

Paynal: Una herramienta de Ingeniería de Software Colaborativa

Alejandro Corbellini, María Manuela Rolón y Silvia Schiaffino

ISISTAN, Facultad de Ciencias Exactas
Universidad Nacional del Centro de la Provincia de Bs. As. Tandil, Bs. As., Argentina
CONICET, Consejo Nacional de Investigaciones Científicas y Técnicas
{sschia, acorbellini}@exa.unicen.edu.ar, mrolon@alumnos.exa.unicen.edu.ar

Resumen En este trabajo se describe una aplicación, llamada Paynal, que provee herramientas *groupware* al entorno de desarrollo Eclipse, permitiendo a sus usuarios interactuar mediante chats, foros y mensajes. Dichas interacciones generan un vínculo social y, en conjunto, las mismas conforman una Red Social. Mediante el uso de métricas de Análisis de Redes Sociales se puede extraer información para la detección de líderes de grupo, la identificación de reemplazantes para desarrolladores que abandonan la empresa, entre otros. Utilizando técnicas de Minería de Texto se pueden extraer tópicos de las interacciones entre dos usuarios, permitiendo discernir temas y habilidades. Dichos tópicos son visualizados utilizando Nubes de *Tags*: una representación compacta y, a su vez, altamente expresiva. Con el objetivo de realizar experiencias reales, se experimentó con datos provenientes de una organización de desarrollo de software, y luego, utilizando dichos datos, se elaboraron conclusiones utilizando las herramientas provistas por Paynal.

Palabras Clave: CSCW, CSE, Nube de tags, Minería de texto, Red Social.

1. Introducción

En 1984, Irene Greif y Paul Cashman de DEC (del inglés, *Digital Equipment Corporation*), organizaron un *workshop* que agrupaba personas interesadas en entender cómo los individuos trabajan en conjunto y cómo la tecnología podría dar soporte a su trabajo. Para describir el *workshop* y los temas que se tratarían, acuñaron el término Trabajo Colaborativo Soportado por Computadoras (CSCW, del inglés *Computer Supported Collaborative Work*). Existen varias definiciones de CSCW, siendo la siguiente una de las más adoptadas por los investigadores:

“CSCW es concebido como un esfuerzo por entender la naturaleza y las características del trabajo cooperativo, con el objetivo de diseñar tecnologías de computadoras que soporten dicho trabajo” [1]

De la misma se desprende que, al analizar esta área de investigación, se deben tener en cuenta dos aspectos fundamentales: en primer lugar, la tecnología y el hardware y software de computadoras; y, en segundo lugar, el trabajo grupal y el fenómeno social [7].

Relacionado con CSCW, se encuentra el concepto de Software Colaborativo o *groupware*. El mismo se define como sistemas basados en computadoras que dan soporte a grupos de personas unidos en una tarea común u objetivo, y proveen una interfaz a un ambiente compartido. Esta tecnología utilizada para comunicar, cooperar, coordinar, y resolver problemas facilita el trabajo de grupos a través de redes de computadoras y servicios inherentes en las mismas. Si bien, este tipo de sistemas existía antes del surgimiento de CSCW, fue adoptado por la misma para su estudio, ya que dichos sistemas materializan muchos de los conceptos elaborados por esta área de investigación. Es por eso que varios autores contemplan el *groupware* como parte de CSCW [3].

Por otro lado, los proyectos de ingeniería de software son inherentemente cooperativos y requieren que un gran número de ingenieros coordinen sus esfuerzos para lograr crear grandes sistemas de software. Es integral a este esfuerzo el alcanzar un entendimiento compartido de los múltiples artefactos, cada uno con su propio modelo, a lo largo del proceso de desarrollo [9].

La gran mayoría de las herramientas de ingeniería de software, tal como editores, compiladores y *debuggers*, no presenta soporte directo para la colaboración. En su lugar, la colaboración es delegada a herramientas de control de versión tales como CVS (del inglés, *Concurrent Versions System*) y SVN (abreviación del sistema de versionado *Apache Subversion*). El control de versiones es un mecanismo altamente efectivo para archivar múltiples versiones de software, sin embargo, los procesos de ingeniería de software a menudo requieren, además, comunicar qué se encuentran haciendo los demás ingenieros para evitar conflictos en el desarrollo. Esto es, la comunicación y la notificación es parte fundamental de la colaboración en la ingeniería de software.

A partir de dichas problemáticas, surge el área de Ingeniería de Software Colaborativa (CSE, del inglés *Collaborative Software Engineering*), el cual se encarga de investigar de qué manera se pueden incluir tecnologías *groupware* en las diferentes etapas del desarrollo de software (Captura de Requerimientos, Análisis, Diseño, Desarrollo, y otros).

Para construir herramientas CSE exitosas, es fundamental tener un conocimiento amplio de varias áreas de Ciencias de la Computación. Entre ellas, podemos destacar CSCW para el soporte *groupware*, *Human Computer Interaction* para proveer formas adecuadas de presentar información, Sistemas Distribuidos para asistir a la comunicación entre procesos, y Administración de Configuración para manejar versionados[5].

Sumado a estos esfuerzos, se han desarrollado trabajos que proponen un nuevo punto de vista de las herramientas CSCW, más acorde con las ideas de la Web Social, con el objetivo de ofrecer los aspectos típicos de estas herramientas (colaboración entre usuarios), y generar contexto colaborativo que muestre las

uniones existentes entre los usuarios del sistema y una información más detallada de su entorno de trabajo desde un punto de vista más social[6].

Las redes sociales son fundamentales en la generación de contexto colaborativo; describen las relaciones existentes entre conjuntos de personas mediante el análisis de los aspectos sociales que comparten y los lazos personales o colaborativos que los unen. Así, usando el Análisis de Redes Sociales (ARS) es posible encontrar las estructuras sociales y grupos de trabajo que aparecen cuando existe colaboración entre personas, entendiendo de esta manera los diferentes roles que interpretan cada una según su contexto actual.

El resto del trabajo se organiza de la siguiente manera. En la Sección 2 se presentará Paynal, una herramienta CSE que permite analizar las interacciones entre integrantes de un proyecto de software. Luego, en la Sección 3 se presentará el diseño de dicha herramienta, detallando las tecnologías utilizadas. Por último, en la Sección 4 se brindarán conclusiones sobre el trabajo desarrollado, incluyendo contribuciones, limitaciones del trabajo y trabajos futuros. Además, en el Apéndice A se pueden encontrar experiencias realizadas con la herramienta, utilizando datos reales de una organización de software.

2. Paynal: Una herramienta CSE para el análisis de interacciones

En este trabajo se propone una aplicación CSE llamada Paynal, la cual extiende el entorno de desarrollo Eclipse¹ agregando herramientas *groupware*. Esta aplicación asiste al programador en la comodidad de su espacio de trabajo, permitiéndole interactuar con sus compañeros de equipo a través del diálogo directo, el envío de notas o mensajes instantáneos, la participación en foros y el intercambio de estados e imágenes.

La aplicación organiza el grupo de usuarios mediante la asignación de roles y privilegios. Cada usuario cuenta con un determinado rol dentro del equipo de trabajo y sus consecuentes permisos para uso y acceso a la información. De esta manera, Paynal establece una jerarquía de usuarios, diferenciando administradores de proyectos, jefes de grupo y desarrolladores.

