="cebollas" ENTONCES* $\boldsymbol{A_j}$ *="limones"*

Es decir, si se compran aguacates y cebollas entonces también se comprarán limones. Las reglas tienen la característica de que son orientadas y por lo tanto esta regla no tiene significado en caso inverso, en este caso, si se compran limones no podemos concluir nada, también existen reglas que son bidireccionales; por ejemplo, si tenemos la siguiente regla:

Si compra="árbol de navidad" ENTONCES compra="esferas" Y si compra="esferas" ENTONCES compra ="árbol de navidad".

Por lo tanto, cumple con el caso inverso y en ambos casos se compran esferas y árbol de navidad.

Existen ciertas variantes de las reglas de asociación, algunas son:

- ***Reglas de dependencias*:** Establece relaciones funcionales entre varios atributos. Expresan patrones de comportamiento entre los datos en función a la aparición conjunta de valores de dos o más atributos.

- ***Reglas de asociación secuenciales:*** Expresan patrones de comportamiento secuencial, es decir, que las asociaciones no ocurren en el mismo momento, si no en sucesivos registros de un intervalo de tiempo.

- ***Reglas de asociación multinivel:*** Utilizan varios niveles de conceptos para expresar las relaciones por lo que es necesario proporcionar además de los datos una jerarquía de conceptos que contiene un árbol de relaciones entre los atributos. Una jerarquía de conceptos, formalmente, define una secuencia de relaciones entre conceptos más específicos a conceptos más generales. Los niveles del árbol se enumeran de arriba abajo, comenzando por el nivel superior 0 que representa todos los conceptos y finalizando en el nivel inferior donde se sitúan los ítems.

2.4. ***Dependencias funcionales:*** Las dependencias funcionales consideran todos los posibles valores. Se definen así: dados los valores $\boldsymbol{A_i, A_j, \ldots A_k}$, se puede determinar el valor de $\boldsymbol{A_r}$, que depende o está en función de los atributos $\boldsymbol{A_i, A_j, \ldots A_k}$, por ejemplo: dada la edad, el nivel de ingresos, el código postal y estado civil, es posible determinar con cierta fiabilidad, que una persona tiene vehículo.

2.5. ***Detección de valores e instancias anómalas:*** Tiene la tarea de encontrar los ejemplos que no son similares al resto. Los ejemplos se agrupan y se observan cuáles quedan desplazados de los grupos mayoritarios, si un ejemplo tiene baja probabilidad de agrupamiento con todos los grupos se puede considerar un caso aislado o anómalo, también se utilizan otros métodos que no se basan en la agrupación, como la medición de distancias (aquellas instancias donde el vecino más próximo esté muy lejos se puede considerar como una instancia anómala).

2.4 Métodos de minería de datos.

Para cada una de las tareas anteriores, existen métodos que se utilizan para generar la solución, algunos de ellos son:

- ***Métodos algebraicos y estadísticos:*** Se utilizan para expresar modelos y patrones mediante fórmulas algebraicas, funciones lineales, funciones no lineales, distribuciones o valores agregados estadísticos, tales como las medias, varianzas, correlaciones, etc. Cuando estos métodos obtienen un patrón lo hacen a partir de un modelo predeterminado donde se estiman algunos coeficientes o parámetros. Los algoritmos más conocidos dentro de este grupo son la regresión lineal (global o local), la regresión logarítmica o la regresión logística. Los discriminantes lineales y no lineales, basados en funciones predefinidas, es decir, los discriminantes paramétricos entran en esta categoría.

- ***Métodos Bayesianos:*** Se basan en estimar la probabilidad de pertenencia a una clase o grupo, mediante la estimación de las probabilidades condicionales inversas, utilizando para ello el teorema de Bayes. Los algoritmos más populares son el clasificador de Näive Bayes, los métodos basados en máxima verosimilitud y el algoritmo EM. Las redes bayesianas generalizan las topologías de las interacciones probabilísticas entre variables y permiten representar gráficamente dichas interacciones.

- ***Métodos basados en conteos de frecuencias y tablas de contingencia:*** Se basan en contar la frecuencia en la que dos o más sucesos se presenten conjuntamente. Cuando el conjunto de sucesos posibles es muy grande, existen algoritmos que van comenzando por pares de sucesos e incrementando los conjuntos sólo en aquellos casos en que las frecuencias conjuntas superen algún umbral. Un algoritmo usado con estos métodos es conocido como *"A priori"*.

- ***Métodos basados en árboles de decisión y sistemas de aprendizaje de reglas:*** Son funciones que además de su representación en forma de reglas, se basan en dos tipos de algoritmos: los algoritmos denominados "divide y vencerás" como el ID3/c4.5 o el CART y los algoritmos denominados "separa y vencerás" como el CN2. La estrategia "*divide y vencerás*" se utiliza cuando se construye o expande un nodo, se considera el subconjunto de casos de entrenamiento que pertenecen a cada clase. Si todos los ejemplos pertenecen a una clase o se verifica alguna regla de parada, el nodo es una hoja del árbol. En caso contrario, se selecciona una pregunta basada en los atributos predictivos del conjunto de entrenamiento (usando una regla de división heurística), posteriormente, se divide el conjunto de entrenamiento en subconjuntos (mutuamente excluyentes) siempre y cuando no existan valores desconocidos o que se usen, y se aplica el mismo procedimiento a cada subconjunto del conjunto de entrenamiento. En cuanto a la estrategia *"separa y vencerás"*, consiste en buscar una solución parcial al problema (una regla en este caso) y una vez encontrada, reducir el problema eliminando todos los ejemplos cubiertos por la solución encontrada.

- ***Métodos relacionales, declarativos y estructurales:*** La característica principal de este conjunto de técnicas es que representan los modelos mediante lenguajes declarativos, como lenguajes lógicos, funcionales y lógicos-funcionales.

- ***Métodos basados en redes neuronales artificiales:*** Se trata de técnicas que aprenden un modelo mediante el entrenamiento de pesos que conectan un conjunto de nodos o neuronas. La topología de la red y los pesos de las conexiones determinan el patrón aprendido.

- ***Métodos basados en núcleo y máquinas de soporte vectorial:*** Intentan maximizar el margen entre los grupos o clases formadas. Para ello se basan en una transformación que puede aumentar su dimensión. Estas transformaciones se llaman kernels o núcleos.

- ***Métodos estocásticos o difusos:*** Se incluyen la mayoría de las métodos que, junto con las redes neuronales, forman lo que se le denomina computo flexible (soft computing). Son métodos donde los componentes aleatorios son fundamentales, como el enfriamiento simulado (simulated annealing), los

métodos evolutivos y genéticos, o bien al utilizar funciones de pertenencia difusas (fuzzy).

- ***Métodos basados en casos, en densidad o distancia:*** Son métodos que se basan en distancia al resto de elementos, ya sea directamente, como los vecinos más cercanos, de una manera más sofisticada mediante la estimación de funciones de densidad. Además de los vecinos más cercanos, otros algoritmos conocidos son los jerárquicos, como Two-step o COBWEB y los no jerárquicos como K-medias (k-means).

2.5 Técnicas usadas en este proyecto.

En esta sección se describen con más detalle las técnicas y métodos que se utilizan en este trabajo. Como se describe más adelante, estos métodos se emplearán para modelar las reglas que definen el comportamiento del agente.

2.5.1 Reglas de Asociación.

Una regla de asociación es una proposición probabilística sobre la ocurrencia de ciertos atributos en una base de datos. La principal característica de esta técnica es que trata con atributos nominales, estos representan características no cuantificables, es decir, representan valores como color, sexo, preferencias de marca, entre otros (Witten, 2005). Las reglas expresan la combinación de valores de los atributos que se repiten con frecuencia. Los ejemplos clásicos del uso de esta técnica son el análisis sobre la compra de productos en un supermercado, estudio de textos, búsqueda de patrones en páginas web, entre otros. La forma de una regla típica de asociación es la siguiente:

Si concierto = “divertido” ***Y*** gente = “alocada” ***ENTONCES*** banda = “buena”

Las sentencias tienen una sintaxis específica con el fin de que la oración no sea muy larga, haciendo posible la definición en una variable, la oración que representa la acción y la omisión del signo de igualdad, implica que tendremos un enunciado más corto y fácil de entender. Tomando el ejemplo anterior:

Definimos c(x), g(x) y b(x), estos representan la acción concierto, gente y banda, entonces nuestra sentencia es más pequeña y queda así:

Si c(divertido) Y g(alocada) ENTONCES b(buena)

Usualmente una regla de asociación puede ser vista como reglas de la forma SI $\alpha \rightarrow \beta$ donde α y β son dos conjuntos de objetos disjuntos. Otra forma utilizada para expresar una regla de asociación es $\beta \leftarrow \alpha$. El conjunto α recibe el nombre de predecesor de la regla y β se le denomina sucesor o consecuente (Hernández, Ramírez y Ferri, 2007).

Para conocer la calidad de la regla se utilizan dos medidas: soporte y confianza. El soporte es también llamado cobertura y se define como el número de instancias que la regla predice

correctamente, es decir, el soporte ($A \longrightarrow B$) representa los registros contenidos en alguna tabla que contiene los elementos A y B simultáneamente, el soporte se define como: $soporte(\mathrm{A} \longrightarrow \mathrm{B}) = \boldsymbol{P}(\boldsymbol{A} \cup \boldsymbol{B})$. La confianza o precisión mide el porcentaje de veces que la regla se cumple y se define como: $confianza(\mathrm{A} \longrightarrow \mathrm{B}) = \boldsymbol{P}(\boldsymbol{B} \mid \boldsymbol{A})$.

Existen gran variedad de tipos de reglas de asociación, presentaremos una clasificación de familias de reglas basadas en los siguientes criterios:

- ***Tipos de los valores utilizados en las reglas:*** Es posible tener reglas que trabajan con atributos binarios, indican la presencia o ausencia de un objeto y se les conoce como reglas booleanas.

- ***Dimensiones de los datos:*** Dada la siguiente regla:

  SI compra = "pescado" ENTONCES compra = "vino"

  Esto se refiere a que trabaja con sólo una dimensión que es compra, se puede incrementar la dimensión si incluimos la dimensión de fecha o cliente, por ejemplo una regla multidimensional sería así:

  SI compra = "pescado", cliente = "Juan", Fecha = "Abril" ENTONCES compra = "vino".

- ***Niveles de abstracción:*** Algunos sistemas o algoritmos permiten incorporar a las reglas diferentes niveles de abstracción representados por conceptos que añaden otros objetos. Este tipo de reglas se conocen como reglas multi-nivel, por ejemplo:

  $$\Big(\mathrm{Edad}(\mathrm{x}, 27) \wedge \big(\mathrm{Compra}(\mathrm{x}, \mathrm{laptop}) \rightarrow \mathrm{Compra}(\mathrm{x}, \mathrm{impresora})\big)\Big)$$

  El ejemplo anterior contiene tres predicados que solo ocurren una vez.

- ***Reglas instantáneas o secuenciales***: Depende si se consideran relaciones en un instante de tiempo, una secuencia o serie.

- ***Aprendizaje de las reglas de asociación:*** El aprendizaje de reglas de asociación se basa en su confianza y soporte. Los algoritmos de aprendizaje trabajan en la búsqueda de reglas que cumplan requisitos mínimos en estas medidas, la tarea de buscar patrones que cumplan estos requisitos es costosa, ya que los conjuntos de objetos al ser analizados crecen exponencialmente con respecto al número de variables de los datos; sin embargo, en los casos reales existen pocos conjuntos frecuentes y los métodos que exigen una confianza o soporte mínimo se benefician de este hecho.

Un método usado para encontrar reglas de asociación es el algoritmo *a priori*, el cual obtiene el conjunto de objetos más frecuente, busca los objetos con determinado soporte y emplea una aproximación iterativa.

Algoritmo a priori: Se basa en el conocimiento previo o "a priori" de los conjuntos frecuentes, esto sirve para reducir el espacio de búsqueda y aumentar la eficiencia. Fue propuesto por (Agrawal & Srikant, 1994). El algoritmo se resume en dos pasos:

1. Se generan todos los itemsets frecuentes que contienen un elemento (un itemset representa los conjuntos de pares atributo-valor llamados ítems, que cubran gran cantidad de instancias), después se genera los itemsets que contengan 2 ítems y así sucesivamente. Se toman todos los posibles pares de ítems que cumplen con las medidas básicas de soporte inicialmente preestablecidas, esto permite ir eliminando posibles combinaciones: aquellas que no cumplan con los requerimientos de soporte no entrarán en el análisis.

2. Se generan las reglas revisando que cumplan con el criterio mínimo de confianza, en caso de tener una conjunción de consecuentes de una regla y ésta cumple con los niveles mínimos de soporte y confianza, sus subconjuntos (consecuentes) también se cumplirán; en el caso contrario, si algún ítem no los cumple no se considerarán sus subconjuntos.

Un ejemplo sencillo sería sobre la compra de productos de un cliente, si éste añade un objeto a la cesta se pone un 1 y de lo contrario un 0. Una transacción representa una compra hecha por algún cliente (TID) y los productos que ha adquirido se muestran con un 1 sobre la tabla 1:

TID	Vino	Soda	Agua	Horchata	Bizcocho	Galleta	Chocolate
T1	1	1	0	0	0	1	0
T2	0	1	1	0	0	0	0
T3	0	0	0	1	1	1	0
T4	0	0	0	1	1	1	1
T5	0	0	0	0	0	1	0
T6	1	0	0	0	0	1	1
T7	0	1	1	1	1	0	0
T8	0	0	0	1	1	1	0
T9	1	1	0	0	1	0	1
T10	0	1	0	0	1	0	0

Tabla 1: Tabla de la cesta de compra

En la tabla 2 existen siete conjuntos que contienen solo un producto los cuales son:

Itemset	Soporte
{Vino}	3
{Soda}	5
{Agua}	2
{Horchata}	4
{Bizcocho}	6
{Galleta}	6
{Chocolate}	3

Tabla 2: Soporte para itemset de 1 elemento

Tomemos la cobertura mínima igual a 2; es decir, los productos que han sido comprados en las transacciones aparecen por lo menos 2 veces en la tabla con el valor 1. Después se generan en base a los conjuntos de un elemento los conjuntos formados por 2 elementos, de las $\binom{7}{2}$=7!/5!=42 posibles combinaciones se tienen 15 que poseen un soporte distinto de cero. Se puede observar en la tabla 1 que el itemset {Vino, Soda} aparece 2 veces, éste cumple el soporte. En la tabla 3 se muestra el soporte para los itemsets de 2 elementos que es distinto de 0.

Itemset	Soporte
{Vino, Soda}	2
{Soda, Agua}	2
{Horchata, Bizcocho}	4
{Vino, Galleta}	2
{Bizcocho, Chocolate}	2
{Soda, Bizcocho}	3
{Vino, Bizcocho}	1
{Soda, Chocolate},	1
{Bizcocho, Galleta}	3
{Soda, Galleta}	1
{Galleta, Chocolate},	2
{Agua, Bizcocho},	1
{Galleta, Chocolate},	2
{Soda, Horchata},	1
{Horchata, Galleta},	3
{Vino, Chocolate}	2

Tabla 3: Soporte para itemsets de 2 elementos

Continuando tendremos las combinaciones para itemsets de tres elementos de $\binom{7}{3} = 7!/\,4! = 210$, los conjuntos son demasiados para listarlos, pero se mostrarán los que tienen soporte distinto de cero en la tabla 4.

Itemset	Soporte
{Vino, Soda, Galleta}	1
{Horchata, Bizcocho, Galleta}	3
{Horchata, Galleta, Chocolate}	1
{ Bizcocho, Galleta, Chocolate}	1
{Vino, Galleta, Chocolate}	1
{Soda. Agua, Horchata}	1
{Agua, Horchata, Bizcocho}	1
{Soda, Agua, Bizcocho}	1
{Vino, Soda, Bizcocho}	1
{Vino, Soda, Chocolate}	1

{Soda, Bizcocho, Chocolate}	1

Tabla 4: Soporte para Itemsets de 3 elementos

Si se sigue iterando, sólo se tienen 2 conjuntos de 4 objetos {Horchata, Bizcocho, Galleta, Chocolate}, {Vino, Soda, Bizcocho, Chocolate}, pero ninguno de ellos cumple con el soporte mínimo que es 2, por lo que no se utilizan y usamos los itemsets que cumplan con el soporte requerido.

Posteriormente de que se seleccionaron los itemsets que cumplen con la cobertura mínima, el siguiente paso consiste en extraer de estos conjuntos de reglas las que tengan un nivel de confianza mínimo. Por ejemplo, para el conjunto de objetos {Horchata, Bizcocho y Galleta} podemos extraer las siguientes reglas de asociación.

El soporte o cobertura se representa con Cb y la confianza con Cf sobre cada regla de la tabla 5. Recordemos que *cf(A⟶B) = soporte(AUB)/soporte(A)*, por ejemplo para la primer regla de la tabla 5 necesitamos lo siguiente:

$$soporte = \frac{\text{número de veces en que el conjunto o elemento tienen 1 en la transacción}}{\text{número total de transacciones}}$$

Entonces tenemos que:

$$\text{soporte}(\{\text{Horchata, Bizcocho}\}) = 4$$

$$confianza(\{Bizcocho, Horchata\} \rightarrow \{Galleta\}) = \frac{\text{soporte}(\{\text{Bizcocho, Horchata}\} \cup \{\text{Galleta}\})}{\text{soporte}(\{\text{Bizcocho, Horchata}\})}$$

Por lo tanto:

$$cf = \frac{\text{soporte}(\{\text{Bizcocho, Horchata}\} \cup \{\text{Galleta}\})}{\text{soporte}(\{\text{Bizcocho, Horchata}\})} = 3/4$$

Sucesivamente se hace el mismo procedimiento para las reglas restantes que se muestran en la tabla 5.

Si compra “bizcocho” Y compra “horchata” ENTONCES compra “galleta”	Cb=4	Cf=3/4
Si compra “bizcocho” Y compra “galleta” ENTONCES compra “horchata”	Cb=3	Cf=3/3
Si compra “galleta” Y compra “horchata” ENTONCES compra“ bizcocho”	Cb=3	Cf=3/3
Si compra “galleta” ENTONCES compra “bizcocho” Y compra “horchata”	Cb=6	Cf=3/6
Si compra” bizcocho” ENTONCES compra “galleta” Y compra “horchata”	Cb=6	Cf=3/6
Si compra “horchata” ENTONCES compra “bizcocho” Y compra “galleta”	Cb=4	Cf=3/6
Si Φ ENTONCES compra “bizcocho” Y compra “galleta” Y compra “horchata”	Cb=10	Cf=3/10

Tabla 5: Soporte y confianza de las regla generadas

La siguiente fase consiste en la creación de reglas a partir de los conjuntos de objetos frecuentes. Si buscamos reglas de asociación de un objeto en la parte derecha del proceso es sencillo: de un conjunto de objetos de tamaño ***i***, se crean ***i*** reglas colocando siempre un único objeto diferente en la parte derecha, por ejemplo, sea el siguiente conjunto de objetos {horchata, bizcocho, galleta}, se construyen las reglas en la tabla 6:

Si compra “bizcocho” Y compra “horchata” ENTONCES compra “galleta”	Cb=4	Cf=3/4
Si compra “bizcocho” Y compra “galleta” ENTONCES compra “horchata”	Cb=3	Cf=3/3
Si compra “galleta” Y compra “horchata” ENTONCES compra “bizcocho”	Cb=3	Cf=3/3

Tabla 6: Reglas para itemset

Para trabajar con reglas que tengan más de un objeto en la parte derecha se puede separar de la siguiente manera, por ejemplo:

Si compra =“bizcocho” Y compra = “horchata” ENTONCES compra = “galletas” Y compra = “vino”.

Esta regla se divide en dos partes con un objeto en la parte derecha:

- Si compra “bizcocho” Y compra “horchata” ENTONCES compra “vino”
- Si compra “bizcocho” Y compra “horchata” ENTONCES compra “galletas”.

El soporte de ambas reglas será idéntico a la regla original por tener la misma parte izquierda. En cuanto a la confianza, se puede asegurar que será igual o mayor a la regla con dos objetos en la parte derecha, ya que el número de registros que cumplan la parte izquierda en donde aparezca *“vino”* será mayor o igual al número de registros que cumplan la parte izquierda en los que aparezca “*vino y galletas”*.

2.5.2 Agrupamiento (Clustering).

El agrupamiento tiene la tarea de formar grupos de objetos con características similares, a estos grupos se les conoce como clases o clústeres. Es una técnica no supervisada y los resultados dependen de la entrada de datos que estemos utilizando.

Para encontrar clústeres similares, los objetos necesitan ser comparados y medir su similitud, para ello se utilizan diferentes métricas, tales como la distancia Euclidiana, distancia de Minkowski o Manhattan. Adicionalmente las distancias se pueden normalizar.

El algoritmo ***k***-medias es utilizado para construir agrupamientos refinados, toma como entrada un número predefinido de clústeres al que llamaremos ***k***. Utiliza un centroide, éste representa un punto en el espacio de objetos con una posición promedio en el grupo, por otra parte, el centroide se calcula obteniendo el promedio de los miembros del grupo.

Asumiendo que los datos son representados como una tabla relacional, cada renglón representa un objeto y cada columna representa un atributo. Por ejemplo, tenemos un conjunto de medidas que representan la altura de algunas personas y queremos saber quién es alto o quién es bajo, en la tabla 7 se muestra el registro de las alturas:

Persona	Altura (cm)
1	195
2	166
3	188

4	195
5	179
6	198
7	161
8	179
9	200
10	191

Tabla 7: Tabla de alturas

Lo siguiente es agruparlos aleatoriamente donde k=1, ver figura 8:

Persona	Altura (cm)	Clase
1	195	○
2	166	□
3	188	□
4	195	○
5	179	□
6	198	□
7	161	□
8	179	○
9	200	□
10	191	○

Grupo 1 ○

Grupo 2 □

Tabla .8: Representación de clases

Centroide ○ : $\frac{195+195+179+191}{4} = 190$

Centroide □ : $\frac{166+188+198+161+179+200}{6} = 182$

El algoritmo es un procedimiento simple e iterativo y concluye con la generación de los clústeres. En cada iteración calcula un centroide más preciso y las coordenadas de este punto son aproximadas a los valores de los atributos de todos los objetos que pertenecen al grupo. El proceso iterativo redefine y reasigna los datos al clúster, finalmente necesita un pequeño número de iteraciones para su convergencia.

Como es iterativo calculamos las distancias euclidianas:

$$\boldsymbol{d = (p1 - p2)^2}$$

Tomemos los valores más grandes del grupo 1 y 2: 200 y 198 por lo que tenemos:

$\boldsymbol{d\_g1 = (200 - 190)^2} = \mathbf{10^2} = 100$ y $\boldsymbol{d\_g2 = (200 - 182)^2} = \mathbf{18^2} = 324$

$\boldsymbol{d\_g1 = (198 - 190)^2} = \mathbf{8^2} = 64$ y $\boldsymbol{d\_g2 = (198 - 182)^2} = \mathbf{16^2} = 254$

Valor	d_g1	d_g2
200	100	324
198	64	254

Tabla 9: Distancias euclidianas de los grupos

En la tabla 9 se muestran los cálculos y por lo tanto $\boldsymbol{d\_g1} < \boldsymbol{d\_g2}$, entonces 200 y 198 ahora pertenecen al grupo1 y así sucesivamente se vuelve a calcular para todos los elementos.

El pseudocódigo de este algoritmo es el siguiente:

1. Selecciona aleatoriamente k puntos los cuales iniciarán como centroides para cada k-esimo clúster.
2. Asigna a cada objeto el centroide más cercano formando k grupos.
3. Calculamos centroides nuevos, tomando el promedio de todos los valores de atributos de los objetos iniciales de algún grupo.
4. Verificar si el centroide del grupo ha cambiado sus coordenadas, si ha cambiado regresar al paso 2.
5. Si no, la detección del grupo ha finalizado y todos los objetos tienen a sus miembros definidos

Algunas desventajas del algoritmo k-medias aparecen cuando las agrupaciones son de diferente tamaño o densidad. Además es posible que genere clústeres vacíos y tenga problemas con valores atípicos (valores diferentes al resto de los datos).

2.6 Minería de la Web.

Dentro del área de la minería de datos existe una rama llamada *minería de la web*. La minería de la web, tal como su nombre lo indica se refiere a la aplicación de las técnicas de minería sobre los datos que se encuentran o generan a partir de la web, por lo que puede ser definida como el descubrimiento y análisis del uso de la información de la web (Akerkar, R, 2008). Los tipos de datos que están asociados a la web, según (Srivastava, 2000), pueden clasificarse de esta manera:

- ***Contenido:*** Elementos que se encuentran en las páginas web, como texto, imágenes, audio, video, hipervínculos y metadatos.
- ***Estructura:*** Datos asociados a la organización del contenido. Esto incluye la estructura de etiquetas contenidas en los html y xml. El primer tipo de información de la estructura de interpágina es cuando un hipervínculo nos lleva de una página a otra.
- ***Uso:*** Datos secundarios, como los que se encuentran en las bitácoras de los servidores web, bitácoras de los servidores proxy, perfiles de usuario, cookies, entre

otros; es decir, toda la información producida de la interacción de los usuarios sobre la web.

- ***Perfil de usuario:*** Datos que indiquen información demográfica; por ejemplo, los datos de un registro de usuario en un sitio web, detalles de su perfil, entre otros.

La minería de la web se divide en tres grandes categorías según el tipo de datos que toma en cuenta:

- ***Minería del contenido de la web:*** Es el proceso que consiste en la extracción de conocimiento a partir del contenido de documentos o sus descripciones.

- ***Minería de la estructura de la web:*** Consiste en encontrar un modelo de la estructura de las ligas en la web y clasificar las páginas, con esto se puede encontrar la relación entre varios sitios web, además de analizar la estructura de los enlaces y determinar problemas en la navegación.

- ***Minería del uso de la web:*** Se define como el estudio de los datos de sesiones y comportamientos generados en la web, trabaja principalmente con datos secundarios que resultan de la interacción de los usuarios con la web.

