



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

DISEÑO Y DESARROLLO DE UNA HERRAMIENTA DE REPRESENTACIÓN Y
VISUALIZACIÓN DE REDES SOCIALES CON CAPACIDADES DISTRIBUIDAS

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

FELIPE ANIBAL RICARDO ESPINOZA CASTILLO

PROFESOR GUÍA:
CLAUDIO GUTIÉRREZ GALLARDO

MIEMBROS DE LA COMISIÓN:
GONZALO NAVARRO BADINO
LUIS MATEU BRULE

SANTIAGO DE CHILE
JULIO 2013

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE:
INGENIERO CIVIL EN COMPUTACIÓN
POR: FELIPE ANÍBAL RICARDO ESPINOZA CASTILLO
PROFESOR GUÍA: CLAUDIO GUTIÉRREZ GALLARDO
FECHA: 12/07/2013

**DISEÑO Y DESARROLLO DE UNA HERRAMIENTA DE
REPRESENTACIÓN Y VISUALIZACIÓN DE REDES SOCIALES CON
CAPACIDADES DISTRIBUIDAS**

A mi familia, a mi polola, a mis amigos y mi profe guía.

Agradecimientos

Gracias a mí por ser tan genial xD.

Índice general

Introducción	1
1. Descripción del Proyecto	4
1.1. Objetivos	4
1.1.1. Objetivo General	4
1.1.2. Objetivos Específicos	4
1.2. Resultados Esperados	5
1.3. Alcances	5
2. Marco Conceptual	6
2.1. Conceptos de Redes Sociales	6
2.1.1. Actor	6
2.1.2. Uniones Relacionales	7
2.1.3. Relaciones	7
2.1.4. Red Social	7
2.2. Conceptos de Desarrollo	8
2.2.1. Desarrollo Web	8
2.2.1.1. Desarrollo Front-End	9
2.2.1.2. Desarrollo Back-End	9
2.2.2. Modelo MVC	9
2.2.3. Frameworks de Desarrollo	10
2.2.3.1. Frameworks Server Side	10
2.2.3.2. Frameworks Client Side	10
3. Especificación del Problema	12
3.1. Relevancia del Problema	12
3.2. Usuarios Objetivo	12
4. Descripción de la Solución	13
4.1. SNM, Social Network Model	13
4.1.1. Elementos de Redes Sociales	13
4.1.2. Definición Matemática	14
4.1.3. Representación Gráfica	15
4.1.4. Representación Como Triples	16
4.2. Herramientas Elegidas	17
4.2.1. Ruby on Rails	17
4.2.2. EmberJS	18

4.2.3. D3.js	18
Conclusión	19

Índice de tablas

4.1. Representación en triples de la red social de la figura 4.2	17
--	----

Índice de figuras

2.1. Modelo MVC	9
4.1. Elementos Gráficos SNM	16
4.2. Ejemplo Red Social de Amistad en SNM	16

Introducción

Existe una red social llamada **LittleSis**[?] que consiste en una base de datos de relaciones de *quien conoce a quien* entre gente política, económica y socialmente poderosa en el mundo de las organizaciones en Estados Unidos cuyo fin es entregar el poder de la información al pueblo.

El origen del nombre de *LittleSis* se relaciona con el personaje *Big Brother* de la novela de George Orwell llamada *Nineteen Eighty-Four*. Este personaje consiste en un dictador que maneja toda la información sobre la población. *Big Brother* posteriormente fue un concepto con el cual se describió un ente poderoso.

De esta forma, al grupo compuesto por las personas poderosas y políticos de un país se les atribuye esta imagen de *Big Brother*, por su poder e influencia en un país. Esta es la motivación para bautizar con el nombre *LittleSis* (*Little Sister*), en oposición a *Big Brother*, a una aplicación cuyo fin es entregar poder de la información a toda la población.

La idea detrás de *LittleSis* tiene la capacidad de ser replicada en muchos países con el mismo fin, esto es, el de entregar poder al pueblo e informarle sobre los posibles conflictos de interés que las autoridades locales puedan tener al lidiar con el ejercicio de su poder, lo que en el caso particular de Chile, existe una iniciativa llamada **Poderopedia**[?] que busca ser una réplica de *LittleSis* para Chile.

En el caso de **Poderopedia**, uno de los líderes de ese proyecto es el ex alumno del DCC, Álvaro Graves, quienes tienen un **modelo RDF publicado en un repositorio en Github***, ellos están desde hace un tiempo cercanos a el lanzamiento de un beta público, de cualquier manera, por lo expresado en las redes sociales y en su página oficial, consiste en esta herramienta de las mismas características de LittleSis, es decir una base de datos centralizada sobre las relaciones del ambiente político chileno.

