UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

APLICACIÓN DE TÉCNICAS DE MINERÍA DE DATOS PARA EL APOYO A LA ADMINISTRACIÓN DE COMUNIDADES VIRTUALES DE PRÁCTICA

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL E INGENIERO CIVIL EN COMPUTACIÓN

ROBERTO ANDRÉS SILVA ÁLVAREZ

PROFESOR GUÍA
SEBASTIÁN A. RÍOS PÉREZ

MIEMBROS DE LA COMISIÓN
CLAUDIO GUTIÉRREZ GALLARDO
GASTÓN ANDRÉS L'HULLIER CHAPARRO
JOSÉ MIGUEL PIQUER GARDNER

SANTIAGO DE CHILE
ABRIL 2011

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL
E INGENIERO CIVIL EN COMPUTACIÓN
POR: ROBERTO SILVA ÁLVAREZ
FECHA: 19/04/2011
PROF. GUÍA: SEBASTIÁN A. RÍOS PÉREZ

"APLICACIÓN DE TÉCNICAS DE MINERÍA DE DATOS PARA EL APOYO A LA ADMINISTRACIÓN DE COMUNIDADES VIRTUALES DE PRÁCTICA"

Internet ha permitido la creación de distintas formas de comunicación entre los individuos, permitiéndoles intercambiar información y, en conjunto, crear conocimiento. Existe una gran diversidad de entidades sociales en la Web, tales como las redes sociales, comunidades virtuales, entre otras, donde cada una posee un objetivo y una razón de ser.

El objetivo del presente trabajo de título es diseñar e implementar una aplicación de análisis que permita proveer de información y apoyar la moderación y administración de comunidades virtuales de práctica, utilizando técnicas de minería de datos, reduciendo la carga de trabajo en esta tarea.

En este tipo de entidades, existe un integrante que posee características particulares, el administrador de la comunidad. Este individuo, debe realizar la labor de mantener el control sobre los distintos eventos que acontecen diariamente, arreglar los posibles problemas, facilitar herramientas e información, todo lo necesario para que la comunidad se mantenga acorde con el objetivo principal: crear y mantener el conocimiento. El problema de la administración, existe principalmente ya que las actividades moderadoras pueden tomar mucho tiempo, al ser una actividad básicamente manual.

La solución propuesta consta del uso de minería de datos originados en la Web, con la intención de analizar los comportamientos de usuario de los integrantes de la comunidad. Mediante el proceso de descubrimiento de conocimiento en bases de datos (KDD), se intenta encontrar un modelo de clusters o grupos de los comportamientos de manera de analizar sus características y así poder indagar en la revisión de los mensajes generados por ellos. Se utilizan dos algoritmos de clustering particional, Self-Organizing Maps (SOM) y la variante del K-means, K-Medoids. El uso de SOM tiene el propósito de encontrar la cantidad de clusters inherentes dentro del modelo.

Se proponen dos modelos sobre medidas de similitud (modelo 1) y disimilitud (modelo 2) de las sesiones de usuario, utilizando dos representaciones del contenido. Los modelos se basan en el uso de medidas que capturan los aspectos más importantes de la navegación en la Web y en características exclusivas de los foros de comunidades virtuales.

La metodología se aplica sobre el foro de la comunidad de Plexilandia.cl. Los resultados varían principalmente en la distribución de cantidad de clusters. El análisis final se basa en dos características principales, el análisis de secuencia y contenido, y en las características de los mensajes ingresados por el usuario durante su navegación. Al evaluar los modelos propuestos, se encuentra que revisando un 85% de todos los mensajes permite encontrar el 88% de los mensajes que requieren moderación para el modelo 1, y al revisar un 65% de todos los mensajes, se encuentra un 61% en el modelo 2. Además, se destaca que un alto porcentaje de los mensajes que requieren más moderación son bien clasificados en este trabajo.

En conclusión, es posible encontrar y pronosticar mensajes que requieren mayor atención estudiando los comportamientos que poseen los usuarios respecto al sistema. Analizando las características de los resultados de manera exhaustiva produce una mejor comprensión del porqué ciertos comportamientos identificados generan o no mensajes relevantes al momento de moderar. Estudiando los comportamientos es posible generar estrategias preventivas y así minimizar la necesidad de moderación en la comunidad.

Se recomienda en trabajos futuros utilizar otros algoritmos de minería de datos, tales como reglas de asociación, buscando causalidad entre el comportamiento de usuario y la moderación o en el caso del clustering, utilizar medidas de similitud o disimilitud que incluyan características personales de usuario que tengan incidencia en la generación de mensajes problemáticos.

Índice

Índice de figuras

Índice de tablas

1.- Introducción

1.1.- Antecedentes generales

Las redes sociales virtuales permiten la interacción a través del mundo, mediante la posibilidad de conexión entre los individuos, aún pertenecientes a lugares geográficamente muy lejanos. Es así, como socialmente, las personas generan amistades, intercambian información, costumbres y muestran sus distintas necesidades [1].

A medida que Internet se hace más y más masiva, la posibilidad de generar este tipo de comunidades se hace cada vez mayor. Esto ha implicado que se generen distintos tipos de entidades sociales, tales como las redes sociales virtuales, las comunidades virtuales de interés y las comunidades virtuales de práctica, entre otras [4].

En particular, las comunidades virtuales de práctica tienen como objetivo el facilitar la interacción entre personas que quieren aprender y compartir sobre algún tópico en particular [8]. Lo relevante en estas comunidades es que se basan en el principio de que lo que se está compartiendo es algo pragmático, cuyos integrantes pueden experimentar y desarrollar a medida que el tiempo pasa, es decir, a través de prácticas comunes a los miembros de la comunidad. Estas comunidades se basan generalmente en herramientas que ayudan a la interacción virtual, tales como foros, Wikis y otras herramientas parecidas.

El fenómeno de las comunidades de práctica no se limita simplemente a la interacción entre personas aficionadas a un tema en particular, sino que existen comunidades de expertos e inclusive estas prácticas se realizan en grandes organizaciones teniendo como objetivo crear canales de información y conocimientos entre las distintas funciones administrativas, permitiendo el uso eficiente de los recursos [8].

En una comunidad virtual de práctica es muy importante generar, guardar y mantener el conocimiento generado. Su éxito se basa específicamente en aquellos individuos que son los que la mantienen viva, ya sea personas que preguntan o responden de acuerdo a lo que conocen [6,7].

Estas comunidades no funcionan de manera automática, sino que requieren de uno o varios administradores del sistema. En este sentido, existe una gran gama de administradores,

dependiendo directamente de la motivación que tenga respecto al trabajo que está realizando [6]. Existen administradores que están relacionados con el comienzo de la comunidad o fundadores, pero a medida que la comunidad alcanza dentro de su ciclo de existencia el estado de madurez, muchas veces es necesario incorporar más personas, lo cual se puede lograr ya sea contratándolos o por ofrecimientos voluntarios de miembros activos.

Las tareas de los administradores pueden ser muy variadas, y estar orientadas por distintas clases de motivaciones [3]. Un asunto de gran relevancia es la enorme cantidad de información con la que los administradores deben lidiar día a día para poder realizar su función. Lo anterior es, mantener la comunidad bien organizada, libre de mensajes que contengan información irrelevante, descubriendo tópicos interesantes para sus miembros, censurando mensajes maliciosos, entre otras acciones.

En este trabajo, la intención principal fue identificar aquellas tareas más importantes que realiza un administrador dentro de una comunidad, analizarlas y diseñar una aplicación de apoyo a la toma de decisiones. Fue necesario realizar un análisis exhaustivo de las diferentes funciones del administrador de la comunidad, de tal modo, que permita capturar aquellas tareas más relevantes para la buena evolución de la comunidad virtual. Posteriormente, evaluar la factibilidad de automatizar dichas tareas tanto como la efectividad de las técnicas propuestas sobre las funciones de administración de una comunidad virtual de práctica [5].

Un punto fundamental del trabajo, fue la aplicación de técnicas de minería de datos para encontrar patrones dentro de las actividades que realizan los administradores. Lo anterior, sirvió para utilizar su tiempo de manera más efectiva, sin tener que realizar inspecciones sobre decenas o centenas de mensajes irrelevantes, generando una mejor administración de la comunidad.

El diseño y construcción de esta herramienta, se basó en una técnica de punta en el área de text mining combinado con web usage mining. Esta técnica permite encontrar patrones intrínsecos de los datos, permitiendo ser analizados y contextualizados, para luego generar instancias que faciliten la interacción de los usuarios en las comunidades virtuales.

1.2.- Descripción del problema y justificación

Uno de los recursos generalmente escaso es el tiempo. El día de cada ser humano está limitado a aproximadamente dieciséis horas, y en este periodo de tiempo, las personas deben desarrollar todas sus actividades.

El administrador de una comunidad virtual de práctica debe estar atento casi diariamente sobre lo que sucede en el sistema. Esto requiere mucho tiempo, donde él (o ellos) debe(n) inspeccionar el sitio para estar en conocimiento de las cosas que han pasado, de los tópicos nuevos y de los posibles cambios que han ocurrido, entre otros. Además, existen comunidades de práctica cuyos integrantes suelen discutir más de lo normal por lo que es necesario revisar si ha pasado algún tipo de “pelea virtual” y actuar para detenerla [8].

Una comunidad virtual de práctica, se crea generalmente con un sólo administrador, puesto que estas tareas son fáciles de hacer al inicio. Pero, al momento de ir avanzando en el ciclo de vida de la comunidad, llega un punto de madurez donde, muy probablemente este trabajo no va a poder ser realizado por una única persona. En este momento, el criterio y las políticas de administración comienzan a mezclarse, y comienzan a depender del administrador que ejecutó ciertas tareas [1].

En varios casos, existen comunidades de práctica que no son creadas bajo el alero de una empresa u organización, sino que son creadas espontáneamente por uno o más miembros fundadores, los cuales toman el rol de administradores. Además, esta labor, por lo general, se realiza en forma de voluntariado (no hay pago de por medio), por lo que muchos administradores usan el tiempo libre (fuera de sus empleos, u otras actividades) para ejercer este rol [9]. Es por esto, que necesitan contar con herramientas que les permitan realizar la administración de la comunidad en forma rápida, sin perder demasiado tiempo en análisis, de todos los mensajes de la comunidad.

Mediante el análisis, diseño e implementación de la aplicación propuesta, se pretendió capturar la información necesaria para el manejo de los administradores, creando así una herramienta que ayude con el apoyo a la toma de decisiones, optimizando así el tiempo utilizado, y aumentando además la eficacia de las acciones tomadas.

El presente proyecto es de gran interés académico, ya que mezcla la utilización de algoritmos de minería de datos con nuevas herramientas sociales que se han generado a razón

de facilitar las interacciones en la Web [2]. Todo se basa en un ambiente altamente colaborativo, donde las personas crean y manejan el conocimiento con claras intenciones de ir depurándolo a medida que pasa el tiempo. Es un tema que está a la palestra de la investigación actual.

Este trabajo se basó principalmente en el diseño y la evaluación de esta aplicación. Se propone la construcción de un sistema, tanto como su adaptación con algoritmos modernos de minería de datos. Finalmente, los resultados obtenidos debieron ser evaluados, de manera de ver qué tan fidedignos son con respecto a la percepción de los mismos administradores.

1.3.- Objetivos

1.3.1.- Objetivo General

Diseñar e Implementar una aplicación que permita proveer de información para apoyar la moderación y administración de comunidades virtuales de práctica, utilizando técnicas de minería de datos, reduciendo la carga de trabajo en esta tarea.

1.3.2.- Objetivos Específicos

1. Estudiar la comunidad virtual de práctica y sus distintas características: historia, estructura, roles, objetivos y herramientas utilizadas para su administración.
2. Identificar un conjunto de tareas realizadas por los administradores y usuarios respecto a la moderación.
3. Diseñar técnicas de minería de datos en busca de patrones de comportamientos de usuario que se ajusten al perfil de tareas que requieran moderación.
4. Analizar tanto el web log de usuario como el texto del foro de la comunidad diseñando e implementando las técnicas propuestas para la comunidad ubicada en www.plexilandia.cl/foro/
5. Diseñar y construir una aplicación de análisis que apoye a la moderación, utilizando el modelo anterior sobre el foro de www.plexilandia.cl.
6. Analizar y evaluar con respecto a la relevancia de la información inferida a la moderación.

1.4.- Metodología

Para poder llevar a cabo este proyecto se utilizó la metodología descrita a continuación:

Lo primero que se realiza, es entender lo mejor posible el problema atacado, vale decir, cómo funcionan generalmente las comunidades virtuales de práctica, cuáles son sus objetivos principales y de qué manera se logra que tengan éxito. En esta etapa, se recopila toda la información necesaria, tanto bibliográfica como de experiencias prácticas en el tema.

Consecutivamente, todo pasa por entender la manera en que los administradores ven la comunidad, qué tipos de intereses tienen, y cuáles son las tareas más comunes que realizan. Esta inspección, se realiza mediante pequeñas entrevistas con administradores, y permite crear las bases del trabajo ulterior. Algunas de las características más importantes descubiertas fueron:

- Importancia de la moderación para el propósito de la comunidad.
- Razones de la existencia de esta actividad.
- Tiempo que requiere llevar a cabo esta actividad.
- Estrategia o reglas preventivas respecto a esta actividad.
- Acciones a tomar en el caso que esta actividad sea muy común dentro de la comunidad.

En este punto, se deben comprender las técnicas asociadas al trabajo, vale decir, la metodología de minería de datos. Para esto, se requiere estudiar estas técnicas y comprenderlas a cabalidad, tratando de utilizarlas de la manera más eficiente posible, aplicando los métodos explicados en publicaciones de investigación asociadas al tema. En consecuencia, se intentan adaptar estas técnicas a la resolución de este problema en particular. Luego, se procesa toda la información, generaran las bases de datos y fijan todos los parámetros para poder ejecutar estas técnicas de acuerdo a lo planeado.

Las técnicas antes mencionadas, permiten generar un análisis completo respecto a las acciones de moderación en una comunidad virtual de práctica. Esto se basa en la hipótesis que el comportamiento de usuario y la data existente en la interacción de los usuarios con el sistema, permiten sustentar las nuevas acciones a tomar por parte de los administradores. Lo anterior, requiere los siguientes procesos:

- Generar el proceso de sesionización de los Web log de la comunidad virtual existente en el foro de www.plexilandia.cl .
- Pre-procesar la data de la comunidad virtual seleccionada para el uso de los algoritmos.
- Aplicar algoritmos de clusterización de manera de encontrar patrones en los datos respecto a la moderación generando un modelo de comportamientos de usuario dentro de la comunidad.
- Diseñar la forma de mostrar el análisis, vale decir, métricas y esquemas.

Seguidamente se desarrolla una aplicación que utilice el modelo generado a partir del análisis pueda facilitar al usuario la clasificación de tópicos de manera correcta.

Al finalizar, se intenta generar un tipo de evaluación de esta aplicación, para lo cual se intentará ver que la herramienta generada sirva para hacer la gestión administrativa de manera eficaz. El resultado es contrastado con la experiencia del experto de la comunidad virtual estudiada, teniendo así una evaluación del rendimiento expuesto por el trabajo realizado.

Metodología Web Mining

CRISP-DM[1] es una metodología muy utilizada para la minería de datos, la cual ve la extracción de ellos como un proceso completo, pasando desde la recopilación de datos, procesamiento, análisis y evaluación [17]. Su enfoque, principalmente se basa en los negocios, sin embargo, se puede utilizar en la investigación. Tanto sus etapas como interrelaciones se aprecian en la Figura 1.

El ciclo se describe partiendo del "entendimiento del negocio", que corresponde en entender el problema y buscar la información necesaria para aplicar una solución, pasando al entendimiento de los datos que serán analizados. Luego de estudiar esos datos, corresponde pasar a una etapa de procesamiento de estos y seguir por una generalización o modelamiento en conjunto con la experimentación necesaria para su posterior análisis y evaluación. Con este ciclo se alimenta con nueva información el entendimiento del negocio y también se producen las implementaciones necesarias.

[1] CRISP-DM: Cross-Industry Standard Process for Data Mining

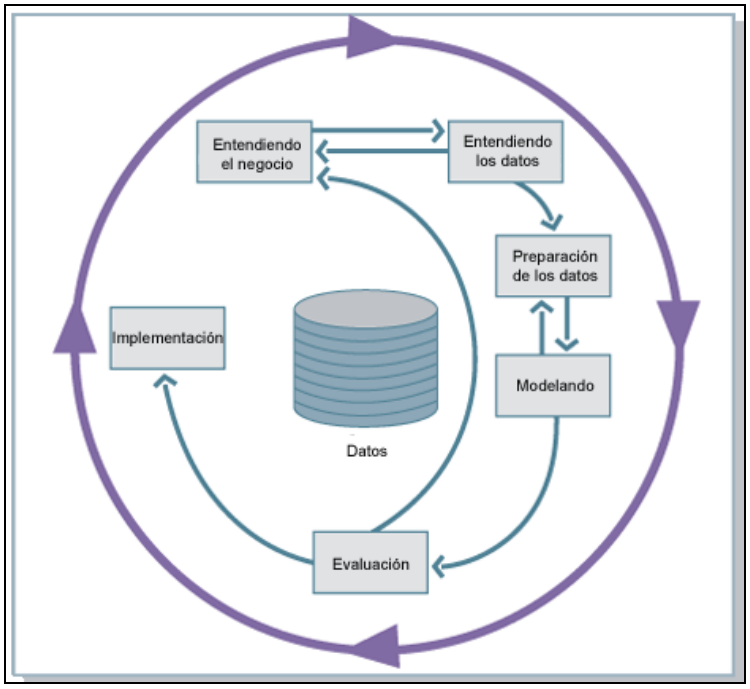

Figura 1: Metodología CRISP-DM
Fuente: http://www.crisp-dm.org/Process/index.htm

1.5.- Resultados Esperados

Como resultado de este proyecto se intenta obtener una implementación que permita dar soporte eficaz y eficiente al momento de llevar acabo las actividades de administración dentro de una comunidad. Lo anterior se evalúa contrastando los resultados, con la importancia del resultado, según los usuarios administradores. Se intenta capturar patrones de comportamiento de los administradores que sean lo más significativos posibles, de manera que los resultados puedan ser un apoyo sustancial al momento de ejercer sus respectivas actividades.

En términos específicos, los resultados esperados son:

1. Generar un marco conceptual que contenga los tópicos más importantes respecto a lo que son las comunidades virtuales y quiénes las integran. Además debe contener el estado del arte de las herramientas actualmente utilizadas para el análisis de comunidades y las herramientas propuestas en este trabajo.
2. Generar capítulo de requerimientos, diseño y clasificación de actividades de moderación en una comunidad virtual de práctica.

3. Identificar, mediante una implementación, comportamientos de la navegación de usuarios de la comunidad virtual del foro de www.plexilandia.cl. En esta instancia se obtiene un modelo de comportamientos navegación incluyendo el tópico involucrado.
4. Se obtiene un análisis que puede clasificar comportamientos de usuario y asociarlos a la moderación.
5. Se evalúa y valida el análisis mediante información aportada por el administrador.

1.6.- Alcances

Los objetivos de este trabajo establecen el diseño e implementación de una herramienta tecnológica que permita analizar y apoyar la administración de comunidades virtuales de práctica, utilizando un conjunto de patrones de comportamientos consensuados como importantes dentro de esta actividad. La idea es determinar cuales son las actividades más comunes por parte de un administrador, e intentar, mediante la información obtenida por parte de este individuo y la existente en la del sistema estudiado, apoyar la toma de decisiones al momento de administrar la comunidad para realizarla de manera más eficiente. Lo anterior será construido para y evaluado para una comunidad, dejando el software con las implementaciones necesarias para ser extendido en trabajo futuro.

2.- Marco Conceptual

2.1.- Comunidades Virtuales

2.1.1.- Comunidades Virtuales de Práctica

La revolución que ha generado en las interacciones humanas la Internet, ha permitido la proliferación de las comunidades virtuales. Muchas de las actividades que se realizaban presencialmente, hoy se realizan de manera online en comunidades virtuales [8].

De acuerdo a [1], las comunidades virtuales de prácticas son "*grupos de personas que comparten un interés, un conjunto de problemas, la pasión sobre un asunto, y quienes profundizan su conocimiento y experticia en esta área interactuando de manera continua*".

Las comunidades virtuales, han sido grandemente estudiadas por varios años, mas la definición de estas entidades sociales no ha sido finalmente consensuada. Dentro de las características más importantes que se identifican se encuentran [10]:

- Los miembros poseen un objetivo, interés, necesidad o actividad que provee la razón primaria para pertenecer a la comunidad.
- Los miembros mantienen una participación activa y continua además de generar uniones emocionales y otras actividades entre los distintos integrantes.
- Los miembros poseen acceso a los recursos compartidos y existen políticas de acceso.
- Reciprocidad de información, el apoyo y el servicio son importantes.
- Existe un contexto compartido respecto a las convenciones sociales, lenguajes y protocolos.

La descripción de las comunidades virtuales se basa en dos grandes ramas teóricas, la descripción social y la descripción con orientación tecnológica. En la primera, se interpreta bajo los cánones del mundo sociológico, en el cual las interacciones existentes dentro de estas comunidades corresponden a vínculos del tipo débiles entre distintos individuos. En la segunda, con orientación tecnológica se suele describir de acuerdo al software que lo apoya, tales como los chat, boletines, UseNet, comunidades basadas en tecnologías Web, entre otros.

Las comunidades virtuales de prácticas tienen estrecha relación con el aprendizaje y la creación de conocimiento, considerando que el aprender interviene en la identidad del ser humano. Para Etienne Wenger [1], las características estructurales de una comunidad virtual de práctica son las siguientes:

- Dominio: un dominio de conocimiento genera un lugar común, inspira a los individuos a participar, guía el aprendizaje y le da sentido a sus acciones.
- Comunidad: la noción de comunidad mantiene los cimientos del aprendizaje. Permite fomentar las interacciones y mejora la voluntad a compartir ideas.
- Práctica: mientras el dominio mantiene la idea general de interés de la comunidad, la práctica es el foco específico de lo que realmente le agrega valor a los integrantes de la comunidad.

2.1.2.- Moderador dentro de una Comunidad virtual

El moderador dentro de una comunidad virtual es quien realiza las actividades necesarias para mantener el objetivo principal de la comunidad. Es por la razón anterior que el moderador o facilitador se hace responsable de mantener el espacio para que la interacción de los individuos permita la creación de conocimiento y la promoción de las sinergias que generan los cambios en la realidad posteriormente [12].

El rol del moderador depende directamente de las normas culturales de la comunidad [10]. Dependiendo del origen de las comunidades y del contexto al cual pertenezcan, determinará el tipo de gobierno que se deberá mantener.

Dentro de las distintas tareas que son realizadas por los moderadores se encuentran:

- Facilitar que el grupo se mantenga focalizado.
- Administrar la lista de suscritos.
- Filtrar mensajes, decidiendo cuales deben ser posteados[2]. Lo anterior, significa marginar el spam, post difamatorios, manteniendo alta la proporción de mensajes relevantes.

[2] Postear = Publicar

- Ser un experto, respondiendo preguntas frecuentes, redirigiendo a la gente hacia los distintos tópicos ya existentes.
- Editar o dar formato a los mensajes.
- Promover preguntas que generen discusión.
- Ayudar en distintos tipos de necesidades a los integrantes de la comunidad.
- Ser el apaciguador de posibles “peleas” dentro de la comunidad.

El nivel de actividad de un moderador varía dependiendo de la comunidad. Depende de la velocidad de los acontecimientos como se utiliza el tiempo en la lectura, haciendo juicios y tomando las acciones en cada post o actualización de la comunidad. Es así, como los moderadores de manera autodidacta o imitando a otros como ellos, terminan aprendiendo a mantener las políticas de la comunidad.

Muchas de las reglas o políticas existentes en las comunidades son propuestas en su creación, permitiendo que queden claras para los integrantes de la comunidad y que no se considere el actuar del moderador como una conducta arbitraria. La razón de la existencia de las políticas dentro de la comunidad es para mantenerla alineada con su objetivo principal de existencia. Suele ocurrir que una excesiva cantidad de normas, entorpece el movimiento del grupo, al promover comportamientos contrarios a los objetivos, como incitar a los integrantes a transgredirlas con el afán de molestar [10].

2.1.3.- Herramientas para administración de Comunidades

La moderación de comunidades virtuales, se basa principalmente en entidades humanas, las que son responsables de ejecutar las reglas preestablecidas en la comunidad, dejando muchas veces vacíos que debe resolver el moderador directamente. La mayoría de las reglas se refieren a situaciones operativas, vale decir, se circunscriben a aspectos fáciles de identificar, como el caso del uso de mayúsculas, que en muchos contextos significa gritar, expresarse de manera efusiva o la utilización de palabras groseras[3].

El desarrollo de herramientas automáticas o de apoyo que pretenden facilitar las tareas de moderación para la buena convivencia en situaciones online, comienzan al nivel de bloqueo de palabras que expertos; o bien, aquellos que administran el medio consideran inapropiadas, por lo que al momento de su uso por parte de los usuarios, estas palabras no aparecen, o se le

[3] Fuente: Administrador Foro Plexilandia.cl

advierte automáticamente al usuario de que comete infracción utilizando lenguaje inapropiado. En este contexto, aparecen una gran cantidad de aplicaciones, inclusive algunas que vienen como funcionalidad dentro de comunidades y redes sociales, que lo realizan utilizando un diccionario preestablecido o con mecanismos de aprendizaje automatizados.

Algunos mecanismos asociados a la moderación, son los vinculados al reconocimiento de spam, o mensajes no deseados, ampliamente utilizados en las aplicaciones de correos electrónicos, que intentan identificar y clasificar los mensajes que llegan a una casilla de un usuario. El grado de sofisticación de esta herramienta se basa en los métodos y heurísticas utilizadas. Generalmente se requiere el entrenamiento de los algoritmos para poder clasificar el correo como spam o no. Normalmente, durante el entrenamiento se utiliza el criterio del usuario.

Los trabajos asociados a la identificación de spam dentro de redes sociales y comunidades virtuales se basan en técnicas de clasificación mediante etiquetas que los mismos individuos colocan sobre los mensajes [25]; o bien, por clasificación automática [26].

Medios masivos como los juegos de video del tipo MMORPG (masively multiplayer online role-playing game), han creado la necesidad de producir herramientas tecnológicas que permitan administrar lo que los distintos usuarios realizan en la aplicación. Estos ambientes son propicios a la catarsis de los individuos, que pueden realizar un sinnúmero de malas prácticas, tales como: engaño, insultos, intimidación, acoso, entre otras. Una herramienta que intenta facilitar la moderación de estas conductas se llama Net Moderator[4].

2.2.- Comportamiento de usuarios en la Web

El comportamiento de los usuarios en la Web, es un tema ampliamente estudiado en los últimos años, ya que de él se pueden extraer distintas características que permiten una mejor implementación del diseño de las distintas aplicaciones, que facilitan al usuario la búsqueda de contenido y el posible análisis de la estructura de links establecida [21]. Para los diseñadores de aplicaciones, es necesario entender la forma más correcta en que deben mostrarse los contenidos, permitiéndoles a los usuarios la tener acceso a la información de la manera que necesitan. Los estudios de comportamiento se encuentran asociados a lo que sucede en el comercio Web, comportamiento de compra, técnicas de publicidad y reclamos [41].

[4] http://www.crispthinking.com/

Específicamente, el conocimiento relativo al comportamiento de los usuarios Web se encuentra en los Web logs. En ellos se encuentra información de los registros de los actuares de los individuos, constituyendo información anónima de la información de las necesidades, de la estructura de los contenidos, la información que consumen y la situación del sitio en general, entre otros [28].

Una forma emergente de análisis de esta información se basa en las técnicas de Data Mining, principalmente estudiando las sesiones de usuarios, utilizando técnicas tales como, clustering, reglas de asociación, clasificación, entre otras.

Conductas de navegación

Para poder entender como se generan las navegaciones de usuario, es necesario mostrar cuales son las posibilidades que existen. Se deben tener en cuenta dos tipos de comportamientos mayoritarios [29]:

- Browsing: el usuario sigue la estructura interna del sitio o aplicación Web visitada
- Searching: se basa principalmente en navegación a partir de búsquedas en motores de búsquedas como Google, entrando a cualquier página del sitio o aplicación Web, sin necesidad de ser la principal.

Ambas clasificaciones de navegación permiten, al momento del diseño de la aplicación, entender si deben ser orientadas a la navegación interna o a la posibilidad de satisfacer al usuario a seguir interactuando a partir de una búsqueda por parte de un buscador externo.