2.7 Minería del uso de la web.

Los datos con los que trabaja la minería del uso de la web son los eventos generados en las interfaces web. Estos eventos pueden ser capturados y almacenados en bitácoras con un formato muy parecido a las que crean los servidores web. Generalmente son archivos de texto, los cuales llevan el registro de la interacción de los usuarios con un conjunto de páginas visitadas, hora de acceso, identificador de usuario. Al hacer el análisis a este tipo de bitácoras podemos encontrar conocimiento que puede ayudarnos a identificar información relacionada con la administración de nuestro sistema, observar cuáles son las páginas más visitadas, saber la hora donde los usuarios visitan el sistema, etc.

La minería del uso de la web es usada en estos sistemas, ya que apoyan a la toma de decisiones, crean una organización más clara y útil para los sitios, hacen más eficiente el comercio electrónico y se pueden generar recomendaciones para una mejor comprensión de sus visitantes.

(Mobasher, 2000) propone que el proceso general de la personalización basada en la web se compone de tres fases: preparación y transformación de datos, descubrimiento de patrones y recomendación.

En la fase de preparación y transformación de datos se utilizan las bitácoras del servidor web o transacciones de datos que provienen directamente de la web, también se realiza la integración de datos de múltiples recursos como: bases de datos y aplicaciones del servidor. Para la fase de descubrimiento de patrones podemos utilizar varias técnicas de minería de datos dependiendo cual es el resultado que se quiera obtener podemos usar: reglas de asociación, clasificación, descubrimiento de patrones secuenciales y modelado

probabilístico. Los resultados de las dos fases anteriores se agregan al modelo de usuario para un uso adecuado en la última fase. Por último, en la fase de recomendación en base a la información recopilada se generan las sugerencias.

2.7.1 Fuente de datos para la minería del uso de la web.

Como se mencionó anteriormente, el principal material de la minería de la web son los eventos generados por el usuario, aunque en algunos casos en la etapa de preparación y descubrimiento de patrones se incluyen los archivos del sitio y sus metadatos.

La bitácora que debe procesar la minería del uso de la web contiene algunos de los elementos de una bitácora común del servidor web, cada entrada en la bitácora de un servidor web corresponde a una petición HTTP. El servidor web se encarga de administrar estas bitácoras que generalmente usan un formato estándar, que incluye:

1. IP del cliente
2. El nombre del usuario
3. Fecha y hora
4. El recurso solicitado vía GET o POST
5. La respuesta del servidor
6. Bytes transmitidos

Un ejemplo de una entrada de estas bitácoras sería la siguiente:

123.456.789.11 usuario [06/05/2012:10:34:21 -0300] "GET index.html HTTP/1.0" '200' 5678

Estos datos deben pasar por una fase de pre-procesamiento que les asignará distintos niveles de significado. En la minería del uso de la web el nivel de significado básico es la *visita a la página web*. Una sesión es la representación de una colección de objetos web o recursos al generarse un evento, cuando el usuario visita algún recurso sobre la interfaz; por ejemplo, la adición de un producto al carrito de compras (Mobasher, 2005).

2.7.2 Preprocesamiento de los datos.

La meta del preprocesamiento de datos es reconstruir la sesión del usuario porque los datos pueden presentar inconsistencias, por ejemplo, pueden contener valores nulos o vacíos. Al analizar el flujo de los eventos generados por cada usuario se puede observar que esta secuencia o conjunto de ligas representan la sesión de usuario. Es común que se utilicen algoritmos y técnicas heurísticas para llevar a cabo el pre-procesamiento de acuerdo a los datos que se desean observar, en el caso de las sesiones, generalmente se divide toda la actividad del usuario registrada en la bitácora de acuerdo a algún criterio, ya sea temporal o a partir del uso de algún recurso específico como la página de inicio de sesión.

Algunos de los problemas que aparecen en el preprocesamiento son que la información puede estar incompleta o ser nula en alguna entrada; pueden contener ruido, en este caso la secuencias pueden ser alteradas por una mala recepción o envío del mensaje.

Existen varios métodos para recuperar los datos de una sesión (Mobasher, 2005), estos son:

Limpieza de datos: Remueve referencias a objetos que no generan eventos como los gráficos, archivos de estilo y archivos de sonido.

Fusión de datos: Se refiere a la mezcla de las bitácoras generadas por múltiples aplicaciones web.

Identificación de sesión: Es el proceso de agregación de una colección de páginas. Este proceso depende de la estructura y contenidos del sitio que visita el usuario.

La mayoría de los recursos pueden ser heurísticamente inferidos a través de un proceso llamado identificación del camino, usando el conocimiento de la estructura del sitio y la información referente a los registros de los usuarios alojados en la bitácora en el servidor.

La identificación de las sesiones es el proceso de agrupación de una colección de recursos o páginas que serán consideradas como la unidad atómica del análisis en cuestión. Este proceso es directamente dependiente de la estructura de los vínculos del sitio así como también su contenido. El nivel de abstracción capturado en una sesión es en parte determinado por el conocimiento del sitio y el análisis requerido.

Generalmente la identificación heurística de sesiones se clasifica en dos tipos: orientada en tiempo y orientada en estructura. La heurística orientada en tiempo se aplica de forma global o local en base a estimaciones de tiempo para distinguir las sesiones mientras que, la heurística orientada en estructura usa la estructura del sitio estático o la estructura de vinculación implícita capturando los campos referidos en la bitácora del servidor.

2.7.3 Minería del uso de la web para la personalización.

En el proceso de construcción de una IUA se mencionó que las teorías de personalización de la web e hipermedia adaptativa deben recoger los datos de la interacción entre el usuario y la interfaz, en este caso, obtenemos los eventos que producen los usuarios**.**

Para construir el perfil de usuario para este tipo de sistema se deben cubrir dos etapas. La primera es que el sistema debe determinar los intereses del usuario, esta tarea puede llevarse a cabo a partir de la observación de la navegación del usuario y con el apoyo de diversas heurísticas clasificar los recursos interesantes para el usuario, también puede ser a criterio del usuario asignar las calificaciones a los elementos del sitio de forma manual.

Y en la segunda etapa se debe construir los diferentes perfiles de usuario en los sistemas de filtrado colaborativo, se utiliza el perfil de usuario activo que se comparará con los perfiles de otros usuarios. El elemento central es buscar la similitud de atributo por atributo de tal manera que se vaya obteniendo información suficiente para hacer la recomendación. Los perfiles se representan como un vector o un conjunto de clasificaciones basado en las

preferencias del usuario en un subconjunto de elementos. Un perfil de usuario activo se utiliza para encontrar a otros usuarios con preferencias similares.

2.8 Resumen.

La minería de datos es una potente herramienta para descubrir la información oculta usando los datos recolectados de la interacción del usuario con la interfaz. Los patrones encontrados a través de las técnicas de minería de datos, dan la pauta para que los analistas puedan encontrar errores en la aplicación, errores de sesión, encontrar comportamiento de navegación caótica, entre otros y de esta manera generar la solución para corregirlos.

Las técnicas de la minería de datos son de gran interés, ya que nos ayudan a clasificar, agrupar, predecir valores futuros, entre otras cosas. Son una herramienta que genera con gran precisión patrones que son de importancia, utilizando métodos estadísticos avanzados que usan árboles de decisión, redes neuronales, reglas de asociación, algoritmos como k-means, Näive Bayes, entre otros.

La minería de datos tiene varias áreas, pero dada nuestra fuente datos que proviene de la web es necesario utilizar minería de la web, ésta a su vez tiene una sub-rama llamada minería del uso de la web, que trabaja principalmente con datos secundarios que resultan de la interacción de los usuarios con alguna aplicación.

Los datos que se utilizan en minería del uso de la web provienen de bitácoras de servidores web y es necesario un pre-procesamiento en los datos para eliminar errores como valores vacíos o nulos, además de valores que puedan alterar los resultados.

Capítulo 3

Descripción general del análisis sobre el comportamiento de navegación del usuario.

En este capítulo se explica la primera parte de la construcción de una interfaz autónoma adaptativa desde el marco práctico basado en las teorías expuestas a lo largo del capítulo 1 y 2. Las técnicas descritas sobre la personalización web han dado paso a que se busque una forma más completa de obtener información del usuario, se mostrará con más detalle como obtener los perfiles del usuario y como el progreso en las tecnologías web permiten el uso de herramientas más sofisticadas para la obtención de información sobre los elementos interactivos de una página web.

Los datos obtenidos de los usuarios serán procesados y deben cumplir con ciertos estándares para utilizarlos de manera más eficiente. Estos datos contienen patrones que revelan información desconocida e importante, como consecuencia reducen el tiempo de búsqueda y análisis.

Existen varias herramientas que analizan bitácoras y reconstruyen la navegación del usuario mediante gráficas, cada nodo de la gráfica simula la transición de una página a otra. Son usadas para arreglar o detectar errores de navegación y el análisis de alguna aplicación web. Otra herramienta muy recurrente se centra en la minería de uso de la web que permitirá encontrar los patrones que producen los usuarios a partir de la interacción de una aplicación web.

Utilizaremos la herramienta Weka para construir los clústeres, su análisis ayudará a definir el comportamiento del agente. Los clústeres más representativos revelan el comportamiento o navegación que es más recurrente entre los usuarios de la plataforma. En este capítulo se mostrará cómo se procesan los datos con Weka, la cual cuenta con un grupo de algoritmos que son usados para la clasificación de los datos.

Uno de los elementos centrales del análisis del comportamiento del usuario es la reconstrucción de una sesión de un usuario del sistema Moodle, para ello se generará una bitácora de uso personalizada, se guardará la información en una base de datos y se procesará para darle formato, posteriormente se calcularán los clústeres y se podrá relacionar al usuario con alguno de éstos.

3.1 Descripción de la idea general de la captura del comportamiento del usuario.

Para usar los métodos contenidos en la personalización de la web vista en el capítulo 1, es necesario obtener la interacción de los usuarios, se sabe que las páginas están escritas en HTML y que la interacción se lleva a cabo a través de artefactos interactivos como vínculos y elementos denominados *input* que suelen ser: botones, campos de texto, listas de selección, checkbox, entre otros. Estos artefactos pueden revelarnos una interacción más detallada, ya que obtienen toda la información acerca de los eventos que se generan en la interfaz.

Obtener la información de los eventos no es difícil, se usa una herramienta muy popular llamada jQuery, esta herramienta es una biblioteca de JavaScript creada inicialmente por John Resig, permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM (W3C, 2004) del documento, manejar eventos, desarrollar animaciones y agregar contenido dinámico a páginas web, es un software libre y de código abierto.

Para capturar la interacción del usuario se genera un script a partir de las funciones de notificación de eventos que son parte del modelo de manejo de eventos de jQuery, algunos de los eventos más usados son: clic, onmouseover, entre otros. Un ejemplo de la sintaxis que debemos usar en este script es el siguiente:

```
$("#target").click(function() {
alert( "Handler for .click() called." );
});
```

Donde "#target" es el elemento que pasará la información del evento clic, generalmente es un input de tipo vínculo o botón. De esta manera al hacer clic sobre un vínculo se genera una notificación que podemos capturar, en este caso podemos usar la información del evento para obtener el nombre del vínculo o su referencia, la hora y fecha en que ocurrió el evento, entre otras cosas.

Estos registros pueden guardarse para poder ser procesados posteriormente. Para este trabajo se decidió guardar la información en una bitácora con formato muy parecido a la de los servidores web, para ello se ocupó una herramienta llamada Log4j que genera bitácoras personalizadas independientes a las del servidor web.

Log4j cuenta con un módulo muy útil para especificar las características que deseamos reportar a través de un archivo de configuración, estos campos son: IP del usuario, el tipo de evento que se produjo, la liga del artefacto en que se disparó el evento, la fecha y la hora. En este mismo archivo de configuración se puede especificar si se requiere guardar la información en un archivo o en una base de datos, para este caso, se decidió que la información se almacenara directamente en una base de datos.

3.2 Propuestas actuales para el análisis de la navegación web.

Dentro del análisis de la navegación web es necesario conocer la secuencia de vínculos que ha visitado el usuario, a esto se le conoce como flujo de clics o *clickstream*; en este proyecto el flujo de clics hace referencia al conjunto de elementos interactivos por los que ha pasado el ratón, es decir, no sólo se tienen clics sino que además se tienen otros tipos de eventos en el flujo.

La interacción del usuario con la web generalmente es capturada en bitácoras, éstas son creadas por el servidor web y son archivos de texto donde se detallan todas las peticiones que se hacen a él, el formato de estos archivos es un estándar que se describirá más adelante.

En este capítulo se explica el funcionamiento de varias herramientas para analizar la navegación web que los usuarios han llevado a cabo, éstas muestran la construcción de gráficas representando visualmente la sesión del usuario describiendo los vínculos que ha visitado, la hora en que lo visitó, el número de veces que lo ha visitado, entre otros. Las gráficas contienen propiedades como color para ayudar a identificar algún movimiento especial por parte del usuario.

Una sesión de usuario es un conjunto de datos que describen la interacción del usuario en un determinado tiempo. Una sesión se puede reconstruir a partir del flujo de clics, pero al reconstruir la sesión puede haber algunos huecos al seguir la pista de las actividades del usuario, considerando esto se deben proponer algunas estrategias que permitan detectar el inicio y fin de una sesión; es decir, poner un límite a la sesión si la inactividad del usuario es evidente. En este caso una sesión se referirá a toda la actividad que hay durante 24 horas.

En este caso se usarán las bitácoras con los datos generados por los usuarios de Moodle, estas bitácoras contienen información sobre el uso de esta interfaz y se usarán los datos contenidos en los campos día, hora, recurso y eventos para construir la sesión del usuario.

3.2.1 Análisis de la navegación mediante herramientas basadas en artefactos interactivos.

Pathalizer es una herramienta que muestra la navegación del usuario de forma gráfica. A esta herramienta se le indican las bitácoras del sitio que se quieren analizar. Una bitácora normalmente incluye la información de todas las peticiones que se realizan al servidor web, pero muchos de estos datos no son relevantes para el análisis de la navegación del usuario, así que a través de la herramienta se hace un filtrado de las peticiones con respecto a los tipos de archivos que provee un servidor; por ejemplo, filtrar del análisis las peticiones y

respuestas del servidor web, imágenes y hojas de estilo. Puede descargarse de la siguiente página: *http://pathalizer.sourceforge.net/*.

El siguiente ejemplo permite observar el funcionamiento de esta aplicación. Los datos de entrada utilizados se extraen de una bitácora de ejemplo del propio sitio web de Pathalizer.

Se da clic en el botón añadir y se elige el archivo que contiene las bitácoras, justo como se muestra en la figura 5:

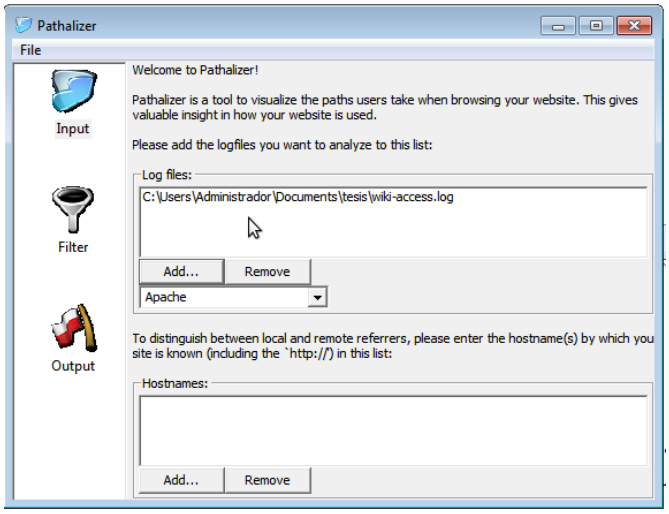

Figura 5: Bitácora para Pathalizer

En la pestaña ***Filter*** se personaliza qué tipo de registros se toman en cuenta y cuáles no; por ejemplo, elementos de la aplicación que no aportan nada durante el análisis de descubrimiento de patrones son: imágenes, css, entre otros. En tal caso Pathalizer no toma en cuenta los recursos que tienen terminación distinta a los filtros siguientes: `ico`, `png`, `jpg`, `gif` y `css`, los cuales ya trae por defecto, pero se pueden añadir otros en su configuración. Por ejemplo, se tienen los siguientes dos registros en una bitácora:

68.251.52.253 - - [19/Jun/2005:06:50:50 +0200] "GET /favicon.ico HTTP/1.1" 200 3262 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.8) Gecko/20050511 Firefox/1.0.4"

61.177.31.179 - - [19/Jun/2005:06:52:36 +0200] "GET /wximages/wxwidgets02-small.png HTTP/1.1" 200 12468 "http://blog.vckbase.com/bastet/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)"

Por otra parte, permite personalizar el número máximo de aristas que se mostrarán en la gráfica, dónde las aristas representan el cambio de una página a otra, lo que permite que la gráfica no sea muy grande y de esta manera llevar un análisis más puntual sobre la parte de la aplicación que es de nuestro interés. En la pestaña de *Filter* podemos personalizar el número de aristas y de nodos que la gráfica tendrá, en la figura 6 se muestran estas opciones.

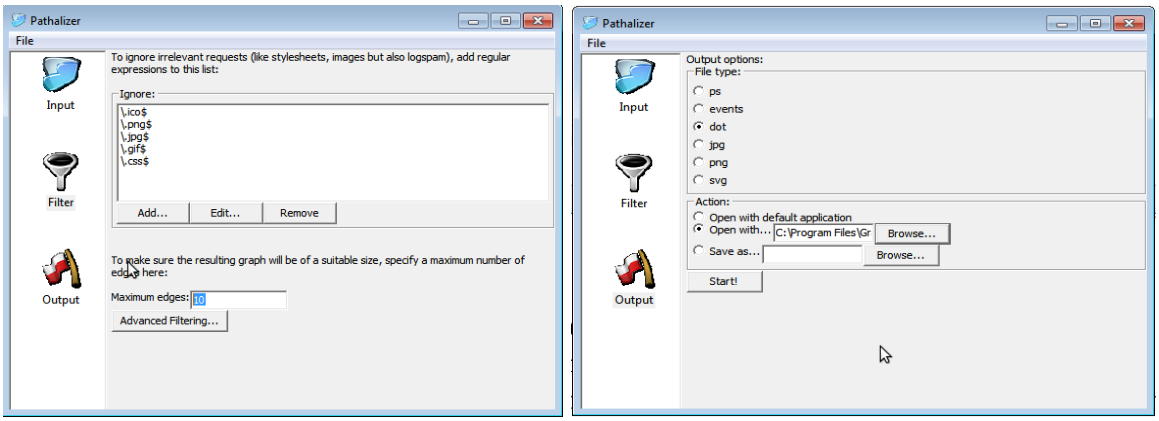

Figura 6: Personalización de la gráfica

Continuando sobre la pestaña de ***output*** se selecciona la opción *dot* el cual es un lenguaje para definir las reglas de construcción de una gráfica, enseguida en la sección ***action*** pulsamos sobre ***"abrir con Gvedit",*** en este paso se utilizan estas opciones para personalizar colores y tamaño de los nodos de la gráfica, ya que si damos la opción de jpg o png la gráfica se verá borrosa y no se distinguirán los elementos.

Para que la aplicación genere la gráfica es necesario presionar el botón start. Pathalizer utiliza la bitácora e identifica los recursos que visita el usuario y crea una gráfica dirigida que representa la vista de cada recurso de manera jerárquica, se representan las páginas que tienen alta demanda así como los recursos más populares del sitio, por último muestra un contador para ver cuántas veces se visitó esa página.

Un ejemplo es la gráfica que se muestra en la figura 7, cada nodo de la gráfica es una página del sitio web, el número que acompaña al nodo indica el número de visitas a esta página. Los nodos pueden tener varias formas según su función dentro de la sesión del usuario:

- De forma predeterminada cada página se representa como una elipse.
- Si el nodo es siempre el primer nodo de una sesión, se cambia a un rectángulo.
- Si el nodo es siempre el último nodo de una sesión, se cambia a un diamante.
- Si el nodo actúa como comienzo y final de sesión, se cambia a un octágono.
- Cada línea representa el camino a otra página y el número asociado a ésta representa cuántas veces ha visitado esa página.

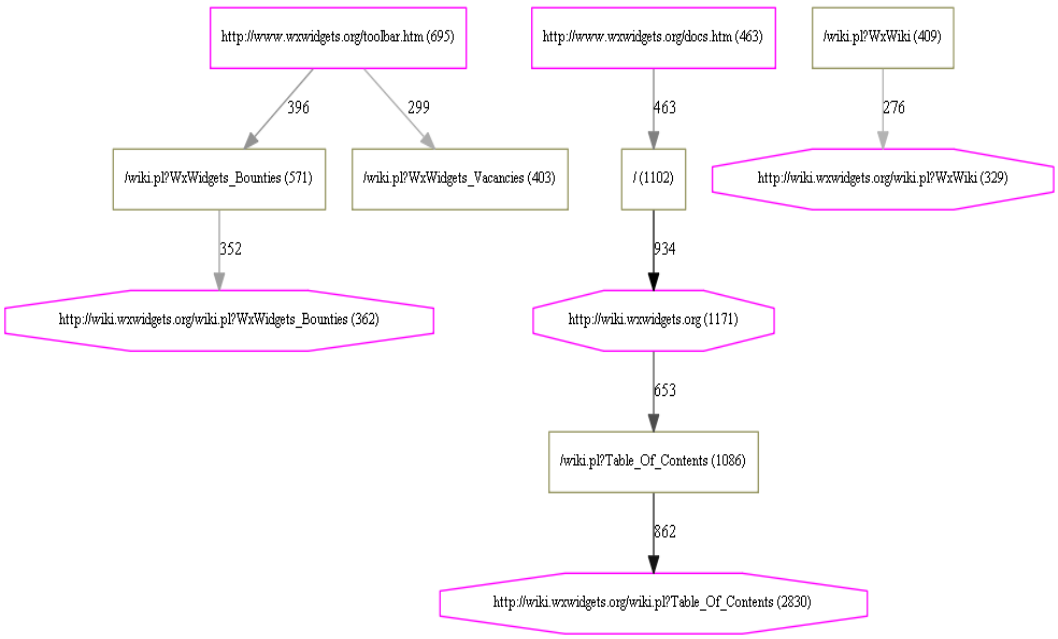

Figura 7: Gráfica generada por Pathalizer

Esta herramienta tiene sus limitantes, si el sitio es muy grande la gráfica crecerá y no podremos analizar con precisión la actividad de los usuarios, ya que no separa las sesiones de usuario y cuenta los accesos en general a las páginas, pero no se sabe qué usuario lo hizo ni en qué momento, sólo muestra un panorama muy general de los movimientos entre páginas.

Por otro lado, StatViz es otra herramienta que permite analizar el uso de un sitio web, StatViz busca dentro de las bitácoras el cambio de una página a otra dentro de una sesión de usuario, en este caso, para identificar la sesión StatViz usa la dirección IP para diferenciar a cada usuario; sin embargo, existe un problema cuando varios usuarios usan una misma IP, es imposible diferenciar qué páginas visitó cada quien, supongamos que las IP son únicas, de esta manera todas las peticiones que vienen de esta IP son parte de una sesión.

El número de clics determina la duración de la sesión, no el tiempo. Cada gráfica generada está diseñada para dar una vista simple de cómo cada visitante se mueve a través de la aplicación. Se puede configurar para graficar tantas sesiones como se requiera.

Esta gráfica muestra el recorrido del usuario a través del sitio web, analiza la navegación mediante la creación de un histograma referente al par (***recurso, clic)*** el más popular de estos pares será representado como un nodo en la gráfica. Por otro lado, cuando hay sesiones individuales permite entender si los usuarios tienen éxito o no al resolver sus tareas.

La figura 8 muestra como se mueve un usuario a través de los recursos del sitio, cada nodo representa una página web y las flechas representan el flujo que va de una página a otra. Las flechas tienen un color basado en la popularidad, también muestran la hora en que se produjo el clic y la profundidad del nodo en la gráfica. Si una página es popular su nodo y la flecha serán de color rojo, degradándose hasta azul si es menos popular. Si se ha llegado a la aplicación desde una referencia externa, el nodo se representará de color café, cuanto más oscuro sea el color de los nodos significa que ha recibido más clics.

En algunos casos también se puede ver la etiqueta BACK la cual se muestra con una flecha punteada, esto representa que un usuario ha regresado a la página anterior y vuelto a la actual. El nodo de color verde representa la página de inicio y el nodo de color rojo representa la página de salida de la aplicación.

