



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ÁREA TÉCNICA

TITULACIÓN DE INGENIERO EN SISTEMAS INFORMÁTICOS Y
COMPUTACIÓN

Análisis de redes sociales aplicados a un RDF Store

TRABAJO DE FIN DE TITULACION

AUTOR: Quichimbo Armijos, Pricila Vanessa

DIRECTOR: López Vargas, Jorge Afranio, Ing.

LOJA – ECUADOR

2014

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE FIN DE TITULACIÓN

Ingeniero.

Jorge Afranio López Vargas.

DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de fin de titulación: "Análisis de redes sociales aplicados a un RDF Store" realizado por e: Quichimbo Armijos Pricila Vanessa, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la documentación del mismo.

Loja, abril de 2014

f).....

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

“Yo Quichimbo Armijos Pricila Vanessa declaro ser autor (a) del presente trabajo de fin de titulación: Análisis de redes sociales aplicados a un RDF Store, de la titulación de Ingeniera en Sistemas Informáticos y Computación, siendo Jorge Afranio López Vargas director del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 67 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado que se realicen a través, o con el apoyo financiero, académico o institucional (operativo) de la Universidad”

f.

Autor: Quichimbo Armijos Pricila Vanessa

Cédula: 1104372626

DEDICATORIA

A mis padres Juan y Madamny.

Mis hermanos: Juan Carlos, Luis Alfredo y María José

AGRADECIMIENTO

A Dios por haberme regalado la vida.

A mis padres que son mi ejemplo y motivación.

A mis hermanos por estar siempre conmigo, apoyándome.

A mis amigos por ser parte de mi vida y por brindarme muchas alegrías.

A mis profesores por enseñarme las bases para un futuro profesional.

A mi tutor de Tesis por compartir sus conocimientos y guiarme en el desarrollo.

ÍNDICE DE CONTENIDOS

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE FIN DE TITULACIÓN.....	II
DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS	III
DEDICATORIA	IV
AGRADECIMIENTO	V
ÍNDICE DE CONTENIDOS.....	VI
ÍNDICE DE TABLAS	IX
ÍNDICE DE ILUSTRACIONES	X
ÍNDICE DE ECUACIONES	XII
RESUMEN.....	XIII
ABSTRACT	XIV
INTRODUCCIÓN.....	XV
ESTADO DEL ARTE	17
1.1. Análisis de Redes Sociales.....	18
1.2. Origen del SNA.....	18
1.3. ¿Por qué estudiar o investigar redes sociales?.....	18
1.4. ¿Cómo se realiza un estudio de una red social?	19
1.4.1. Estudiando las características generales.....	19
1.4.2. Estudiando las posiciones de los actores	19
1.4.3. Estudiando los grupos que tiene.....	20
1.4.4. Visualización.....	20
1.5. Medidas de SNA.....	20
1.5.1. Densidad (Density)	20
1.5.2. Centralidad (Degree Centrality)	20
1.5.3. Cercanía (Closeness Centrality)	22
1.5.4. Intermediación (Betweenness Centrality).....	23
1.5.5. Vector Propio (Eigenvector Centrality)	24
1.5.6. Distancia.....	24
1.5.7. Flujo (Flow Betweeness)	25
1.5.8. Poder (Bonacich power)	25
1.6. Tecnología de representación del conocimiento – RDF	25
1.6.1. Definición.....	25
1.6.2. Descripción.....	26
1.6.3. Notaciones de RDF	26
1.6.4. Jena	27

1.7.	SPARQL	28
1.8.	Twitter.....	29
1.8.1.	Definición.....	29
1.8.2.	Funcionamiento	29
1.8.3.	Metadatos de Twitter	32
1.8.4.	API de Twitter	37
1.9.	AllegroGraph	37
1.9.1.	Definición.....	37
1.9.2.	Características.....	38
1.9.3.	Arquitectura	40
1.9.4.	Tecnologías semánticas compatibles.....	42
1.9.5.	Requisitos del sistema.....	44
1.10.	Otras herramientas que soportan RDF y SNA.....	44
1.10.1.	Gephi	44
1.10.2.	Proyecto R	46
1.11.	Aplicaciones que utilizan RDF y Allegrograph	47
1.12.	Trabajos relacionados que aplican RDF y SNA.....	48
	VOCABULARIO DE TWITTER	50
2.1.	Vocabularios.....	51
2.1.1.	DC (Dublin Core).....	52
2.1.2.	SIOC (Semantically-Interlinked Online Communities).....	52
2.1.3.	MOAT (Meaning of a tag).....	53
2.1.4.	GEO	53
2.1.5.	GN (Geo Names).....	53
2.1.6.	FOAF (Friend of a friend).....	54
2.1.7.	TW (Twitter).....	54
2.1.8.	RDF Schema	55
2.1.9.	OWL (Web Ontology Language).....	55
2.2.	Conceptualización	56
2.3.	Namespaces.....	62
2.4.	Ontología.....	63
2.5.	Creación de RDF	64
2.5.1.	Ontología/Vocabulario	64
2.5.2.	Instanciación.....	64
	ANÁLISIS DE REDES SOCIALES EN LAS ONTOLOGÍAS	68
3.1.	SNA en la Web Semántica.....	69

3.2.	Proceso para realizar el análisis de redes sociales en Allegrograph	70
3.3.	SNA en Allegrograph	71
3.4.	SNA en Allegrograph versión 3	72
3.5.	SNA en Allegrograph versión 4 y consultas PROLOG	74
3.6.	SNA en Allegrograph versión 4 y consultas SPARQL	76
3.7.	Función para obtener los resultados	78
	RESULTADOS	79
	CONCLUSIONES	83
	RECOMENDACIONES	86
	TRABAJOS FUTUROS	88
	BIBLIOGRAFÍA	89
	ANEXOS	92
A.	Crear ontología en Protégé	92
B.	Instalar Allegrograph	98
C.	Instalar y configurar gruff	101
D.	Tutorial AGWebView	107
E.	Fórmulas matemáticas de las métricas de centralidad	111
E.1.	Cálculo del grado de centralidad	111
E.2.	Cálculo del grado de cercanía	112
E.2.	Cálculo del grado de intermediación	112
F.	Base de Datos	114
G.	Ejemplo de un tuit según el vocabulario en formato N3 (Notación 3)	115

ÍNDICE DE TABLAS

Tabla 1.- Grado de centralidad	21
Tabla 2.- Descripción de clases	56
Tabla 3.- Descripción de propiedades de objetos.....	57
Tabla 4.- Descripción de propiedades de datos.....	58
Tabla 5.- Prefijos y Namespace	62
Tabla 6.- Resultados de aplicar las métricas de SNA.....	79
Tabla 7.- Características del computador	82
Tabla 8.- Cantidad de datos por tabla	114

ÍNDICE DE ILUSTRACIONES

Ilustración 1.- Grado de entrada y grado de salida.....	21
Ilustración 2.- Cercanía.....	22
Ilustración 3.- Intermediación.....	24
Ilustración 4.- Estructura de una tripleta.....	26
Ilustración 5.- Nombre de usuario y cantidad de Tweets, Following y Followers	30
Ilustración 6.- Usuario y menciones	30
Ilustración 7.- Etiquetas	31
Ilustración 8.- Retweets	31
Ilustración 9.- Contribuyentes (Contributors) – Tweet.....	32
Ilustración 10.- Coordenadas – Tweet.....	32
Ilustración 11.- Fecha de Creación (Created at) –Tweet	33
Ilustración 12.- Etiquetas (Hashtag) – Tweet.....	33
Ilustración 13.- Multimedia (Media) – Tweet.....	33
Ilustración 14.- Enlaces (URL) – Tweet.....	34
Ilustración 15.- Menciones de Usuario (User Mention) – Tweet.....	34
Ilustración 16.- Número de favoritos (Favorite count) – Tweet	34
Ilustración 17.- Número de retuits (Retweet count) – Tweet.....	34
Ilustración 18.- Lenguaje (lang) – Tweet	35
Ilustración 19.- Lugar (Place) -Tweet	35
Ilustración 20.- Código de origen (source) – Tweet	35
Ilustración 21.- Texto (text) – Tweet.....	36
Ilustración 22.- Usuario (user) – Tweet.....	36
Ilustración 23.- Arquitectura Allegrograph.....	41
Ilustración 24.- Arquitectura Allegrograph, Jena, Java DataBase	41
Ilustración 25.- Esquema de conexión a un repositorio	42
Ilustración 26.- Allegrograph Web View.....	43
Ilustración 27.- Allegrograph Gruff.....	44
Ilustración 28.- Ontología.....	63
Ilustración 29.- Arquitectura de SNA en la Web Semántica	70
Ilustración 30.- Proceso para realizar el análisis de redes sociales en Allegrograph	70
Ilustración 31.- Resultados grado de centralidad	80
Ilustración 32.- Resultados grado de cercanía	81
Ilustración 33.- Resultados grado de intermediación.....	82
Ilustración 34.- Error al ejecutar "sna:groupDensity	85

Ilustración 35.- Añadir IRI a la ontología.....	92
Ilustración 36.- Crear namespaces y prefijos	93
Ilustración 37.- Crear recursos	94
Ilustración 38.- Crear Object Property	95
Ilustración 39.- Crear Data Property.....	96
Ilustración 40.- Guardar ontología.....	97
Ilustración 41.- Descomprimir archivo tar.gz	98
Ilustración 42.- Instalar Allegrograph.....	98
Ilustración 43.- Instalar Allegrograph, realizar configuraciones.....	99
Ilustración 44.- Instalar Allegrograph, nombre de la cuenta y contraseña.....	99
Ilustración 45.- Iniciar Allegrograph	100
Ilustración 46.- Comprobar si Allegrograph está instalado correctamente	100
Ilustración 47.- Nuevo triple-store	101
Ilustración 48.- Configurar el nuevo triple-store.....	102
Ilustración 49.- Cargar RDF/XML	102
Ilustración 50.- Abrir triple-store	103
Ilustración 51.- Configurar al abrir un triple-store	104
Ilustración 52.- Mostrar todos las tripletas.....	104
Ilustración 53.- Gráfico de nodos en el gruff.....	105
Ilustración 54.- Query View	105
Ilustración 55.- Table View	106
Ilustración 56.- Loguearse en AGWebView	107
Ilustración 57.- Crear repositorio	107
Ilustración 58.- Repositorio nuevo creado	108
Ilustración 59.- Cargar RDF en AGWebView.....	108
Ilustración 60.- Seleccionar archivo AGWebView.....	109
Ilustración 61.- View Statements.....	109
Ilustración 62.- Query SPARQL AGWebView	110
Ilustración 63.- Grafo de ejemplo	111
Ilustración 64.- Modelo Relacional de la Base de Datos	114

ÍNDICE DE ECUACIONES

Ecuación 1.- Densidad.....	20
Ecuación 2.- Centralidad.....	22
Ecuación 3.- Cercanía	23
Ecuación 4.- Intermediación.....	24

RESUMEN

En este trabajo se presenta la forma de realizar el análisis de redes sociales utilizando Allegrograph en sus dos versiones, herramienta que permite almacenar tripletas RDF (Resource Description Framework) y que a partir de la versión 3 implementa métricas de SNA (Social Network Analysis), entre las principales tenemos: closeness (cercanía), betweenness (intermediación), cliques (grupos), degree centrality (grado de centralidad); los resultados obtenidos permitirán identificar perfectamente los nodos que sirven de intermediarios para transmitir información, los que tienen más vínculos, los nodos que se comunican con el resto de nodos más fácilmente, los más conocidos, es decir, los que son más influyentes en la red. Además, se presenta un vocabulario para Twitter que permitirá expresar la información recolectada desde esta red social en formato de la Web de datos y se describe la manera de efectuar la creación del RDF utilizando Jena.

PALABRAS CLAVES: SNA, RDF, Twitter, Allegrograph, Degree, Closeness, Betweenness.

ABSTRACT

In this work it is presented the way of carrying out the analysis of social networks using AllegroGraph in its two versions, a tool that allows to store RDF triples (Resource Description Framework), and that from version 3 it implements metrics of SNA (Social Network Analysis), among the main, we have: closeness (proximity), betweenness (intermediation), cliques (groups), degree centrality (degree centrality); the results obtained will allow to identify perfectly the nodes that serve as intermediaries to convey information, those with more links, the nodes that communicate with the rest of nodes easily, the most known, that is to say, those that are most influential in the network. In addition, it is presented a vocabulary for Twitter that will allow expressing the gathered information from this social network on the format Web data, and it is described the way of making the creation of the RDF using Jena.

KEYWORDS: SNA, RDF, Twitter, Allegrograph, Degree, Closeness, Betweenness.

INTRODUCCIÓN

El surgimiento de nuevas tecnologías basadas en la Web ha permitido la interacción y comunicación de las personas por medio del envío de mensajes, blogs, wikis donde la colaboración y el intercambio de información juegan un papel importante, además, los usuarios se convierten en creadores de contenido en una comunidad virtual.

El término Web 2.0 comprende aquellos sitios web que facilitan el compartir información y representa la evolución de los sitios web estáticos donde los usuarios se limitaban a la observación pasiva de contenidos hacia las aplicaciones web enfocadas en el usuario final. Entre los servicios de red social que ofrece la Web 2.0 se encuentra Twitter que en la actualidad se ha convertido en la red social de microblogging más popular, su rápido crecimiento radica en su simplicidad y permite a sus usuarios estar en contacto en tiempo real con personas de su interés a través de mensajes breves a los que se les denomina tuits. Twitter maneja una gran cantidad de información relevante para los usuarios, es por ello que se la ha utilizado para realizar el análisis de redes sociales.

El Análisis de las Redes Sociales se ha desarrollado como herramienta de medición y análisis de las estructuras sociales que emergen de las relaciones o de la ocurrencia de determinados eventos entre actores sociales diversos.

El objetivo principal de este proyecto es realizar el Análisis de Redes Sociales en un RDF Store, en este caso Allegrograph¹, utilizando métricas de SNA, entre las principales tenemos: closeness (cercanía), betweenness (intermediación), cliques (grupos), degree centrality (grado de centralidad), para de esa manera obtener los recursos o nodos que son más conocidos en la red, los que se comunican y sirven de intermediarios para transmitir información; en otras palabras, los que son más importantes e influyentes en la red. Es importante destacar que Allegrograph, diseñado para almacenar tripletas RDF (sujeto, predicado, objeto), se considera como una base de datos gráfica que toma en cuenta cada dato almacenado para tener cualquier número de relaciones, las mismas pueden ser vistas como enlaces que en un conjunto forman una red o gráfico y está diseñado para conseguir altas velocidades en las operaciones de carga de tripletas.

La solución se presenta en 3 capítulos que han sido planteados para el desarrollo de la investigación: el primer capítulo es el estado del arte donde se da una introducción a los

¹ Allegrograph.- herramienta para almacenar tripletas RDF que incluye análisis de redes sociales.

conceptos, herramientas y tecnologías semánticas compatibles con el Análisis de Redes Sociales; en el segundo capítulo se trata acerca de la manera de realizar una ontología y luego se describe el proceso de creación de un RDF a partir de una base de datos relacional; el tercer capítulo se detalla la manera de utilizar las métricas de SNA en Allegrograph en sus dos versiones; finalmente se analiza los resultados, se describen conclusiones, recomendaciones y trabajos futuros.

CAPÍTULO 1
ESTADO DEL ARTE

1.1. Análisis de Redes Sociales

Análisis de Redes Sociales o en su traducción al inglés Social Network Analysis (SNA), generalmente estudia la conducta de los actores (individuos, organizaciones, naciones, comunidades, etc) a nivel micro, los patrones de relaciones (amistad, parentesco, religión, afiliaciones, etc) a nivel macro y las interacciones entre los dos niveles, además, describe las estructuras relacionales que surgen cuando los actores colaboran, coinciden o se comunican a través de diferentes procesos o acuerdos, de esta manera la estructura que emerge de la interrelación se traduce en la existencia de una red social (Sanz, 2003).

El concepto de red social parte de un fenómeno que ha sido permanente durante la historia de la humanidad y que ha marcado el desarrollo de todas las estructuras sociales, políticas y económicas, sin embargo, hay que resaltar que el fenómeno de las redes sociales se ha visto magnificado gracias a las nuevas tecnologías (Monsalve, 2008).

Como fruto de las relaciones directas e indirectas entre actores se pueden identificar estructuras relacionales, las mismas que nos pueden ayudar a comprender, predecir e incluso gestionar mejor los resultados de la acción humana (Sanz, 2003).

1.2. Origen del SNA

Históricamente el análisis de redes sociales aparece como una de las primeras disciplinas en utilizar la teoría de grafos, luego de que se estableciera hacer las ciencias sociales algo más formal dando origen a la sociometría². El análisis de redes sociales comenzó creciendo lentamente, a veces a grandes saltos y demostrando cosas peculiares en la sociología de las organizaciones y etnias, ahora ha invadido la investigación sobre la Web y de cómo se hace ciencia (Monsalve, 2008).

Las nuevas tecnologías han potenciado el estudio de las redes como un fenómeno de movimiento social y transformando también las dinámicas simbólicas de la sociedad, promoviendo el desarrollo de las diferentes formas culturales y sus evoluciones y además dando nuevos usos a los sistemas icónicos existentes (Cardozo).

1.3. ¿Por qué estudiar o investigar redes sociales?

Existen varias razones para el estudio de las redes sociales entre las cuales podemos considerar las siguientes: el estudio de cómo se hace y se mide la ciencia que es de interés por gran parte de la comunidad científica; el modelamiento, la simulación y el diseño de

² Sociometría.- método cuantitativo para medir las relaciones sociales.

algoritmos, habilidades y conocimientos clave que han hecho que el análisis de redes sociales escale en gran tamaño; en las ciencias sociales ya que el análisis de redes sociales se utiliza activamente en la sociología, antropología, ciencia política, gestión organizacional, medios de comunicación, etc.

Además, permite a una organización identificar puntos vulnerables o fuertes dentro de la misma.

1.4. ¿Cómo se realiza un estudio de una red social?

Primero se necesita recopilar información fiable y expresarla como grafo³. Luego analizar el grafo para determinar propiedades de la red social original. Según (Monsalve, 2008) existen cuatro maneras de analizar la información.

1.4.1. Estudiando las características generales

- *Redes de mundo pequeño (small world networks)*.- se trata de que las personas estamos a seis grados de separación o seis personas de distancia de todo el mundo según el sociólogo Duncan Watts. Este fenómeno ocurre en las redes que tienen una conectividad especial que hace que la distancia promedio entre dos actores cualquiera sea muy pequeña en comparación con el tamaño (número de actores) de la red (Polanco, 2008).
- *Redes libres de escala (scale free networks)*.- muchas redes como la citación de artículos científicos, la Web y muchas otras tienen una distribución de grados que sigue una ley de potencias o similar, porque si tomamos un subgrafo de esa red, lo más probable es que los grados se sigan distribuyendo como ley de potencias (Universidad de Chile).

1.4.2. Estudiando las posiciones de los actores

El concepto tras la posición de un actor a una red corresponde al acceso que tiene al resto de la red, en principio, se conoce que dos actores ocupan el mismo lugar en la red si comparten los mismos vecinos, pero en general, se desea ir más lejos. Con esta necesidad se definen métricas (ver más adelante Medidas de SNA) para medir la posición de un actor en una red de acuerdo a ciertos criterios.

³ Grafo.- es un conjunto de puntos (vértices) de los cuales algún par de ellos están conectados por líneas (aristas).

1.4.3. Estudiando los grupos que tiene

La detección de comunidades, grupos, cliques⁴ es tema de elevado interés en las redes sociales. La definición es fácil cuando se trata de una estructura formal, es decir, cuando existe un grupo definido o un grupo de adherentes que dicen formar parte del mismo. Todo se vuelve más complicado cuando se trata de una estructura informal, puede ser: un grupo de amigos que son todos o casi todos amigos entre sí, pero ellos a su vez tienen más amigos que no son comunes que desarrollarán redes sociales alrededor de ellos y que les servirá como elemento aglutinante.

1.4.4. Visualización

La visualización sirve como método para descubrir propiedades de ésta, aunque tiene menos peso teórico en el análisis, pero tiene la ventaja de alimentar rápidamente la intuición del investigador.

1.5. Medidas de SNA

1.5.1. Densidad (Density)

La densidad de la red es la proporción de vínculos en la red en relación con el total de vínculos posibles (redes escasas versus redes densas).

Se calcula dividiendo el número de relaciones existentes (-R) entre las posibles. El cálculo total de todas las relaciones posibles se hace multiplicando el número total de nodos por número total de nodos menos uno (Polanco, 2008).

$$D = \frac{R}{N(N - 1)}$$

Ecuación 1.- Densidad

1.5.2. Centralidad (Degree Centrality)

Se define al grado de un nodo como el número de enlaces que posee un nodo, es decir, el número de relaciones que tiene el nodo con los otros nodos. Para los grafos dirigidos⁵ el grado del nodo se divide en dos formas: in-degree y out-degree la suma de estos nos proporciona el grado nodal (degree) para cada uno de los vértices de la red (Velázquez & Aguilar, 2005).

⁴ Clique.- grupo de personas que comparten intereses en común.

⁵ Grafo dirigido.- es un tipo de grafo en el cual el conjunto de aristas tiene una dirección definida. Ej. A --> B

- *In-degree (grado de entrada)*.- es la suma de las relaciones referidas hacia un actor por otros.
- *Out-degre (grado de salida)*.- es la suma de las relaciones que los actores dicen tener con el resto.

Si un actor recibe muchos vínculos a menudo se dice que es prominente o de prestigio, o sea muchos otros actores buscan entablar vínculos con él, y esto puede indicar su importancia. Los actores que inusualmente tienen un alto grado de salida son actores que son capaces de intercambiar con muchos otros, o hacer a muchos otros conscientes de su punto de vista. Los actores que muestran alta centralidad de grados de salida se dice que son actores influyentes (Hanneman).

En la Ilustración 1, se presenta una red de amistad de ejemplo con 7 actores A, B, C, D, E, F, G, donde se podrá apreciar claramente el grado de entrada y de salida de cada nodo.

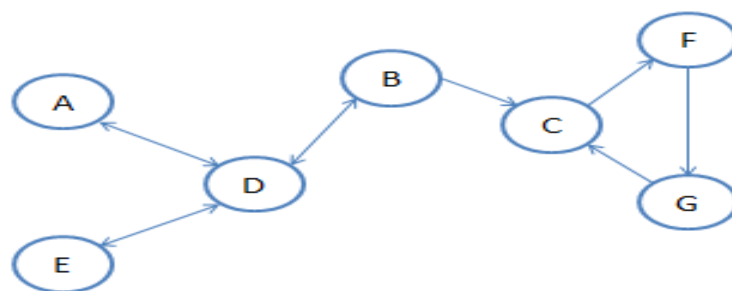


Ilustración 1.- Grado de entrada y grado de salida

Tabla 1.- Grado de centralidad

NODOS	IN-DEGREE	OUT-DEGREE
A	1	1
B	1	2
C	2	1
D	3	3
E	1	1
F	1	1
G	1	1

Con los datos que se muestran en la Tabla 1 se puede apreciar que D tiene una relación de amistad con A, B y E por lo cual su grado de salida es 3, el más alto de todos; en el mismo ejemplo, D es mencionado por A, E y B, por lo tanto su grado de entrada es de 3.

La fórmula matemática para el cálculo del grado de centralidad es la siguiente: (para más detalle revisar el Anexo E)

$$C_d(P_i) = \sum_{k=1}^N a(P_i, P_k)$$

Ecuación 2.- Centralidad
Fuente: (Freeman, 1979)

1.5.3. Cercanía (Closeness Centrality)

Es una condición especial que un actor ejerce un papel claramente central al estar conectado con los otros nodos, tiene la capacidad para llegar en pocos pasos a cualquier nodo de la red (Mejia, 2010).

Este punto de vista de la centralidad, en el que un nodo está próximo a todos los otros nodos del grafo, está relacionado también con el control de la comunicación. Pero como (Freeman, 1979) subraya, se trata de una manera algo diferente a la centralidad basada en la intermediación: aquí un nodo es visto como central en la medida en que puede evitar el posible control de los otros. Una posición central es aquella que no es dependiente de los otros como intermediarios del mensaje.

Matemáticamente se representa como el inverso a la suma de las distancias, Ecuación 3.

En la Ilustración 2, el actor D y C son los nodos que se encuentran conectados con la mayoría de nodos en la red, los nodos que más influencia tiene en la red.

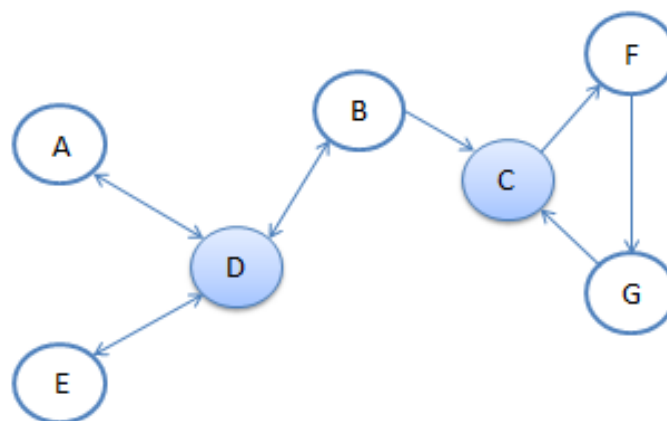


Ilustración 2.- Cercanía

La fórmula matemática para el cálculo del grado de centralidad es la siguiente: (para más detalle revisar el Anexo E)

$$C_c(P_i) = \frac{N - 1}{\sum_{k=1}^N d(P_i, P_k)}$$

Ecuación 3.- Cercanía
Fuente: (Freeman, 1979)

1.5.4. Intermediación (Betweenness Centrality)

La importancia de un actor recae en su intermediación, esta se enfoca en el “control de la comunicación” y se interpreta como la posibilidad que tiene un nodo o actor para intermediar las comunicaciones entre pares de nodos (Velázquez & Aguilar, 2005).

Para este análisis se consideran todos los posibles caminos geodésicos⁶ entre todos los pares posibles, la medida de intermediación de un nodo se obtiene al contar la veces que este aparece en los caminos (geodésicos) que conectan a todos los pares de nodos de la red, a estos actores también se los denomina actores puente.

Es necesario destacar que para que un nodo tenga grado de intermediación por lo menos debe tener un grado de entrada y uno de salida, además, estar en o los caminos geodésicos entre los pares de nodos que se quieren conectar.

La idea que (Freeman, 1979) sugiere es que un individuo puede muy bien estar ligeramente conectado a los otros (es decir, centralidad de grado baja) y sin embargo ser un intermediario esencial en los intercambios.

Según (Polanco, 2008) un nodo puede más fácilmente influenciar el grupo filtrando o distorsionando la información en circulación, este nodo está también en mejor posición para asegurar la coordinación de la red, de todos estos puntos de vista, ocupa una posición central.

Los actores B, C, D como se muestra en la Ilustración 3, son los nodos de intermediación o actores puente de la red porque une dos grupos de nodos, y es el encargado de comunicación de la red. Si no estuviera alguno de ellos se dividiría el grafo.

⁶Caminos Geodésicos.- son las rutas más cortas que un actor debe seguir para llegar a otros nodos.