Con toda sus potencialidades, tanto *LittleSis* como *Poderopedia*, poseen un fin muy particular. En este sentido, sólo tienen la capacidad de cubrir la información sobre las personas muy importantes del mundo político, social y económico poderoso. Pero si una persona quisiera recolectar información sobre grupos de personas (por ejemplo el grupo de poder de una pequeña ciudad) para formar una red sociales, estas aplicaciones no sirven. Tampoco estas

herramientas permiten la creación de redes sociales privadas, que cumplan los objetivos que el usuario desee, pero al mismo tiempo permitirle interactuar con los datos de otras redes sociales pertenecientes a otras personas o comunidades.

El objetivo de esta memoria es llenar ese vacío, diseñando y desarrollando una herramienta para representar redes sociales, y que tengan capacidades distribuidas, donde los usuarios que poseen sus redes sociales públicas o privadas, puedan unirlos a voluntad.

Esta herramienta posee una gran gama de aplicaciones posibles, tanto para estudios de tipo sociológico, histórico, biológico, político, etc. Un caso de ejemplo reciente es el caso de las redes de autoridades e intereses posibles dentro del ambiente educacional chileno, lo que puede dar una visión más informada sobre el entramado de intereses detrás de los problemas denunciados por el conflicto estudiantil que se vive desde el año 2011.

Para lograr desarrollar esta herramienta se deben sortear una serie de desafíos técnicos, que aunque pareciera una aplicación bastante intuitiva, hasta el momento no hemos encontrado una aplicación con estas características.

Entre sus mayores desafíos está el de tener una aplicación de fácil instalación y uso amigable para los usuarios. Por otro lado, con lo que respecta al modelamiento de datos, debe basarse en algún estándar flexible de representación de redes sociales que permita la interoperabilidad. Esta memoria partirá de la base y la experiencia del recientemente titulado doctor en computación Mauro San Martín, cuya tesis de doctorado consistió en diseñar un modelo para el manejo de redes sociales, de esta forma, se aprovechará el conocimiento de su experiencia de investigación, aplicándolo a un trabajo práctico y útil para la investigación y estudio de varias disciplinas.

También habrá que desarrollar el aspecto de la visualización de datos, la integración con otras herramientas enfocadas al análisis de redes sociales, el modelamiento y uso de la información de redes sociales.

Dado que la estructura de redes sociales está fuertemente ligada a grafos, y se necesita un estándar aceptado (para el que existan herramientas de desarrollo disponibles), se ha optado por representarlo en el modelo **RDF**. Este estándar es además es una representación que permitirá todo tipo de usos incorporándose a tecnologías de la **Web Semántica**. Finalmente, además de los anteriores, hay desafíos de negocios que se vinculan con lo técnico, como hacer algunos casos de prueba de usabilidad para entregar una herramienta de valor a usuarios y contactar diversas personas que puedan estar interesadas en la herramienta.

En este informe de avance en el tema de memoria se expondrá los avances hechos en el semestre. En la sección se presenta el objetivo general del trabajo, de manera que el lector

comprenda resumida y claramente la visión y alcance de este trabajo. Luego en la sección de objetivos específicos se descompone el objetivo general en una serie de objetivos medibles incluyendo además objetivos tentativos con respecto al tiempo de lo que se pudiera extender este trabajo. Los avances principales en la memoria de expresan en la sección donde se expone la especificación del sistema. Luego, se discute el enfoque mismo de trabajo de esta memoria en la sección de plan de trabajo, en donde se dará a conocer la metodología general para desarrollar los objetivos, definir los entregables finales del trabajo y haciendo una estimación de tiempos por medio de una carta gantt.

Capítulo 1

Descripción del Proyecto

1.1. Objetivos

A continuación se presentan los objetivos para este trabajo de memoria.

1.1.1. Objetivo General

El objetivo de este trabajo consiste en crear una herramienta por la cual personas que estudian diversos tipos de redes sociales como: sociólogos, periodistas, biólogos, etc; puedan representar, administrar y visualizar redes sociales de mediana escala, contando además con la capacidad de combinar redes sociales con otros usuarios de la herramienta para obtener redes sociales con información más completa ofreciendo esto como un servicio centralizado con la capacidad de ser distribuido

1.1.2. Objetivos Específicos

De lo escrito anteriormente en el objetivo general, se desprenden los siguientes objetivos intermedios:

1. Implementar un modelo de representación de redes sociales en RDF.
2. Hacer una interfaz amigable de ingreso de datos: actores y relaciones.
3. Desarrollar un sistema centralizado que administre la asignación de identificadores únicos a los actores que los usuarios agregan a sus redes sociales.
4. Unir redes sociales, combinando la información y relaciones de actores en común.
5. Visualización
 - (a) De la estructura general de las redes sociales creadas.
 - (b) Específica de elementos de interés en esas redes sociales.