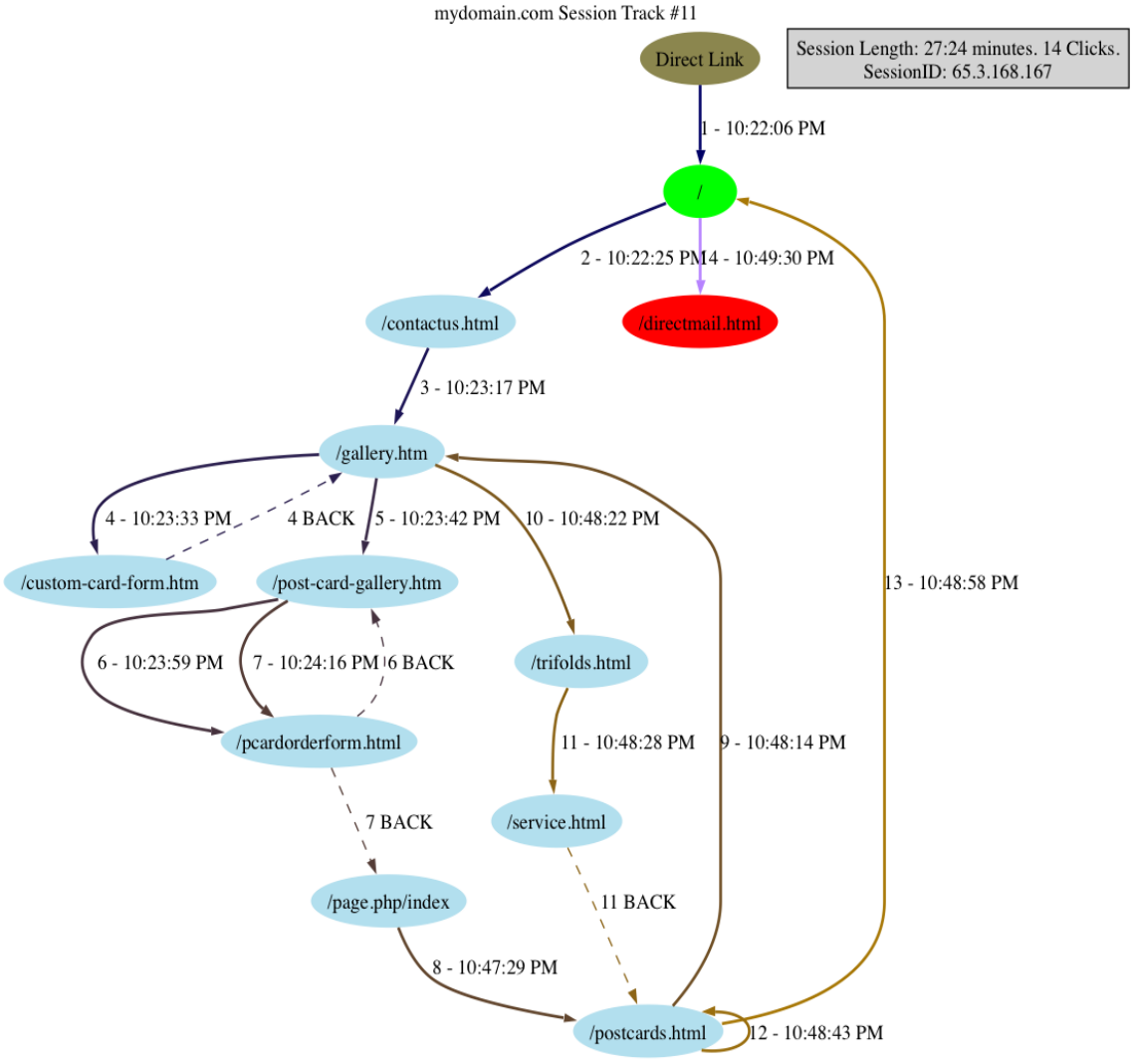

Figura 8: Gráfica generada por StatViz

Este análisis es más completo, pero aun así no es posible determinar la actividad del usuario de manera rigurosa, no se conoce cómo el usuario interactúa con los elementos restantes de la interfaz, como los botones, ligas, clics sobre cuadros de texto o diálogo, entre otros y no se puede determinar si la interfaz es adecuada o fácil de usar para ellos.

3.2.2 Modelos cognitivos computacionales de la navegación web.

(Pirolli, 2007), explica que los modelos cognitivos están enfocados en entender y predecir la manera en que los usuarios navegan sobre la web, para lograr esto se necesita considerar cómo las personas perciben estas estructuras de información con respecto a sus objetivos, ya que su percepción determina su conducta al tratar de resolver una tarea.

El modelo Scent-based Navigation and Information Foraging in the ACT architecture (SNIF-ACT)[G1] , desarrollado por (Pirolli y Fu ,2003), intenta simular cómo los usuarios buscan información en la web, para ello trata de cuantificar la relevancia de los vínculos que se le van presentando al usuario hasta que llega a su meta, el modelo asume que los usuarios evalúan los vínculos de forma secuencial y toman la decisión de dar clic sobre el vínculo o regresar a la página anterior en base a la relevancia de la información, la cual se calcula con ayuda de un modelo bayesiano. Este mecanismo es dinámico y proporciona información de cómo y cuándo los usuarios deciden hacer clic en un vínculo o retirarse de esa página basándose en experiencias previas.

Otro modelo importante es el de COLIDES, Comprehension-based Linked model of Deliberate Search (Kitajima, 2007). Kitajima menciona que las personas en realidad consumen la información que pueden entender. El funcionamiento de COLIDES se basa en el modelo construcción-integración de Kintsch (1998) para la comprensión de textos, planificación de acciones y resolución de problemas.

El proceso de construcción-integración hace referencia al proceso que siguen las personas cuando las representaciones significativas de entidades externas e internas; texto, objetos de interfaz y conexiones objeto-acción, son construidas y elaboradas a partir de información recuperada de la memoria, lo que inicia un proceso de activación por propagación restringido para las necesidades a satisfacer que integra la información relevante y elimina la irrelevante.

Otro modelo es COLIDES+ (Juvina y Oostendorp, 2007) propone que la relevancia de los objetos de la interfaz no dependen del conocimiento del usuario, sino que además se debe considerar el contexto de la navegación; por ejemplo, la historia de la interacción con las páginas y analizar el recorrido del usuario por el sitio para determinar qué tan cerca se encuentra de alcanzar su objetivo.

Los modelos mencionados anteriormente se basan en el análisis de la interacción a partir de clics sobre los vínculos, por lo que es necesario extender estos modelos a partir de la captura de más información de la interacción del usuario. En el siguiente modelo (De la Cruz, G., 2011) indica la necesidad de desarrollar un modelo del comportamiento del usuario basado en el uso de los artefactos interactivos de una interfaz de usuario web. Para ello se plantean los siguientes pasos:

1. Definir un esquema de análisis del comportamiento del usuario usando minería de datos.
2. Implementar el modelado del comportamiento del usuario para una aplicación web.

Esta propuesta indica que el análisis de la navegación web a un nivel más detallado arroja más información sobre el comportamiento del usuario, en su trabajo plantea cómo podría analizarse una interfaz a partir de los artefactos interactivos.

3.2.3 Observaciones a las propuestas actuales.

Los modelos propuestos anteriormente tienen limitantes al obtener la interacción del usuario, estos muestran el recorrido entre las páginas que han visitado los usuarios. Para obtener más información del usuario se puede observar el uso de otros artefactos interactivos, por ejemplo: el paso del cursor sobre algún vínculo, el uso de cajas y botones de selección, por mencionar algunos. Estos elementos complementan la actividad de los usuarios y es posible un análisis más detallado para obtener información sobre su comportamiento.

Se propone que para la captura de la interacción se utilicen herramientas como Javascript y jQuery, que son capaces de obtener la información de los eventos generados por los artefactos interactivos.

Finalmente con estas herramientas se puede implementar el modelo propuesto en (De la Cruz, G., 2011). Una ventaja adicional de usar estas herramientas es que no se modifica ningún elemento de la interfaz a observar, para este proceso se deben añadir los scripts que nos permitan llevar a cabo la captura de información de la interacción del usuario.

3.3 Métodos de captura del comportamiento.

La manera más sencilla de captura sobre la sesión de usuario se encuentra en las bitácoras de los servidores web, éstas cuentan con información de las peticiones hechas a recursos de la aplicación, donde se usa un estándar para la descripción de éstos y generalmente se almacenan en archivos de texto.

3.3.1 Uso de bitácoras web para capturar el comportamiento del usuario.

Los servidores web generan bitácoras de acceso a los recursos, estos son archivos de texto donde se guardan los detalles de las actividades de una aplicación web. La bitácora de un servidor web tiene cierto formato estandarizado por (W3C). Una entrada en la bitácora se conforma de varias partes, la estructura básica de una bitácora de Apache es la siguiente:

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

Dónde:

- 127.0.0.1: Es la dirección IP del cliente (host remoto) que hizo la petición al servidor.
- user: Este es el identificador de usuario de la persona que solicita el recurso determinado por la autentificación HTTP.
- [10/Oct/2000:13:55:36 -0700]: La hora a la que el servidor recibió la petición y se desglosa de la siguiente manera:

  [día/mes/año:hora:minuto:segundo zona_horaria]

  Día = 2*dígitos.

  Mes = 3*letras, generalmente son las tres letras iniciales del mes.

  Año = 4*dígitos.

  Hora = 2*dígitos.

  Minuto = 2*dígitos.

  Segundo = 2*dígitos.

  Zona = (`+' | `-') 4*dígitos.

- "GET /apache_pb.gif HTTP/1.0": La petición al recurso está contenida entre dobles comillas. El primer campo GET se refiere al método usado por el navegador. En el segundo campo aparece el recurso al que hacen la petición /apache_pb.gif, y tercero HTTP/1.0: corresponde al protocolo de transferencia de hipertexto y la versión.
- 200: Es el código de estado que el servidor envía de vuelta al navegador, es decir, si la petición se realizó satisfactoriamente, también existen varios códigos para reportar peticiones y posibles errores.
- 2326: La última entrada indica el tamaño del paquete regresado por el cliente web, no incluidas las cabeceras de respuesta.

El análisis de estas bitácoras nos permite obtener información sobre nuestra aplicación como:

- Generar reportes estadísticos de la actividad por semanas o por mes.
- Listar la actividad por día y por hora para corregir errores.
- Listar los códigos de estado de HTTP de todas las peticiones.
- Saber cuál es la página más visitada y cuántas veces ha sido accedida.

Estas bitácoras tienen información restringida a la actividad de la aplicación, pero si queremos enfocarnos sobre la actividad del usuario, una bitácora de este tipo no proporciona información para rastrear la navegación que el usuario ha realizado.

3.3.2 Bitácoras personalizadas con Log4j.

Existen herramientas de software libre orientadas a capturar información sobre el uso de una aplicación web. Una de éstas es conocida como log4j, es usada para la personalización de las bitácoras de una aplicación web y es posible obtener información más detallada.

Para utilizar Log4j debemos configurar un archivo de propiedades llamado *Log4j.properties*, el formato puede ser texto plano o un XML. En este caso utilizamos el formato de texto plano que contiene los *appenders*, éstos son un conjunto de instrucciones para conectarnos a alguna base de datos: la dirección donde se aloja la base, el usuario, contraseña del manejador de base de datos y el formato en que se guardarán los datos. Para la inserción de los datos ya sea en archivo de texto, base de datos o un XML se utilizan los layouts que se encargan de dar el formato necesario a cada uno de ellos.

De esta manera, en la bitácora personalizada se pueden añadir campos que no existen en la bitácora tradicional de un servidor web. En esta propuesta, se añadió una cadena que incluye la información de los eventos que producen las acciones de los usuarios sobre la interfaz.

La estructura de la bitácora personalizada es la siguiente:

- ***Campo fecha:*** Es un campo de tipo Date en el que se recibe la hora y fecha de la petición ocurrida en el servidor.
- ***Campo clase:*** Método o función en el que se procesa la información (este atributo puede omitirse).
- ***Campo prioridad o loglevel:*** Existen tipos de información del servidor los cuales tienen prioridades INFO, DEBUG, entre otras (también puede omitirse).
- ***Campo mensaje:*** Cuarteto con la siguiente información:
    - IP del cliente.
    - Tipo de evento generado:(mouseover).

- Recurso: Ruta del recurso.
- Hora: Momento exacto en el que se produjo el evento (hr, min, seg, ms).

Un ejemplo de la bitácora personalizada se puede ver en la tabla 10:

fecha	clase	prioridad	mensaje
2011-11-23 15:06:58	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,http://ihm.ccadet.unam.mx/moodle/user/view.php?id=322; 132.248.181.55}
2011-11-23 15:07:12	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1192,15:57:20:530;mouseout,http://ihm.ccadet.unam.mx/moodle/mod/resource/view.pl
2011-11-23 15:07:17	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,http://ihm.ccadet.unam.mx/moodle/user/view.php?id=4; 132.248.181.55}
2011-11-23 15:07:28	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,http://ihm.ccadet.unam.mx/moodle/user/view.php?id=322; 132.248.181.55}
2011-11-24 10:43:51	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,http://ihm.ccadet.unam.mx/moodle/course/view.php?id=30,11:35:0:184;onclick,http://ihm.ccadet.unam.mx/moodle/course/view.php?id=30,11:35:0:79
2011-11-24 10:43:56	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,http://ihm.ccadet.unam.mx/moodle/calendar/view.php?view=upcoming; 132.248.36.15}
2011-11-24 18:46:26	log4j.MeteDatos	INFO	{127.0.0.1;onFocus,password,19:36:32:681;onblur,password,19:36:36:412;onblur,password,19:36:36:412; 132.248.181.52}
2011-11-24 18:46:38	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1202,19:36:50:884;mouseout,http://ihm.ccadet.unam.mx/moodle/mod/resource/view.pl
2011-11-24 18:47:06	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,http://ihm.ccadet.unam.mx/moodle/mod/assignment/view.php?id=1247,19:36:55:267;mouseout,http://ihm.ccadet.unam.mx/moodle/mod/assignment/vi
2011-11-24 18:47:15	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1282,19:37:24:868;mouseout,http://ihm.ccadet.unam.mx/moodle/mod/resource/view.pl
2011-11-24 18:47:17	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,view.php?id=1283; 132.248.181.52}
2011-11-24 18:47:27	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,view.php?id=1283; 132.248.181.52}
2011-11-24 18:47:29	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,view.php?id=1283; 132.248.181.52}
2011-11-24 18:47:32	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,http://ihm.ccadet.unam.mx/moodle/file.php/29/IR/TDM/build/DocVector.class,19:37:45:549;mouseout,http://ihm.ccadet.unam.mx/moodle/file.php/29/
2011-11-24 19:05:09	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,view.php?id=1283; 132.248.181.52}
2011-11-26 13:00:38	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1220,13:50:41:307;mouseout,http://ihm.ccadet.unam.mx/moodle/mod/resource/view.pl
2011-11-26 13:00:42	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,http://ihm.ccadet.unam.mx/moodle/course/view.php?id=29,13:50:52:573;onclick,http://ihm.ccadet.unam.mx/moodle/course/view.php?id=29,13:50:53:
2011-11-26 13:00:46	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,http://ihm.ccadet.unam.mx/moodle/user/index.php?id=29,13:50:56:434;onclick,http://ihm.ccadet.unam.mx/moodle/user/index.php?id=29,13:50:57:37;c
2011-11-26 13:01:04	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,http://ihm.ccadet.unam.mx/moodle/user/view.php?id=3; 201.124.43.63}

Tabla 10: Ejemplo de una bitácora personalizada

Con log4j podemos insertar los datos directamente a una base de datos, siguiendo el estándar definido en el archivo *Log4j.properties* con la siguiente información:

log4j.rootCategory=ALL, BD

log4j.appender.BD=org.apache.log4j.jdbc.JDBCAppenderlog4j.appender.BD.driver=com.mysql.jdbc.Driver

log4j.appender.BD.URL=jdbc:mysql://localhost:3306/tesislogs

log4j.appender.BD.user=$$$$$

log4j.appender.BD.password=$$$$$$

log4j.appender.BD.layout=org.apache.log4j.PatternLayout

log4j.appender.BD.layout.ConversionPattern=%d %-5p %C.%M(%L)===> %m%n

log4j.appender.BD.sql=INSERT INTO tesislogs.logueos (fecha,clase,prioridad,mensaje) VALUES ('%n%d{yyyy-MM-dd HH:mm:ss}','%c','%p','%m ')

El campo prioridad tiene el objetivo de dar información sobre los movimientos que las aplicaciones generan en el servidor. Por último, se encuentra el campo mensaje, el cual contiene una cadena muy grande que representa el conjunto de movimientos que se hacen en la aplicación por algún usuario.

Para que toda esta información pueda ser procesada por una computadora es necesario generar una estructura en la que podamos identificar cada entrada distinta, separándolos por

medio de punto y coma cada renglón y una coma para separar los campos como se muestra en la siguiente secuencia:

{mouseover,http://ihm.ccadet.unam.mx/moodle/mod/assignment/index.php?id=29,21:1:43:920;onclic,http://ihm.ccadet.unam.mx/moodle/mod/assignment/index.php?id=29,21:1:44:430;onclic,http://ihm.ccadet.unam.mx/moodle/mod/assignment/index.php?id=29,21:1:44:430;201.103.56.87 }.

En esta cuarteta se guardan los eventos que produce el usuario sobre la página, por lo que en toda su sesión se generarán varios registros con este formato, en la tabla 11 se muestra la una entrada de la bitácora que contiene la información explicada anteriormente.

fecha	clase	prioridad	mensaje
2011-11-23 15:06:58	log4j.MeteDatos	INFO	{127.0.0.1;mouseover,http://ihm.ccadet.unam.mx/moodle/user/view.php?id=322; 132.248.181.55}

Tabla 11: Representación de una entrada de la bitácora

3.4 Análisis de las bitácoras.

Como se ha venido indicando, las bitácoras pueden ser analizadas para encontrar información acerca del uso del sitio, algunos problemas de servicios sobre la aplicación, entre otros. Dentro de las bitácoras suele haber información que es recurrente y que puede mostrarnos indicios del problema existente, lo que se buscará en el análisis de estas bitácoras serán patrones de comportamiento de los usuarios.

3.4.1 Qué es un patrón.

Un patrón es una secuencia de un flujo de clics pertenecientes a una sesión que se presenta de manera recurrente.

Este ejemplo tiene la intención de mostrar la idea intuitiva de los patrones que pueden encontrarse en una secuencia de datos. Supongamos que cada renglón representa una secuencia y cada columna contiene un símbolo de la secuencia, como se ve en la tabla 12.

A	**B**	**C**	**B**	**A**	**D**
B	A	A	C	B	D
C	B	A	C	A	D
A	B	C	C	B	C

Tabla 12: Secuencias

Se comparan todas las secuencias una contra otra sobre el primer símbolo, de modo que en la primer columna tenemos que la secuencia 1 y 4 contienen una A, después en la segunda posición observamos que la secuencia 1 y 4 ambas contienen una B, si se compara la columna 3 se encuentra que las secuencias 1 y 4 contienen a ***C***, si continuamos observamos que en la cuarta columna la secuencia 1 y 4 son diferentes y termina la comparación. Para

encontrar los patrones es necesario comparar una secuencia con respecto a las demás y así agruparlas. Es decir, los que tienen más elementos parecidos pueden añadirse en un mismo grupo, en este caso, la secuencia 4 se encuentra más cercano a parecerse a la secuencia 1 que las restantes y por otro lado, las secuencias 2 y 3 son más parecidas entre sí, por lo que existen dos grupos distintos.

A	B	C	B	A	D
B	A	A	C	B	D
C	B	A	C	A	D
A	B	C	C	B	C

Tabla 13: Representación de patrones

El primer conjunto tendrá a las secuencias ABCBAD y ABCCBC, el segundo conjunto contiene a BAACBD y CBACAD. Esto ha sido un ejemplo representado en la tabla 13, pero para realizar esta agrupación es necesario hacer uso de herramientas más sofisticadas.

3.4.2 Preprocesamiento de datos para la búsqueda de patrones.

Para encontrar los patrones es necesario usar la información con los datos que se han recolectado en la tabla logueos, esta tabla como se mencionó anteriormente, es una bitácora personalizada creada con log4j, donde el campo mensaje es una cadena que contiene la navegación del usuario. Esta cadena representa todos los recursos generados y claramente la hora del evento en que fue activado para evitar conflictos de tiempo y ordenación.

Es necesario desglosar el campo mensaje e introducir estos datos en una nueva tabla con su campo correspondiente, el campo IP representa la dirección del usuario, el recurso, el evento que se generó al visitar ese recurso, el día y la hora.

idlog_des	ip	evento	recurso	dia	hora
1	132.248.181.54	mouseover	index.php?id=33	2012-03-20	18:53:26:573
2	189.180.35.126	mouseout	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1459	2012-03-01	22:24:22:295
3	132.248.181.54	onclick	index.php?id=33	2012-03-20	18:53:27:826
4	189.180.35.126	onmouseleave	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1459	2012-03-01	22:24:22:295
5	132.248.181.54	onclick	index.php?id=33	2012-03-20	18:53:27:826
6	189.180.35.126	mouseover	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1548	2012-03-01	22:24:22:295
7	132.248.181.54	mouseover	../course/view.php?id=33	2012-03-20	18:54:38:751
8	189.180.35.126	mouseout	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1548	2012-03-01	22:24:22:320
9	132.248.181.54	onclick	../course/view.php?id=33	2012-03-20	18:54:39:322
10	189.180.35.126	onmouseleave	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1548	2012-03-01	22:24:22:320
11	132.248.181.54	mouseout	../course/view.php?id=33	2012-03-20	18:54:39:879
12	189.180.35.126	mouseover	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1576	2012-03-01	22:24:22:813
13	132.248.181.54	onmouseleave	../course/view.php?id=33	2012-03-20	18:54:39:879
14	189.180.35.126	mouseout	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1576	2012-03-01	22:24:23:17
15	132.248.181.54	mouseover	http://ihm.ccadet.unam.mx/moodle/user/index.php?id=33	2012-03-20	sin hora
16	189.180.35.126	onmouseleave	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1576	2012-03-01	22:24:23:17
17	132.248.181.54	mouseover	http://ihm.ccadet.unam.mx/moodle/user/index.php?id=33	2012-03-20	sin hora
18	189.180.35.126	mouseover	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1463	2012-03-01	22:24:23:320
19	132.248.181.54	mouseover	http://ihm.ccadet.unam.mx/moodle/user/view.php?id=3	2012-03-20	sin hora
20	189.180.35.126	mouseout	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1463	2012-03-01	22:24:23:411

Tabla 14: Desglose de la tabla logueo

Una vez que tenemos la información del usuario desglosada como se muestra en la tabla 14, se puede utilizar un esquema de visualización de la actividad del usuario parecido al de Statviz, con un gráfico que indicará el tipo de evento que se produjo sobre el objeto interactivo: dar clic, cambiar el foco, entre otros. La nomenclatura se muestra en la figura 9.

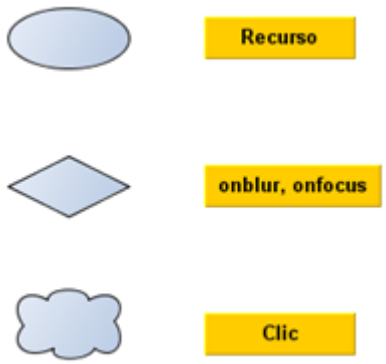

Figura 9: Representación de los cambios de estado

La nube representará el evento clic cuando el usuario pulse en cualquier vínculo y se anexará la navegación del usuario, el rombo representa todos los eventos que no son clics y por último el ovalo representa al recurso.

Para ilustrar el uso de este esquema de visualización, se elige una sesión de un usuario al azar de la base de datos, es decir, se filtra en la tabla por una IP al azar, en este caso se escogió la IP 201.137.12.249 y con fecha 2012-03-20, hay que recordar que anteriormente se definió una sesión como la actividad del usuario durante un lapso de 24 horas, como lo muestra a continuación en la tabla 15, ésta contiene todos los registros del usuario con IP 201.137.12.249 de ese día.

idlog_des	ip	evento	recurso	dia	hora
25595	201.137.12.249	onblur	password	2012-03-20	21:26:13:644
25598	201.137.12.249	onblur	password	2012-03-20	21:26:13:644
25601	201.137.12.249	mouseover	http://lhm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1477	2012-03-20	21:26:22:572
185	201.137.12.249	mouseover	http://lhm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1477	2012-03-20	21:26:22:572
85955	201.137.12.249	mouseover	http://lhm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1477	2012-03-20	21:26:22:572
25604	201.137.12.249	mouseout	http://lhm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1477	2012-03-20	21:26:22:594
25607	201.137.12.249	onmouseleave	http://lhm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1477	2012-03-20	21:26:22:594
187	201.137.12.249	mouseout	http://lhm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1477	2012-03-20	21:26:22:594
189	201.137.12.249	onmouseleave	http://lhm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1477	2012-03-20	21:26:22:594
85957	201.137.12.249	mouseout	http://lhm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1477	2012-03-20	21:26:22:594
85959	201.137.12.249	onmouseleave	http://lhm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1477	2012-03-20	21:26:22:594
25610	201.137.12.249	mouseover	http://lhm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1476	2012-03-20	21:26:22:598
191	201.137.12.249	mouseover	http://lhm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1476	2012-03-20	21:26:22:598
25613	201.137.12.249	mouseout	http://lhm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1476	2012-03-20	21:26:22:620
25615	201.137.12.249	onmouseleave	http://lhm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1476	2012-03-20	21:26:22:620
193	201.137.12.249	mouseout	http://lhm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1476	2012-03-20	21:26:22:620
195	201.137.12.249	onmouseleave	http://lhm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1476	2012-03-20	21:26:22:620
25618	201.137.12.249	mouseover	http://lhm.ccadet.unam.mx/moodle/mod/assignment/view.php?id=1620	2012-03-20	21:26:22:624
197	201.137.12.249	mouseover	http://lhm.ccadet.unam.mx/moodle/mod/assignment/view.php?id=1620	2012-03-20	21:26:22:624
25621	201.137.12.249	onclick	http://lhm.ccadet.unam.mx/moodle/mod/assignment/view.php?id=1620	2012-03-20	21:26:23:236

Tabla 15: Fragmento de una sesión

La representación gráfica del flujo de navegación de la sesión completa sobre el usuario 201.137.12.249 se muestra en la figura 10:

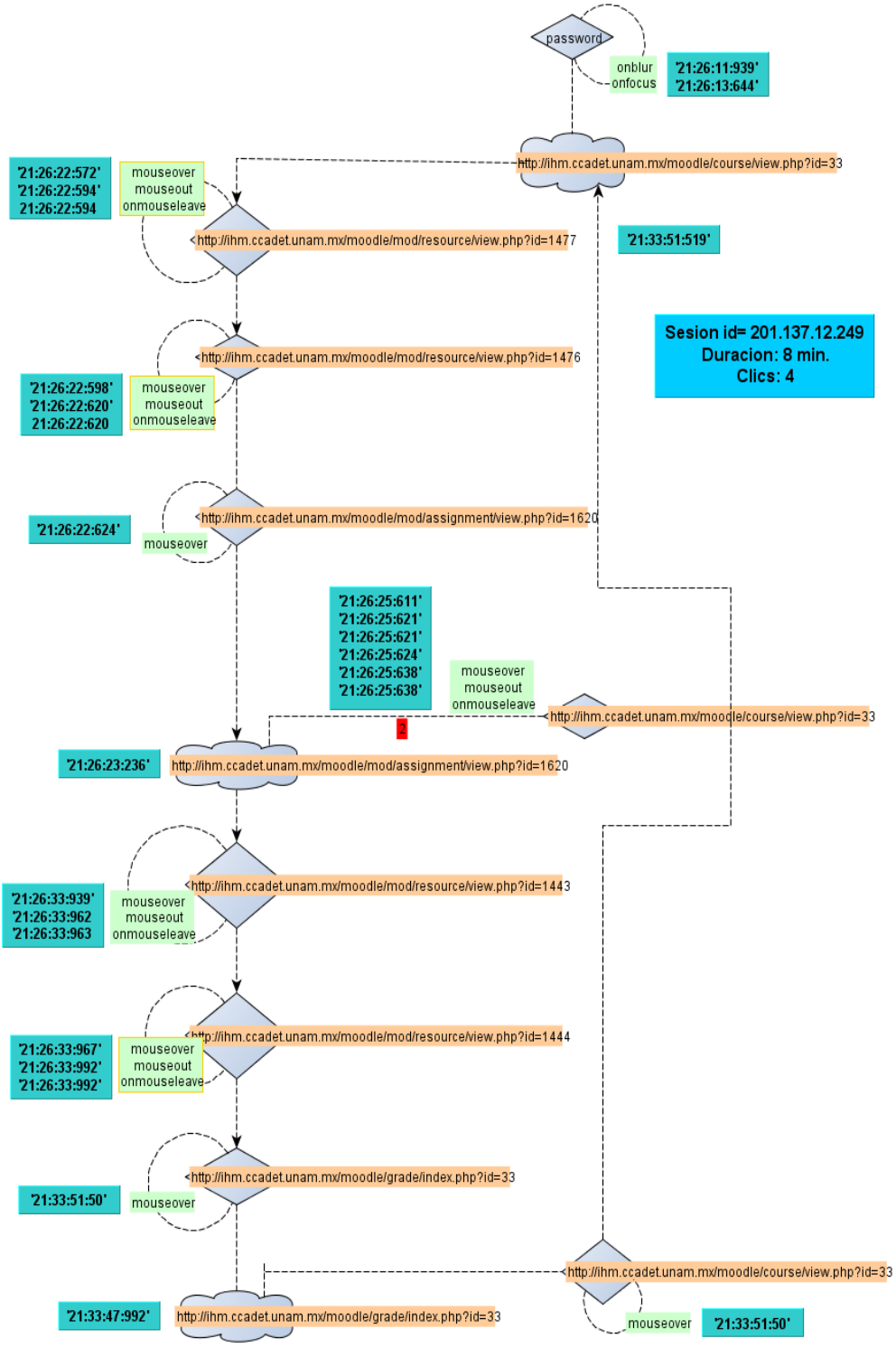

Figura 10: Representación del flujo de navegación.