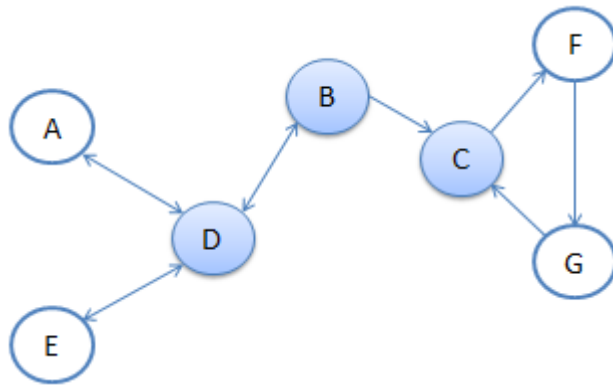


Ilustración 3.- Intermediación

La fórmula matemática para el cálculo del grado de centralidad la siguiente: (para más detalle revisar el Anexo E)

$$C_B(P_k) = \sum_{j=1}^N \sum_{k=1}^N \frac{g_{ij}(P_k)}{g_{ij}}$$

Ecuación 4.- Intermediación
Fuente: (Freeman, 1979)

1.5.5. Vector Propio (Eigenvector Centrality)

Los nodos que poseen un valor alto de esta medida de centralidad están conectados a muchos nodos que a su vez están bien conectados, por lo tanto, son buenos candidatos para difundir información, divulgar rumores o enfermedades, etc. Los nodos más centrales pertenecen a centros de grandes grupos cohesivos⁷ (Rodríguez & Mérida).

1.5.6. Distancia

La distancia nos indica el esfuerzo para que un nodo alcance a otro (Hanneman).

Las distancias entre actores en una red puede ser una importante característica macro de una red en su conjunto. Cuando las distancias son grandes, puede tomar un largo período de tiempo a una información difundirse a través de una población. Puede ser también que algunos actores sean bastante poco conscientes de este hecho y que estén influenciados por otros –incluso si son técnicamente accesibles, el costo puede ser demasiado alto para comportar intercambios (Hanneman).

⁷Grupos cohesivos.- es la medida de intensidad de la interacción social dentro de un grupo.

1.5.7. Flujo (Flow Betweenness)

Indica la posición de intermediación que ocupa cada actor en todos los tipos de contactos y comunicaciones que mantiene con otros actores. Calcula el flujo de intermediación en la red en cada vértice para conocer los valores de centralidad de flujo que tiene toda la red (Rodríguez & Mérida).

1.5.8. Poder (Bonacich power)

Plantea que la centralidad de cada actor está en función de: la cantidad de conexiones que tiene el actor y cuántas conexiones tienen los actores de su vecindario, cuantas menos conexiones tengan los actores de su vecindario más poderoso será el actor. Existen dos tipos de índices de poder (Rodríguez & Mérida):

- *Peso negativo*.- las conexiones con actores bien conectados.
- *Peso positivo*.- las conexiones con actores conectados débilmente.

1.6. Tecnología de representación del conocimiento – RDF

1.6.1. Definición

RDF (Resource Description Framework), en su traducción al español significa “Marco de Descripción de Recursos”, es uno de los pilares fundamentales de la Web Semántica, un lenguaje para describir recursos e identificar unívocamente entidades (organizaciones, lugares, personas, etc) y relaciones y hechos sobre las mismas, además, representa el conocimiento humano en formato digital. Fue impulsado por el creador de la Web Tim Berners Lee y es reconocido y monitorizado por la W3C⁸.

Uno de los aspectos clave es que trabaja con información distribuida, es decir, que las aplicaciones RDF pueden unir archivos RDF publicados en diferentes fuentes de datos, enlazando los documentos que utilicen vocabularios comunes, permitiendo de esa manera flexibilidad al momento de expresar hechos sobre una amplia gama de información.

Esta tecnología es diseñada para representar conocimiento y no para datos, esto significa que RDF es concebido particularmente con significado, todo lo mencionado en RDF significa algo.

⁸ W3C.- World Wide Web Consorsrtium.

1.6.2. Descripción

El objetivo principal de la W3C al crear las especificaciones RDF era crear una fórmula genérica para representar el conocimiento humano, se trataba de crear una sintaxis amplia que permita expresar cualquier hecho y a la vez tan estructurada que cualquier software pudiese interpretarlo de manera automática.

Para conseguirlo la forma más básica que utilizaron es crear tripletas, las mismas que son piezas de conocimiento que tienen la siguiente estructura:

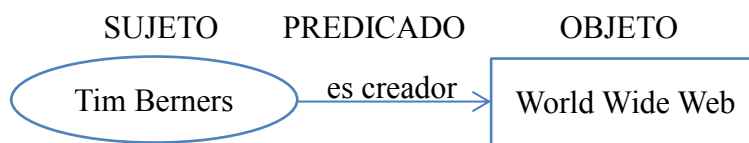


Ilustración 4.- Estructura de una tripleta

En otras palabras, una tripleta es una versión simple de una oración humana que puede ser entendida por las máquinas.

Los sujetos y objetos son recursos referenciables, están identificados mediante una URI⁹ (Identificador Único de Recursos), aunque los objetos pueden ser referenciables o valores literales como: string, date, etc. A continuación se presenta un ejemplo, donde se introducen prefijos para facilitar la lectura de los URI (Apache Jena - RDF).

@prefix vb: <http://www.vocabulario.com/> .			
@prefix dt: <http://www.data.com/> .			
	TRIPLETA		ORACIÓN HUMANA
	Sujeto	Predicado	Objeto
dt:Tim	vb:es_un	dt:cientifico .	Tim es un científico
dt:Tim	vb:es_creador	dt:Web .	Tim es creador de la Web

1.6.3. Notaciones de RDF

Un fichero RDF es un conjunto de tripletas, por tanto puede ser representado en JSON, CSV, XML, incluso pueden ser almacenado en una base de datos con tres columnas. Los

⁹ URI.- Uniform Resource Identifier http://es.wikipedia.org/wiki/Uniform_Resource_Identifier

documentos RDF suelen tener la extensión .rdf, sólo cuando están basados en RDF/XML, existen otras variantes de notaciones utilizadas más a menudo.

- Notation 3

Se la conoce más comúnmente como N3, es diseñada pensando en la legibilidad de los humanos porque es mucho más compacto y fácil de leer. Permite definir espacios de nombres o prefijos que facilitan la lectura.

```
@prefix dt: <http://www.data.com/> .  
@prefix vb: <http://www.vocabulario.com/> .  
dt:Tim   vb:es_creador   dt:Web
```

- N-Triples

Tiene un formato más simple, por lo tanto, es más fácil para el software analizar y generar.

```
<http://www.data.com/Tim>  
    <http://www.vocabulario.com/es_creador>  
    <http://www.data.com/Web> .
```

1.6.4. Jena

Para trabajar con documentos RDF y crear aplicaciones para la Web Semántica existe un conjunto de librerías, siendo Jena uno de los frameworks más utilizados.

Jena es un Framework para Java que provee un API para gestionar grafos RDF, donde los grafos son representados con un modelo abstracto “model”, el mismo que puede tener un origen de datos de archivos, URLs o una combinación de estos (Apache Jena, 2011).

Jena soporta la serialización de grafos RDF a:

- Base de datos relacional
- Turtle
- Notación N3

1.7. SPARQL

SPARQL Protocol and RDF Query Language, es un lenguaje estandarizado para consulta de grafos RDF, donde los desarrolladores y usuarios finales pueden representar y utilizar los resultados obtenidos en las búsquedas a través de una gran variedad de información como datos personales, redes sociales, metadatos sobre recursos digitales como música e imágenes (W3C, W3C Sparql Query Language for RDF, 2008).

Utiliza la comparación de patrones gráficos los cuales están basados en tripletas que poseen variables de consulta que reemplaza al sujeto, predicado y objeto en el término RDF, a continuación se menciona un ejemplo de una consulta simple, para obtener el nombre del creador de la Web.

```
@prefix dt: <http://www.data.com/> .           //Se establece el prefijo "dt" que hace referencia a
                                                "http://www.data.com" para facilitar la lectura de URI

SELECT ?name                                     //Pedirá que se muestre el nombre
{
    ?name dt:creador 'Web' .                    //Selecciona los nombres que tengan como
                                                predicado "dt:creador" y contenga la cadena de
                                                búsqueda "Web"
```

El resultado será el siguiente:

name
Tim

1.8. Twitter

1.8.1. Definición

Uno de los servicios que ofrece la Web 2.0 es Twitter, que en la actualidad se ha convertido en la red social de microblogging más popular, su rápido crecimiento radica en su simplicidad, permite a sus usuarios estar en contacto en tiempo real con personas de su interés a través de mensajes breves a los que se les denomina tweets, que están limitados a 140 caracteres.

En inglés, el vocablo twitter significa «gorjear, trinar, parlotear», de ahí que su símbolo sea un pajarito, y ofrece interesantes posibilidades de comunicación tanto para uso individual como para empresas según (Gil).

Twitter presenta nuevos avances en relación a otros medio de comunicación, debido a que abandona la reciprocidad obligatoria de otras redes sociales, es decir, que no se necesita ser amigo de alguien para seguir sus publicaciones. Una red que permite suscribirse a cuentas de personas o empresas que resultan relevantes, se convierte en un panel de lectura personalizado por cada usuario en función de sus intereses y en un flujo de información constante.

Twitter cuenta con millones de usuarios, por lo tanto, millones de tweets circulan diariamente, es una de las redes más destacadas para encontrar patrones sociales y la gran cantidad de información que almacena es de gran ayuda e interés para el desarrollo de la presente tesis.

1.8.2. Funcionamiento

Twitter pone en contacto a personas con intereses comunes, a través de los posts (tweets) que se publican, cada usuario sigue la actividad de otros (following) y cada usuario es seguido por otros (followers), el usuario es representado con un palabra anteponiéndole el signo de arroba “@”, como se muestra en la Ilustración 5.



Ilustración 5.- Nombre de usuario y cantidad de Tweets, Following y Followers

Es necesario partir de los componentes básicos para crear y disfrutar de este medio de comunicación.

1.8.2.1. Usuario - @

Al usuario de lo identifica con el signo de arroba que se encuentra junto al nombre de usuario de la persona. Si se desea escribir un tweet haciendo referencia a otro usuario, bastará con escribir @ y el nombre de usuario, de esa manera.

En la Ilustración 6, @jorgaf es el autor de tweet, @manuelmax y @nopiedra son los usuarios a los que se hace referencia (menciones) en la publicación.

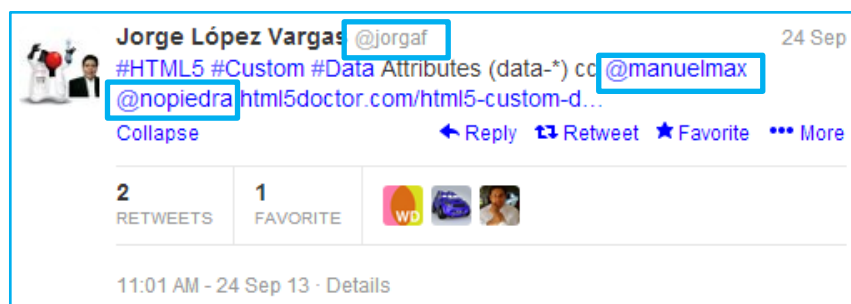


Ilustración 6.- Usuario y menciones

1.8.2.2. Etiquetas -

Cuando se desea destacar una o varias palabras en el tweet o para identificar temáticas, eventos, lugares, etc y convertirla en una categoría global, se escribe la almohadilla delante de la palabra "#". #HTML5 #Custom #Data son las etiquetas o hashtags, si alguien hace una búsqueda con ese término aparecerán los resultados que contengan esa etiqueta como se muestra en la Ilustración 7.

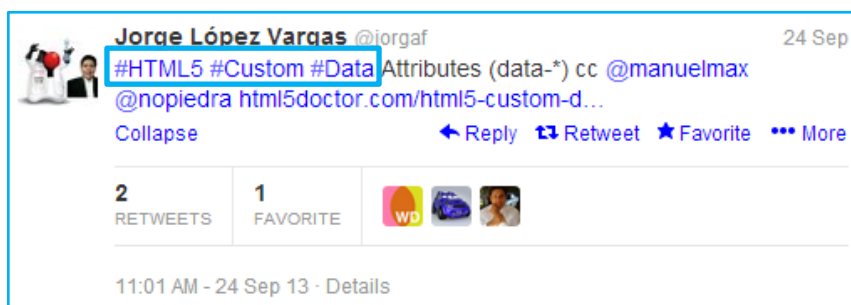


Ilustración 7.- Etiquetas

1.8.2.3. Retweet – RT

Al realizar un retweet le estamos dando un mayor valor a un mensaje y que lo consideramos relevante para compartirlo los seguidores. Existen dos maneras de realizar esta acción, como se muestra en la Ilustración 8.

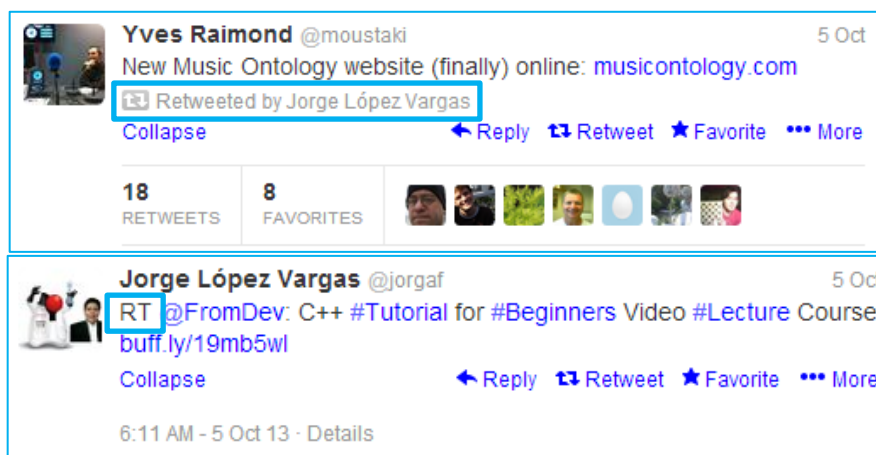


Ilustración 8.- Retweets

En la primera imagen se observa que “Jorge López Vargas” ha retuiteado una publicación de @moustaki, además se puede apreciar que 18 personas han hecho un retweet; en la segunda imagen se observa que a una publicación se le añade las letras “RT” seguido del usuario (@FromDev) y el contenido del tweet principal.

1.8.3. Metadatos de Twitter

Existen alrededor de 40 metadatos que no son visibles para el usuario, pero son de mucha utilidad para quienes realizan aplicaciones o investigaciones de minería de datos debido a que contiene información relevante en cuanto al usuario, el lugar, número de retuits, etc.

A continuación se detalla la información existente detrás de cada tuit, según (Tweet Developers).

1.8.3.1. Contribuyentes (Contributors)

Una colección breve de objetos de usuario, donde se encuentra el identificador único del usuario y el nombre de usuario.

```
"contributors":
[
  {
    "id":819797,
    "id_str":"819797",
    "screen_name":"episod"
  }
]
```

Ilustración 9.- Contribuyentes (Contributors) – Tweet
Fuente.- (Tweet Developers)

1.8.3.2. Coordenadas (Coordinates)

Representa la información geográfica del tuit, según lo informado por el usuario o la aplicación cliente. El primer dato representa la longitud y el segundo la latitud.

```
"coordinates":
{
  "coordinates":
  [
    -75.14310264,
    40.05701649
  ],
  "type":"Point"
}
```

Ilustración 10.- Coordenadas – Tweet
Fuente.- (Tweet Developers)

1.8.3.3. Fecha de Creación (Created at)

Cuándo el tuit fue creado, según el UTC (Tiempo Universal Coordinado).

```
"created_at": "Wed Aug 27 13:08:45 +0000 2008"
```

Ilustración 11.- Fecha de Creación (Created at) – Tweet
Fuente.- (Tweet Developers)

1.8.3.4. Etiquetas (Hashtags)

Representa las etiquetas que se encuentran en el contenido del tuit, se puede observar información como: en qué posición del texto se encuentra la etiqueta.

```
"hashtags": [{"indices": [32, 36], "text": "lol"}]
```

Ilustración 12.- Etiquetas (Hashtag) – Tweet
Fuente.- (Tweet Developers)

1.8.3.5. Multimedia (Media)

Representa elementos multimedia que contiene el tuit, además, el tipo de multimedia, tamaño, url, índices que indican en qué parte del texto de encuentra ubicado.

```
"media": [{"type": "photo", "sizes": {"thumb": {"h": 150, "resize": "crop", "w": 150}, "large": {"h": 238, "resize": "fit", "w": 226}, "medium": {"h": 238, "resize": "fit", "w": 226}, "small": {"h": 238, "resize": "fit", "w": 226}}, "indices": [15, 35], "url": "http://t.co/rJC5Pxsu", "media_url": "http://p.twimg.com/AZVLmp-CIAAbkyy.jpg", "display_url": "pic.twitter.com/rJC5Pxsu", "id": "114080493040967680", "id_str": "114080493040967680", "expanded_url": "http://twitter.com/yunorno/status/114080493036773378/photo/1", "media_url_https": "https://p.twimg.com/AZVLmp-CIAAbkyy.jpg"}]
```

Ilustración 13.- Multimedia (Media) – Tweet
Fuente.- (Tweet Developers)

1.8.3.6. Enlaces (Url)

Representan los URL incluidos en el texto. El *url* nos muestra el URL acortado que hace encajar más fácil las URL largas en el límite de los 140 caracteres, *display url* significa la versión de URL que se muestra al usuario y *expanded_url* es el URL original.

```
"urls": [{"indices": [32, 52], "url": "http://t.co/IOwBrTZR",  
"display_url": "youtube.com/watch?v=oHg5SJ\u2026",  
"expanded_url": "http://www.youtube.com/watch?v=oHg5SJYRHA0"}]
```

Ilustración 14.- Enlaces (URL) – Tweet
Fuente.- (Tweet Developers)

1.8.3.7. Menciones de Usuario (User Mention)

Representa los usuarios mencionados dentro del tuit. *Name* nos indica el nombre que se muestra del usuario referenciado, *screen_name* es el nombre de usuario y *id_str* es el nombre de usuario.

```
"user_mentions": [{"name": "Twitter API", "indices": [4, 15],  
"screen_name": "twitterapi", "id": "6253282", "id_str": "6253282"}]
```

Ilustración 15.- Menciones de Usuario (User Mention) – Tweet
Fuente.- (Tweet Developers)

1.8.3.8. Número de favoritos (Favorite count)

Indica cuántas veces el tuit ha sido marcado como favorito por otros usuarios.

```
"favorite_count": 1138
```

Ilustración 16.- Número de favoritos (Favorite count) – Tweet
Fuente.- (Tweet Developers)

1.8.3.9. Número de retuits (Retweet count)

Indica cuántas veces el tuit ha sido retuiteado por otros usuarios.

```
"retweet_count": 1585
```

Ilustración 17.- Número de retuits (Retweet count) – Tweet
Fuente.- (Tweet Developers)

1.8.3.10. Lenguaje (lang)

Indica el lenguaje detectado por la máquina del idioma del texto, se muestra “und” cuando el lenguaje no ha sido detectado.

```
"lang": "en"
```

Ilustración 18.- Lenguaje (lang) – Tweet
Fuente.- (Tweet Developers)

1.8.3.11. Lugar (places)

Indica de una forma detallada el lugar donde fue emitido el tuit, contiene la siguiente información: dirección de las calles, coordenadas, país, código de país, tipo de lugar, una url donde se encuentra más información acerca del lugar.

```
"place":  
{  
  "attributes": {},  
  "bounding_box":  
  {  
    "coordinates":  
    [[  
      [-77.119759, 38.791645],  
      [-76.909393, 38.791645],  
      [-76.909393, 38.995548],  
      [-77.119759, 38.995548]  
    ]],  
    "type": "Polygon"  
  },  
  "country": "United States",  
  "country_code": "US",  
  "full_name": "Washington, DC",  
  "id": "01fbe706f872cb32",  
  "name": "Washington",  
  "place_type": "city",  
  "url":  
  "http://api.twitter.com/1/geo/id/01fbe706f872cb32.json"  
}
```

Ilustración 19.- Lugar (Place) -Tweet
Fuente.- (Tweet Developers)

1.8.3.12. Código de origen (Source)

Se utiliza para publicar el tuit, como una cadena con formato HTML.

```
"source": "\u003Ca  
href=\"http://itunes.apple.com/us/app/twitter/id409789998?  
mt=12\" rel=\"nofollow\" \u003ETwitter for Mac\u003C/a\u003E"
```

Ilustración 20.- Código de origen (source) – Tweet
Fuente.- (Tweet Developers)

1.8.3.13. Texto (text)

El texto actual en formato UTF-8 de las actualizaciones de estado.

```
"text":"Tweet Button, Follow Button, and Web Intents javascript  
now support SSL http:\\\\t.co\\9fbA0oYy ^TS"
```

Ilustración 21.- Texto (text) – Tweet
Fuente.- (Tweet Developers)

1.8.3.14. Usuario (User)

Información del usuario quien posteó el tuit como: número de tuits, url de la imagen de perfil, nombre de usuario, número de seguidores, número de personas a quien sigue, descripción, ubicación, identificador único de usuario, color de fondo del perfil, fecha de creación.

```
"user":{"statuses_count":3080, "favourites_count":22,  
"protected":false, "profile_text_color":"437792",  
"profile_image_url": "...", "name":"Twitter API",  
"profile_sidebar_fill_color":"a9d9f1", "listed_count":9252,  
"following":true, "profile_background_tile":false,  
"utc_offset":-28800, "description":"The Real Twitter API. I  
tweet about API changes, service issues and happily answer  
questions about Twitter and our API. Don't get an answer? It's  
on my website.", "location":"San Francisco, CA",  
"contributors_enabled":true, "verified":true,  
"profile_link_color":"0094C2", "followers_count":665829,  
"url":"http:\\\\dev.twitter.com", "default_profile":false,  
"profile_sidebar_border_color":"0094C2",  
"screen_name":"twitterapi", "default_profile_image":false,  
"notifications":false, "display_url":null,  
"show_all_inline_media":false, "geo_enabled":true,  
"profile_use_background_image":true, "friends_count":32,  
"id_str":"6253282", "entities":{"hashtags":[], "urls":[],  
"user_mentions":[], "expanded_url":null, "is_translator":false,  
"lang":"en", "time_zone":"Pacific Time (US & Canada)",  
"created_at":"Wed May 23 06:01:13 +0000 2007",  
"profile_background_color":"e8f2f7", "id":6253282,  
"follow_request_sent":false,  
"profile_background_image_url_https": "...",  
"profile_background_image_url": "...",  
"profile_image_url_https": "..."} }
```

Ilustración 22.- Usuario (user) – Tweet
Fuente.- (Tweet Developers)

1.8.4. API¹⁰ de Twitter

El servicio API ha sido utilizado por muchos investigadores, empresas, y las instituciones gubernamentales que desean extraer conocimientos de conformidad con un conjunto diverso de preguntas relativas a los medios de comunicación social. (Morstatter, Pfeffer, Liu, & Carley, 2013) mencionan que el inconveniente esencial de la API de Twitter es la falta de documentación sobre qué y cuánto de datos los usuarios pueden obtener.

Los datos de los medios de comunicación social es a menudo difícil de obtener porque la mayoría de los sitios restringen el acceso a sus datos. El “API de Twitter Streaming” es una capacidad proporcionada por Twitter que permite que cualquiera pueda recuperar información al proporcionar parámetros como: palabras clave (palabras, frases o hashtag), límite geográfico y el ID del usuario.

Los hashtag son importantes en la comunicación en Twitter ya que los usuarios los emplean para resaltar los contenidos que producen, lo que permite que otros usuarios encuentren sus tuits y facilitar la comunicación en la plataforma. La adición de un hashtag a un tuit es equivalente a unirse a una comunidad de usuario que discuten sobre el tema, además, los hashtag son también utilizados por Twitter para calcular los trending topics del momento que anima a los usuarios para publicar en estas comunidades (Morstatter, Pfeffer, Liu, & Carley, 2013).

1.9. AllegroGraph

1.9.1. Definición

AllegroGraph es una base de datos y el marco de aplicación para la creación de aplicaciones de Web Semántica. Es capaz de almacenar los datos y metadatos como triples; consultar estos triples a través de varios APIs consulta como SPARQL (el lenguaje de consulta estándar W3C) y Prolog (Franz, AllegroGraph 4.10 Social Networking Analysis, 2013).

Presenta eficiencia para las consultas de base de datos relacional de rango y soporta los tipos de dato: fecha, hora, longitud, latitud, número de teléfono, etc.

Los clientes utilizan a esta herramienta como una base de datos de eventos con el análisis de redes sociales (relaciones entre actores) y el razonamiento geo-espacial (lugar: dónde

¹⁰ API (Application Programming Interface).- con junto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

algo sucedió o dónde algo sucederá) y temporal (tiempo: cuándo o en qué orden algo sucederá) (Franz, AllegroGraph for Social Network Analysis).

Está diseñado para conseguir altas velocidades en las operaciones de carga y consulta. La carga de tripletas, a través de sus parseadores de RDF¹¹ a XML y N-Triples facilita el trabajo sobre todo con grandes ficheros. Usando un hardware estándar, puede cargar gigabytes de datos RDF en minutos.

1.9.2. Características

- SPARQL 1.1

Según (W3C, Sparql 1.1) es un conjunto de especificaciones que permite consultar y manipular el lenguaje RDF en la Web o en un triple store RDF. En relación a SPARQL 1.0 se añaden las funciones de concat, sum, min, max, avg, insert (añade tripletas), delete (elimina tripletas), load (carga de tripletas), clear (elimina todas las tripletas), create (crear un grafo), drop (eliminar un grafo específico).

- SPARQL v1.1 para razonamiento geoespacial temporal y análisis de redes sociales

Se hace uso de las propiedades mágicas para hacer uso de SPARQL v1.1 en razonamiento geoespacial temporal y análisis de redes sociales. Una propiedad mágica es un predicado de una consulta SPARQL que produce enlaces con coincidencias en el subgrafo, este tipo de extensiones proporcionan un entorno de consulta mucho más rico.

- Almacenamiento y velocidad de carga

Está diseñado para la velocidad d carga máxima y la velocidad de las consultas, la línea de productos siempre ha empujado de Allegrograph siempre se ha enfocado sobre el rendimiento a partir de la versión 1.0 en el 2004 que fue el primer producto en cargar e indexar 1 billón de tripletas utilizando hardware de x86 y 64bits. La capacidad de gestionar de forma automática todos los recursos de hardware disponibles para maximizar las capacidades de carga, indexación y búsqueda de nuevo eleva el nivel de rendimiento de almacenamiento RDF.

¹¹ RDF.- Resource Description Framework es un lenguaje de objetivo general para representar información de la web.

- Razonamiento temporal y geoespacial

Almacena tipos de datos geoespaciales y temporales como estructura de datos nativas, combinado con los mecanismos de indexación y de consulta.

- Tipos de datos nativos y consultas eficientes

Almacena un amplio rango de tipos de datos directamente en su triple representación, esto favorece la existencia de consultas eficientes y una significativa reducción del tamaño del almacén de los datos.

- Reitificación¹² directa

Permite que los identificadores de tripletas sean el sujeto u objeto de otra tripleta, la ventaja de esta aproximación es que se puede reducir el número total de tripletas en el almacén hasta un tamaño más manejable y se reduce el tiempo de consulta porque una sola consulta puede devolver más datos.

- Clustering

Cuando se almacena un gran conjunto de datos en un único sistema de procesamiento, aproximadamente el 60% del tiempo se emplea en la carga de tripletas y el 40% indexándolas. Si se cargan los datos de forma masiva en un sistema de procesamiento múltiple o en un clúster de máquinas independientes, se puede hacer casi toda la indexación al proceso de carga.