6. Representar algunas redes con el modelo implementado anteriormente, que varíen en tamaño y en complejidad.

1.2. Resultados Esperados

Para llevar a cabo los objetivos expuestos en esta memoria, los resultados que se esperan consisten en: un sistema que permita el modelamiento de redes sociales en donde debe existir una aplicación que permita de manera cómoda crear estas redes sociales, complementar su información para posteriormente unir las redes de un usuario con otro en caso de que este lo quiera. Además de lo anterior, los datos generados por el sistema deben estar en un formato amigable para computadores, que faciliten la posterior interoperabilidad con otras aplicaciones y fuentes de conocimiento.

1.3. Alcances

Es importante mencionar, que con el objetivo de que el trabajo a realizar cumpla con las limitaciones de tiempo correspondientes a una memoria de ingeniería, el problema de la creación y manipulación de redes sociales por personas se acota en los siguientes aspectos:

- El tamaño de redes sociales, se considerará pequeño con un número de alrededor 100 actores por red social, debido a que las redes serán principalmente creadas de forma manual, por tanto debe manejar un número que sea posible de alcanzar por los usuarios de la aplicación, de esta forma, ahorrando problemáticas asociadas con la escalabilidad de la aplicación para redes sociales más grandes.
- Dependiendo de las aplicaciones para las cuales se use el modelamiento de redes sociales, puede requerirse ver los cambios temporales que sufren estas, para analizar la estructura de una red social y su evolución en el tiempo. Dicho problema no será abordado en esta memoria.

Capítulo 2

Marco Conceptual

En este capítulo se describen los conceptos necesarios para que el lector se pueda familiarizar con mayor facilidad con los conceptos usados en el resto del informe, describiendo lo que se entiende por los conceptos usados en términos de redes sociales y conceptos asociados al desarrollo de la solución.

En caso de así preferirlo, el lector puede saltar las secciones que estime convenientes dentro de este capítulo, en caso de dominar los conceptos y usar este capítulo como referencia en caso de necesitarlo.

2.1. Conceptos de Redes Sociales

A continuación se expondrán los conceptos fundamentales que se usarán en la memoria en relación a redes sociales. Estos conceptos fueron extraídos desde la tesis de doctorado del alumno del DCC, Mauro San Martín[14], debido a que representan lo necesario para el entendimiento de este trabajo, por lo tanto su reusó e inclusión en el informe para que el lector no necesite revisar la memoria de Mauro para poder ver estos conceptos.

Para construir la definición formal de redes sociales, es necesario definir algunos conceptos claves previamente, siguiendo las definiciones clásicas de este dominio[15].

2.1.1. Actor

Un actor es una entidad social, la cual está bajo estudio junto con sus interacciones sociales. En estricto rigor, los actores pueden ser definidos como individuos, corporaciones o unidades sociales colectivas. Ejemplos de actores son gente en un grupo, departamentos en una empresa o agencias de servicio público en una ciudad. El uso del término actor no significa que estas entidades necesariamente tienen la habilidad de actuar. Más allá, la mayoría de las

aplicaciones en redes sociales se enfocan en colecciones de actores que son de un mismo tipo (por ejemplo, gente en un grupo de trabajo). A veces, sin embargo, la investigación necesita mirar a actores de diversos niveles o tipos conceptuales, o desde diversos conjuntos. Los datos pueden incluir atributos no relacionales asociados a los diferentes actores.

2.1.2. Uniones Relacionales

Los actores están conectados hacia otros por uniones sociales. El rango y tipo de estas uniones puede ser muy grande. La característica principal de una unión es que establece una conexión entre un par de actores. Algunos de los ejemplos más comunes de uniones empleadas en el análisis de redes sociales son:

- La evaluación de una persona por otra, ej: amistad declarada, gusto o respeto.
- Transferencia de recursos materiales, ej: transacciones de negocios, prestar o pedir prestado.
- Asociación o afiliación, ej: atender conjuntamente a un evento social, o pertenecer al mismo club social.