Como se observa en la Figura 1, los usuarios interactúan con la aplicación a través del chat, el foro y/o los mensajes instantáneos. Todas estas acciones son almacenadas en un repositorio para luego extraer y procesar esta información y presentarla gráficamente. Además, es posible extraer conocimiento acerca de los principales temas de interés y habilidades de cada usuario.

La información recolectada concerniente a las relaciones que se suceden entre los usuarios del sistema permite construir una Red Social. Conceptualmente, una Red Social es una estructura compuesta por uno o más grafos cuyos nodos representan actores o unidades sociales discretas y los arcos representan las relaciones entre ellos. El análisis de la Red Social en un proyecto de desarrollo de software permite, por ejemplo, identificar los desarrolladores más activos, los cuales forman el núcleo del proyecto [8]. Para llevar a cabo este análisis, se dispone de

¹ Página Oficial de la Fundación Eclipse, <http://www.eclipse.org>

ciertas métricas que permiten estudiar las redes sociales y emitir conclusiones en cuanto, por ejemplo, a la participación de cada actor dentro de la estructura.

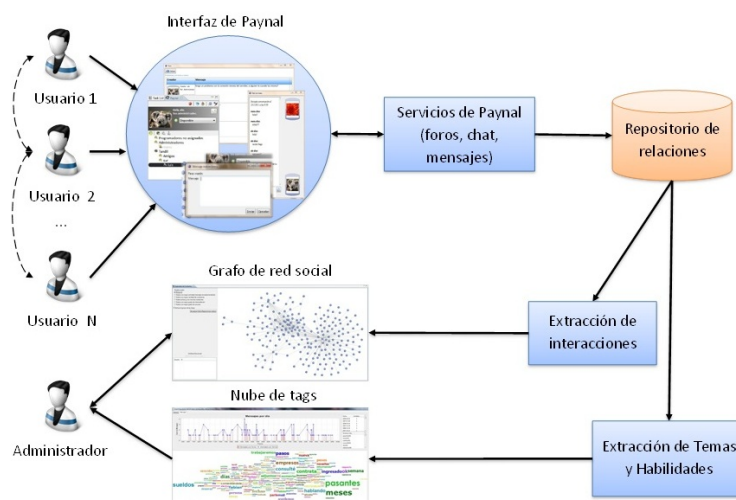


Figura 1. Esquema general del funcionamiento de Paynal.

Las herramientas proporcionadas por Paynal pueden ser útiles a diferentes usuarios de la aplicación para conocer a los integrantes de un equipo de desarrollo. Entre las diferentes conclusiones que se pueden obtener a partir de las mismas se encuentran: la detección de líderes de grupo, la identificación de reemplazantes para desarrolladores que dejan la empresa, el reconocimiento de temas de conversación y habilidades de los usuarios, entre otras.

A continuación, se detallarán las métricas de Análisis de Redes Sociales utilizadas en Paynal. Además, se explicará cómo Paynal detecta temas discutidos entre dos desarrolladores y la representación elegida para visualizar los mismos.

2.1. Análisis de Redes Sociales en Paynal

El ARS es una herramienta que permite obtener información acerca de las interacciones entre individuos. Dado que requiere contar con elementos de naturaleza cualitativa, resulta necesario el empleo de técnicas para ordenar la información de los individuos y representarlos así mediante un grafo o red. Las redes o grafos se constituyen como las herramientas más importantes para representar una red social de manera más ilustrativa.

Paynal proporciona una vista de relaciones en la que se representa la red social subyacente a la aplicación. Dicha representación se realiza mediante un grafo no dirigido construido a partir de las interacciones registradas por los usuarios. En efecto, la entrada de datos para la construcción de esta red social

está constituida por las conversaciones de chat, los mensajes instantáneos y las discusiones en el foro, almacenados en el repositorio de la aplicación.

Los nodos o actores de la red son todos los usuarios de Paynal, desde desarrolladores hasta jefes de proyectos; y los arcos o vínculos corresponden a las interacciones entre ellos. El grosor de los arcos indica el grado de interacción entre dos nodos: cuanto mayor sea la cantidad de interacciones, mayor será el grosor del arco que los une.

Existen distintos enfoques para medir la estructura de las redes sociales: por un lado, el análisis de la estructura general de la red y el grado de integración que la caracteriza; y por el otro, el lugar que cada actor ocupa en la red. Para el análisis de la estructura general, se han desarrollado algoritmos y procedimientos que proporcionan información referida a dicha estructura, como componentes, integración, densidad, unipolaridad y centralización. Para el estudio de la posición del actor, se emplean algoritmos que ofrecen información de centralidad de los actores participantes en la red; los más básicos son grado, cercanía e intermediación.

Los indicadores de centralidad permiten analizar la red conjunta e individualmente arrojando distintos resultados: grado de conectividad de la red, individuos con el mayor o menor número de interacciones, intermediación de actores en las relaciones y la cercanía entre los individuos a través de sus interacciones.

Dentro de la teoría de grafos y del ARS, existen ciertos conjuntos de medidas de centralidad de un vértice en un grafo que determinan la importancia relativa de un nodo en la red [2]. La centralidad representa la contribución de un nodo a la posición que ocupa en la red, sea en forma de importancia, influencia, relevancia o prominencia. Algunas de las medidas de centralidad implementadas en Paynal se describen a continuación:

- Grado de centralización (*Centrality*): es la cantidad de actores a los cuales un individuo se encuentra directamente unido. Cuanto mayor sea este número, mayor importancia tendrá el nodo en la red.
- Grado de intermediación (*Betweenness*): es la frecuencia con la que un actor aparece en los caminos más cortos que conectan a todos los pares de nodos de la red.
- Grado de cercanía (*Closeness*): es la capacidad de un nodo de llegar a todos los actores de la red. Se calcula sumando las distancias más cortas de un actor para llegar a los demás actores, y luego, obteniendo la inversa de dicha suma. Los valores altos de cercanía indican una mejor posición en la red para distribuir información o acceder a la misma.

Otra métrica que se le presenta a los administradores en Paynal es la de Equivalencia estructural (*Structure equivalence*). Dos nodos son estructuralmente equivalentes cuando sus relaciones con el resto de los nodos de la red son idénticas. En otras palabras, dos actores son estructuralmente equivalentes si es posible intercambiarlos sin alterar las propiedades de la red.

Paynal provee diversas perspectivas o vistas del grafo, que permiten resaltar nodos con mayor actividad en cuanto a mensajes enviados y recibidos, nodos con mayor cantidad de conexiones, nodos más activos y con muchas conexiones

(la suma de ambos indicadores), nodos con mayor grado de intermediación, y aquellos con mayor grado de cercanía. Además, al seleccionar un usuario en el grafo, se presentan en una tabla los usuarios con mayor similitud estructural a dicho usuario. En la Figura 2 se puede observar el grafo, resaltando aquellos nodos con mayor grado de centralización.