Existen mecanismos estadísticos que permiten conocer las conductas de navegación de los sitios en Internet. Se pueden utilizar los Web logs u otras herramientas que basan su análisis principalmente en métricas estadísticas tales como el número de usuarios de un sitio, las visitas de usuarios por día, el uso diario de ciertas páginas y búsquedas. Estos informes suelen ser gratuitos, y se destaca entre ellos Google Analytics[5].

[5] http://www.google.com/intl/es_ALL/analytics/

2.3.- Procesamiento de datos y proceso KDD

Debido a que los algoritmos utilizados en análisis de grandes volúmenes de datos están diseñados para funcionar con modelos numéricos, el proceso KDD permite resolver el problema que genera utilizar datos no estandarizados. Este proceso se resume en la Figura 2.

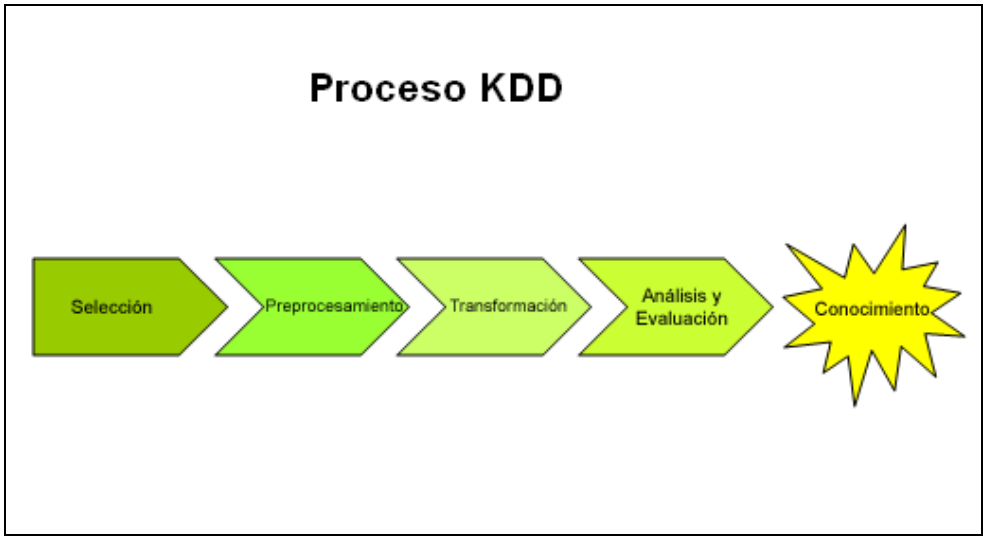

Figura 2: Proceso de Knowledge Discovery in Databases
Fuente: elaboración propia

2.3.1.- Datos de aplicaciones Web

Los datos que vienen de la Web suelen ser muy diversos. Pueden ser estructurados, como el caso de las bases de datos, semi-estructurados como las páginas Web en el lenguaje HTML (del inglés Hiper Text Markup Language) o sin estructura como texto plano, imágenes, entre otros.

Al momento de ingresar a un documento HTML, se generan peticiones hacia el servidor que provee los distintos recursos que deben ser desplegados por el browser. En ese momento estas acciones se graban en el Web log [22], tanto como para accesos a páginas como objetos multimedia. Con lo anteriormente descrito, se denota el hecho que al clic de un usuario se generan múltiples registros que probablemente contienen información irrelevante.
Se puede ver un ejemplo simple de un Web Log, en la Tabla 1. Este se define generalmente por:

- **Dirección IP:** la dirección IP del usuario con que accede a la Web.
- **Identity:** información de identificación proporcionada por el cliente.

- **Tiempo:** el tiempo en el cual el servidor responde a la petición.
- **Tipo de requerimiento, protocolo y archivo pedido:** identifica la petición hecha por el cliente y la versión del protocolo de transferencia.
- **Estado:** Código representado por un número entero, dónde se señala el estado de una petición. Algunos de los más frecuentes son 200 (exitoso) 404, 403, etc.
- **Bytes:** número de bytes entregados en la petición.
- **Referrer:** línea enviada por el cliente que indica la fuente original donde se genera la petición.
- **User Agent**: versión del navegador o aplicación usados por la aplicación cliente.

Tabla 1: Ejemplo de web log

Id	IP Address	User ID	Time	Method/URL/Protocol	Status	Size	Referrer	Agent
1	123.456.78.9		[25/Apr/2010:03:54:41]	"GET A.html HTTP/1.0"	200	3290		Mozilla/4.0
2	123.456.78.9		[25/Apr/2010:03:54:46]	"GET B.html HTTP/1.0"	200	1000	A.html	Mozilla/4.0
3	123.456.78.9		[25/Apr/2010:03:54:48]	"GET P.html HTTP/1.0"	200	2600		Mozilla/4.0
4	123.456.78.9		[25/Apr/2010:03:54:51]	"GET L.html HTTP/1.0"	200	2500		Mozilla/4.0
5	123.456.78.9		[25/Apr/2010:03:54:53]	"GET A.html HTTP/1.0"	200	3290		Mozilla/4.0
6	123.456.78.9		[25/Apr/2010:03:54:55]	GET C.html HTTP/1.0	200	1500	A.html	Mozilla/4.0
7	123.456.78.9		[25/Apr/2010:03:55:00]	"GET D.html HTTP/1.0"	200	1450		Mozilla/4.0
8	209,458.782		[25/Apr/2010:03:55:10]	"GET A.html HTTP/1.0"	200	3290		Mozilla/4.0
9	209,458.784		[25/Apr/2010:03:55:30]	"GET A.html HTTP/1.0"	200	3290		Mozilla/4.0
10	209,458.782		[25/Apr/2010:03:57:10]	"GET A.html HTTP/1.0"	200	3290		Mozilla/4.0

Los Web logs poseen la información ordenada en tiempo de petición y generalmente no poseen información privada del usuario en particular.

Para poder analizar el comportamiento de un usuario, es necesario reconstruir su sesión real, vale decir, el conjunto de actividades realizadas por el usuario desde el momento que llego al sistema hasta que se fue en un determinado tiempo. Este proceso suele ser bastante complejo por el ruido existente en los datos tales como:

- **Proxy y firewalls**: si una institución utiliza Proxy o firewall, la IP real es enmascarada usando una única IP real externa. En ese caso, los Web logs contendrán muchos registros originados con la misma IP sin necesidad de pertenecer a un comportamiento de usuario único.

- **Asincronismo de la Web**: muchas veces no existe forma de identificar a los usuarios, en estos casos muchas veces se utilizan cookies o métodos de reconstrucción de sesiones.

- **Web crawlers**: los crawlers son robots utilizados por motores de búsqueda que realizan la acción de colectar la información de la Web para alimentar sus buscadores, tales como el caso de Google o Yahoo!. Estas peticiones al servidor no son necesarias, por lo cual es necesario identificarlas al momento de generar las sesiones de usuario.

2.3.2.- Pre-Procesamiento de los datos

Los datos que vienen de la Web vienen con una gran cantidad de elementos que no son útiles para el análisis, o al menos, no vienen en un formato que favorezca el uso de las metodologías y técnicas de la minería de datos.

Particularmente en el texto, existe una gran cantidad de problemas, tales como las palabras que no tienen importancia llamadas Stopwords. Estas palabras se encuentran generalmente en todos los documentos y no tienen gran importancia semántica, sino que sirven para armar oraciones, tales como ilativos, pronombres, artículos, entre otros. Estas palabras deben ser eliminadas.

Otra técnica ampliamente utilizada es el stemming que tiene la intención de encontrar la raíz de una palabra. Así múltiples conjugaciones verbales con género o número distinta, pueden ser agrupadas con dicha raíz.

2.3.3.- Proceso de Sesionización

Teniendo los web logs se hace necesario estudiar las rutas tomadas por cada usuario individualizando sus sesiones. Este proceso se llama proceso de sesionización [21], cuyo propósito es encontrar las sesiones reales de un individuo, para lo que se han propuesto heurísticas de manera de encontrar sesiones únicas para un tiempo determinado. En ese contexto existen del tipo proactiva o reactiva [23]:

Estrategia Proactiva

En el caso de las estrategias proactivas, la intención es colocar objetos que permitan identificar a un usuario. Por ejemplo, el uso de cookies que son enviadas hacia los clientes al

momento de ingresar al sitio por primera vez. Es así, que desde la segunda visita es posible identificar un usuario de manera fidedigna. No obstante, la problemática que tiene esta estrategia se basa en la privacidad y que además pueden ser eliminadas por los usuarios.

La utilización de cookies es la manera más fidedigna para asegurar la reconstrucción de sesiones de usuarios, pero poseen la dificultad de su configuración y la utilización de manera errónea por parte de entes inescrupulosos. Sin embargo, permiten una mejora en la experiencia de navegación en la Web y un acercamiento a lo que posiblemente requieren los usuarios. Los tipos de cookies son [24]:

a) Persistentes o de sesiones: Una cookie del tipo persistente se queda grabada en un computador, pudiendo identificar a usuarios que ingresan a un sitio de manera frecuente. En cambio, una de sesión, sólo persiste hasta el momento que el usuario abandona el sitio.

b) Primera o tercera parte: Una cookie de primera parte es ofrecida por el sitio que las utiliza, mientras que una de tercera parte se ofrece de otro sitio por entidades que ofrecen este servicio, aunque pueden ser bloqueadas.

c) No personales o personales: Las no personales sólo rastrean visitantes y usuarios sin buscar información específica, en cambio las personales identifican usuarios en particular, con todos sus datos.

Estrategias reactivas

Sólo utilizan la información de los web logs y son heurísticas orientadas al tiempo o a la navegación. El primer caso, son heurísticas que intentan reconstruir sesiones con la premisa que tienen una duración máxima (típicamente 30 minutos [21]) o al tiempo entre distintas páginas accedidas es menor a una diferencia de tiempo. En el segundo caso, se basa en el conocimiento de la estructura del sitio Web, admitiendo transacciones dentro de una misma sesión si es posible acceder a una página a partir de otra, en el caso de no ser posible, se identifica como una nueva sesión.

Al momento de sesionizar, se requiere hacer un pre-procesamiento de la información, en los siguientes pasos:

1. Limpiar datos: esto es eliminar los registros que correspondan a objetos multimedia que pertenezcan a una página, puesto que estos no son pedidos explícitamente por el usuario.
2. Limpiar registros no humanos: eliminar registros que han sido accedidos por robots crawlers u otros agentes robóticos.
3. Identificar usuarios distintos: se realizan identificaciones con IP-Agente y cookies
4. Identificación de sesiones de usuario: se identifican las páginas accedidas en un tiempo y se especula sobre el tiempo que toma el usuario en cada página y cuando decidió dejar el sitio
5. Completar rutas: esto se realiza dado que la reconstrucción de sesiones se ve perjudicada por abrir nuevas ventanas en un navegador o el uso del botón back.

Para la identificación de usuarios, se suele utilizar IP y el Agente, puesto que muchas veces existen ISP que proveen la misma dirección IP a un conjunto de usuarios, por lo que usar ambos campos, en estos casos suele ser más efectivo.

En general existen condiciones mínimas para considerar una sesión como real [26]. Sea *L* un conjunto de registros en el web log y $R = \{r_1,...,r_n\}$ el conjunto de sesiones iniciales encontradas en *L* luego de la sesionización.

Las condiciones mínimas para que r_i sea una sesión son:

1. Estar compuesta por objetos pedidos durante la sesión ordenados por tiempo. Entonces, $\forall r_i, \forall k \in \{2,...,n\}, r_{i,k}.timestamp > r_{i,k-1}.timestamp$.
2. Solamente objetos pedidos en *L* pueden estar en *R*, es decir, $\bigcup_{r_i \in R} \left(\bigcup_{j=1}^{n} r_{i,j} \right) = L$.
3. Cada pedido en *L* pertenece exactamente a una sesión de *R*, es decir, $\forall r_i \in R, \forall j \in \{1,...,n\}, \nexists i' \neq i, j' / r_{i,j} = r_{i',j'}$.

Utilizando lo anterior, se reducen los errores dentro del proceso. Sin embargo, el proceso no es perfecto y siempre se deberán revisar y generar estudios estadísticos en busca de errores.

2.3.4.- Transformación de datos

La transformación de los datos se debe principalmente que la mayoría de las herramientas de análisis utiliza modelos matemáticos con entrada numérica. Muchos de la información estudiada es texto o son representaciones multimedia como las fotografías, videos, audio, entre otros. Eso genera la necesidad de crear un modelo que permita transformar los elementos reales en representaciones matemáticas consistentes con las características que se desean considerar dentro del análisis.

Modelo de transformación para Sesiones de usuarios

Con la intención de registrar la secuencia de navegación de un usuario, y datos sobre las preferencias de navegación, se considera el vector de comportamiento de usuario (UBV) como [42]:

Ecuación 1

$$V_i = [(p_{i,1}, t_{i,1}), \ldots, (p_{i,n}, t_{i,n})]$$

donde p_{ij} es la *j*-ésima página de la sesión *i*, y t_{ij} es el tiempo utilizado en su visita.

Este vector facilita la manipulación de la información de la sesión, y permite entender de manera más simple del comportamiento de usuario. Esta representación termina siendo la base del modelo que, si se necesita, podría incluir nuevas características.

Modelo de transformación para Text mining

En el caso del contenido, suele utilizarse el modelo vector space model (en español, modelo de espacio vectorial), que es en síntesis la representación vectorial del texto asociado a pesos relativos en el corpus del documento.

Luego de realizado el pre-procesamiento del texto, que se basa en la eliminación de stopwords y realizar el proceso de stemming, se procede a generar los vectores característicos.

Sea *R* el número de términos distintos extraídos del conjunto de documentos y *Q* el número de documentos. La representación vectorial suele ser una matriz *M* de dimensión *RxQ* donde cada componente m_{ij} representa el peso de la palabra *i* en el documento *j*. El modelo

más utilizado dentro del contexto de la recuperación de la información es modelar m_{ij} como el TFIDF descrito en la Ecuación 2.

Ecuación 2: TF-IDF

$$m_{ij} = f_{ij} \cdot \log(Q/n_i)$$

donde f_{ij} es la frecuencia de la palabra *i* en el documento *j*, *Q* es el número total de documentos y n_i es el número de documentos donde se encuentra la palabra *i*.

2.4.- Web Mining

Dentro de los esfuerzos más destacados realizados en el contexto de la Web, se encuentran los diversos estudios que han intentando capturar la información relevante para los usuarios de los sistemas con la intención de proveerles todas las herramientas necesarias, tanto para que se mantengan en contacto con el sitio, como internar el comportamiento del usuario, para generar una adaptación propicia a lo que los interesados requieren para su mayor comodidad. Nace de este modo el Web mining, minería de la Web, cuyo objetivo es poder proveer herramientas de análisis de la información sobre la experiencia de usuario [16].

La minería de datos se define como el proceso de extracción de patrones a partir de los datos. Tales metodologías sumadas a los datos de la Web más ciertos modelos, corresponden al Web mining, que se basa fundamentalmente en el descubrimientos de patrones en la estructura, contenido y uso de sitios Web [17].

2.4.1.- Web mining y componentes

El proceso del Web mining puede verse como un proceso de al menos tres etapas fundamentales donde la entrada son los datos Web [15], las que se detallan a continuación.

1. Pre-procesamiento: En esta etapa sucede todo lo anterior a la aplicación de técnicas y algoritmos de la minería de datos. Esto requiere la limpieza de los datos, que generalmente presenta errores o se requiere cumplir con un formato en particular para poder ejecutar los algoritmos. Los errores, provienen de la forma en que están representados los datos, por ejemplo, si los datos vienen en palabras, existen errores ortográficos y tipográficos que afectan al análisis de patrones de manera lexicográfica

por lo que es necesario eliminar palabras que no poseen relevancia y tratar de acotar el conjunto final. En esta etapa también se define el modelo de datos a utilizar el cual permitirá la aplicación de los algoritmos y el posterior análisis de los resultados.

2. Minería de Datos: Dentro de las tareas más comunes se encuentran:
    - Clustering: permite el descubrimiento de grupos o estructuras de individuos similares sin conocer la estructura real de los datos.
    - Clasificación: permite generalizar una estructura conocida para utilizarla sobre nuevos datos.
    - Regresión: modelar los datos con una función que se ajuste con el menor error.
    - Reglas de Asociación: infiere reglas de asociación lógicas entre las distintas variables del sistema.

3. Validación de Resultados: En esta etapa, se requiere hacer un análisis respecto a los resultados obtenidos a partir de los modelos y aplicación de la minería de datos. Se debe analizar si existe un sobre ajuste de los datos o si el modelo obtenido representa genéricamente, es decir, independientemente de los datos, los patrones encontrados.

El Web mining utiliza el contenido de las páginas, su estructura y otros estimadores con la intención de ayudar al usuario a encontrar la información que requiere [16]. De esta manera se definen tres aspectos dentro de los distintos datos, los cuales corresponden al uso (datos de preferencia en la navegación de usuario) [18], contenido (información en los documentos, metadatos, información multimedia) y estructura de hyperlinks.

Dependiendo del tipo de dato estudiado, nacen las tres principales áreas de clasificación del Web mining que se describen a continuación. Además, una explicación gráfica de la Taxonomía de Web Mining en la Figura 3.

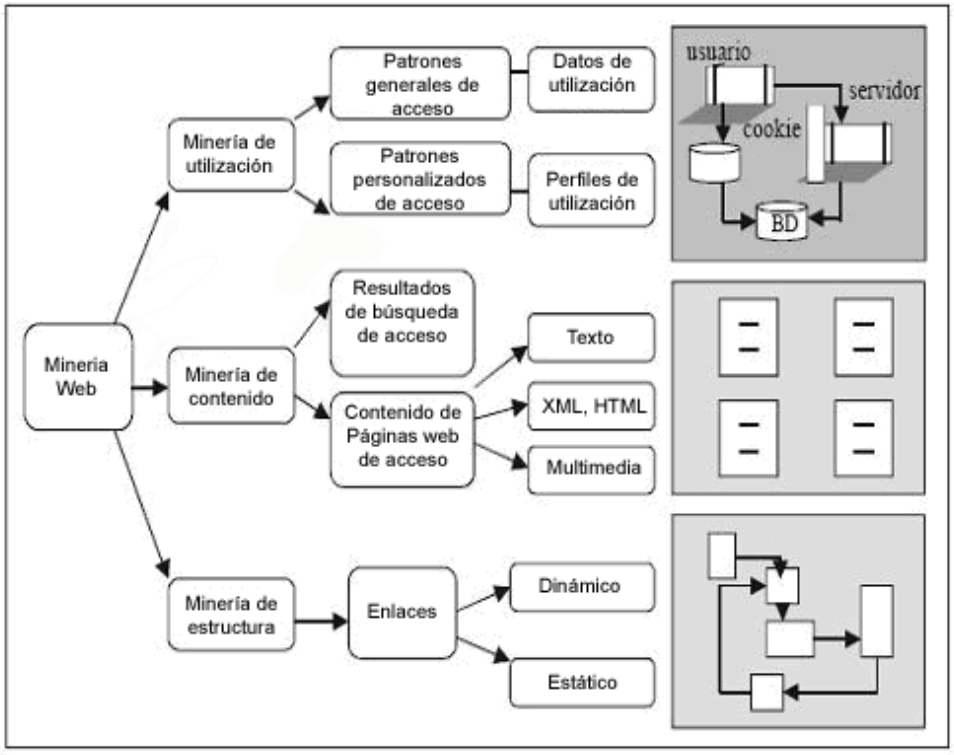

Figura 3: Taxonomía de Web mining
Fuente: http://desarrolloparaweb.blogspot.com/2010/04/tecnicas-para-conocer-nuestra-audiencia.html (oct 2010)

a) Web usage mining (WUM): Proceso de obtención y análisis del comportamiento de usuario obtenido de elementos de navegación, tales como, los web log del servidor, entre otros. Permiten la personalización y en mecanismos de recomendación en la Web [44].

b) Web content mining (WCM): Proceso de análisis sobre el contenido de la Web, con la intención de indagar en cuales son las palabras, oraciones, más significativas para los usuarios dentro del sistema [45].

c) Web structure mining (WSM): Proceso de extracción y análisis de patrones respecto de la estructura de links existente dentro del sistema. El estudio se encuentra focalizado en las conexiones existentes entre las páginas y su posible popularidad. Generalmente, se estudia considerando el sitio como un grafo dirigido [46].

Para este trabajo se describen las técnicas WUM y WCM con mayor profundidad en las siguientes secciones.

2.4.2.- Web usage mining

El análisis utilizando Web usage mining es una herramienta que ha ido tomando fuerza desde ya varios años, puesto que se aplica directamente a la personalización de la Web y al incremento en la complejidad de los sitios Web [12]. Trabaja en particular con la minería sobre datos de uso, sin embargo, se deben realizar algunas modificaciones para se utilizadas sobre distintos tipos de datos relacionados con el uso.

La información que proviene de las distintas aplicaciones de la Web, proviene de distintos niveles. Estos pueden ser según [13]:

1. Servidor: generalmente tienen que ver con los Logs o bitácoras de navegación de los visitantes. Estas bitácoras suelen ser muy importantes para el análisis del comportamiento de los usuarios de la aplicación, pues describen fidedignamente el camino recorrido y las posibles acciones tomadas por los usuarios.
2. Cliente: esta información se obtiene al hacer modificaciones en los navegadores o creaciones de aplicaciones que corren por el lado del usuario. Para poder utilizar este tipo de generación de información se requiere que el usuario explícitamente apruebe su uso.
3. Proxy o aplicación intermedia: mediante un modelo de predicción del comportamiento de los usuarios, una aplicación puede hacer más eficiente la navegación de los usuarios.

Los Web logs son un medio de extracción de información en la Web, contienen información de navegación de los distintos usuarios. Uno de sus usos más importantes es poder reconstruir, por varios métodos de inferencia, la navegación completa de un usuario en un tiempo determinado, partiendo de su dirección IP y otras características. Lo anterior, se realiza mediante métodos tales como la sesionización, que consta en reconstruir la navegación y el tiempo utilizado por el usuario en cada página o acción de la aplicación Web [13].

Como en todas las técnicas derivadas de la minería de datos existen al menos tres subprocesos en los que se divide su utilización. Pre-procesamiento, descubrimiento de patrones y análisis de patrones [14].

2.4.3.- Web content mining

El objetivo del Web content mining es encontrar contenido importante dentro de los documentos Web. En síntesis, es muy similar a las técnicas de Recuperación de la Información. No obstante, no sólo se limita al texto sino a otros objetos multimedia presentes en las páginas Web, tales como fotografías, música y videos.

Existen dos importantes estrategias en el Web content mining, en primer lugar, la minería de documentos (web page content mining), y en segundo lugar, la minería de búsqueda, asociada a motores de búsqueda [20].

Antes de aplicar cualquier técnica de Web content mining, es necesario transformar los datos de las páginas Web en un vector de características. Muchas veces esto se logra mediante el uso del modelo de espacio vectorial [20].

El modelo de espacio vectorial se basa prácticamente en las palabras más importantes dentro de los documentos, dándole una dimensión dentro del espacio hiper-dimensional generado por las distintas palabras dentro del contexto de los documentos estudiados. De esta manera un documento se representa por un vector de características en el espacio de palabras canónicas obtenidas llamadas tokens.

2.4.4.- Algoritmos supervisados y no supervisados

En Web mining, se utilizan una gran cantidad de algoritmos. Se pueden mencionar algoritmos predictivos, de clustering, de clasificación, entre otros. Dado que la mayoría de los análisis intentan encontrar gustos, tendencias sobre el uso de recursos o sobre texto, los algoritmos de clustering son los más utilizados.

Los algoritmos utilizados en el Web mining se clasifican en dos tipos [27]: supervisados y no supervisados. Los algoritmos supervisados están diseñados para ser utilizados con datos ya clasificados. En el caso que los datos no han sido clasificados, están los algoritmos no supervisados que intentan clasificar los distintos datos en una clase o varias clases.

Los algoritmos no supervisados intentan clasificar bajo similitudes subyacentes entre los distintos elementos que los hacen parecerse más que con los demás, por lo que generan grupos a los que se les llama clusters.

En Web mining suelen usarse algoritmos no supervisados puesto que los documentos usualmente no poseen ninguna clasificación previa.

2.5.- Algoritmos de clustering

Se explicó en secciones anteriores que para Web mining unos de los tipos de algoritmos más utilizados es el de clustering. Es por eso que en esta sección serán revisados más profundamente.

Los algoritmos de clustering han sido utilizados en una gran gama de problemáticas, generalmente teniendo la intención de encontrar conjuntos o grupos que contengan un grado de similitud o disimilitud dentro de un universo de elementos. La medida de similitud o disimilitud entre los elementos es la que define la distribución de los elementos, generando subconjuntos, que bajo este criterio, tienen una mayor relación entre si. Dependiendo de la medida estudiada, los elementos que se relacionan de manera más fuerte con otros suelen poseer características similares, incluso en aspectos no incluidos en la medida utilizada.

Estos algoritmos de clustering o agrupamiento, permiten clasificar elementos del universo de manera natural, vale decir, de acuerdo a las características generales de los elementos y la medida elegida, se generan grupos o clusters que no se conocían con anterioridad al análisis. Es así como a estos algoritmos, dentro de la minería de datos, se les considera como técnicas de aprendizaje no supervisado, puesto que la relación entre los elementos no tiene relación con una variable objetivo de manera a priori.

De la clasificación de los algoritmos de clustering, se mencionan al menos tres [21]:

- Clustering jerárquico: se construyen jerarquías entre los elementos de manera iterativa
- Clustering de particionamiento: se construye una partición del conjunto universo de datos.
- Clustering basado en densidad: se basa en el concepto físico de densidad. Dependen de la cantidad de elementos en un espacio delimitado por un radio específico.

Algoritmos de clustering de particionamiento

Matemáticamente, una secuencia de subconjuntos $A_i, i \in I$ del conjunto *A* es una partición sobre *A* si

1. $A_i \neq \phi, \forall i \in L$
2. $\bigcup_{i \in L} A_i = A$
3. $A_i \cap A_j \neq \phi \Rightarrow A_i = A_j$

Los algoritmos de clustering de particionamiento tienen el objetivo de crear una partición de los elementos, partiendo por el uso de la medida de similitud o disimilitud empleada, con la intención que los elementos en cada una de los subconjuntos se encuentren lo más relacionadas entre si.

Existen una gran variedad de algoritmos, tales como el K-means y sus variantes, c-fuzzy, QT y otros que se basan en grafos [30]. Además, existen herramientas que se pensaron en otros contextos que se pueden utilizar como técnicas de clustering como es el caso de las redes de Kohonen o también llamadas SOM.

2.5.1.- K-means

K-means es un algoritmo de clustering de particionamiento que intenta generar una partición de n elementos en k clusters en cual cada elemento pertenece al conjunto de su media más cercana, en el caso de una medida de disimilitud [40].

Dado un conjunto de observaciones $(x_1, ..., x_n)$, donde cada observación es un vector de valores reales de dimensión *d*, el algoritmo de K-means intenta obtener una partición de las *n* observaciones en *k* conjuntos, con $k \leq n$, $S = \{S_1, ..., S_k\}$ que minimice la suma de cuadrados ínter cluster, es decir, resuelva [43]:

Ecuación 3: conjunto solución k-means

$$\arg\min_{s} \sum_{i=1}^{k} \sum_{x_j \in S_i} \left\| x_j - \mu_i \right\|^2$$

donde μ_i es la media de los puntos del conjunto S_i.

El algoritmo suele ser implementado de manera iterativa como se muestra a continuación:

Dado una partición inicial $c_1^1,...,c_k^1$, obtenida generalmente por heurísticas y aleatorizaciones, el algoritmo procede alternándose en los siguientes pasos:

1. Paso de asignación: Asigna cada observación al cluster con el centroide más cercano.

2. Paso de actualización de centroides: Calcula el nuevo prototipo o centroide de las observaciones en el cluster. Sea $V_{jt} = \{v_1,...,v_{q_j^t}\}$ el conjunto de q_j^t vectores asociados al centroide c_j^t con $j = 1...k$. El próximo centroide se determina como c_j^{t+1}, como la media de V_j^t.

3. El algoritmo termina cuando $c_j^t \approx c_j^{t+1}$.

Esta técnica corresponde a una heurística, por lo que no se puede aseverar que el resultado convergerá a un mínimo global, lo que depende estrictamente del conjunto de medias iniciales.

K-means posee una gran cantidad de variantes las que cambian aspectos tales como el determinismo de la partición, generando efectos probabilísticas y de lógica difusa al momento de incluir un elemento a un cluster o aquellos que intentan optimizar el conjunto inicial de medias para una mejor convergencia [31,32]. Una de las variantes más conocidas es aquella que utiliza elementos del conjunto analizado, como puede ser la mediana, denominado K-Medoids [47].