Por otra parte, se utiliza el contenido del campo llamado recurso que aloja a todos los registros sin repetir, todos deben ser únicos para poder ser diferenciados, y en la columna código se encuentra el símbolo asociado al recurso correspondiente como se muestra en la tabla 16.

id_rec	recurso	identificador	codigo
1	http://ihm.ccadet.unam.mx/moodle/file.php/33/moddata/assignment/157/261/exame...	file.php/33/moddata/assignment/157/261/examen_4.pdf	A1
2	http://ihm.ccadet.unam.mx/moodle/course/category.php?id=4	course/category.php?id=4	B1
3	http://ihm.ccadet.unam.mx/moodle/mod/forum/user.php?course=1	mod/forum/user.php?course=1	C1
4	http://ihm.ccadet.unam.mx/moodle/course/category.php?id=3	course/category.php?id=3	D1
5	http://ihm.ccadet.unam.mx/moodle/mod/assignment/view.php?id=1466	mod/assignment/view.php?id=1466	E1
6	http://ihm.ccadet.unam.mx/moodle/mod/forum/post.php?forum=233	mod/forum/post.php?forum=233	F1

Tabla 16: Registros de la tabla recursos.

a) Reconstrucción de sesiones.

Una sesión del usuario comúnmente comienza en la página de inicio, donde se realiza un proceso de identificación y autorización a través de un nombre de usuario y contraseña, pero todo esto depende de las facilidades de los sitios web, los navegadores y las propias costumbres del usuario, ya que algunos sistemas permiten que nunca se cierre la sesión, de manera que al apagar o suspender la computadora es posible que más adelante se siga trabajando sin necesidad de ir la página de inicio. Estos son hábitos muy comunes entre los usuarios y complica la identificación de las sesiones, así que, por esta razón se propone que una sesión tenga tiempo límite alrededor de un lapso de 24 horas, sería suficientemente significativo para el análisis del comportamiento del usuario.

Con la ayuda de las bitácoras personalizadas podemos filtrar los datos y obtener sólo la actividad del usuario 201.137.12.249 del día 20-03-2012 como se muestra en la tabla 17.

idlog_des	ip	evento	recurso	dia	hora
179	201.137.12.249	onFocus	password	2012-03-20	21:26:11:939
181	201.137.12.249	onblur	password	2012-03-20	21:26:13:644
183	201.137.12.249	onblur	password	2012-03-20	21:26:13:644
185	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1477	2012-03-20	21:26:22:572
187	201.137.12.249	mouseout	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1477	2012-03-20	21:26:22:594
189	201.137.12.249	onmouseleave	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1477	2012-03-20	21:26:22:594
191	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1476	2012-03-20	21:26:22:598
193	201.137.12.249	mouseout	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1476	2012-03-20	21:26:22:620
195	201.137.12.249	onmouseleave	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1476	2012-03-20	21:26:22:620
197	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/mod/assignment/view.php?id=1620	2012-03-20	21:26:22:624
199	201.137.12.249	onclick	http://ihm.ccadet.unam.mx/moodle/mod/assignment/view.php?id=1620	2012-03-20	21:26:23:236
201	201.137.12.249	onclick	http://ihm.ccadet.unam.mx/moodle/mod/assignment/view.php?id=1620	2012-03-20	21:26:23:236
203	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/course/view.php?id=33	2012-03-20	21:26:25:611
205	201.137.12.249	mouseout	http://ihm.ccadet.unam.mx/moodle/course/view.php?id=33	2012-03-20	21:26:25:621
207	201.137.12.249	onmouseleave	http://ihm.ccadet.unam.mx/moodle/course/view.php?id=33	2012-03-20	21:26:25:621
209	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/course/view.php?id=33	2012-03-20	21:26:25:624
211	201.137.12.249	mouseout	http://ihm.ccadet.unam.mx/moodle/course/view.php?id=33	2012-03-20	21:26:25:638
213	201.137.12.249	onmouseleave	http://ihm.ccadet.unam.mx/moodle/course/view.php?id=33	2012-03-20	21:26:25:638
215	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/user/view.php?id=306	2012-03-20	sin hora
217	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/mod/forum/view.php?id=1443	2012-03-20	21:26:33:939

Tabla 17: Bitácora desglosada

A través de este flujo de clics se observa que por cada recurso se disparan varios eventos. Los eventos representan gran parte del comportamiento del usuario, un evento no tiene jerarquía alguna, sólo nos proporciona información del elemento en el que ha puesto su atención y al encontrar un clic se sabe que ha visitado un recurso específico. Existen varios tipos de eventos como: onmouse, onclick, onfocus, entre otros. Los objetos interactivos tienen asignados ciertos tipos de eventos, en la página de documentación de jquery se describe qué tipos de eventos tiene cada objeto.

Al tener la información de las sesiones es importante hacer una limpieza de los datos en este tipo de bitácoras, se deberá tener cuidado con los eventos que genera el navegador web, aún no existe ninguna herramienta que permita capturar estos eventos.

En la tabla 17 se puede ver que algunos recursos no tienen hora, estos datos son producidos por eventos del navegador; sin embargo, éstos no pueden ser capturados por herramientas como javascript o jquery, es por esta razón que se introduce la cadena "*sin hora*", el recurso se almacena por que el valor queda en la variable y no ha sido limpiada o vuelta a inicializar, es un error que no se tenía previsto hasta que se analizaron los datos al tratar de obtener un valor de tipo fecha, la variable se vuelve nula y esto quiere decir que no es posible obtener ese recurso, ya que los datos no existen, si el valor es nulo o indefinido se le agrega una excepción y a la variable se le asigna la cadena "*sin hora*". Como se explicó en el capítulo 2 sobre el pre-procesamiento de datos estos valores pueden quitarse por que no aportan información sobre la interacción del usuario con el sistema.Al no tener una hora que los represente es imposible saber en qué momento se disparó el evento y seguir la reconstrucción de la navegación del usuario, es decir, el flujo de navegación se interrumpió así que se decidió omitirlos y continuar con el siguiente recurso que si cumpliera con tener hora. En la sesión de la tabla 18 pareciera que se repiten los recursos, pero se debe notar que el evento no es el mismo.

idlog_des	ip	evento	recurso	dia	hora
179	201.137.12.249	onFocus	password	2012-03-20	21:26:11:939
181	201.137.12.249	onblur	password	2012-03-20	21:26:13:644
183	201.137.12.249	onblur	password	2012-03-20	21:26:13:644
185	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1477	2012-03-20	21:26:22:572
187	201.137.12.249	mouseout	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1477	2012-03-20	21:26:22:594
189	201.137.12.249	onmouseleave	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1477	2012-03-20	21:26:22:594
191	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1476	2012-03-20	21:26:22:598
193	201.137.12.249	mouseout	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1476	2012-03-20	21:26:22:620
195	201.137.12.249	onmouseleave	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1476	2012-03-20	21:26:22:620
197	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/mod/assignment/view.php?id=1620	2012-03-20	21:26:22:624
199	201.137.12.249	onclick	http://ihm.ccadet.unam.mx/moodle/mod/assignment/view.php?id=1620	2012-03-20	21:26:23:236
201	201.137.12.249	onclick	http://ihm.ccadet.unam.mx/moodle/mod/assignment/view.php?id=1620	2012-03-20	21:26:23:236
203	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/course/view.php?id=33	2012-03-20	21:26:25:611
205	201.137.12.249	mouseout	http://ihm.ccadet.unam.mx/moodle/course/view.php?id=33	2012-03-20	21:26:25:621
207	201.137.12.249	onmouseleave	http://ihm.ccadet.unam.mx/moodle/course/view.php?id=33	2012-03-20	21:26:25:621
209	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/course/view.php?id=33	2012-03-20	21:26:25:624
211	201.137.12.249	mouseout	http://ihm.ccadet.unam.mx/moodle/course/view.php?id=33	2012-03-20	21:26:25:638
213	201.137.12.249	onmouseleave	http://ihm.ccadet.unam.mx/moodle/course/view.php?id=33	2012-03-20	21:26:25:638
215	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/user/view.php?id=306	2012-03-20	sin hora
217	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/mod/forum/view.php?id=1443	2012-03-20	21:26:33:939
223	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/mod/forum/view.php?id=1444	2012-03-20	21:26:33:967
225	201.137.12.249	mouseout	http://ihm.ccadet.unam.mx/moodle/mod/forum/view.php?id=1444	2012-03-20	21:26:33:992
227	201.137.12.249	onmouseleave	http://ihm.ccadet.unam.mx/moodle/mod/forum/view.php?id=1444	2012-03-20	21:26:33:992
229	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/grade/index.php?id=33	2012-03-20	21:26:34:367
231	201.137.12.249	onclick	http://ihm.ccadet.unam.mx/moodle/grade/index.php?id=33	2012-03-20	21:26:34:788
233	201.137.12.249	onclick	http://ihm.ccadet.unam.mx/moodle/grade/index.php?id=33	2012-03-20	21:26:34:788
235	201.137.12.249	mouseover	?id=33	2012-03-20	sin hora
237	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1477	2012-03-20	21:33:46:709
239	201.137.12.249	mouseout	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1477	2012-03-20	21:33:46:719
241	201.137.12.249	onmouseleave	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1477	2012-03-20	21:33:46:719
243	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1476	2012-03-20	21:33:46:723
245	201.137.12.249	mouseout	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1476	2012-03-20	21:33:46:770
247	201.137.12.249	onmouseleave	http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1476	2012-03-20	21:33:46:770
249	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/mod/assignment/view.php?id=1620	2012-03-20	21:33:47:547
251	201.137.12.249	onclick	http://ihm.ccadet.unam.mx/moodle/mod/assignment/view.php?id=1620	2012-03-20	21:33:47:992
253	201.137.12.249	onclick	http://ihm.ccadet.unam.mx/moodle/mod/assignment/view.php?id=1620	2012-03-20	21:33:47:992
255	201.137.12.249	mouseover	http://ihm.ccadet.unam.mx/moodle/course/view.php?id=33	2012-03-20	21:33:51:50
257	201.137.12.249	onclick	http://ihm.ccadet.unam.mx/moodle/course/view.php?id=33	2012-03-20	21:33:51:519
259	201.137.12.249	mouseout	http://ihm.ccadet.unam.mx/moodle/course/view.php?id=33	2012-03-20	21:33:53:67
267	201.137.12.249	onmouseleave	http://ihm.ccadet.unam.mx/moodle/course/view.php?id=33	2012-03-20	21:33:53:68

Tabla 18: Sesión del usuario 201.137.12.249

La interfaz de la aplicación Moodle tiene varias secciones sobre la página principal, es posible analizarla por áreas como se muestra en la figura 11, de esta manera es más sencillo observar con que áreas interactúa más el usuario y como está organizada la interfaz. Los elementos están organizados por cierta relevancia e interés de los alumnos, así que la parte más importante se sitúa en el centro de la página.

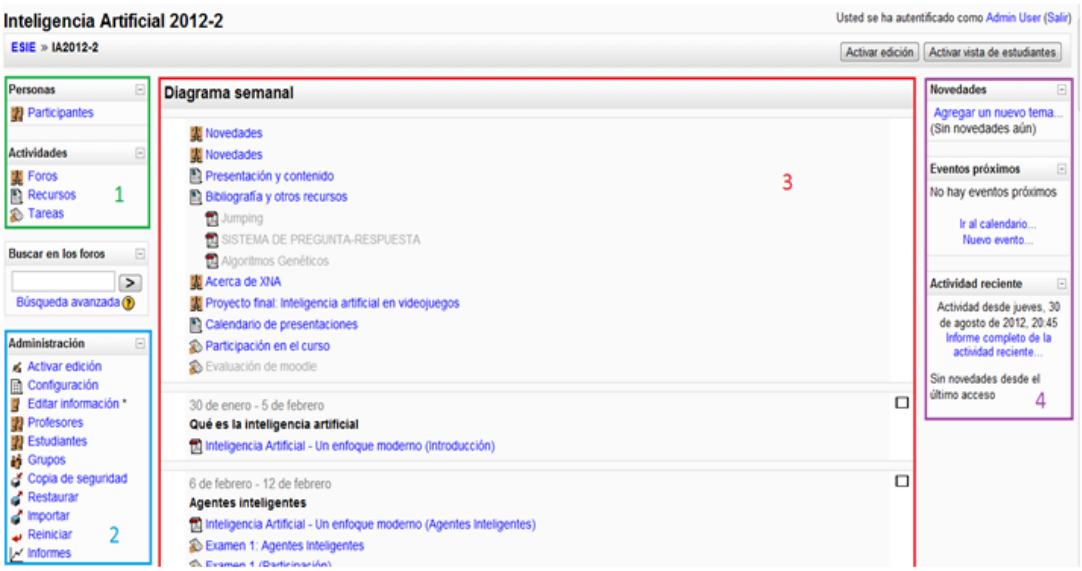

Figura 11: Áreas de la interfaz

Como se había mencionado anteriormente, en este trabajo analizaremos el comportamiento de los usuarios de la plataforma Moodle sobre el curso de inteligencia artificial. Se observará cómo los alumnos interactúan con los contenidos de éste curso.

En la página inicial puede verse que hay varias áreas, en especial en el centro, se despliegan los recursos asociados al curso mediante vínculos, esta es la parte más importante para la interacción, ya que a partir de esta sección los usuarios acceden a los diferentes recursos que se usan a lo largo del curso. Para mostrar cómo la bitácora personalizada nos ayuda a analizar la manera en que el alumno recorre el curso, describiremos algunas de las entradas de la bitácora. En nuestro ejemplo, observando la tabla 19, en el cuarto renglón se encuentra el recurso:

ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1447

Este recurso corresponde a un archivo PDF ubicado en la parte central de la página inicial, (ver figura 12). Por el tipo de eventos que aparecen en la base de datos (*mouseover*, *mouseover* y *onmouseleave*) se puede identificar que el usuario pasa el cursor sobre éste vínculo sin dar clic sobre ella.

Figura 12: Recurso PDF.

Siguiendo este mismo proceso observamos que el usuario mueve el cursor sobre el área central hasta que llega al siguiente recurso como se muestra en la figura 13:

http://ihm.ccadet.unam.mx/moodle/mod/resource/view.php?id=1548

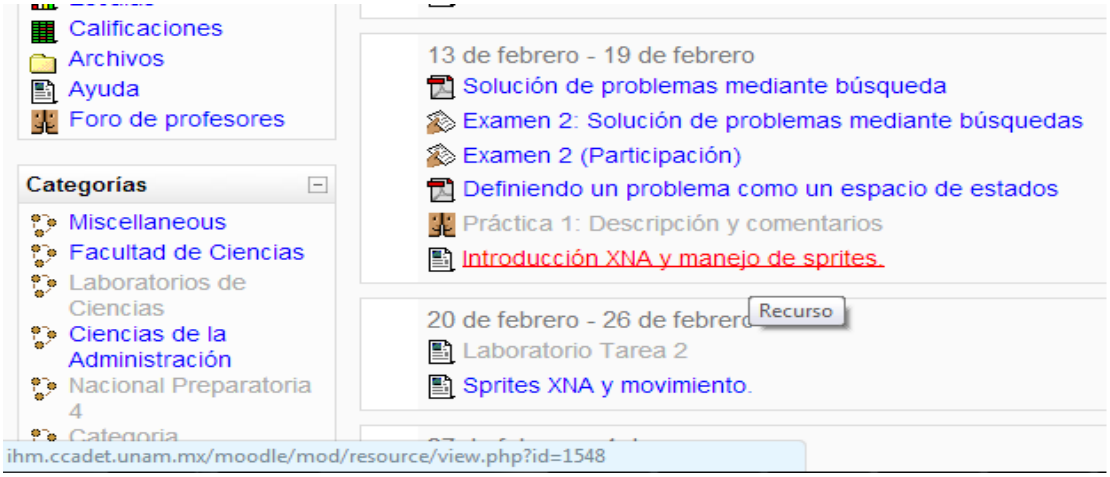

Figura 13: Ruta de un recurso

El recurso anterior corresponde a una tarea del curso y enseguida da clic, en la siguiente página nuevamente mueve el cursor sobre algunos vínculos y luego de 7 minutos regresa a la página principal.

Este ejemplo, es la muestra del análisis de una sesión pequeña, pero si tenemos muchos datos se vuelve más complejo. Cada interfaz es diferente dependiendo su funcionalidad por lo que la interacción debe ser analizada de acuerdo al objetivo de la aplicación. Una ventaja de los patrones es que pueden ayudarnos a visualizar los problemas generales que los usuarios pueden tener con la interfaz. Por otra parte, si el usuario pasa sobre algún vínculo sin estar consciente de esto es imposible descartarlo, los eventos se disparan al interactuar

con ellos y esta información es almacenada en la bitácora, pero casi nunca será el mismo a menos que se pongan de acuerdo entre los usuarios, aunque éste no es el caso, ningún usuario tiene conocimiento acerca de la captura de su interacción. Más adelante se explicará por qué no afectan estos recursos a la solución.

b) Extracción de patrones y asignación de perfil.

Existen varias herramientas que podemos utilizar para extracción de los patrones, algunas de ellas son de licencia libre como por ejemplo: orange, rapidMiner, Knime, entre otros. En este proyecto se decidió utilizar Weka, por que cuenta con un manual y ejemplos acerca de las técnicas y métodos que tiene implementadas para aplicar minería de datos, esto se explicará a continuación.

i. La Herramienta Weka.

Para encontrar los patrones se utiliza Weka la cual es una herramienta de aprendizaje automático que usa técnicas de minería de datos. Está implementada en el lenguaje java, es software libre y fue desarrollada en Waikato, Nueva Zelanda. Contiene una serie de paquetes de código abierto y los usuarios pueden contribuir con algoritmos nuevos con sólo añadirlos, puede ser descargada de la página: http://www.cs.waikato.ac.nz/ml/weka. Esta herramienta ayuda a filtrar datos para su procesamiento y se utilizan varias tareas de minería de datos como: clasificación, agrupamiento, asociación y visualización. La figura 14 muestra los algoritmos que están implementados en Weka.

Técnicas de Minería de Datos				**Implementación en WEKA**
TÉCNICAS	**No supervisadas**	Agrupamiento	Numérico	weka.clusterers.SimpleKMeans.java
			Conceptual	weka.clusterers.Cobweb.java.
			Probabilístico	weka.clusterers.EM.java
		Asociación	A Priori	weka.associations.Apriori.java.
		Predicción	Regresión	weka.classifers.LinearRegression.java weka.classifers.LWR.java
			Árboles de Predicción	weka.classifers.m5.M5Prime.java
			Estimador de Núcleos	weka.classifiers.KernelDensity
	Supervisadas	Clasificación	Tabla de Decisión	weka.classifiers.DecisionTable.java
			Árboles de Decisión	weka.classifiers.ID3.java weka.classifers.j48.J48.java weka.classifers.DecisionStump.java
			Inducción de Reglas	weka.classifers.OneR.java weka.classifers.Prism.java weka.classifers.j48.PART.java
			Bayesiana	weka.classifiers.NaiveBayesSimple.java weka.classifiers.VFI.java
			Basado en Ejemplares	weka.classifiers.IBk.java weka.classifers.kstar.KStar.java.
			Redes Neuronales	weka.classifiers.neural.NeuralNetwork.java.

Figura 14: Técnicas de Minería de Datos

ii. Conversión de datos para analizar en Weka.

Weka fue diseñado para soportar las diferentes etapas de la minería de datos, por lo tanto su interfaz permite hacer un pre-procesamiento de los datos que van a ser analizados. Los datos se pueden leer de un archivo de texto plano, con separadores entre los valores o una relación en formato propio de la herramienta llamado arff, en la que cada atributo (normalmente numérico o nominal) está descrito por un número fijo de columnas. Los archivos arff llevan cierto formato, en la primera línea debe llevar el nombre de los atributos con los que se quiere trabajar, los datos deberán estar separados por comas y con un signo de interrogación los valores vacíos.

El siguiente paso es generar la estructura del archivo arff, sobre la tabla log_desglozado se tiene el campo día donde se guardó el día en que se visitaron los recursos, también la IP que identifica al usuario y los eventos. Es posible filtrar respecto a los campos IP, día y hora, para obtener la navegación del usuario respecto al orden en que fueron accedidos. Pero existe un problema: la ruta de los recursos es muy larga así que es necesario hacerla más corta, para ello a cada recurso de la secuencia se le asignó un código que facilita este proceso a Weka.