2.1.3. Relaciones

El conjunto de uniones entre un tipo específico de miembros de un grupo se llama una relación. Por ejemplo, el conjunto de las amistades entre pares de niños en un salón de clases, o el conjunto de uniones diplomáticas entre pares de naciones en el mundo, son uniones que definen relaciones. Para cualquier grupo de actores, podemos encontrar diversas relaciones; por ejemplo, además de las relaciones diplomáticas entre países, podemos encontrar la existencia de comercio en un determinado año. Las relaciones (o uniones específicas) pueden tener atributos que las describen. Por ejemplo en el caso del comercio, su cantidad de transacciones total puede haber sido guardado.

Con lo expuesto anteriormente, finalmente podemos definir una red social.

2.1.4. Red Social

Una red social consiste en uno o muchos conjuntos finitos de actores, junto con las relaciones definidas entre ellos. La presencia de información relacional es una característica crítica de las redes sociales. Una red social es un caso particular de red, de esta manera su estructura puede ser formalizada como un grafo.

En adición al uso de conceptos relacionales, Wasserman y Faust[15] se deben tener en cuenta las siguientes consideraciones:

- Actores y sus acciones son vistos como interdependientes, más que independientes, unidades autónomas.

- Uniones relacionales (conexiones) entre actores son canales de transferencia o "flujo" de recursos (material o no material).
- Modelos de red enfocados en individuos muestran el ambiente estructural de la red, además de proveer la capacidad de definir limitaciones en nivel individual.
- Los modelos de red conceptualizan estructura (social, económica, política, entre otras) como patrones generales de relaciones entre actores.

2.2. Conceptos de Desarrollo

A continuación se explican los términos en lo que a desarrollo que se refiere que tienen relación con el trabajo realizado en esta memoria.

2.2.1. Desarrollo Web

El desarrollo web se denomina al desarrollo de software cuyo fin es la creación de aplicaciones que se ejecuten en computadores que puedan ser accedidos vía la *world-wide web*. Este estilo de desarrollo conlleva a que una aplicación que pueda estar ejecutándose en un servidor pueda entregar resultados a clientes provenientes de cualquier parte del mundo, usando una gran diversidad de dispositivos, plataformas de software, etc. La cual es frecuentemente accesada por las personas vía un navegador web.

Esta propiedad de multiplataforma de las aplicaciones desarrolladas para la web, junto con propiedades de accesibilidad desde diversos medios a las aplicaciones, ha hecho que el desarrollo web sea un enfoque de desarrollo ampliamente usado en la industria en los últimos 10 años.

En términos generales el desarrollo web posee una arquitectura clásica de cliente/servidor, en donde una aplicación corriendo en un servidor proporciona resultados y ejecuta instrucciones proveniente de clientes, que pueden ir desde un navegador web de p.c. o dispositivo móvil, o servir de API para otras aplicaciones. A continuación se adjunta un diagrama que muestra los principales componentes cuando se habla de desarrollo web.

En la imagen se ve un ejemplo simple de una aplicación que posee un servidor de ejecución de la aplicación, con un servidor de base de datos, que son accedidos vía teléfono, tablet o computador por medio de un navegador web o de una interfaz API que use una aplicación nativa.

Dentro del desarrollo web, también se encuentran subcategorías que son expuestas a continuación.

2.2.1.1. Desarrollo Front-End

Es el desarrollo orientado a todos los componentes con los cuales interactúa un usuario de la aplicación, enfocado en la interfaz y sus componentes gráficos, además de la experiencia usuario, la usabilidad de la aplicación, etc. Es el desarrollo del cliente que el usuario usa para acceder al core de la aplicación.

En términos de tecnologías, se asocia el desarrollo front-end al uso de tecnologías como: HTML, CSS, Javascript, entre otras.

2.2.1.2. Desarrollo Back-End

Es el desarrollo de la aplicación que almacena y posee la lógica común a todos los clientes de la aplicación, en donde el código de este tipo de desarrollo es que se ejecuta en el servidor. Además generalmente en el desarrollo back-end se incluye todo lo relacionado a persistencia de datos generados a través del uso de la aplicación.