El algoritmo también posee problemas con elementos lejanos u outliers, por lo que en muchas ocasiones es preferible utilizar elementos propios del conjunto original como prototipos, tales como las medianas.

Ventajas y desventajas de K-means

Como ventajas del algoritmo K-means, se encuentra que es un algoritmo muy sencillo de diseñar e implementar para todo tipo de elementos, además el tiempo de ejecución requerido suele ser muy corto por lo que es posible ejecutarlo varias veces y así comparar sus resultados. Sin embargo, las desventajas se van principalmente en la necesidad de saber cuantos clusters existen en los datos (k), podría no encontrarse un mínimo global sino inclusive un máximo, en el peor caso la convergencia es lenta con la posibilidad de no converger y posee problema con los elementos “lejanos” o llamados outliers, aunque esto se puede resolver utilizando medianas en vez de medias.

2.5.2.- Self-organizing Feature Maps

SOM (Self-organizing map) o SOFM (Self-organizing feature map) es una red neuronal artificial que se entrena utilizando aprendizaje no supervisado para producir una representación de menor dimensionalidad, del espacio real llamada mapa. En la mayoría de los casos esta representación es en dos dimensiones. La mayor diferencia entre SOM y redes neuronales es que las primeras utilizan el sentido de vecindad, manteniendo una topología respecto al espacio original.

SOM es útil para poder visualizar espacios multidimensionales en espacios con menos dimensiones, en forma de un escalamiento. Este modelo fue descrito por Tuevo Kohonen, por lo que a menudo se le llama mapa de Kohonen [37].

Los mapas consisten principalmente en un conjunto de neuronas, que pasan a tener las mismas características de un nodo del conjunto original de elementos. Estas neuronas son posicionadas en una grilla, que puede ser hexagonal o rectangular. El objetivo principal de estas redes es poder describir un mapeo de los elementos iniciales en un mapa de menor dimensión, conservando las diferencias iniciales. La forma de mapear las relaciones es encontrando la

neurona que se parezca más o tenga menor distancia al elemento, y así entrenar la red con los elementos e ir difundiendo los efectos respecto a una topología.

Componentes y funciones de SOM

SOM es una red neuronal artificial, por lo que se basa en el principio del aprendizaje.

El entrenamiento utiliza aprendizaje competitivo, vale decir, que cada neurona compite por los elementos ingresado al entrenamiento. Particularmente, cuando un elemento original se presenta ante el modelo, se calcula la medida de similitud o disimilitud, y se encuentra la neurona que posee la mejor relación con el elemento, a la cual se le denomina la unidad de mejor ajuste (best matching unit - BMU). Los valores escalares o pesos que representan a la neurona cambian respecto al elemento bajo un aprendizaje que se describe en la Ecuación 4.

Ecuación 4: Ecuación de aprendizaje SOM

$$Wv_{t+1} = Wv_t + \Theta(h,t)\alpha(t)(D_t - Wv_t)$$

con:

- t es la iteración actual
- λ es el limite de la iteración temporal
- W_v es el vector de pesos
- D es el vector de entrada
- $\Theta(t)$ es la función de distancia respecto al BMU, usualmente llamada función de vecindad
- $\alpha(t)$ es la función de aprendizaje respecto al tiempo
- h función de la diferencia de radios entre el BMU y otra neurona de la grilla

Dependiendo de los puntos elegidos por la función de vecindad, es posible definir distintas topologías, tales como:

- Topología abierta: la idea es mantener la noción de espacio geométrico bi-dimensional en el proceso de aprendizaje. Las neuronas situadas en el borde rompen abruptamente con el aprendizaje, por ende, el efecto es más notorio cuando el BMU esta cerca de ellas.

- Topología cilíndrica: se conectan dos bordes del mapa, propagando el aprendizaje al otro lado. Sin embargo, los otros bordes quedan sin conexión.
- Topología toroidal: se conectan todos los bordes, y se intenta mantener la continuidad del espacio.

Luego la evolución del algoritmo de entrenamiento del SOM es como sigue:

1. Aleatoriamente crear los pesos de los vectores prototipos dentro del mapa
2. Seleccionar un vector de entrada
3. Recorrer cada nodo del mapa
    - Usando la medida de similitud o disimilitud de vector de entrada con el vector de pesos del mapa
    - Encontrar el nodo que maximiza o minimiza la medida de similitud o disimilitud respectivamente
4. Actualizar los nodos dentro de la vecindad del BMU, haciéndolos más cercanos al vector de entrada con la Ecuación 4.
5. Incrementar t y repetir desde 2 mientras $t < \lambda$

Ventajas y Desventajas de SOM

Como ventajas de este algoritmo de clustering, están la no necesidad de conocer la cantidad de clusters de los datos a priori (k) y que el cluster depende de centroides o prototipos que se van creando naturalmente con los datos de entrada. No obstante, las desventajas suelen estar asociadas a que las implementaciones suelen ser más complejas puesto que se necesita mantener el mapa actualizado siempre, es requisito definir una gran cantidad de parámetros y existen conceptos complicados como topologías y funciones de aprendizaje, se deben realizar una cantidad considerable de iteraciones, puede existir un sobre ajuste desmedido y no representar el universo de datos como suele suceder en las redes neuronales y para interpretarlas se suelen necesitar representaciones gráficas complicadas.

2.5.3.- Medidas de similitud, disimilitud y distancias

Los elementos de un conjunto, en muchas ocasiones, deben ser comparados entre ellos para poder clasificarlos respecto a sus características. Medidas de similitud, disimilitud y distancias son descripciones numéricas de cuán cerca o lejos se encuentran dos elementos.

En el estricto sentido matemático, las distancias poseen un conjunto de reglas que las determinan, en particular las métricas que se define como a continuación:

Una métrica sobre un conjunto X, es una función (denominada la función de distancia o simplemente distancia)

$d : X \times X \rightarrow \Re$, con $\Re$ el conjunto de los números reales. Para todo *x, y, z* en *X*, esta función debe satisfacer las siguientes condiciones:

1. No negatividad

Ecuación 5

$$d(x, y) \geq 0$$

2. Identidad de indiscernibles

Ecuación 6

$$d(x, y) = 0 \Leftrightarrow x = y$$

3. Simetría

Ecuación 7

$$d(x, y) = d(y, x)$$

4. Desigualdad triangular

Ecuación 8

$$d(x, z) \leq d(x, y) + d(y, z)$$

Existe una gran cantidad de categorizaciones de distancias que no cumplen con todas las reglas anteriores, tales como las pseudométricas, quasimétricas y premétricas, entre otras. La gran mayoría de estas métricas relaja las restricciones más estrictas, y que generalmente cuentan en los espacios euclidianos como la desigualdad triangular.

Dentro de las distancias más conocidas se encuentran las distancias de Minkowski la cual se define matemáticamente como:

Para un punto $(x_1,...,x_n)$ y un punto $(y_1,...,y_n)$, la distancia de Minkowski de orden $p \geq 1$, se define como:

Ecuación 9: Distancia Minkowski orden p

$$L_p = \left(\sum_{i=1}^{N} |x_i - y_i|^p \right)^{1/p}$$

Se muestran las distancias de Minkowski más conocidas:

- Distancia City Block o Manhattan (o L_1)

Ecuación 10

$$d(x, y) = \sum_{i=1}^{N} |x_i - y_i|$$

- Distancia Euclidiana (o L_2)

Ecuación 11

$$d(x, y) = \sqrt[2]{\sum_{i=1}^{N} |x_i - y_i|^2}$$

- Distancia Chevychev (o L_∞)

Ecuación 12

$$d(x, y) = máx(|x_i - y_i|)$$

En ciertos contextos, se prefiere utilizar medidas de similitud. Una medida de similitud entre dos puntos x e y en X, es una función $s(x, y) : X \times X \rightarrow \Re$ que satisface las siguientes condiciones:

1. Máxima similitud

Ecuación 13

$$s(x, x) = 1$$

2. Similitud positiva

Ecuación 14

$$s(x, y) \geq 0$$

3. Simetría

Ecuación 15

$$s(x, y) = s(y, x)$$

En la investigación para la recuperación de la información, una medida de similitud altamente utilizada es la distancia coseno. Esta distancia se define como a continuación:

Dado dos vectores de atributos, A y B, la medida de similitud coseno, se representa usando el producto punto vectorial y las normas de la Ecuación 16.

Ecuación 16: Similitud Coseno

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^{N} A_i \times B_i}{\sqrt[2]{\sum_{i=1}^{N} (A_i)^2} \times \sqrt[2]{\sum_{i=1}^{N} (B_i)^2}}$$

Esta distancia, en el contexto del las frecuencias, como es el caso del modelo TFIDF, se encuentra dentro del conjunto [0,1]. En la Figura 4, se describe geométricamente que significa esta similitud, en el caso (a) el valor de la similitud es cero, los vectores son ortogonales, en (b) la similitud es más grande, y en (c) la similitud es máxima. Esta medida de similitud no cumple con todas las definiciones de una distancia o métrica, en particular la restricción de la desigualdad triangular, pero para text mining es mucho más certera que el uso de distancias como las de Minkowski.

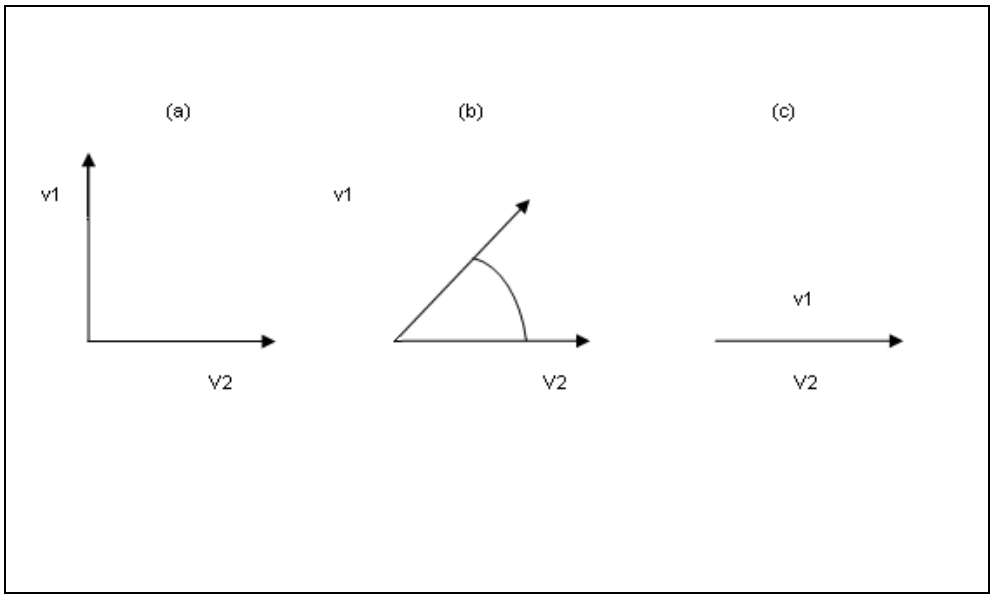

Figura 4: Esquema de la geometría de la similitud coseno
Fuente: elaboración propia

Existen otros tipos de distancias, que sirven para comparar instancias no numéricas, tales como cadena de caracteres o bits. Dentro de aquellos se encuentran las distancias de Levenshtein y Hamming, entre otras.

La distancia de Levenshtein o también llamada distancia de edición, contabiliza la cantidad de inserciones, intercambios y eliminaciones de caracteres mínima que se deben realizar para transformar una cadena de caracteres en otra. En el web usage mining, el uso de esta distancia ha sido de gran interés, siendo utilizada en varios trabajos, especialmente intentando capturar las disimilitudes existentes en las combinaciones de páginas utilizadas del sistema por los usuarios [18,19,38,39].

2.5.4.- Análisis de clustering

El objetivo del clustering de particionamiento es dividir el conjunto universo de datos en subgrupos. Estos deben ser significativos, útiles o poseer ambas características. En varias ocasiones el análisis generado por los algoritmos de clustering quiere ser utilizado para propósitos como el resumen de la información o poder generar un modelo de predicción significativa respecto a nuevos datos comparables con el modelo y sus características.

Luego de generar el clustering, se requiere evaluar el modelo analizando si los clusters están bien formados, y si el modelo matemáticamente cumple con el objetivo general, generar una agrupación natural que minimice o maximice, dependiendo del caso, las relaciones de similitud o disimilitud.

El problema se presenta ya que la mayoría de los algoritmos de clustering poseen una cuota de aleatoriedad, principalmente al escoger las semillas iniciales, por lo que es muy probable que en distintas ejecuciones se lleguen a resultados completamente distintos.

Para analizar los resultados de los algoritmos de clustering, existen dos enfoques utilizados en la literatura. El primero orientado al análisis no supervisado basado en el uso de la matemática, mientras que el segundo utiliza información de clasificación supervisada [33].

En el enfoque no supervisado se encuentra principalmente el análisis por cohesión y separación. La cohesión de un cluster se calcula como la suma de las distancias interior, mientras que la separación es la suma de las distancias de los elementos a otros clusters. Intuitivamente, al intentar generar un modelo de clusters, el objetivo es que los clusters tengan un grado de cohesión bajo, y una separación lo más grande posible.

En el caso de clusters con prototipo el cálculo de estas medidas se realiza con las siguientes fórmulas [33]:

- Cohesión

Ecuación 17

$$cohesión(C_i) = \sum_{x \in C_i} d(x, c_i)$$

- Separación

Ecuación 18

$$separación(C_i) = m_i \sum_{x \in C_i} d(c_i, c)$$

donde m_i es la cardinalidad de C_i y c es el centroide o mediana de todo el conjunto de elementos.

Cuando se consta con información externa, se puede contrastar la clasificación que tienen ya los elementos con la resultante de los algoritmos de clustering. Este enfoque suele darse por medidas tales como entropía, pureza, F-measure, entre otros. Estas medidas suelen utilizarse para observar el rendimiento del modelo de clasificación.

Definiendo los más conocidos se encuentran [33]:

- Entropía: el grado en que cada cluster se constituye de elementos de una misma clase. Asumiendo que la probabilidad de que un cluster *i* contenga elementos de la clase *j* es $p_{ij} = m_{ij} / m_i$, donde m_{ij} es la cantidad de elementos de la clase *j* en el cluster *i*, m_i es la cantidad de elementos en el cluster *i*, la entropía del cluster *i* se define como $e_i = -\sum_{j=1}^{L} p_{ij} \log_2 p_{ij}$, con *L* el número de clases. Finalmente la entropía del modelo es $e = \sum_{i=1}^{K} \frac{m_i}{m} e_i$ con K el número de clusters y m el total de elementos.
- Pureza: otra medida que se basa en lo mismo que la entropía. En este caso la pureza se define para el cluster *i* como $p_i = \max_j pij$, mientras que la pureza total es $e = \sum_{i=1}^{K} \frac{m_i}{m} p_i$
- Precisión: la fracción de que los elementos en un cluster pertenezcan a una cierta clase. La precisión de un cluster *i* respecto a una clase *j* es p_{ij}.
- Recall: la medida de la cual un cluster contiene todos los elementos de una clase. El recall de un cluster *i* con respecto a una clase *j* es m_{ij}/m_j donde m_j es el total de elementos de la clase *j*.

- F-Measure: es una combinación de la precisión y el recall que mezcla la idea de que un cluster tenga sólo elementos de una clase, y que contenga a todos de esa clase. F-Measure de un cluster *i* respecto a una clase *j* es como se muestra en la Ecuación 19.

Ecuación 19

$$F(i, j) = (2 \times precision(i, j) \times recall(i, j)) / (precision(i, j) + recall(i, j)) .$$

2.6.- Representaciones en text mining

Una de las representaciones más utilizadas en text mining sobre los documentos es el vector space model (en español, modelo de espacio vectorial). Este modela los documentos como un vector algebraico, basándose generalmente en la indexación de los términos. Generalmente, cada escalar del vector representa un peso de importancia que posee el término indexado en el documento.

Uno de los modelos más utilizados en recuperación de la información es el TFIDF, el cual se basa principalmente en la frecuencia que poseen los términos sobre los documentos. Esto fue explicado en la sección 2.3.4.

El modelo TFIDF, para una gran cantidad de documentos, genera vectores de alta dimensionalidad, por lo que en muchos casos, genera una desventaja para su análisis comparativo posterior. Una de esas desventajas es que los vectores resultantes poseen una gran cantidad de ceros, por lo que no aportan información valiosa, y en general generan una mayor distancia al momento de la comparación. Otra desventaja es al momento de utilizarlos en algoritmos de clustering u otras técnicas de text mining, puesto el algoritmo tomará un mayor tiempo en realizar los cómputos.

Luego, existen variadas técnicas que intentan incorporar mayor semántica, mejorar la carga semántica al representar los textos con los cuales se trabaja al mismo tiempo y también reducir la dimensionalidad de la representación vectorial. A continuación se detallan dos técnicas al respecto.

Metodología de extracción de conocimiento que se basa en el uso de lógica difusa sobre conceptos obtenidos por la información y criterio de expertos que conocen lo que se desarrolla en los textos y así obtener la relevancia a partir de los conceptos. Esto se condice con que la representación de objetos es mucho mejor con conceptos que con palabras.

El enfoque permite tener un grado de certeza al momento de definir un documento respecto a un concepto. El proceso de identificación de conceptos, permite incorporar el conocimiento de los analistas, expertos y administradores al proceso de minería [2,35].

Al momento de estudiar los conceptos, estos se asocian a términos, siendo definido el puntaje que tiene cada termino por concepto. El valor cero significa que no existe relación, el valor uno representa que el termino tiene una alta relación.

El uso de ésta metodología en el contexto del text mining, se asume que un documento puede ser representado por la relación difusa [Conceptos x Documentos], que es un matriz, en la que en cada fila hay un concepto y cada columna representa un documento. Los términos son palabras que definen conceptos. La ecuación determina la relación que hay entre los conceptos, términos y documentos, utilizando las operaciones x y $\otimes$, como operaciones difusas.

Ecuación 20

$$[\text{Conceptos} \times \text{Documentos}] = [\text{Conceptos} \times \text{Términos}] \otimes [\text{Términos} \times \text{Documentos}]$$

Utilizando, algunas teorías de lógica difusa, la multiplicación matricial anterior, termina siendo la suma limitada a uno, de la multiplicación de los términos de un documento por el grado de pertenencia a cada concepto.

En el contexto anterior, es aplicable a representaciones numéricas de términos sobre documentos, tales como el TFIDF.

Latent Dirichlet Allocation (LDA)

Un modelo de tópicos puede ser considerado como un modelo probabilístico que relaciona documentos y palabras a través de variables que representan los tópicos principales inferidos del texto mismo [35,36]. En este concepto, cada documento puede ser considerado una mezcla de tópicos que probabilísticamente generan palabras dentro del cuerpo del documento.

LDA es un modelo de tópicos. Es un modelo bayesiano donde los tópicos son inferidos de distribuciones de probabilidad a partir de un conjunto de entrenamiento. La idea principal es que cada tópico es modelado como una distribución de probabilidades sobre un conjunto de palabras representadas por el vocabulario, y cada documento como una distribución de probabilidades sobre un conjunto de tópicos. Estas distribuciones se representan como distribuciones multinomiales de Dirichlet.

Este proceso no requiere la ayuda de expertos, puesto que la clasificación proviene estrictamente del conjunto de entrenamiento.

Matemáticamente para el LDA, dado los parámetros α y β, y la distribución conjunta de mezcla de tópicos θ, la idea es determinar la distribución de probabilidad para generar de un conjunto de tópicos T, un mensaje compuesto por un conjunto S de palabras $w = (w_1,\ldots,w_s)$

Ecuación 21

$$p(\theta, z, w \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{s=1}^{S} p(z_s \mid \theta)\, p(w^s \mid z_s, \beta)$$

donde $p(z_s \mid \theta)$ puede ser representado por la variable aleatoria θ_i, de manera que el tópico z_s esté presente en el documento i $(z_s^i = 1)$. Una expresión final se deduce integrando la ecuación (anterior) sobre la variable aleatoria y sumando sobre los tópicos en $z \in T$. Dado lo anterior, la distribución marginal de un mensaje puede ser definido como:

Ecuación 22

$$p(w \mid \alpha, \beta) = \int p(\theta \mid \alpha) \left(\prod_{s=1}^{S} \sum_{z_s \in T} p(z_s \mid \theta)\, p(w^s \mid z_s, \beta) \right) d\theta$$

Luego, la finalidad de LDA es estimar las distribuciones para construir un modelo generativo para los cuerpos de documentos dados.

De la misma manera propuesta en el caso del Concept-based, se puede utilizar LDA y utilizarlo en conjunto con el TFIDF para finalmente generar la representación de los documentos a partir de los tópicos, o denominada representación Topic-based [34].

3.- Solución propuesta

3.1.- Identificación de acciones de moderación

Cada comunidad virtual de práctica tiene sus propias reglas preestablecidas que le permiten a esta entidad social lograr sus objetivos fundamentales. Cada regla o norma se ha creado con la intención de facilitar la interacción entre los distintos individuos pertenecientes, pero basados muchas veces en la perspectiva del administrador, respecto a sus experiencias y subjetividades.

Teniendo en cuenta la subjetividad de cada comunidad, identificar el comportamiento de los usuarios y moderadores varía con cada comunidad, y parece necesario estudiar cada caso de manera particular. No obstante, existen patrones o acciones que han sido heredadas de experiencias pasadas y siguen siendo habituales en el mundo de las comunidades virtuales.

3.1.1.- Acciones del moderador

Dentro de las acciones del moderador más importantes se encuentran las siguientes:

Identificación de individuos problemáticos

Muchas veces dentro de las comunidades virtuales, aparecen individuos con características problemáticas, que al momento de interactuar suelen tergiversar las opiniones de los demás integrantes, o muchas veces insultan explícitamente a los distintos usuarios. En este caso, lo que se suele hacer es advertirle de manera reiterada respecto a su comportamiento, aconsejándole tomar una conducta más apropiada acorde con el interés común. La actividad requiere un tiempo pequeño en ser ejecutada y suele realizarse un par de veces antes de ejecutar una sanción mayor. No obstante, requiere de un administrador que esté 100% involucrado con la comunidad, para que advierta a estos individuos y ejecute las acciones que corresponda oportunamente.

Cierre de conversación

Dentro de una comunidad, suelen existir preguntas o comentarios en los cuales terminan aportando una gran cantidad de miembros, dado el gran interés que puede suceder respecto a

ese tema. Lo necesario es terminar con la conversación cuando ya se ha resuelto completamente, para que tanto los nuevos integrantes como personas que encuentren casualmente el tema en instantes muy posteriores no abran el tema, sino que si tienen una duda puedan utilizar la información aportada o generen una nueva conversación referenciando a la anterior (si viene al caso). La ejecución de esta acción por parte de un moderador es instantánea al momento de identificarlo, lo cual requiere que se esté revisando constantemente dentro de la comunidad.

Ban o restricción de ingreso de un integrante (Baneo)

Al momento que se identifica a un integrante de la comunidad como conflictivo, se apela a un cambio de actitud dentro de su interacción con los demás una cantidad determinada de instancias. En estas instancias el castigo es marginar al miembro de la comunidad a estar alejado de las circunstancias normales y de interacción con el resto de los integrantes por un cierto periodo de tiempo, donde la intención es que exista alguna especie de recapacitación respecto a lo sucedido.

Identificación temprana de una conversación muerta

En el caso de las interacciones entre los individuos pertenecientes a la comunidad virtual, muchas veces la conversación parte sin tener un propósito o no puede tener un desarrollo por alguna razón intrínseca. Lo anterior suele suceder cuando la consulta posee un grado de contradicción o no es del interés del resto de la comunidad. El moderador, como experto en la comunidad, identifica estas conversaciones y procede a dar instrucciones o cerrarla.

Conversaciones mal clasificadas o existentes dentro de la comunidad

Las comunidades virtuales suelen clasificar su contenido de manera que sea más fácil el acceso hacia este por parte de los individuos. Una de las acciones más recurrentes dentro de las comunidades virtuales por parte de los administradores es facilitar la buena clasificación de la información por parte de los miembros. Se intenta que el individuo tenga especial precaución en informarse de lo que ya existe dentro de la comunidad y sus prácticas comunes antes de publicar un mensaje.

3.1.2.- Identificación de comportamiento de usuarios revisables

Basándose principalmente en entrevistas con administradores y moderadores de comunidades virtuales de práctica, los mensajes ingresados al sistema por usuarios son revisados por un modelo de actuar que cada moderador va creando con la experiencia.

Es un acuerdo conjunto, que generalmente cada individuo que juega un rol de moderador, en su primera experiencia tiende a revisar absolutamente todo, puesto que es la única manera de poder ver si existen problemas, o si se necesita intervenir en ayuda de los novatos, entre otras circunstancias. La revisión total de los mensajes de la comunidad implica un esfuerzo sobre humano, y en general, es ineficiente respecto al tiempo utilizado, puesto que la mayoría de las veces el comportamiento de los individuos involucrados es correcto. Luego, los moderadores comienzan a crear mecanismos más eficientes que se basan principalmente en características que proveen las aplicaciones de la comunidad, como la información de mensajes y el orden de ellos por fecha de ingreso.

Algunos indicadores que permiten la priorización al momento de revisar la comunidad en busca de mensajes inapropiados son los que se muestran en la Tabla 2.

Tabla 2 Indicadores de mensajes revisables

Indicador	Razón de su uso
Cantidad de mensajes o respuestas en una Discusión	La cantidad de respuestas representa la importancia que tiene para la comunidad, a mayor cantidad de respuestas, más necesaria es su revisión
Cantidad de vistas de una Discusión	Las vistas representan importancia, pero si existen muchas vistas puede darse a razón del morbo de los individuos por causa de alguna mala acción de los que generaron mensajes
Discusión larga con pocos individuos involucrados	Pocos individuos dentro de una conversación larga, puede existir porque es un tema específico a esas personas o sino porque es una conversación muerta
Mensajes generado por un usuario con mucha actividad	Si el usuario es muy activo, puede ser porque entrega mucho apoyo necesario o porque quiere molestar.
Mensaje producido por un novato en la comunidad	Los novatos no conocen los mecanismos de la comunidad, suelen cometer errores, por lo que revisar sus mensajes es preferible
Mensaje o discusión fuera de clasificación	La clasificación existe para una mejor navegación en la comunidad, es regla general mantenerla, por lo que siempre se esta revisando que se cumpla tal regla
Últimos mensajes dentro del sistema	Suelen priorizarse los últimos mensajes, puesto que son los que no han sido revisados

3.2.- Diseño de algoritmos

En este trabajo se utilizó el proceso KDD, en el cual se encuentra una secuencia de pasos y supuestos que se van aplicando a cada uno de sus subprocesos. La intención es utilizar este proceso para poder generar un modelo de grupos de comportamientos de usuarios en la comunidad virtual de práctica. En consecuencia generar un análisis de estos comportamientos con la intención de encontrar relaciones entre éstos y los asuntos de la moderación.

En este subcapítulo se explican cada uno de los supuestos, tanto en el diseño, como implementación genérica de los subprocesos que permitieron llevar a cabo el objetivo de este trabajo.

3.2.1.- Descripción de los datos

En este trabajo, los datos utilizados corresponden a aquellos que permiten modelar la navegación de usuario en una plataforma Web con la información del contenido de dicha navegación. En ese contexto, la entrada del proceso son los Web log de usuario, que contienen información de la navegación más el contenido de las vistas realizadas.

Los Web logs, representan los requerimientos de los usuarios a un servidor Web, guardándose en ellos la IP, el agente, el tiempo, la URL[6], entre otros datos importantes.

Los datos del contenido, es texto el cual debe ser procesado para su uso en los contextos explicados en la sección 2.3.2. El contenido en una comunidad virtual de práctica tiene poco de ser estático, y es importante construirlo de acuerdo a las hipótesis más fidedignas.

Los datos a considerar en este trabajo, son los que representen la navegación de un usuario, tanto por las páginas que están accediendo, como por el contenido textual que están visualizando. Objetos multimedia como fotografías, videos no son considerados, puesto que requieren del uso de técnicas avanzadas que permitan generalizar su semántica o significado en la metodología utilizada.

[6] URL: Uniform Resource Locator

3.2.2.- Pre-Procesamiento de los datos

En el diseño de la etapa de pre-procesamiento se deben advertir dos instancias, primero, lo que tiene que ver con los Web logs de usuario, con el contenido de las páginas Web.