El código es compuesto por una letra del alfabeto (A-Z) concatenado con un número entero, si la letra del alfabeto reinicia en ***A*** entonces se incrementa el entero, existe otro caso donde si el vínculo tiene el evento clic entonces se le añade la palabra "*clk*" más el número entero correspondiente.

id_rec	recurso	identificador	codigo
1	http://ihm.ccadet.unam.mx/moodle/file.php/33/moddata/assignment/157/261/examen_4.pdf	file.php/33/moddata/assignment/157/261/examen_4.pdf	A1
2	http://ihm.ccadet.unam.mx/moodle/course/category.php?id=4	course/category.php?id=4	B1
3	http://ihm.ccadet.unam.mx/moodle/mod/forum/user.php?course=1	mod/forum/user.php?course=1	C1
4	http://ihm.ccadet.unam.mx/moodle/course/category.php?id=3	course/category.php?id=3	D1
5	http://ihm.ccadet.unam.mx/moodle/mod/assignment/view.php?id=1466	mod/assignment/view.php?id=1466	E1
6	http://ihm.ccadet.unam.mx/moodle/mod/forum/post.php?forum=233	mod/forum/post.php?forum=233	F1
7	http://ihm.ccadet.unam.mx/moodle/course/category.php?id=2	course/category.php?id=2	G1
8	http://ihm.ccadet.unam.mx/moodle/course/category.php?id=1	course/category.php?id=1	H1
9	mailto:al3chikis@gmail.com	mailto:al3chikis@gmail.com	I1
10	view.php?id=1473	view.php?id=1473	J1
11	view.php?id=1476	view.php?id=1476	K1
12	http://docs.moodle.org/es/mod/resource/view	http://docs.moodle.org/es/mod/resource/view	L1
13	http://ihm.ccadet.unam.mx/moodle-test/mod/resource/view.php?id=1242	http://ihm.ccadet.unam.mx/moodle-test/mod/resource/view.php?id=1242	M1
14	view.php?id=1477	view.php?id=1477	N1
15	view.php?id=1271	view.php?id=1271	O1

Tabla 19: Asignación de recurso-código

Lo siguiente es generar una tabla donde se guardaran estas secuencias de códigos llamada navegación, esta tabla contiene el campo sec_nav que guarda la cadena y para separar cada código se utilizó el símbolo ***"-"*** como puede verse en la tabla 20:

id_nav	ip_nav	usr_nav	sec_nav	dia_nav
1	189.146.41.229	##mod/resource/view.php?id=1202##password##user/index.php?id=29##mo...	-S14-I49-M20-I14-T3-I11-Q9-T3-I14-N11-I11-M20-Q9-Q9-S14-S14-M20-I11-T14-I11-S8-S14-M20-...	2011-12-31
2	201.141.43.54	##course/view.php?id=34##course/view.php?id=34##course/view.php?id=3...	-Z8-clk1506-clk1506-W8-P34-W8-P34-P34-Z8-O30-W8-W8-Z8-H18-clk1506-clk874-Z8-P34-Z8-...	2012-05-24
3	201.110.155.44	##user/index.php?id=33##user/index.php?id=33##view.php?f=229##mod/re...	-Z18-Z18-clk1020-P15-H41-H41-F40-X8-M39-M39-clk847-P15-M6-Z18-Z18-X8-clk1504-clk1020-...	2012-05-24
4	201.141.30.54	##mod/assignment/view.php?id=1567##view.php?id=1558##mod/assignmen...	-X50-J26-B48-H17-U25-Q16-W8-X50-clk1224-clk769-W49-Z8-Q16-I12-X50-W49-H17-B18-U25-J...	2012-05-24
5	187.145.179.23	##https://docs.google.com/document/d/1HVZg8wkvU5YDbk420w7F37G2kn...	-G41-C10-C26-Z13-Z13-D26-V25-Z13-G41-clk1535-clk1047-D26-C26-V25-D26-V25-H30-V25-C1...	2012-05-25
6	189.230.204.104	##calendar/view.php?view=day##password##password##index.php?id=34#...	-E34-I49-I49-E16-L8-E16-I49-E16	2012-05-25
7	201.110.155.44	##username##mod/resource/view.php?id=1446##mod/resource/index.php?i...	-R1-M15-J31-H18-R36-J31-J31-D50-D50-H41-Z8-Z18-J31-I49-P15-D50-R36-P15-O26-L8-M39-I4...	2012-05-25
8	201.141.57.54	##calendar/view.php?view=upcoming##mod/resource/view.php?id=1523##u...	-E15-I12-O8-Z8-Z8-I49-I12-Z8-I12-I49	2012-05-25
9	187.145.96.244	##mod/forum/view.php?id=1451##mod/forum/discuss.php?d=641##mod/foru...	-B44-L20-B44-O44-B44-E10-O20-P18-E31-L18-H33-Q18-O20-B44-D44-P34-Z15-clk1133-O20-L8...	2012-05-26
10	189.178.161.197	##message/discussion.php?id=3##mod/resource/view.php?id=1454##mod/r...	-Y35-Z15-J31-H45-clk1152-clk380-M39-clk1152-clk665-P15-clk1048-G48-H41-M39-O26-P15-clk...	2012-05-26
11	201.141.49.118	##view.php?id=1657##course/view.php?id=34##index.php?id=34##view.php...	-D13-clk1506-clk395-D13-clk395-Z8-Z8-E16-D13-H18-E16-D13-Z8-D13-Z8-E16-D13-D13	2012-05-27
12	189.179.109.243	##../forum/view.php?f=230##mod/forum/view.php?id=1450##mod/forum/ind...	-M34-D44-M39-H41-D44-T32-J31-M34-clk871-clk1122-O33-clk1504-clk1229-H41-X8-X8-J31-J3...	2012-05-27
13	201.141.100.23	##password##sin recurso##index.php?id=33##password##index.php?id=33#...	-I49-clk450-clk1668-I49-F15-T22-T22-I49-clk566-clk1155-T22-T22-Y1-F15-K45-T22-X8-Y1-K45-...	2012-02-29
14	187.162.43.96	##mod/forum/view.php?id=1444##mod/assignment/view.php?id=1464##mod...	-P44-Y1-Y1-G2-clk1643-O44-D44-G2-F15-G14-X8-Y34-P15-G14-Y34-clk1668-G2-Y1-X8-Y3-Y1-G...	2012-02-29
15	189.143.138.52	##index.php?id=33##mod/resource/view.php?id=1455##index.php?id=33##...	-F15-B16-F15-K22-clk928-Z15-K22-Y1-R36-F15-Z15-B16-R36-Y1-Y1-X8-Z15-R36-X8-K22-Y1-clk...	2012-02-29
16	189.179.204.123	##message/index.php?tab=search##message/index.php?tab=search##mod/f...	-G44-G44-O26-O26-R7-I49-clk1472-clk1457-B44--clk665-R7-E34-G44-G44-C7-F5-R7-G44-O8-G...	2012-02-29
17	189.228.164.237	##mod/forum/view.php?id=1450##mod/resource/view.php?id=1547##mod/r...	-D44-L42-K43-D48-L42-D44-L2-L8-D48-L42-D48-L2-D48-P44-L42-P34-Y1-G2-D44-D48-L42-L2-c...	2012-02-29
18	187.207.23.50	##mod/assignment/view.php?id=1567##mod/resource/view.php?id=1523##c...	-X50-I12-Z8-M42-A43-K43-K43-clk1079-clk1593-W16-A43-Y11-G1-X50-Y11-X50-A43-K43-G1-Y1...	2012-02-28
19	187.195.97.109	##mod/forum/view.php?id=1522##mod/resource/view.php?id=1555##mod/f...	-A26-I43-A26-A26-I43-Z8-W16-W16-Z8-I43-L8-Z8-I49-H18-W16	2012-02-28

Tabla 20: Secuencia de abreviación

De la tabla anterior podemos obtener los datos y generar el archivo con las entradas: día, IP del usuario, y los recursos de la navegación separada por una coma:

Día, usuario, r1, r2, r3,.., rk

Los recursos que ocupan estos campos son de dos tipos: los que pertenecen a eventos que no son de tipo onclic como: onmouseover, onfocus, entre otros y los de tipo clic que se usarán para formar la secuencia específica que siguió cada usuario. Los eventos que se utilizan están definidos por el tipo de input que contiene la interfaz de Moodle como: vínculos, campos de texto, botones, checkbox y radiobotones, así que los eventos que se disparan son: onclick, omnouseleave, mouseover, mouseout, onfocus y onblur.

La longitud de cada renglón del archivo dependerá de la cantidad de recursos que el usuario visite por día, los atributos se encuentran en la parte superior del archivo que es muy parecido al formato csv, los campos pueden nombrarse de forma que se pueda identificar ese valor más adelante, los campos están representados por: día, usuario y los restantes con un valor numérico que comienza en 0 y termina en 448, esto se debe a que existen cadenas muy largas y otras muy cortas, la secuencia más corta contiene un elemento que es I49 que correspondería al 0.5 % de los casos y la más larga contiene más de 500 símbolos que corresponde al 2% del total de los recursos. Estos elementos no son muestras representativas por que no abarcan todo el conjunto de registros que tenemos, sólo son una pequeña porción de éstos, por lo que si se omiten permitirán quitar casos en los que su presencia puede alterar los resultados, sacamos el promedio de las longitudes de las cadenas lo que nos llevó a quedarnos con 448 campos.

Una entrada del archivo debe verse como la siguiente secuencia:

21/12/2013, 189. 179. 179. 20, H9, D6, D6, X7, O6, H9, O6, X7, X7, ?, G2, X3, Y3, X3, X2 , ?, D3

iii. Agrupamiento usando k-medias en Weka.

En este trabajo utilizaremos la técnica de agrupamiento k-medias. Como se describió en el capítulo anterior, la clasificación suele trabajar con técnicas estadísticas generando reglas. Los elementos de tipo cadena y documentos utilizan técnicas de reconocimiento de secuencias o símbolos, aunque la mayoría de los algoritmos implementados en Weka no pueden manejar directamente atributos de tipo String, por ejemplo, k-medias utiliza la distancia euclidiana que sólo recibe entradas numéricas, pero en caso de tener entradas de tipo cadena o símbolos realiza el siguiente procedimiento:

Comienza con la carga del archivo con el objeto ArrfReader, este objeto usa un BufferedReader y para leer el archivo aplica el método getData(), regresa una instancia, a la que se le aplica la función getStructure() que regresa información del conjunto de datos, este valor se modifica en caso de ser de tipo cadena o atributos relacionales, la instancia y el tipo de datos son pasados como parámetros a la función readInstance(), que lee la instancia usando tokens y regresa el objeto.

Esto puede observarse en la interfaz de Weka al cargar el archivo:

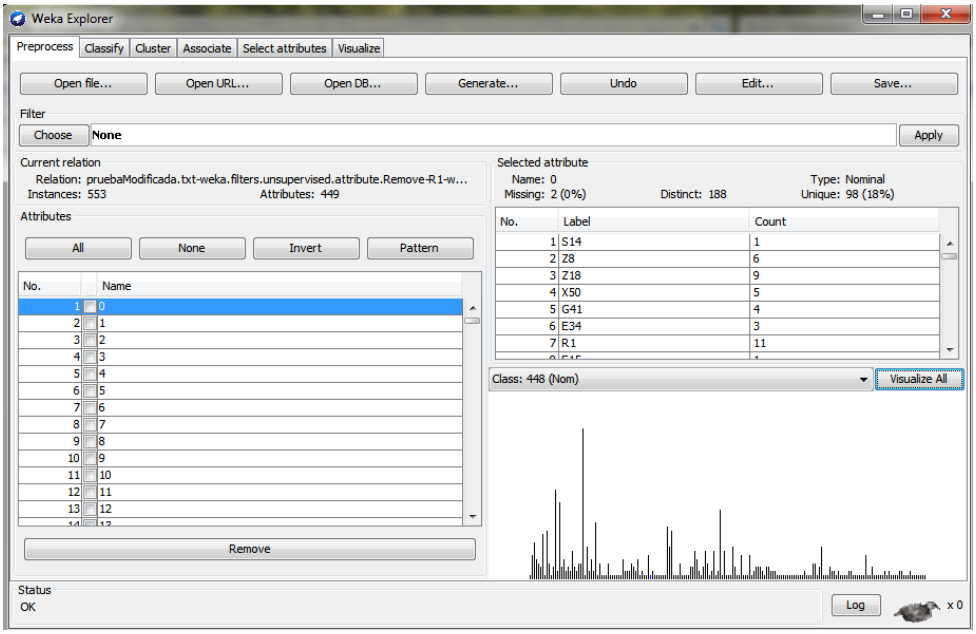

Figura 15: ArrfReader desde explorador

Nos devuelve la información contenida acerca de nuestros datos y muestra la cantidad de ***datos (553)***, el ***número de atributos (449), atributos perdidos (2)***: que son los atributos día y usuario, el ***tipo de datos (nominal)*** que quiere decir que los toma como cadenas, sin embargo, como se mencionó antes k-medias utiliza valores numéricos para encontrar la distancia euclidiana si y solo sí el tipo de datos es numérico, pero si es de tipo cadena utiliza una variante muy simple que se explicará a continuación (api Weka):

```
500     protected double difference(int index, double val1, double val2) {
501       switch (m_Data.attribute(index).type()) {
502         case Attribute.NOMINAL:
503           if (Utils.isMissingValue(val1) ||
504              Utils.isMissingValue(val2) ||
505              ((int) val1 != (int) val2)) {
506             return 1;
507           }
508           else {
509             return 0;
510           }
511
512         case Attribute.NUMERIC:
513           if (Utils.isMissingValue(val1) ||
514              Utils.isMissingValue(val2)) {
515             if (Utils.isMissingValue(val1) &&
516                Utils.isMissingValue(val2)) {
517               if (!m_DontNormalize)  //We are doing normalization
518                 return 1;
519               else
520                 return (m_Ranges[index][R_MAX] - m_Ranges[index][R_MIN]);
521             }
522             else {
523               double diff;
524               if (Utils.isMissingValue(val2)) {
525                 diff = (!m_DontNormalize) ? norm(val1, index) : val1;
526               }
```

Figura 16: función de distancia

Las funciones de distancia en Weka heredan de la clase “NormalizableDistance”, donde se define como calcular la distancia entre dos atributos, como puede observarse en la imagen la función de distancia hace la distinción según el tipo de datos que se obtiene del archivo, en nuestro caso es nominal y si los dos símbolos son iguales o alguno de ellos es vacío se le asigna 1, en caso contrario se asigna 0.

Supongamos que tenemos las siguientes secuencias de símbolos: {b-a-b, b-b, b-a, a-a-b, a, a-b-a} y queremos saber si b-a-b es más cercana a b-b, aquí no se puede aplicar una distancia numérica, en vez de esto, aplica una función de comparación que consiste en comparar carácter por carácter y para ello las cadenas deben tener la misma longitud, por ejemplo:

Tenemos b-b su longitud es 2 y para b-a-b es 3, Weka rellena los lugares vacíos con el símbolo ‘?’, así las dos cadenas tendrán longitud 3 y se puede comparar símbolo a símbolo de la siguiente manera

- Si el primer símbolo es igual al primer símbolo de la otra cadena o si alguno de los dos símbolos tiene el carácter ‘?’ se le asigna 1, en caso contrario es cero y así se recorre token por token.

b-a-b es más cercana *a* **b-b-?** =1+0+1=2

b-a-b es más cercana *a* **a-a-b** =0+1+1=2

b-a-b es más cercana *a* **b-a-?** =1+1+1=3

b-a-b es más cercana *a* **a-?-?** =0+1+1=0

b-a-b es más cercana *a* **a-b-a** =0+0+0=0

Y entonces el cluster1={a-b-a}, cluster2={b-b-?, a-a-a, a-?-?} y cluster3={b-a-?}.

En caso de que las secuencias de símbolos sea una combinación distinta de estos símbolos por ejemplo, sea b-a-b la primera secuencia y a-b-b la segunda, se puede observar que contienen los mismos símbolos y podría presentar un comportamiento similar, aunque para este algoritmo son completamente distintos.

Este procedimiento se aplica para todas las secuencias que se generaron en el archivo arff, con el propósito de encontrar las secuencias más representativas en este caso los resultados son los clústeres. Estos clústeres representan las secuencias que más repiten los usuarios, no importa si en alguna entrada es diferente porque un evento se hizo al azar o inconscientemente, la función se encarga de determinar a qué clúster pertenece.

También es posible utilizar la distancia de Levenshtein para determinar la distancia entre dos secuencias de símbolos, fue implementada a mediados del siglo XX con el propósito de medir la diferencia entre dos secuencias de símbolos, la similitud entre secuencias puede definirse como la cantidad de operaciones de edición que se requieren para transformar una secuencia en otra. Las operaciones de edición que se consideran son*: insertar un símbolo* y *borrar un símbolo*, la interpretación de similitud debe entenderse como sigue según (Cáceres. A, 2008):

Definición 1. Sean $\mathbf{x} =< \boldsymbol{x_1}, \boldsymbol{x_2}, \ldots \boldsymbol{x_n} >$ y $\mathbf{y} =< \boldsymbol{y_1}, \boldsymbol{y_2}, \ldots \boldsymbol{y_m} >$ dos secuencias finitas de símbolos en algún alfabeto finito C donde**y** es una subsecuencia de **x,** denotado por $\mathbf{y} \subset \mathbf{x}$, si existe un conjunto de índices $\{i_1, i_2, \ldots i_k\}$ $\boldsymbol{en\ x}$ donde $\mathbf{1} \leq k \leq \mathbf{n}$ tal que $\mathbf{y} =< \boldsymbol{x_{i1}}, \boldsymbol{x_{i2}}, \ldots \boldsymbol{x_{im}} >$**.**

Definición 2. Una subsecuencia $\boldsymbol{y}$ es una subsecuencia común para las secuencias $\boldsymbol{x_a}$ y $\boldsymbol{x_b}$, denotado por $\mathbf{y} \subset (\boldsymbol{x_a}, \boldsymbol{x_b})$, si $\mathbf{y} \subset \boldsymbol{x_a}$ y $\mathbf{y} \subset \boldsymbol{x_b}$.

Definición 3. Sea C' el conjunto de secuencias finitas generadas con símbolos de C; y la imagen de la función asocia un valor entero no negativo a cada par de secuencias.

La similitud entre dos secuencias $x, y \in C'$, denotada por $S(x, y)$ está dada por:

$$S(x, y) = \max\{|z|: z \subset (\mathrm{x}, \mathrm{y})\}; con\ z \in C'$$

Donde |z| indica la longitud de la secuencia z, es decir, la cantidad de símbolos que contiene.

Nótese que S(x,y)=0 cuando x no tiene símbolos comunes por lo que:

$S(x, y) = \max\{|z|: z \subset (\mathrm{x}, \mathrm{y})\}$; $con\ z \in C'$cuando contiene símbolos en común.

La distancia de Levenshtein mejora los resultados, pero no está implementado en Weka, aunque puede desarrollarse y añadirse.

El siguiente paso es encontrar los patrones y Weka utiliza k-medias para generar éstos. El algoritmo representa a cada clúster con la media ponderada de sus puntos, denominado centroide y se describe el algoritmo a continuación:

Pseudocódigo del algoritmo K-medias:

1. Elegir k-ejemplos que actúan como semillas (k representa el número de clústeres).
2. Para cada ejemplo, añadir el ejemplo a la clase más similar.
3. Calcular el nuevo centroide de cada clase, que pasan a ser las nuevas semillas. Se toma el promedio de todos los valores de los elementos pertenecientes al clúster.
4. Verificar si los centroides han cambiado sus coordenadas, si es así volver al paso 2.
5. Si no, la detección del clúster ha finalizado y todos los objetos pertenecen a un clúster.

Para construir los clústeres usamos las diferentes implementaciones que nos ofrece Weka, en la siguiente sección se describe paso a paso la forma en que se usó la herramienta para la identificación de estos grupos. Como primer paso, se debe abrir el explorador de Weka y abrir el archivo que contienen las sesiones representadas mediantes las secuencias mencionadas anteriormente. En la figura 17 se muestra la selección del archivo donde se cargan los datos, mientras que en la figura 18 se puede observar un fragmento de su contenido.

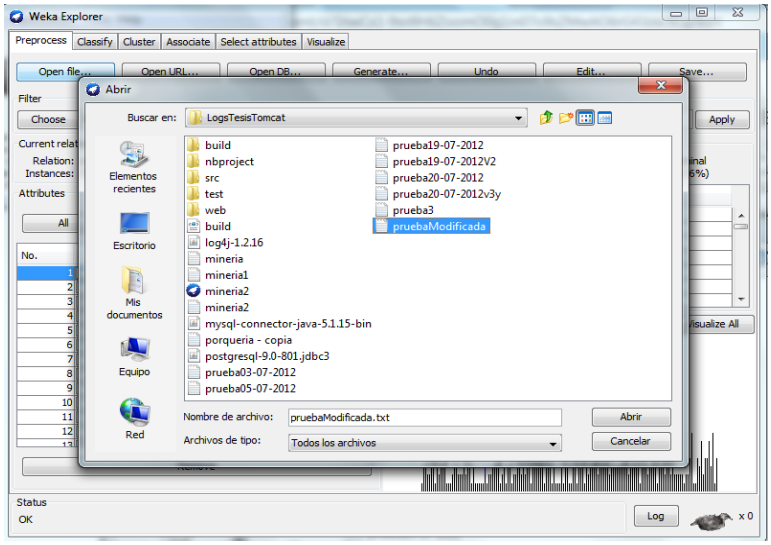

Figura 17: Seleccionar el archivo para Weka

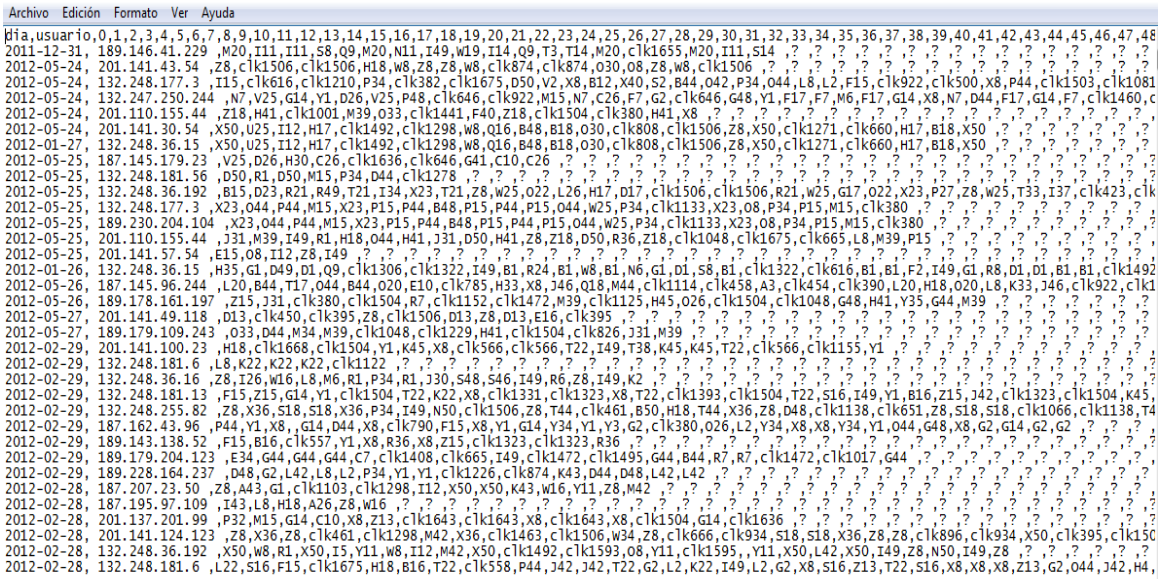

Figura 18: Archivo arff preparado para Weka

Cuando el archivo se ha cargado, en el panel inferior izquierdo se pueden seleccionar los atributos que no se utilizarán y se quitan con el botón remover. Para las pruebas se eliminaron los atributos día y usuario, ya que para generar los clústeres son necesarios los datos referentes a los recursos por qué se buscan símbolos en común, el día y el usuario no contienen información acerca de esto por eso pueden omitirse.

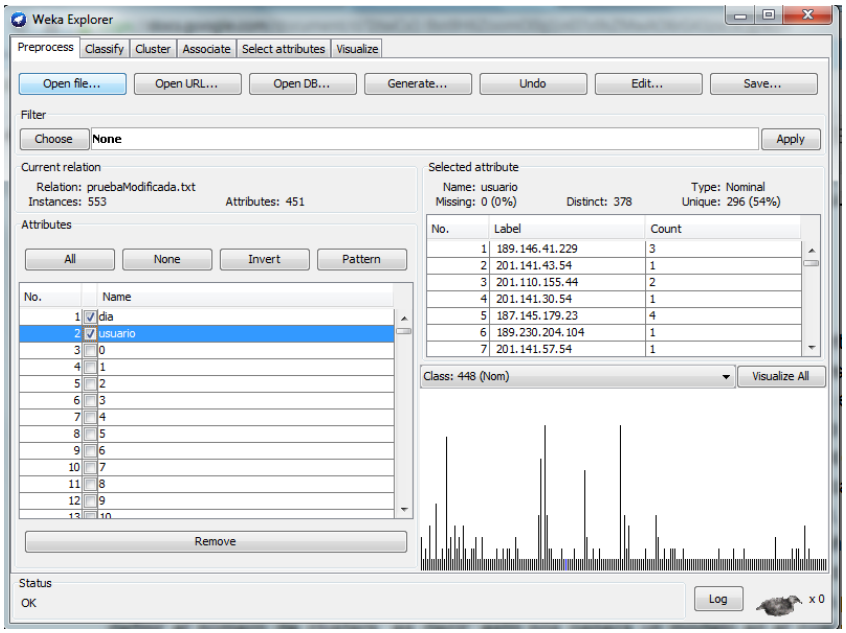

Figura 19: Exclusión de atributos

En el panel "clúster" encontramos distintos algoritmos que tiene implementados la herramienta para calcular los clústeres, para ello debemos seleccionar el algoritmo k-medias que se encuentra al dar clic en la pestaña *"clúster"*, se despliega la lista de opciones, como se muestra en la figura 20.

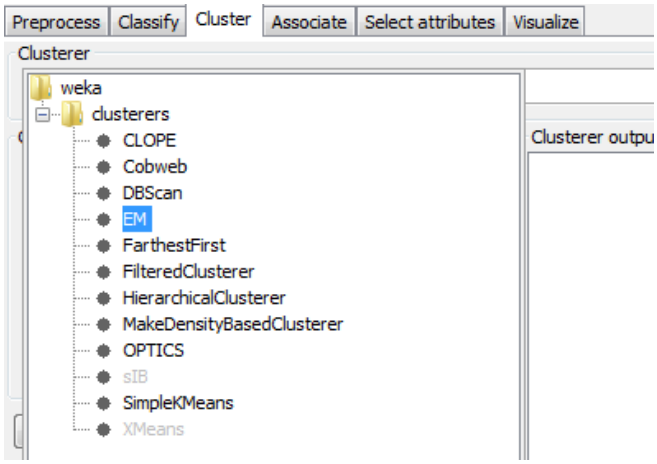

Figura 20: Selección de algoritmo K-Medias

Para el algoritmo k-medias es necesario indicar el número de clústeres *k* que se generarán, con esto selecciona *k* elementos aleatoriamente que se tomarán como referencia para construir cada clúster y se le conoce como semilla, aunque también puede fijarse como se observa en la figura 21.

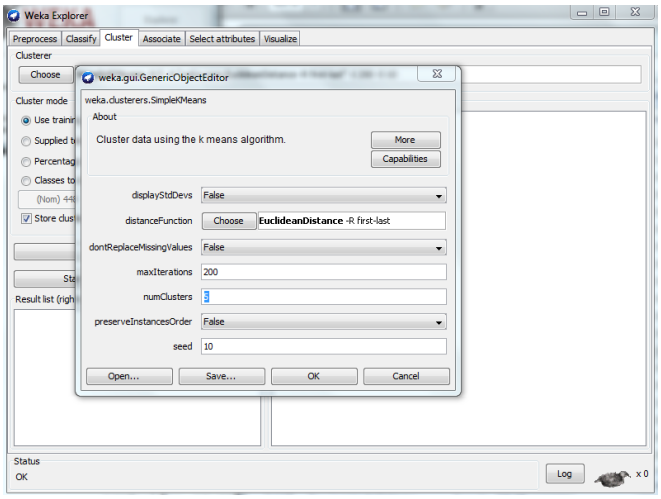

Figura 21: Asignación del número de clústeres

La representación mediante centroides tiene la ventaja de que muestra un significado gráfico y estadístico inmediato. La suma de las discrepancias entre un punto y su centroide expresado a través de la distancia apropiada se usa como una función objetivo. La función objetivo al analizar secuencias simbólicas es calcular mediante la comparación de símbolo a símbolo estas secuencias, como ya se ha explicado.

El reporte muestra el porcentaje de ejemplares de las secuencias que están asociados a cada clúster de acuerdo a lo que encontró el algoritmo. El resultado presenta 5 clústeres que pueden verse en la figura 22, estos presentan el número de instancias frecuentes que contiene y un porcentaje que indica cual clúster tiene más o menos instancias.