2.2.2. Modelo MVC

El modelo MVC (Modelo, Vista, Controlador)[8], consiste en un modelo para separar la lógica de una aplicación agrupándola en clases u otras unidades modulares, de acuerdo con la responsabilidad que estos módulos cumplan dentro de un sistema. A modo de ejemplo, a fin de ilustrar este concepto, se puede dar un caso de una aplicación que registre compras en un sistema de tienda online, un ejemplo de los módulos asociados a compras en MVC puede ser el siguiente:

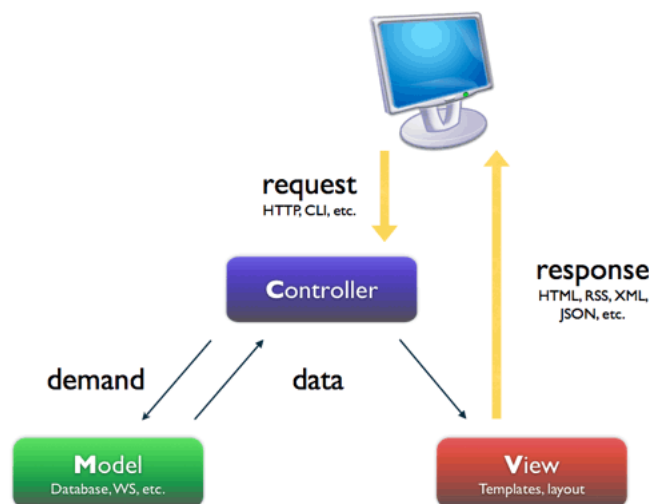


Figura 2.1: Modelo MVC

- **Modelo:** el modelo corresponde a una clase **Compra** que contiene toda la lógica de negocio asociada a las compras, que además se asocia directamente a como se almacena una compra en la base de datos.
- **Vista:** un ejemplo de vista para una compra, puede ser una interfaz **HTML** en la cual el cliente efectúe operaciones sobre la compra, por ejemplo agregar productos, cabe destacar que la vista no ejecuta las acciones, sólo se encarga de recibirlas y enviarlas al último componente del modelo MVC, el controlador.
- **Controlador:** de acuerdo a lo recién expresado, el controlador es el encargado de coordinar uno o más modelos para ejecutar las acciones capturadas en la vista. En el caso de la compra, el controlador es quien efectivamente procesaría el pago de la misma.

2.2.3. Frameworks de Desarrollo

Un framework de desarrollo define un marco en el cual desarrollar una aplicación, el cual provee una gran cantidad de funcionalidad común lista de manera de no desarrollar un proyecto desde cero, pero además de sólo funcionalidad, provee de una estructura lógica con la cual se escribe el código, que está influida por muchas otras personas que han usado el framework en aplicaciones reales.

2.2.3.1. Frameworks Server Side

Dentro de los frameworks de desarrollo web existe una categoría llamada *Sever Side*, la cual consiste en que el código de la aplicación corre desde un servidor en internet, lo que permite que si la computación es común para muchos clientes, los resultados de esa computación pueden ser usados múltiples veces.

Ejemplos actuales de esta categoría de frameworks son: **Ruby on Rails**[9], **Sinatra**[10] que utilizan el lenguaje **Ruby**; **Django**[5], **Pylons**[7] para **Python**; **Spring**[11] para **Java** y **CakePHP**[3] para **PHP** entre otros.

2.2.3.2. Frameworks Client Side

En el último tiempo, surgió una nueva categoría de frameworks de desarrollo web llamada *Client Side*, en la cual el código de la aplicación se ejecuta en el computador del usuario de la aplicación, ahorrando recursos necesarios en un servidor, además de ahorrar el tiempo de latencia entre el cómputo de una respuesta y su transmisión al equipo del usuario.

Comúnmente estos frameworks son escritos para ser usados con el lenguaje **Javascript**, pues posee la propiedad que todos los navegadores web implementan un motor de **Javascript** y por lo tanto el usuario no necesita instalar nada más. Ejemplos de frameworks client side son:

AngularJS[1], EmberJS[13], Meteor[6] y alternativamente BackboneJS[2] que es una librería más que un framework.

Capítulo 3

Especificación del Problema

3.1. Relevancia del Problema

El problema es relevante de resolver, debido a que se está entregando una herramienta que le permite a personas que no tengan conocimientos avanzados de computación, almacenar redes sociales en forma gráfica interactivamente.

Además estas redes sociales anteriormente creadas, se pueden complementar uniéndolas con otras redes sociales creadas por otros usuarios de la aplicación, aprovechando el conocimiento de más personas sobre estructuras sociales determinadas.

Otro aspecto clave de resolver este problema con este tipo de aplicación corresponde a que los datos generados por sus usuarios son compatibles con los formatos y tecnologías pertenecientes a la web semántica, por lo cual esta información puede ser relacionada con otras fuentes de conocimiento posteriormente.

3.2. Usuarios Objetivo

La herramienta que se desarrollará para esta memoria está enfocada principalmente en personas en cuya profesión requiera el estudio de diversos tipos de redes sociales, las cuales pueden encontrarse en campos como: sociología, biología, historia, negocios etc. Es decir personas con un conocimiento básico de redes sociales y el estudio de estas, aplicadas a algún campo en particular.