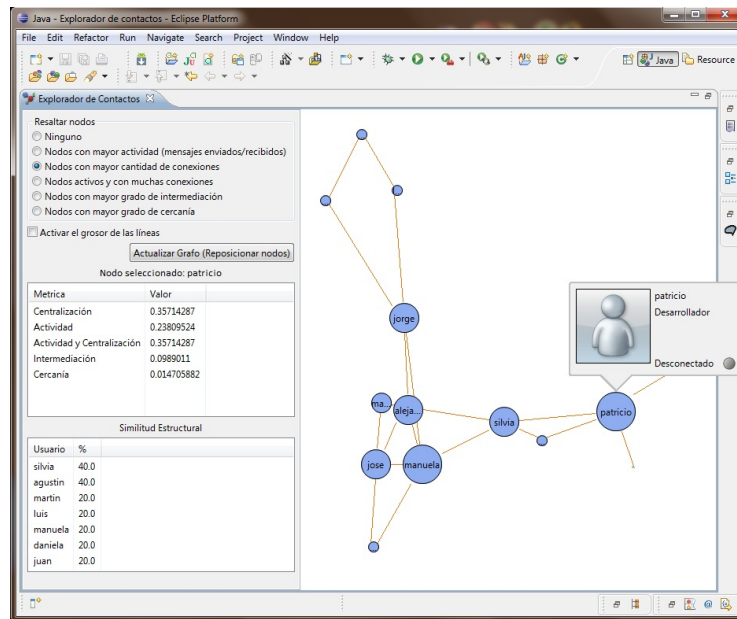


Figura 2. Grafo de la red social subyacente, resaltando los nodos con mayor centralización.

En el Apéndice A se detalla cómo fueron utilizadas las métricas mencionadas para el análisis de una red social creada a partir de interacciones reales.

2.2. Procesamiento inteligente de texto en Paynal

En Paynal, el texto de las relaciones es analizado para extraer información acerca de los términos más relevantes y, de esta manera, permitir al usuario discernir temas y habilidades que poseen los participantes de la interacción. Para realizar este análisis, se emplearon técnicas de Minería de Texto, las cuales tienen como objetivo filtrar progresivamente el texto hasta llegar a los términos más relevantes.

En la Figura 3 se muestra el *pipeline* de análisis utilizado por Paynal para extraer términos de los mensajes directos. En primer lugar se realiza un pre-procesamiento para eliminar texto que, de antemano, se sabe que no aporta información importante. Este proceso es *ad-hoc*, es decir, depende exclusivamente

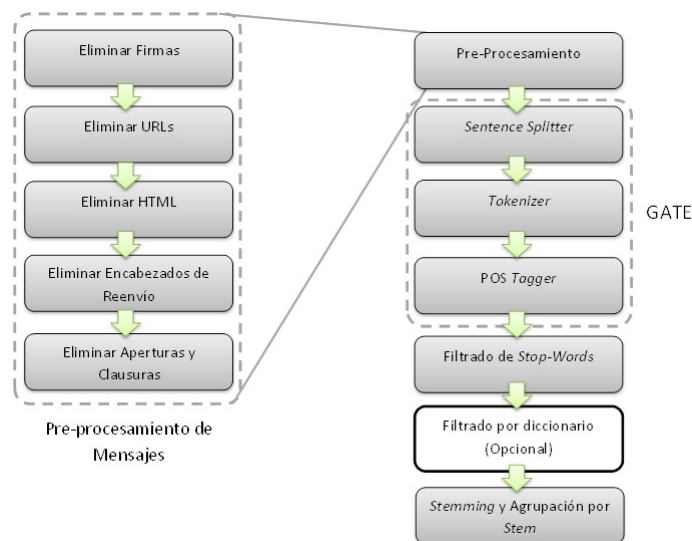


Figura 3. Pipeline de extracción de texto utilizado por Paynal.

del tipo de texto que se está evaluando (mensajes, foros o chats). En el caso de los mensajes directos, este pre-procesamiento consiste de:

- Eliminación de firmas: Las firmas son bloques de texto que se colocan al final de un *email* para brindar información adicional sobre el remitente del mismo. Dicho texto aparece duplicado en todos los mensajes, y por lo tanto, puede afectar de manera negativa el valor de importancia obtenido por cada palabra. Teniendo en cuenta que en lenguaje natural dos líneas de texto con información relevante no suelen estar repetidas, se puede interpretar una firma como un conjunto de líneas que se encuentran repetidas a lo largo del texto. El módulo de eliminación de firmas se encarga de buscar las firmas y eliminar todas, excepto la primera firma encontrada.
- Eliminación de URLs: Las URLs suelen afectar negativamente el desempeño del análisis, agregando palabras como “com”, “www” ó “http”.
- Eliminación de HTML: Este módulo se encarga de extraer el texto que se encuentra entre *tags* HTML. En los *emails* se suele agregar HTML para darle estructura al texto contenido, por ejemplo, agregando marcos con imágenes. Los *tags* HTML afectan negativamente al análisis, por lo tanto, se debe extraer el texto que contienen y eliminarlos de la entrada.
- Eliminación de Aperturas y Clausuras: Se encarga de eliminar las frases de apertura y de clausura más comunes. De esta manera, dichas frases no afectan el desempeño del análisis.

Una vez realizado el pre-procesamiento del texto, se pasa el texto a los módulos que se encargan de asignar un rol gramatical a cada palabra. El primero de ellos, el *Sentence Splitter*, tiene la tarea de delimitar sentencias en el texto. Para

realizar esto, el *Sentence Splitter* busca signos de fin de sentencia tal como “.”, “?”, “:”, y otros. El segundo módulo, el *Tokenizer*, se encarga de marcar *Tokens* en cada sentencia, es decir, marcar palabras, números y símbolos. Para ello, debe buscar espacios (espacios simples, tabulaciones, saltos de línea, entre otros) en el texto o signos de puntuación como “,” y “;”. Por último, se realiza el *Tagging* de Partes del Lenguaje (POST, del inglés *Part-of-Speech Tagging*) cuyo objetivo es asignar o etiquetar la categoría gramatical a cada uno de los *Tokens* en la sentencia. Este proceso se realiza en base a la definición de la palabra o el contexto en el que se encuentre.

Los tres módulos descritos (*Sentence Splitter*, *Tokenizer* y *POS Tagging*), fueron diseñados dentro del marco de la herramienta GATE² (del inglés, *General Architecture for Text Engineering*), una *suite* de herramientas de análisis de texto en Java. GATE provee un entorno de desarrollo para combinar diferentes procesadores de texto (*Splitters*, *Tokenizers*, entre otros) en *pipelines*. A su vez, cada *pipeline* recibe como entrada un *Corpus*, es decir, un conjunto de documentos de texto a ser analizados. Para extraer los términos de las interacciones de Paynal, se diseñó un *pipeline* a partir de procesadores provistos por un *plugin* de GATE, llamado *SpanishPlugin*, el cual provee procesadores especializados para el idioma español. El *Corpus* que se envía a dicho *pipeline* está formado por todas las interacciones que llevaron a cabo 2 usuarios dados.