Para el caso de los Web logs de usuario, se elimina todo lo que no sea una petición explícita por parte del cliente, en general, todos los objetos multimedia, como fotografías, y se dejan sólo las acciones o páginas que muestran contenido, asumiendo la hipótesis que el tiempo utilizado dentro de esas páginas tiene relación con la importancia que su contenido tiene para el usuario. Se eliminan todas las instancias en que los accesos tuvieron errores y aquellos accesos donde explícitamente se nota que fueron realizados por entidades no humanas, es decir, robots. En síntesis, al final de este proceso, la información procesada, contiene acciones estrictamente ligadas a despliegue o ingreso de información al sistema.

En el caso del texto, el pre-procesamiento se basa especialmente en eliminar las stopwords y utilizar stemming dentro de las palabras claves.

3.2.3.- Sesionización

Para generar la sesionización, se utilizan heurísticas de reconstrucción, principalmente basadas en tiempo de navegación. Web logs anónimos, sin el uso de cookies, obliga el uso de estrategias de sesionización retroactivas. Existen comunidades, en los que la aplicación que permite el intercambio de información posee archivos de logs de usuario más sofisticados, pudiendo generarse sesiones con menor error. El proceso diseñado en este punto puede ser generalizado a esos casos según corresponda.

Para este trabajo las sesiones a investigar son aquellas en los cuales los usuarios ingresaron información pública al sistema, vale decir, poseen al menos un ingreso en modo de post por sobre la aplicación. La manera de identificar estas sesiones, es realizar el proceso de sesionización para todas las entradas y luego filtrar aquellas que hayan generado un post al menos.

Luego, de la sesionización, se propone un post-proceso que filtre aquellas acciones que no tengan directa relación con mensajes o discusiones dentro de la comunidad. Lo anterior, es

a razón que otras acciones que no tengan que ver con mensajes generan ruido y no tienen que ver con el posible accionar de los moderadores de la comunidad.

3.2.4.- Transformación

Para la transformación de los datos, requerida para el uso de los distintos algoritmos de minería de datos, se decide usar una representación vectorial de la secuencia. Para esto es necesario utilizar un largo de sesión fijo mínimo, permitiéndoles a sesiones más largas ser incluidas truncando su largo hasta el largo mínimo elegido. Lo anterior, permite que una mayor cantidad de sesiones pueda ser estudiada puesto que sesiones de un largo fijo puede reducir los resultados de gran manera.

El vector que representa una sesión lleva un identificador de la página o acción dentro del sistema que el usuario requirió, que representa la URL, el tiempo utilizado y el contenido de ésta. Cada escalar del vector en secuencia representa en orden de visita el comportamiento del usuario en una sesión.

Al momento de modelar la secuencia de la sesión, para este diseño es importante modelar el contenido de las acciones. Dado que en la sección de sesionización se advirtió que se dejarían activas las acciones que tuvieran que ver directamente con mensajes en la comunidad, el contenido de cada acción se obtiene de los mensajes de la comunidad. En este punto se pueden decidir que es lo que el usuario realmente veía al momento de mirar una discusión, esto puede ser el post que inicia la discusión o a quién desea responderle.

La representación del texto, se basa en el modelo TF-IDF, definiendo cada documento dentro del conjunto de vistas, como un vector de características.

Pero el modelo TFIDF posee tanto ventajas como desventajas al momento de representar documentos. Esto genera la necesidad de utilizar otras herramientas para luego generar un análisis más robusto que permita comparar los resultados obtenidos. Para lo anterior, se han elegido las representaciones Concept-based y LDA.

3.2.5.- Diseño de algoritmos de clustering

En este trabajo se utilizó el clustering particional, con la intención de encontrar patrones de comportamiento dentro de las sesiones de usuario. Para esto se trabajó con dos algoritmos que permiten obtener modelos de clustering: K-medoids y SOM.

El input para estos algoritmos varía, dependiendo de la construcción del prototipo generado para cada algoritmo. En general, se requiere la representación de las sesiones y la distancia determinada para el experimento.

SOM o red neuronal de Kohonen

El algoritmo SOM, que en este trabajo tiene la intención de proveer el número de clusters inherentes en el conjunto de sesiones y así utilizar K-medoids, requiere toda la representación de los elementos que vayan a ser utilizadas para comparar con sus prototipos.

Recordando que este algoritmo funciona con una red o grilla de neuronas, que vienen a ser prototipos de los elementos estudiados, necesitando ser comparados para encontrar la mejor neurona, y luego adaptar todos los prototipos de acuerdo a una regla de entrenamiento y cuidando la topología. En su diseño, se requiere incluir representaciones de los aspectos que serán utilizados en la comparación, tales como la representación de la secuencia, el contenido, entre otros.

Para el diseño de este trabajo, el algoritmo tiene el propósito de encontrar el número de grupos inherentes del conjunto de datos. La manera de la frecuencia de elementos cuya neurona más símil se distribuye dentro de la grilla, muestra finalmente cuantos clusters naturales tienen los datos. Sin embargo, en muchas ocasiones la distribución implica que ciertos grupos no poseen elementos necesarios para considerarse clusters, por lo que la cantidad real de clusters termina siendo menor a lo que el algoritmo predice. Es por lo anterior, que para efectos prácticos se utilizará como cantidad real de clusters el número de clusters importantes dentro del modelo, vale decir que contengan a la mayoría de los elementos y que sean suficientemente grandes para ser considerados clusters.

K-medoids

El diseño del clásico K-means sólo requiere la utilización de prototipos y las distancias existentes entre los elementos que generarán el modelo de clustering.

El diseño del algoritmo en este trabajo varía de su versión original, puesto que el espacio de las sesiones no es un espacio euclidiano en R^n. Este espacio se define a partir de vectores que representan todas las características, tanto del texto como la secuencia, pero además contienen una cuota de texto, como es el caso de las URL's. Así que en vez de aprovecharse del concepto centroide, lo mejor es utilizar elementos del mismo conjunto como prototipos. El nombre que se le da en la literatura a esta variante es K-Medoids.

En este caso se utilizó el elemento mediana, que es el objeto que para cada conjunto es el más cercano a todos los demás de su mismo conjunto. La mediana se obtiene como el elemento que dentro de un conjunto minimiza la suma de distancias (o maximiza la suma de las similitudes) a todos los otros. Matemáticamente se explica en la Ecuación 23 en la versión para medidas de disimilitud.

Ecuación 23: Definición mediana

$$mediana = \arg\underset{x \in C}{mín}\{d(x, y), y \in C\}$$

Este algoritmo, permite el análisis de manera más certera y confiable que el SOM, puesto que está diseñado para esto. En este trabajo se utiliza la información del cluster realizado con SOM para capturar la cantidad de clusters de entrada para este algoritmo. Sin embargo, es un algoritmo aleatorio, puesto que los clusters iniciales se eligen al azar, por ende en cada ejecución los resultados podrían, con alta probabilidad, ser diferentes. Para la elección del modelo de cluster, se utilizan los conceptos de cohesión y separación. Para este trabajo se propone el uso del indicador de la Ecuación 24.

Ecuación 24: Índice cohesión-separación

$$indicador = \frac{cohesión}{separación}$$

donde,

Ecuación 25

$$cohesión = \sum_i cohesión(C_i)$$

Ecuación 26

$$separación = \sum_i separación(C_i)$$

Ambos conceptos explicados en la sección 2.5.4, generan el indicador utilizado en este trabajo. Para la elección entre los resultados dentro de un conjunto de experimentos, se usa el modelo que genere de valor máximo para el caso de medidas de similitud y el de valor mínimo en el caso de medidas de disimilitud. Con lo anterior se utilizan los modelos con mejor similitud o disimilitud global, respectivamente.

3.2.6.- Medidas de similitud y disimilitud entre sesiones

En este trabajo, el uso de las medidas de similitud o disimilitud para segmentar los comportamientos de usuario dentro de las sesiones de usuarios es fundamental. En la literatura existen una cantidad grande de experimentos que intentan capturar lo más importante dentro de la navegación de usuario para poder comparar los comportamientos y así poder encontrar aquellos con mayor similitud o menos disimilitud. Los algoritmos de clustering permiten el uso de ambas tipo de medidas, sólo es necesario implementarlos como problemas de maximización para la similitud o minimización para la disimilitud.

Para este trabajo se postulan algunos modelos, que en parte dependen de las medidas. Se utilizarán medidas en la literatura y postuladas directamente para este trabajo.

Modelo 1

La primera medida se basa en el concepto de similitud desarrollado por Velasquez et al. [18].

En dicho estudio se aplica una medida de similitud a la navegación de usuarios a páginas Web, con la intención de capturar la importancia de las páginas, la secuencia de páginas visitadas, el tiempo y el contenido. Esto se puede apreciar en la definición de su medida de similitud expresada en la Ecuación 27.

Ecuación 27: Similitud entre sesiones

$$sm(\alpha,\beta) = dG(\Gamma(\alpha),\Gamma(\beta))\frac{1}{\eta}\sum_{k=1}^{\eta}\tau_k * dp(p_{\alpha,k}^h, p_{\beta,k}^h)$$

donde la descripción de cada elemento se muestra a continuación:

- α,β : son las secuencias de las sesiones o comportamiento de usuarios.
- $\Gamma(\alpha)$: es la representación numérica de la secuencia de navegación.
- $\eta = mín\{C^\alpha, C^\beta\}$: es el mínimo entre las cardinalidades o largos de las secuencias de navegación comparadas.
- $dp(p_{\alpha,k}^h, p_{\beta,k}^h)$: es la similitud entre las k-ésimas páginas de α,β.
- $\tau_k = mín\left\{\frac{t_\alpha^k}{t_\beta^k}, \frac{t_\beta^k}{t_\alpha^k}\right\}$: que representa la similitud de los tiempos relativos.
- dG : es una función que intenta encontrar la similitud entre las secuencias comparadas a partir de la distancia de edición o Levenshtein. Su definición, se encuentra en la Ecuación 28. *L* es la distancia de edición *E* es la función que devuelve el largo de secuencia y *S* es la cadena de caracteres de la secuencia.

Ecuación 28: Similitud de secuencia

$$dG(G_1,G_2) = 1 - 2\frac{L(S_1,S_2)}{\|E(G_1)\| + \|E(G_2)\|}$$

En el caso de este trabajo, la representación de la secuencia no se puede hacer con los identificadores de las páginas visitadas, puesto que las URL son muchas, porque dependen del controlador[7] y las variables que identifican la muestra del contenido específico. Luego, los identificadores simples son diferentes en casi todas las representaciones de las sesiones y la similitud sería cero en la mayoría de las ocasiones. Sin embargo, se propone en este trabajo utilizar la abstracción de la acción, sin las variables, adquiriendo un sentido semántico útil para la comparación de secuencia representa de acuerdo a la intención del usuario en la aplicación.

7 Controlador: dentro de un modelo modelo-vista-controlador, es el responsable de realizar las acciones que, generalmente, el usuario pide a través de la vista.

Modelo 2

La segunda medida, que se crea a partir de nociones de distancias ya utilizadas en contextos de Web usage mining, utiliza el contenido la URL, la secuencia y la información de clasificación del post dentro de la comunidad virtual de práctica.

Este trabajo consta principalmente en la aplicación de técnicas de Web usage mining sobre comunidades virtuales de práctica, las cuales en muchas ocasiones son aplicaciones Web. Sin embargo, existen muchas características dentro de estas técnicas que están pensadas para la navegación sobre un portal o un sitio Web donde los contenidos están definidos por los administradores. En el caso de las comunidades, los contenidos los definen otros usuarios, mediante el ingreso de nuevos mensajes y respuestas.

Una comunidad, para ser manejada de una manera más fácil, suele ser dividida en categorías donde se indica qué tipo de comportamientos y contenidos deben ser ingresados. En muchas comunidades, las categorías suelen reflejar el interés de los integrantes, por lo que las personas suelen usar aquella categoría más acorde con sus intereses.

Bajo las premisas antes discutidas, parece coherente considerar la clasificación explícita de la comunidad en la diferencia de comportamientos de usuarios.

Se postula la medida de disimilitud de la Ecuación 29.

Ecuación 29: Disimilitud de sesiones

$$d(S_1, S_2) = (1 + \Gamma(S_1, S_2)) \sum_{i \in \{1,\ldots,N)} \{(1 + \left| Cat_{s_1 i} - Cat_{s_2 i} \right|) \cdot (1 + \Gamma(Url_{s_1 i}, Url_{s_2 i})) \cdot (1 + d_c(C_{s_1 i}, C_{s_2 i})) - 1\}$$

donde la descripción de cada elemento se realiza a continuación:

- S_i : representa una sesión
- Γ : es la función distancia de edición o Levenshtein
- C_i : es el contenido i-ésimo de la sesión
- Url_i : es i-ésima URL de la secuencia
- Cat_i : es la categoría del i-esimo contenido de la sesión

Como explicación general de esta medida de disimilitud, intenta, mediante multiplicadores expresar que la diferencia de la sesión se basa, en la URL de la acción realizada, la categoría en la comunidad del contenido de la sesión, en la disimilitud del contenido y en la disimilitud que existe en la secuencia de la sesión. El valor -1 incluido en la ecuación es necesario para que cuando todas las disimilitudes sean cero, implique que la suma sea realmente cero, es decir, la disimilitud con la misma sesión es 0.

En la distancia anterior se identifica d_c como una modificación de la medida de similitud coseno. Dado que es una medida de similitud, para ser utilizada en un contexto de disimilitud, entendiendo que su conjunto recorrido es [-1,1] y su máxima similitud se da en 1, se postula el uso de d_c como en la Ecuación 30.

Ecuación 30: Disimilitud coseno

$$d_c(C_i, C_j) = 1 - \cos(C_i, C_j)$$

4.- Aplicación en una comunidad virtual de práctica

Para los administradores de comunidades virtuales, el objetivo principal es mantener a la comunidad funcionando a máxima capacidad. Dado que es un contexto social, las interacciones de los usuarios se verán perjudicadas por actuares de individuos que, tanto por falta de conocimiento o por voluntad propia, generan textos que no corresponden a la clasificación, son agresivos, o bien, no tienen ninguna coherencia con el objetivo de la comunidad. Con el fin de estudiar distintas causas, encontrar patrones de comportamiento y generalizarlos sobre sus resultados dentro de la comunidad.

Utilizando los modelos propuestos, se han realizado varios experimentos a una comunidad virtual de práctica real.

4.1.- Descripción de la comunidad

La comunidad estudiada es el foro de Plexilandia.cl, cuya URL corresponde a www.plexilandia.cl/foro. Esta comunidad se encuentra funcionando desde Septiembre del año 2002 y consta dentro de la base de datos con la información de más de 2000 usuarios. Su página de inicio y presentación se muestra en la Figura 5.

Figura 5: Página inicio foro Plexilandia
Fuente: Elaboración propia

El contenido de esta comunidad se basa principalmente en amplificadores de sonido, que son desarrollados y perfeccionados por los mismos usuarios, compartiendo el conocimiento

obtenido por llevar a cabo esta práctica. El nombre plexilandia viene del concepto “plexi”, que es como se le solía llamar a los amplificadores en los años sesenta, puesto que en la parte frontal tenían polimetilmetacrilato o “plexiglas”.

El perfil de usuario de esta comunidad, suele ser de alto nivel técnico, generalmente con estudios en ingeniería electrónica o carreras afines. La información suele ser muy específica, y aunque se acepta personas que no posean los conocimientos con anterioridad, se logra identificar quienes aportan y quiénes no. Gracias a buscadores como Google, es como llega la mayoría de los usuarios.

El foro está construido basado en un sistema para foros gratuitos llamado phpBB. Esta plataforma codificada en el lenguaje PHP, permite crear y modificar fácilmente comunidades. Además, las páginas terminan siendo una vista de un controlador que maneja las variables que desean ser mostradas.

Particularmente, esta comunidad posee una estructura jerárquica, como suele suceder en este tipo de foros, constando de seis categorías. Estas categorías han sido desarrolladas a través de los años, con las experiencias y necesidades existentes en la comunidad.

Como en la generalidad de las comunidades, el contenido es completamente dinámico, varía dependiendo de lo que vayan ingresando los usuarios. Para reducir este efecto el estudio se realizó en un periodo anterior al actual, para que los posibles efectos y cambios sean menores.

Para el análisis de este trabajo se utilizó la información recopilada de los datos con los que se consta en esta comunidad. El intervalo de tiempo utilizado fue de 6 meses, en detalle: tiempo del primer registro: 2009-03-31 20:55:16, tiempo del último registro: 2009-09-22 00:01:51.

4.2.- Almacenamiento de Datos

El almacenamiento de los datos del foro de la comunidad se basa en la forma que lo realiza el sistema phpBB, que es en una base de datos del tipo MySQL, con una estructura establecida. La base de datos de plexilandia consta de 33 tablas, y en la Figura 6 se muestran las más útiles para este trabajo. Las tablas que tienen la información de los posts y los tópicos,

poseen la información del foro al que pertenecen, el identificador del usuario que generó la información y el tiempo en que se realizó la acción. Luego, mediante estas tablas es posible encontrar la información del usuario, conversación, mensaje y foro ingresado en el sistema. Además, entre las tablas existen relaciones, tales como que una conversación o tópico fue creado por un usuario, y tiene una cantidad de mensajes. A la vez, una conversación pertenece a un foro y los mensajes fueron creados por un usuario.

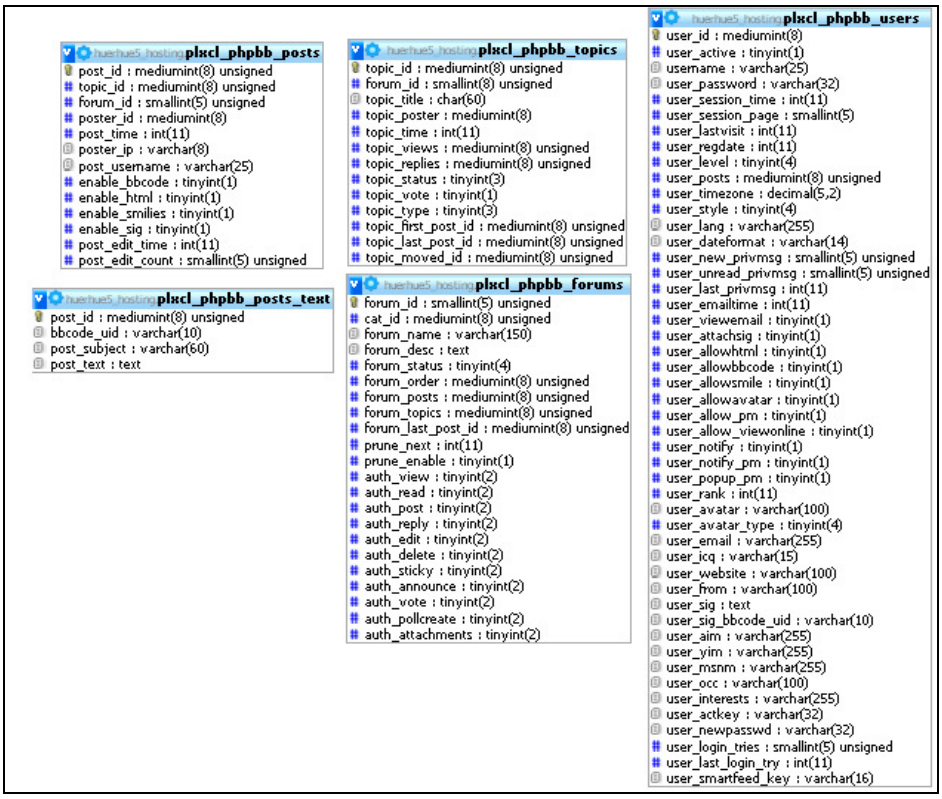

Figura 6: Modelo de datos Plexilandia
Fuente: Elaboración propia

Los datos de la navegación son grabados por el servidor, en donde se guardan en cada línea la información de cada request. En la Figura 7, se muestran ejemplos de líneas del Web log de plexilandia. Se denota que en este log, la información es anónima y sólo del usuario se guarda su dirección IP.

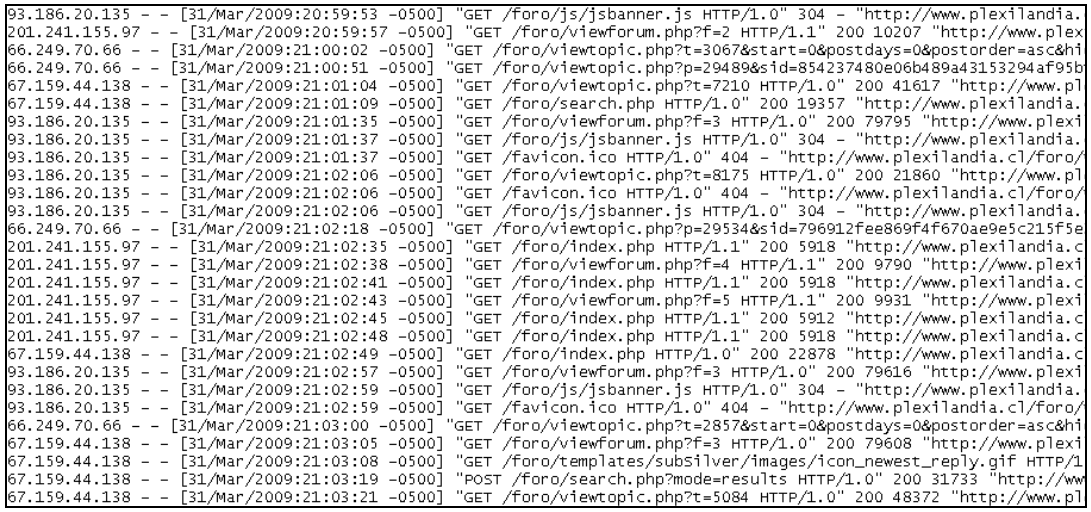

Figura 7: Web log Apache Plexilandia
Fuente: Elaboración propia

4.3.- Selección y pre-procesamiento

Una primera hipótesis de uso dentro del análisis del foro de la comunidad virtual de práctica Plexilandia, es que el navegador le pide al servidor de la comunidad información que se interpreta como una acción, particularmente, un controlador. Luego, en esta etapa, se consideran como acciones explicitas del usuario las de la Tabla 3.

En este punto es importantísimo eliminar aquellas entradas dentro de los Web logs que hayan sido ejecutadas por robots, generalmente utilizados por los motores de búsqueda en Internet. El proceso identificó como robots a 561 direcciones IP. La manera de indagar en el hecho que eran robots fue utilizando el user agent, imputando el hecho que era un robot si no era de los tres tipos convencionales, vale decir, Mozilla, Opera ó BlackBerry. Luego los user agent encontrados con esas características fueron 104, y dentro de la Tabla 4 se muestran los más importantes.

Otro aspecto que se modela dentro del pre-procesamiento del log es el modelo de identificación de posts. Cabe destacar que dentro de las acciones, el post es un proceso de varios pasos. En primer lugar, se genera una vista donde el usuario escribe en un formulario la información del post, y luego envía el post. Por esta razón, por parte de la base de datos, el mensaje es ingresado sólo al momento de enviar el formulario. En el caso de esta aplicación esto se puede identificar encontrando la URL /foro/posting.php con método POST. Luego, como del web log, no es posible identificar el usuario, se utiliza una heurística para encontrar el post, utilizando la información de IP, el tiempo en que se envió y en la información de la URL de

referencia. Por esta razón, esta información se adjunta explícitamente al modelo de URL, para que en el proceso de transformación, donde se genera la representación vectorial de la sesión, puedan interpretarse como posts de usuario. Un ejemplo de esto es la siguiente URL: /foro/posting.php?ip=146.83.193.31&fechats=1245691757, que indica que un usuario generó un post teniendo la dirección IP 146.83.193.31 y en el instante con UNIX TIMESTAMP 1245691757.

Tabla 3: Acciones php del foro de Plexilandia

Controlador	Descripción de la acción
/foro/index.php	Ver página principal
/foro/viewtopic.php	Ver un tópico o conversación
/foro/admin/admin_users.php	Administrar usuarios
/foro/admin/index.php	Ver página de inicio de administración
/foro/faq.php	Desplegar FAQ
/foro/groupcp.php	Administración de grupos
/foro/login.php	Ingreso al sistema
/foro/memberlist.php	Mostrar la lista de miembros
/foro/modcp.php	Administración de moderación
/foro/privmsg.php	Envío de mensaje privado
/foro/posting.php	Ingreso de mensaje
/foro/profile.php	Revisar la información del usuario
/foro/search.php	Búsqueda de contenidos
/foro/smartfeed.php	Relacionado a RSS
/foro/smartfeed_url.php	Relacionado a RSS
/foro/viewforum.php	Ver contenido de un foro
/foro/viewonline.php	Muestra quién está conectado

Tabla 4: Posibles Crawlers

Microsoft Data Access Internet Publishing Provider Protocol Discovery
ShackCrawlAlpha/Nutch-0.9
Googlebot-Image/1.0
Feedfetcher-Google;
Googlebot/2.1 (+http://www.google.com/bot.html)

4.4.- Sesionización

La sesionización se realizó mediante algoritmos reactivos, puesto que los Web logs de Apache configurados al nivel basal, poseen la información de usuario como la IP y user agent solamente.

La estrategia utilizada es la que agrupa las sesiones por IP y user agent, con la identificación de una sesión basada en la diferencia de tiempo entre una acción o request a otra de no más de 30 minutos. Esta opción es satisfactoria, puesto que en general, no se aprecian dentro de los datos, las problemáticas de IP's multiusuarios, en general.

Una hipótesis importante por parte de la sesionización, es el hecho que la sesión posee acciones relevantes para el análisis y otras que no. Tal es el ejemplo de las acciones que llevan a la página del índice, o llevan a páginas de preparación de la acción de postear. Estas acciones, son parte de la estructura y protocolos de la aplicación Web, que no poseen contenido relevante para el usuario, sino que existen para ingresar la información necesaria antes de ver lo que realmente se está buscando. Por esa razón, en este trabajo se consideró realizar un post-proceso, que elimina estas acciones de la sesión, dejando aquellas que representen lo que el usuario realmente hace dentro del sitio.

El post-proceso filtra las acciones por las cuales no se puede definir un contenido específico relacionado a un post encontrado dentro de la base de datos. Las acciones que quedan luego de la sesionización para ser modeladas se encuentran en la Tabla 5.

Tabla 5: Acciones utilizadas en Plexilandia

URL	Método
/foro/viewtopic.php	GET
/foro/posting.php	POST

La razón de realizar este filtrado en este momento, es para que el proceso de sesionización, que se basa principalmente en el tiempo, no corte sesiones que en el caso de filtradas podrían tener sus acciones muy separadas en el tiempo.

4.5.- Transformación

La transformación de lo que se define en este trabajo como sesión se divide en dos partes: La transformación o representación vectorial de la secuencia de acciones realizadas por un usuario y la representación vectorial del contenido visto o revisado por el usuario con cada acción.

El total de sesiones obtenidas en el proceso de sesionización es de 72855. Pero el conjunto de sesiones válidas para el análisis, es decir, cuyos usuarios generen posts explícitos se reduce a 2613. El histograma de sesiones válidas para el análisis se puede apreciar en la Figura 8 y Tabla 6. De los largos se puede inferir que el largo promedio de sesiones es 7.7 páginas por sesión. Sin embargo, ignorar sesiones con largo menor a 7 significa una pérdida del 57% de los datos. Lo anterior se puede ver del histograma puesto que se nota que los datos se encuentran muy dispersos. Finalmente, se elige el largo de tamaño 5, con eso los vectores de menor largo representan un 35% y se incluyen sesiones de largo máximo 30 dejando un 1% fuera del análisis. No se elige un largo menor a 5, puesto que se intenta analizar comportamiento de usuario, y un largo menor dejaría de lado el concepto de análisis de secuencia que se quiere tener.

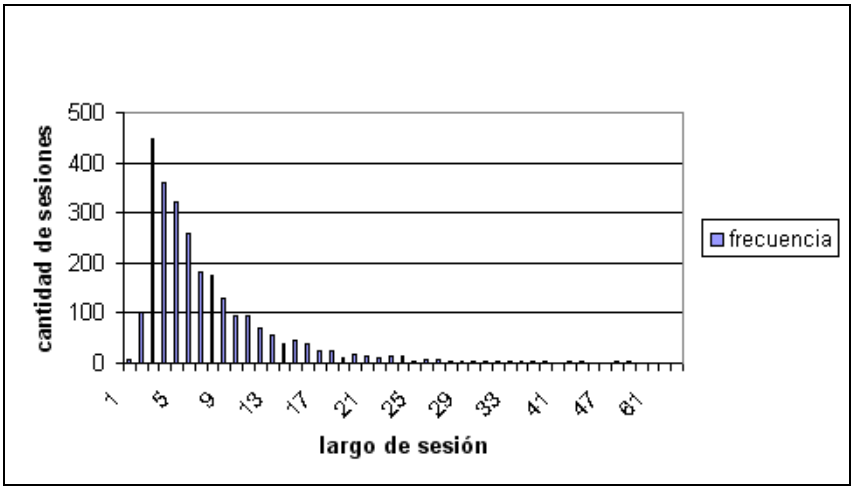

Figura 8: Histograma cantidad de sesiones
Fuente: Elaboración propia

Luego, de saber cuáles son estas sesiones, se indexan sus URL's de manera lexicográficamente, para mantener la noción de cercanía semántica de las acciones de manera localizada y que exista relación de vecindad entre los identificadores.