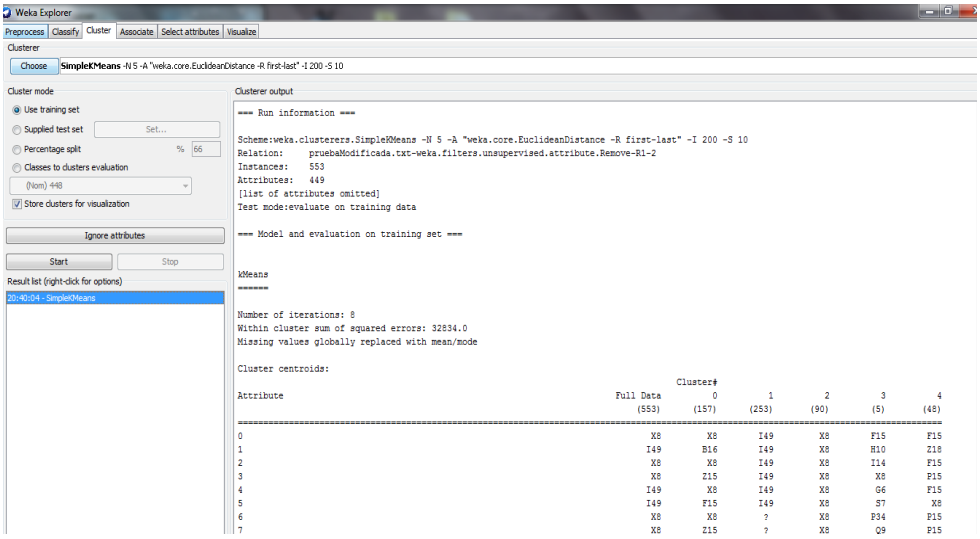

Figura 22: Generación de los clústeres

Cuando se utilizó k-medias se obtuvieron varias agrupaciones, estos resultados tienen que ser analizados y para ello se hicieron 5 pruebas variando la semilla inicial, el número de clústeres y el número de iteraciones, con el objetivo de encontrar el comportamiento general que describa a la mayoría de las secuencias.

iv. Observaciones

Se realizaron 5 pruebas cambiando el número de iteraciones, la semilla y variando el número de clústeres finales: 5, 7, 9, 12 y 18. Como resultado, la herramienta generan un modelo de evaluación y presenta como están formados los clústeres, listando una secuencia de los elementos en cada columna correspondiente al clúster, esto puede observarse en la figura 23.

	Cluster#											
Full Data	0	1	2	3	4	5	6	7	8	9	10	11
(553)	(116)	(162)	(10)	(16)	(41)	(15)	(39)	(15)	(51)	(13)	(9)	(66)
X8	L8	X8	G2	N39	X50	F15	P15	B44	X8	M15	S16	P34
I49	I49	I49	N35	I43	Z15	clk392	M15	B16	X8	F15	G2	Z15
X8	Z8	X8	Q34	R40	I49	clk1180	R1	D50	F15	I18	S16	X8
X8	I49	X8	K37	K43	I49	S17	M15	P15	H19	S2	X8	X8
I49	Z15	I49	W8	Z8	G2	X8	clk1047	R36	F15	L20	S16	X8
I49	M6	I49	Q34	I43	I49	F15	J31	S16	L2	I49	Z15	X8
X8	Z8	?	I49	Z8	P34	F20	B44	B44	X8	J31	F15	X8
X8	X8	?	I49	Z8	Z8	Z15	E36	clk392	X8	O44	P15	X8
?	P34	?	H20	Z8	B16	X8	P15	P15	T22	J31	I49	X8
?	X8	?	K37	I49	P34	V22	M15	P15	X8	O20	clk409	P15
?	P34	?	J31	clk516	Z8	A3	B44	Z18	X8	V2	X8	X8
?	B44	?	clk1331	clk516	Z8	J31	I49	L2	X8	O20	X8	Z15
?	B44	?	O34	R49	I49	F7	Z15	I49	X8	H41	X8	X8
?	B44	?	W8	V20	Z15	X8	I49	L2	X8	N7	Z18	X8
?	X8	?	I49	N50	Z8	Z18	D44	L2	X8	X8	Z8	X8
?	X8	?	Q34	Z8	I12	N23	M15	P15	T22	R24	J42	X8
?	B44	?	J31	K43	I49	B16	V23	Z15	F15	S20	G2	X8
?	Z15	?	W8	N39	Z8	I18	P15	L22	X8	Q41	S16	P15
?	P34	?	S16	M42	I49	F15	B44	B12	X8	Z8	S16	X8
?	I11	?	K37	R40	T45	Q18	M15	Z15	X8	clk1346	B16	Z15
?	?	?	H18	Y20	S8	clk1468	R36	P15	S16	F7	G2	X8
?	?	?	Q34	Z8	P34	clk790	Z15	J46	F15	R1	F15	X8
?	?	?	clk874	R49	I49	R36	M39	Z15	L22	B44	G2	Z15
?	?	?	clk873	N39	P34	clk468	B44	O44	X8	N7	G2	X8
?	?	?	Q34	clk519	I49	clk468	P15	P34	X8	F7	B16	Z15

Figura 23: Resultado obtenido para 12 clústeres

En la tabla 21 se observan los resultados de las 5 pruebas, el desglose de los porcentajes indican que ocurrieron ciertas variaciones donde las ocurrencias numéricas muestran los clústeres con el porcentaje de mayor a menor, generalmente los clústeres con mayor porcentaje significa que cubren más secuencias que uno que tenga un porcentaje menor.

# clúster	18 clústeres	12 clústeres	9 clústeres	7 clústeres	5 clústeres
0	145 (26%)	155 (28%)	197 (36%)	221 (40%)	233 (42%)
1	66 (12%)	65 (12%)	107 (19%)	107 (19%)	114 (21%)
2	58 (10%)	55 (10%)	68 (12%)	89 (16%)	89 (16%)
3	40 (7%)	45 (8%)	44 (8%)	61 (11%)	65 (12%)
4	22 (4%)	24 (4%)	28 (5%)	29 (5%)	53 (10%)
5	17 (3%)	22 (4%)	29 (5%)	29 (5%)	
6	13 (2%)	17 (3%)	18 (3%)	18 (3%)	
7	19 (3%)	33 (6%)	29 (5%)		
8	29 (5%)	34 (6%)	34 (6%)		
9	37 (7%)	49 (9%)			
10	33 (6%)	39 (7%)			
11	15 (3%)	16 (3%)			
12	4 (1%)	49 (9%)			
13	2 (0%)				
14	11 (2%)				
15	10 (2%)				
16	17 (3%)				
17	16 (3%)				

Tabla 21: Resultado de las 5 pruebas

Estos resultados dan una idea aproximada de cuál es la frecuencia de aparición de la secuencia de navegación, pero lo que necesitamos es encontrar los clústeres más representativos.

El procedimiento para la búsqueda de estos clústeres representativos inicia con compararsímbolo a símbolo cada clúster con respecto a los símbolos del otro clúster, los clústeres tienen la misma longitud, ya que representan secuencias y si una es más corta que la otra lo rellena con el signo de interrogación ***"?"*** y todos tienen tamaño de 448 símbolos en total. Para iniciar se necesita saber la procedencia de cada símbolo, ya que representa un vínculo y este puede ser recuperado desde la tabla recursos que contiene los códigos y vínculos asociados, cada vínculo está relacionado a una sección del Moodle, recordemos que estas podían ser separadas en assignements, resources, foros, entre otras.

Por ejemplo el clúster 2 y el clúster 10 que se encuentran en la figura 24 son 3 clústeres con secuencias cortas, pero si las analizamos se observa un comportamiento general:

S1= I49-I49-Y1-X8-I49-I49-X8-X8-I49-I49

S2=X8-X8-X8-M15-F15-X8-F15-J31-X8-F15-X8-X8-X8-F15-X8-X8

El código I49 indica que se ha introducido el password por lo que el usuario ha iniciado sesión y se dirige a la página inicial que está representada por el código X8, se queda ahí o hace un movimiento a otra página y regresa a X8. El comportamiento general de estos dos clústeres distintos es muy similar, expresan un comportamiento de los usuarios que entra a la aplicación y revisan uno o dos recursos. Se ha concluido que esta acción refleja varias posibilidades, el usuario es nuevo y se queda un largo tiempo leyendo cierto recurso, el usuario ya conoce dónde está ese recurso y se dirige directamente a él o la mayoría de los usuarios no saben cómo usarla y se dan por vencidos. Este análisis nos sirve para la construcción de las reglas para el agente.

De esta manera se hace lo mismo con los clústeres restantes y se encontraron 4 clústeres que representan el comportamiento en general, por lo que es suficiente usar el que contiene 5 clústeres y a continuación se describe su comportamiento general:

Clúster 0: Muestra una pequeña secuencia de navegación de 2 páginas, esto significa que los usuarios suelen acceder a través de la página de inicio del curso y en seguida visitar un recurso y concluye la sesión.

Clúster 1: Muestra otra relación entre la página inicial del curso y 2 recursos del sitio. Este comportamiento es un poco más amplio y en particular los recursos visitados corresponden a tareas, tiempo después cierra la sesión. Este tipo de comportamiento representa que algunos de los usuarios fueron directamente a las tareas y como no saben qué recursos pueden utilizar parar resolverla como consecuencia pierden cierta interacción con el curso.

Clúster 2: Se observa que otra secuencia recurrente es el de iniciar sesión en el curso, antes de entrar escriben su usuario y contraseña e inmediatamente los redirecciona a la página inicial del curso y no continúan trabajando más, esto sucede cuando están a la espera alguna noticia del sitio ya sea nuevo material o resultados.

Clúster 3: Entran y salen de páginas varias veces, luego de analizar algunos ejemplares de este clúster se observa que los usuarios parece que no encuentran el recurso que le ayudará a resolver su tarea así que busca sobre varias áreas del curso en concreto sobre las que contienen actividades recientes.

Clúster 4: Recorre la página de manera ordenada de forma descendente para encontrar los recursos sobre el área central o área 3 de la página.

	Cluster#				
Full Data	0	1	2	3	4
(554)	(233)	(114)	(89)	(65)	(53)
X8	X8	X8	I49	J42	F15
I49	X8	X8	I49	L22	B16
X8	X8	X8	I49	R1	R36
X8	M15	X8	I49	P15	S17
I49	M15	X8	I49	G2	X8
I49	X8	X8	I49	S16	F15
X8	?	X8	X8	L22	X8
X8	?	X8	X8	I49	X8
?	?	X8	I49	P15	I49
?	?	X8	I49	X8	V22
?	?	X8	I49	B16	X8
?	?	X8	I49	L2	X8
?	?	X8	I49	T22	X8
?	?	X8	?	Z15	X8
?	?	X8	?	L22	Z18
?	?	X8	?	M15	B44
?	?	X8	?	L22	Z15
?	?	P15	?	S16	X8
?	?	X8	?	S16	F15
?	?	Z15	?	L22	I49
?	?	X8	?	G2	P34
?	?	X8	?	F15	F15
?	?	X8	?	Z15	M15
?	?	X8	?	B44	B44
?	?	X8	?	B16	clk468
?	?	X8	?	F15	F15

Figura 24: Clústeres representativos

Con el fin de obtener una representación más práctica de lo que pasa con el clúster 3 se construyó una gráfica, ésta se muestra en la figura 25, donde el cambio de color indica que regresa a un recurso que previamente ya fue visitado, también tiene su código asociado y el nombre del vínculo para reconocer los nodos. Se observa que pasa varias veces por algunos nodos de forma ordenada y no visita recursos que están contenidos en tareas y recursos que posiblemente se utilizarían para resolver las tareas.

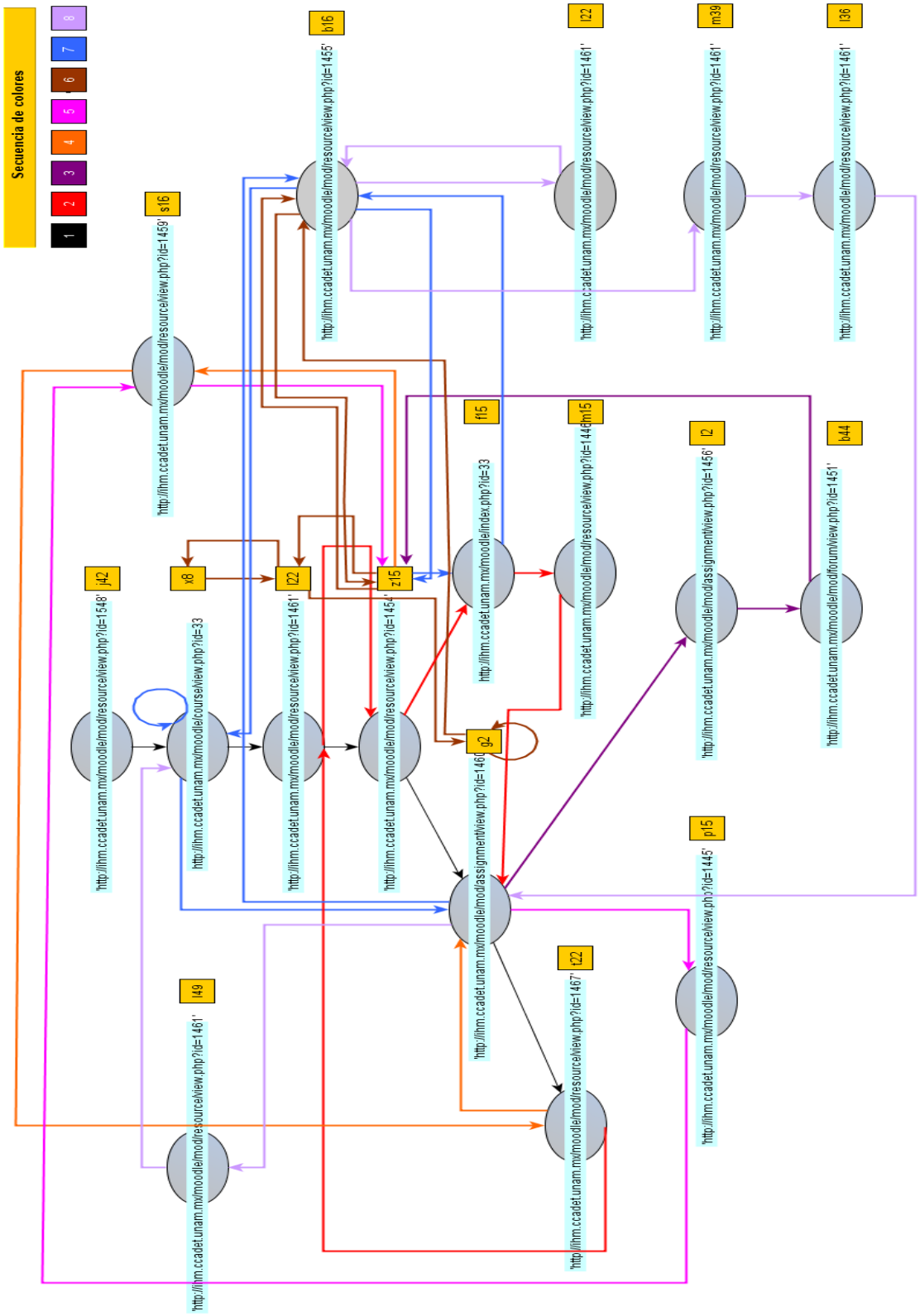

Figura 25: Representación gráfica del clúster 4

Estos 5 clústeres son representativos y se usarán para clasificar a los usuarios a partir de que se encuentren secuencias parecidas, este proceso se describirá en el siguiente capítulo.

3.5 Resumen.

En este capítulo se han presentado un conjunto de herramientas que nos ayudan a explicar la navegación del usuario y cómo visualmente podemos representar esta información para analizarla con más facilidad. La información sobre el usuario puede ser ampliada si se obtiene más datos acerca de su interacción con la interfaz.

Las bitácoras guardan de manera inalterable la interacción de los usuarios, pero podemos ir más allá de las bitácoras tradicionales de los servidores web. Es posible generar bitácoras personalizadas para guardar la información que se necesita.

También se mencionaron algunos modelos para el análisis de la navegación los cuales son la base para que nuestro modelo de captura de información sobre el usuario, pueda recopilar datos mediante el uso de la interfaz.

Por otra parte, se describió el proceso de análisis de las bitácoras, donde se pueden encontrar los patrones en la navegación, el proceso para reconstruir las sesiones de cada usuario el uso de la herramienta Weka para construir el clasificador, el cual nos genera un conjunto de clústeres representativos, su análisis ayudará a clasificar a los usuarios y posteriormente a definir las reglas con las cuales el agente debe entrenarse de acuerdo con este conocimiento que encontramos.

Capítulo 4

Aplicación.

Los agentes adaptables son programas que realizan tareas específicas para un usuario, poseen un grado de inteligencia suficiente para ejecutar parte de sus tareas de forma autónoma y logran interactuar con su entorno de forma útil. Existen 3 categorías para los agentes adaptables los cuales son: agentes biológicos, agentes de hardware y agentes de software. En particular, se usarán los agentes de software, que son aplicaciones con la capacidad de decidir cómo actuar para alcanzar sus metas.

En este capítulo definiremos qué son los agentes autónomos adaptativos (AAA), qué características tienen, cuáles son sus tareas y cómo podemos construirlos. Se utilizarán los agentes cuyo comportamiento esté guiado por una serie de reglas con las que modelamos su entorno. Se presentará un breve panorama de las tecnologías que existen para el desarrollo de este tipo de agentes.

Para mostrar el funcionamiento de los AAA sobre una interfaz de usuario usaremos la plataforma Moodle por lo tanto, el objetivo del agente será generar sugerencias personalizadas para cada usuario. Moodle por si solo es complejo, ya que contiene varios módulos y casi siempre los usuarios no saben cómo usarlo, así que estas sugerencias ayudaran al usuario a usarla de manera más eficiente.

En el capítulo 3, se explicaron las técnicas y procesos para manipular la información que se utilizará para generar las reglas que harán que el agente pueda funcionar de manera autónoma; aprenda acerca del usuario y genere sugerencias útiles.

Posteriormente, se integrará este agente a Moodle sin que la interfaz se vea afectada, se añadirá en la parte derecha inferior un recuadro que contenga la sugerencia.

4.1 Agente autónomo adaptable.

Un agente es una entidad que percibe información del entorno que lo rodea por medio de sensores y produce cambios en su medio ambiente mediante salidas o efectores (Russell, 2003). Por otra parte (Wooldridge, 1995) define al agente como un sistema de hardware o un sistema de cómputo basado en software que contiene las siguientes propiedades:

- **Autonomía:** Los agentes operan sin la intervención directa de humanos u otros, y tienen alguna clase de control sobre sus acciones y estado interno.
- **Habilidad social:** Los agentes interactúan con otros agentes vía alguna clase de lenguaje de comunicación de agentes.

- **Reactividad:** Los agentes perciben su ambiente y responden en una forma oportuna a los cambios que ocurren en él.
- **Proactividad:** Los agentes no actúan simplemente en respuesta a su ambiente, ellos exhiben un comportamiento dirigido a metas tomando la iniciativa.

Otras características importantes que nuestro agente debe tener son:

- **Razonamiento y aprendizaje:** Ambos aspectos son necesarios para que el agente sea capaz de comportarse inteligentemente.
- **Adaptabilidad:** La capacidad de realizar objetivos y tareas en distintos dominios de forma incremental y flexible.

Los agentes tienen ciertas características y comportamientos de acuerdo al ambiente en el cual se utilizan. En nuestro caso el agente actuará en una aplicación web, por lo que usaremos particularmente un agente con las características de autonomía y adaptabilidad.

Para construir un agente es necesario definir su arquitectura. (Maes, 1994) define a la arquitectura de agentes como:

"Una metodología particular para construir agentes. Especifica cómo el agente puede ser descompuesto en la construcción de un conjunto de módulos y cómo pueden interactuar. El conjunto total de módulos y sus interacciones tienen que proveer una respuesta a la pregunta de cómo el sensor y el estado actual interno del agente determinan las acciones y estado futuro interno del agente. Una arquitectura incluye técnicas y algoritmos que soportan esta metodología."

Wooldridge sugiere la siguiente clasificación de las arquitecturas:

Arquitectura deliberativa: Este tipo de arquitectura utiliza modelos de representación simbólica del conocimiento, donde las decisiones se toman utilizando mecanismos de razonamiento lógico basados en la correspondencia de patrones y la manipulación simbólica con el propósito de alcanzar los objetivos del agente.

Arquitectura Reactiva: Se caracteriza por no tener como elemento principal de razonamiento un modelo simbólico, no utiliza razonamiento simbólico complejo y generalmente se usan en robótica por medio de heurísticas, toma de decisiones, entre otros.

Arquitectura híbrida: Esta arquitectura pretende combinar las características de las dos arquitecturas anteriores. Una propuesta para ello es construir un agente compuesto de dos subsistemas: uno deliberativo, que utilice un modelo simbólico para que genere planes y el otro reactivo, centrado en reaccionar ante los eventos que tengan lugar en el entorno que no

requieran de un mecanismo de razonamiento complejo. Se basa en una estructuración por capas, las cuales son:

- **Vertical:** Sólo una capa tiene acceso a los sensores y actuadores.
- **Horizontal:** Todas las capas tienen acceso a los sensores y actuadores.

Para este tipo de arquitectura el número de capas máximo son 3 niveles las cuales son:

- **Reactivo o de más bajo nivel:** Se toman decisiones acerca de qué debe hacer en base a estímulos recibidos del entorno en tiempo real.
- **Conocimiento o nivel intermedio:** Se centra en el conocimiento que el agente posee del medio, normalmente con la ayuda de una representación simbólica.
- **Social:** En esta última capa se manejan los aspectos sociales del entorno, incluyendo información de otros agentes, deseos, intenciones, entre otros.

4.2 Construcción de un AAA para una IUA.

Para la construcción de un AAA se ha utilizado una arquitectura deliberativa la cual se basa en que los agentes que la implementan permiten cambiar la percepción del entorno de acuerdo al conocimiento que tienen de éste.

Es por esta razón, que definimos un conjunto de reglas que le indicarán al agente qué acciones debe realizar ante un conjunto de cambios esperados del ambiente, de tal manera que, su comportamiento se base en la observación de patrones para lograr formar una sugerencia.

En este caso el entorno del agente es una interfaz web y los cambios que puede percibir en él son los eventos que el usuario genera al usar la página web. Como se ha visto en el capítulo 3, los datos de la interacción se pueden procesar y analizar por medio de minería del uso de la web para encontrar patrones de navegación, posteriormente estos se analizan y se generan reglas para la toma de decisiones por medio de la herramienta Weka. Generalmente la interfaz de usuario es la base para construir una IUA porque se necesita de una IU previamente construida y funcionando para la obtención de los datos, con el propósito que el agente pueda retroalimentarse de esta información y generar sugerencias de acuerdo al perfil de los usuarios. Se define a una IUA como aquella que muestra sugerencias a los usuarios para ayudar a resolver las tareas guiándolo a través de la interfaz. El agente autónomo adaptativo se construye para hacer los cálculos que sean necesarios y generar sugerencias, es el encargado de procesar los datos y constantemente se retroalimenta de las acciones del usuario para decidir que sugerencia se adapta a sus intereses.

4.3 Implementación de la IUA sobre Moodle.

En este punto, para construir una IUA sólo falta explicar cómo clasificar el comportamiento de los usuarios que usan esta aplicación, clasificar al usuario es la parte más importante para construir este agente, ya que al observar sus comportamientos y compararlos con los clústeres se puede entender cuáles son las secciones o actividades donde los usuarios necesitan ayuda de manera más más específica, pero no parecida algunos recursos pueden variar. Las reglas del autómata se van construyendo en base a que conocemos el comportamiento de los usuarios, cómo detectamos en que actividades pueden tener problemas, con los puntos mencionados podemos generar las reglas para la construcción de la sugerencia.

Para iniciar es necesario identificar el comportamiento del usuario, se hace con respecto a qué clúster corresponde su interacción actual, esto es porque cada clúster representa cierta navegación común entre los usuarios.

Para poder asignarle un clúster a un usuario se debe construir un clasificador usando Weka, en particular, utilizando el algoritmo llamado One Rule.

One Rule (OneR): Es un algoritmo de clasificación que genera un árbol de decisión de un único nivel y es capaz de inferir reglas de clasificación a partir de un conjunto de instancias. El algoritmo crea una regla para cada atributo en los datos de entrenamiento, luego escoge la regla con la tasa de error más pequeña. Para crear una regla por cada atributo debe determinarse la clase más frecuente.

El algoritmo OneR implementado en WEKA es muy efectivo deduciendo una regla basada en un único atributo. Las desventajas que se pueden advertir en este algoritmo son:

- El algoritmo trata todos los atributos numéricamente evaluados como continuos, usa un método directo para dividir el rango de valores en intervalos disjuntos. Esto introduce un riesgo de sobreajuste en el caso de atributos evaluados de forma continua, por ejemplo: identificadores únicos secuenciales, números de teléfono.
- El *"sobreajuste*" de atributos nominales con valores cercanos o únicos como: nombres de personas, direcciones de correo electrónico, etc.
- Selección aleatoria de un atributo cuando las tasas de error son iguales.
- Selección aleatoria de una clase cuando dos o más clases dan la misma tasa de error con un atributo.

Previamente a este paso, los clústeres ya se habían obtenido al aplicar el algoritmo k-medias previamente. Desde la pestaña *Clúster*, dando clic derecho sobre el modelo generado escogemos la opción *Visualize cluster assignments* (figura 26).

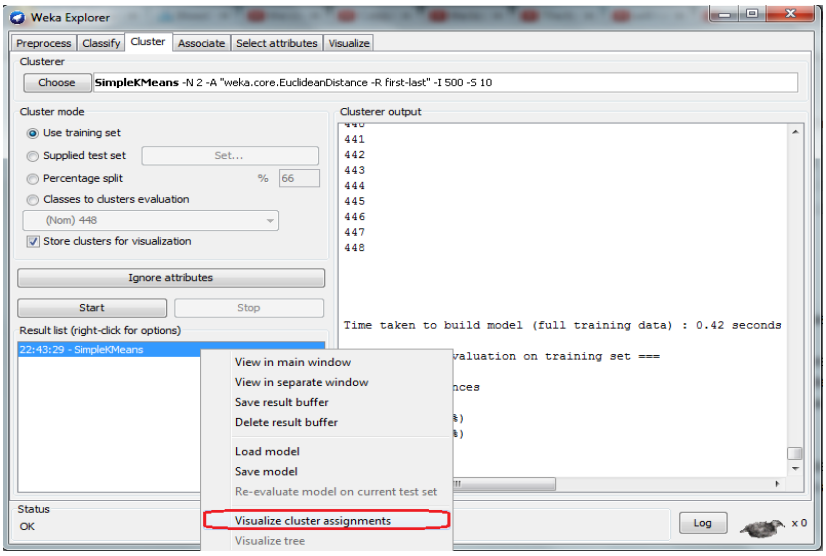

Figura 26: Opción para visualizar clústeres

Al elegir esta opción, se genera una nueva ventana que muestra la distribución de los datos con respecto al clúster asignado, dándole un color diferente a cada uno (figura 27) y existe la opción de guardarlo como un archivo de tipo *arff*, en el cual se incluye una última columna, ésta indica a que clúster pertenece cada individuo, el archivo será utilizado para entrenar nuestro clasificador.