Una vez obtenidos y categorizados los términos, es posible ignorar aquellos que carecen de significado relevante para el análisis de las relaciones; este tipo de términos o palabras reciben el nombre de *stop words*. Ejemplos de *stop words* son los artículos y las preposiciones, entre otros. Las *stop words* se pueden filtrar antes o después del procesamiento del texto, ya que son fácilmente identificables. El problema de quitarlas antes de analizar el texto, es que el *POS Tagger* no podrá utilizarlas para reconocer el contexto de las palabras y funcionará de forma incorrecta. De esta manera, en este trabajo se utilizó el filtrado de *stop-words* luego de haber clasificado las palabras morfológicamente.

Por último, se utilizó el método de *Stemming* que permite reducir una palabra a su correspondiente raíz. Por ejemplo, las palabras “inteligente” e “inteligencia” se reducen a la raíz “intelligen”. Este método es útil en motores de búsquedas para realizar consultas exhaustivas e indexación. En este caso en particular, el proceso de stemming permite agrupar dos o más términos bajo un único *stem* o raíz y, de esta manera, reducir la cantidad de palabras a mostrar.

Una vez obtenida la lista de términos final, se retorna al cliente para ser mostrado al usuario en una Nube de *Tags*. La Nube de *Tags* utiliza la frecuencia con la que aparecen las palabras para definir el tamaño de cada una de ellas. A continuación se presentará el concepto de Nube de *Tags*, explicando diferentes usos en la actualidad y cómo fue utilizada en Paynal.

² Página Oficial de GATE, <http://gate.ac.uk/>

Tags lo mayor posible, colocando *tags* alrededor de cualquier otro *tag*, siempre que estos *tags* no intersequen. Se alternan los lados donde se debe ubicar el nuevo *tag* y, de esta manera, no se sesga el crecimiento de la nube a un lado en particular, sino que se genera un crecimiento en espiral. Una lista de *tags*, denominada “frontera”, es encargada de almacenar los *tags* que tienen alguno de sus lados libre. A medida que los *tags* de la frontera son utilizados para ubicar otros *tags*, son removidos de la misma. En la Figura 4 se muestra el resultado final de aplicar el algoritmo sobre un conjunto de términos extraídos de una relación.

3. Diseño e Implementación de Paynal

Como se indica en la Figura 5, la arquitectura de Paynal se corresponde con el patrón cliente-servidor, con el agregado de un repositorio donde se almacenan datos de usuario e interacciones. El cliente Paynal se encuentra implementado como un *plug-in* de Eclipse que provee en su mayor parte, funcionalidad visual. Es por este motivo que una de las tecnologías principales utilizadas por el cliente es la biblioteca gráfica SWT⁵ (del inglés, *Standard Widget Toolkit*).

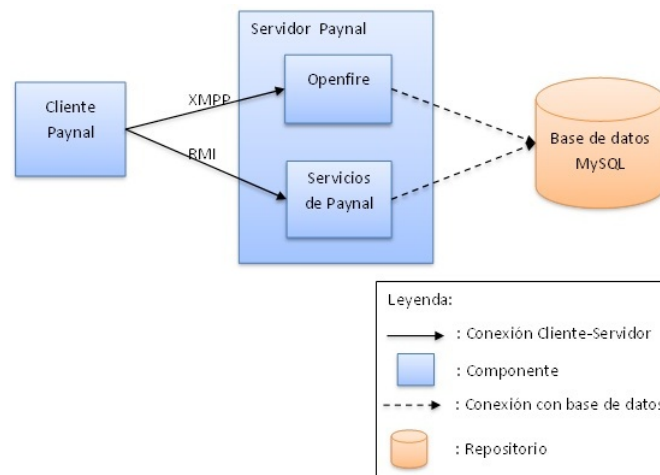


Figura 5. Arquitectura general de Paynal.

El servidor de Paynal provee servicios de administración de usuarios, grupos y proyectos, y de comunicación por chat, foros y mensajes. Posee los servicios necesarios para obtener la Red Social del proyecto y, además, para realizar el análisis de interacciones (extracción de términos). El cliente de Paynal se comu-

⁵ Página oficial de SWT, <http://www.eclipse.org/swt/>.

nica con los diferentes servicios del servidor mediante RMI⁶ (del inglés, *Remote Method Invocation*). RMI es un mecanismo de invocación de métodos remotos en Java que abstrae todos los detalles de la comunicación a bajo nivel, tales como serialización/deserialización de objetos, la ubicación física de los servicios, entre otros. Utilizando RMI, la programación del cliente se torna mucho más sencilla, ya que las invocaciones a métodos de los distintos servicios son similares a las invocaciones a métodos locales.

Por otro lado, la funcionalidad central del sistema de chat, se delegó sobre el servidor libre Openfire⁷, disminuyendo, así, el tiempo de desarrollo de Paynal. Openfire posee varias ventajas, entre ellas, provee una interfaz de administración web, conexión segura mediante SSL/TLS, permite especificar la base de datos donde guardar los mensajes y datos de usuario, entre otras. De esta manera, el desarrollo del sistema de chat se centró en proveer una interfaz dentro del entorno de desarrollo Eclipse y almacenar el historial de chats en el repositorio de Paynal.

Con el objetivo de disminuir la dependencia con la base de datos, se optó por utilizar una capa de abstracción para realizar el mapeo objeto-relacional, entre los objetos persistentes de Paynal y las tablas de la base de datos. La tecnología elegida para éste propósito fue Hibernate⁸, una biblioteca de mapeo objeto-relacional para Java que brinda una mayor portabilidad de la aplicación a cambio de un mínimo overhead de performance. Para ello utiliza archivos XML donde el usuario indica de qué forma se asocian las clases de la aplicación con las tablas del esquema relacional.

A continuación, en la Sección 3.1, se explicará la forma en que Paynal muestra los usuarios del sistema y cómo se administran los mismos. En la Sección 3.2 se detalla cómo fue implementado el grafo de la Red Social y los algoritmos utilizados para calcular las métricas de ARS.

3.1. Administración de usuarios

Paynal provee servicios de administración básicos como son la identificación de usuarios mediante login y diferentes niveles de autorización para realizar acciones. Estos niveles de autorización están asociados al rol que cada usuario cumple en el sistema. Los usuarios de Paynal pueden tener uno de los siguientes roles:

- Administrador: Puede realizar tareas avanzadas en el sistema. Tiene la capacidad de crear o eliminar usuarios, grupos y proyectos. Puede ver todos los proyectos que existen actualmente en el sistema e interactuar con cualquier usuario en el mismo. Puede moderar todos los foros del sistema, es decir,

⁶ Página Oficial de RMI, <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136424.html>.

⁷ Página Oficial de Openfire, <http://www.igniterealtime.org/projects/openfire/>.

⁸ Página Oficial de Hibernate, <http://www.hibernate.org/>.

puede eliminar mensajes de foro que crea incorrectos o incluso foros completos. También es el único usuario al que se le permite usar las herramientas de análisis de redes sociales.

- Jefe de Grupo: Administra un grupo de trabajo. Se le permite crear usuarios y asignarlos solo dentro de su grupo de trabajo. Además, solo puede interactuar con los usuarios que se encuentran asignados a su proyecto.
- Desarrollador. Un desarrollador puede realizar solo las tareas básicas en el sistema, tales como chatear, crear foros dentro de su proyecto o enviar mensajes. Además, solo puede ver el estado de conexión de los demás usuarios dentro de su grupo de trabajo.