Tabla 6: Frecuencia sesiones con post

largo	sesiones	largo	sesiones
< 3	111	13	55
3	447	14	39
4	359	15	45
5	321	16	38
6	259	17	25
7	181	18	26
8	175	19	9
9	128	20	19
10	95	20 < x <30[8]	70
11	94	x > 30	37
12	71		

Seguido, se genera una representación vectorial de la sesión, definida como una secuencia de identificadores, verificando en el mismo momento si la sesión generó un posteo real dentro de la comunidad dentro de las 5 acciones representadas. Se requiere que dentro de las acciones haya al menos un post, puesto que de esta manera se intentan identificar aspectos relevantes de la sesión.

En este momento se genera el modelo de tiempos de las acciones de cada sesión, basándose en la diferencia entre tiempo del request y el tiempo del siguiente request. En el proceso de sesionización se registra el tiempo en que la siguiente acción comienza, como se muestra en la Tabla 7.

Tabla 7: Ejemplo tiempo entre acciones

IP	URL	tiempo inicial	tiempo final
146.83.193.31	/foro/viewtopic.php?t=8386	1245691698	1245691725
146.83.193.31	/foro/posting.php?ip=146.83.193.31&fechats=1245691757	1245691757	1245691762

Con la información se modela el tiempo dentro de la acción como la diferencia entre tiempo final y el tiempo inicial. No obstante, esto no se puede realizar para con la última acción de la secuencia. Si es necesario obtener esa información se utiliza el promedio de las acciones anteriores.

[8] 20 < x < 30: Representa que el largo es mayor estricto que 20 y menor estricto que 30

En el caso del contenido, se toman en cuenta aquellas que se encuentren en sesiones con post. Cada acción genera un modelo de contenido, que se basa en lo que el usuario pudo haber buscado con la acción. El contenido, como es dinámico dentro del foro, es modelado de la forma que se muestra en la Tabla 8.

Tabla 8: Modelo para la acción posting.php

URL	Descripción de acuerdo a la BD plexilandia
/foro/viewtopic.php?t=8171	El primer post del topic 8171
/foro/posting.php?ip=146.83.193.31&fechats=1245691757	El primer post desde 20 segundos después de fechats de la IP 146.83.193.31
/foro/posting.php?ip=146.83.193.31&fechats=1245691757&p=7800	Este caso es una edición del post 7800

La acción de postear, no es simple, sino que contempla muchos pasos. Un caso normal, contempla la necesidad de estar logueado, seguir con la acción de responder donde se despliega la información del post a que se está respondiendo, y una acción de confirmación. El modelo que se utilizó en este trabajo concentra todo este mecanismo en la acción de mandar al servidor la información en acción tipo POST. Esto asegura el hecho que efectivamente se posteó, y que no fue un intento sin éxito.

En este mismo contexto, dada la información existente dentro de los Web logs, no es posible saber el identificador en la base de datos de cuál es el post generado por el usuario. Para encontrar cual es el post del usuario de la sesión se realiza una búsqueda de posts donde tanto la IP del web log como la IP del post en la base de datos sean la misma. Además se requiere que el tiempo de creación del post sea mayor o igual que la fecha del request en el Web log menos una diferencia de tiempo, que para el caso de este trabajo es de 20 segundos. Esta diferencia viene a razón de la posible desincronización entre php y el servidor MySQL.

Es así como cada identificador de acción se asocia con un identificador de post, muchos identificadores de acción pueden asociarse con un mismo identificador de post, debido a que sea una edición de post, o es una observación, que para el modelo se considera una acción distinta. Es por esta razón que para poder generar el TFIDF de estos documentos, no se hace con el identificador de acción sino con los identificadores de post que son únicos.

Luego del generar los vectores de TFIDF, se requiere la creación de los vectores de Conceptos del modelo Concept-based y de Tópicos del LDA.

La información que permiten generar estas tablas, estaban previamente calculadas por trabajos anteriores, por lo que simplemente se utilizan para, mediante la matemática correspondiente generar las representaciones.

4.6.- Algorítmos de clustering

Para el análisis en este trabajo se utilizaron dos algoritmos de clustering de partición conocidos, las redes de Kohonen o SOM y K-medoids.

El orden de uso es relevante, puesto que SOM se utiliza más que para generar un mejor clustering, para obtener la cantidad de clusters que existen dentro del universo de datos, valor que K-medoids recibe como dato.

La implementación de ambos algoritmos se basó principalmente en la librería opensource JavaML[9]. Esta librería consta de una gran cantidad de herramientas orientadas a realizar minería de datos. Sin embargo, no consta con facilidades de generalización de algoritmos, por lo que se modificaron para poder ser utilizados en este trabajo. Específicamente, se modificó el algoritmo de SOM, representando la sesión con todos los atributos necesarios para las distancias, y el K-medoids, para que usara una nueva forma de cálculo del elemento mediana.

SOM

La implementación de SOM debe generalizarse para el caso en que las neuronas, como prototipo de sesión, sean modeladas de la misma manera que las sesiones del experimento.

La neurona prototipo debe constar con aquellas características que la medida de similitud o disimilitud requiera. En el caso del trabajo, constan de vectores que representan la secuencia de identificadores, la secuencia de tiempos de la sesión, el valor del largo de la sesión, la secuencia de categorías de la comunidad en la sesión y la secuencia de vectores característicos del contenido de la sesión.

[9] http://java-ml.sourceforge.net/

La librería JavaML tiene implementados las funciones de vecindad circular y gaussiana y funciones de propagación, exponencial, lineal e inversa. Además, es importante fijar el radio de aprendizaje, el radio inicia, la forma de la grilla y su tamaño.

K-medoids

En este trabajo se propone el uso del K-medoids en el cual se usó la mediana o elemento cuyas distancias hacia los otros elementos del conjunto es la menor, de esta manera se manejan aspectos como los outliers de mejor manera.

El gran problema de este algoritmo en cuanto a tiempo, es que se tarda en calcular las distancias entre elementos. Esto se debe hacer para asignar cada elemento a un cluster y para poder calcular la mediana. En este trabajo, dada la cantidad de elementos con que se trabajó, se decidió pre-calcular las distancias, dejándolas guardadas con anterioridad en el sistema. La decisión de guardar las distancias de manera previa, genera una gran ventaja al momento del análisis, puesto que este cálculo no se requiere hacer nuevamente, y el algoritmo puede ser repetido una cantidad representativa de veces para encontrar el mejor resultado.

4.7.- Modelo de datos y diagrama de clases del sistema propuesto

En la Figura 9 se muestra el modelo de datos utilizado en todo el proceso. En él se encuentran las tablas más importantes que permiten mantener dentro del sistema la información de las sesiones y su contenido. Es relevante en este punto, clarificar que en la figura se muestra una instancia de la base de datos, en donde el usuario ha elegido que la sesión se represente por un máximo de cinco acciones. Esto es dado ya que a medida que se va avanzando en el proceso, las tablas de representación de las sesiones se van creando dinámicamente de acuerdo a las decisiones del usuario analista.

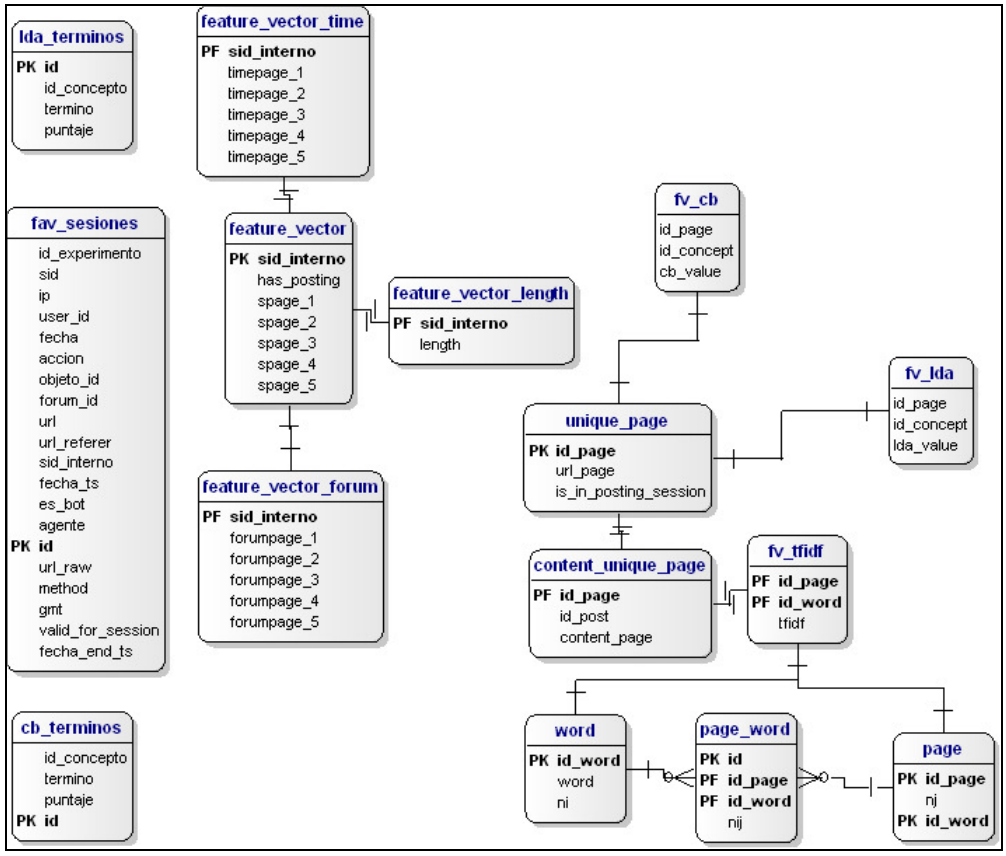

Figura 9: Esquema base de datos de Generador de Sesiones y Clustering
Fuente: elaboración propia

Se sigue con la descripción de cada tabla:

- fav_sesiones: es la tabla en la que se guarda el detalle de cada línea válida luego del proceso de selección y pre-procesamiento de los web log de apache. Sobre esta tabla se lleva el proceso de sesionización.
- feature_vector: es la tabla modelo para guardar la secuencia que se basa en una secuencia de números enteros procesados en la tabla unique_page.
- feature_vector_time: al momento de generar los vectores característicos se guarda el modelo del tiempo de la misma manera.
- feature_length: se guarda el largo de la sesión original, sirve cuando es truncada.
- feature_forum: modela la clasificación de los post en el foro de Plexilandia.
- cb_terminos y lda_terminos: son las tablas que guardan los modelos de Concept-based y LDA obtenidos de trabajos anteriores.
- unique_page: guarda el valor de únicas URL en el sistema.
- content_unique_page: guarda el match entre unique_page y el post asociado y su contenido.
- fv_tfidf: guarda el modelo tfidf para el contenido.

- word: guarda las palabras y la cantidad de elementos que las contienen
- page: guarda la información de cuantas palabras tiene un post.
- page_word: guarda la relacion N-N entre word y page.
- fv_cb: el contenido en representación del modelo Concept-based.
- fv_lda: el contenido en representación del modelo LDA.

La implementación de la base de datos, para este trabajo se hizo sobre postgresql v8.4.

El proyecto fue programado en Java, funcionando en la versión jdk1.6.0_13, y se divide en dos proyectos. Uno de ellos se llama AnalizadorSesiones e implementa el proceso KDD para este trabajo. Se muestra en la Figura 10 el diagrama de clases de este proyecto sólo con las más importantes clases. Se utilizaron librerías para la conexión de base de datos, para MySQL, el conector MySQL-java v5.0.7 y jdbc de postgresql v8.2-505. Y para ciertos algoritmos, la librería Snowball para el stemming en español.

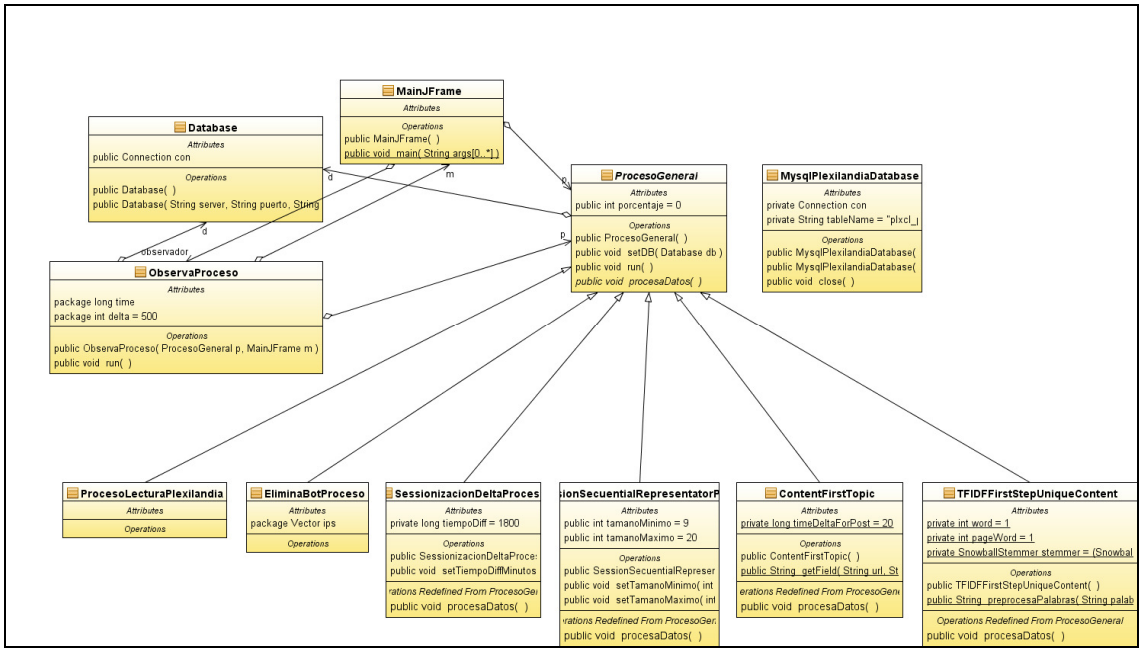

Figura 10: Diagrama de clases Generador Sesiones
Fuente: elaboración propia

Se sigue con la descripción de cada clase:

- MainJFrame.java: es la interfaz gráfica de la aplicación, es la encargada de mostrar los controladores necesarios para poder realizar cada paso del proceso, y mientras los algoritmos corren, muestra el avance en una barra de progreso.

- Database.java: es la interfaz que permite la conexión con la base de datos en todo el proceso.
- ObservaProceso.java: permite ir mostrando el avance del algoritmo, actualizando los valores mostrados en el Frame constantemente. Corresponde a lo que en programación orientada a objetos se conoce como el patrón Observer.
- ProcesoGeneral.java: es el proceso observado, de él heredan los algoritmos.
- MysqlPlexilandiaDatabase.java: es la interfaz que permite hacer todas las operaciones con la base de datos de Plexilandia.
- ProcesoLecturaPlexilandia.java: lee el web log de plexilandia, selecciona y preprocesa.
- EliminaBotProceso.java: indaga en la eliminación de bots.
- SessionizationDeltaProcess.java: sesioniza usando la estrategia reactiva temporal.
- SessionSecuentialRepresentation,java: genera los vectores característicos de la sesión y sus características afines.
- ContentFirstTopic: llena la tabla content_unique_page con la información de plexilandia, usa la heurística que ver una discusión implica ver el primer post de ella.
- TFIDFFirstUniqueStep: genera el TFIDF del contenido modelado.

La cantidad de procesos, es mayor a la mostrada en el diagrama, y por espacio se decidió omitirlos. El concepto de proceso permite ordenar dentro de la aplicación el orden en que se deben ejecutar los algoritmos.

Respecto a las relaciones en el modelo de clases, esas se distribuyen en los siguientes tipos:

- Herencia del ProcesoGeneral: cada subsección del trabajo se implementa en un proceso, que suele utilizar uno o más algoritmos. Este proceso, cuya función principal heredada es procesaDatos, utiliza la clase encargada de proveer la conexión con la base de datos y además va indicando en la variable porcentaje el avance de la ejecución. El paso de parámetros se basa en métodos del tipo “set” y “get”.
- Comunicación con interfaz: la clase que genera la interfaz con el usuario se denota MainJFrame, y es la encargada de permitir el ingreso de los parámetros y mostrar ciertos resultados dados por los procesos. Es la que utiliza al

observador, con la intención que se vaya mostrando el avance de los procesos y así le permitirle al usuario estimar el tiempo que queda en la ejecución.

- Observer: la clase ObservaProceso, tiene la intención de apoyar la interfaz del usuario, ya que los procesos suelen tomar varios minutos en ejecutar.

Se pueden ver imágenes de la interfaz de la aplicación en la sección de Apéndices C.

El segundo proyecto es una adaptación de la librería Java ML nombrada en la sección 4.6. Esta librería de código libre, está diseñada para el manejo de datos, generalmente representado en vectores característicos. Sin embargo, en este trabajo se utiliza el concepto de sesión de usuario que posee una representación más compleja y por ende se debe implementar el algoritmo de SOM para que vaya generando el aprendizaje a cada característica de la sesión de usuario modelada. Luego, la solución propuesta, considera la implementación de SOM y K-medoids, con las modificaciones pertinentes.

La estructura de la modificación de la librería se puede apreciar en la Figura 11. En ella se pueden ver aquellas clases utilizadas dentro del contexto de los algoritmos implementados.

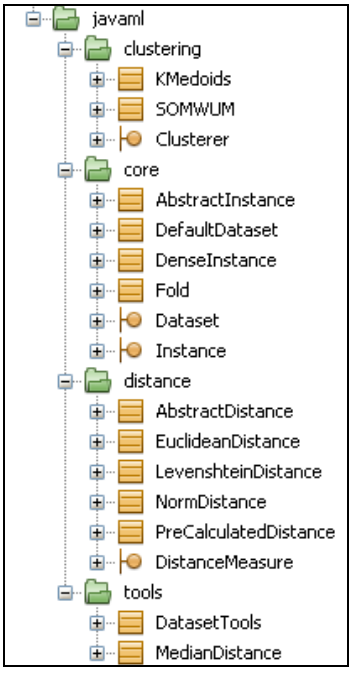

Figura 11: Estructura JavaML en la aplicación
Fuente: Elaboración propia

Se sigue con la descripción general de las clases:

- En el package clustering: Kmedoids y SOMWUM, son clases que generan el modelo de clustering. Están modificados de la versión original para soportar las características de las sesiones. Para el caso del SOMWUM, el algoritmo es favorecido implementando sus subprocesos de manera paralela con Threads, pudiendo tomar menos tiempo que siendo secuencial.
- En el package core: Instance, representa la sesión, y mantiene en su representación las características de su identificador y la secuencia de acciones.
- En el package distance: se agregan ciertas distancias para el trabajo, como la distancia de edición y una interfaz para utilizar distancias precalculadas existentes en la base de datos.
- En el package tools: se encuentran herramientas para encontrar la mediana, entre otras.

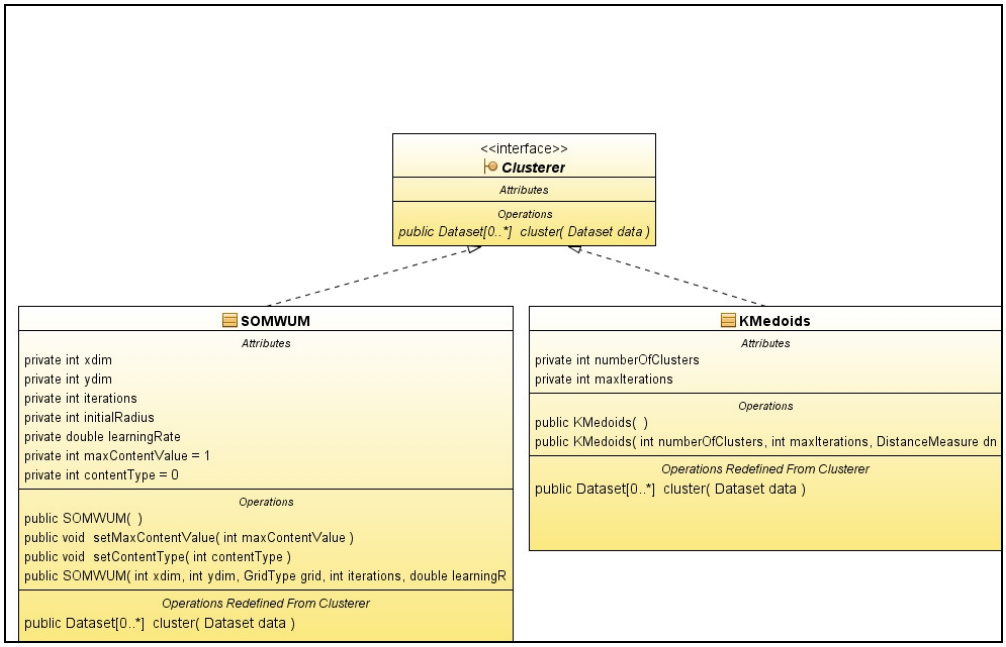

Figura 12: Diagrama de clases de implementación algoritmos de clustering
Elaboración propia

El caso de las clases que implementan los algoritmos requiere un poco más de atención, puesto que requieren interactuar con los datos de las sesiones y por ende utilizar la base de datos. Además se requiere traspasar la dimensionalidad de las representaciones de la secuencia de acciones de la sesión como la del texto. El diagrama de clases de la Figura 12 permite ver la interfaz pública de la implementación de los algoritmos. Basándose en el diseño que tiene la librería original, se aprecia que los algoritmos heredan de una interfaz Clusterer,

que tiene un método llamado cluster, el cual entrega un arreglo de Dataset, que tiene el resultado del clustering.

4.8.- Aplicación de los algoritmos

Para cada modelo propuesto en el diseño se realizaron los siguientes pasos:

1. Se ejecuta el algoritmo SOM, con la intención de obtener un modelo de clustering que permita inferir la cantidad de grupos naturales en los datos respecto a la distancia analizada.
2. Se realiza un análisis del número de clusters, concluyendo en la cantidad de clusters reales de los datos, basándose en un porcentaje de confianza.
3. Se ejecuta K-medoids, una cierta cantidad de veces, con el k obtenido con anterioridad.
4. Se selecciona el resultado del K-medoids que tenga el mejor indicador de cohesión-separación posible.

Para el algoritmo SOM, las variables de entrada de la ejecución de todos los experimentos se encuentran en la Tabla 9.

Tabla 9 Parámetros del algoritmo SOM

Parámetro	Valor
Tamaño de grilla	6x6
Tipo de grilla	Cuadricular
Número de iteraciones	1000
fracción de aprendizaje	0,1
radio inicial	8
función de aprendizaje	lineal
función de vecindad	gaussiana

De los resultados del SOM, se consideran no clusters a los grupos obtenidos tales que al estar ordenados de menor cantidad de elementos a mayor, los que no captan 5% del total de los datos, esto mientras la cantidad agregada no supere el 15% del total de elementos. Luego es posible encontrar la cantidad de clusters que serán utilizados para el K-medoids.

Se ejecuta K-medoids 5 veces por modelo, eligiendo el que tenga mejor índice cohesión-separación.

4.9.- Ejecución y resultados

Para la presentación de resultados, se deciden utilizar los modelos de distancias usadas, teniendo en cuenta que cada distancia es analizada con dos representaciones de textos independientes explicadas en la sección 2.6.- Representaciones en text mining.

Modelo 1

Enfoque texto representado Concept – based

En el caso de la representación del contenido con el enfoque basado en conceptos (Concept-based) para el algoritmo SOM se obtiene un modelo de cluster con distribución de frecuencias como se muestra en la Figura 13.

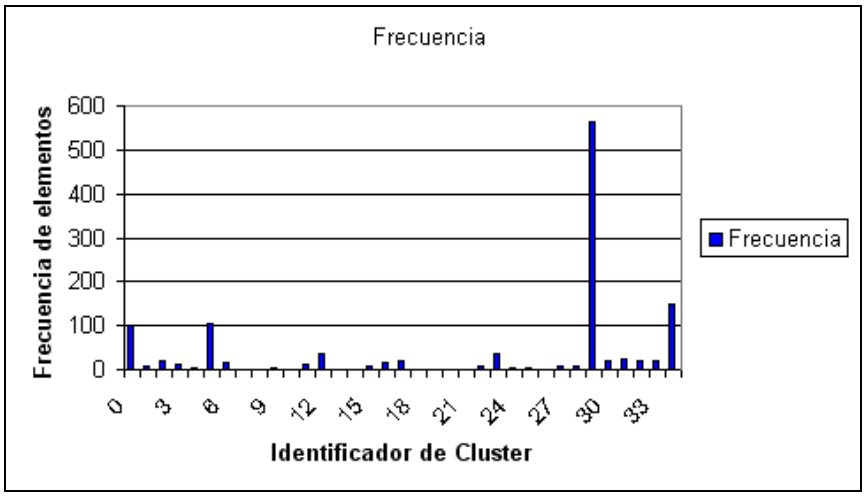

Figura 13: Histograma resultado SOM
Fuente: Elaboración propia

El tamaño de la grilla del algoritmo SOM utilizado fue de 6x6, es decir, 36 neuronas prototipo que permiten la creación de a lo más 36 clusters. De la cantidad de clusters de la Tabla 10, se observan 35 clusters alcanzados, con un promedio de 35,7 elementos por cluster y

una desviación estándar de 97,6. Se aprecia dentro del histograma que existen al menos 4 clusters con cantidades predominantes. No obstante, se verifica que existen muchos clusters con cantidades cercanas al promedio calculado, viendo el histograma y la desviación estándar, esto quiere decir que estos clusters no se pueden ignorar estadísticamente.

Tabla 10: Frecuencias SOM

ID Cluster	Cantidad	ID Cluster	Cantidad	ID Cluster	Cantidad
0	101	15	7	30	21
1	9	16	15	31	24
2	19	17	20	32	21
3	14	18	2	33	19
4	6	19	1	34	151
5	103	20	2		
6	17	21	1		
7	2	22	10		
8	1	23	36		
9	4	24	5		
10	1	25	6		
11	11	26	2		
12	37	27	7		
13	2	28	8		
14	1	29	564		

Lo anterior se refleja fidedignamente al indagar en la cantidad a utilizar de los clusters mediante la restricción propuesta en la sección 4.8, es decir, ignorar la clasificación dada por el algoritmo en no más del 15% de los elementos. Como resultado final, se obtienen 10 clusters significativos los que se muestran en la Tabla 11.

Tabla 11: Resultados SOM procesados

ID Cluster	Cantidad
29	564
34	151
5	103
0	101
12	37
23	36
31	24
30	21
32	21
17	20

Consecutivamente se lleva a cabo el experimento con el algoritmo K-medoids con cantidad de clusters igual a 10. Explicado anteriormente, se realizan 5 experimentos, y se calcula el índice cohesión-separación, eligiendo para este modelo el que posee el valor máximo (ya es una medida de similitud). Las estadísticas se pueden apreciar en el histograma, y en la tabla de cantidades de la Figura 14 y la

Tabla 12, respectivamente. Donde la cantidad de elementos posee una media de 125 y una desviación de 51,7. Además, el mejor índice encontrado de cohesión-separación obtenido es de 0,3.

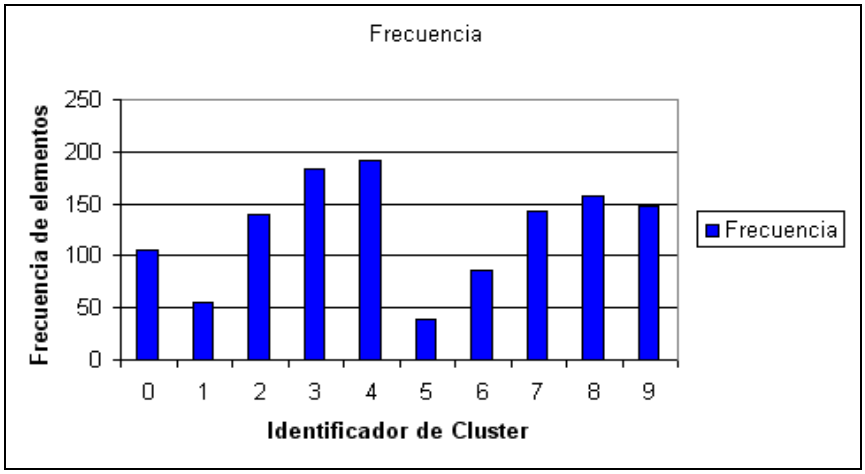

Figura 14: Histograma resultados K-medoids
Fuente: elaboración propia

Tabla 12 Resultados K-medoids

ID Cluster	Cantidad
0	106
1	56
2	139
3	184
4	192
5	39
6	86
7	143
8	158
9	147

Enfoque texto representado con LDA

Como en el caso anterior, los resultados del algoritmo SOM se presentan en un histograma de frecuencias por cluster, en la Figura 15.