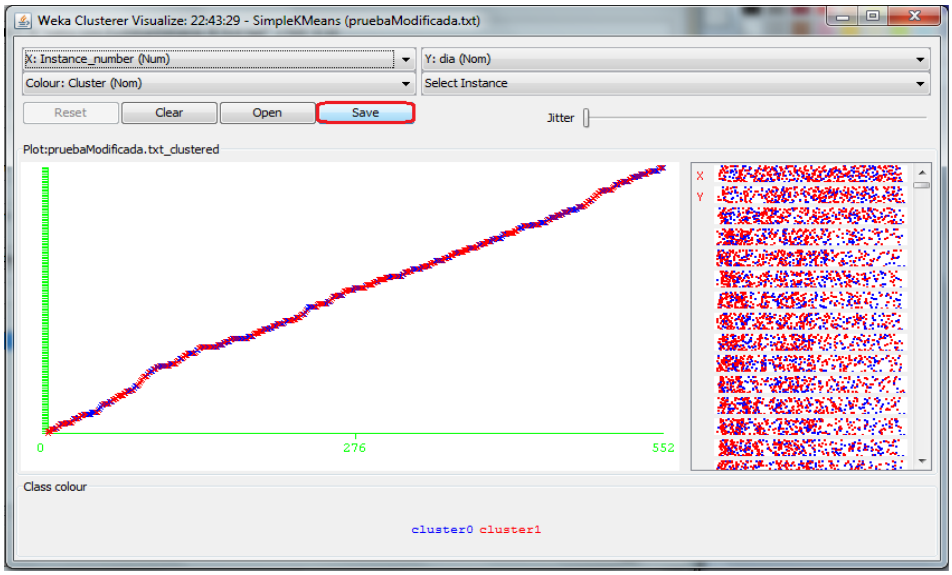

Figura 27: Dispersión de los clústeres 0 y 1

Se carga este nuevo archivo en el explorador de Weka y se pulsa sobre el botón *Open file*, se selecciona el archivo que se generó anteriormente y se observa que contiene una última columna llamada clúster (figura 28).

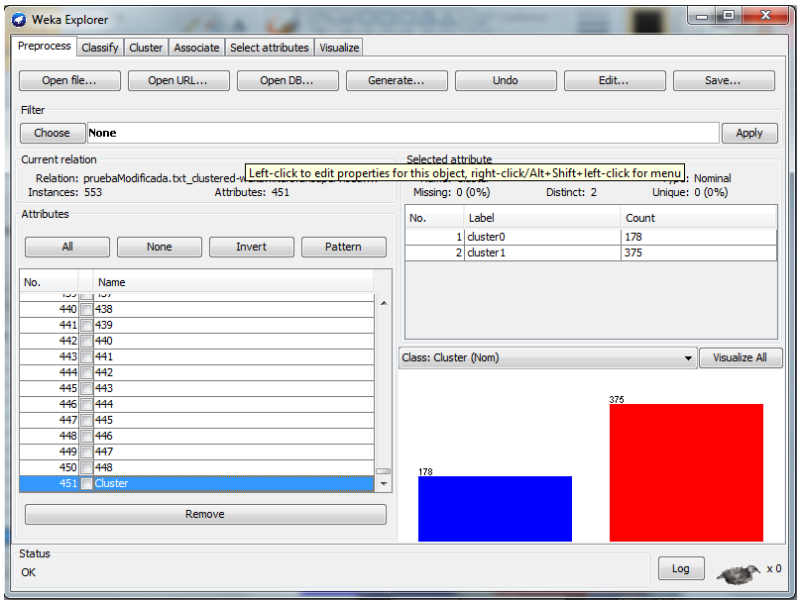

Figura 28: Columna clúster

Pulsamos sobre la pestaña *Classify*, en seguida escogemos el algoritmo *OneR*, por último pulsamos *Start*.

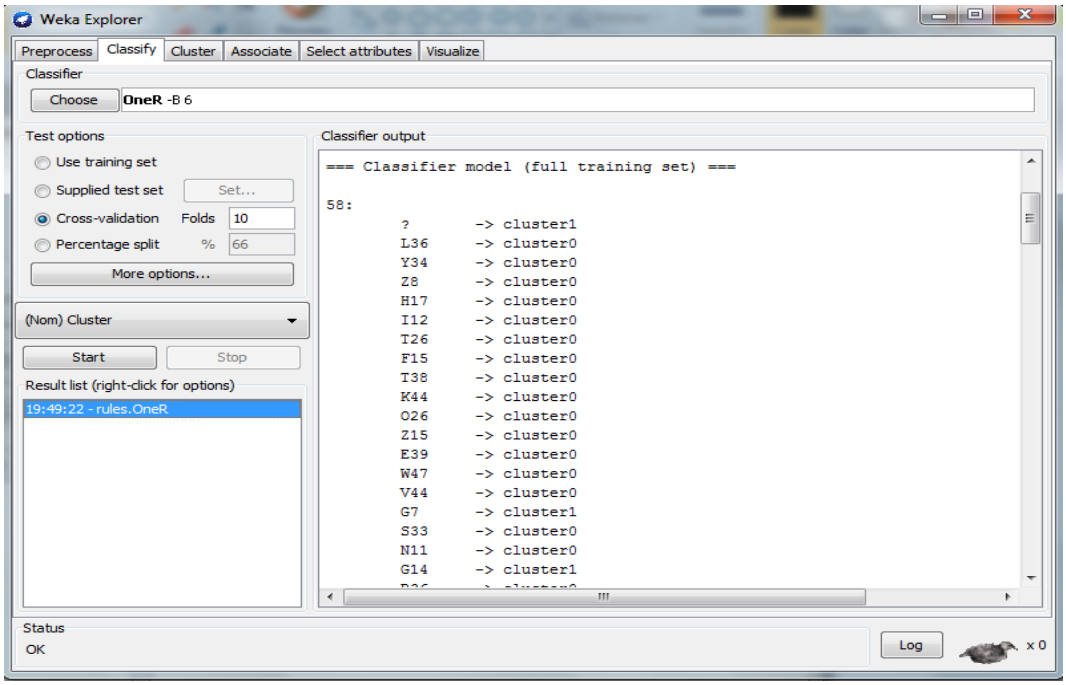

Figura 29: Reglas de clasificación según el clúster

En la figura 29 se muestra el conjunto de reglas generadas, el objetivo de estas reglas es clasificar a los usuarios, esto se hace a partir de que el usuario ha hecho algunos movimientos sobre la interfaz, se toma el último vínculo visitado y se busca en las reglas, si éste se encuentra en el listado de la figura 29 entonces le asignamos el clúster

correspondiente que pueden ser c = {0,1, 2, 3, 4}. Posteriormente deben construirse las reglas de asociación en base a lo que se observa en los clústeres obtenidos con k-medias, como representan un comportamiento general es necesario analizar cómo resolver los problemas que describen estos clústeres, por ejemplo: es posible saber que recursos son más visitados, las secciones del Moodle que los alumnos utilizan más, que documentos PDF son más leídos y también es posible basarse en el temario del profesor para ver que se está viendo en clase, es la forma más razonable de crear las reglas para el autómata haciendo el desglose de las actividades del curso, por ejemplo: si tenemos la resolución de una tarea los alumnos deberán leer sobre un tema en concreto y el profesor es el responsable de subir ese material, por lo cual se sugerirán los recursos que el profesor proponga para esta parte de su curso.

Un breve análisis de los clústeres da información acerca de qué reglas pueden construirse es posible mapear los clústeres a la interfaz, las rutas de los códigos pueden obtenerse de la tabla recursos para tener una idea clara sobre el área de Moodle a la que pertenecen.

El clúster 1 y el clúster 3 son los que necesitan más ayuda porque en el clúster 1 se observa que el comportamiento general es ir inmediatamente al apartado de novedades donde se accede a ver tareas; sin embargo, ya no hace nada. Entonces la sugerencia consistiría en mostrarle todos los recursos de esa semana con el propósito de que alguno de ellos pueda resolver esa tarea. Por otro lado, el clúster 3 presenta un recorrido desorganizado, así que cuando de clic a un recurso se le mostrarán los recursos relacionados con éste, de esta manera irá a una sección en concreto y dejará de buscar por todas partes.

El clúster 4 muestra el comportamiento donde recorre de manera ordenada la parte central del sitio, pero esto puede complicar las cosas si existen muchos recursos al buscar el último, no es muy agradable para el usuario. Para evitar que busquen de esta manera se sugieren los recursos actuales de la semana.

Y por último, los clústeres 0 y 2 son los que contienen la navegación más corta donde están en espera de alguna novedad en el sitio o solamente han dejado abierta la sesión en el navegador, así que si el usuario no está viendo la página de la aplicación no tiene mucho sentido pero si ha accedido y la está viendo, se le recomienda los recursos de la semana.

Con el análisis de estos clústeres sobre la interfaz, resalta el hecho de que hay comportamientos especiales en ciertas áreas del Moodle y es posible formar una serie de reglas que construyan sugerencias dado el caso que se presente.

Las reglas que cubren mejor estos comportamientos son las siguientes:

1. Si el usuario es nuevo deberá registrarse e inscribirse a un curso. Hasta que pueda entrar al curso no se sabe nada de su perfil y no puede ser clasificado así que debemos esperar a que comience a interactuar y se obtengan algunos eventos.

2. Si el usuario ha interactuado con la interfaz y por lo menos ha generado algunos eventos, obtenemos el último evento y lo comparamos con las reglas para asignarle el clúster correspondiente, pero si el registro no se encuentra en las reglas de clasificación que obtuvimos anteriormente entonces se le asigna el clúster #4 por defecto.

3. Si el usuario ya tiene un clúster asignado entonces se muestra la sugerencia, el agente mostrará todos los recursos relacionados con respecto a la tarea que se debe resolver :

    a. Los recursos se muestran por orden de acuerdo a la fecha de subida a la plataforma, si el usuario ya visitó el primero, entonces este recurso se intercambiará a la última posición y así sucesivamente con los restantes.

El resultado de este módulo es la sugerencia que es una cadena de caracteres, se debe construir una función la cual reciba esta cadena y genere dinámicamente una sección en el HTML donde se añada esta cadena como un elemento de tipo vínculo. Se observará una pequeña ventana en la parte inferior derecha de la interfaz donde aparecerán las sugerencias. Esta implementación es un script que se ha añadido al Moodle.

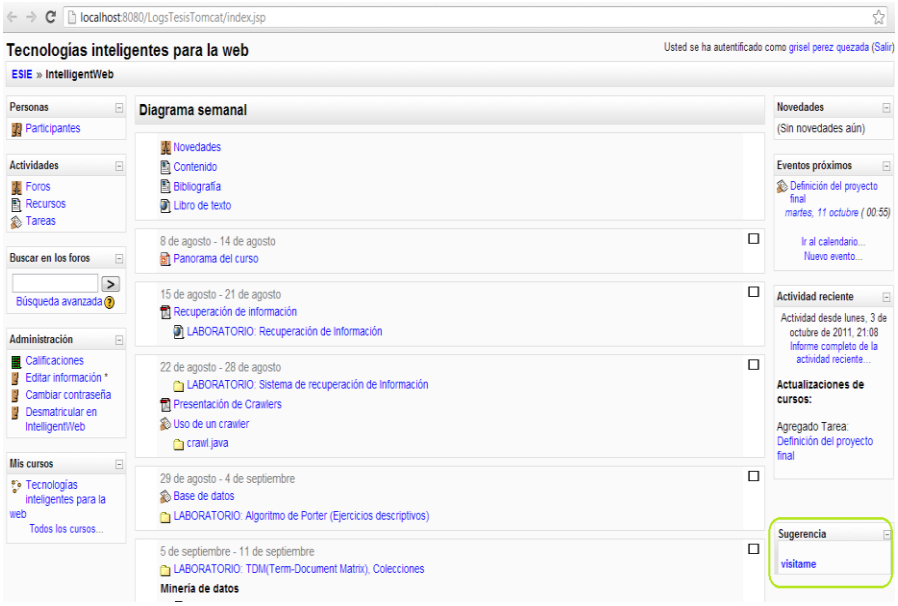

Figura 30: Integración del AAA a la IUA

4.4 Resumen.

En este capítulo se explicó el concepto de agente autónomo adaptable para una interfaz de usuario adaptativa. En el capítulo 1 se define que una interfaz de usuario adaptiva es aquella que ayuda a los usuarios a poder resolver las tareas de manera más eficiente y lo guía a través de la interfaz con la finalidad de hacer más sencillo su uso, apoyándose de la información generada sobre sí misma.

El agente autónomo adaptivo es la entidad que hace que la interfaz pueda ser más amigable para el usuario sin tener que ser una presencia intrusiva hacia las acciones de los usuarios y su característica principal es que se retroalimenta de las acciones del usuario para decidir una sugerencia que se adapte a los intereses del usuario, también tiene la capacidad de ser autónomo ya que él mismo puede encontrar los perfiles y por último utiliza las reglas programadas para tomar la decisión de asignar la mejor sugerencia que puede servirle al usuario.

La herramienta Weka fue usada para construir las reglas que clasifican al usuario con los mismos datos que se usaron para generar los clústeres junto conla descripción de este proceso.

El análisis de la interfaz y de los clústeres en conjunto se ha interpretado con un modelo simple: mediante la suposición de las posibles acciones que pueden tomar los usuarios al estar frente a determinada tarea; sin embargo, aplicando algún modelo de aprendizaje podrían generarse reglas más exactas.

Capítulo 5

Análisis del Agente Autónomo Adaptable.

El agente construido en este proyecto ha sido integrado a una plataforma educativa llamada Moodle. Se ha observado que los usuarios tienen problemas para poder realizar algunas de sus tareas, en particular les es difícil encontrar los recursos que buscan. Este sistema no contiene un mapa del sitio por lo que a los usuarios les es complicado saber en qué sección se encuentran, también en algunos cursos podemos encontrar demasiado contenido en las páginas sobre todo en la página inicial donde se despliegan todos los recursos. Si el usuario ve una sobrecarga de información esto podría posiblemente causar desagrado y quitarle la intención de revisar el sitio.

El agente observa la navegación del usuario y lo clasifica, encontrando los patrones con el algoritmo ***k-medias*** y con ayuda del algoritmo ***one rule*** puede decidir de forma muy poca costosa a que clúster pertenece, como ya se explicó en el capítulo 4. El costo computacional es muy bajo ya que es un algoritmo muy simple basado en comparaciones entre los datos, cuyo objetivo es encontrar la regla que produce menor tasa de error.

A partir de la clasificación, construimos un sistema basado en reglas que dirige el comportamiento del agente para encontrar la sugerencia más apropiada para el usuario. El procesamiento de estas reglas se basa en: el temario del curso, el análisis de la interfaz y los clústeres encontrados, hacemos uso de ellos para calcular el recurso que está relacionado o contenido en alguna semana para hacer las recomendaciones.

Las reglas con las que trabaja el agente son generadas a partir del análisis de los clústeres, ya que los clústeres nos permiten visualizar los recursos que son más interesantes para los usuarios a partir de la organización propuesta para cada curso. De acuerdo al criterio y temario propuesto por cada profesor es posible usar el material que el profesor tiene disponible para dar su curso para ubicar los recursos que aparecerán en la sugerencia generando un conjunto de reglas más precisas. Como puede verse estos son algunos factores con los que se debe trabajar.

Los usuarios no tienen conocimiento alguno sobre el rastreo de su información, los datos capturados fueron usados para mostrar el proceso de construcción de una IUA y se registró la navegación de todos los alumnos pertenecientes al curso de inteligencia artificial del semestre 2012-2, se utilizaron los datos obtenidos de febrero a junio de 2013, con 35 alumnos inscritos en éste curso.

5.1 Descripción del modelo.

Para mostrar el desempeño del agente se ha escogido un usuario que utilizó la plataforma. Siguiendo el proceso mencionado anteriormente, se mostrarán algunos resultados al tener este tipo de agente sobre una plataforma educativa.

Una característica importante que se ha percibido es que si se utiliza este agente en un curso o semestre posterior, funciona utilizando las reglas generadas, haciendo cambios mínimos sobre la asignación de los recursos a las semanas, ya que éstas varían según el año. El agente fue probado en el curso de inteligencia artificial 2014-1 con la versión de Moodle 1.6, agregar este agente a la nueva versión del Moodle 2.1 lleva el mismo procedimiento: se insertan los estilos donde se mostrará el agente y el script que captura la navegación del usuario. Esta última versión de Moodle muestra programación orientada a objetos mientras que versiones anteriores es estructural. En dado caso que se tuviera una versión más reciente de Moodle, lo único que debemos hacer es incrustar nuestros scripts en la cabecera del HTML de la interfaz para que estos funcionen.

El curso de Moodle en que el agente trabaja es en el de inteligencia artificial. En el caso base tenemos un nuevo alumno que tiene la IP: 178.247.110.999. El usuario introduce su login y su contraseña para poder entrar al curso.

En el capítulo 3 se definieron 5 clústeres, donde el clúster #2 representa el patrón de inicio de sesión con la secuencia X8-I49 que son los recursos que representan los campos password y el login. El agente no se hará presente en este caso.

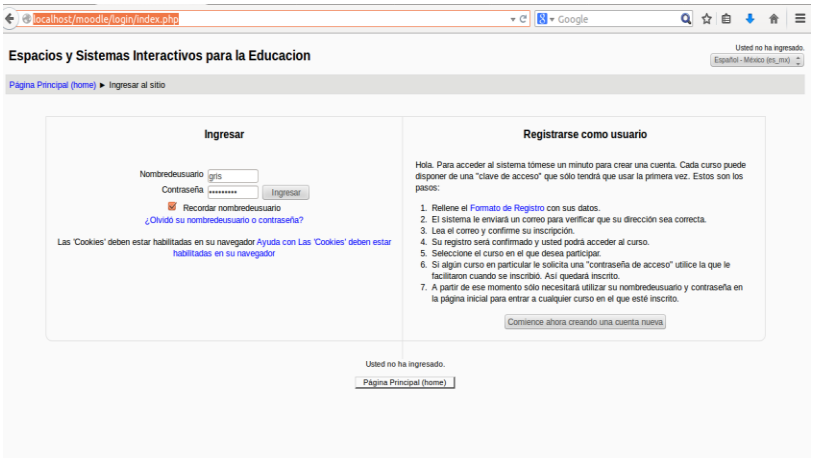

Figura 31: Página principal de Moodle

Enseguida, el usuario entra a su sesión le muestra todos los cursos a los que está inscrito como puede observarse en la siguiente figura:

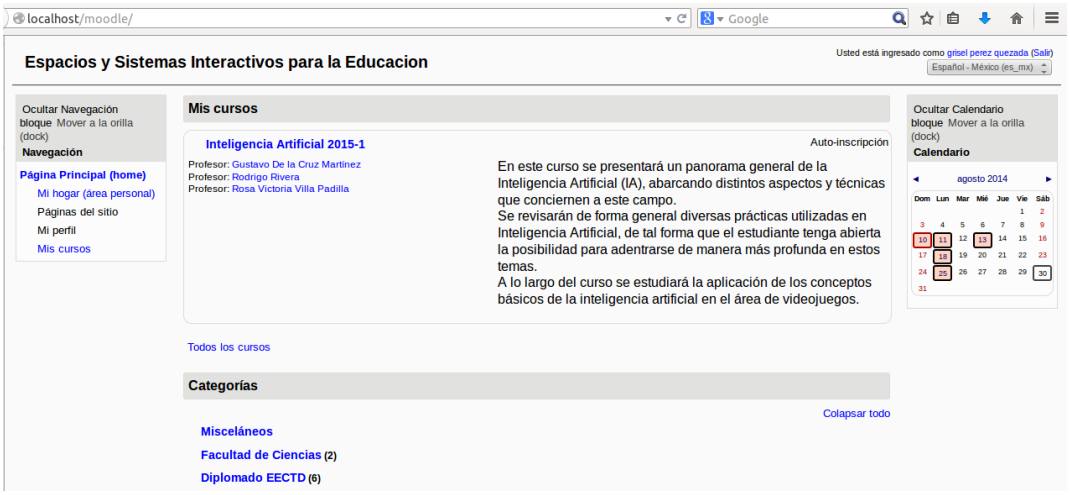

Figura 32: Mis cursos en Moodle

Para iniciar ha quedado inscrito al curso de inteligencia artificial donde se mostrarán las sugerencias. El agente sobre esta sección aún no tiene aparición, ya que los datos que se muestran es información personal del usuario y la administración de su cuenta.

Para ilustrar el comportamiento de la propuesta se describen diferentes casos para mostrar las recomendaciones con respecto al contenido del curso. Las sugerencias se presentarán en las páginas del curso sobre una sección en la parte inferior derecha.

Se recuerda que el comportamiento general de los clústeres fue presentado en el capítulo 3. Los clústeres 1 y 3 presentan a los usuarios que tienen un comportamiento desordenado; visitan varias veces alguna sección y luego pasan a otra hasta que encuentran lo que buscaban o desisten de ello. En el clúster 4 se observa un comportamiento más ordenado, visita los recursos por semana de manera ordenada y es el clúster asignado por defecto. En el caso de tener un usuario nuevo que se encuentre en la página inicial y que no haya interactuado se le asigna el clúster por defecto, porque el agente aún no tiene información sobre el usuario.

El agente debe verificar la información más reciente del usuario, es decir, que recursos ha visitado en cada página lo que lleva a que por cada diferente página que visite se debe volver a clasificar al usuario para poder generar la sugerencia actual. Si el usuario ya ha interactuado con la interfaz entonces puede ser clasificado y en el siguiente recurso que visite se mostrará una sugerencia de acuerdo a lo que ha visitado.

Para comenzar se considera que el usuario es nuevo y que la interacción comienza en la página inicial del curso, la sugerencia muestra todos los recursos que introducen al usuario en el contenido del curso, como se muestra en la figura 33. Se sugieren todos los recursos básicos del curso propuestos por el profesor, por ejemplo el temario, bibliografía, entre otros, con el propósito de dar un panorama general de lo que se verá en el curso.

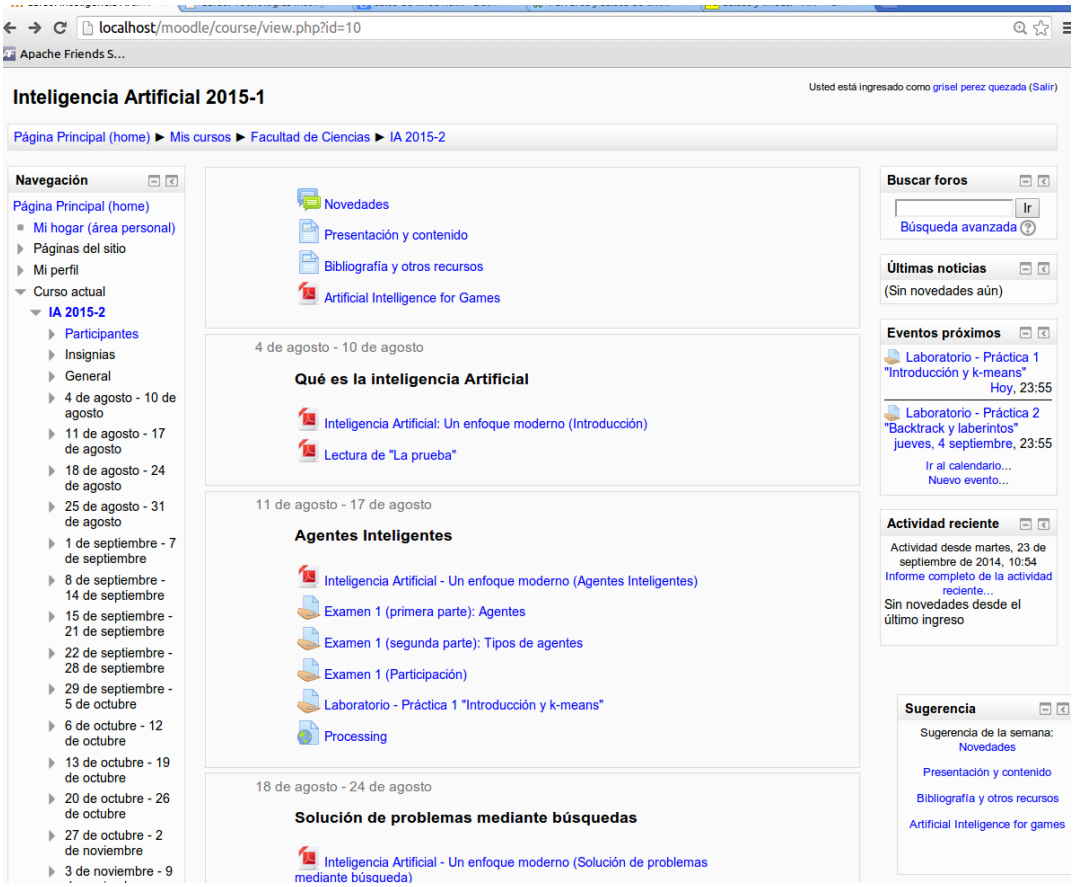

Figura 33: Pagina que contiene los recursos del curso en Moodle

Si el usuario ha interactuado con el sistema, entonces ya puede ser clasificado con los criterios mencionados en el capítulo 3, dependiendo de su navegación se busca en este caso cual es el último recurso y se compara con el conjunto de reglas generadas por One Rule.