Los usuarios del sistema pueden ser asignados a grupos de trabajo, y así mismo, los grupos de trabajo forman parte de proyectos. Esta organización permite asemejar la organización de usuarios en el sistema a aquella que utiliza la empresa de software. Además, facilita la definición de ámbitos de trabajo y así filtrar los elementos visibles a un usuarios según el rol que tiene asignado.

Como se observa en la Figura 6, la interfaz principal de Paynal presenta un árbol donde los usuarios se organizan en grupos de desarrollo y, a su vez, dichos grupos, forman parte de diferentes proyectos. A partir de dicho árbol se pueden realizar las tareas de administración de usuarios, grupos y proyectos. Además, permite iniciar chats y enviar mensajes seleccionando los usuarios que se muestran en el árbol. La barra de herramientas superior, permite cerrar la sesión del usuario actual, abrir la interfaz de foros, iniciar una conferencia, ver logs e iniciar la herramienta de análisis de redes sociales.



Figura 6. Árbol de usuarios de Paynal.

3.2. Análisis de Redes Sociales y Minería de Texto

Para representar el grafo de la Red Social, y así realizar diferentes análisis sobre el mismo, se evaluaron diferentes frameworks gráficos. El framework

que mejor se desempeñó, balanceando facilidad de uso y performance, fue Zest⁹, un framework para visualizar grafos en SWT. Posee varios algoritmos de posicionamiento de nodos, entre ellos, el algoritmo *Spring Layout*, un algoritmo basado en fuerza que trata los nodos como masas que tienen fuerza gravitatoria y carga eléctrica, y los arcos se tratan como resortes con resistencia y elasticidad. Dicho algoritmo es muy útil a la hora de dibujar la Red Social, ya que separa automáticamente subgrafos que no poseen conexiones, mejorando su visualización.

Por otro lado, a la hora de implementar diferentes métricas a ser visualizadas sobre el grafo de Zest, se requería de la utilización de algoritmos de grafos conocidos. Por ejemplo, para calcular el grado de cercanía se utilizó el algoritmo de Dijkstra, ya que obtiene los caminos más cortos desde uno nodo origen. En el caso del grado de intermediación se necesitan los caminos más cortos entre todos los pares de nodos, por lo que se optó por utilizar el algoritmo de Floyd-Warshall. En Paynal se utilizó la biblioteca JGraphT¹⁰, una biblioteca Java que brinda una gran cantidad de algoritmos estándar de grafos.

4. Conclusiones

En este trabajo se presentó una aplicación CSE que provee herramientas *groupware* al entorno de desarrollo Eclipse. Dicha aplicación, a la que se denominó Paynal, permite a sus usuarios interactuar de distintas maneras, desde el diálogo directo, como el chat, hasta la participación en discusiones de foros. Se mostró cómo Paynal, a través de sus servicios, favorece la colaboración y cooperación entre los miembros del equipo de desarrollo.

Por otro lado, se construyó la red social subyacente a Paynal, conformada a partir del conjunto de relaciones establecidas entre los usuarios. La construcción de esta red se realizó empleando grafos no dirigidos, cuyos nodos representan los usuarios y los arcos, los vínculos entre ellos. A partir de ella, se elaboró un análisis que permitió estudiar tanto la conducta de los miembros dentro de la red, como el comportamiento de la estructura en general.

Además, se aplicaron técnicas de Minería de Texto a fin de extraer información acerca de los principales temas tratados en una relación en particular. Entre los procesos utilizados para dicha extracción, se destacan: *Part-of-Speech Tagging*, para averiguar la función gramatical de una palabra (verbo, sustantivo, etc.); y *Stemming*, para reducir cada palabra a su *stem* y así agrupar diferentes palabras bajo un mismo *stem*. Los tópicos extraídos de las relaciones se visualizaron en una Nube de *Tags*, destacando aquellos con mayor frecuencia de ocurrencia. De igual modo, se incluyeron gráficos estadísticos que permiten analizar estos datos cuantitativos de manera más detallada.

4.1. Contribuciones

La principal contribución del presente trabajo pertenece al área de CSE, a partir de la creación de una herramienta que permite a los desarrolladores

⁹ Página Oficial de Zest, <http://www.eclipse.org/gef/zest/>.

¹⁰ Página Oficial de JGraphT, <http://www.jgrapht.org/>.

colaborar sin abandonar su entorno de trabajo. Este tipo de herramientas es escaso en la industria de software, por lo que las organizaciones deben recurrir a herramientas externas al entorno de desarrollo.

Se contribuyó, además, a la relación entre las áreas de CSE y Redes Sociales, permitiendo analizar desde la herramienta la Red Social subyacente formada por la colaboración entre los usuarios. Esto es posible ya que cada interacción entre desarrolladores forma un vínculo social. De esta manera, se construyó un grafo de interacciones donde los nodos representan a los usuarios de la herramienta, y los arcos a las interacciones.

Existen en la actualidad algunos ambientes de desarrollo colaborativo similares a Paynal. Uno de ellos es el proyecto Jazz de IBM, el cual integra programación, comunicación y gestión de proyectos [4]. A diferencia de Paynal, en Jazz el análisis de redes sociales se debe realizar de manera externa, obteniendo los datos través de la API provista por Jazz o directamente de su repositorio [8]. Por su parte, Paynal se enfoca en la integración de herramientas de análisis de interacciones en un entorno de desarrollo colaborativo, con el objetivo de proveer una visión rápida sobre la estructura de la organización.

4.2. Limitaciones

Es claro que la aplicación de la herramienta se encuentra acotada al desarrollo de software, ya que se trata de una aplicación CSE. Otra limitación es que sólo se puede extraer información de interacciones por texto. En el caso de videoconferencias o conversaciones por voz, se debería aplicar algoritmos de reconocimiento de voz y transformar las charlas en texto.

Una de las limitaciones de Paynal se halla en que la extracción de temas y habilidades está perfeccionada para el idioma español. Si bien, la aplicación permite añadir nuevos idiomas al reconocimiento de texto, se deben aplicar reglas y técnicas específicas para el nuevo idioma, lo que conlleva un gran esfuerzo de análisis e implementación.

4.3. Trabajo Futuro

Existen varias mejoras a realizar sobre las herramientas en diferentes aspectos de la misma. Una de ellas es la introducción de más métricas de ARS para brindar a los administradores mejores puntos de vista al analizar la Red Social.

Otra extensión posible es la de mejorar el soporte CSE de la herramienta. Por ejemplo, se podría agregar un diseñador UML colaborativo que permita interactuar a varios diseñadores. Otro ejemplo, sería la inclusión de pizarras compartidas para realizar bocetos entre los desarrolladores. Además, se puede mejorar el conocimiento que poseen los demás usuarios sobre los artefactos que otros usuarios se encuentran modificando. Por ejemplo, incluyendo indicadores gráficos sobre los archivos de código fuente para saber si alguien se encuentra modificando dicho archivo.