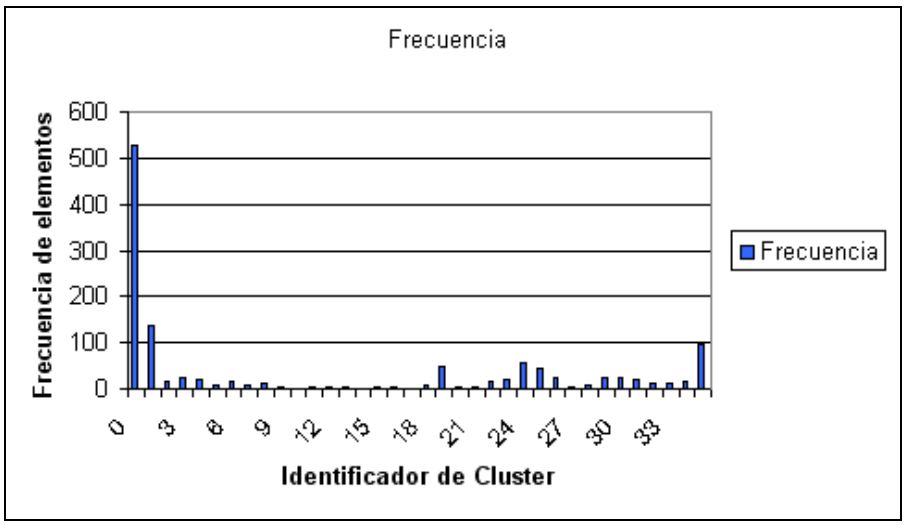

Figura 15: Histograma resultados SOM
Fuente: Elaboración propia

Tabla 13: Frecuencias resultados SOM

ID Cluster	Cantidad	ID Cluster	Cantidad	ID Cluster	Cantidad
0	526	15	5	30	23
1	135	16	3	31	20
2	16	17	2	32	11
3	26	18	10	33	12
4	21	19	49	34	16
5	9	20	6	35	98
6	15	21	4		
7	7	22	15		
8	14	23	21		
9	4	24	57		
10	2	25	43		
11	3	26	24		
12	5	27	6		
13	6	28	10		
14	2	29	24		

De la Tabla 13, se observa que el modelo SOM utiliza toda la grilla para poder conformar los clusters, es decir, 36. El promedio de cantidad por cluster es 34.7 y la desviación estándar de 88.5. Del histograma de la Figura 15, se aprecian a la vista 6 cluster predominantes, pero tal como en el caso del enfoque Concept-Based el promedio bajo y la desviación estándar no superando 100, y como existen clusters con cantidades mucho más grandes, implica que los clusters pequeños no pueden ser ignorados.

Al utilizar la premisa, que el número de clusters significativos son aquellos tales que no se dejen fuera más del 15% del total de elementos, quedan 13 clusters que se visualizan en la Tabla 14.

Tabla 14: Resultados SOM procesados

ID Cluster	Cantidad
0	526
1	135
35	98
24	57
19	49
25	43
3	26
29	24
26	24
30	23
23	21
4	21
31	20

En el caso del uso de K-medoids, se realizó lo mismo que en el enfoque anterior. El histograma y detalle se pueden ver en la Figura 16 y Tabla 15 respectivamente. Cabe destacar que el promedio de elementos en el modelo es 96,15 y la desviación estándar de 34,35. El mejor índice cohesión-separación alcanzado en los experimentos es de 0,2.

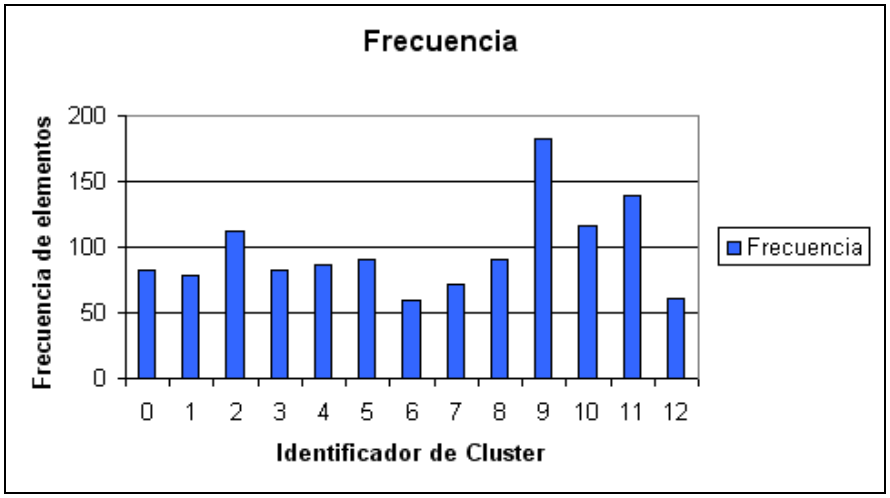

Figura 16: Histograma K-medoids
Fuente: Elaboración propia

Tabla 15 Frecuencias K-medoids

ID Cluster	Cantidad
0	82
1	78
2	112
3	82
4	87
5	90
6	59
7	71
8	90
9	183
10	116
11	139
12	61

Modelo 2

Enfoque texto representado Concept – based

Del experimento del algoritmo SOM utilizando el modelo 2, con representación de texto utilizando Concept-based, los resultados se aprecian en el histograma de la Figura 17.

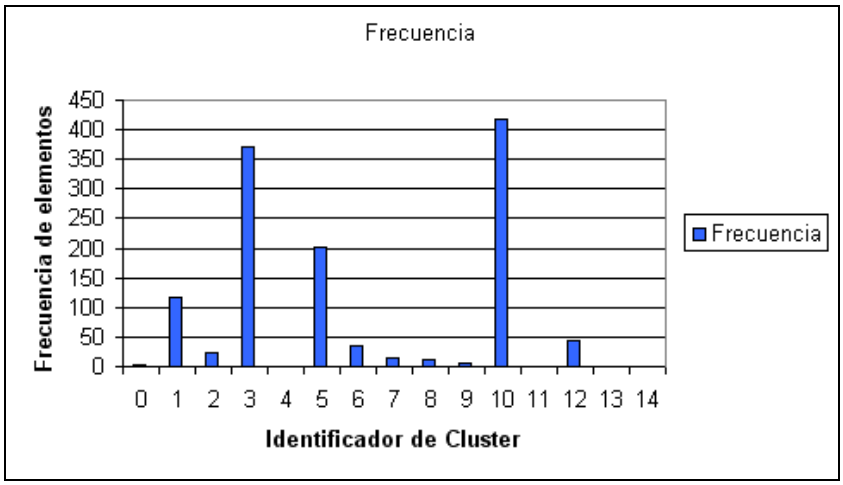

Figura 17: Histograma resultados SOM
Fuente: Elaboración propia

Tabla 16: Frecuencia resultados SOM

ID Cluster	Cantidad	ID Cluster	Cantidad
0	2	8	13
1	117	9	6
2	23	10	418
3	371	11	1
4	1	12	43
5	203	13	1
6	34	14	1
7	16		

De la Tabla 16 se puede apreciar que el algoritmo SOM termina agrupando las sesiones solamente en 15 clusters. El promedio de cantidad por cluster es de 83,3 elementos y la desviación estándar de 138,12. La desviación estándar permite entender que los cluster están bastante dispersos respecto a la cantidad, y los elementos predominantes si son significativos. Al analizar por simple inspección se observa la existencia de 3 a 4 clusters. Recurriendo al filtro de clusters basado en el 15% de elementos no significativos quedan finalmente 4 clusters indicados en la Tabla 17.

Tabla 17: Resultados SOM procesados

ID Cluster	Cantidad
10	418
3	371
5	203
1	117

Utilizando el hecho que SOM encontró que el número de clusters era 4, el resultado obtenido por K-medoids se representa en un histograma y cantidades en la Figura 18 y Tabla 18, respectivamente. Obteniendo un promedio de elementos de 312,5 y una desviación estándar de 158,6. El índice, que en este caso debe ser el mínimo entre los experimentos, es de 0,4.

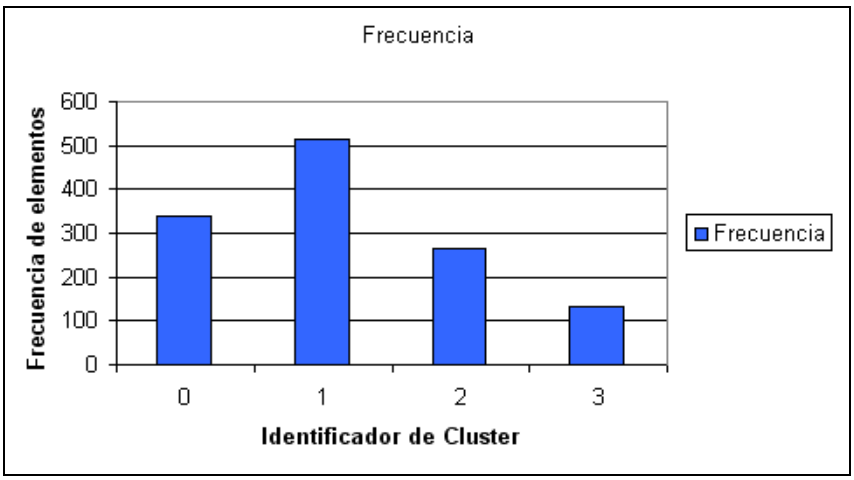

Figura 18: Histograma K-medoids
Fuente: Elaboración propia

Tabla 18: Frecuencias resultado K-medoids

ID Cluster	Cantidad
0	340
1	513
2	264
3	133

Enfoque texto representado con LDA

Del algoritmo SOM, se obtienen los resultados visibles dentro del histograma de cantidades de elementos en la Figura 19.

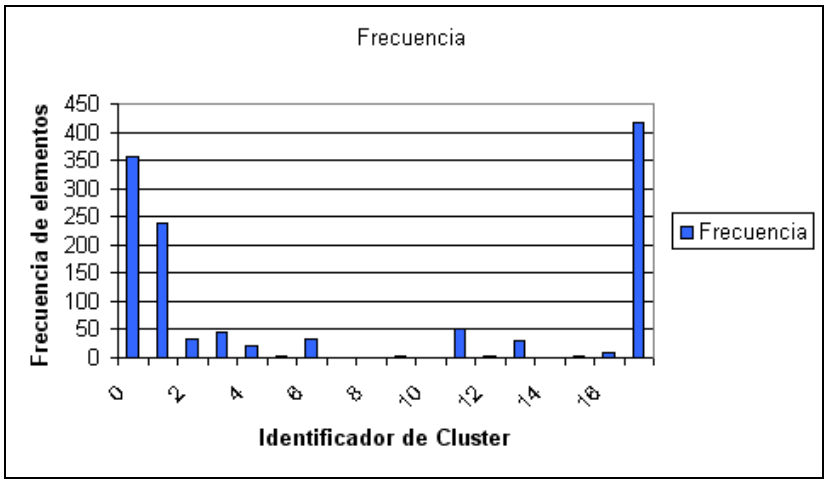

Figura 19: Histograma resultados SOM
Fuente: Elaboración propia

Tabla 19: Frecuencias resultados SOM

ID Cluster	Cantidad	ID Cluster	Cantidad
0	355	9	4
1	238	10	1
2	34	11	51
3	46	12	2
4	20	13	30
5	2	14	1
6	34	15	4
7	1	16	8
8	1	17	418

De la Tabla 19, se observa que el algoritmo logra agrupar los elementos en 18 clusters de un total de 36 posibles. La cantidad promedio es 69,4 y la desviación estándar es 128,1. Al igual que en el enfoque de la representación Concept-based, la desviación estándar indica que la diferencia entre los clusters más pequeños y los más grandes es sustancial, por lo que son representativos. De la Figura 19, se aprecia que al menos 4 clusters son importantes en cantidad. Utilizando la metodología para encontrar la cantidad de clusters importantes, dejando fuera sólo al 15%, la cantidad de clusters termina siendo 5, indicados en la Tabla 20.

Tabla 20: Resultados SOM procesados

ID cluster	Cantidad
17	418
0	355
1	238
11	51
3	46

Utilizando el resultado del SOM, los resultados de K-medoids se expresan en el histograma y tabla de cantidades, en la Figura 20 y Tabla 21, respectivamente. El promedio de cantidades termina siendo de 250 y la desviación estándar de 75,5. El mejor índice de cohesión-separación para este modelo fue de 0,11.

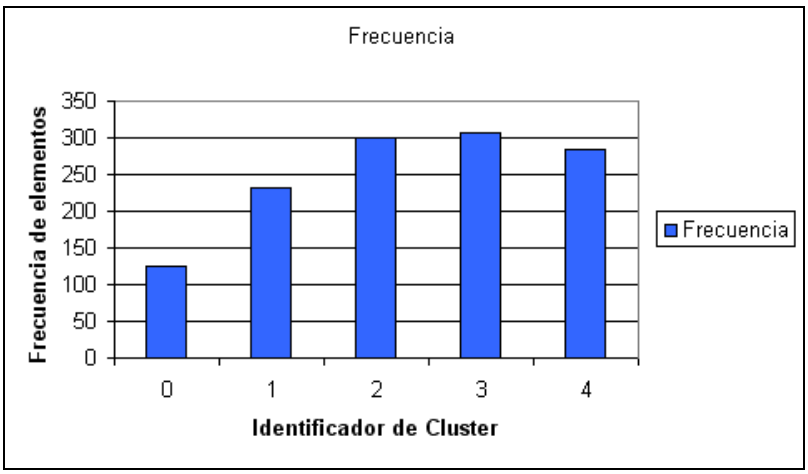

Figura 20: Histograma resultados K-medoids
Fuente: Elaboración propia

Tabla 21: Frecuencias K-medoids

ID Cluster	Cantidad
0	126
1	231
2	301
3	307
4	285

5.- Análisis de Resultados

Dado que los resultados de este trabajo son un modelo de clusters de datos de sesiones de usuario, en una comunidad virtual de práctica en la Web, el análisis debe ser realizado sobre las variables más importantes dentro del contexto de las técnicas del Web usage mining y Web content mining. En síntesis, las características que definen la navegación de usuario en este trabajo son la secuencia de acciones tomadas por el usuario en conjunto con su contenido.

Sin embargo, el objetivo principal de este trabajo es encontrar dentro de los comportamientos de usuario indicios que permitan utilizar la información obtenida luego de generar un modelo de clustering con la intención de identificar mensajes que deben ser revisados por los moderadores con una mayor probabilidad. Por lo tanto, se propone buscar aspectos para la moderación dentro de la secuencia como el contenido en el comportamiento de usuario analizado.

Cabe destacar que la importancia a cada mensaje, en el contexto de la moderación, se la dan los moderadores y los administradores de la comunidad, por lo que este aspecto debe ser considerado en el análisis.

5.1.- Metodología de análisis y evaluación

La minería de datos, termina resumiéndose en el uso de distintos algoritmos que intentan encontrar patrones o información nueva en el conjunto de datos. Los algoritmos utilizados suelen generar resultados que no son fáciles de comprender y generalmente requieren de expertos que puedan interpretarlos y validarlos, y así poder ser utilizados.

En este trabajo, los datos utilizados, en síntesis, son comportamientos de usuarios representados por las sesiones de usuarios dentro del sistema, pero se les relaciona directamente a las acciones de moderación y la necesidad de revisar lo que un comportamiento produce para la comunidad.

En búsqueda de entender la causalidad que existe entre el comportamiento de usuario, la información que se comparte, y la necesidad de revisar y moderar tales mensajes se propone analizar los resultados del clustering a partir de un cómo (basado en la secuencia y el contenido) y en un qué (basado principalmente en las características de los mensajes

generados por el usuario en la sesión). Se propone que este análisis sea realizado a partir de las premisas rescatadas de los expertos generales del trabajo, el uso del conocimiento que proveen las representaciones del texto y las características que se consideran importantes al momento de moderar, sin la necesidad de incluir más información supervisada.

El análisis por parte de la secuencia y contenido, tiene la intención de reconocer aquellos clusters de comportamiento de usuario que sean más propensos a generar mensajes que posteriormente deben ser moderados, o al menos, revisados. La forma de realizar el análisis se concentra principalmente en la rapidez del usuario en generar el mensaje, puesto que un usuario que lo genera al principio de la navegación, puede estar haciéndolo sin la debida conciencia. Sin embargo, es necesario correlacionar, con la ayuda de las técnicas de representación de texto elegidas, la secuencia con su contenido principal.

El análisis por características de los mensajes de las sesiones, se basa principalmente en aquellas cualidades que tienen las conversaciones y que los moderadores consideran importantes al momento de revisar y moderar. Estas características se encuentran en la sección 3.1.2. Para este análisis se propone generar una cuantificación, dentro de cada cluster, para cada comportamiento de usuario, y poder decidir cuál de estos clusters debiera ser moderado. En este momento, cada sesión de cada cluster se clasifica que necesita moderación o no, bajo cierta característica. Con las fracciones de elementos que se consideren así, se estima la probabilidad de cada cluster respecto a cada característica. Se propone que los clusters se pueden ordenar respecto al grado de moderación basado en la suma de estas probabilidades.

La conjetura de los análisis anteriormente descritos, tiene la intención de proveer de la información importante al analista para poder decidir cuales comportamientos de usuario dentro de la comunidad tienen alta inferencia dentro de la moderación. Sin embargo, la intuición y la estadística deben ser evaluadas respecto a lo que un experto realizaría de manera supervisada. Para esto, se propone utilizar una clasificación de sesiones obtenida directamente de un moderador y contrastarla con los resultados del análisis propuesto para este trabajo.

La evaluación de este trabajo se realiza a partir de conceptos conocidos como precisión, recall y F-measure, explicados en la sección 2.5.4. Para estas definiciones, se debe medir la capacidad del modelo de predecir la necesidad de moderación de la manera más correcta posible, vale decir, que el modelo debe encontrar la mayor cantidad de mensajes que requieran moderación. Además, la intención de usar estas medidas es poder comparar los resultados de los distintos modelos propuestos.

Todo el proceso del análisis y evaluación propuestos en este trabajo se puede observar en la Figura 21. Esta metodología se realiza para cada uno de los modelos, permitiendo la comparación en la etapa de evaluación mediante el uso de herramientas como precisión, recall y F-measure.

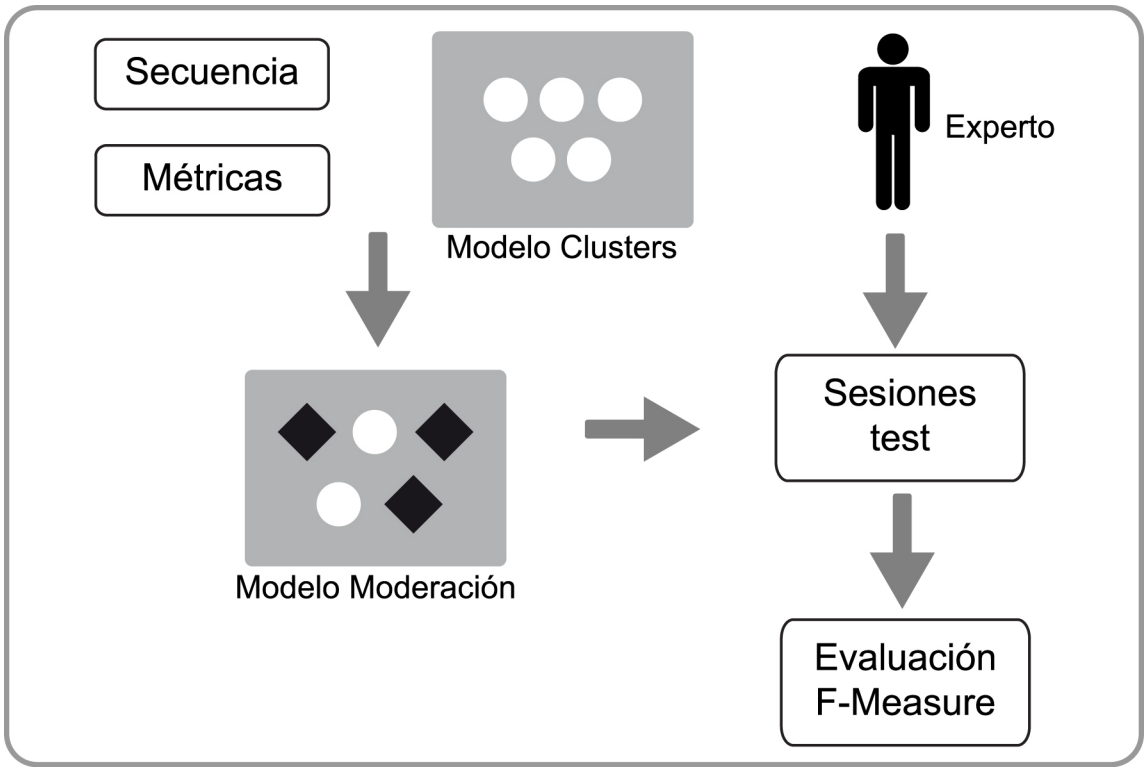

Figura 21: Metodología de evaluación
Fuente: Elaboración propia

5.1.- Análisis de secuencia y contenido

Para analizar cada cluster, en el sentido de comportamiento de usuario, se eligen como características más importantes la secuencia de eventos y el contenido navegado por el usuario en una sesión.

El análisis se llevará a cabo a cada cluster de cada modelo y seguirá la siguiente metodología:

1. Identificar la sesión mediana de cada cluster.
2. Encontrar los vecinos más cercanos de la mediana.

3. Analizar la secuencia que aparece con mayor frecuencia dentro de los elementos más internos del cluster, y reconocerla como la secuencia característica del cluster.
4. Analizar la representación del contenido, y según los conceptos o tópicos de la representación del texto (CB o LDA, según corresponda) más relevantes describir la secuencia de contenido.

Utilizando 35 vecinos más cercanos para todos los modelos los resultados del análisis, y se recuerda que por construcción de la solución para la comunidad de Plexilandia, las secuencias se constituyen solamente de dos acciones que se resumen en los conceptos de ver y generar un mensaje (post). Los conceptos o tópicos que se generan por las metodologías concept-based y LDA, se encuentran en la sección de Apéndices A.

Modelo 1

Se diferencian los resultados para las distintas representaciones del texto.

Enfoque Concept-based

El resultado para este modelo y representación consta de 10 clusters. Luego el análisis de secuencia y contenido de cada cluster se muestran en la Tabla 22.

Tabla 22: Secuencia y contenido Modelo 1 CB

Cluster	Secuencia (s)	Contenido
Cluster 0	Ver, Ver, Ver, Ver, Post	Contenido: amplificadores, amplificadores a tubo y efectos.
Cluster 1	Ver, Post, Ver, Ver, Ver	Amplificadores y amplificadores a tubo.
Cluster 2	Ver, Ver, Post, Ver, Ver Ver, Ver, Post, Ver, Post	Amplificadores y preguntas de inicios.
Cluster 3	Ver, Ver, Ver, Post, Ver	Amplificadores.
Cluster 4	Ver, Post, Ver, Ver, Ver	Amplificadores y preguntas de inicios.
Cluster 5	Ver, Ver, Post, Ver, Ver	Amplificadores, Amplificadores de tubo, efectos e pregunta de inicios.
Cluster 6	Ver, Ver, Ver, Ver, Post	Amplificadores.
Cluster 7	Ver, Ver, Post, Ver, Ver	Amplificadores y efectos.
Cluster 8	Ver, Ver, Ver, Post, Ver Ver, Post, Ver, Post, Ver	Amplificadores y efectos.
Cluster 9	Ver, Post, Ver, Ver, Ver	Amplificadores y efectos.

El análisis sobre este modelo y representación de texto permite ver que se encuentran 5 combinaciones de secuencias principales, que se distribuyen entre los distintos clusters, generalmente con cambios de contenidos. Sin embargo, las secuencias por clusters no necesariamente son únicas, lo que es complejo de entender puesto que la medida de similitud para este modelo se basa en acercar a aquellas que tienen una secuencia parecida, aunque la similitud en estos casos sigue siendo alta ya que no pasa de ser sólo una diferencia en la secuencia. Esta cercanía entre secuencias diferentes debe darse a razón del contenido o del tiempo, que también son características de la medida de este modelo.

El otro aspecto relevante al análisis es lo complicado que es analizar el contenido con los vectores característicos del concept-based en este trabajo, y la razón puede encontrarse en la poca cantidad de conceptos que han sido capturados o que existe un sesgo dentro de los mensajes de esta comunidad que tiende a darle la mayor relevancia a cierto grupo pequeño de conceptos. Sin embargo, se puede ver el hecho que existen clusters donde predominan conceptos específicos de la comunidad, como los relacionados con amplificación, y otros donde aparte se tratan de preguntas sobre inicios en la comunidad.

Enfoque LDA

El resultado para esta representación consta de 13 clusters, el análisis se muestra en la Tabla 23.

En este modelo y representación de texto se identifican 6 tipos de secuencias de sesiones de usuarios significativas, donde se observa lo mismo que en el análisis de este modelo con representación concept-based, existen clusters con una mezcla de secuencias representantes. Como en el caso anterior, esto se puede dar por el uso del tiempo de navegación en la medida de similitud usada.

En el caso de la representación de LDA para este trabajo, la declaración de tópicos que representan un documento genera una más variada definición del contenido, por lo que el análisis puede decir cosas más contundentes. Por ejemplo, dentro de los clusters, se encuentran algunos que contienen contenido irrelevante, al menos para el experto, por lo que tienen mayor probabilidad a representar mensajes en los cuales se debe moderar o al menos leer con una mayor prioridad.

Tabla 23: Secuencia y contenido modelo 1 LDA

Cluster	Secuencia (s)	Contenido
Cluster 0	Ver, Post, Ver, Ver, Ver	Cajas y diferencias de sonido.
Cluster 1	Ver, Post, Ver, Ver, Post	Felicitaciones varias y cosas irrelevantes.
Cluster 2	Ver, Post, Ver, Ver, Ver	Técnicas de soldado, acople de efectos, felicitaciones y cosas irrelevantes.
Cluster 3	Ver, Post, Ver, Ver, Ver	Felicitaciones y cosas irrelevantes.
Cluster 4	Ver, Ver, Ver, Ver, Post	Apreciación sonido de bandas, transformadores amplificadores de tubo.
Cluster 5	Ver, Ver, Ver, Post, Ver	Transformadores amplificaciones a tubo.
Cluster 6	Ver, Post, Ver, Ver, Ver	Transformadores amplificaciones a tubo.
Cluster 7	Ver, Ver, Post, Ver, Ver	Contenido Irrelevante.
Cluster 8	Ver, Ver, Post, Ver, Ver	Conexión y construcción de cajas.
Cluster 9	Ver, Ver, Ver, Post, Ver Ver, Post, Ver, Post, Ver	Cajas para efectos y chasis para amplificadores.
Cluster 10	Ver, Ver, Ver, Post, Ver Ver, Post, Ver, Post, Ver	Problema de acople en efectos de distorsión.
Cluster 11	Ver, Ver, Post, Ver, Ver	Contenido irrelevante.
Cluster 12	Ver, Post, Ver, Ver, Ver	Apreciación sonido bandas y felicitaciones varias.

Al momento de hacer un análisis conjunto, entre la secuencia y el contenido, se encuentra que generalmente los clusters que tienen mayor información irrelevante tienen una secuencia cuyo post esta dentro de las primeras acciones. Esto podría darse a razón que un post al inicio de una sesión tiene una intención menos específica, y termina siendo algo con menor importancia para la comunidad.

Modelo 2

Enfoque Concept-based

Para el enfoque concept-based se encontraron 4 clusters y el análisis se ve en la Tabla 24.

Tabla 24: Secuencia y contenido modelo 2 CB

Cluster	Secuencia (s)	Contenido
Cluster 0	Ver, Ver, Post, Ver, Ver	Amplificadores y efectos.
Cluster 1	Ver, Post, Ver, Post, Ver	Amplificadores y amplificadores a tubo.
Cluster 2	Ver, Ver, Ver, Ver, Post	Amplificadores a tubo, efectos e preguntas de inicios.
Cluster 3	Ver, Ver, Post, Ver, Ver	Amplificadores, inicios e Instrumentos de cuerda.