Capítulo 4

Descripción de la Solución

A continuación se describe la solución propuesta con motivo de este trabajo de memoria, partiendo de conceptos más teóricos para luego abordar algunos temas de decisiones de implementación y mostrar el resultado final de la aplicación que se desarrolló.

4.1. SNM, Social Network Model

Uno de los puntos principales de esta memoria, es poner en práctica el modelo realizado por el alumno de doctorado en el DCC, Mauro San Martín[14], quien investigó y propuso un modelo llamado *Social Network Model* o SNM. Por lo tanto, la generación de redes sociales en esta aplicación tendrán en cuenta este modelo, trabajo que fue guiado por el mismo profesor guía de esta memoria, aprovechando la experiencia e investigación previa sobre el tema. Entonces, se procederá a definir y explicar brevemente en que consiste este modelo.

4.1.1. Elementos de Redes Sociales

Para el modelo de Mauro se tienen algunos elementos pertenecientes a las redes sociales, que fueron tratados en los conceptos de redes sociales en la sección 2.1, a los cuales se les agregan algunas restricciones:

- Los *Actores*: estos tienen un identificador único y un conjunto de atributos, además pueden participar en cualquier número de relaciones.
- Las *Relaciones* también tiene un identificador único, un conjunto de atributos y un número de actores participantes. El número de participantes puede ser uno o más, y puede cambiar sin afectar el resto de las propiedades de la relación.
- Los *Atributos* tienen un significado asociado y un valor literal. Un atributo es identificado por el identificador del objeto al cual está añadido (acto o relación), por su significado y su valor literal. La clase de un objeto (actor o relación) es un tipo de atributo especial llamado *familia*.

- *Actores, Relaciones, Atributos* y sus conexiones forman una red social. Compartiendo y reusando metadata al nivel, por ejemplo, proveniencia de conjuntos de datos.

4.1.2. Definición Matemática

Dado lo anterior, el modelo *SNM* describe una red social como un grafo de la siguiente forma:

Definición 4.1 (*Red Social Generalizada*) Una red social generalizada es definida como un multigrafo dirigido tripartito con etiquetas junto con una familia de funciones etiquetadoras f y un conjunto de familia de etiquetas L_f :

$$G = (N, E, L_N, L_E, L_f, \iota, \nu, \varepsilon, f)$$

Donde:

- El conjunto de nodos $N = A \cup T \cup C$ es una unión disjunta del conjunto de actores A , con el conjunto de relaciones T y el conjunto de atributos C .
- Existe una colección finita de familias (subconjuntos) de actores $\mathcal{A} = \{A_1, A_2, \dots, A_k\}$ de manera tal que cada $A_i \subseteq A$ y $\cup_{1 \leq i \leq k} A_i = A$.
- Donde existe una colección finita de familias (subconjuntos) de relaciones $\mathcal{T} = \{T_1, T_2, \dots, T_j\}$ donde cada $T_i \subseteq T$ y $\cup_{1 \leq i \leq k} T_i = T$.
- El conjunto de arcos $E = E_{AT} \cup E_{AC} \cup E_{TC}$ es la unión disjunta del conjunto de arcos entre actores y relaciones E_{AT} , con el conjunto de arcos entre actores y atributos E_{AC} , el conjunto de arcos entre atributos y relaciones E_{TC} .
- El conjunto de etiquetas de nodo $L_N = L_A \cup L_T \cup L_C$ es la unión disjunta de los conjuntos de etiquetas de actores L_A , de etiquetas de relaciones L_T y el de etiquetas de atributos L_C .
- El conjunto de etiquetas de arcos $L_E = L_{AT} \cup L_{AC} \cup L_{TC}$ es la unión disjunta de los conjuntos de etiquetas de arcos entre actores y relaciones L_{AT} (roles de participación), de etiquetas de arcos entre actores y atributos L_{AC} (significado de atributos de actores) y el de etiquetas de arcos entre relaciones y atributos L_{TC} (significado de atributos de relaciones).
- El conjunto de etiquetas de familias $L_f = L_{f_A} \cup L_{f_T}$ es la unión disjunta entre el conjunto de etiquetas de familias de actores L_{f_A} y el conjunto de etiquetas de familias de relaciones L_{f_T} .
- $\iota = \{\iota_{AT}, \iota_{AC}, \iota_{TC}\}$ es el conjunto de funciones de incidencia tales que $\iota_{AT} : E_{AT} \rightarrow A \times T$ es una función de incidencia que asocia cada arco de participación a un actor y su relación; $\iota_{AC} : E_{AC} \rightarrow A \times C$ es una función de incidencia que asocia un arco de significado a un actor y a un atributo; $\iota_{TC} : E_{TC} \rightarrow T \times C$ es una función de incidencia que asocia cada arco de significado a una relación y un atributo.
- $\nu = \{\nu_A, \nu_T, \nu_C\}$ es un conjunto de funciones etiquetadoras de nodos tales que $\nu_A : A \rightarrow L_A$ es una función biyectiva desde actores a las etiquetas de actores; $\nu_T : T \rightarrow L_T$ es una función biyectiva desde relaciones a las etiquetas de las relaciones; $\nu_C : C \rightarrow L_C$ es una función biyectiva desde atributos a las etiquetas de atributos.