La formación de grupos es un tópico importante en las organizaciones, por lo que sería conveniente agregar algoritmos de detección de grupos para recomendar

posibles equipos de trabajo. Esta tarea se puede realizar analizando tanto la estructura de la Red Social, cómo los temas hablados entre los usuarios. Este tipo de algoritmos sirve, además, para comparar los equipos presentes actualmente en la organización.

Referencias

1. L. J. Bannon and K. Schmidt. *CSCW: four characters in search of a context*, pages 3–16. North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, 1991.
2. S. P. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55 – 71, 2005.
3. P.H. Carstensen and K. Schmidt. Computer supported cooperative work: New challenges to systems design. In *In K. Itoh (Ed.), Handbook of Human Factors*, pages 619–636, 1999.
4. L.T. Cheng, S. Hupfer, S. Ross, J. Patterson, B. Clark, and C. de Souza. Jazz: a collaborative application development environment. In *Conference on Object Oriented Programming Systems Languages and Applications: Companion of the 18 th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*. Association for Computing Machinery, Inc, One Astor Plaza, 1515 Broadway, New York, NY, 10036-5701, USA, 2003.
5. C. Cook. Collaborative software engineering: An annotated bibliography. *Department of Computer Science and Software Engineering, University of Canterbury University of Canterbury. Computer Science and Software Engineering*, 0:44, 2004.
6. D. Gallego Vico, I. Martínez Toro, and J. Salvachúa Rodríguez. Generación de contexto colaborativo a partir de herramientas cscw 2.0. *IX Jornadas de Ingeniería Telemática (JITEL 2010)*, 2010.
7. M. Koch, T. Gross, et al. Computer-supported cooperative work-concepts and trends. In *Information Systems and Collaboration: State of the Art and Perspectives*. 11th Conf. of the Association Information and Management (AIM), 2006.
8. T. Nguyen, A. Schroter, and D. Damian. Mining jazz: An experience report. In *Infrastructure for Research in Collaborative Software Engineering. 2008 of Conference*, 2008.
9. J. Whitehead. Collaboration in software engineering: A roadmap. In *2007 Future of Software Engineering*, FOSE '07, pages 214–225, Washington, DC, USA, 2007. IEEE Computer Society.

A. Resultados Experimentales

En la presente sección se describen diferentes maneras de utilizar la información provista por Paynal para analizar la organización tanto a nivel grupal, como a nivel individual. Para fortalecer la veracidad del análisis, se utilizan datos de una empresa de desarrollo de software.

A.1. Datos experimentales

Los datos experimentales corresponden a un grupo de personas trabajando en una empresa de desarrollo de software. Dichos datos provienen de una copia de respaldo realizada desde *Oracle Collaboration Suite* compuesta en su totalidad por *emails*. Los mismos están compuestos aproximadamente por 833 usuarios y 9827 mensajes enviados entre los mismos.

Es evidente que la copia de respaldo no puede ser importada de manera directa en Paynal. Los archivos generados por Oracle Team Suite, deben ser tratados con el objetivo de transformarlos en mensajes entre usuarios de Paynal.

La copia de respaldo está compuesta, a su vez, de los respaldos de los espacios de trabajo de cada uno de los empleados. Cada uno de dichos respaldos de usuario se compone por:

- Un conjunto de archivos cuyos nombres están formados por el nombre de usuario, un guión bajo, y un número que identifica unívocamente a una carpeta de *emails* en el espacio de trabajo. Cada archivo contiene los *emails* correspondientes a cada carpeta.
- Un archivo con extensión “foldermap” y cuyo nombre se corresponde con el nombre de usuario al cual pertenece. Este tipo de archivo contiene las asociaciones entre los números identificadores de los archivos y el de la carpeta original.

De esta manera, utilizando los archivos con extensión “foldermap” se puede realizar un primer filtrado de mensajes, descartando, por ejemplo, aquellos mensajes que se encuentren en las carpetas con nombre “Spam” o “Correo Basura”. A su vez, cada archivo contiene un conjunto de *emails* en formato Internet Mail (el formato estándar de mails en internet) y sus datos internos (el texto y los archivos adjuntos) se encuentran codificados en formato MIME (del inglés, *Multipurpose Internet Mail Extensions*).

Para realizar la importación de forma programática, se creó una aplicación que obtiene los *emails* a partir de los archivos “foldermap”. Una vez obtenido cada conjunto de *emails* concatenados, se realiza una división en *emails* separados. Utilizando la biblioteca estándar “javax.mail” y más específicamente la clase *MimeMessage* para extraer la siguiente información:

- El campo *From* del mensaje, que contiene la dirección *email* de origen.
- Los campos *To* y *Cc* del mensaje, que contiene las direcciones *email* de destino.

- Aquellas partes del mensaje que contengan texto (corresponden a las partes del mensaje del tipo “text”, por ejemplo, “text/plain” o “text/html”).

A partir de las direcciones encontradas, se crean usuarios de Paynal de manera automática, y luego, se crean mensajes directos entre los usuarios involucrados. Se utilizó el mecanismo de mensajes directos, ya que éste posee la mayor similitud con la estructura de un *email*.

Una vez cargados todos los mensajes directos, se puede comenzar a utilizar Paynal con los datos importados. La Red Social formada a partir de los datos experimentales se presenta en la Figura 7. En la misma sólo se aprecia una fracción de las interacciones que se encuentran en el repositorio.

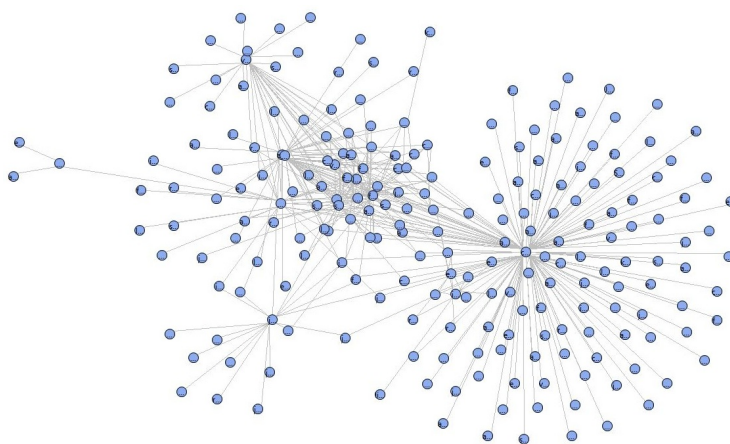


Figura 7. Grafo de la Red Social formada por los datos experimentales.

Esto se debe a que los datos completos se traducen en un grafo de 812 nodos y 2110 interacciones. Dicho grafo se torna visualmente complejo (los nodos se agolpan debido a la falta de espacio), por lo que se optó por incluir en la Red Social sólo aquellas interacciones que tengan determinada “fuerza”, es decir, que entre los usuarios involucrados en cada una de ellas se hayan enviado una cierta cantidad de mails. Esta decisión partió del hecho de que, de las 2110 interacciones, 847 de ellas, es decir, el 40%, corresponden a interacciones con un solo mensaje enviado.

De esta manera, se filtraron los mails utilizando el criterio de sólo representar aquellas interacciones que posean más de N mensajes. En la Tabla 1 se puede ver como diferentes cotas de mensajes representan diferentes cantidades de nodos, y el porcentaje del total que dicha cantidad representa.