- Bases de datos distribuidas para tripletas

Soporta consultas en bases de datos distribuidas, además, permite abrir hilos de ejecución seguros de varias bases de datos desde una sola aplicación (solo lectura de las bases de datos).

- Consultas Prolog

AllegroGraph RDF Prolog ofrece poderoso razonamiento estándar de la industria, el dominio específico de la construcción de conceptos de alto nivel (que requieren reglas

¹² Reitificación.- operación que se realiza sobre grafos RDF, también se puede definir como “sentencias sobre sentencias”.

complejas o procesamiento numérico). AllegroGraph Prolog es una opción debido a que muchos casos de uso son difíciles (o muy complicado) para modelo con solo RDF / RDFS y OWL.

- Análisis de redes sociales

AllegroGraph incluye una biblioteca SNA que trata a un triple-store¹³ como un grafo de relaciones, con las funciones para la medición como importancia y centralidad, todo esto para formar un marco de reconocimiento de actividades para analizar las redes de manera flexible y eventos en grandes volúmenes de datos estructurados y no estructurados (Franz, AllegroGraph 4.10 Social Networking Analysis, 2013)

- Nodal neighbors
- Ego group
- Density
- Actor degree centrality
- Actor closeness centrality
- Actor betweenness centrality
- Group degree centrality
- Group closeness centrality
- Group betweenness centrality

1.9.3. Arquitectura

Proporciona una arquitectura del protocolo REST, está adaptado a varios idiomas como: Sesame Java, Sesame Jena, Python utilizando las firmas de Sesame, y Lisp. Hay adaptadores de código abierto a través de proyectos comunitarios para C #, Ruby, Clojure, Scala, y Perl.

¹³ Triple-store.- construido en una base de datos para almacenamiento y recuperación de triple. Un triple es una entidad de datos compuesta por sujeto (recurso), predicado (propiedad o relación) y objeto (valor).

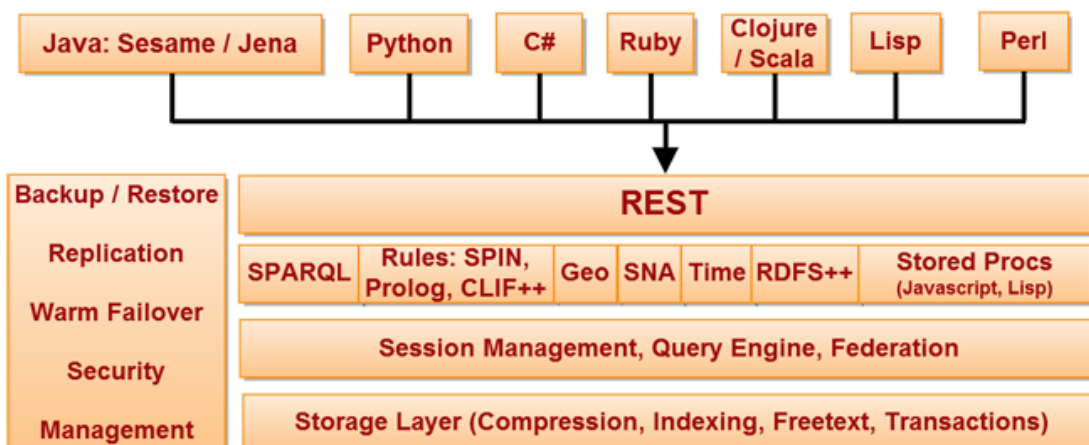


Ilustración 23.- Arquitectura Allegrograph
Fuente: (Franz, AllegroGraph for Social Network Analysis)

La distribución del cliente Java en Allegrograph contiene el API de Jena, la misma que ofrece un acceso conveniente y eficiente al servidor de Allegrograph basada en una aplicación Java a través del puerto 10035, este API provee métodos para crear, consultar datos RDF y para gestionar las tripletas almacenadas.

El API de Jena se encuentra envuelto en una simple arquitectura de Allegrograph, un archivo de base de datos, cliente Java y los archivos .java como se muestra en la Ilustración 24.

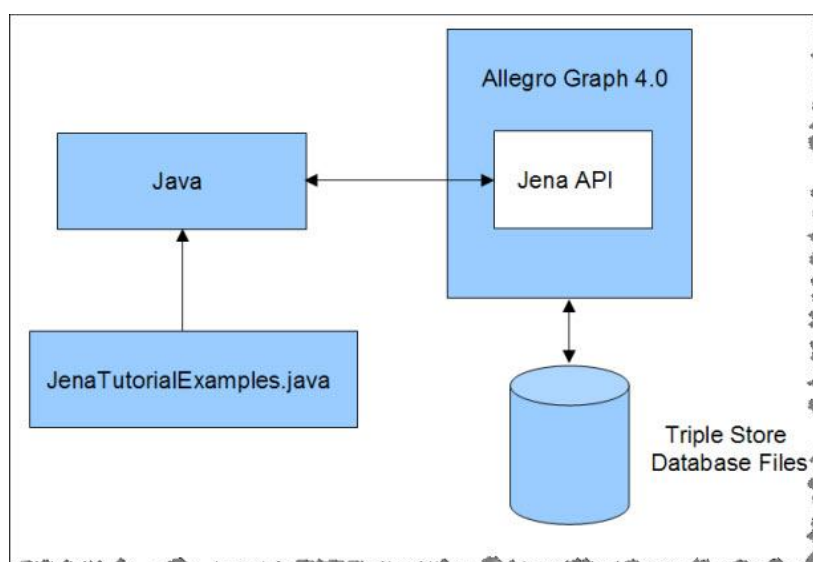


Ilustración 24.- Arquitectura Allegrograph, Jena, Java DataBase
Fuente: (Franz, Jena Tutorial)

El proceso general para generar el objeto de conexión que nos permite manipular las

tripleas de un repositorio en específico sigue el esquema de la Ilustración 25.

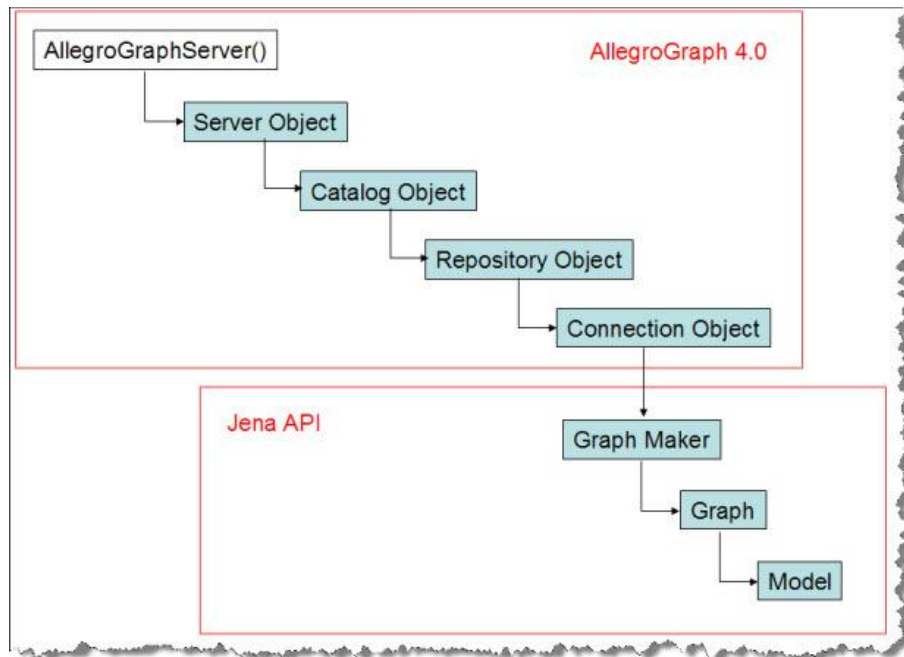


Ilustración 25.- Esquema de conexión a un repositorio
Fuente: (Franz, Jena Tutorial)

1.9.4. Tecnologías semánticas compatibles

- AGWebView

El AllegroGraph Web View es una interfaz gráfica de usuario para explorar, consultar y gestionar triple stores de AllegroGraph, además, utiliza la interfaz HTTP de AllegroGraph para proporcionar estos servicios a través de un navegador web.

Usando AGWebView puede:

- Explorar catálogos y repositorios disponibles
- Crear y eliminar repositorios
- Carga de datos de RDF en un repositorio
- Consultas SPARQL y Prolog, que se pueden guardar y volver a utilizar
- Ver y agregar espacios de nombres

- Gestión de usuarios AllegroGraph
- Terreno geoespaciales resultados de la consulta en una pantalla de Google Maps.

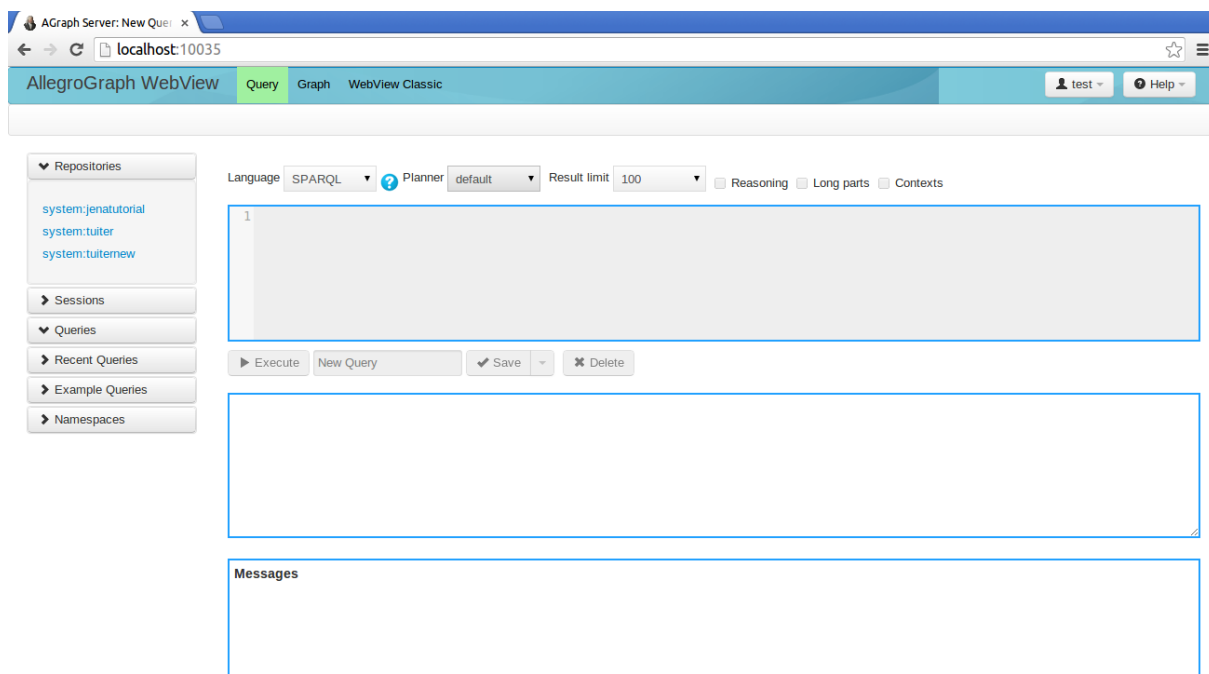


Ilustración 26.- Allegrograph Web View

- Gruff

Gruff es un navegador interactivo de triple stores, gestor de consultas y editor que se basa en AllegroGraph. La información puede ser consultada en forma de gráficos visuales de nodos y líneas de enlace que se nos extienden de forma automática, y también como tablas de propiedades de los nodos particulares. Las consultas se pueden escribir textualmente como SPARQL o código Prolog, o diseñados gráficamente como diagramas de nodos y líneas de enlace. Los datos pueden ser creados y editados mediante la cumplimentación de las tablas de valores de propiedad. Los diferentes puntos de vista y las herramientas están estrechamente integrados a la instalación rápida navegación, consulta y edición.

Mediante la selección de recursos y predicados particulares, se puede construir un gráfico visual que muestra una variedad de las relaciones, mostrar las tablas de todas las propiedades de los recursos seleccionados o generar tablas con consultas SPARQL, y los recursos de las tablas se puede agregar a la gráfica visual.

Es una herramienta complementaria a las estadísticas tradicionales, como el pensamiento visual con interfaces interactivas ahora se reconoce para facilitar el razonamiento. Este es un software para análisis exploratorio de datos (Gephi).

- Características
 - Visualización en tiempo real

Motor gráfico de visualización más rápida de la comprensión de aceleración y el descubrimiento de patrones en grandes gráficos. Impulsado por su ad-hoc motor OpenGL, Gephi está empujando el sobre en la forma en la exploración de la red interactiva y eficiente. Itera a través de la visualización mediante el filtrado dinámico. Soporta redes de hasta 50.000 nodos y 1000000 aristas.

- Análisis de redes sociales

Creación sencilla de conectores de datos sociales para mapear las organizaciones comunitarias y las redes de mundo pequeño. Las estadísticas y el marco de indicadores ofrecen las métricas más comunes para el análisis de redes sociales como: intermediación, cercanía, diámetro, coeficiente de agrupamiento, ruta más corta, detección de comunidades, densidad.

- Poster Creation

Promueve el trabajo científico con los mapas imprimibles de alta calidad.

- Interfaz ergonómica

Basado en NetBeans UI.

- Alto rendimiento

Motor de renderizado 3D.

- Requisitos del sistema

Gephi es de código abierto que se ejecuta sobre Windows, Linux y Mac OS X, requiere Java versión 6 y posteriores.

Los requisitos mínimos del hardware CPU 500 MHz + 128 MB RAM + OpenGL 1.2, el rendimiento es en función del tamaño del gráfico.

Si está utilizando Gephi en Linux, se debe obtener los controladores gráficos más recientes para asegurar la aceleración de hardware OpenGL.

- Plugin Semantic Web Import (Demairy)

El plugin está destinado a permitir la importación de datos semánticos en Gephi. Los datos importados se obtienen mediante el procesamiento de una solicitud SPARQL en los datos semánticos. Los datos se pueden acceder después de tres maneras:

- Accediendo a archivos locales rdf, rdfs utilizando el motor Corese integrado para aplicar la solicitud SPARQL.
- Accediendo a un endpoint remoto SPARQL REST. En ese caso, la solicitud SPARQL se aplica de forma remota y el gráfico se construye localmente mediante el análisis del resultado enviado por el REST endpoint;
- Accediendo a un endpoint remoto SPARQL SOAP. En cuanto REST endpoint, la gráfica resultante se construye a partir del resultado devuelto por el endpoint.

También proporciona las siguientes funcionalidades: las estadísticas sobre los tipos de datos presentes en el RDF obtenido (si el las relaciones rdf:type son proporcionadas en la solicitud de construcción), un filtro del tipo de los nodos, filtrado de los nodos con base en una solicitud SPARQL "Select".

- Gephi Toolkit (Gephi)

El kit de herramientas es un JAR que se puede reutilizar en nuevas aplicaciones Java y lograr las tareas que se pueden hacer en Gephi automáticamente, a partir de un programa de línea de comandos, por ejemplo. La capacidad para utilizar las funciones de Gephi como este en otras aplicaciones Java aumentar las posibilidades y promete ser muy útil.

Tiene paquetes esenciales en una biblioteca estándar de Java, que cualquier proyecto Java puede utilizar como: gráfico, diseño, filtros, entrada/salida, análisis de redes sociales, cliques.

1.10.2. Proyecto R

R es un lenguaje y entorno para computación y gráficos estadísticos que proporciona una amplia variedad de estadística (modelos lineales y no lineales, pruebas estadísticas clásicas, análisis de series temporales, clasificación, agrupamiento...) y las técnicas gráficas, y es altamente extensible (Project R).

Mediante el uso de la librería 'igraph' se puede manejar grandes gráficos muy bien y proporciona funciones para la generación aleatoria y gráficos regulares, visualización gráfica, los índices de centralidad y mucho más. La librería SPARQL permite importar directamente los resultados de las consultas SELECT en el sistema estadístico de R desde un endpoint (Robert van Hage).

R está disponible para sistemas operativos Windows, Mac y Linux.

1.11. Aplicaciones que utilizan RDF y Allegrograph

- DBpedia Deutschland 1.0 Release

Es la parte Alemana de DBpedia que extrae información estructurada de Wikipedia y la pone a disposición en datos enlazados en la web. Se compone de más de 100 millones de hechos extraídos de Wikipedia, esto permite a las máquinas responder preguntas tales como: ¿Qué impresionistas nacen en Berlín? ¿Qué elementos químicos se encuentran en la tabla periódica? (Glahn, 2009)

- Proyecto de Eli Lilly – Entorno farmacéutico competitivo utilizando tecnologías semánticas

Desarrollo de un gran número de medicamentos y tratamiento de muchas enfermedades.

Utilizan la integración semántica junto con un modelo de datos RDF y ontologías que proporcionan un marco de gran alcance en la prestación de los diccionarios y las relaciones taxonómicas que ayudan a la inferencia de los datos para el descubrimiento de conocimiento (Ranga).

Aplican estadísticas y teoría de grafos como: clustering, page-rank algorithms.

- Pfizer

Es una de las primeras empresas en reconocer el poder de la Web Semántica a los esfuerzo de impacto para el descubrimiento de fármacos, actualmente trabajan en proyectos como verificación del compuesto de la pureza y análisis de droga de estabilidad (Zaino, 2010).

- TwitLogic

Introduce un agregador de datos semántico que agrupa a un conjunto de formatos compactos para contenido estructurado microblog con la Web Semántica utilizando vocabularios y las mejores prácticas con el fin de aumentar la Web Semántica en tiempo real, que se rijan por el usuario de los datos.

Hace uso de métricas simples que se definen como el número de instancias disponibles. Si los datos tomados en cuenta es un subconjunto representativo de datos del mundo, entonces los valores más altos deben indicar que la mayoría de las interacciones alrededor de un tema en particular se están produciendo en un tipo particular de medio y por lo tanto el uso de tales fuentes podría resultar en una mayor precisión. Para calcular la relevancia de un tema determinado, el número de enlaces no ponderados a ese elemento se calcula y para mejorar los resultados los enlaces ponderados podrían ser considerados (De Vocht, 2011).

1.12. Trabajos relacionados que aplican RDF y SNA

Para la Web Semántica se han puesto en práctica aplicaciones que propician el análisis de redes sociales que toman como premisa un vocabulario RDF para describir información sobre los individuos y sus relaciones en una red social, conocido como FOAF (Friend of a Friend), que se extiende con el motivo de expresar el nivel de confianza entre los individuos de la red para finalmente mostrar su aplicación práctica (Rodríguez & Ronda).

Algunas aplicaciones que aplican RDF y SNA son los Sistemas de Recomendación Semánticos, que se menciona a continuación.

- FilmTrust

Un sitio web que integra un sistema de recomendación semántico colaborativo de películas que utiliza el vocabulario FOAF como base para la creación de una red social de confianza. En esta red cada individuo debe valorar la confianza que les merecen aquellos usuarios a los que añaden a su red de amigos (a partir de los cuales se generan recomendaciones de películas). El sistema adicionalmente dispone de un mecanismo que permite determinar la precisión de las recomendaciones generadas (Golbeck).

- Modelo de Kobayachi y Saito

Propone un sistema para recomendar artículos periodísticos y noticias extraídos de canales RSS (RDF Site Summary or Rich Site Summary)¹⁴, utilizando como apoyo un tesauro¹⁵ cuyos términos ayudan a representar tanto las materias de las noticias como las necesidades de información de los usuarios (Kobayashi, 2007).

- ReConRank: A Scalable Ranking Method for Semantic Web Data with Context

Se presenta un enfoque que adapta el conocido algoritmo PageRank/HITS a los datos de la Web Semántica. Nuestro método combina rangos de la gráfica RDF con rangos de la gráfica contexto, es decir, los datos fuentes y su vinculación. Se presentan los resultados de evaluación del desempeño sobre la base de un gran conjunto de datos RDF obtenido de la web (Hogan, Harth, & Decker).

- Sistemas de Inteligencia Web: Análisis de Redes Sociales

Desarrollo de un esquema algorítmico que combina la extracción de redes sociales con la búsqueda de información. De esta forma se forma un marco de trabajo que permite especificar fácilmente las necesidades informativas de los usuarios, extrayendo información que no son cubiertas por las bases de datos convencionales, mediante la implementación de sistemas de búsquedas dirigidos por heurísticas (De la Rosa Troyano).

¹⁴ RSS.- es un vocabulario RDF basado en XML que permite la catalogación de información de tal manera que sea posible encontrar información precisa adaptada a las preferencias de los usuarios.

¹⁵ Tesauro.- vocabulario controlado con el que podemos traducir el contenido de los documentos expresados en lenguaje natural a un lenguaje documental, evitando la ambigüedad del primero.

CAPÍTULO 2
VOCABULARIO DE TWITTER

Para desarrollar la ontología para Twitter se ha basado en la metodología NeOn¹⁶ que cuenta con escenarios que se apoyan en los aspectos de colaboración de desarrollo de ontologías y reutilización. Debido a que la construcción de ontologías a partir de cero está consumiendo recursos, la reutilización de ontologías reduce tiempo y costes en la proceso de desarrollo, se propaga a las buenas prácticas y cuando las ontologías están bien desarrolladas se vuelven a utilizar aumentando la calidad general de los modelos ontológicos según (Suárez-Figueroa & Gómez-Pérez).

En cuanto al vocabulario para Twitter se reutiliza algunos vocabularios como: DC, SIOC, MOAT, GEO, GN, FOAF que se ajustan perfectamente a lo que queremos modelar, ya que permiten la descripción de los datos mediante el uso de sus propiedades para garantizar un modelo de representación que permitan ser consumidos por aplicaciones web facilitando la interoperabilidad.

Debido a que se necesitan algunas propiedades que no se encuentran en los vocabularios internacionalmente reconocidos para la representación de la información se crea un vocabulario (TW) para poder modelarlas.

Se menciona también los lenguajes RDF Schema y OWL que permiten describir y dar significado a los vocabularios y las relaciones entre esos términos.

2.1. Vocabularios

Un vocabulario es un diccionario de términos semánticos con los cuales se pretende representar o modelar un dominio, con la finalidad de facilitar la comunicación y el intercambio de información entre diferentes sistemas y entidades. Los elementos de este diccionario son generalmente categorías (o clases), atributos (o propiedades o características) y las relaciones entre ellos según (Gruber).

Los vocabularios van a ser expresados usando RDF que es un lenguaje de objetivo general para representar la información de la web. Se basan en convertir las declaraciones de los recursos en expresiones con la forma sujeto-predicado-objeto (conocidas como tripletas).

Es importante mencionar que el sujeto y predicado se indican mediante un URI (Universal Resource Identifier), mientras que el objeto puede ser URI o literal.

¹⁶ NeOn.- Metodología para la construcción de redes ontológicas.

2.1.1. DC (Dublin Core)

Define un conjunto básico de atributos que sirven para describir una amplia gama de recursos de la red que conlleva: recursos (objeto que se va a describir), elementos (características o propiedades) y valores (cualidad definida por el elemento) que refinan la semántica de tal forma que pueden ser útiles para la localización y recuperación del recurso.

El recurso puede ser un objeto que tiene una identidad definida como cualquiera de los elementos del DC y esta a su vez, un valor que la califica como una cadena de caracteres. Por ejemplo, el elemento título está formado por el identificador título del recurso (en este caso, dc.title) y por el valor constituido por toda la cadena de caracteres que forman el título del recurso (Senso & De la Rosa Piñero).

Las propiedades que se reutilizan de este vocabulario son para representar la fecha de creación y lenguaje del tuit:

- created
- language

2.1.2. SIOC (Semantically-Interlinked Online Communities)

SIOC, proporciona los principales conceptos y propiedades necesarias para describir la información de las comunidades en línea (foros, wikis, blogs, etc) en la Web Semántica.

Se ha logrado recientemente la adopción significativa a través de su uso en una variedad de aplicaciones de software comercial y de código abierto, y se utiliza comúnmente en combinación con el vocabulario FOAF para expresar perfil personal y la información de las redes sociales (Berrueta & Brickley).

Las propiedades de objeto y de datos que se reutilizan se muestran a continuación:

- Propiedades de objeto

- addressedTo
- has_creator
- links_to

- Propiedades de datos

- content
- link

2.1.3. MOAT (Meaning of a tag)

MOAT proporciona un entorno de colaboración para que la Web 2.0 productora de contenidos de significado a sus etiquetas utilizadas para categorizar el contenido asociado a un URI de un recurso, además, modela las relaciones entre las etiquetas y su significado de una forma legible para las máquinas. Utiliza SIOC para describir el ítem etiquetado (Passant & Laublet, 2008).

Para representar las etiquetas o hashtags se reutiliza del vocabulario moat la propiedad "name".

2.1.4. GEO

Proporciona a la comunidad Web Semántica un espacio de nombres para la representación de latitud, longitud y otra información acerca de las cosas especialmente localizados, usando WGS84¹⁷ como punto de referencia (Brickley, Basic Geo (WGS84 lat/long) Vocabulary).

De una manera más simple, es un vocabulario para definir ubicaciones geográficas de los recursos RDF.

De este vocabulario se reutilizan las propiedades latitud (lat) y longitud (long).

2.1.5. GN (Geo Names)

Permite añadir información semántica geoespacial a la World Wide Web como: nombre del país, código de país, etc

Debido a que ahora los topónimos Geonames tienen una URL única con un servicio RDF correspondiente que se encuentran almacenados en una base de datos geográfica que contiene todos los países y más de ocho millones de nombres de lugares, esto nos facilita hacer un enlazado de datos y obtener más información acerca del lugar donde se realizó el tweet.

¹⁷ WGS84 (World Geodetic System 84).- que permite localizar cualquier punto de la tierra.

Para representar el código del país y el nombre del lugar donde se porteó el tuit se reutilizan las siguientes propiedades:

- country_code
- name

2.1.6. FOAF (Friend of a friend)

Integra tres tipos de redes: las redes de colaboración humana, amistad y asociación que permiten conectar a personas y a la información mediante la web, además, describe relaciones utilizando RDF para que puedan ser procesadas fácilmente por las máquinas.

De este vocabulario se reutiliza lo siguiente:

- name
- nick

2.1.7. TW (Twitter)

TW es un vocabulario propio que se ha creado con el fin de poder representar: el contador de retweets, contador de favoritos, código html, país, dirección de las calles donde se emitió el tweet, lugar, url expandido, url acortado.

Las siguientes clases, propiedades de objetos y de datos han sido creadas mediante este vocabulario:

- Clases

- place
- tweet
- url
- user

- Propiedades de objetos

- posted_from

- Propiedades de datos

- country
- place_type
- street_address
- display_url
- extended_url
- source
- retweet_count
- favorite_count

2.1.8. RDF Schema

RDF Schema es una extensión semántica de RDF que consiste en un lenguaje primitivo de ontologías que proporciona los elementos básicos para la descripción de vocabularios (Brickley, RDF Vocabulary Description Language 1.0: RDF Schema, 2004).

A continuación se mencionan las propiedades más importantes para la definición de esquemas aplicados a los vocabularios de recursos que utilizan RDF:

- *rdf:range*.- se utiliza para indicar que los valores de una propiedad son instancias de una o más clases.
- *rdf:type*.- se utiliza para indicar que un recurso es una instancia de una clase.
- *rdf:domain*.- se utiliza para indicar que cualquier recurso que tiene una determinada propiedad es una instancia de una o más clases.