- $\varepsilon = \{\varepsilon_{AT}, \varepsilon_{AC}, \varepsilon_{TC}\}$ es el conjunto de funciones etiquetadoras de arcos tales que $\varepsilon_{AT} : E_{AT} \rightarrow L_{AT}$ es una función desde arcos de participación a sus etiquetas; $\varepsilon_{AC} : E_{AC} \rightarrow L_{AC}$ y $\varepsilon_{TC} : E_{TC} \rightarrow L_{TC}$ son funciones desde arcos de significado a sus etiquetas.
- $f = \{f_A, f_T\}$ es el conjunto de funciones etiquetadoras de familias tal que $f_A : \mathcal{A} \rightarrow L_{f_A}$ es una función de familias de actores a las etiquetas de familias de actores y $f_T = \mathcal{T} \rightarrow L_{f_T}$ es una función de familias de relaciones a las etiquetas de familias de relaciones.
- La siguiente condición se mantiene para todos los arcos entre el mismo par de actores y relaciones, Para todo e_1 y e_2 de manera tal que $\iota(e_1) = \iota(e_2) = (u, v)$ con $u \in A$ y $v \in T$, $e_1, e_2 \in E \Leftrightarrow \varepsilon(e_1) \neq \varepsilon(e_2)$.
- Cada función de etiquetado en ν, ε y f , excepto ν_C , deben ser invertibles.
- Para una relación $r \in T$ entre dos actores $a_1, a_2 \in A$, tal que existe $e_1, e_2 \in E$, y $\iota(e_1) = (a_1, r), \iota(e_2) = (a_2, r)$ con etiquetas $\varepsilon(e_1) = p_1, \varepsilon(e_2) = p_2$. La dirección de r puede ser especificada por el par ordenado de las etiquetas de participación, eso es que una dirección (p_1, p_2) indica que r comienza en a_1 y termina en a_2 , la dirección opuesta es representada por (p_2, p_1) .

De lo anterior se extrae información relevante sobre las características de las redes sociales y sus elementos:

- Un **Nodo** puede ser un *Actor*, *Relación* o *Atributo*.
- Las relaciones pueden ser de uno a múltiples actores.
- Los actores juegan un **Rol** en la relación.
- Los *Actores* y *Relaciones* pertenecen a **Familias de Actores** y **Familias de Relaciones** respectivamente.
- Una **Familia** de relación o actor, define un conjunto de actores/relaciones en común.

4.1.3. Representación Gráfica

En su tesis de doctorado[14], Mauro define un lenguaje gráfico para representar redes sociales con su modelo, el cual se adjunta a continuación por motivos de completitud del trabajo y expresar como este modelo influye posteriormente el diseño de la interfaz del usuario.