De esta manera el usuario podrá observar la sugerencia asociada a los recursos que están relacionados con las actividades que está viendo en ese momento. La semana actual en la que se ejecuta esta prueba es la cuarta semana de clases, por lo que en la página inicial el agente muestra los recursos correspondientes a dicha semana que comprende del 25 al 31 de agosto. Esta es una ventaja para el usuario, ya que no tiene que usar el scroll y recorrer los recursos hasta llegar a la semana actual para consultar los recursos que serán vistos a lo largo de esta semana, esto puede verse en la figura 32.

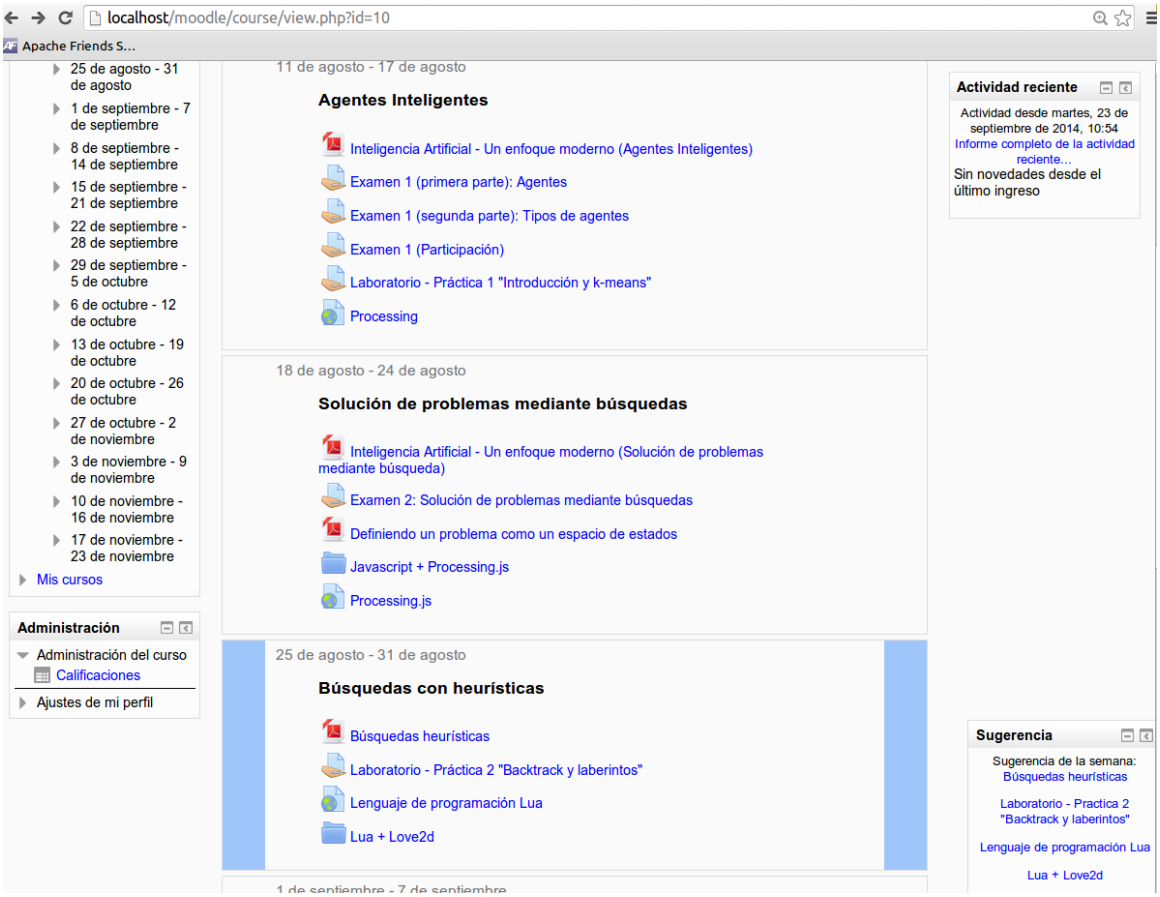

Figura 34: Ejemplo de los recursos de la semana 4

En este ejemplo estamos en la semana #4 que comprende del 25 al 31de agosto y contiene estos 4 recursos, la sugerencia también los contiene.

1. Búsquedas heurísticas
2. Laboratorio – Práctica 2 “backtrack y laberintos”
3. Lenguaje de programación Lua
4. Lua + love2d

El agente debe reconocer que cuando el usuario elige un recurso de una semana, debe actualizar sus recomendaciones y determinar que los recursos más útiles son los de la semana actual, además debe intercambiar el orden de los recursos que ya fueron visitados hasta el final, pero conservando siempre el orden propuesto por el profesor.

Existe una excepción, ya que algunos de los recursos apuntan a PDF’S o a páginas externas donde es imposible que el agente pueda mostrar una sugerencia. Sólo podrá hacerlo si las páginas visitadas son parte de la aplicación Moodle.

En este caso, el usuario elige el recurso "***Búsquedas heurísticas***" de la semana 4 que apunta a un PDF del curso como se muestra a continuación:

Figura 35: Ejemplo de recursos externos o archivos

Una vez que el navegador muestra el archivo PDF, la única manera de regresar al curso es el botón ***atrás*** del navegador y se regresa a la página inicial del curso, pero como el vínculo con nombre "***Búsquedas heurísticas"*** ya fue visitado este se intercambia hasta el final del listado y el agente nos sugiere el siguiente recurso de la semana que no ha sido visitado y que está relacionado con las actividades de esta semana que es:

2. ***Laboratorio – Practica 2 "backtrack y laberintos"***

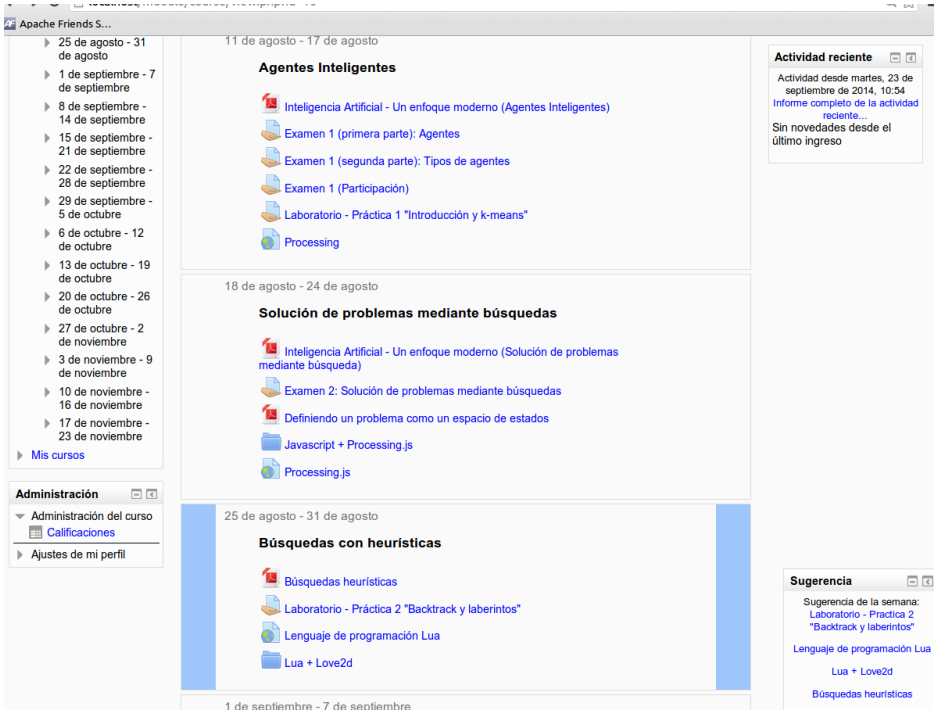

Figura 36: Siguiente recurso de la semana 4

Por último, en caso de que el usuario revise un recurso de semanas anteriores o alguno posterior, el sistema automáticamente le sugerirá las actividades de la semana actual esto se debe a que cada vez que el usuario visita una página, es clasificado con la navegación actual y el agente en base a esta selección muestra las sugerencias personalizadas.

5.2 Fallas.

Algunas fallas pueden presentarse y tal vez desorientar al usuario a causa de que las áreas del Moodle son extensas, en esta simple implementación se hicieron las reglas para llevar la secuencia de las visitas a los recursos. Sin embargo para poder sugerir un foro, no tenemos ningún punto de partida para decidir si algún foro está relacionado con algún recurso o tarea. Los recursos pueden sugerirse por que el maestro lleva un temario por semana, para solucionar el problema de los foros, tendríamos que preguntar explícitamente al profesor sobre el contenido y tema de cada foro. Por el momento se manejan los foros y situaciones que no se pueden manejar desde el sistema de la siguiente manera:

- Si el usuario entra a una tarea cualquiera el sistema por default le dará la recomendación de ingresar a los recursos relacionados con la semana actual.

- Si el usuario entra a uno de los temas del foro, el sistema recomendará por default los recursos relacionados con la semana actual.

- Si el usuario en su sesión anterior no pudo terminar de leer algún recurso, entonces cuando vuelva a iniciar otra sesión, este esperaría continuar leyendo donde se quedó, se sugerirá los recursos relacionados con la actividad.

Por otra parte existe una falla que puede ser controlada por el agente, cuando se tiene un recurso que es externo al Moodle, por ejemplo, la referencia de un artículo en internet o un documento creado en google, éstos no contienen la estructura de una liga en Moodle, es decir, no contiene el id en la ruta, estas ligas generan un error y se decidió en este caso ignorar estos recursos para evitar que el módulo deje de funcionar.

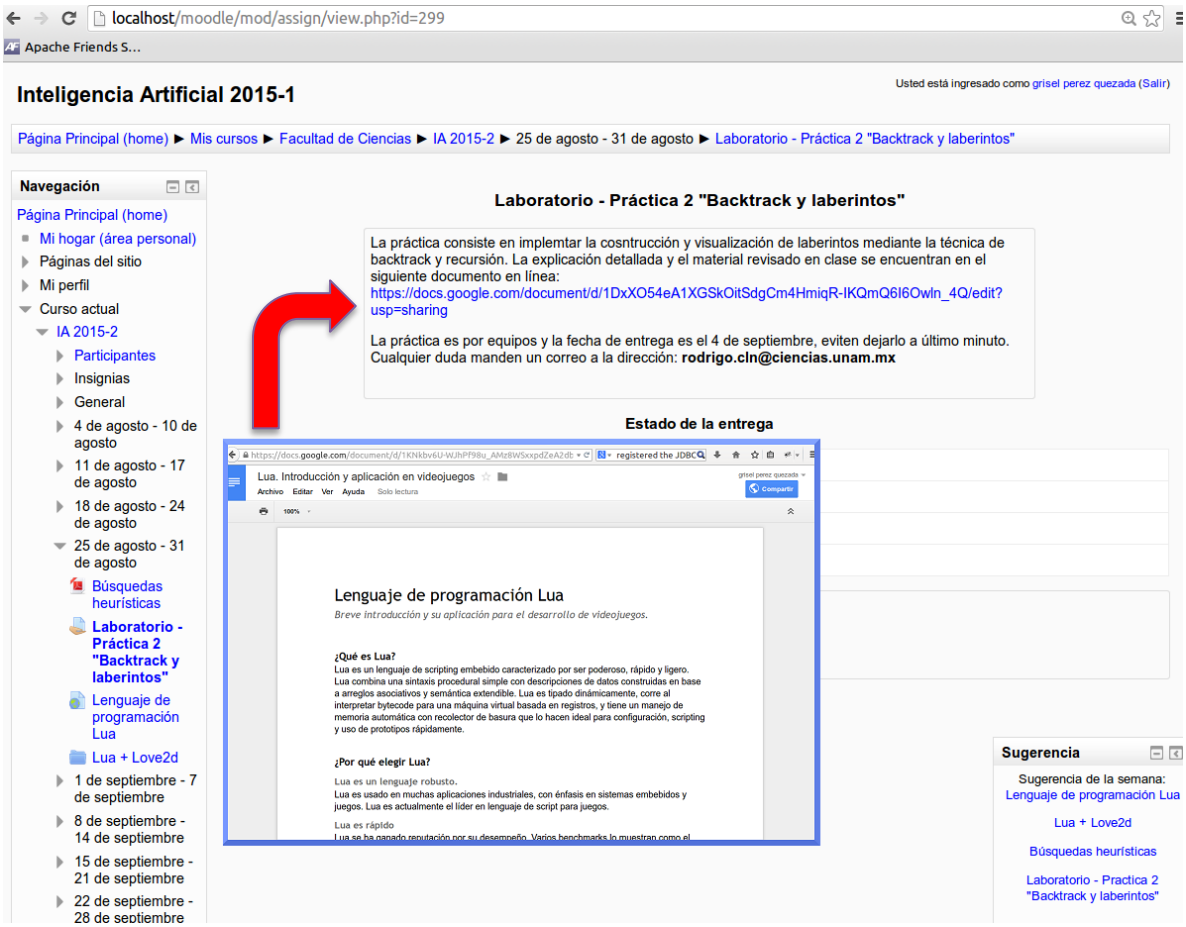

Figura 37: Recurso externo a Moodle

Pero realmente la solución es muy sencilla, filtramos todas los vínculos que son de este tipo y las guardamos en una lista, estos vínculos tendrán un procesamiento diferente y se necesitará de la ayuda del profesor para poder recomendar adecuadamente, esto generará otro conjunto de reglas que se añadirían al agente.

5.3 Ventajas y Desventajas sobre su uso.

Ventajas:

1) Como el agente hace un análisis en base a los recursos visitados previamente por el usuario, existe cierta tendencia a que resuelvan la tarea de la misma manera o muy similar, por lo que habrá ciertos recursos que sobresalgan más que el resto y esos serán los elegidos para hacer la sugerencia. Por lo que la sugerencia será muy precisa en cuanto al tiempo en el que se encuentra la sesión.

2) Los patrones observados sobre la navegación generan un conjunto de reglas las cuales ayudan al agente a decidir qué recurso puede ser el más indicado de acuerdo a los intereses del usuario, así es que la sugerencia será personalizada.

3) Una sugerencia de este tipo ayuda a que los estudiantes puedan encontrar recursos que ayuden a resolver su tarea más rápidamente y sin buscar demasiado en la interfaz de Moodle.

4) Los alumnos siempre tienen la información adecuada y actualizada para resolver todas las tareas.

Desventajas:

1) Si la sugerencia no es útil en la búsqueda de la solución de la tarea, entonces el usuario no lo utilizará y la ignorará.

2) Si por alguna razón el usuario se atrasa en el curso, podría causar confusión la sugerencia provista por el agente, ya que este podría sugerir un recurso distinto al que se quedó el alumno.

3) El agente puede sugerir, más el usuario es el que decide si es de su interés.

4) Es necesaria la opinión del profesor para determinar la importancia de los recursos, si se implementara algún modelo de aprendizaje sobre plataformas educativas se podría prescindir del profesor.

Capítulo 6

Conclusiones.

6.1 Panorama general.

En este proyecto de tesis se ha mostrado como construir una interfaz de usuario adaptativa, iniciando con el uso de las teorías de dos autores Mobasher y Brusilovosky que han estudiado ampliamente el campo de la interacción y aprendizaje del usuario sobre las interfaces web. Ellos describen que es posible reconstruir la navegación del usuario y después analizar su interacción dentro de la aplicación y observar sobre qué elementos ha interactuado. Por medio de estas observaciones podemos saber los intereses que tiene cada usuario de forma particular. El modelo de (De la Cruz, G., 2011), propone extender la captura de todos los eventos generados por los elementos interactivos que contiene una interfaz, con el objetivo de descubrir intereses o recursos que llamen la atención de cada usuario sin que éstos lo muestren explícitamente.

Para la captura de la información se utilizó una herramienta llamada log4j que ayudó a generar bitácoras personalizadas siguiendo un formato específico muy parecido al de un servidor web pero con la adición de los campos que contienen los recursos, tipo de evento, hora y fecha en que se disparó el evento, para darle seguimiento a la navegación del usuario. Log4j tiene la opción de poder guardar directamente los datos a una base de datos con los archivos de configuración de la herramienta.

Las bitácoras deben ser pre-procesadas para quitar valores nulos o valores que pueden alterar los resultados en el descubrimiento de nueva información. En la etapa de la aplicación de técnicas de minería de datos se encontraron ciertos datos que alteraban los resultados como: todos los usuarios que provenían de una intranet salen con la misma IP esto dificulta la tarea de diferenciar que usuario usó la interfaz, así como los datos nulos generados por los eventos del navegador.

Al quitar los valores que alteran los resultados debemos utilizar minería de datos para encontrar patrones ocultos en la navegación del usuario, se utilizaron técnicas de minería de datos en específico los agrupamientos para la construcción de clases que contengan a los usuarios con comportamiento similar. Utilizando el algoritmo k-medias y convirtiendo la navegación en secuencias es posible usarlo para encontrar distancias entre secuencias, que da como resultado un conjunto de clústeres, posteriormente se aplica el algoritmo oneR para la clasificación de los usuarios.

Finalmente, con un análisis de los clústeres encontrados sobre los elementos de la interfaz se construyen las reglas para el agente autónomo adaptable que se encarga de la construcción y visualización de las sugerencias.

La IUA muestra a cada usuario una sugerencia de acuerdo a sus intereses, en un recuadro en la parte inferior derecha; si el usuario es nuevo sugerirá los recursos que lo orienten acerca del contenido del curso. Cada vez que el usuario visite algún recurso el agente sugerirá el contenido relacionado a ese recurso, intercambiando la posición de éste si ya fue visitado anteriormente.

6.2 Observaciones

El modelo que se describió en la sección anterior es una forma de implementación de una IUA, las teorías que escogimos en este caso hipermedia adaptativa y personalización de la web son dos vertientes que en lo personal resuelven el problema de identificar como los usuarios dejan rastro de su actividad en el sistema.

Utilizar otro tipo de teoría no podría ser posible ya que se necesitan los datos de la interacción entre el sistema y el usuario para poder medir el buen uso. La mayoría de las interfaces contienen objetos interactivos y estáticos, pero si contiene elementos estáticos como imágenes, éstas no generan eventos y sería imposible encontrar el rastro del recorrido del usuario.

Con respecto al análisis de datos en la fase de minería de datos existen varios algoritmos y funciones con ciertas características, es importante tener el conocimiento de las tareas que ayudan a resolver este problema.

El modelo propone que a partir de este paso se construya un agente autónomo adaptable; sin embargo, existen varios tipos de agente que cuentan con otras características. Si se construyera otro tipo de agente no podríamos modelar reglas coherentes para que el agente genere una sugerencia acertada. Es por esta razón que las características de autonomía y adaptabilidad son las ideales para poder generar reglas muy precisas para cada usuario.

Es un modelo bastante sencillo de realizar porque utilizamos herramientas de licencia libre con lo que ha sido suficiente para la captura, procesamiento y análisis de los datos, el procesamiento de los datos no incapacita a la máquina, los algoritmos son muy ligeros y no afectan ningún proceso. Sin embargo, es necesario hacer los análisis correspondientes para lograr tener sugerencias certeras, además de hacer una evaluación para medir su eficiencia.

Con respecto a las reglas construidas para modelar al agente sobre la plataforma Moodle, sería posible tener reglas más específicas para describir lo que puede o no hacer el agente si se pudiera tener más información acerca de cómo cada profesor imparte su clase, ya que es

muy ambiguo apoyarse en un temario y basarse en las fechas u horas que se muestran ahí, porque en realidad ocurren acciones que alteran este orden.

Por otra parte el modelo cognitivo utilizado consiste en la repetición de las acciones sobre la interfaz, básicamente las personas cuando aprenden a utilizar una interfaz acceden a los mismos elementos una y otra vez, en el caso de la aplicación Moodle los alumnos se desplazan a través de tareas específicas sobre el área que más les atrae, es por esta razón que es posible obtener el parecido entre secuencias y es de mucha utilidad en este caso el algoritmo k-medias.

6.3 Trabajo a futuro.

El agente no ha sido evaluado para determinar si es útil para los usuarios; es decir, hacer pruebas para determinar si las sugerencias que se muestran en realidad sirven al usuario.

Solucionar el problema donde los usuarios provienen de una intranet usando el usuario con el que ingresa en la plataforma. Extender el análisis para la generación de sugerencias para profesores y administradores.

Introducir un modelo educativo de aprendizaje sobre la web para sustentar de manera más confiable la propuesta de las reglas usadas para este agente.

Rehacer todo el proceso para construir la IUA con la misma materia, pero de otro año, con el propósito de analizar si las reglas sirven para grupos posteriores o si existen patrones nuevos que no habían aparecido.

Hacer un análisis para determinar si la posición del agente es el adecuado, observar si es visto por los usuarios.

Bibliografía

- Agrawal, R., Srikant, R. (1994). Fast Algorithms for Mining Association Rules. CA: IBM Almaden Research center.

- Brusilovsky, P. (2007). Adaptative Navigation Support. En brusilovsky, Peter, Alfred Kobsa y Wolfrang Nejdl (editores): The adaptative Web, volume 4321 de Lecture Notes in Computer Science, páginas 263-290. Springer Berlin / Heidelberg, 2007, ISBN 978-3-540-72078-2. http://dx.doi.org/10.1007/978-3-540-72079-9_8, 10.1007/978-3-540-72079-9_8.

- Cáceres González, A. (2008). La métrica de Levenshtein. Revista de ciencias básicas UJAT p 35-43.

- Césari Matilde. (sin año) Aprendizaje automático con Weka. http://ccia.ei.uvigo.es/docencia/MRA/practicas/MATERIAL_WEKA.pdf (visitado 11/06/2014)

- Chakrabarti, S. (2003). Mining the web: Discovering knowledge from hypertext data. SF: Morgan Kaufmann Publishers.

- De la Cruz Martínez, G., y Gamboa, F. (2011). Using user interaction to model user comprehension on the web navigation. International JOURNAL OF Computer Information Systems and Industrial Management Applications, 3:878-885, 2011.

- Dourish P., y Chalmers M. (1994). Running Out of Space: Models of Information Navigation. Cambridge: Rank Xerox Research Centre.

- Fayyad, U. (1996). From Data Mining to Knowledge Discovery in Databases. AI MAGAZINE.

- Hall, M., Eibe F., Holmes G., Pfahringer B, Reutemann P., & Witten Ian H. (2009) The WEKA data mining software: an update. SIGKDD Explor. Newls. http//doi.acm.org/10.1145/1656274.1656278.
- Hernandez Orallo, J., Ramirez Quintana, M. y Ramirez Ferrí, Cesar. (2004). Introducción a la minería de datos. Pearson Prentice Hall. http://books.google.com.mx/books?id=x3LuAAAACAAJ.

- Hewett et. al. (1996). ACM SIGCHI Curricula for Human-Computer Interaction.

- Kay, A. (1990). User Interface: A personal view. In B. Laurel, editor, The Art of Human-Computer Interface Design. Adison-Wesley, Reading, Mass.

- Kitajima, Muneo, Polson P., y Blackmon, M. (2007). COLIDES and SNIF-ACT: Complementary Models for searching and Sensemaking on the web. En proceedings of Human Computer Interaction Consortium (HCIC) Winter Workshop. http://autocww2.colorado.edu/~blackmon/Papers/KitajimaPolsonBlackmonHCIC3355_2007.pdf

- Log4j, http://logging.apache.org/log4j/2.x/ (Visitado 24/05/2014).

- Mas, A. (2005). Agentes de software y sistemas multi-agente: conceptos, arquitecturas y aplicaciones. Madrid: PEARSON EDUCACIÓN, S.A.

- Maes, P. (1994). Agents that Reduce Work and Information Overload. Communications of the ACM, pag. 31-40.

- Mandel. (1997).Theo:The elements of user interface design. NY:Jhon Wiley & Sons. ISBN 0-471-16267-1.

- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

- Mobasher, B. (2007). Data Mining for Web Personalization.En brusilovsky, Peter, Alfred Kobsa Y Wolfgang Nejdl (editores): The Adaptive Web, volumen 4321 de Lecture Notes in Computer Sciene, páginas 90-135. Spinger Berlin / Heidelberg, ISBM 978-2-540-72078-2. http://dx.doi.org/10.1007/978-3-540-72079-9_3, 10.1007/978-3-540-72079-9_3.

- Ostendorp, H. y Juvina I. (2007). Using a cognitive model to generate web navigation support. International Journal of Human-Computer Studies. http://wwwsciencedirect.com/science/article/pii/S1071581907000845.

- Pathalizer, http://pathalizer.sourceforge.net/ (Visitado 10/07/2014).

- Pirolli, Peter, 2008. Cognitive Models of Huan Information paginas 443-470. Jhon Wiley & Sons, Inc. ISBN 9780470713181 http://dx.doi.org/10.1002/9780470713181.ch17.

- Pirolli, Peter y Wai tat Fu, 2003. SNIF-ACT: a model of information foraging on the world wide web. En proceedings of the 9th international conference on User modeling. UM'03, páginas 45-54, Berlin, Heidelberg. Springer-Verlag, ISBM 3-540-40381-7. http://dl.acm.org/citation.cfm?id=1759957.1759968.

- Pirolli, Peter L. T. (2007). Information Foraging Theory: Adaptative Interaction with Information. New York : Oxford University Press, Inc.

- Rajendra A. y Pawan L. (2008) Building and Intelligent Web: Theory and practice. USA: Jones and Bartlett Publishers, Inc.

- Ross E. (2000). Intelligent User Interfaces: Survey and Research Directions. University of Bristol.

- Shneiderman, B. y Plaisant, C. (2004). Designing the User Interface: Strategies for Effective Human-Computer Interaction. Boston: Pearson Addison Wesley.

- StatViz, http://statviz.sourceforge.net/ (Visitado 1/07/2014).

- Weka API, 2012. Versión 3.7.6. http://grepcode.com/file/repo1.maven.org/maven2/nz.ac.waikato.cms.weka/weka-dev/3.7.6/weka/core/NormalizableDistance.java#500 (Visitado 20/03/2015).

- Wooldridge, M., y Jennings, N. Agentes inteligentes: Teoría y práctica -Articulo. http://www.cs.upc.edu/~bejar/aia/aia-web/wooldridge95intelligent.pdf.