2.1.9. OWL (Web Ontology Language)

OWL se puede utilizar para representar explícitamente el significado de los términos en vocabularios y las relaciones entre esos términos, además, tiene más facilidades para expresar significado y semántica que RDF y RDF-S, y por lo tanto OWL va más allá de estos idiomas en su capacidad de representar a máquina el contenido interpretable en la

Web (McGuinness & Van Harmelen).

- *owl:Class*.- define un grupo de individuos que van de la mano, ya que comparten algunas propiedades.
- *owl:DataTypeProperty*.- se utiliza para relacionar un clase con un valor (literal).
- *owl:ObjectProperty*.- se utiliza para relacionar una clase con otra clase.

2.2. Conceptualización

Tabla 2.- Descripción de clases

DESCRIPCIÓN DE CLASES (<i>owl:Class</i>)	
tw:place	
URI	http://data.twitter.com/ontology#Place
Label	Place
Descripción	Clase que representa al lugar donde se emitió el tuit
tw:tweet	
URI	http://data.twitter.com/ontology#Tweet
Label	Tweet
Descripción	Clase que representa el tuit
tw:url	
URI	http://data.twitter.com/ontology#Url
Label	Url
Descripción	Clase que representa los urls que posee el tuit
tw:user	

URI	http://data.twitter.com/ontology#User
Label	User
Descripción	Clase que representa el usuario que posteó el tuit

Tabla 3.- Descripción de propiedades de objetos

DESCRIPCIÓN DE PROPIEDADES DE OBJETOS (<i>owl:ObjectProperty</i>)	
sioc:addressedTo	
URI	http://rdfs.org/sioc/ns#addressedTo
Label	Place
Descripción	Relación que especifica que un usuario es mencionado en el tuit
sioc:has_creator	
URI	http://rdfs.org/sioc/ns#hasCreator
Label	has_creator
Descripción	Relación que especifica que un usuario es el creador del tuit
sioc:links_to	
URI	http://rdfs.org/sioc/ns#links_to
Label	links_to
Descripción	Relación que especifica que un URL fue extraído del contenido del tuit

tw:posted_from	
URI	http://data.twitter.com/ontology#posted_from
Label	posted_from
Descripción	Relación que especifica el lugar donde fue posteado el tuit

Tabla 4.- Descripción de propiedades de datos

DESCRIPCIÓN DE PROPIEDADES DE DATOS <i>(owl:DataTypeProperty)</i>	
dc:created	
URI	http://purl.org/dc/terms/#created
Label	created
Descripción	Fecha de creación del tuit
dc:language	
URI	http://purl.org/locwd/schema#language
Label	language
Descripción	Lenguaje que se escribió el tuit
foaf:name	
URI	http://xmlns.com/foaf/0.1/name
Label	name
Descripción	Nombre del usuario que posteó el tuit
foaf:Nick	

URI	http://xmlns.com/foaf/0.1/nick
Label	Nick
Descripción	Alias del usuario que posteo el tuit
geo:lat	
URI	http://www.w3.org/2003/01/geo/wgs84_pos#lat
Label	lat
Descripción	Latitud (ubicación) de donde se posteo el tuit
geo:long	
URI	http://www.w3.org/2003/01/geo/wgs84_pos#long
Label	long
Descripción	Longitud (ubicación) de donde se posteo el tuit
gn:country_code	
URI	http://www.geonames.org/ontology#country_code
Label	Country_code
Descripción	Código del país donde se posteo el tuit
gn:name	
URI	http://www.geonames.org/ontology#name
Label	name
Descripción	Nombre del lugar donde se posteo el tuit
moat:name	
URI	http://moat-project.org/ns#name_hash

Label	name
Descripción	Nombre del hashtag que se encuentra en el contenido del tuit
sioc:content	
URI	http://rdfs.org/sioc/ns#content
Label	content
Descripción	Contenido del tuit
sioc:link	
URI	http://rdfs.org/sioc/ns#link
Label	link
Descripción	URL de un recurso que es extraído del tuit
tw:country	
URI	http://twitter.com/vocab/tw#country
Label	country
Descripción	Nombre del país donde se emitió el tuit
tw:place_type	
URI	http://twitter.com/vocab/tw#place_type
Label	place_type
Descripción	Tipo de lugar (ej. ciudad) donde se emitió el tuit
tw:street_address	
URI	http://twitter.com/vocab/tw#street_address
Label	street_address

Descripción	Dirección donde se emitió el tweet
tw:display_url	
URI	http://twitter.com/vocab/tw#display_url
Label	display_url
Descripción	URL que se muestra en el tuit
tw:extended_url	
URI	http://twitter.com/vocab/tw#extended_url
Label	extended_url
Descripción	URL extendido que se muestra en el tuit
tw:source	
URI	http://twitter.com/vocab/tw#source
Label	source
Descripción	Código HTML del tuit
tw:retweet_count	
URI	http://twitter.com/vocab/tw#retweet_count
Label	retweet_count
Descripción	Número de veces que ha hecho retweet al tuit
tw:favorite_count	
URI	http://twitter.com/vocab/tw#favorite_count
Label	favorite_count
Descripción	Número de veces que se ha marcado como favorito al tuit

2.3. Namespaces

Debido a que las tripletas en RDF se hacen muy largas porque el sujeto, predicado y objeto (no siempre) se representan mediante URIs, se utilizan prefijos para evitar la ambigüedad a la hora de utilizar los namespaces. A continuación se mencionan los espacios de nombres con los que vamos a trabajar:

Tabla 5.- Prefijos y Namespace

Prefijos	Namespace
DC	http://purl.org/dc/elements/1.1/
FOAF	http://xmlns.com/foaf/0.1/
GEO	http://www.w3.org/2003/01/geo/wgs84_pos#
GN	http://www.geonames.org/ontology#
MOAT	http://moat-project.org/ns#
OWL	http://www.w3.org/2002/07/owl#
SIOC	http://rdfs.org/sioc/ns#
TW	http://data.twitter.com/ontology#
RDFS	http://www.w3.org/2000/01/rdf-schema#
RDF	http://www.w3.org/1999/02/22-rdf-syntax-ns#
XSD	http://www.w3.org/2001/XMLSchema#

2.4. Ontología

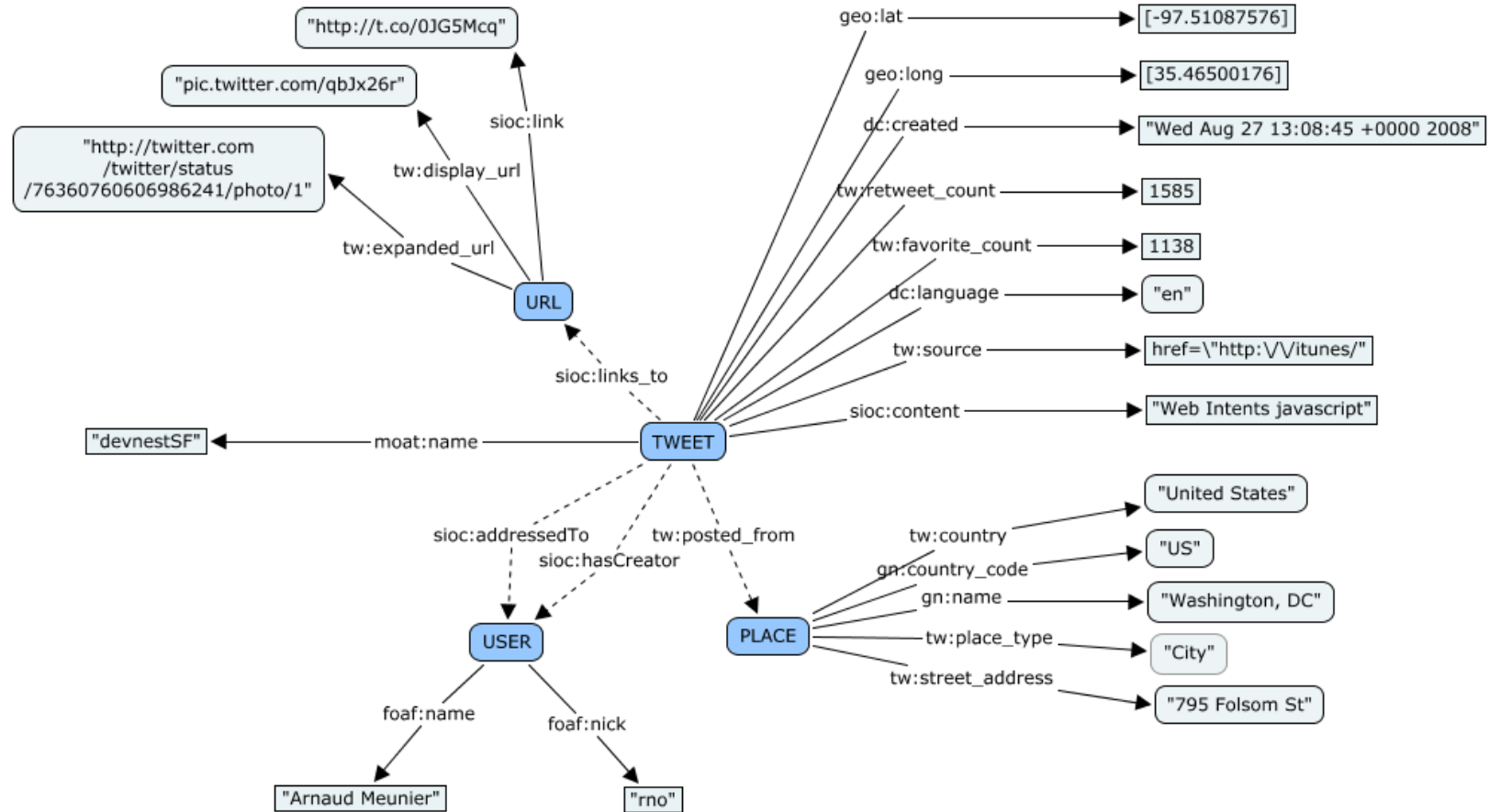


Ilustración 28.- Ontología

2.5. Creación de RDF

2.5.1. Ontología/Vocabulario

Para crear el RDF se hace uso de Protégé, un editor de ontologías de código abierto, a través de un cliente Web o un cliente de escritorio, donde se pueden desarrollar en variedad de formatos incluyendo OWL, y XML Schema. Protégé está basado en Java, es extensible y proporciona un entorno de plug-and-play que hace que sea una base flexible para la creación rápida de prototipos y creación de aplicaciones según (Stanford Center for Biomedical Informatic Research). Para más información acerca de la creación de la ontología en Protégé, consultar el Anexo A Crear ontología en Protégé.

2.5.2. Instanciación

Se utiliza Jena para realizar la instanciación de la información que se encuentran en la base de datos MYSQL con el RDF creado en Protégé. A continuación se detalla la manera de realizar todo este proceso:

Se crea un modelo vacío.

```
OntModel model = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM);
```

Usar el FileManager para encontrar el archivo de entrada.

```
InputStream is = FileManager.get().open(ruta);
```

Leer el archivo RDF/XML.

```
model.read(is, "");
```

Declaración de espacios de nombres que se requieren gestionar la ontología.

```
String namespace = "http://data.twitter.com/ontology#";
```


Se realiza una consulta a la base de datos para obtener los información que va a ser instanciada en este caso: el id del tweet, id del usuario, nombre del usuario, hashtag.

```
Statement st = conn.createStatement();
String query = "SELECT t.TWETT_ID, t.FROM_USER_ID, us.NAME nameus,
tt.NAME namehs
FROM tweett t, tagged tg, tag_t tt, user_t us
WHERE t.TWETT_ID = tg.TWETT_ID
AND tg.TAG_ID = tt.TAG_ID
AND t.FROM_USER_ID = us.USER_ID";

ResultSet rs = st.executeQuery(query);
```

Se obtiene el URI de la clase “tweet” del RDF. En la variable “strTweet_id” se almacena el id del tweet de la consulta realizada y se le concatena las letras “tw” (tw123) que van a servir para diferenciar del id del usuario “us” (us123) y de los enlaces “url” (url123). En “tweet_id” se crea un recurso (<http://data.twitter.com/ontology#tw123>) de tipo Tweet. En “contentIndividual” se crea un individual donde se va a almacenar el texto del tweet con su respectiva propiedad. El método “getURi” se lo describe más adelante y luego se procede a almacenar la tripleta creada en la base de datos para usarla posteriormente.

```
Resource tweetClass = model.getResource(getURi("tweet"));

String strTweet_id = "tw" + rs.getString("tweett_id");

Resource tweet_id = model.createResource(namespace + strTweet_id, tweetClass);

Individual contentIndividual = model.createIndividual(namespace + strTweet_id,
tweetClass);
contentIndividual.setPropertyValue(model.getDatatypeProperty(getURi("content")),
model.createLiteral(strContent));

query = "INSERT INTO tripletas (sujeto, predicado, objeto) VALUES ("
+ (namespace + strTweet_id) + "," + getURi("content") + "," + strContent + "));";
st.executeQuery(q);
```

Se utiliza la siguiente línea para presentar en pantalla el RDF en el formato que se desea, poder variar entre “N-TRIPLE”, “RDF-XML” o “TURTLE”.

```
model.write(System.out, "N-TRIPLE");
```

65

Para almacenar el RDF en un archivo se usa "FileOutputStream", el archivo se guardará en el directorio donde se encuentra el proyecto.

```
FileOutputStream out = new FileOutputStream("tuitInst.rdf");
model.write(out, "");
```

El método "createTripleTB()" para crear una tabla en la base de datos donde se almacenarán las tripletas.

```
public static void createTripleTB(){
    Statement stmt = conn.createStatement();
    String q = "CREATE TABLE IF NOT EXISTS tripletas ("
        + " sujeto varchar(200) NOT NULL,"
        + " predicado varchar(200) NOT NULL,"
        + " objeto varchar(300) NOT NULL"
        + ") ENGINE=InnoDB DEFAULT CHARSET=utf8;";
    stmt.executeUpdate(q);
    System.out.println("** Creating table...");
    stmt.executeUpdate("truncate table tripletas");
    System.out.println("** truncate table...");
}
```

El método "getURi" devuelve el URI de un DataTypeProperty o de un ObjectProperty que se encuentra almacenado en la base de datos en una tabla de equivalencias (ver más adelante).

```
public static String getURi(String propiedad) {
    Statement stmt = conn.createStatement();
    String query = "select uri from equivalences where propiedad='" + propiedad + "'";
    ResultSet rs = stmt.executeQuery(query);
    String uri = "";
    while (rs.next()) {
        uri = rs.getString(1);
    }
    return uri;
}
```

El código que se describe a continuación es de la clase Equivalencias, primero se crea la tabla "equivalences", luego mediante una consulta SPARQL se obtiene del archivo RDF el URI (<http://data.twitter.com/ontology#Tweet>), el tipo (class) y la propiedad (Tweet) y se los almacena en la tabla anteriormente creada.

```

Statement st = conn.createStatement();
st.executeUpdate("CREATE TABLE IF NOT EXISTS equivalences ("
    + " propiedad varchar(20) NOT NULL,"
    + " URI varchar(100) NOT NULL,"
    + " type varchar(20) NOT NULL"
    + ") ENGINE=InnoDB DEFAULT CHARSET=utf8;");

OntModel model = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM);
InputStream is = FileManager.get().open(ruta);
model.read(is, "");

String queryString =
    "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>"
    + "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>"
    + "PREFIX owl: <http://www.w3.org/2002/07/owl#>"
    + "select distinct * "
    + "WHERE {"
    + "{?uri rdf:type owl:DatatypeProperty .}"
    + "OPTIONAL {?uri rdf:type ?type} ."
    + "FILTER (?type = owl:DatatypeProperty) ."
    + "}"
    + "UNION"
    + "{?uri rdf:type owl:ObjectProperty .}"
    + "OPTIONAL {?uri rdf:type ?type} ."
    + "FILTER (?type = owl:ObjectProperty) ."
    + "}"
    + "UNION"
    + "{?uri rdf:type owl:Class .}"
    + "OPTIONAL {?uri rdf:type ?type} ."
    + "FILTER (?type = owl:Class) ."
    + "}}";

Query query = QueryFactory.create(queryString);
QueryExecution qe = QueryExecutionFactory.create(query, model);
ResultSet results = qe.execSelect();
for (; results.hasNext(); ) {
    QuerySolution sol = results.nextSolution();
    Object uri = sol.getResource("uri");
    Object type = sol.getResource("type").getLocalName();
    Object propiedad = sol.getResource("uri").getLocalName();

    st.execute("INSERT INTO equivalences (propiedad, URI, type) VALUES ("
        + propiedad + ", " + uri + ", " + type + ")");
}

```

CAPÍTULO 3
ANÁLISIS DE REDES SOCIALES EN LAS ONTOLOGÍAS

3.1. SNA en la Web Semántica

El concepto de la Web Semántica es una idea original de Tim Berners-Lee que es el creador de la World Wide Web.

Según (Berners-Lee, Lassila, & Lassila, 2001), la Web semántica no es una Web independiente, sino una extensión de la actual en la que se da un significado bien definido, permitiendo a los ordenadores y a la gente trabajar en cooperación. En un futuro próximo, se marcará el comienzo de una nueva funcionalidad significativa donde las máquinas son mucho más capaces de procesar y entender los datos que se limitan a entender en la actualidad.

La visión de la Web Semántica puede verse desde tres perspectivas diferentes: (a) un tipo de biblioteca universal que puede ser fácilmente accedido y usado por los seres humanos en su adquisición de información día a día, (b) la columna vertebral para el software o agentes computacionales, (c) método para federar bases de conocimiento particulares y bases de datos para realizar las tareas prevista por los seres humanos y sus agentes (Marshall & Shipman).

(Ereteo, Gandon, & Buffa) mencionan que los frameworks de la Web Semántica proporcionan un modelo gráfico (RDF), un lenguaje de consulta (SPARQL) y definición de esquemas (RDFS y OWL) para representar e intercambiar conocimiento en línea. Recientemente, las interacciones a través de la Web 2.0 han suscitado mucha atención en la comunidad de la Web Semántica. Las descripciones basadas en RDF de datos sociales proporcionan un grafo que ofrece una manera más poderosa y significativa para representar las redes sociales que los modelos tradicionales de SNA.

En la actualidad, la mayoría de los datos solo se puede acceder a través de las API o haciendo crawling¹⁸ a las páginas web y algunos datos sociales están disponibles en un formato semántico y pueden ser explotados fácilmente.

En la arquitectura de SNA en la Web Semántica, se utiliza grafos RDF para representar los datos sociales utilizando ontologías existentes junto con ontologías de dominio específico, definiciones SPARQL para calcular las funciones de SNA y el acceso a los datos a través de crawling o API como se muestra en la

Ilustración 29 (Ereteo, Gandon, & Buffa).

¹⁸ Crawling.- el término literalmente significa arrastrador en inglés y es uno de los varios nombres que tienen los “robots de búsqueda” que utilizan los motores de búsqueda para indexar páginas web en sus bases de datos.

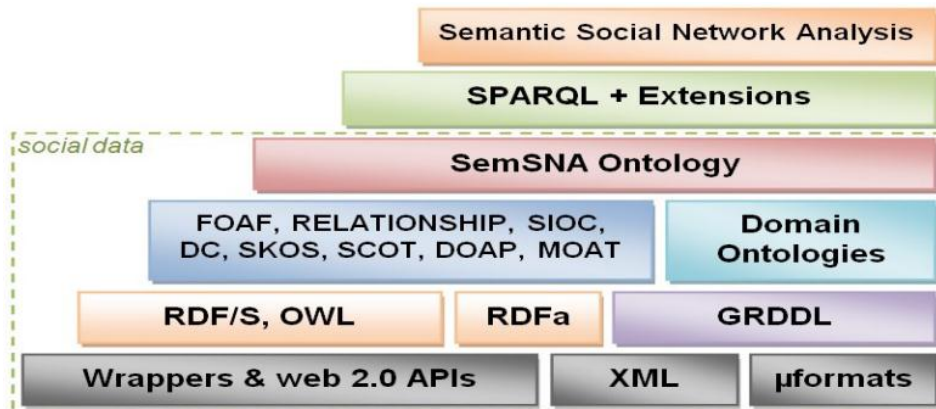


Ilustración 29.- Arquitectura de SNA en la Web Semántica
Fuente: Semantic Social Network Analysis (Ereteo, Gandon, & Buffa).

3.2. Proceso para realizar el análisis de redes sociales en Allegrograph

Para la creación de la ontología se utiliza Protégé que tiene como entrada los vocabularios, el mismo que genera un archivo RDF que luego en Jena se procede a realizar la instanciación de los datos para obtener las tripletas a partir de una base de datos relacional MySQL, las tripletas se las almacena en el triplestore Allegrograph y se procede a aplicar las métricas para realizar el análisis de redes sociales, como se muestra en la Ilustración 30.- .

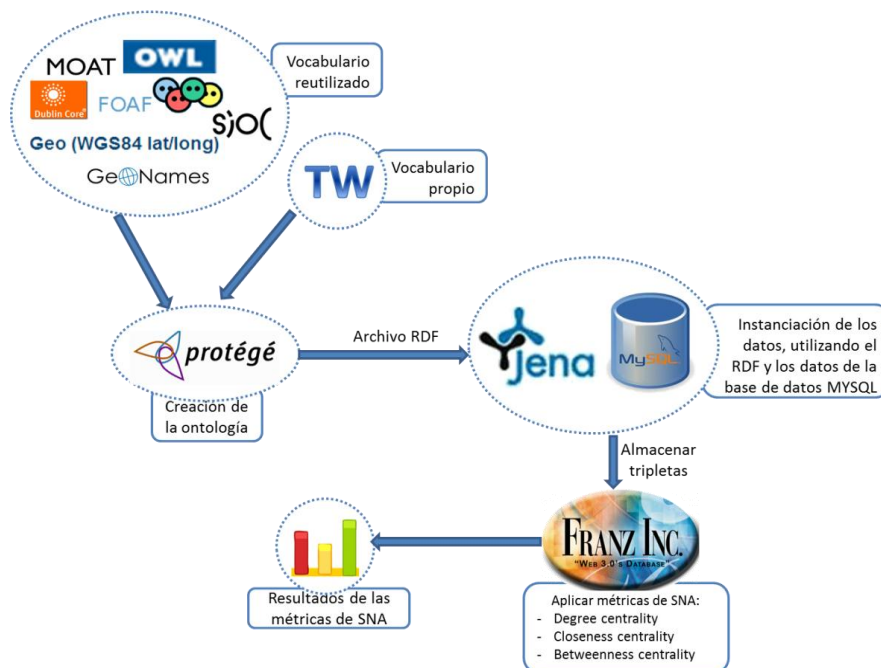


Ilustración 30.- Proceso para realizar el análisis de redes sociales en Allegrograph

3.3. SNA en Allegrograph

Allegrograph incluye algoritmos sofisticados para realizar el Análisis de Redes Sociales, que pueden examinar un grafo RDF de relaciones entre actores. Desde Java se puede hacer uso de la mayoría de las características de SNA en Allegrograph, ya sea a través de consultas, funciones como se describe a continuación:

- El API de Java Sesame AllegroGraph contiene funciones de configuración que te permiten crear un entorno SNA listos para consultas.
- A partir de Java en la versión 3 de Allegrograph mediante el uso de funciones o métodos (ej. `getNodeDegree()`).
- A partir de Java, se realiza consultas Prolog a AllegroGraph. Algunas de las funciones SNA tienen equivalentes Prolog denominados “functors” que se puede llamar directamente desde una consulta functors (ej. `nodal-degree`).
- A partir de Java, se realiza consultas SPARQL a AllegroGraph. Algunas de las funciones de SNA se accede mediante el uso de magic properties (ej. `sna:nodalNeighbors`).
- Dentro de una consulta Prolog, podemos abrir una ventana a Lisp y alcanzar funciones SNA Lisp del AllegroGraph.

Antes de describir con más detalle las formas para realizar el Análisis de Redes Sociales, es necesario familiarizarse con ciertos términos que se mencionarán a menudo:

- Magic Properties

Las magic properties son un predicado en una consulta SPARQL que trabajan con la librería de SNA. En el transcurso del presente proyecto a las magic properties se las denominará funciones de SNA:

- Generador

Un generador especifica un conjunto de vecinos que se puede llegar desde un nodo dado, en otras palabras, es uno o más predicados existentes en el archivo RDF, el resultado del generador es la unión de los nodos de todas las cláusulas.

El primer elemento de una cláusula es una cadena que identifica su naturaleza que puede ser: `objects-of`, `subjects-of`, o `undirected`.

- Objects-of (objeto-de).- esta cláusula recoge todos los nodos que son objetos de triples con los predicados especificados.
- Subjects-of (sujeto-de).- esta cláusula recoge todos los nodos que son sujetos de triples con los predicados especificados.
- Undirected (no dirigido).- esta cláusula recoge todos los nodos que son sujetos u objetos de triples con los predicados especificados.

El generador puede ser creado como un arreglo de objetos, lista de URIs, mediante consultas SPARQL o consultas PROLOG.

- Repositorio

Un repositorio es el lugar donde se almacena el archivo RDF en el servidor de Allegrograph, al mismo que se lo puede crear utilizando el API o AGWebView.

- Catálogo

Un catálogo es el lugar donde se almacenan varios repositorios en el servidor de Allegrograph, cabe recalcar que en la versión 4 de Allegrograph el catálogo por defecto se denomina “system” y se pueden crear más utilizando el API o AGWebView.

Para la presente investigación se ha especializado el campo de análisis de redes sociales, puesto que para estos análisis se debe seleccionar un ámbito o contexto específico para obtener resultados más precisos y claros; por este motivo, los tuits que van a ser analizados son todos aquellos que hayan sido etiquetados con el hashtag “accesibiity”.