En este modelo y representación actual, la cantidad de clusters termina siendo bastante bajo. Sin embargo, las secuencias representativas, que en este caso son 4, terminan siendo las únicas en cada cluster para este análisis que se basa en una estrategia de analizar sólo la mediana y sus vecinos más cercanas. Lo anterior, ratifica el hecho que la secuencia es importante para la creación de cada grupo, y que aunque se pierdan algunas secuencias, estas son las más importantes dentro de las sesiones del sitio. Es interesante también observar que las secuencias son distintas, esto quiere decir que el objetivo de que la secuencia sea algo importante se ha cumplido.

El análisis del texto, dado que se usa el enfoque concept-based en este trabajo, no posee una gran cantidad de conceptos representativos, así que no aporta mucha información, salvo que al menos existen aspectos diferentes. Singularmente, los mensajes referidos a inicios tienden a estar dentro de sesiones con posteo en las últimas posiciones.

Enfoque LDA

En este enfoque se hallaron 5 clusters, los que se analizan a partir de la Tabla 25.

Tabla 25: Secuencia y contenido modelo 2 LDA

Cluster	Secuencia (s)	Contenido
Cluster 0	Ver, Post, Ver, Post, Ver	Interruptores para efectos y transformadores para amplificadores a tubo.
Cluster 1	Ver, Ver, Ver, Ver, Post	Cajas para efectos y transformadores para amplificadores a tubo.
Cluster 2	Ver, Ver, Ver, Post, Ver	Cajas para efectos y problema de acople de distorsión.
Cluster 3	Ver, Ver, Post, Ver, Ver	Transformadores para amplificadores a tubo.
Cluster 4	Ver, Post, Ver, Ver, Ver	Solicitudes de ayuda, felicitaciones varias y referencias dónde comprar.

Luego, se encuentran 5 secuencias identificadas como importantes para los distintos clusters. Igual que en el la representación por concept-based, la cantidad de clusters es bastante pequeña, pero consiste de clusters que al menos en su centro, son bastante estables respecto a la secuencia y al contenido.

El contenido, según la metodología de análisis permite definirse mediante una gran gama de tópicos, vale decir, que en este modelo existe una mejor correlación de lo que significa una secuencia con su contenido. No obstante, esto es sólo en el conjunto de los vecinos más cercanos de cada mediana, no se puede aseverar esto en el borde.

Para el modelo 2, los resultados suele estar mejor clasificados en el contenido, aunque no todas las secuencias existentes dentro de las sesiones de usuario tengan un cluster que las represente. En conjunto, al menos 5 secuencias se consideran importantes y en el análisis se encuentran recalcadas fuertemente.

Análisis y diferencias en el análisis de secuencia y contenido entre modelos

En general, ambos modelos obtienen clusters bastante distintos, aunque en esencia identifican, como importantes, las mismas secuencias de usuario dentro de los datos.

La discrepancia entre los resultados es radical, y se debe a las diferencias que generan las dos medidas involucradas, pero este análisis permite darse cuenta que existen ventajas y desventajas entre los modelos.

De las ventajas que tiene el modelo 1, es que genera un modelo de clusters con muchos grupos, pero estos grupos tienden a representar ampliamente el espectro de las sesiones, aunque a este nivel no es posible ver si esto es útil para la moderación o no. Otra ventaja parece tener al momento de identificar sesiones con información no relevante para la comunidad, lo que en realidad se sustenta en la identificación de tópicos para el LDA por parte del experto, pero dada la simplicidad del análisis no es posible aseverarlo completamente.

Las desventajas que muestra el modelo 1 en este análisis, se concentran principalmente en la dificultad de identificar una secuencia imperante para cada cluster, puede significar que la medida de similitud genere clusters un poco desordenados y que la secuencia no sea tan relevante como suele ser en el Web usage mining. Posiblemente esto es a causa del uso de tiempos relativos en la medida.

Las ventajas del modelo 2, parten del hecho que son pocos clusters, así que el análisis termina siendo más simple. Además, las secuencias encontradas suelen ser más representativas por cada cluster, y los contenidos mejor distribuidos, no se repiten en los

distintos clusters. Esto último, puede que deba a que la definición de la medida de disimilitud consta con la clasificación de la comunidad estudiada.

Sin embargo, las desventajas se basan en que al ser pocos clusters, el análisis es menos potente ya que se requiere un análisis aún más exhaustivo para hallar las características del patrón real.

Análisis secuencia-contenido y moderación

Dado que el objetivo de este trabajo es encontrar relaciones entre las distintas etapas del análisis propuesto y la moderación, en esta sección se genera una pequeña discusión al respecto.

Según información obtenida de la literatura y expertos, considerada en la sección 3.1, al analizar la secuencia de la navegación de usuario se pueden vislumbrar algunas características que, de manera intuitiva, podrían generar mensajes que debieran ser moderados.

A partir de la secuencia, se logra indagar si el usuario tenía la intención de generar un post rápidamente o si cuidadosamente tuvo la oportunidad de leer y buscar información en la comunidad. Considerar que las navegaciones de usuario que generan mensajes en las posiciones iniciales requieren una mayor tasa de moderación es razonable. Sin embargo, hay que distinguir que este comportamiento puede deberse al menos por dos razones, es un mensaje nuevo o el usuario tenía la intención clara en ese momento. En cambio, para analizar si las navegaciones de usuario que generan mensajes tardíamente en la secuencia, requiere entender de qué estaba hablando, puesto que posiblemente es una persona que buscó información y no la encontró, lo que también requiere intervención de los moderadores.

Por otro lado, el contenido también ayuda, puesto que si los temas hablados no están centrados en tópicos específicos, existe una alta probabilidad que traten de temas que requieren moderación. Dentro del mismo contexto, necesitarán moderación, temas relacionados con novatos en la comunidad, preguntas generales y peticiones de ayuda.

5.2.- Análisis del modelo mediante métricas para la moderación

Luego de generar un análisis exploratorio de los clusters de sesiones obtenidos en este trabajo e intentar inquirir circunstancias de navegación relacionadas con la moderación dentro de la comunidad, parece necesario crear un análisis un poco más profundo, particularmente de aspectos no incluidos dentro de la similitud o disimilitud de los elementos.

En este trabajo se intenta encontrar una relación entre la necesidad de moderar con el comportamiento que tienen los usuarios dentro del sistema. Luego, parece necesario estudiar qué probabilidad existe que un comportamiento genere un post que luego el moderador deberá revisar.

Expresado dentro de la sección 3.1.2., existen características cuantitativas de los mensajes que los hacen más revisables o menos revisables por el moderador. Se propuso para este trabajo utilizar un análisis a partir de algunas de estas variables con el propósito de medir el grado de revisión tiene cada cluster.

Cada sesión de usuario posee al menos un post generado por el usuario, son estos mensajes lo que deben o no ser revisados y moderados. Las características que se usaron en este análisis se muestran a continuación:

1. La cantidad de respuestas que posee la discusión del mensaje ingresado en total.
2. La cantidad de vistas que tiene la discusión del mensaje ingresado.
3. La cantidad de mensajes ingresados por la sesión actual.
4. La razón existente entre los mensajes de la discusión y la cantidad de usuarios distintos que interviene en ella.
5. La cantidad de mensajes que el usuario de esta sesión ingresa en la comunidad en una semana.
6. Si el usuario de la sesión ha ingresado al sistema el último mes.

Luego, es necesario ajustar cuando significativamente los valores antes explicados se consideran suficientemente importantes para ser considerados en la moderación. Una forma estadística de decidirlo es elegir a partir de los promedios.

A partir de la información de todas las sesiones estudiadas, observando los promedios de las métricas analizadas, es posible ver que una sesión se considerará revisable si sus post poseen las siguientes características:

1. El máximo largo de discusión es mayor a 16.
2. El mínimo largo de discusión es menor a 2.
3. La cantidad de mensajes generados en una sesión es mayor a 1.
4. La razón entre el largo de la discusión sobre cantidad de usuarios distintos dentro de esa discusión es mayor a 2.
5. La cantidad de mensajes generado por el usuario de la sesión en una semana es mayor a 5.
6. La cantidad de vistas máxima de discusión es mayor a 1000.
7. Si el usuario de la sesión se registró a lo más hace un mes.

Luego, cada característica genera un porcentaje de sesiones dentro de un cluster que representa que tanto es la moderación necesaria en esa característica. Se postula el uso de la suma de estos porcentajes para poder apreciar de manera absoluta la diferencia que existe entre los distintos clusters de un modelo.

A continuación se presentan los resultados de este análisis para los modelos planteados en este trabajo. El detalle del puntaje del análisis se deja en la sección Apéndices B.

Modelo 1

Enfoque Concept-based

Los resultados para el análisis se pueden apreciar en la Tabla 26.

Dado que el modelo contiene 10 clusters, se aprecia que los puntajes finales de cada cluster son muy variados, se postula que los clusters con menor puntaje debieran corresponder a sesiones de usuarios menos relevantes al momento de moderar. El cluster que tiene mayor puntaje es el con identificador 0 con 200 y el que posee el menor es el con identificador 6 con 119,76. La diferencia en los puntajes se basa principalmente entre la característica 1 y la 4.

Tabla 26: Puntaje revisión modelo 1 CB

ID Cluster	Puntaje
0	200
1	187,5
2	151,79
3	166,3
4	157,29
5	128,2
6	119,76
7	149,65
8	177,84
9	163,26

El segundo cluster con mayor puntaje (187,5), con ID 1, posee el puntaje más alto de usuarios nuevos y un no despreciable porcentaje de elementos con muchas respuestas y vistas de discusión, y según los resultados y el análisis sobre secuencia, es un cluster con pocos elementos y la secuencia característica genera los mensajes muy cercano al inicio de la sesión, por ende, parece ser un gran candidato a ser revisado. Haciendo este mismo análisis con el caso contrario, el cluster con ID 6, no posee usuarios nuevos, la cantidad de sesiones es relativamente baja, y la secuencia característica indica que el mensaje se ingresa luego de haber observado varios otros.

Enfoque LDA

Los resultados para el análisis se pueden apreciar en la Tabla 27.

Tabla 27: Puntajes revisión modelo 1 LDA

ID Cluster	Puntaje
0	150
1	146,15
2	148,21
3	139,02
4	170,11
5	182,22
6	181,35
7	190,14
8	142,22
9	157,92
10	194,82
11	151,79
12	162,29

Se aprecia que el cluster con menos puntaje es el 3, y el que posee un mayor puntaje es el 10. Sin embargo, estos clusters no presentan valores importantes respecto a los demás, es decir, sus puntajes son todos altos.

En el caso del cluster 7, con el segundo más alto puntaje 190,14, posee discusiones largas y son sesiones con bastante usuarios nuevos en comparación a otros clusters, pero aún más relevante para la validación de este análisis es que según el análisis de contenido, el tópico más importante es contenido irrelevante o que no aportan a la comunidad según el experto. El cluster 1 que es que no posee sesiones de novatos, es uno de los que posee menos puntaje, 146,15, su secuencia muestra que el mensaje se ingresa luego de revisar en la comunidad y generalmente son felicitaciones, lo que no debieran generar problemas que deban ser revisados con posterioridad.

Modelo 2

Enfoque Concept-based

Los resultados para el análisis se pueden apreciar en la Tabla 28.

Tabla 28: Puntajes revisión modelo 2 CB

ID Cluster	Puntaje
0	146,17
1	171,73
2	174,24
3	139,84

En este modelo la diversidad de puntajes y comportamientos estudiados es menor, pero tener un modelo más pequeño para este trabajo trae la ventaja de una mayor facilidad al hacer el análisis. El cluster con menor puntaje es el 3 y el con mayor es el 2. La diferencia es bastante pequeña, es decir, 34.4 puntos, lo que hace difícil discriminar el porqué de esta diferencia. Sin embargo, las diferencias dentro de este análisis tienen que ver con que las sesiones difieren en las medidas 3, 4 y 5, siendo sesiones en la que se generan más mensajes en ellas, las discusiones son entre menos personas y entre personas que suelen postear mucho más

durante una misma semana. El cluster 1 que posee el segundo mayor puntaje de moderación, tiene ingreso de mensajes en la segunda posición, por lo que probablemente pueden ser demasiado apresurados e indican la necesidad de ser revisados.

Enfoque LDA

Los resultados para el análisis se pueden apreciar en la Tabla 29.

Tabla 29: Puntajes revisión modelo 2 LDA

ID Cluster	Puntaje
0	266,66
1	165,8
2	145,51
3	153,42
4	138,94

Aunque el modelo sólo provee de pocos clusters, los puntajes se distribuyen en un intervalo mucho más grande que los otros modelos, lo que refleja que existen diferencias significativas en los puntajes. El cluster 0 que es el que tiene el mayor puntaje, sólo tiene elementos con al menos 2 mensajes ingresados en la sesión, es por esa razón que su puntaje es tan alto. Sin embargo, son sesiones cuyas discusiones de pocos usuarios (alto puntaje característica 4), con usuarios muy activos (alto puntaje característica 5) y generalmente de personas nuevas (característica 7). El cluster 4 con el puntaje más bajo, aunque según el análisis de contenido tienden a ser solicitudes de ayuda, genera poco puntaje probablemente porque las respuestas son cortas y concisas por parte de los otros usuarios, por lo que para el moderador, no debieran tener tanta prioridad. No obstante, el cluster 2 que es el siguiente en puntaje, tanto por las características como por la secuencia parece ser contener mensajes que tuvieron tiempo para ser ingresados concienzudamente y por ende no requieren tanta revisión por parte de los moderadores.

Finalmente, luego de haber analizado por las características propuestas en este trabajo cada modelo de clustering obtenido en este trabajo, cabe destacar que el poder de clasificación debe ser evaluado.

5.3.- Encuesta y Evaluación

Para entender realmente si el modelo de comportamiento de usuario aporta con información relevante a los aspectos de la moderación, se realizó una encuesta, con la intención de capturar la intención real de revisión de distintos post generados en la comunidad en un tiempo determinado. En el caso específico de este trabajo, la encuesta se realizó a uno de los moderadores de Plexilandia, quién lleva participando de la comunidad desde hace más de cinco años y tiene muchas experiencias dentro de otras comunidades generalmente asociadas a la música.

La encuesta se diseñó a partir de la información de 100 sesiones de usuarios capturadas por el sistema, de donde la información a evaluar corresponde a cada post generado por el usuario de la sesión. La información provista consta de la información de la discusión, tales como las vistas, las respuestas, el título, autor, clasificación del post, el texto del post del usuario y del post original de la discusión. Para capturar el nivel de revisión o moderación de cada sesión se le pidió al moderador experto que calificara con una nota de 1 a 5 cada post, considerándose el valor máximo para los casos donde existan dos o más mensajes en la sesión.

Como resultado de esta encuesta se obtiene el detalle de la calificación que se explica en la Tabla 30 y Figura 22 del histograma de frecuencias de elementos, donde se muestran la cantidad de sesiones clasificadas para cada nivel de calificación. En ellos se aprecia que existe una mayoría predominante de sesiones calificadas con bajo índice de revisión, según la información del experto, esto es algo común dentro de la comunidad estudiada.

Tabla 30: Puntajes de elementos evaluados

calificación	cantidad
1	25
2	33
3	33
4	5
5	4

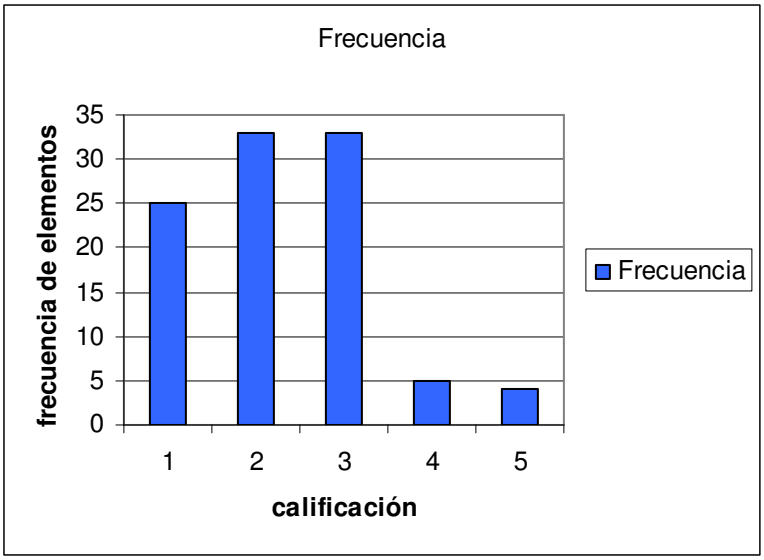

Figura 22: Histograma elementos evaluados
Fuente: Elaboración propia

Luego de obtener los resultados de la encuesta, se procede a evaluar el análisis propuesto dentro de este trabajo. Para lo anterior, es necesario utilizar los modelos de clustering estudiados, basándose en los modelos propuestos. Para la evaluación se sigue la metodología siguiente:

1. Basándose en los análisis anteriormente presentados, cada comportamiento de usuario (o cluster de cada modelo propuesto) será identificado como con necesidad de moderación o no.
2. Mediante las definiciones de cada modelo (medidas y resultados) se distribuyen las sesiones evaluadas al grupo con mayor relación.
3. Se analizan los resultados.

Entonces, la manera de seleccionar si un cluster requiere moderación más que otro se basa la estructura de la secuencia, principalmente en que el cluster sea representado sólo por una que tenga un mensaje ingresado no más allá de la segunda posición y el puntaje obtenido del análisis de métricas de moderación sea alto (mayor a la cantidad media entre el que tiene más puntaje y el que tiene menos puntaje).

Para la validación, se consideran importantes para ser revisados o moderados si su puntaje es mayor o igual a tres y se realiza una comparación entre lo que clasifica el modelo obtenido por la metodología del trabajo y la encuesta del moderador.

Para el análisis se calcula el F-measure de los distintos modelos, usando la precisión y el recall de cada modelo, basándose en la suma de elementos relevantes revisados, los elementos totales relevantes y los elementos totales revisables a partir de usar un subconjunto de clusters de los modelos que requieren moderación según el análisis.

Modelo 1

Enfoque Concept-based

En la Tabla 31, se ve la clasificación de los 100 elementos con el modelo de clustering.

Tabla 31: Elementos relevantes por cluster modelo 1 CB

ID cluster	total clasificados	total relevantes
0	8	4
1	7	2
2	7	2
3	17	7
4	13	7
5	10	3
6	9	4
7	10	3
8	5	3
9	14	7

Por los supuestos implicados del análisis para considerar un cluster más revisable que el resto, los clusters con ID's 0, 1, 3, 4, 8 y 9 deben ser revisados. Esto implica que revisando un 64% de todos los elementos se pueden identificar 71% de los elementos que deben ser moderados. La precisión es de 0,46 y el recall llega a 0,71. Calculando F_1-Measure se obtiene como resultado 0.56.

Enfoque LDA

En la Tabla 32, se ve la clasificación de los 100 elementos evaluados para este enfoque.

Los clusters que debieran ser revisados son ID's 0, 1, 2, 3, 4, 5, 6, 7, 10, 12. Estadísticamente, implica haber revisado 85% de todos los elementos encontrando un 88% de los elementos revisables. Las medidas de precisión y recall terminan siendo 0,43 y 0,88 respectivamente. Luego, el F_1-Measure queda en 0,58.

Tabla 32: Elementos relevantes por cluster modelo 1 LDA

ID cluster	total clasificados	total relevantes
0	10	5
1	13	3
2	5	2
3	4	2
4	8	5
5	7	3
6	6	4
7	9	5
8	4	2
9	6	2
10	15	6
11	5	1
12	8	2

Modelo 2

Enfoque Concept-based

En la Tabla 33, se ve la clasificación de los 100 elementos evaluados para este enfoque.

Tabla 33: Elementos relevantes por cluster modelo 2 CB

ID cluster	total clasificados	total relevantes
0	26	10
1	42	18
2	23	8
3	9	6

Los clusters que se identifican revisables según el análisis anterior tienen ID's 1 y 2. Lo que implica que se revisa 65% del total para capturar un 61% de los elementos revisables. La precisión es de 0,4 y el recall de 0,61. El F_1-Measure del modelo es 0,48.

Enfoque LDA

En la Tabla 34, se ve la clasificación de los 100 elementos evaluados para este enfoque.

Tabla 34: Elementos relevantes por cluster modelo 2 LDA

ID cluster	total clasificados	total relevantes
0	9	5
1	24	9
2	19	8
3	23	10
4	25	10

En este caso, los clusters que debieran revisarse según el análisis anterior poseen ID's 0, 1 y 4. Con esto se revisa un 58% para obtener un 57% de los elementos relevantes para la revisión. La precisión es de 0,41 y el recall de 0,57. El F_1-Measure de este enfoque es de 0,48.

Evaluación general

Como análisis general, es importante notar el hecho que los datos relevantes corresponden a un 42% de los datos totales para la evaluación, por ende, la mayoría de los datos no son relevantes y generan mucho ruido en el análisis. Esto puede ser así por el tipo de comunidad, la cual ya se encuentra consolidada, y las personas que participan generan mensajes específicos que no son relevantes al momento de revisar por los moderadores. De hecho, el mismo administrador considera que la moderación en la comunidad, con los años, suele ser más relajada.

Un hecho interesante es el que la precisión de los modelos suele estar cerca de 0,42 que es la fracción de elementos que necesitan moderación dentro de la toda la validación. Por esa razón, la comparación entre los modelos se da principalmente por el recall obtenido, lo que generalmente favorece al modelo 1, que tiende revisar un mayor porcentaje. Sin embargo, al momento de comparar la clasificación para elementos con puntaje sobre la moderación, entre 4 y 5, todos los modelos clasifican correctamente al menos un 55% de los elementos, e inclusive un 100% de esos elementos en el modelo 1 con enfoque LDA. El detalle se muestra en la Tabla 35.

Tabla 35: Porcentaje elementos de alta moderación por modelo

modelo	porcentaje clasificación
modelo 1 CB	77%
modelo 1 LDA	100%
modelo 2 CB	55%
modelo 2 LDA	77%

Dentro de los modelos propuestos, aquellos que se generan con LDA, que en particular en este trabajo tiene una representación de dimensión considerablemente mayor al enfoque CB, permite una mejor clasificación, esto se ve al momento de revisar los F-Measure y el hecho que en ambos casos se encuentra un porcentaje mayor de elementos que requieren moderación, revisando menos del total. Este hecho se da probablemente porque los modelos tienen un número mayor de clusters que sus símiles del enfoque Concept-based. Sin embargo, la diferencia no es radicalmente mayor.

Indagar si un grupo o comportamiento de usuario requiere más moderación que otro, requiere un análisis bastante grande, pero depende de las características generales y del posible juicio de los expertos. Sin embargo, un análisis simple de las características de los mensajes, las secuencias de la sesión y el contenido permiten obtener resultados relevantes, puesto que en el peor caso, se revisa lo mismo que antes.

Un hecho dentro de esta evaluación fue que la mayoría de los clusters que se definieron como con mayor grado de moderación a partir de la secuencia con acción de generar mensajes en las primeras posiciones termina teniendo alta precisión, esto debe ser a razón que son sesiones que crean discusiones, por lo tanto suelen ser mensajes con información nueva o son sesiones de usuarios desinformados, la cual es la razón por la que se eligieron estos clusters.

6.- Conclusiones

Las comunidades virtuales de práctica son entidades sociales que dependen de las personas que las conforman. El actuar de las personas, las actividades que realizan, la disciplina en la que se desarrollan y las experiencias de la vida van moldeando los comportamientos de todos los individuos pertenecientes a la comunidad y de aquellos quienes la administran. El estudio del comportamiento de los usuarios en la manera que interactúan en la aplicación que soporta la comunidad tiene un gran impacto en todas las dimensiones existentes de ésta.

Los moderadores de las comunidades realizan una gran cantidad de tareas, que deben ser realizadas todos los días, y muchas veces por día. Dentro de las tareas más comunes están la de revisar cada mensaje ingresado, respondiendo preguntas técnicas y ayudando a los novatos. Además, se encuentran tareas asociadas al diseño de nuevas categorizaciones para facilitar las discusiones en el foro. Así, también deben ejecutar las reglas instauradas dentro de la comunidad, advertir y castigar a aquellos usuarios que quiebran las reglas de manera reiterada, sin dejar pasar posibles abusos que sean perjudiciales para el resto de la comunidad. Para llevar a cabo todas tareas en un tiempo muy limitado cada día, los administradores suelen elegir un conjunto de personas que realicen estas tareas para subsecciones de la comunidad, cada una de ellas tiene sus criterios, pero muchas veces, se basan en características operativas, tales como la cantidad de respuestas de una discusión o si los usuarios involucrados son novatos.

Un estudio de esta envergadura necesita un buen diseño, que utilice metodologías ya probada en otros contextos, tales como el KDD y la minería de datos. La rigurosidad de estas técnicas permite generar un diseño reusable no sólo en el análisis de una comunidad sino que cualquiera que se base en los mismos principios de interacción. En el trabajo se utilizaron algoritmos de clustering particional, con la intención de encontrar clusters o grupos de comportamientos de navegación de usuarios que requirieran de moderación. Se postuló el uso de modelos con distintas medidas de similitud y disimilitud, además de varias representaciones del texto, permitiendo realizar una comparación tanto por compatibilidad con los algoritmos como para interpretación semántica del contenido.

La información de la navegación de usuario de cada comunidad permite tomar las decisiones necesarias respecto al diseño optado en este trabajo. Dadas las acciones más

importantes, el diseño debe implementarse para que el estudio sea representativo al momento de decidir que significa que un comportamiento de usuario requiera la intervención de un moderador. Para el caso del foro de Plexilandia.cl, lo que representa un comportamiento es la secuencia de mensajes observados y generados por el usuario dentro de una sesión.

Para sustentar el trabajo, se implementó la propuesta de solución en un software, que permite al analista realizar cada paso de la metodología, pudiendo advertir aquellos aspectos más importantes al momento de ir avanzando sobre el proceso. El diseño del software se sustenta principalmente en tratar de permitir la inclusión de varias versiones de los algoritmos, para que mediante las interfaces, el usuario pueda realizar los experimentos que considere necesarios, y además de darle la extensibilidad necesaria para en el futuro ser utilizada en otras comunidades.

El primer análisis, proveniente del web usage mining de los inicios, que intenta capturar la navegación del usuario a partir de la secuencia de navegación permite analizar las posibilidades de que estas generen la necesidad de ser revisadas y moderadas de una manera simple, partiendo de intuiciones lógicas con respecto a la velocidad de respuesta y su impacto final en la comunidad. Una respuesta apresurada, con alta probabilidad genera situaciones que deben ser inspeccionadas. Analizando aspectos del contenido también, sin necesidad de ser experto, se puede interpretar como importante o no para la moderación.

En busca de un mejor análisis se postula el uso de características de los comportamientos de usuarios sobre la posible moderación. No obstante, no dejan de ser aspectos que dependen de factores culturales y de las reacciones de los individuos dentro de la comunidad. A mayor reacción, las diferencias en los comportamientos de usuarios son más significativas y pueden inferir mucho más en los aspectos de la moderación. En otras palabras, si la comunidad no reacciona a situaciones anormales, las características no cambiarán de manera significativa entre comportamientos, por lo que el análisis estará sesgado.

A modo de validación del trabajo, el clustering de comportamientos de usuarios permite mediante su análisis metodológico y experto encontrar un nivel aceptable de elementos que requieren revisión, revisando una fracción de los datos, por ende utilizando sólo una fracción del tiempo. En este trabajo se postula el uso del análisis de secuencia-contenido y las características de moderación, para decidir si un comportamiento de usuario requiere más moderación que otro. Dentro de la evaluación de esta hipótesis, realizada en conjunto con el experto de la comunidad sobre 100 comportamientos de usuario del foro de Plexilandia.cl, el

modelo 1 alcanza un 88% de captación de los elementos importantes para la moderación revisando un 85% del total de elementos que debiera revisar, y el modelo 2 alcanza a capturar un 61% en 65% de elementos revisados. Los resultados pueden parecer no tan buenos, pero se descubre que los modelos tienden a clasificar correctamente a aquellos que requieren más moderación. Una ventaja del modelo 1 es que permite visualizar mucho mejor las diferencias dentro de los comportamientos de usuario, al generar un modelo con una gran cantidad de clusters, pero esto mismo es una desventaja, porque es necesario tomar más atención al analizar una mayor cantidad de grupos. El modelo 2 tiene la ventaja de poder ser analizado con mayor rapidez, pero pierde mucho en recall.