Las métricas aplicadas sobre la Red Social no se ven afectadas en gran medida por la reducción de interacciones de menor fuerza. En general, se observó que hasta con el filtrado de 10 mensajes el análisis de las métricas brinda resultados similares, por lo que se tomó dicho valor como cota máxima de filtrado. Cabe

| Cantidad mínima de mensajes | Número de interacciones | Número de nodos involucrados | Porcentaje del total de interacciones |
|-----------------------------|-------------------------|------------------------------|---------------------------------------|
| 1 | 2110 | 812 | 100 % |
| 2 | 1263 | 515 | 60 % |
| 3 | 993 | 413 | 47 % |
| 4 | 801 | 351 | 38 % |
| 5 | 685 | 307 | 32 % |
| 10 | 221 | 435 | 10 % |
| 20 | 143 | 237 | 6 % |

Tabla 1. Cantidad de nodos e interacciones restantes luego de filtrar por número de mensajes.

aclarar que, a los fines del análisis, el mismo se puede realizar con el número total de nodos obteniendo directamente los valores concretos de los nodos en cada métrica, omitiendo la representación del grafo.

Una vez introducidos los mensajes de la organización en el repositorio de Paynal, se puede comenzar a trabajar utilizando el grafo de la Red Social. Se pueden realizar varios análisis según ciertos objetivos de la organización utilizando las métricas provistas.

Se presentan 4 objetivos a alcanzar desde la organización:

- Identificar posibles líderes de grupo.
- Controlar el flujo de información.
- Colocar monitores de información, o averiguar cuáles nodos pueden funcionar como tales.
- Reemplazar un desarrollador con otro sin perder conectividad en la red.

A.2. Identificar posibles líderes de grupo

En la búsqueda de posibles líderes de grupo, se pueden tener en cuenta 2 indicadores: que el nodo posea muchas conexiones y que sea activo en la organización. Para el primero se utiliza la métrica de grado de centralización. Como ocurre con otras métricas, el tamaño de los nodos permite distinguir los diferentes valores obtenidos por cada actor. De esta manera, se observa que el actor con mayor número de conexiones directas es Carolina, seguido por el actor Alejandro. De esta manera, Carolina contribuye fuertemente en la Red Social, tratándose de un usuario clave para la interconexión de la red.

Se podría concluir que dicho nodo puede llevar a cabo la administración del grupo de personas con las que se conecta, sin embargo, se debe tener en cuenta qué tanto se relaciona con cada uno de los usuarios. Por este motivo, en Paynal se pueden visualizar los nodos más activos y, además, se provee un indicador mixto que combina los indicadores de actividad y centralización. De esta manera, se puede decir, que un nodo que posee un alto valor en el indicador mixto es un

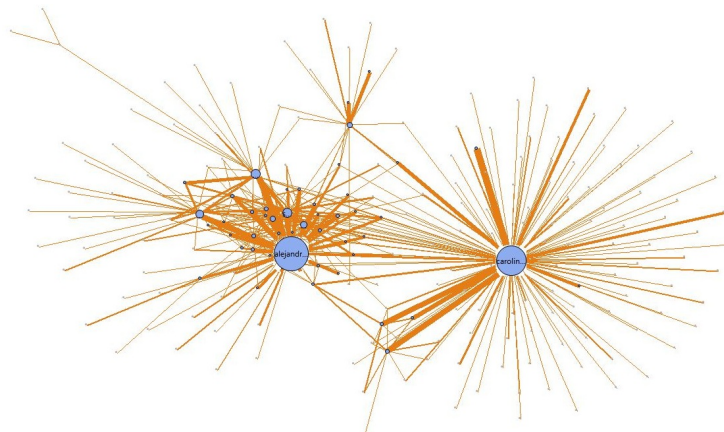


Figura 8. Grado de centralización y actividad aplicados de forma conjunta sobre el grafo.

posible líder de grupo. En la Figura 8 se puede observar el indicador mixto de centralización y actividad aplicado al grafo. En él se puede observar cómo tanto Carolina como Alejandro obtuvieron altos valores en ambas métricas por lo que pueden desempeñarse como líderes de los grupos.

| | Centralización | Actividad | Centralización y Actividad |
|-----------|----------------|-----------|----------------------------|
| Alejandro | 0,2909 | 0,3467 | 0,4704 |
| Carolina | 0,65 | 0,2861 | 0,5935 |

Tabla 2. Valores de centralización, actividad y el indicador mixto para los nodos Alejandro y Carolina

En la tabla 2 se pueden observar los valores obtenidos en las métricas de centralización, actividad, y el indicador mixto de centralización y actividad; tanto por Alejandro como por Carolina.

A.3. Controlar el flujo de información

Para controlar el flujo de información que sale o entra a la Red Social, se puede intentar encontrar aquellos nodos por los que fluye mayor cantidad de información. La métrica a utilizar es la de grado de intermediación. El grado de intermediación de un nodo, como se mencionó antes, se calcula observando qué cantidad de caminos más cortos entre todos los pares de nodos pasan por el nodo en cuestión.

En la Figura 9, Carolina es el nodo con mayor grado de intermediación. Si bien no se aprecia en la misma, le siguen más alejados, con valores similares

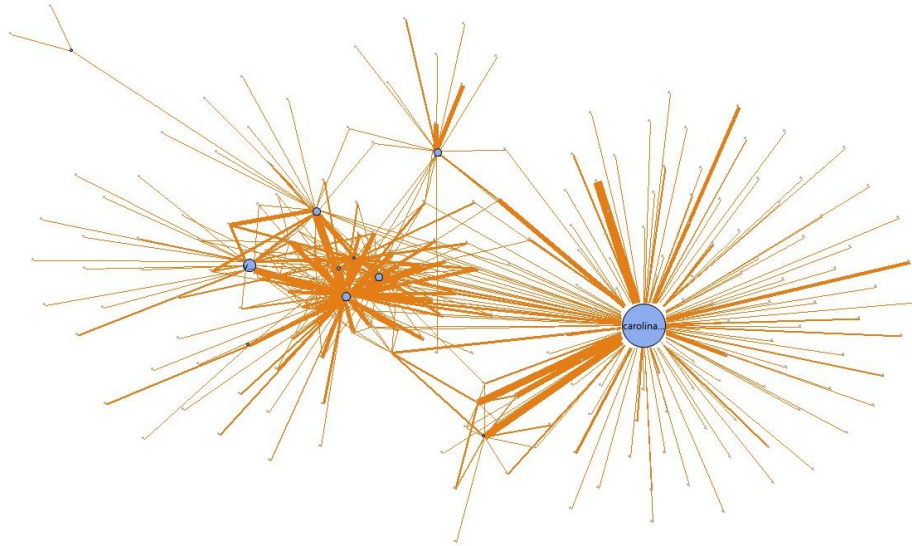


Figura 9. Grado de intermediación visualizado en el grafo de Paynal.

entre ellos, los nodos correspondientes a los usuarios Federico, Vanesa, Juan, Alejandro y Germán.

| | Grado de intermediación |
|-----------|-------------------------|
| Carolina | 0,0839 |
| Vanesa | 0,0122 |
| Federico | 0,0078 |
| Alejandro | 0,0057 |
| German | 0,0036 |
| Juan | 0,0034 |

Tabla 3. Valores obtenidos en la métrica de grado de intermediación.