3.4. SNA en Allegrograph versión 3

Conectarse al servidor de Allegrograph y crear un repositorio donde se va a almacenar el grafo, luego registrar el espacio de nombres con el que se va a trabajar en este caso “http://data.twitter.com/ontology#” y su prefijo “tw”. Cargar el grafo de datos en el repositorio.

```
AllegroGraphConnection ags = new AllegroGraphConnection();
AllegroGraph ts = ags.renew("snametrics", AGPaths.TRIPLE_STORES);
ts.registerNamespace("tw", "http://data.twitter.com/ontology#");
AGLoadNtriples.loadNTriplesWithTiming(ts, AGPaths.dataSources("tuitter.rdf"));
```


Obtener una instancia de la extensión SNA para poder hacer uso de sus ventajas y crear el generador de la siguiente manera: asignar un nombre “generator”, crear un arreglo de objetos donde el primer elemento es “undirected” porque se necesita recoger todos los nodos que sean sujetos y objetos, los demás elementos son los predicados, a cada predicado se le antepone el signo de admiración “!” para poder aplicar las funciones de SNA.

```
SNAExtension sna = ts.getSNAExtension();
sna.registerGenerator("generator",
    new Object[]{ "undirected", moat:name", "!foaf:name", "!sioc:addressedTo" });
```

Mediante un ciclo repetitivo se itera por todos los nodos aplicando los métodos como: `getNodeDegree()`, `getNodeNeighbors()`, `getBetweennessCentrality()`, `getClosenessCentrality()`, `getNodeDegreeCentrality()`, se debe tomar en cuenta que a las tres últimas funciones se les añade un parámetro “group”, el mismo que contiene un arreglo de objetos en donde están todos los nodos a analizar. El tipo de dato de “`getNodeNeighbors`” es “UPI[]” un arreglo de UPIs¹⁹ que almacenará todos los nodos vecinos.

```
long nodalDegree = sna.getNodeDegree("nodo", "generator");
UPI[] nodalNeighbors = sna.getNodeNeighbors ("nodo", "generator");
double cc = sna.getClosenessCentrality("nodo", group, "generator");
double bc = sna.getBetweennessCentrality ("nodo", group, "generator");
double dc = sna.getNodeDegreeCentrality ("nodo", group, "generator");
```

¹⁹ UPI (Unique Part Identifier).- Identificador único de un nodo

3.5. SNA en Allegrograph versión 4 y consultas PROLOG

Conectarse al servidor de Allegrograph <http://localhost:10035>, el username “test” y el password “xyzyzy” son valores que vienen preestablecidos para poder acceder al servidor, al menos que ingrese al AGWebView y cree otros usuarios. Se procede a crear un catálogo si se desea o caso contrario se accede al que viene por defecto “system”. Se crea un repositorio “miRepository”, pero antes se elimina los repositorios que tengan ese nombre en el catálogo para evitar que se dupliquen los datos, se inicializa el repositorio y se realiza la conexión añadiendo el archivo RDF que se tiene localmente. El “AGValueFactory” nos permitirá crear URIs que se utilizarán en los generadores. Si se necesita saber cuántas tripletas se han cargado en el servidor, utilizar la función “size”.

```
AGServer agserver = new AGServer(SERVER_URL, USERNAME, PASSWORD);
AGCatalog catalog = agserver.getCatalog(CATALOG_ID);
    catalog.deleteRepository(REPOSITORY_ID);
AGRepository myRepository = catalog.createRepository(REPOSITORY_ID);
    myRepository.initialize();
AGRepositoryConnection connRC = myRepository.getConnection();
    connRC.add(new File("tuitarInst.rdf"), null, RDFFormat.RDFXML);
AGValueFactory vf = connRC.getValueFactory();
    println("Loaded " + connRC.size() + " tuitarInst.rdf triples.");
```

Crear URIs para relacionar los predicados, primero añadiendo un prefijo “tw” a “<http://data.twitter.com/ontology#>”, luego vincular el namespace al nombre del predicado. Para crear un generador se declara un ArrayList de tipos URI en este caso “generator” y mediante la función “Collections.addAll” se agregan al generador los predicados, al momento de registrar el generador se tiene que tomar en cuenta lo siguiente: el primer parámetro es el nombre del generador, el segundo y tercer parámetro se establecen si se desea trabajar con “objects-of” y “subjects-of” respectivamente, en este caso se lo deja en “null”, el cuarto parámetro cuando se necesita trabajar con “undirected” por tal razón se establece “generator” que contiene el ArrayList con todos los predicados creados anteriormente, el quinto parámetro es reservado para cuando se cree un generador a través de una cláusula PROLOG.

```

connRC.setNamespace("http://data.twitter.com/ontology#", tw);
connRC.setNamespace("http://moat-project.org/ns#", moat);
connRC.setNamespace("http://data.twitter.com/ontology#", sioc);
connRC.setNamespace("http://moat-project.org/ns#", foaf);

URI namehs = vf.createURI(moat, "namehs");
URI name = vf.createURI(foaf, "name");
URI addressedTo = vf.createURI(sioc, "addressedTo");

List<URI> generator = new ArrayList<URI>(1);
Collections.addAll(generator, namehs, name, addressedTo);
connRC.registerSNAGenerator("generator ", null, null, generator, null);

```

El bloque de código PROLOG que se utiliza en la mayoría de consultas establece lo siguiente: los puntos de partida o nodos que se va a aplicar las métricas de SNA “!tw:123” (se antepone el signo “!” para que sea identificado como nodo) y que se pueden encontrar siguiendo los predicados a cierta profundidad especificada, en este caso la profundidad es “3”; utilizar el generador “generator” que define los predicados; se declara algunas variables “?variables” en las que se almacenarán los objetos resultantes, donde se podrá iterar sobre los resultados e imprimir sus valores; “group-degree-centrality”, “group-closeness-centrality”, “group-betweenness-centrality” son funciones SNA que han sido adaptadas para el uso de consultas PROLOG.

```

queryString = "(select (?centrality)
    (ego-group !tw:123 3 generator ?group)
    (group-degree-centrality ?group generator ?centrality))";

queryString = "(select (?centrality)
    (ego-group !tw:123 3 generator ?group)
    (group-closeness-centrality ?group generator ?centrality))";

queryString = "(select (?centrality)
    (ego-group !tw:123 3 generator ?group)
    (group-betweenness-centrality ?group generator ?centrality))";

```

3.6. SNA en Allegrograph versión 4 y consultas SPARQL

Las funciones de SNA utilizan generadores para especificar qué nodos en el grafo son vecinos, para crear un generador a las funciones de SNA hay que identificarlas con un URI que utiliza como espacio de nombres el prefijo “sna” que es la abreviatura de “http://franz.com/ns/allegrograph/4.11/sna/”.

Cuando una función de SNA especifica un generador llamado “<generator>”, Allegrograph buscará una definición existente, caso contrario, en el triple store se almacenará lo siguiente:

```
graph sna:sna { _:b sna:hasName <generator> }
```

Si se encuentra, las tripletas asociadas con ese nodo en blanco, se pueden utilizar para utilizar el generador. Un generador se añade a la tienda utilizando una instrucción de actualización de SPARQL, donde se debe definir: los espacios de nombres con sus respectivos prefijos (ej. prefix tw: <http://data.twitter.com/ontology#>), con “sna:Generator” se especifica que es un generador, “sna:hasName” se utiliza para dar un nombre al generador, “sna:undirected” para combinar el “subjects-of” y “objects-of” en que definirá los vecinos como sujetos y objetos de un predicado determinado.

```
String query = "prefix sna: <http://franz.com/ns/allegrograph/4.11/sna/>
  prefix moat: <http://moat-project.org/ns#>
  prefix tw: <http://data.twitter.com/ontology#>
  prefix sioc: < http://rdfs.org/sioc/ns# >
  prefix foaf: < http://xmlns.com/foaf/0.1/>
  insert data {
    graph sna:sna {
      [ a sna:Generator ;
        sna:hasName sna:generator1 ;
        sna:undirected (moat:name foaf:name sioc:addressedTo) ]
    }
  }
connRC.prepareUpdate(QueryLanguage.SPARQL, query).execute();
```

Muchas de las funciones SNA se definen en términos de nodos y grupos, para especificar un grupo se requiere de un generador, un nodo de inicio y una profundidad, estos grupos actúan como nodos en blanco y no tienen ningún significado fuera de una consulta determinada, sin embargo, en una consulta se puede utilizar otras propiedades mágicas para examinar el grupo: para obtener el grado de centralidad de un actor por cada actor en un grupo, se crea un grupo con la función “sna:egoGroup” y luego utilizar “sna:actorDegreeCentrality”; utilizar “sna:members” para iterar sobre los miembros de un grupo.

Las medidas de centralidad soportadas por las propiedades mágicas son:

- sna: nodalNeighbors
- sna: actorDegreeCentrality
- sna: actorClosenessCentrality
- sna: actorBetweennessCentrality
- sna: groupDegreeCentrality
- sna: groupClosenessCentrality
- sna: groupBetweennessCentrality

```

query = "prefix tw: <http://data.twitter.com/ontology#>
        prefix sna: <http://franz.com/ns/allegrograph/4.11/sna/>\
        select (max (?centrality) as ?centrality) {
            ?group sna:egoGroup (sna:generator1 tw:123 3) .
            ?centrality sna:groupDegreeCentrality (sna:generator1 ?group) .
        }";

query = "prefix tw: <http://data.twitter.com/ontology#>
        prefix sna: <http://franz.com/ns/allegrograph/4.11/sna/>
        select ?centrality {
            ?group sna:egoGroup (sna:generator1 tw:123 3) .
            ?centrality sna:groupClosenessCentrality (sna:generator1 ?group) .
        }";

query = "prefix tw: <http://data.twitter.com/ontology#>
        prefix sna: <http://franz.com/ns/allegrograph/4.11/sna/>
        select ?centrality {
            ?group sna:egoGroup (sna:generator1 tw:123 3) .
            ?centrality sna:groupBetweennessCentrality (sna:generator1 ?group) .
        }";

```

3.7. Función para obtener los resultados

Al aplicar las métricas de SNA a cada uno de los nodos, al resultado y al nodo se los añade a un ArrayList diferente, para luego mediante una función o método obtener el mayor valor y su respectiva posición.

```
static ArrayList<Integer> mayor(ArrayList<Float> egoGroup) {
    ArrayList<Integer> posicion = new ArrayList<Integer>();
    int pos = 0;
    float mayor = egoGroup.get(0);
    for (int i = 0; i < egoGroup.size(); i++) {
        if (egoGroup.get(i) > mayor) {
            mayor = egoGroup.get(i);
            pos = i;
        }
    }
    mayor = egoGroup.get(pos);
    for (int i = 0; i < egoGroup.size(); i++) {
        if (egoGroup.get(i) == mayor) {
            posicion.add(i);
        }
    }
}
```

El resultado final se presentará de la siguiente manera:

```
Métrica: betweenness centrality
Posición del resultado mayor: [4]
Dato: 0.6467
Nodo: http://data.twitter.com/ontology#tw123
```

RESULTADOS

De la base de datos de Twitter, en específico se va a analizar los tuits relacionados con el hashtag “accessibility” porque es el hashtag más veces ha sido mencionado.

Se aplica las métricas de Análisis de Redes Sociales a 3217 tripletas, al ejecutar se tarda un tiempo de 270 minutos con 35 segundos (4 horas y media aproximadamente), es demasiado tiempo para ejecutar pocas tripletas, pero esto es debido a las características del computador (ver Tabla 7), cabe destacar que Allegrograph ejecuta millones de tripletas en pocos minutos porque hacen uso de clústeres con gran capacidad.

Los resultados de las métricas se pueden observar en la Tabla 6, considerando que “Closseness Centrality” y “Betweeness Centrality” se presentan: el valor más alto 1 y el más bajo 0.

Los nodos obtenidos en los resultados con prefijo “idUs” son los identificadores de usuario, prefijo “hs” son hashtags y prefijo “us” son nombres de usuarios.

Tabla 6.- Resultados de aplicar las métricas de SNA

MÉTRICA	DATO	RECURSO (hashtag, usuario)
Degree Centrality	234	idUs1513
Closseness Centrality	0.0907407402992248	hsToronto, hsParalympics, hsMicrosoft, hsOntario, hsplainlanguage, hscommunication, hsdevelopment, hsbooks, hsfirefox, hsSafari, hscaptivate, hsLearning, hsOffice2010, hsFeatures, hsChicago-Area, hsa fb, hsosx, hsWindows8, hsJFW, hsReview, hsBildschirm lupe, hsMLB, hslax, hsmobilea11y, hsbva, hsJellybean, hsrisilebilirlik, hs5raeasonsIHateFaceBook, hsqtatspi, hsonlineed

Betweenness Centrality	0.999939560890197	usdtlds, usdotpeople, usl_alford, , usAccessSanDiego, usRachel_Menzies, usaskmeaks, usTonyEddicott, usKen_Saunders, ustinitun, usthemaplenet, usbowiesinspace9, usvenalic, usSharepointAus, usHigherEdOnlineL, usmlaumoraes, usriescog, ussvknyc, usMetDigital, usBlackheathLife, usDaphneRepair, uspostpolionews, usHanigsberg, usAggelos_Taplatz, usHRESMcMaster, usShortieDesigns, usnakulshenoy, usgbatuyong
------------------------	-------------------	--

El nodo “idUs1513” que corresponde al nombre de usuario “dsai2012”, con 234 de grado de centralidad es el usuario que tiene más conexiones, tiende a tener posiciones favorables y por lo tanto más poder debido a las oportunidades y alternativas que tiene en relación a los otros actores. En la Ilustración 31 se puede apreciar los nodos con mayor grado de centralidad, donde se puede encontrar el nodo anteriormente mencionado que es el más popular y el más conocido.



Ilustración 31.- Resultados grado de centralidad



Ilustración 33.- Resultados grado de intermediación

Tabla 7.- Características del computador

CARACTERÍSTICA	DESCRIPCIÓN
SISTEMA OPERATIVO	2 particiones: Ubuntu 12.04 y Windows 7 Ultimate
PROCESADOR	Intel Core i7 CPU 2.20 GHz
MEMORIA (RAM)	6.00 GB
TIPO DE SISTEMA	Sistema operativo de 64 bits (Linux y Windows)
MODELO	HP Pavilion dv6 Notebook PC

CONCLUSIONES

Una vez culminado el proyecto, utilizando la base de datos de Twitter y realizando el análisis de sus datos mediante el uso de métricas de SNA en Allegrograph se puntualiza las siguientes conclusiones:

- Reutilizar vocabularios facilita la interoperabilidad y la flexibilidad con la que se pueden utilizar las propiedades y clases de diferentes vocabularios ontológicos para de esa manera compartir conocimiento a través de una estructura estandarizada.
- Cuando se instancia los datos en Jena, se debe tener especial cuidado al crear recursos y literales; si se crea un recurso se puede relacionar con otros conceptos, en cambio, si se crea un literal este representa un dato final.
- Aplicar métricas para realizar el análisis de los datos de Twitter nos ayuda a obtener los nodos o recursos con más influencia; los más conocidos (grado de centralidad), los que se comunican fácilmente con otros nodos (grado de cercanía), los que transmiten información entre grupo de nodos en la red (grado de intermediación).
- Mediante el grado de centralidad se puede llevar a cabo una valoración sobre la importancia de una entidad específica dentro de un contexto determinado, en este caso tenemos que el usuario “dsai2012” con identificador “idUs1513” es el más relevante dentro de la data analizada porque tiene el mayor número de conexiones relacionadas a la etiqueta “accessibility”.
- Cuando se utilizan las funciones SNA en el Gruff no es posible graficar el resultado, es un gran inconveniente ya que el uso de los gráficos ayuda al momento de realizar la interpretación de los mismos.
- La ventaja de utilizar Allegrograph es que las métricas de SNA se pueden aplicar a todo el grafo, pero también se puede aplicar a algunas propiedades (rdf), todo depende de lo que se desea analizar.

- Una forma de realizar el análisis de redes sociales en Allegrograph es mediante PROLOG, que es un lenguaje de propósito general que necesita de bastante conocimiento para utilizarlo debido a que no permite escribir reglas y expresar conceptos en triples.
- Allegrograph permite realizar el análisis de redes sociales a través de funciones SNA que pueden utilizar patrones con múltiples entradas y salidas, e incluye propiedades para hacer consultas SPARQL y de esa manera interactuar fácilmente con la base de datos.
- Allegrograph, Gephi y Proyecto R son herramientas que nos permiten realizar el análisis de redes sociales a partir de archivos RDF o tripletas. La mejor herramienta es Allegrograph porque permite trabajar con una gran cantidad de datos y a través de las las funciones SNA se puede realizar el análisis a un conjunto de datos (agrupando por predicados). Gephi y Proyecto R realizan el análisis a todo el archivo RDF, no hay como realizar filtros (agrupar por predicados). Proyecto R permite importar resultados directamente desde un endpoint con la desventaja de que tiene un límite de accesos mediante las consultas SPARQL.
- La implementación de las propiedades mágicas para realizar el Análisis de Redes Sociales aun es un proceso inmaduro, debido a que no todas las propiedades que están documentadas en la página oficial están disponibles para su uso, se presenta un mensaje que dice “No Results”, como sucedió al tratar de ejecutar “sna:cliquesOf” y “sna:groupDensity” (ver Ilustración 34).
Nota: Gary, unos de los empleados de Allegrograph, contestó “*Oh dear! It turns out that sna:groupDensity is documented but not implemented. We will remedy this oversight as quickly as possible. Please accept my apologies for the confusion*”, cuando se le escribió preguntando acerca del error.

Edit query

Query language: SPARQL ▾ Query planner: default ▾ Result limit: 100 ▾

```
1 prefix sna: <http://franz.com/ns/allegrograph/4.11/sna/>
2 prefix tw: <http://data.twitter.com/ontology#>
3
4 select ?density {
5   ?group sna:egoGroup (sna:coCreators tw:tw86927441 3) .
6   ?density sna:groupDensity (sna:coCreators ?group)
7 }
```

as

Result

Download as SPARQL JSON ▾

density

No results

Ilustración 34.- Error al ejecutar "sna:groupDensity"

RECOMENDACIONES

Después de haber realizado el trabajo de fin de titulación se ha llegado a una serie de conclusiones expuestas anteriormente, las mismas que me permiten presentar las siguientes recomendaciones:

- Se recomienda utilizar Allegrograph en una máquina virtual cuando se desee ejecutar pocos datos, caso contrario, tardaría demasiado tiempo en presentar los resultados debido a los bajos recursos. Si la cantidad de datos es elevada y no cuenta con Linux o MacOS, es mejor realizar una partición al disco e instalar los sistemas operativos compatibles con Allegro.
- Es aconsejable, antes de realizar la ejecución de las métricas de SNA, detener y reiniciar el servidor de Allegrograph porque se quedan almacenados en memoria los resultados anteriores y se puede interpretar erróneamente los resultados.
- Al utilizar el AGWebView, si se presenta el mensaje “No Results”, puede ser causado porque se escribe de manera incorrecta alguna de las funciones SNA, se debe tener cuidado al digitar, por ej. escribir mayúsculas en lugar de minúsculas (sna:DegreeCentrality en lugar de sna:degreeCentrality).
- El namespace y el prefijo con el que se trabaja las funciones SNA siempre van a ser “<http://franz.com/ns/allegrograph/4.11/sna>” y “sna” respectivamente, debido a que son estándares para el uso de estas propiedades. En la documentación oficial de Allegrograph presenta el siguiente error: se ha escrito el prefijo “san” para el namespace antes mencionado y al momento de ejecutar se presenta el error “No Results”.
- Revisar cuidadosamente la base de datos que se tiene, luego analizar y seleccionar las tablas necesarias para hacer el tratamiento de la información y así evitar que se eleve el tiempo de ejecución.
- En análisis de redes sociales se puede realizar utilizando SPARQL y Prolog, se recomienda utilizar SPARQL porque tiene potencial para ser explotado, es un lenguaje que permite recuperar información por medio de sentencias que se asemejan a las utilizadas en modelos relacionales (SQL), esto permite que las

personas que tenga conocimientos previos en modelos relaciones puedan ajustarse a la sintaxis de SPARQL.

- Cuando se aplica las métricas de SNA a 3217 tripletas se tarda aproximadamente 5 horas porque las características del computador personal (procesador i7, RAM 6GB) que se utiliza es de muy bajos recursos relacionado a los equipos (procesadores a partir de 32 cores y RAM 1TB en adelante) que usan los creadores de Allegrograph, ejecutan 1 billón de tripletas en tan solo 36 minutos con 49 segundos. Si se desea realizar el análisis de redes sociales a una gran cantidad de tripletas en menos tiempo se tiene que utilizar equipos con altos recursos.

TRABAJOS FUTUROS

Generador de RDF

Elaborar un aplicativo para generar RDF, en donde se realice el proceso de instanciación de los datos, a partir de una base de datos relacional y de esa manera evitar confusión cuando se crea clases, instancias y propiedades.

Aplicación Web

Desarrollar una aplicación Web para realizar el Análisis de Redes Sociales a través de magic properties o Prolog, de manera que el usuario pueda interactuar fácilmente y presentar los resultados en tablas y gráficos.

Análisis de Redes Sociales

En el presente proyecto se aplica el análisis de redes sociales a un grupo de datos relacionados con el hashtag “accessibility”, por tal razón, en trabajos futuros se podría realizar el análisis de las relaciones entre personas que hay en Twitter incluyendo menciones, retweets, following, followers.

BIBLIOGRAFÍA

- Robert van Hage, W. (s.f.). *SPARQL for R Tutorial - Hollywood Social Network Analysis*.
Obtenido de http://semanticweb.cs.vu.nl/R/sparql_hollywood/sparql_hollywood.html
- Apache Jena - RDF. (s.f.). *The core RDF Api*. Obtenido de
<https://jena.apache.org/documentation/rdf/index.html>
- Apache Jena. (2011). *Apache Jena*. Obtenido de <https://jena.apache.org/>
- Berners-Lee, T., Lassila, o., & Lassila, O. (2001). *The Semantic Web*. Obtenido de
<http://www.cs.umd.edu/~golbeck/LBSC690/SemanticWeb.html>
- Berrueta, D., & Brickley, D. (s.f.). *SIOC Core Ontology Specification*. Obtenido de
<http://www.w3.org/Submission/sioc-spec/>
- Brickley, D. (10 de febrero de 2004). *RDF Vocabulary Description Language 1.0: RDF Schema*. Obtenido de <http://www.w3.org/TR/rdf-schema/>
- Brickley, D. (s.f.). *Basic Geo (WGS84 lat/long) Vocabulary*. Obtenido de
<http://www.w3.org/2003/01/geo/>
- Cardozo, G. (s.f.). *Historia del concepto de red social*. Obtenido de
http://api.ning.com/files/XTj6PLCPPuFqdfb7UgUqoC3MJP2Gp8uETdvZtgGysgOtkjOSDtmFBx*D5*s9HLJlrxDDys-Q3WRdjAwQBay*fSCWgNomHy/Historiadelconceptoderedsocial.pdf
- Corbera, S. (Diciembre de 2007). *Estudio sobre Sistemas de Gestión del Conocimiento para la Web Semántica*. Obtenido de
http://163.117.147.101/SEMSE/index2.php?option=com_docman&task=doc_view&gid=14&Itemid=39
- De la Rosa Troyano, F. (s.f.). *Sistemas de Inteligencia Web: Análisis de Redes Sociales*.
Obtenido de <http://www.lsi.us.es/~ffrosat/publicaciones/Tesis.2011.v75.vf.pdf>
- De Vocht, L. (Junio de 2011). *Researcher Profiling based on Semantic Analysis in Social Networks*. Obtenido de
<http://laurengoessemsemantic.files.wordpress.com/2011/07/masterproef.pdf>
- Demairy, E. (s.f.). *Semantic Web Import*. Obtenido de
<https://marketplace.gephi.org/plugin/semanticwebimport/>
- Ereteo, G., Gandon, F., & Buffa, M. (s.f.). *Semantic Social Networks Analysis*. Obtenido de
<http://arxiv.org/ftp/arxiv/papers/0904/0904.3701.pdf>
- Franz, I. (2013). *AllegroGraph 4.10 Social Networking Analysis*. Obtenido de
<http://www.franz.com/agraph/allegrograph/>
- Franz, I. (s.f.). *AllegroGraph for Social Network Analysis*. Obtenido de
http://www.franz.com/agraph/services/conferences_seminars/Franz_Webinar_5-7-08.pdf
- Franz, I. (s.f.). *Jena Tutorial*. Obtenido de
<http://www.franz.com/agraph/support/documentation/current/java-tutorial/jena-tutorial-40.html>
- Freeman, L. (1979). *Centrality in Social Networks Conceptual Clarification*. Obtenido de
<http://leonidzhukov.ru/hse/2013/socialnetworks/papers/freeman79-centrality.pdf>
- Gephi. (s.f.). *Gephi*. Obtenido de <https://gephi.org/>
- Gil, E. (s.f.). *Twitter*. Obtenido de <http://blogfolioeduingil.blogspot.com/2013/02/twitter.html>
- Glahn, J. (20 de Julio de 2009). *DBPedia Deutschland 1.0 Release - The German part of the Wikipedia for Machines*. Obtenido de <http://www.corporate-semantic-web.de/reader/items/dbpedia-deutschland-10-release-the-german-part-of-the-wikipedia-for-machines.html>
- Golbeck, J. (s.f.). *Semantic Web Interaction trough Trust Network Recommender Systems*.

- Obtenido de <http://www.cs.umd.edu/~golbeck/pubs/Golbeck%20-%202005%20-%20Semantic%20Web%20Interaction%20through%20Trust%20Network%20Recommender%20Systems.pdf>
- Gruber, T. (s.f.). *Ontology*. Obtenido de <http://tomgruber.org/writing/ontology-definition-2007.htm>
- Hanneman, R. (s.f.). *INTRODUCCIÓN A LOS MÉTODOS DEL ANÁLISIS DE REDES SOCIALES*. Obtenido de <http://www.ceunes.ufes.br/downloads/2/jjuniorsan-Manual%20-%20HANNEMAN%20-%20COMPLETO%20-%20Espanhol.pdf>
- Hogan, A., Harth, A., & Decker, S. (s.f.). *ReConRank: A Scalable Ranking Method for Semantic Web Data with Context*. Obtenido de National University of Ireland: <http://ir.library.nuigalway.ie/xmlui/bitstream/handle/10379/492/paper.pdf?sequence=1>
- Kobayashi, I. (30 de Junio de 2007). *A study on an Information Recommendation System that Provides Tropical Information Related to User's Inquiry for Information Retrieval*. Obtenido de http://download.springer.com/static/pdf/306/art%253A10.1007%252Fs00354-007-0033-5.pdf?auth66=1363276895_987fff11034dd3eb7326f4ae914672f4&ext=.pdf
- Marshall, C., & Shipman, F. (s.f.). *Which Semantic Web?* Obtenido de <http://www.csd.tamu.edu/~marshall/ht03-sw-4.pdf>
- McGuinness, D., & Van Harmelen, F. (s.f.). *OWL Web Ontology Language*. Obtenido de <http://www.w3.org/TR/owl-features/>
- Mejia, C. P. (Febrero de 2010). *Análisis de Redes Sociales a Gran Escala*. Obtenido de <http://www.cs.cinvestav.mx/TesisGraduados/2010/tesisCristianMejia.pdf>
- Miller, L., & Brickley, D. (s.f.). *FOAF Vocabulary Specification 0.98*. Obtenido de <http://xmlns.com/foaf/spec/>
- Monsalve, M. (11 de Diciembre de 2008). *Análisis de redes sociales: un tutorial*. Obtenido de <http://users.dcc.uchile.cl/~mnmonsal/SNA.pdf>
- Morstatter, F., Pfeffer, J., Liu, H., & Carley, K. (2013). *Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose*. Obtenido de <http://www.public.asu.edu/~fmorstat/paperpdfs/icwsm2013.pdf>
- Passant, A., & Laublet, P. (28 de Abril de 2008). *Meaning Of A Tag: A Collaborative Approach to Bridge the Gap Between Tagging and Linked Data*.
- Polanco, X. (2008). *Análisis de redes: Introducción*. Obtenido de http://hal.archives-ouvertes.fr/docs/00/21/83/97/PDF/XP_Analisis_de_redes_2006.pdf
- Project R. (s.f.). *The R Project for Statistical Computing*. Obtenido de <http://www.r-project.org/>
- Ranga, G. (s.f.). *Building a Pharmaceutical Competitive Landscape using Semantic Technologies*. Obtenido de Eli Lilly & Company: http://www.iscb.org/cms_addon/conferences/cshals2009/presentations/GudivadaCFeb09.pdf
- Rodríguez, J., & Mérida, F. (s.f.). *Guía práctica de Redes Sociales*. Obtenido de webcasus.usal.es/edenred/documentos/guia_.doc
- Rodríguez, K., & Ronda, R. (s.f.). *Web Semántica: un nuevo enfoque para la organización y recuperación de información en la web*. Obtenido de ACIMED: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352005000600003
- Sanz, L. (Junio de 2003). *Análisis de Redes Sociales: o como representar las estructuras sociales subyacentes*. Obtenido de <http://digital.csic.es/bitstream/10261/1569/1/dt-0307.pdf>
- Senso, J., & De la Rosa Piñero, A. (s.f.). *Evolución del Dublin Core Metadata Initiative*. Obtenido de <http://www.ugr.es/~jsenso/curriculum/dcmi.pdf>

- Stanford Center for Biomedical Informatic Research. (s.f.). *Protégé*. Obtenido de <http://protege.stanford.edu/>
- Suárez-Figueroa, M., & Gómez-Pérez, A. (s.f.). *Reusing Domain Ontologies as a Whole*. Obtenido de <http://www.neon-project.org/web-content/media/book-chapters/Chapter-10.pdf>
- Tweet Developers. (s.f.). *Tweets*. Obtenido de <https://dev.twitter.com/docs/platform-objects/tweets>
- Universidad de Chile. (s.f.). *Revista BITS de Ciencia*. Obtenido de http://dcc.comopapel.com/files/revista_bits_de_ciencia/2/download/Revista%20BITS%20de%20Ciencias%20Vol%202.pdf
- Velázquez, A., & Aguilar, N. (Junio de 2005). *Manual Introductorio al Análisis de Redes Sociales*. Obtenido de http://revista-redes.rediris.es/webredes/talleres/Manual_ARIS.pdf
- W3C. (2008). *W3C Sparql Query Language for RDF*. Obtenido de <http://www.w3.org/TR/rdf-sparql-query/>
- W3C. (s.f.). *Sparql 1.1*. Obtenido de <http://www.w3.org/TR/sparql11-overview/>
- Zaino, J. (17 de Mayo de 2010). *Making a Semantic Web Business Case at Pfizer*. Obtenido de http://semanticweb.com/making-a-semantic-web-business-case-at-pfizer_b621

ANEXOS

A. Crear ontología en Protégé

Protégé es un editor de ontologías de código abierto, soporta el modelado a través de un cliente Web o un cliente de escritorio, las ontologías se pueden desarrollar en una variedad de formatos, incluyendo OWL y XML Schema.