Un hecho importante en el uso de las representaciones del texto es que el modelo de LDA permite un mejor análisis. Existen dos razones fundamentales, primero genera modelos de cluster con más clusters, entonces genera más posibilidad de segmentar por comportamientos de usuarios más específicos, y segundo, la representación, genera un conjunto de tópicos bastante amplio, que luego de ser analizados por el experto, mantiene información semántica útil para entender lo que tiene el contexto pudiendo ser entendido por analistas que no necesariamente entienden de la actividad de la comunidad.

6.1.- Trabajo Futuro

Cabe señalar que este fue un trabajo experimental y de investigación con grandes posibilidades de ser desarrollado bajo distintos contextos, aspectos y puntos de vista. La moderación incluye a todos los individuos de la comunidad y tanto sus comportamientos como sus intenciones y pensamientos pueden llegar a influir en las situaciones futuras.

Seguir trabajando en esto, al menos, implica dos líneas de investigación, primero, encontrar un mejor análisis respecto a las posibles variables que definen los comportamientos, y segundo encontrar comportamientos o características asociadas a la moderación a partir de otras técnicas de minería de datos.

Probablemente utilizar medidas de similitud o disimilitud que incluyan características asociadas a aspectos que deben ser moderados en la navegación de usuario genere mejores resultados, o al menos segmente los resultados de mejor forma. Estas pueden ser de usuario, tales como la frecuencia en que estos generan mensajes en la comunidad, o también respecto al impacto que puede tener el mensaje a partir de ciertas palabras que se asocian a discordia.

Apéndices

A.- Tablas de los enfoques Concept-based y LDA

Para la creación del modelo para concept-based se usa a los expertos del sitio. La construcción de sus componente fue realizado en trabajos anteriores del grupo de investigación del profesor Sebastián Ríos. Se definen los conceptos en la Tabla A. 1.

Tabla A. 1: Conceptos del modelo CB

ID	Descripción
1	Generar una comunidad hispanoparlante con interés comunes en torno a la construcción propia (DIY do it yourself) de efectos. Amplificadores y equipos de audio musicales.
2	constituir un repositorio de conocimientos en torno a la construcción de amplificadores de tubo
3	constituir un repositorio de información en torno a la construcción de efectos
4	constituir un repositorio de información en torno a la construcción de instrumentos de cuerda
5	ser un punto de partida y referencia para quienes se inician en el DIY relacionado con música y audio
6	crear un repositorio de información en torno al audio profesional

El caso del LDA se construyó de la misma manera en un trabajo anterior a este. Aunque el algoritmo es no supervisado al momento de generar las probabilidades respecto a los tópicos, se le pide a un experto sobre la información estudiada ayuda para darle un nombre a los tópicos encontrados. Estas definiciones dadas por un experto de la comunidad de Plexilandia se muestran en la Tabla A. 2.

Tabla A. 2: Tópicos modelo LDA

id	nombre
1	Distorsión
2	no sirve
3	Impresión de Placas para efectos
4	Preguntas clásicas usuarios nuevos
5	Solicitudes de ayuda
6	Técnicas de soldado (de efectos y amplificadores)
7	Conexión de cables/cableado
8	Específicamente: por favor redacten bien

9	Conexión y construcción de cajas
10	no sirve
11	explicación de fallas
12	felicitaciones varias
13	maderas para construcción de guitarras
14	Tiendas de venta de insumos electrónicos
15	referencias a libros y textos en ingles
16	Modificaciones al Amplificador JCM Marshall modelo slash
17	componentes electrónicos (énfasis en condensadores)
18	amplificadores a tubo
19	conversación en base a referencias de posts pasados
20	no sirve
21	concepto de marcas de artesanos
22	no sirve
23	sitio web y foro plexilandia
24	imágenes y vídeos del avance de lo que se construye.
25	efectos de modulación
26	referencias de lugares donde comprar
27	diferencias de sonido dependiendo de diversos factores
28	no sirve
29	proceso de detección de fallas
30	ajuste de transistores
31	apreciación del sonido de bandas
32	transformadores para amplificadores a tubo
33	rectificador en amplificadores a tubo
34	no sirve (conversación)
35	muchas palabras mal escritas
36	normas y buena convivencia en Plexilandia
37	cajas para efectos y chasis para amplificadores
38	esquemas de efectos
39	distintos efectos
40	interruptores para efectos
41	no sirve
42	plexijunta
43	no sirve
44	solicitudes de compra y venta de componentes
45	modelos y marcad de guitarras
46	etapas de distorsión
47	software y hardware para aplicaciones de sonido
48	opinión y recomendación de marcas v/s precio
49	problema de acople en efectos de distorsión
50	aislación acústica

B.- Detalle del puntaje de análisis por métricas de moderación

A continuación se presentan las tablas de puntaje de los modelos de clustering del trabajo

Modelo 1
Enfoque Concept-based

Tabla B. 1: Puntaje Moderación Modelo 1 CB

ID cluster	Característica	Porcentaje	ID Cluster	Característica	Porcentaje
0	1	53,77358491	1	1	35,71428571
	2	2,830188679		2	3,571428571
	3	15,09433962		3	19,64285714
	4	51,88679245		4	44,64285714
	5	30,18867925		5	35,71428571
	6	41,50943396		6	37,5
	7	4,716981132		7	10,71428571
	Suma o Puntaje	200		Suma o Puntaje	187,5
ID cluster	Característica	Porcentaje	ID Cluster	Característica	Porcentaje
2	1	25,17985612	3	1	28,80434783
	2	1,438848921		2	1,086956522
	3	21,58273381		3	18,47826087
	4	36,69064748		4	42,39130435
	5	43,16546763		5	39,67391304
	6	19,42446043		6	27,17391304
	7	4,316546763		7	8,695652174
	Suma o Puntaje	151,7985612		Suma o Puntaje	166,3043478
ID cluster	Característica	Porcentaje	ID Cluster	Característica	Porcentaje
4	1	21,35416667	5	1	28,20512821
	2	2,083333333		2	0
	3	30,72916667		3	20,51282051
	4	39,58333333		4	25,64102564
	5	36,45833333		5	25,64102564
	6	18,22916667		6	23,07692308
	7	8,854166667		7	5,128205128
	Suma o Puntaje	157,2916667		Suma o Puntaje	128,2051282
ID cluster	Característica	Porcentaje	ID Cluster	Característica	Porcentaje
6	1	25,58139535	7	1	27,27272727
	2	0		2	1,398601399
	3	8,139534884		3	19,58041958
	4	33,72093023		4	32,86713287
	5	31,39534884		5	41,95804196
	6	20,93023256		6	24,47552448
	7	0		7	2,097902098
	Suma o Puntaje	119,7674419		Suma o Puntaje	149,6503497
ID cluster	Característica	Porcentaje	ID Cluster	Característica	Porcentaje
8	1	37,97468354	9	1	30,6122449
	2	0,632911392		2	0
	3	20,88607595		3	21,08843537
	4	46,20253165		4	44,89795918
	5	38,60759494		5	37,41496599
	6	29,11392405		6	25,85034014
	7	4,430379747		7	3,401360544
	Suma o Puntaje	177,8481013		Suma o Puntaje	163,2653061

Enfoque LDA

Tabla B. 2: Puntaje moderación modelo 1 LDA

ID cluster	Característica	Porcentaje	ID Cluster	Característica	Porcentaje
0	1	23,1707317	1	1	21,7948718
	2	0		2	2,56410256
	3	17,0731707		3	34,6153846
	4	43,902439		4	30,7692308
	5	35,3658537		5	39,7435897
	6	24,3902439		6	16,6666667
	7	6,09756098		7	0
	Suma o Puntaje	150		Suma o Puntaje	146,153846
ID cluster	Característica	Porcentaje	ID Cluster	Característica	Porcentaje
2	1	22,3214286	3	1	29,2682927
	2	1,78571429		2	1,2195122
	3	24,1071429		3	12,195122
	4	41,9642857		4	39,0243902
	5	34,8214286		5	34,1463415
	6	17,8571429		6	15,8536585
	7	5,35714286		7	7,31707317
	Suma o Puntaje	148,214286		Suma o Puntaje	139,02439
ID cluster	Característica	Porcentaje	ID Cluster	Característica	Porcentaje
4	1	39,0804598	5	1	41,1111111
	2	1,14942529		2	1,11111111
	3	12,6436782		3	12,2222222
	4	42,5287356		4	47,7777778
	5	40,2298851		5	35,5555556
	6	32,183908		6	36,6666667
	7	2,29885057		7	7,77777778
	Suma o Puntaje	170,114943		Suma o Puntaje	182,222222
ID cluster	Característica	Porcentaje	ID Cluster	Característica	Porcentaje
6	1	30,5084746	7	1	38,028169
	2	3,38983051		2	1,4084507
	3	15,2542373		3	25,3521127
	4	47,4576271		4	47,8873239
	5	47,4576271		5	39,4366197
	6	30,5084746		6	32,3943662
	7	6,77966102		7	5,63380282
	Suma o Puntaje	181,355932		Suma o Puntaje	190,140845
ID cluster	Característica	Porcentaje	ID Cluster	Característica	Porcentaje
8	1	24,4444444	9	1	32,7868852
	2	0		2	1,09289617
	3	13,3333333		3	16,3934426
	4	38,8888889		4	42,6229508
	5	36,6666667		5	33,3333333
	6	22,2222222		6	26,7759563
	7	6,66666667		7	4,91803279
	Suma o Puntaje	142,222222		Suma o Puntaje	157,923497
ID cluster	Característica	Porcentaje	ID Cluster	Característica	Porcentaje
10	1	31,8965517	11	1	31,6546763
	2	0,86206897		2	2,15827338
	3	46,5517241		3	15,8273381
	4	40,5172414		4	32,3741007
	5	43,1034483		5	35,2517986
	6	26,7241379		6	26,618705
	7	5,17241379		7	7,91366906
	Suma o Puntaje	194,827586		Suma o Puntaje	151,798561

ID cluster	Característica	Porcentaje			
12	1	31,147541			
	2	0			
	3	19,6721311			
	4	39,3442623			
	5	40,9836066			
	6	29,5081967			
	7	1,63934426			
	Suma o Puntaje	162,295082			

Modelo 2

Enfoque Concept-based

Tabla B. 3: Puntaje moderación modelo 2 CB

ID cluster	Característica	Porcentaje	ID Cluster	Característica	Porcentaje
0	1	26,17647	1	1	31,77388
	2	1,764706		2	0,974659
	3	15		3	24,95127
	4	39,11765		4	43,07992
	5	36,76471		5	37,4269
	6	19,70588		6	28,07018
	7	7,647059		7	5,45809
	Suma o Puntaje	146,1765		Suma o Puntaje	171,7349
ID cluster	Característica	Porcentaje	ID Cluster	Característica	Porcentaje
2	1	34,84848	3	1	29,32331
	2	0,757576		2	2,255639
	3	23,48485		3	12,03008
	4	42,80303		4	32,33083
	5	40,15152		5	33,83459
	6	29,16667		6	26,31579
	7	3,030303		7	3,759398
	Suma o Puntaje	174,2424		Suma o Puntaje	139,8496

Enfoque LDA

Tabla B. 4: Puntaje moderación modelo 2 LDA

ID cluster	Característica	Porcentaje	ID Cluster	Característica	Porcentaje
0	1	31,74603	1	1	35,930736
	2	2,380952		2	0,4329004
	3	100		3	17,316017
	4	53,1746		4	41,558442
	5	45,2381		5	38,095238
	6	26,19048		6	29,437229
	7	7,936508		7	3,030303
	Suma o Puntaje	266,6667		Suma o Puntaje	165,80087

ID cluster	Característica	Porcentaje	ID Cluster	Característica	Porcentaje
2	1	32,55814	3	1	29,315961
	2	0,664452		2	1,9543974
	3	4,651163		3	16,938111
	4	40,86379		4	35,830619
	5	32,89037		5	39,413681
	6	27,90698		6	24,7557
	7	5,980066		7	5,2117264
	Suma o Puntaje	145,515		Suma o Puntaje	153,4202
ID cluster	Característica	Porcentaje			
4	1	25,26316			
	2	1,403509			
	3	8,77193			
	4	40			
	5	36,14035			
	6	21,75439			
	7	5,614035			
	Suma o Puntaje	138,9474			

C.- Interfaces de la implementación de algoritmos Generación de Sesiones

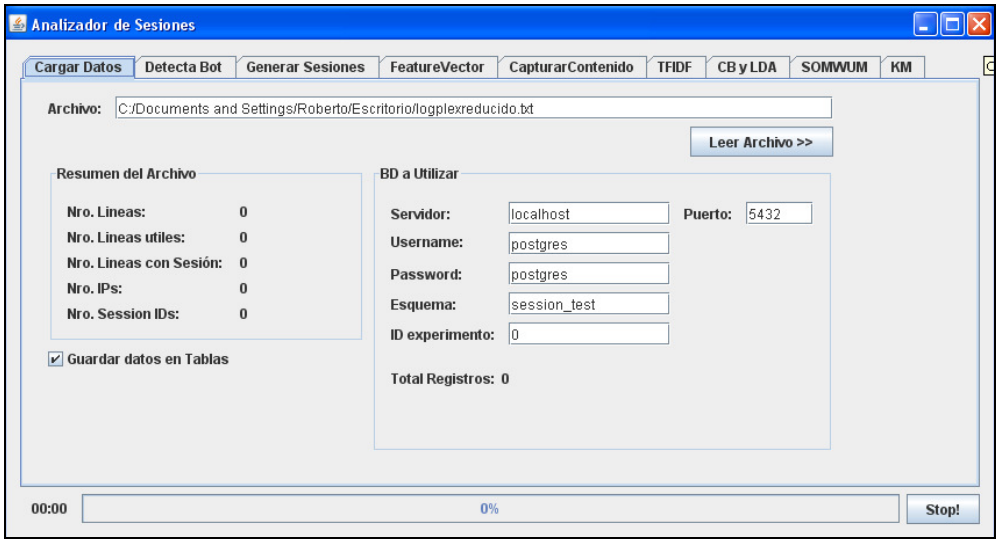

Figura C. 1: Interfaz Ingreso de información de logs
Fuente: Elaboración propia

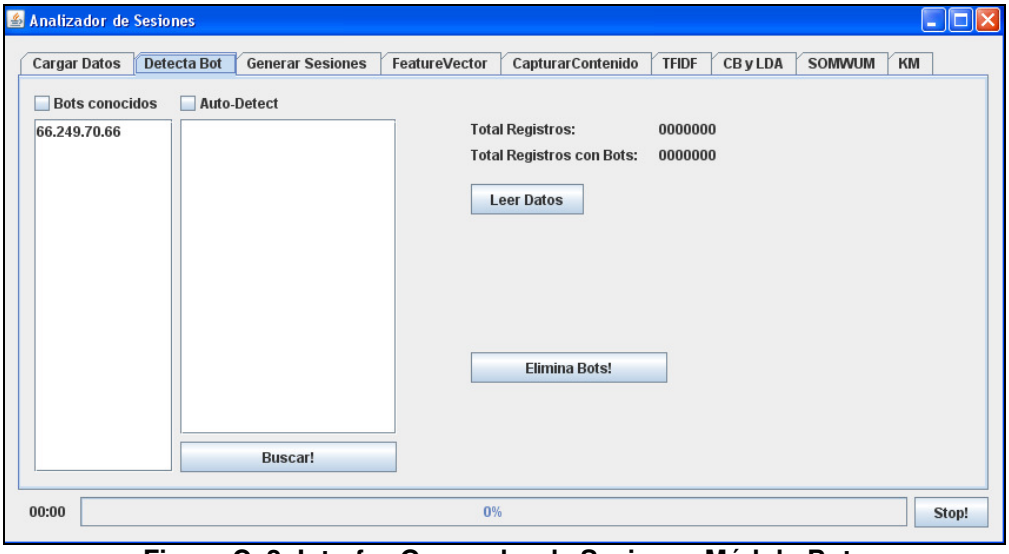

Figura C. 2: Interfaz Generador de Sesiones Módulo Bots
Fuente: Elaboración propia

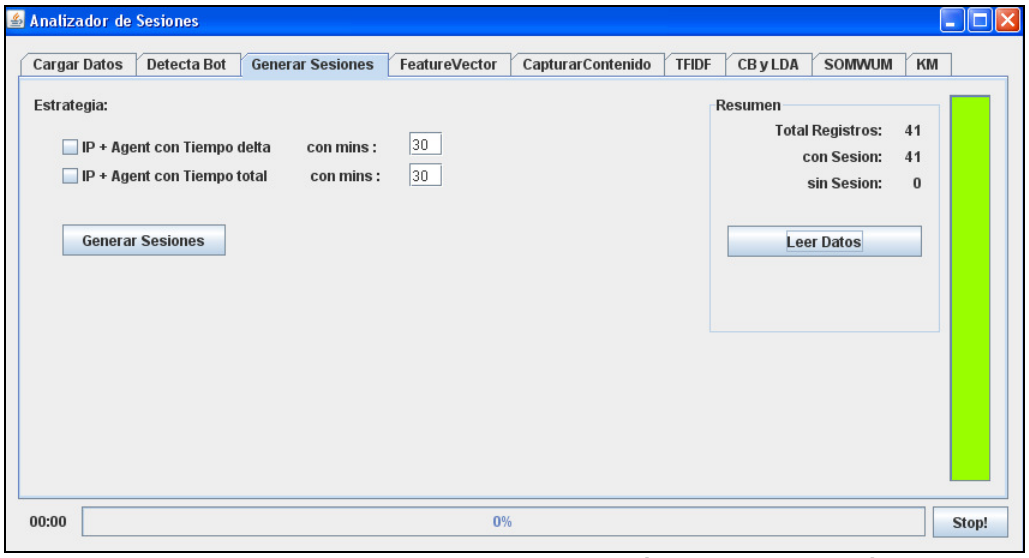

Figura C. 3: Interfaz Genera Sesiones Módulo Sesionización
Fuente: Elaboración propia

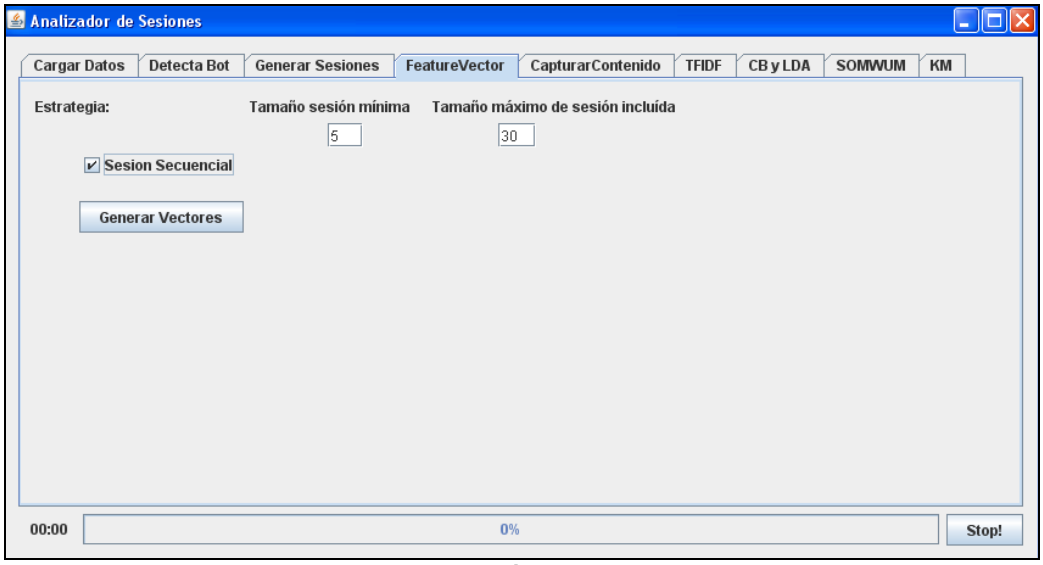

Figura C. 4: Interfaz obtención de secuencias de sesiones
Fuente: Elaboración propia

7.- Referencias

1. E.Wenger, R. A. McDermott, and W. Snyder. Cultivating communities of practice. Harvard Business Press, 2002.

2. Ríos et al. Virtual Communities of Practice's Purpose Evolution Analysis Using a Concept-Based Mining Approach. Knowledge-Based and Intelligent Information and Engineering Systems (2009) vol. 2 pp. 480-489

3. G. Probst and S. Borzillo. Why communities of practice succeed and why they fail. European Management Journal, 26(5):335-347, 2008.

4. W. Kim, O. Jeong, and S. Lee. On social web sites. Information Systems, 35(2):215-236, 2010.

5. Fortuna, B., E. M Rodrigues, y N. Milic-Frayling. "Improving the classification of newsgroup messages through social network analysis." En Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, 877–880, 2007.

6. Moore, Trevor D., y Mark A. Serva. Understanding member motivation for contributing to different types of virtual communities: a proposed framework. En Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research: The global information technology workforce, 153-158. St. Louis, Missouri, USA: ACM, 2007.

7. Yu, Jie, Zhenhui Jiang, y Hock Chuan Chan. Knowledge contribution in problem solving virtual communities: the mediating role of individual motivations. En Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research: The global information technology workforce, 144-152. St. Louis, Missouri, USA: ACM, 2007.

8. Koh, Joon, Young-Gul Kim, Brian Butler, y Gee-Woo Bock. "Encouraging participation in virtual communities." Commun. ACM 50, no. 2 (2007): 68-73.

9. Kim, Amy Jo. Community Building on the Web: Secret Strategies for Successful Online Communities. Addison-Wesley Longman Publishing Co., Inc., 2000.

10. Preece, J. and Diane Maloney-Krichmar (2003) Online Communities. In J. Jacko and A. Sears, A. (Eds.) Handbook of Human-Computer Interaction, Lawrance Eribaum Associates Inc. Publishers. Mahwah: NJ. 596-620.

11. Gairín-Sallán, Joaquín, David Rodríguez-Gómez, y Carme Armengol-Asparó. Who exactly is the moderator? A consideration of online knowledge management network moderation in educational organisations. Comput. Educ. 55, no. 1 (2010): 304-312.

12. Han, J., H. Cheng, D. Xin, y X. Yan. “Frequent pattern mining: current status and future directions.” Data Mining and Knowledge Discovery 15, no. 1 (2007): 55–86.

13. Srivastava, Jaideep, Robert Cooley, Mukund Deshpande, y Pang-Ning Tan. “Web usage mining: discovery and applications of usage patterns from Web data.” SIGKDD Explor. Newsl. 1, no. 2 (2000): 12-23.

14. Rao, V. V.R.M, V. V Kumari, y K. Raju. “Understanding User Behavior using Web Usage Mining” (2010).

15. Han, J., y M. Kamber. Data mining: concepts and techniques. Morgan Kaufmann, 2006.

16. Scime, A., y Inc Books24x7. Web Mining: applications and techniques. Idea Group Pub., 2005.

17. Markov, Z., y D. T Larose. Data mining the Web: uncovering patterns in Web content, structure, and usage. Wiley-Blackwell, 2007.

18. VELASQUEZ, J. D, H. YASUDA, T. AOKI, y R. WEBER. “A new similarity measure to understand visitor behavior in a web site.” IEICE TRANSACTIONS on Information and Systems 87, no. 2 (2004): 389–396.

19. Velásquez, J. D, R. Weber, H. Yasuda, y T. Aoki. “A methodology to find web site keywords” (2004).

20. Lee, D. L, H. Chuang, y K. Seamons. “Document ranking and the vector-space model.” Software, IEEE 14, no. 2 (2002): 67–75.

21. Velásquez, Juan D., y Vasile Palade. “Adaptive Web Sites A Knowledge Extraction from Web Data Approach.” En Proceeding of the 2008 conference on Adaptive Web Sites: A Knowledge Extraction from Web Data Approach, 1-272. IOS Press, 2008.

22. Pei, J., J. Han, B. Mortazavi-Asl, y H. Zhu. “Mining access patterns efficiently from web logs.” Knowledge Discovery and Data Mining. Current Issues and New Applications (2000): 396–407.

23. Velásquez, J. D, y J. I Fernández. “Towards the Identification of Important Words from the Web User Point of View.” En Procs. on Int. Workshop on Intelligent Web Based Tools (IWBT-07), CEUR-WS database, 17–26.

24. Accurate Analytics Require Cookies by Bryan Einsenberg http://www.clickz.com/clickz/column/1706798/accurate-analytics-require-cookies [consulta: 15 de octubre del 2010].

25. Koutrika, Georgia, Frans Adjie Effendi, Zolt\'n Gyöngyi, Paul Heymann, y Hector Garcia-Molina. “Combating spam in tagging systems: An evaluation.” ACM Transactions on the Web (TWEB) 2 (Octubre 2008): 22:1–22:34.

26. Kosmopoulos, A., G. Paliouras, y I. Androutsopoulos. “Adaptive spam filtering using only naive bayes text classifiers.” En Proceedings of the Fifth Conference on Email and Anti-Spam (CEAS), 2008.

27. Theodoridis, Sergios, y Konstantinos Koutroumbas. Pattern recognition. Academic Press, 2003.

28. Dell, R. F, P. E Román, y J. D Velásquez. “Web user session reconstruction using integer programming.” En 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 385–388, 2008.

29. Serrano-Cobos, J. “Combinación de logs internos y externos en la predicción de estacionalidad de búsquedas para el rediseño de webs.” El profesional de la información, Vol. 18, No. 1, pp. 11–19. Enero-Febrero, 2009

30. Xu, R., D. C Wunsch, y Inc Books24x7. Clustering. IEEE Press, 2009.

31. Kanungo, T., D. M Mount, N. S Netanyahu, C. D Piatko, R. Silverman, y A. Y Wu. “An efficient k-means clustering algorithm: Analysis and implementation.” Pattern Analysis and Machine Intelligence, IEEE Transactions on 24, no. 7 (2002): 881–892.

32. Bradley, P. S, y U. M Fayyad. “Refining initial points for k-means clustering.” En Proceedings of the Fifteenth International Conference on Machine Learning, 91–99, 1998.

33. Tan, P. N, M. Steinbach, V. Kumar, y others. Introduction to data mining. Pearson Addison Wesley Boston, 2006.

34. L’Huillier, G., H. Alvarez, S. A Ríos, y F. Aguilera. “Topic-Based Social Network Analysis for Virtual Communities of Interests in the Dark Web” (2010).

35. Alvarez, H., S. Ríos, F. Aguilera, E. Merlo, y L. Guerrero. “Enhancing Social Network Analysis with a Concept-Based Text Mining Approach to Discover Key Members on a Virtual Community of Practice.” Knowledge-Based and Intelligent Information and Engineering Systems (2010): 591–600.

36. Blei, D. M, A. Y Ng, y M. I Jordan. “Latent dirichlet allocation.” The Journal of Machine Learning Research 3 (2003): 993–1022.

37. Kohonen, T. “Self-organized formation of topologically correct feature maps.” Biological cybernetics 43, no. 1 (1982): 59–69.

38. Hay, B., G. Wets, y K. Vanhoof. “Clustering navigation patterns on a website using a sequence alignment method.” En Proc. Intelligent Techniques for Web Personalization: 17th Int. Joint Conf. Artificial Intelligence, 1–6, 2000.

39. Scherbina, A., y S. Kuznetsov. “Clustering of Web Sessions Using Levenshtein Metric.” Advances in Data Mining (2005): 438–449.

40. Wu, Xindong, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey McLachlan, et al. “Top 10 algorithms in data mining.” Knowledge and Information Systems 14, no. 1 (Enero 1, 2008): 1-37.

41. Y. GAO. Web System Design and Onlines Consumer Behavior. Ramapo College of New Jersey, USA. Idea Group Publishing, 2005.

42. Velásquez, J. D, H. Yasuda, T. Aoki, y R. Weber. “Using the KDD process to support Web site reconfigurations.” En Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on, 511–515, 2003.

43. Patel, B. C, y G. R. Sinha. “An Adaptive K-means Clustering Algorithm for Breast Image Segmentation.” International Journal of Computer Applications IJCA 10, no. 4 (2010): 24–28.

44. Mobasher, B. “Web usage mining.” Encyclopedia of Data Warehousing and Data Mining. Idea Group Publishing (2005): 1216–1220.

45. Xu, G., Y. Zhang, y L. Li. “Web Content Mining.” Web Mining and Social Networking (2011): 71–87.

46. Baeza-Yates, R., y P. Boldi. “Web Structure Mining.” Advanced Techniques in Web Intelligence-I (2010): 113–142.

47. ROUSSEEUW, L. K.P.J. “CLUSTERING BY MEANS OF MEDOIDS.” Statistical data analysis based on the L1-norm and related methods (1987): 405.