El control sobre qué datos fluyen hacia adentro o hacia afuera de la red se puede realizar en los nodos antes mencionados. Este tipo de nodos tiene una posición de privilegio al conectar diferentes partes de la red; sin embargo, sufren de una desventaja. En general, mientras mayor cantidad de caminos cortos pasen por un nodo intermediario, es decir, mientras mayor grado de intermediación posea, mayor será el riesgo de desconexión de ciertas partes de la red. El extremo de dicha situación se daría en el caso de nodos “puente” que conectan dos redes distintas, o bien, dos sectores de una red, a partir de sus únicos dos arcos. La supresión de este actor dejaría ambas redes o sectores sin comunicación alguna.

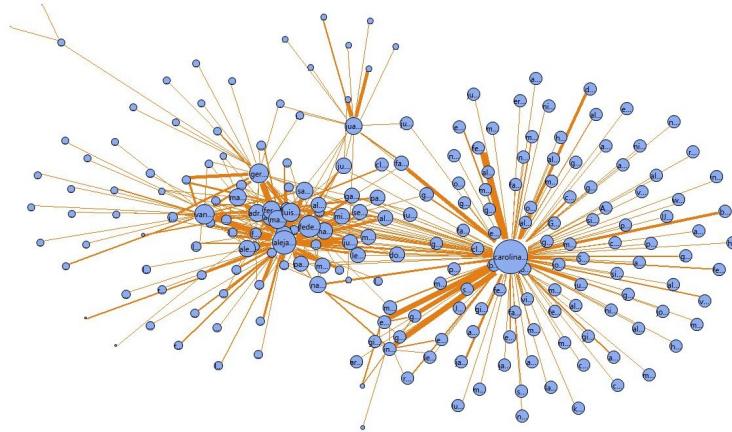


Figura 10. Grado de cercanía visualizado en la Red Social de Paynal.

Un ejemplo de nodo “puente” es Germán (nodo que conecta el sub-grafo superior izquierdo compuesto por tres nodos, con el resto de la red).

En la Tabla 3 se pueden observar los valores de intermediación obtenidos por Federico, Vanesa, Juan, German, Alejandro y Carolina.

A.4. Colocar monitores de información

| | Grado de Cercanía |
|-----------|-------------------|
| Carolina | 0,0016 |
| Alejandro | 0,0013 |
| Federico | 0,00125 |
| Vanesa | 0,00119 |
| German | 0,00118 |
| Luis | 0,001152 |
| Mariano | 0,001146 |
| Fernando | 0,001131 |
| Analia | 0,001116 |
| Adrian | 0,001111 |
| Sandra | 0,001108 |
| Juan | 0,001106 |

Tabla 4. Valores obtenidos en la métrica de grado de cercanía.

Existen nodos en la red donde se puede monitorear de manera mucho más sencilla toda la información que fluye a través de la misma. Este tipo de nodos

tienen los caminos más cortos a todos los demás nodos de la red. Para averiguar qué tipo de nodos tienen dicha característica, se utiliza la métrica de grado de cercanía.

En la Figura 10, se muestran los nodos con mayor grado de cercanía. Como se puede observar, los nodos donde se puede monitorear de manera más efectiva la red son Carolina, Alejandro, Federico y Vanesa. Le siguen no muy separados los nodos German, Juan, Luis, Mariano, Fernando, Sandra y Analía.

En la Tabla 4 se pueden observar los valores obtenidos en la métrica de cercanía por cada uno de los nodos mencionados.

A.5. Identificar reemplazos para un desarrollador

Cuando un nodo se elimina porque el usuario que lo representa deja la organización, se debe intentar mantener la conectividad que dicho usuario brindaba a la red. Esto se logra buscando un nodo que se conecte a los mismos nodos que el nodo a reemplazar, esto es, se busca un nodo con alta similitud estructural. Idealmente, los nodos comparten todos los nodos con los que se conectan obteniendo 100 % de similitud estructural.

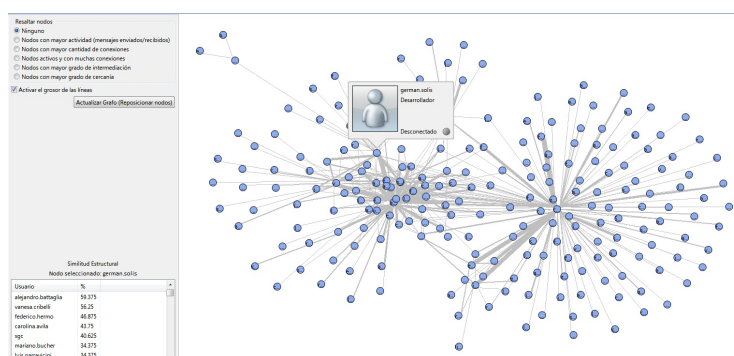


Figura 11. Similitud estructural visualizado para el nodo Germán.

La Figura 11 muestra la selección del nodo Germán y la tabla de similitud estructural asociada. Se puede observar que el usuario Alejandro es el que más se asemeja a Germán, por lo que sería el mejor reemplazo ante un posible retiro. Luego, lo siguen Vanesa y Federico con porcentajes no muy distantes.

Es evidente que al reemplazar un nodo por otro que no posee un 100 % de similitud estructural, se deben volver a entablar las relaciones con aquellos nodos no compartidos.

En el análisis del reemplazo de un nodo, se deben tener en cuenta, además, las aptitudes del nodo a reemplazar. Para ello, se puede analizar los temas del nodo reemplazante y compararlos con el del nodo a reemplazar. También, se

puede observar la nube de tags de cada nodo y obtener conclusiones sobre los temas que tratan.

| | Nodos con mayor similitud estructural |
|-----------|--|
| Carolina | Alejandro(20 %) y Federico(20 %) |
| Alejandro | Federico (64 %) y Carolina (45 %) |
| Vanesa | Alejandro (61 %), German (44 %) y Federico(44 %) |
| Germán | Alejandro(56 %), Vanesa(53 %), Federico(43 %) y Carolina(40 %) |
| Federico | Alejandro (93 %) y Carolina (65 %) |

Tabla 5. Nodos con mayor similitud estructural para los nodos más relevantes de la red.

En la Tabla 5 se pueden observar los nodos que tienen mayor similitud estructural para cada uno de los actores relevantes de la Red Social. Los actores más relevantes son aquellos que obtuvieron buenos valores en todas las métricas; éstos son: Carolina, Alejandro, Vanesa, Germán y Federico. Como se puede observar en la tabla, un nodo puede reemplazar a otro nodo teniendo alta similitud estructural, pero no siempre se cumple la inversa. Por ejemplo, Carolina puede reemplazar con cierta facilidad a Federico, ya que se comunica con el 65 % de los nodos de Federico. Sin embargo, Federico es poco probable que pueda reemplazar a Carolina, ya que sólo comparte la comunicación con el 20 % de los nodos de Carolina.