El enlace para realizar la descarga del programa es el siguiente:
<http://protege.stanford.edu/download/registered.html>.

A continuación se detalla los pasos para crear una ontología, lo primero que se debe realizar es añadir un IRI²⁰ a la ontología.

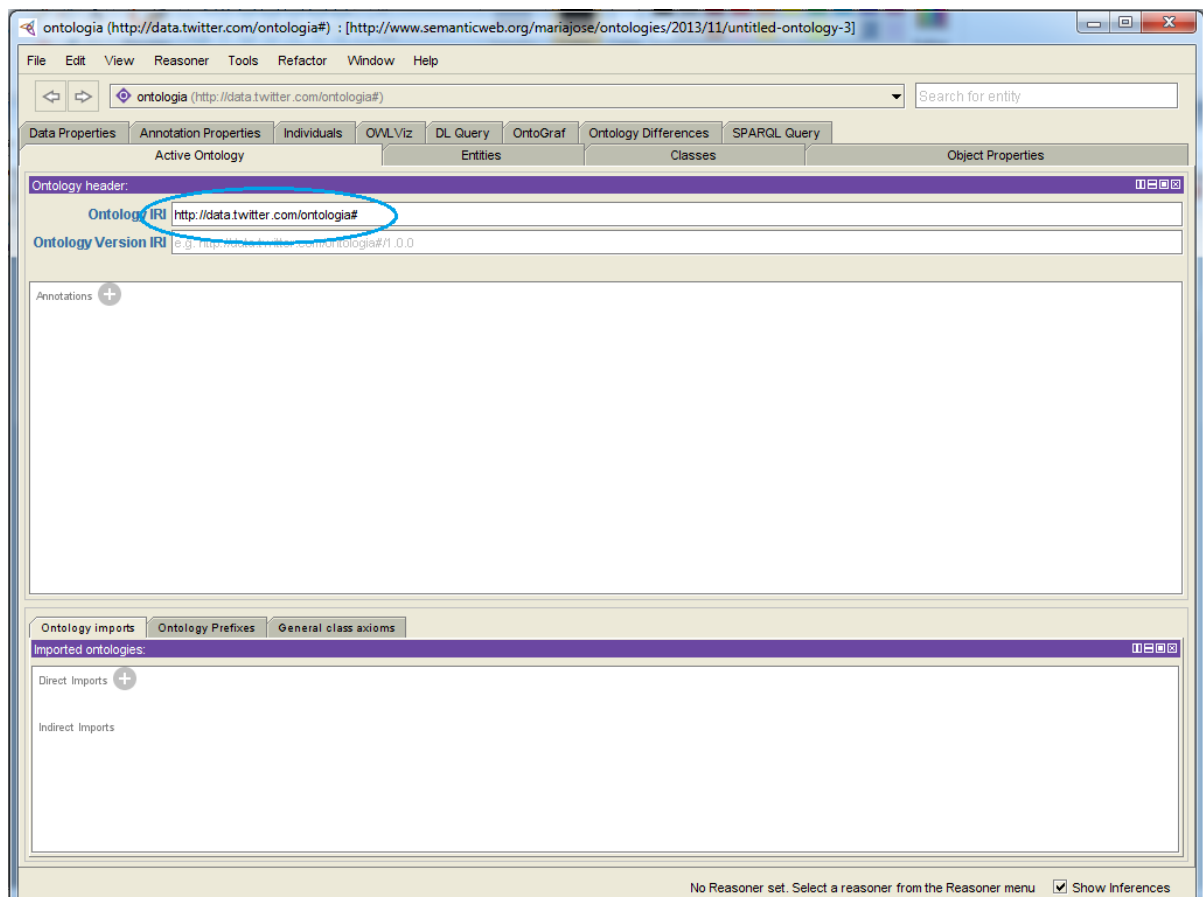


Ilustración 35.- Añadir IRI a la ontología

²⁰ IRI.- Internationalized Resource Identifiers

Crear namespaces y prefijos que se van a utilizar en la ontología: clic en “Ontology Prefixes”, clic en ícono “Add prefix”, ingresar el prefijo y su correspondiente valor.

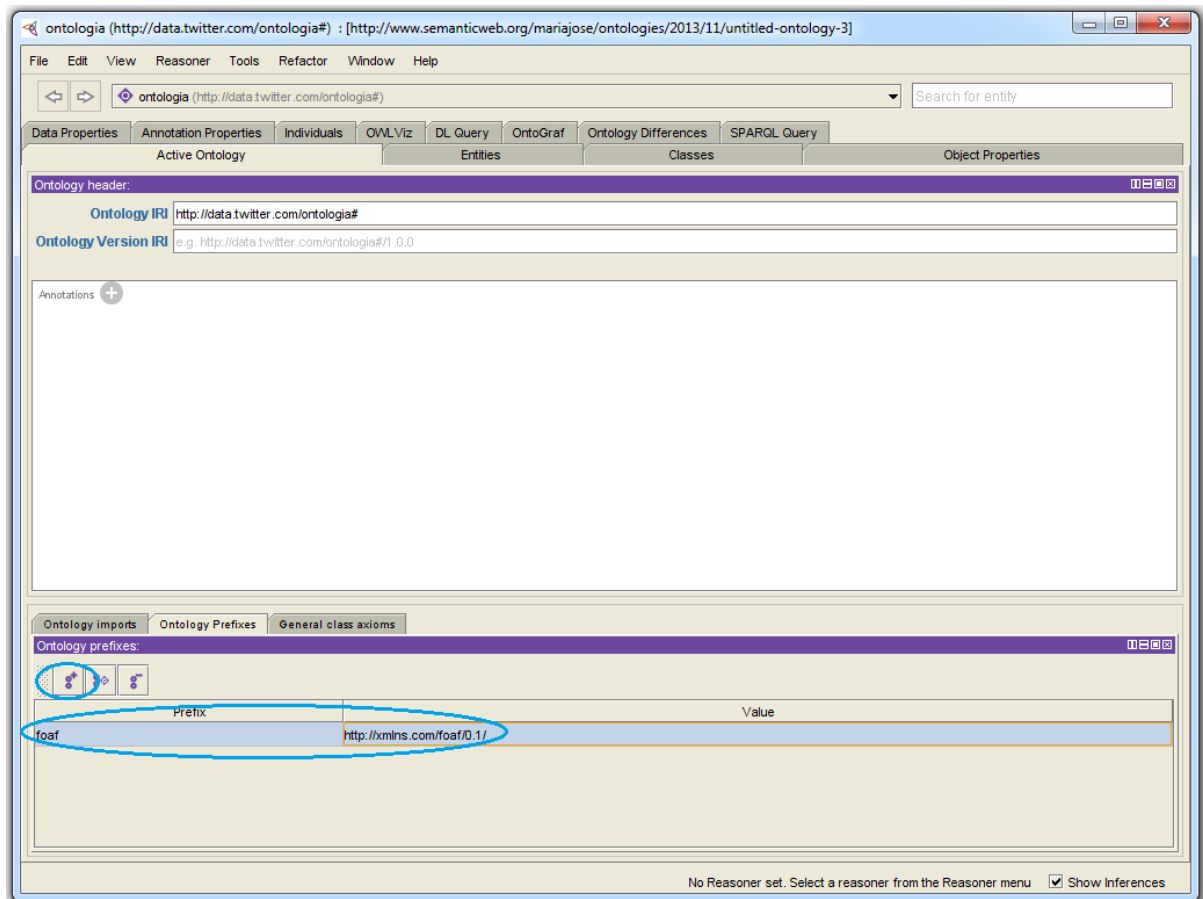


Ilustración 36.- Crear namespaces y prefijos

Para crear los recursos que en Protégé se les denomina clases: clic en “Classes”, clic en “Create new OWLClass” e ingresar el nombre de la clase, el IRI es por defecto el que se definió al iniciar Protégé.

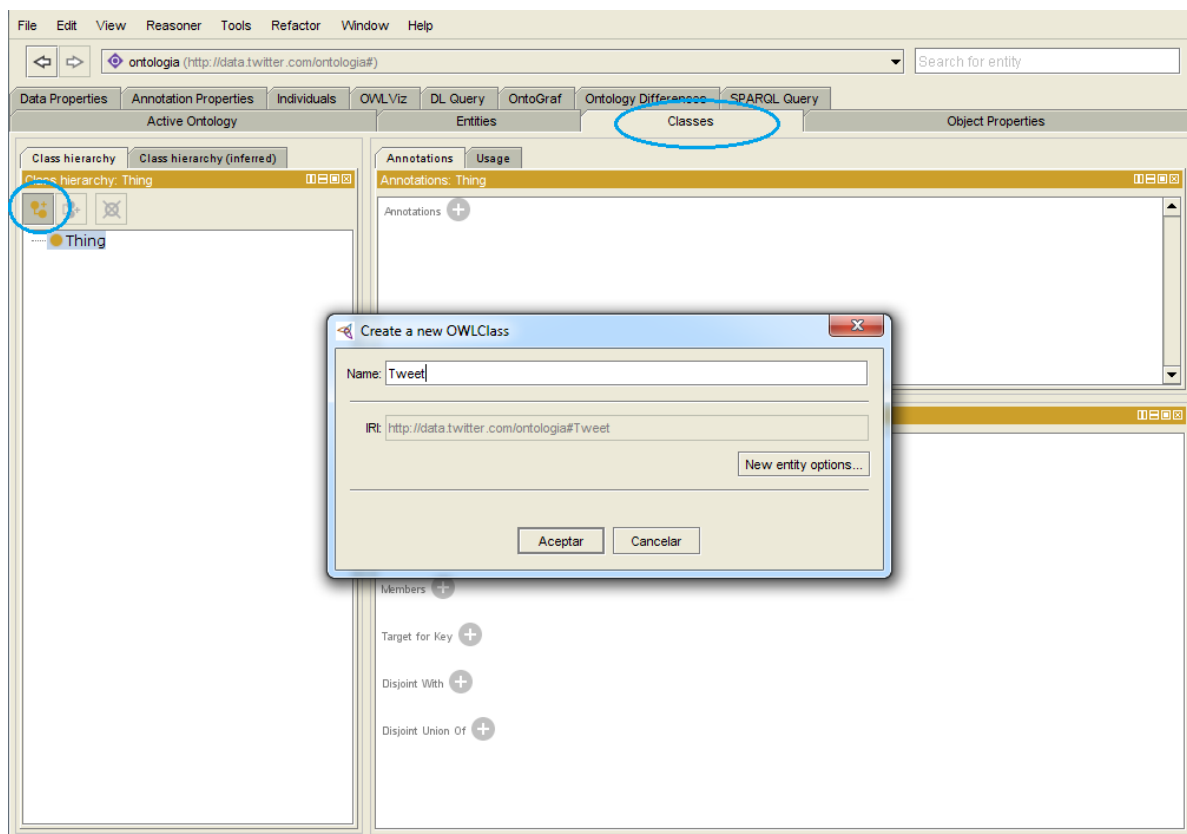


Ilustración 37.- Crear recursos

Para crear object property: clic en “Object Properties”, clic en “Create a new OWLObjectProperty e ingresar el nombre de la propiedad.

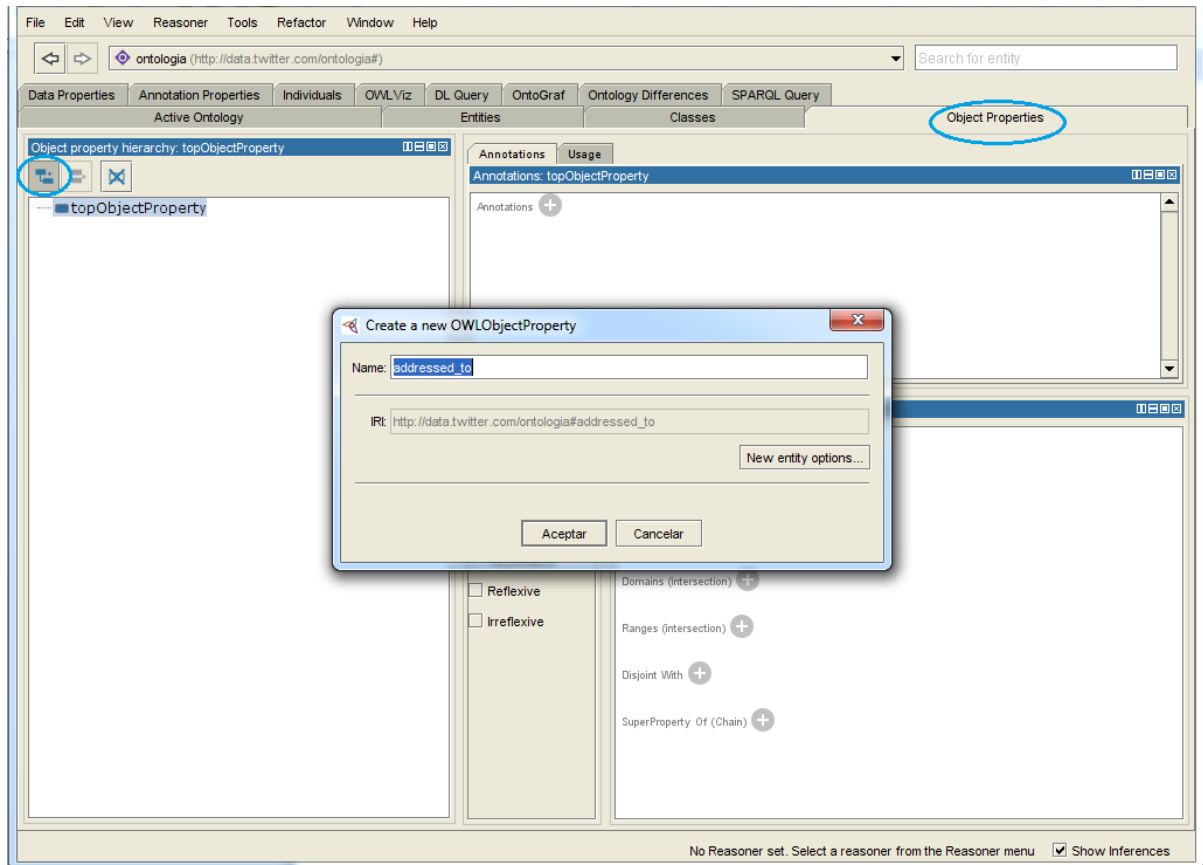


Ilustración 38.- Crear Object Property

Para crear data property: clic en “Data Properties”, clic en “Create a new OWLDataProperty”, al ingresar un nombre a una propiedad hay como realizarla de dos maneras, la primera es escribiendo un nombre y el IRI por defecto es el que se lo definió al inicio, la segunda es digitar el prefijo y un nombre separado por los dos puntos (ej. foaf:name) y automáticamente el IRI se actualizará al namespace correspondiente al prefijo.

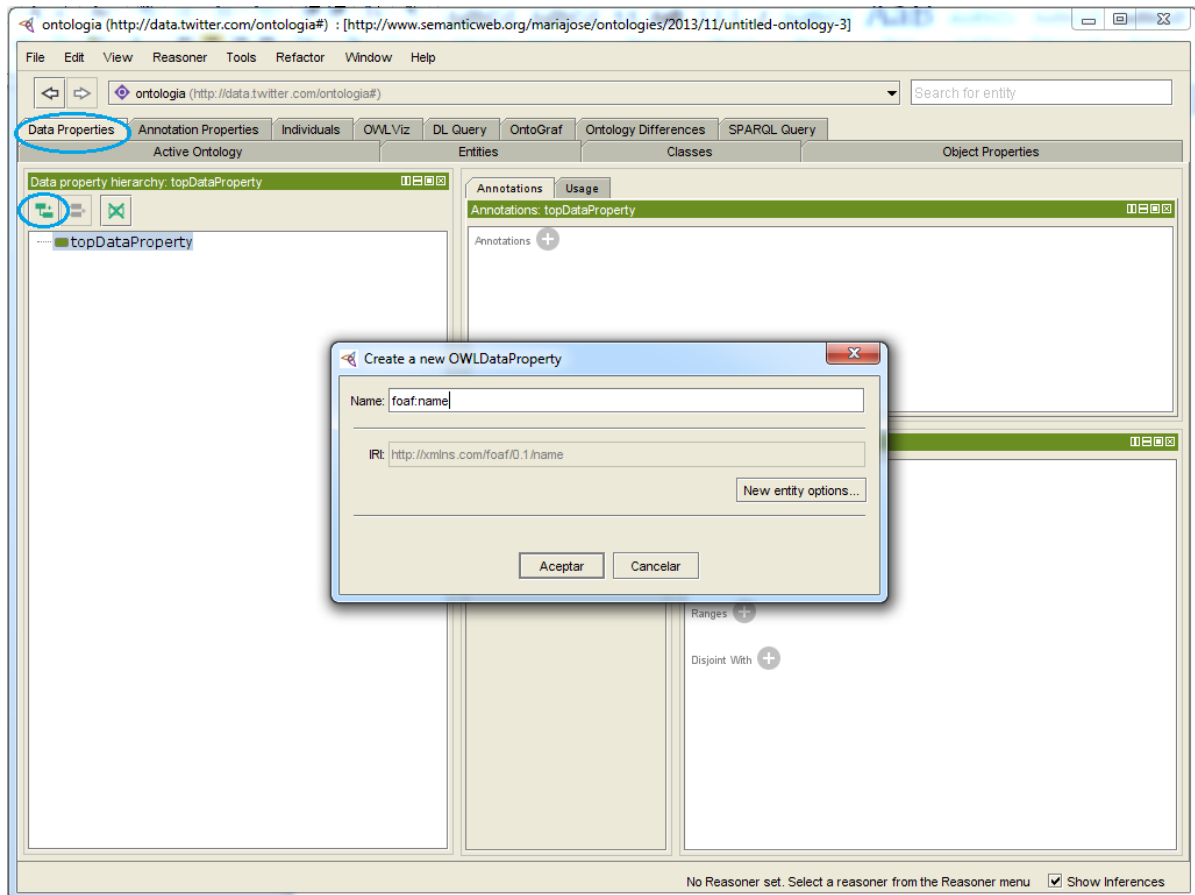


Ilustración 39.- Crear Data Property

Para guardar la ontología: clic en “File”, clic en “Save” y se mostrará una pantalla donde se debe elegir el formato, en este caso clic en RDF/XML, luego en guardar.

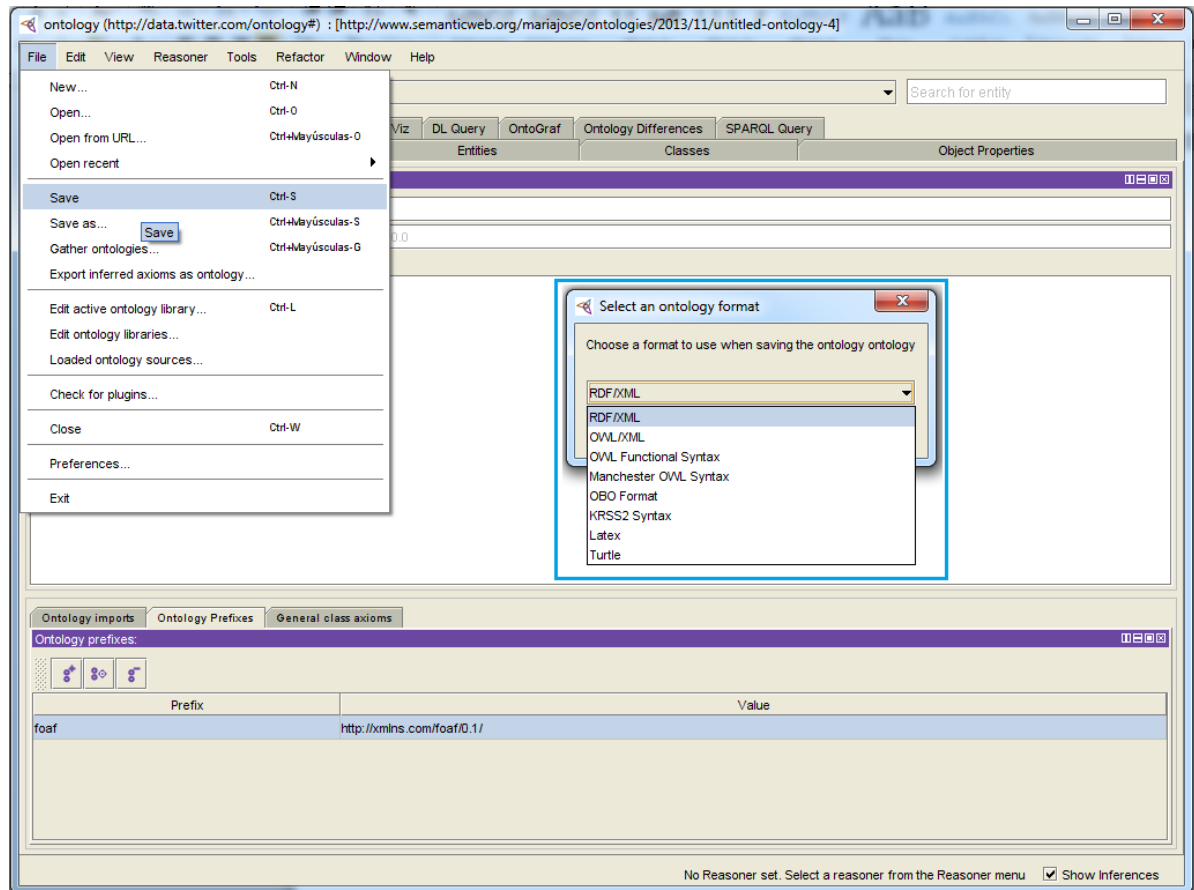
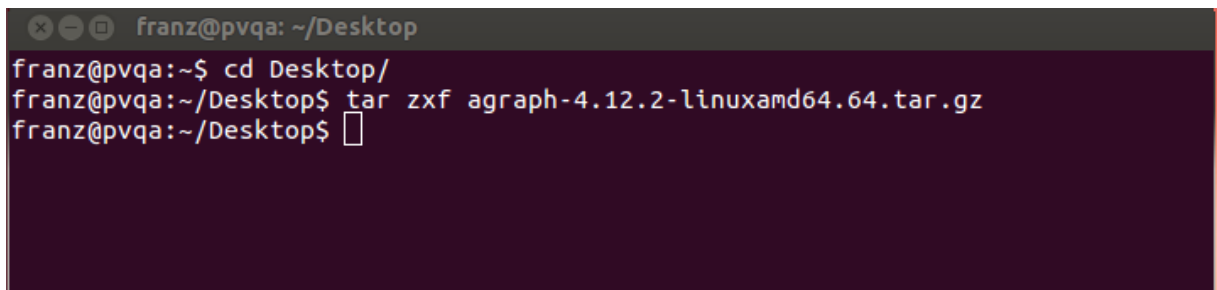


Ilustración 40.- Guardar ontología

B. Instalar Allegrograph

Allegrograph se ejecuta en forma nativa en Linux x86-64bits, en Windows y Mac se sugiere instalar una máquina virtual con sistema operativo Linux, accediendo al siguiente enlace http://www.franz.com/downloads/clp/ag_validate_survey se puede descargar los archivos: para Ubuntu tar.gz, Fedora .rpm y para los usuarios Windows y Mac se encuentra una imagen de la máquina virtual en donde ya vienen instalados el servidor, el Gruff, y la versión Web.

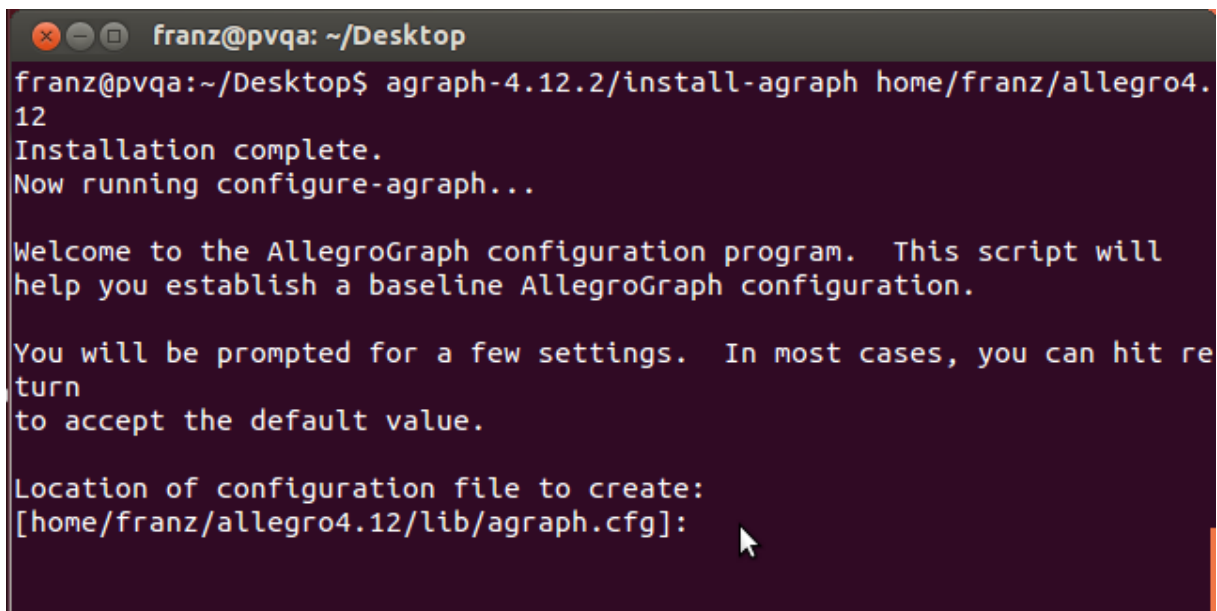
Para instalar Allegrograph en Ubuntu se descarga el archivo con extensión tar.gz, abrir un terminal y acceder a la carpeta donde se encuentra el archivo, descomprimir utilizando el comando “tar xzf archivo.tar.gz”.



```
franz@pvqa: ~/Desktop
franz@pvqa:~$ cd Desktop/
franz@pvqa:~/Desktop$ tar xzf agraph-4.12.2-linuxamd64.64.tar.gz
franz@pvqa:~/Desktop$
```

Ilustración 41.- Descomprimir archivo tar.gz

Luego, mediante línea de comandos escribir la ruta donde se encuentra el archivo “install-agraph”, dar un espacio y la ruta donde se quiere instalar el triple-store.



```
franz@pvqa: ~/Desktop
franz@pvqa:~/Desktop$ agraph-4.12.2/install-agraph home/franz/allegro4.12
Installation complete.
Now running configure-agraph...

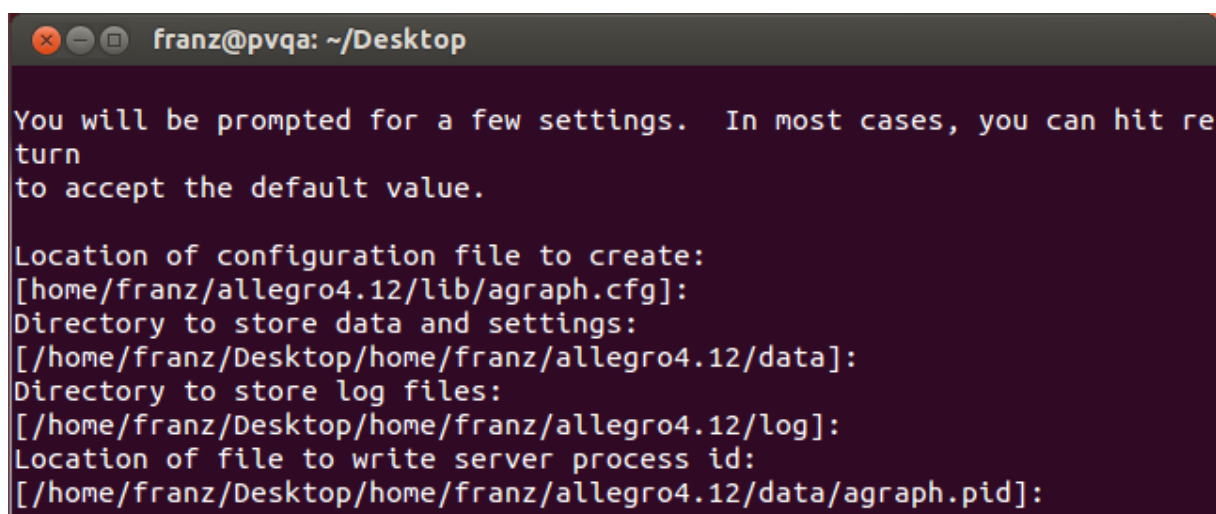
Welcome to the AllegroGraph configuration program. This script will help you establish a baseline AllegroGraph configuration.

You will be prompted for a few settings. In most cases, you can hit return to accept the default value.

Location of configuration file to create:
[home/franz/allegro4.12/lib/agraph.cfg]:
```

Ilustración 42.- Instalar Allegrograph

Durante la instalación se solicitará hacer algunas configuraciones, de preferencia aceptar los valores por defecto dando un “enter”.

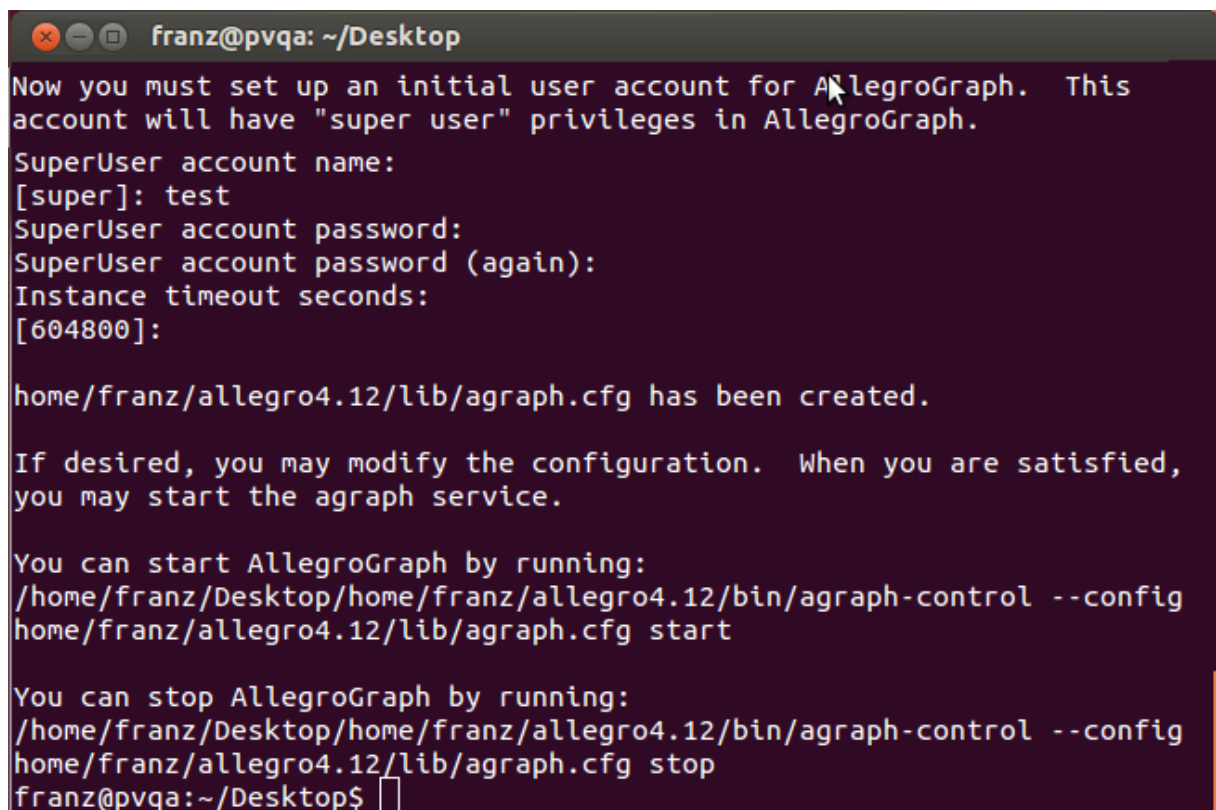


```
franz@pvqa: ~/Desktop
You will be prompted for a few settings.  In most cases, you can hit re
turn
to accept the default value.

Location of configuration file to create:
[home/franz/allegro4.12/lib/agraph.cfg]:
Directory to store data and settings:
[/home/franz/Desktop/home/franz/allegro4.12/data]:
Directory to store log files:
[/home/franz/Desktop/home/franz/allegro4.12/log]:
Location of file to write server process id:
[/home/franz/Desktop/home/franz/allegro4.12/data/agraph.pid]:
```

Ilustración 43.- Instalar Allegrograph, realizar configuraciones

El siguiente paso es establecer una cuenta de usuario inicial para Allegrograph, esta cuenta debe tener privilegios de súper usuario. En el nombre de la cuenta ingresar “test” y en contraseña “xyzzzy”.



```
franz@pvqa: ~/Desktop
Now you must set up an initial user account for AllegroGraph.  This
account will have "super user" privileges in AllegroGraph.
SuperUser account name:
[super]: test
SuperUser account password:
SuperUser account password (again):
Instance timeout seconds:
[604800]:

home/franz/allegro4.12/lib/agraph.cfg has been created.

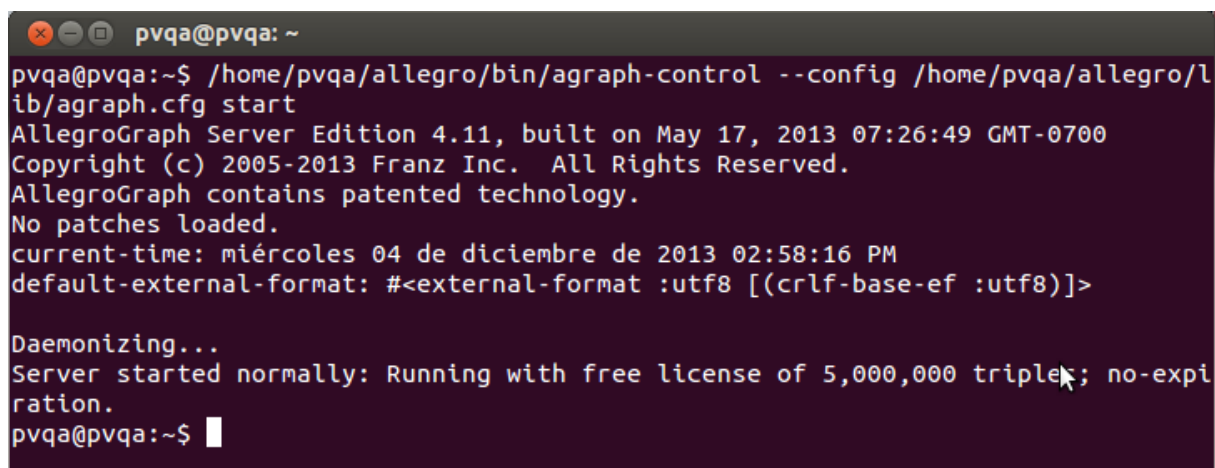
If desired, you may modify the configuration.  When you are satisfied,
you may start the agraph service.

You can start AllegroGraph by running:
/home/franz/Desktop/home/franz/allegro4.12/bin/agraph-control --config
home/franz/allegro4.12/lib/agraph.cfg start

You can stop AllegroGraph by running:
/home/franz/Desktop/home/franz/allegro4.12/bin/agraph-control --config
home/franz/allegro4.12/lib/agraph.cfg stop
franz@pvqa:~/Desktop$
```

Ilustración 44.- Instalar Allegrograph, nombre de la cuenta y contraseña

Para iniciar Allegrograph ejecutar el siguiente comando: “ruta-allegro/bin/agraph-control --config ruta-allegro/lib/agraph.cfg start”.



```
pvqa@pvqa: ~  
pvqa@pvqa:~$ /home/pvqa/allegro/bin/agraph-control --config /home/pvqa/allegro/lib/agraph.cfg start  
AllegroGraph Server Edition 4.11, built on May 17, 2013 07:26:49 GMT-0700  
Copyright (c) 2005-2013 Franz Inc. All Rights Reserved.  
AllegroGraph contains patented technology.  
No patches loaded.  
current-time: miércoles 04 de diciembre de 2013 02:58:16 PM  
default-external-format: #<external-format :utf8 [(crlf-base-ef :utf8)]>  
  
Daemonizing...  
Server started normally: Running with free license of 5,000,000 triples; no-expiration.  
pvqa@pvqa:~$
```

Ilustración 45.- Iniciar Allegrograph

Una de las maneras de comprobar si se ha instalado correctamente es acceder a cualquier navegador e ingresar a la siguiente dirección “localhost:100535”, se mostrará una pantalla para hacer loguearse como en la siguiente imagen:

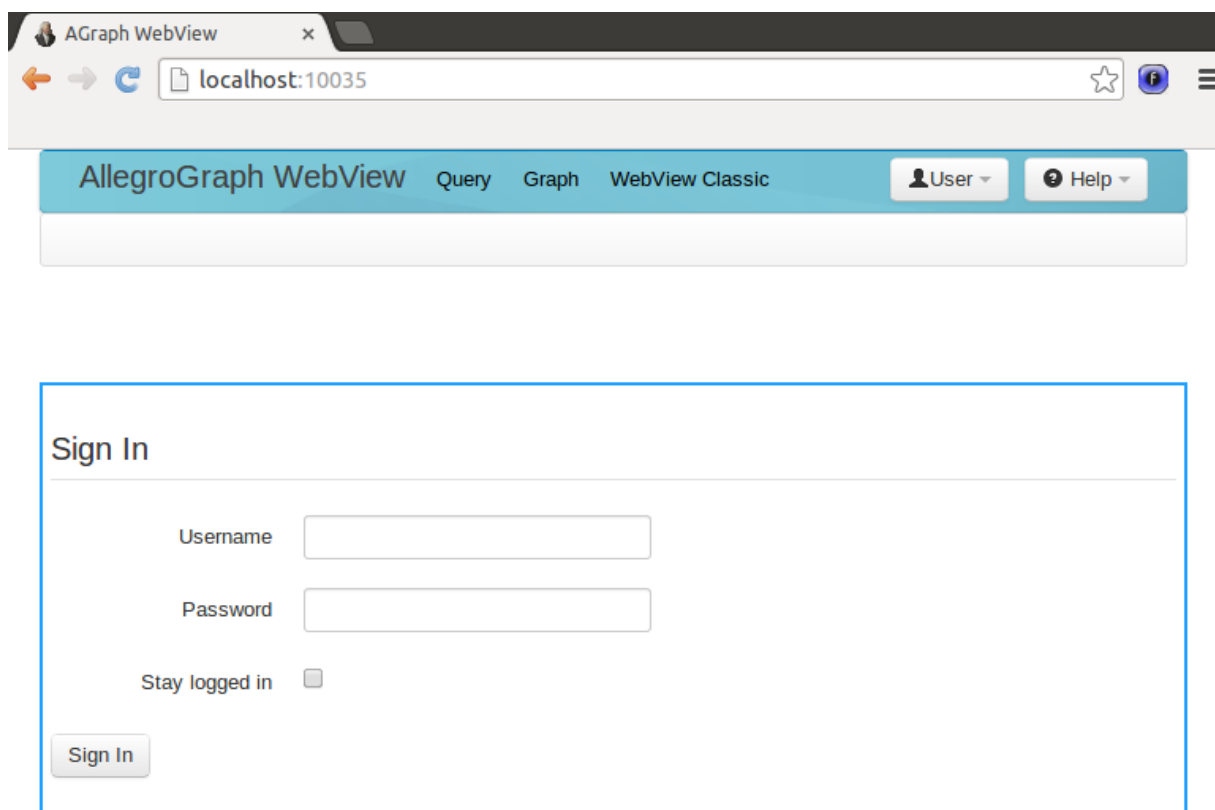


Ilustración 46.- Comprobar si Allegrograph está instalado correctamente

C. Instalar y configurar gruff

El archivo de descarga del gruff es .zip para Windows, .tar.gz para Linux y dmg para Mac OS X, se extrae los archivos en la ubicación preferida, Para descargar el gruff, ingresar al siguiente enlace <http://www.franz.com/agraph/gruff/index.html#download>.

Para iniciar el gruff se ejecuta el archivo “gruff”, como es un navegador interactivo para Allegrograph se necesita iniciar el servidor, para Linux se utiliza el siguiente comando:

```
/home/pvqa/allegro/bin/agraph-control --config /home/pvqa/allegro/lib/agraph.cfg start
```

Crear un nuevo triple store: clic en “File”, clic en “New Triple-Store”.

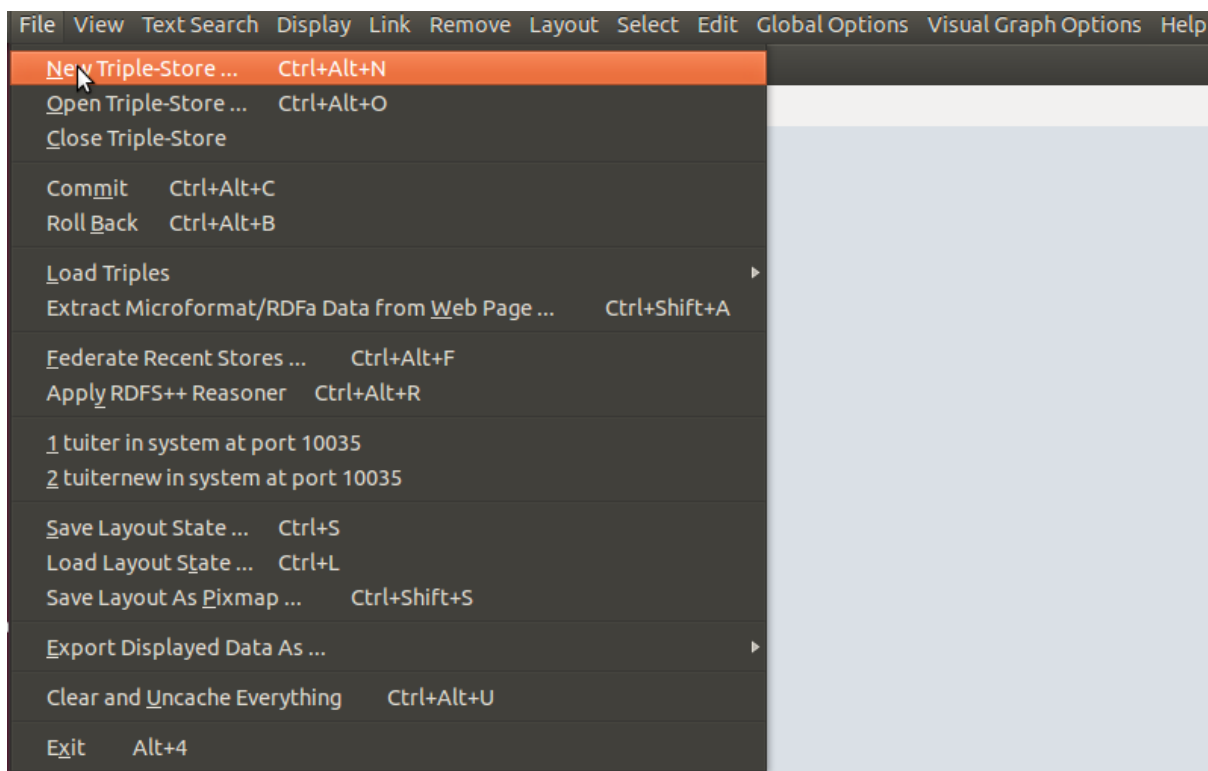


Ilustración 47.- Nuevo triple-store

Se muestra una pantalla donde se debe ingresar: el nombre de la máquina del servidor (server machine), el número de puerto que por defecto para Allegrograph es 10035 (port number), el usuario para acceder al servidor es “test” (user), la contraseña es “xyzyy” (password), seleccionar el catálogo por defecto que es “system” y finalmente ingresar un nombre al store que se va a crear.

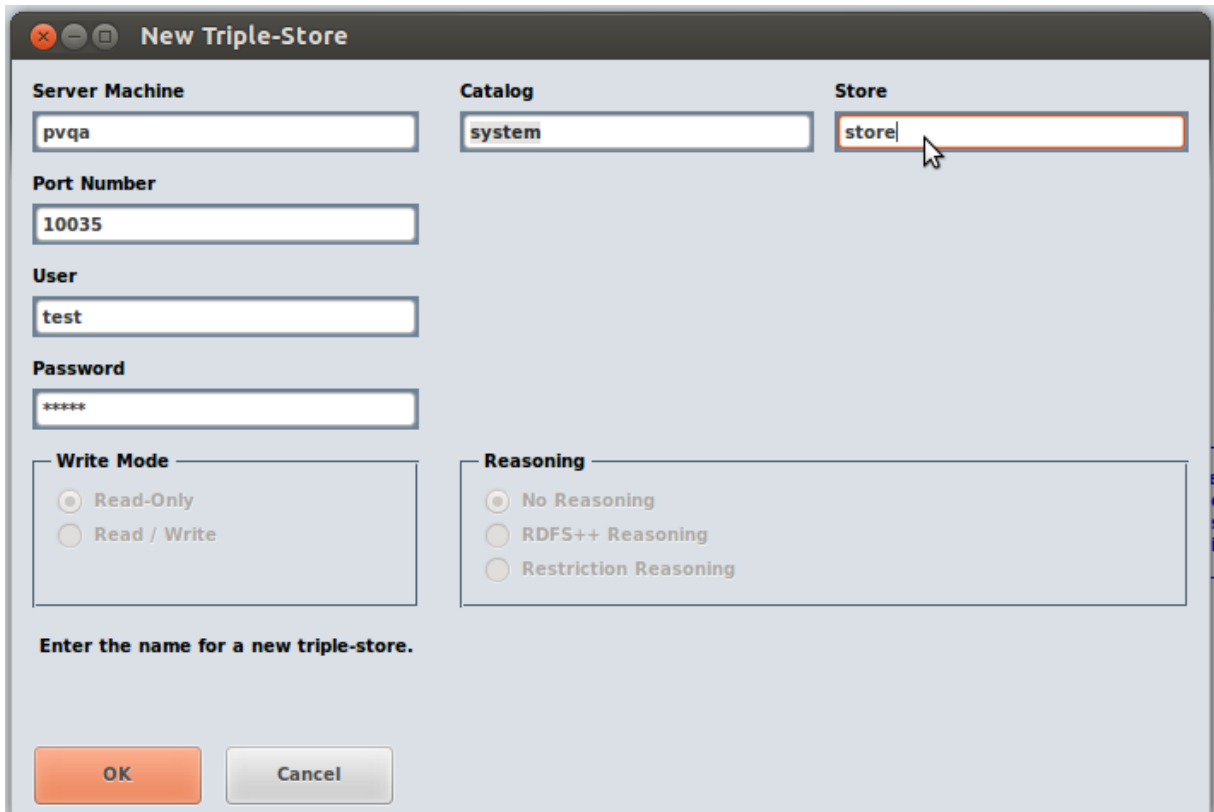


Ilustración 48.- Configurar el nuevo triple-store

Ya se tiene un espacio creado para almacenar las tripleteas, se procede a cargar el archivo RDF en el triple-store: clic en “File”, clic en “Load Triples”, clic en “RDF/XML”. Seleccionar desde donde se desea cargar el archivo, en este caso, clic en “File”, buscar el archivo y clic en “Ok”.

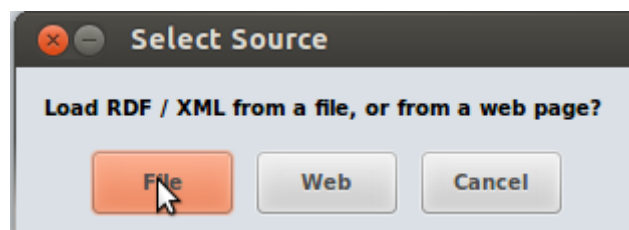


Ilustración 49.- Cargar RDF/XML

Si en el servidor se encuentra algún triple-store y se lo desea abrir, debe seguir lo siguientes pasos: clic en “File”, clic en “Open Triple-Store”.

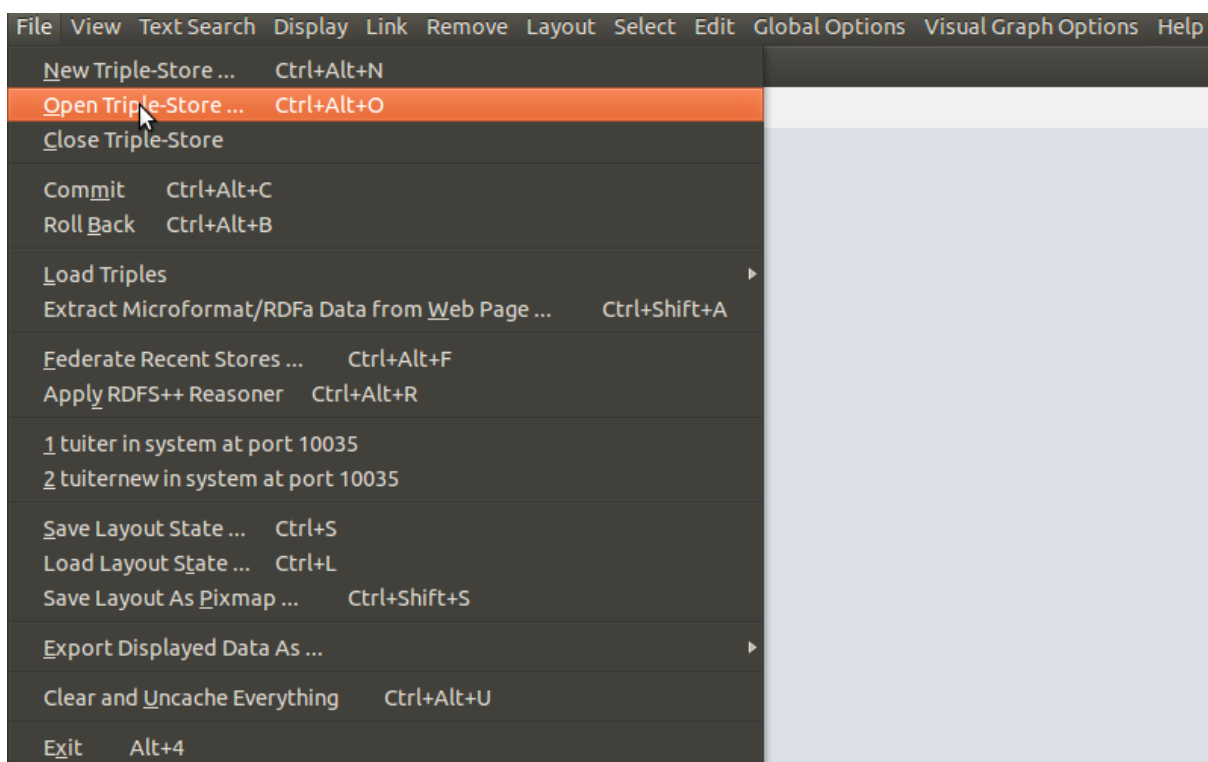


Ilustración 50.- Abrir triple-store

Para abrir el triple-store se debe ingresar la siguiente información: nombre de la máquina (server machine), número de puerto 10035 por defecto (port number), usuario “test” (user) y contraseña “xyzyzy” (password), seleccionar un catálogo y el store.

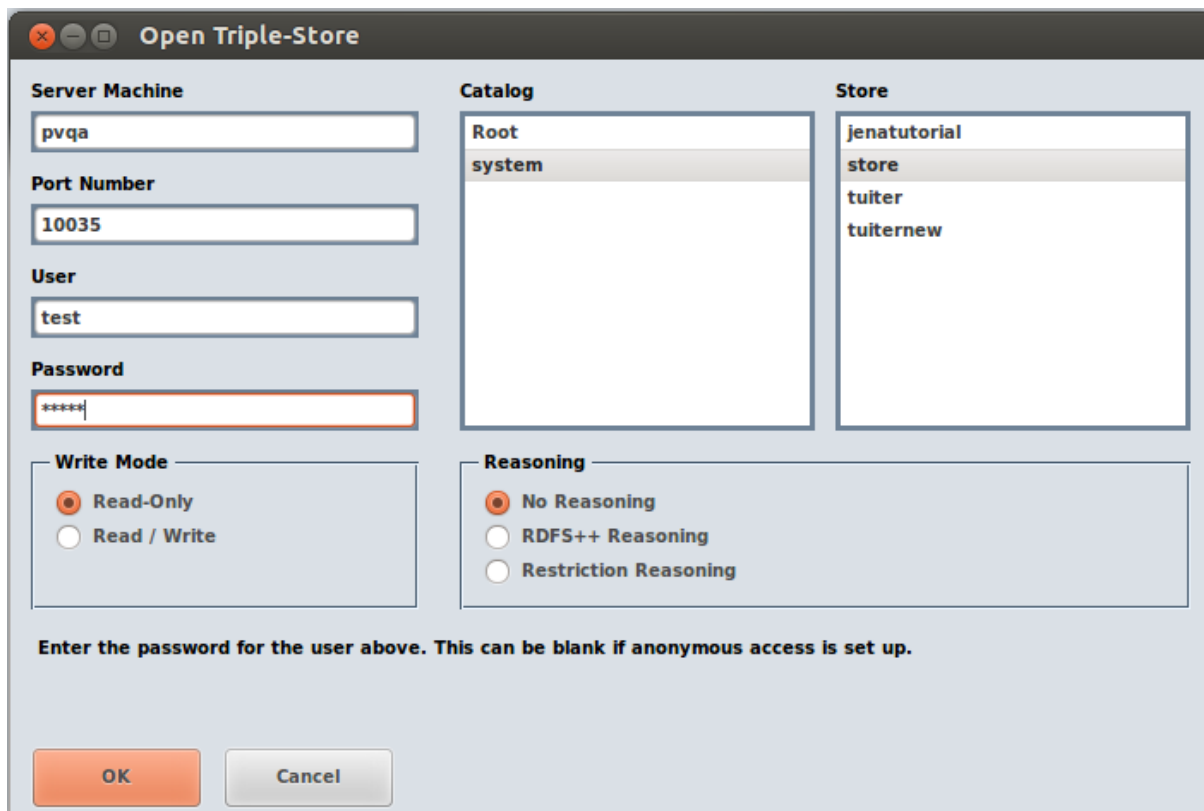


Ilustración 51.- Configurar al abrir un triple-store

Para mostrar en el gruff el gráfico con todos los nodos y sus relaciones: clic en “Display”, clic en “Display All Triples”.

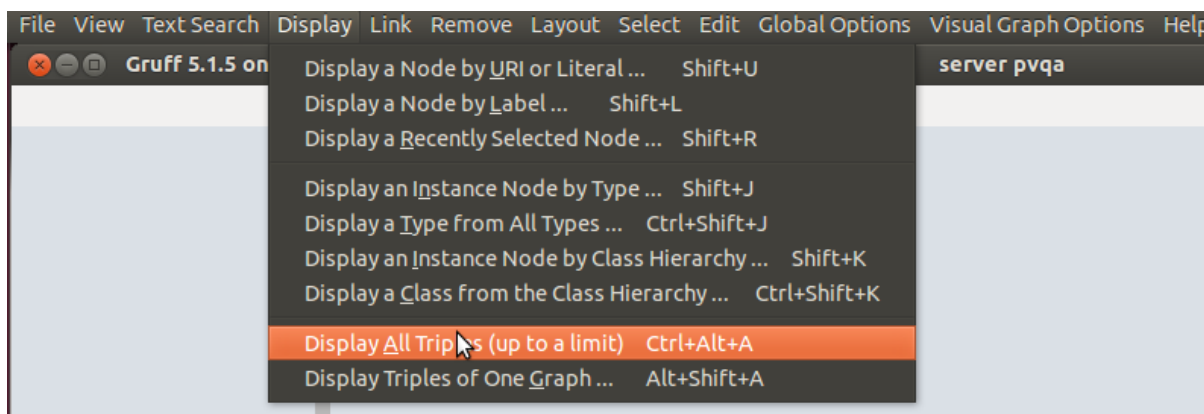


Ilustración 52.- Mostrar todas las tripletas

Cuando se abre o crea un triple-store, el gráfico de nodos en el gruff se muestra como en la siguiente imagen:

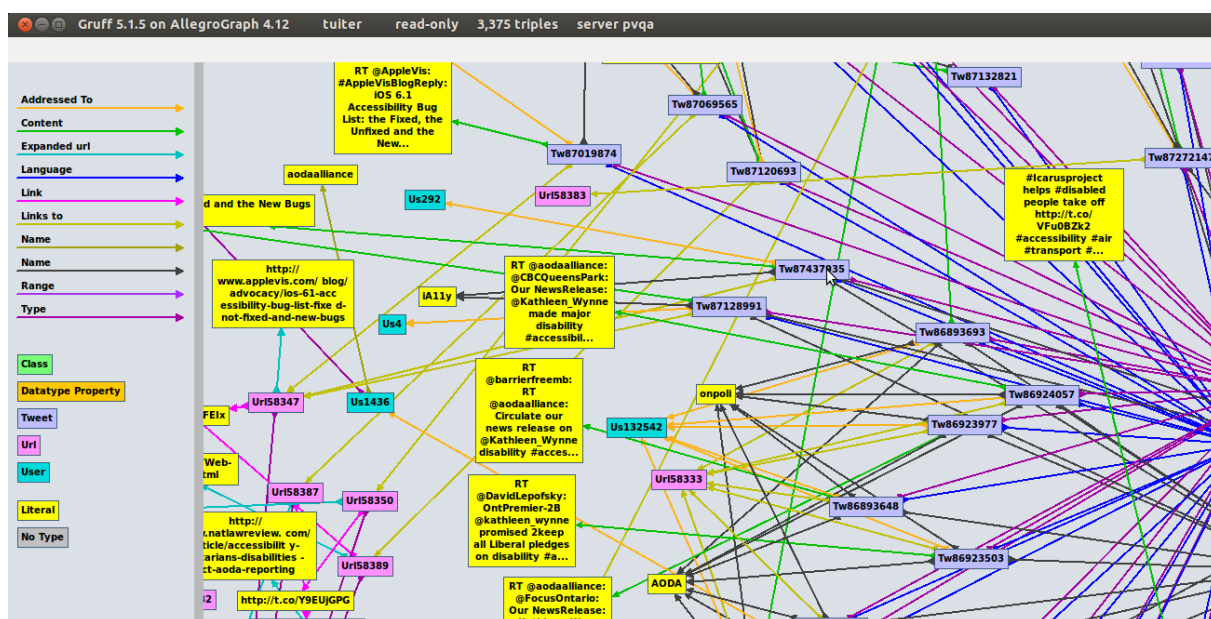


Ilustración 53.- Gráfico de nodos en el gruff

Una de las ventajas de utilizar esta herramienta es que se puede hacer consultas SPARQL, para realizar las mismas: clic en “View”, clic en “Query View” o simplemente presionando la tecla “W”. Además, presenta las opciones de crear un gráfico visual, escribir un reporte de texto y guardar el resultado en un archivo CSV.

Query

```
SELECT ?s ?p ?o WHERE {
  ?s ?p ?o .
}
```

1000 Results

?s	?p	?o
Tw87101934	Type	Tweet
Tw87101934	Addressed To	Us284
Tw87101934	Content	MRT @CRPDisabilities Game #Accessibility Information Symbol - http://t.co/TWW7BoIN #a11y
Tw87101934	Name	accessibility
Tw87101934	Name	a11y
Tw87101934	Language	En
Tw87101934	Links to	Uri58421
Tw87077510	Type	Tweet
Tw87077510	Content	Monitor urban accessibility issues #accessibility http://t.co/0L54KPC...

Explicit Nodes from Query

Explicit Predicates from Query

Ilustración 54.- Query View

Si se desea hacer seguimiento a un nodo en específico se utiliza la vista de tabla: clic en “View”, clic en “Table View” o presionando la tecla “T”. Haciendo clic en cada una de las propiedades o a los valores se puede acceder a más información.

The screenshot shows a web browser window with the title 'Gruff 5.1.5 on AllegroGraph 4.12'. The address bar shows 'tulner read-only 3,375 triples server pvqa'. The main content area displays a table view for a specific node, 'Tw87101934'. The table has two columns: 'Property' and 'Value'. The 'Property' column contains the following entries: 'Addressed To', 'Content', 'Language', 'Links to', 'Name', and 'Type'. The 'Value' column contains the following entries: 'Us284', 'MRT @CRPDIsabilities Game #Accessibility Information Symbol - http://t.co/WW7BoIN #a11y', 'En', 'Uri58421', 'a11y accessibility', and 'Tweet'. The table is styled with alternating light blue and white rows. Above the table, there are navigation buttons: 'Revisit', left and right arrows, and 'Show All Triples'. Below the table, there is a large empty area with a mouse cursor pointing to it.

Property	Value
Addressed To	Us284
Content	MRT @CRPDIsabilities Game #Accessibility Information Symbol - http://t.co/WW7BoIN #a11y
Language	En
Links to	Uri58421
Name	a11y accessibility
Type	Tweet

Ilustración 55.- Table View

D. Tutorial AGWebView

El AGWebView es una interfaz gráfica de usuario para explorar, consultar y gestionar triplestores de Allegrograph.

Al momento de instalar Allegrograph, AGWebView se instala por defecto, para acceder se ingresa a cualquier navegador a la siguiente dirección “localhost:10035”, se muestra una pantalla para loguearse, en username digitar “test” y en password “xyzyz”. Hay que tener siempre en cuenta que para que funcione el AGWebView el servidor de Allegrograph debe estar iniciado con el siguiente comando:

```
/home/pvqa/allegro/bin/agraph-control --config /home/pvqa/allegro/lib/agraph.cfg start
```

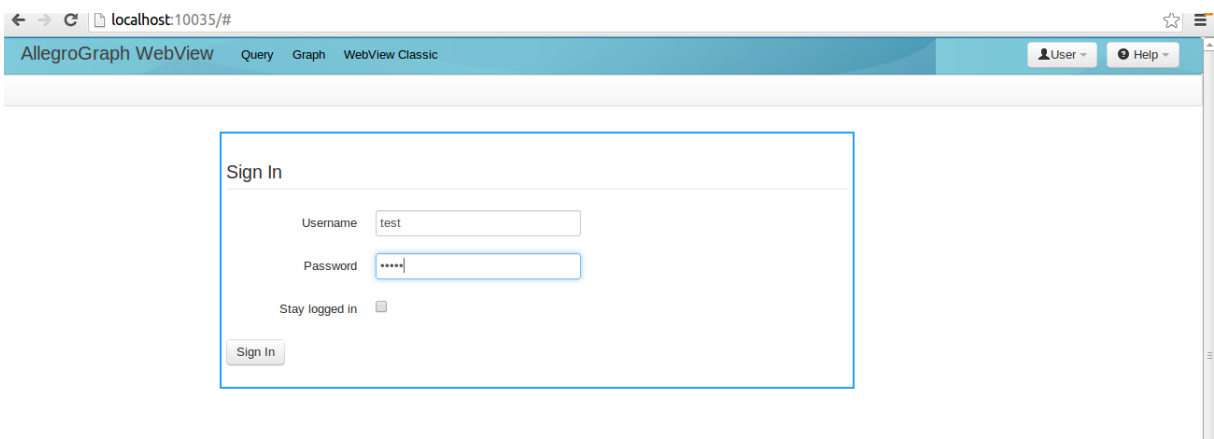


Ilustración 56.- Loguearse en AGWebView

Si ya se tiene creados repositorios, basta con hacer clic en el nombre del repositorio y se puede iniciar a gestionar las tripletas. En el caso de que no se tenga ningún repositorio, se procede a crear uno, en la opción “Create a Repository-Name” escribir un nombre para el nuevo repositorio y finalmente clic en “Create”.

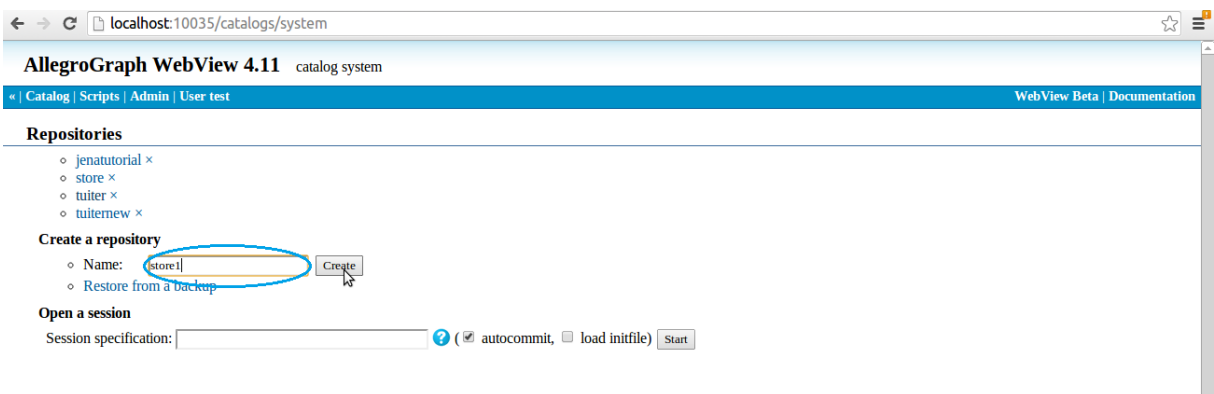


Ilustración 57.- Crear repositorio

Se ha creado un repositorio como ejemplo con el nombre de “store1” como se muestra en la Ilustración 58, el siguiente paso es cargar el archivo en el repositorio, haciendo clic en “from an uploaded file” como en la Ilustración 59, finalmente se presenta una pantalla (Ilustración 60) donde se selecciona el archivo y clic en “Ok”.

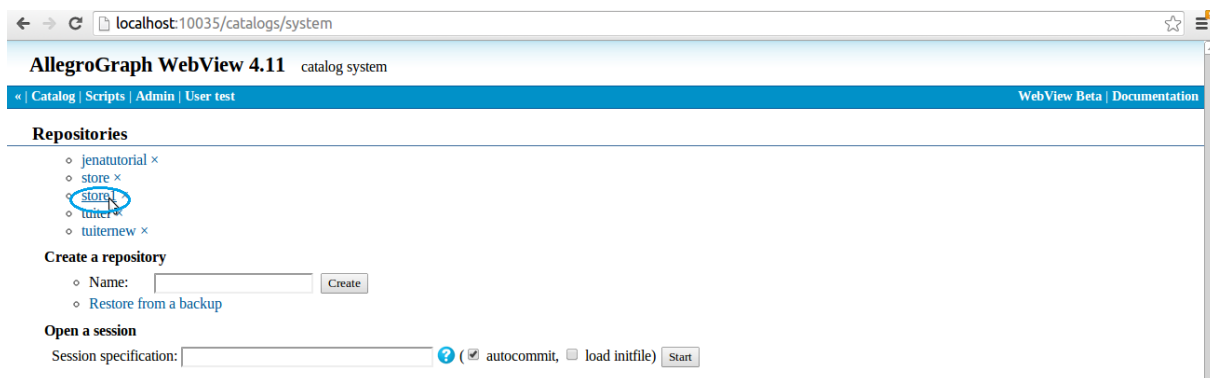


Ilustración 58.- Repositorio nuevo creado

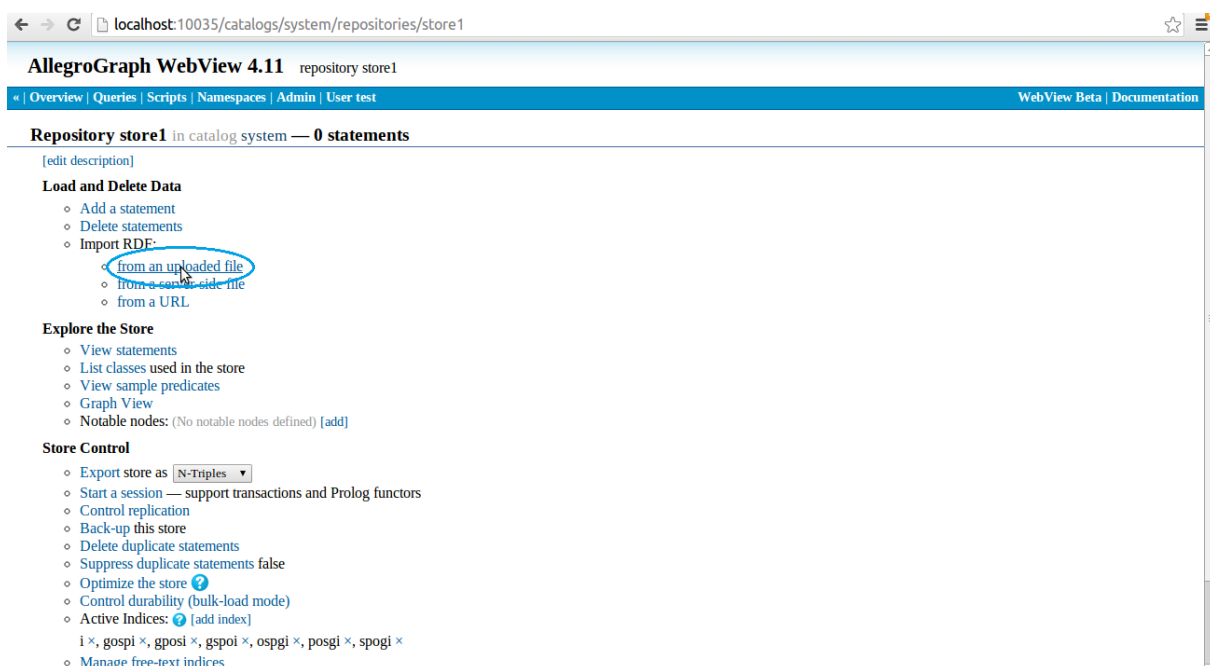


Ilustración 59.- Cargar RDF en AGWebView

En AGWebView se puede realizar consultas SPARQL y al ejecutar el resultado se muestra en columnas en la parte inferior de la pantalla, una opción interesante que posee esta herramienta es que al resultado hay como descargarlo en formato: SPARQL JSON, SPARQL XML, SPARQL TTL, SPARQL CSV, CSV, CSV (con sintaxis N-Triples).

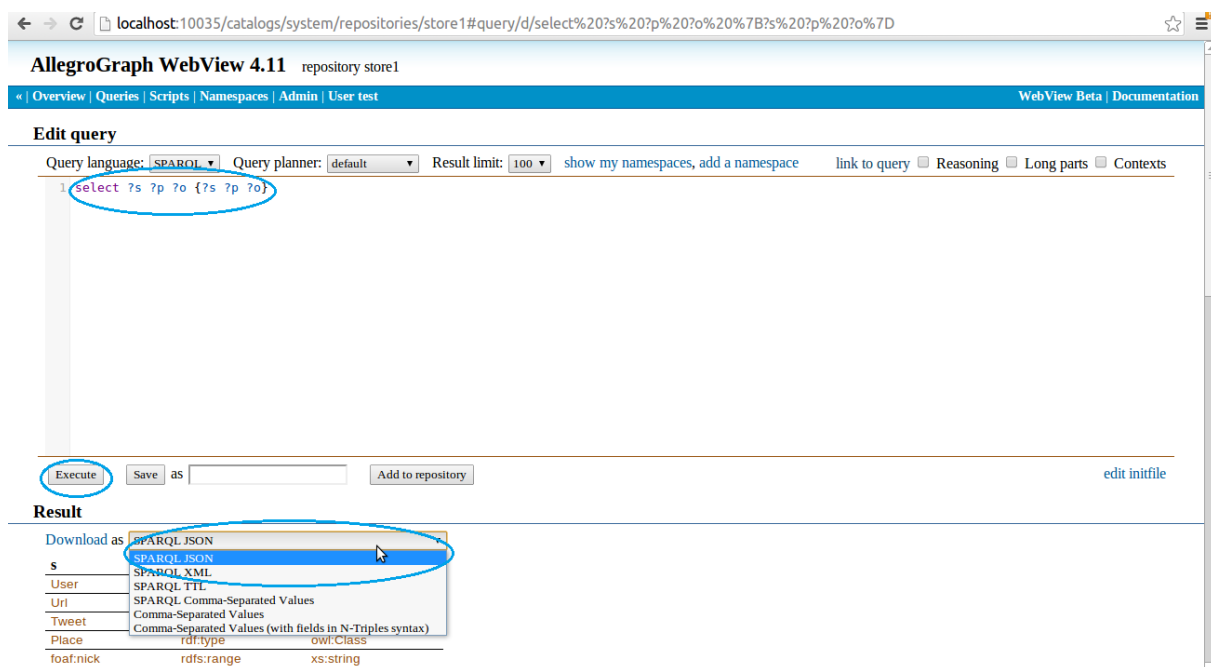


Ilustración 62.- Query SPARQL AGWebView

E. Fórmulas matemáticas de las métricas de centralidad

Un grafo con 7 nodos (A, B, C, D, E, F, G) se tomará como ejemplo para explicar el cálculo de las medidas cercanía, grado de centralidad e intermediación.

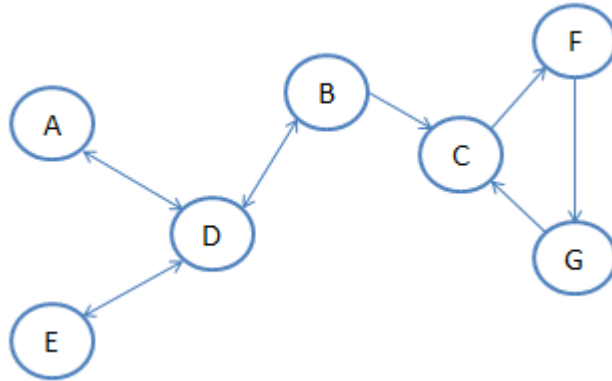


Ilustración 63.- Grafo de ejemplo

E.1. Cálculo del grado de centralidad

$$C_d(P_i) = \sum_{k=1}^N a(P_i, P_k)$$

$$C_d(P_i) = 1 + 1 = 2$$

Sumatoria de los enlaces de entrada y salida P_i, P_k

NODOS	IN-DEGREE	OUT-DEGREE	GRADO DE CENTRALIDAD
A	1	1	2
B	1	2	3
C	2	1	3
D	3	3	6
E	1	1	2
F	1	1	2
G	1	1	2

E.2. Cálculo del grado de cercanía

$$C_c(P_i) = \frac{N - 1}{\sum_{k=1}^N d(P_i, P_k)}$$

$$C_c(P_i) = \frac{7 - 1}{2} = 3$$

$d(P_i, P_k)$ es el número de enlaces geodésicos P_i y P_k . En esta expresión la suma es basada en las distancias de P_k al n-1 que es adyacente a todos los otros puntos. N es el total de nodos.

NODOS	IN- DEGREE	OUT- DEGREE	$d(P_i, P_k)$	CERCANÍA
A	1	1	2	3
B	1	2	3	2
C	2	1	3	2
D	3	3	6	1
E	1	1	2	3
F	1	1	2	3
G	1	1	2	3

* Cabe mencionar que en está métrica los valores de cercanía son bajos.

E.2. Cálculo del grado de intermediación

$$C_B(P_k) = \sum_{j=1}^N \sum_{i=1}^N \frac{g_{ij}(P_k)}{g_{ij}}$$

$$C_B(P_k) = \frac{0}{17} = 0$$

g_{ij} es el número de enlaces geodésicos entre P_i y P_j . El punto potencial es P_k para el control de información que pasa entre P_i y P_j entonces, $g_{ij}(P_k)$ es el número de enlaces geodésicos entre P_i y P_j que contienen a (P_k) . Para determinar la centralidad del punto (P_k) se suma sus valores de intermediación parciales para todos los pares de nodos.

En esta tabla se escriben los nodos que sirven de intermediarios entre un par de nodos.

NODOS	A	B	C	D	E	F	G
A		D	DB		D	DBC	DBCF
B	D				D	C	CF
C	BD						F
D			B			BC	CBF
E	D	D	DB			DBC	DBCF
F			G				
G						C	

En la siguiente tabla se muestra el número de veces que un nodo es intermediario entre un par de nodos.

NODOS	$g_{ij}(P_k)$
A	0
B	10
C	9
D	13
E	0
F	5
G	1

Tabla que muestra la suma de los caminos geodésicos entre P_i y P_j y el resultado final, es decir, el valor de intermediación de cada nodo.

NODOS	A	B	C	D	E	F	G	g_{ij}	$g_{ij}(P_k)$	Intermediación
A	0	2	3	1	2	4	5	17	0	0
B	2	0	1	1	2	2	3	11	10	0.909
C	0	0	0	0	0	1	2	3	9	3
D	1	1	2	0	1	3	4	12	13	1.083
E	2	2	3	1	0	4	5	17	0	0
F	0	0	2	0	0	0	1	3	5	1.666
G	0	0	1	0	0	2	0	3	1	0.333

F. Base de Datos

La base de datos con la que se trabaja se encuentra almacenada en MYSQL, se denomina “twitterdb” y cuenta con 10 tablas de las cuales sólo se trabaja con 4 debido a que los datos que se necesitan se encuentran en las mismas. En la tabla “twett” se encuentra la mayoría de información que contiene un tuit, la tabla “tag_t” contiene las etiquetas, la tabla “user_t” posee la información del usuario y finalmente la tabla “tagged que es una tabla intermedia. Como se puede observar en la Tabla 8 la cantidad de información con la que se trabaja en el desarrollo del proyecto es una parte de la base de datos principal. En la Ilustración 64 se puede apreciar el modelo relacional de la base de datos.

Tabla 8.- Cantidad de datos por tabla

Tabla	Cantidad
Twett	315252
tag_t	21590
user_t	148996

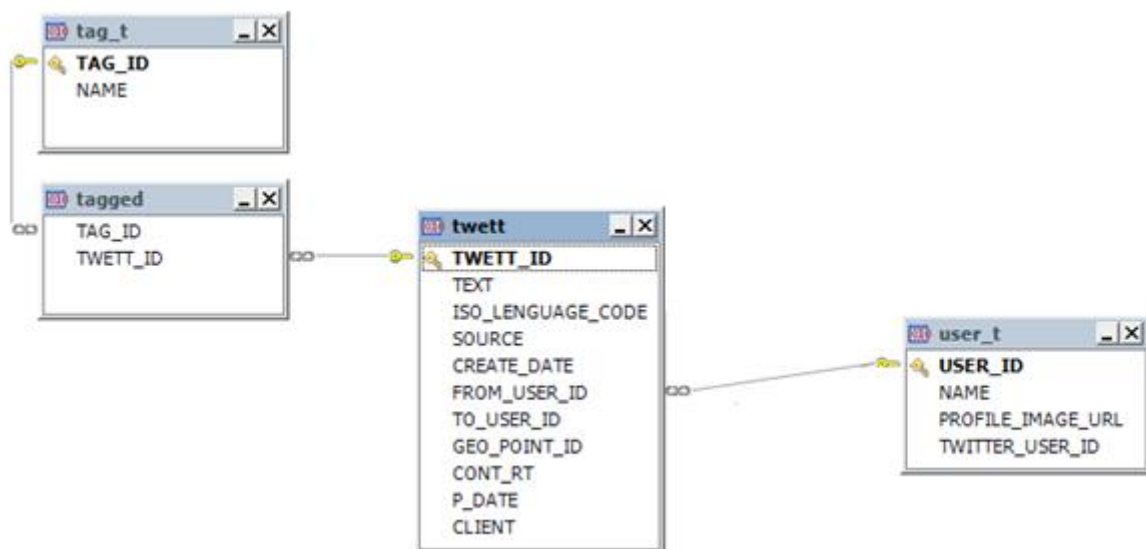


Ilustración 64.- Modelo Relacional de la Base de Datos

G. Ejemplo de un tuit según el vocabulario en formato N3 (Notación 3)

```
@prefix sioc: <http://rdfs.org/sioc/ns#>
@prefix foaf: <http://xmlns.com/foaf/0.1/>
@prefix moat: <http://moat-project.org/ns#>
@prefix tw: <http://data.twitter.com/ontology#>

tw:86949790 moat:name "accessibility"
tw:86949790 sioc:addressedTo tw:33392
tw:33392 foaf:name "ReducedMobility"
```