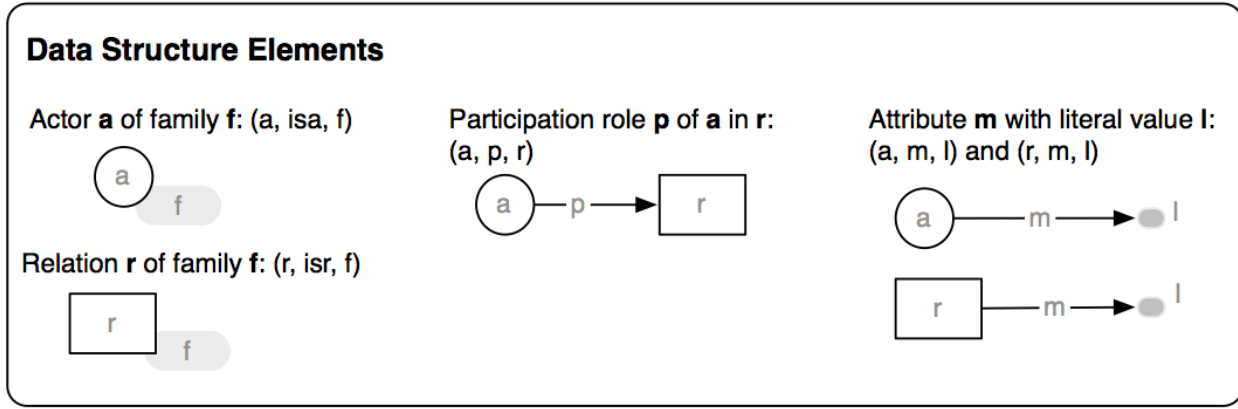


Figura 4.1: *Elementos Gráficos SNM*. Desde izquierda a derecha, los 4 bloques de construcción gráfica de una red social: un actor (arriba) y su etiqueta de familia, una relación y su etiqueta de familia, un rol de participación de un actor en una relación y atributos sobre actores y relaciones. Además por cada bloque se muestra la equivalencia en forma de triple.

Dados estos elementos gráficos podemos modelar una red de ejemplo como:

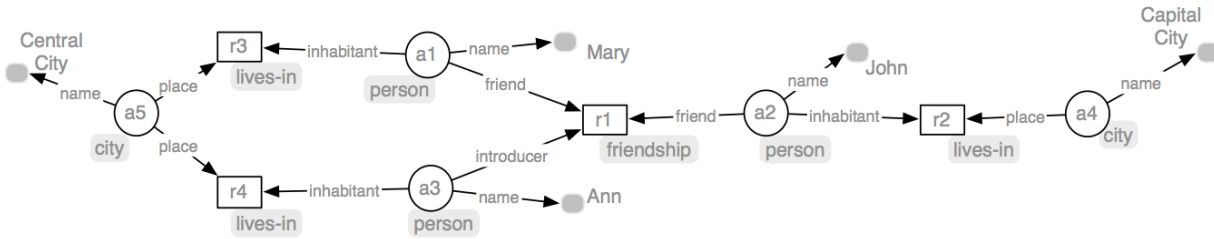


Figura 4.2: *Ejemplo Red Social de Amistad en SNM*. Una red social representando relaciones de amistad (nodos cuadrados) entre Mary y John, quienes fueron presentados por Ann (los actores se muestran como nodos circulares y sus atributos como puntos grises). Las ciudades de residencia también son representadas como actores.

4.1.4. Representación Como Triples

El modelo SNM, al ser en una representación de grafo, es posible representarlo como triples y vice-versa (la demostración se encuentra en la memoria de Mauro[14]). Lo cual es particularmente útil para usar este modelo dentro de un contexto de bases de datos relacionales, razón por la cual veremos una de estas transformaciones a continuación.

Es posible traducir una red social dada, representada como tres conjuntos de triples (N, R, M) , en donde N es el conjunto de triples de *Tipos*, R el de *Roles* y M de *Atributos*; en una red social generalizada G con el siguiente algoritmo que produce un nodo o arco por cada triple:

- Para cada triple $(\nu_A(a), isa, f_A(A_i))$ en N , se añade un nodo actor a en A_i .
- Para cada triple $(\nu_T(t), isr, f_T(T_i))$ en N , se añade un nodo de relación t a T_i .
- Para cada triple $(\nu_A(u), \varepsilon_{AT}(e), \nu_T(v))$ en R , se añade un arco de participación e a E_{AT} con nodos finales $u \in A$ y $v \in T$.
- Para cada triple $(\nu_A(u), \varepsilon_{AC}(e), \nu_C(v))$ en M , se añade un nodo de atributo v a C , y su arco de significado e a E_{AC} con nodos finales $u \in A$ y $v \in C$.
- Para cada triple $(\nu_T(u), \varepsilon_{TC}(e), \nu_C(v))$ en M , se añade un nodo de atributo v a C , y su arco de significado e a E_{TC} con nodos finales $u \in T$ y $v \in C$.

Por ejemplo se muestra la siguiente tabla que representa este conjunto de triples, que es similar a lo que se tendría en una base de datos relacional para el caso de la red de amistades de la figura 4.2.

N: Typing			R: Roles			M: Attributes		
a1	isa	'person'	a1	friend	r1	a1	name	'Mary'
a2	isa	'person'	a2	friend	r1	a2	name	'John'
a3	isa	'person'	a3	introducer	r1	a3	name	'Ann'
a4	isa	'city'	a2	inhabitant	r2	a4	name	'Capital City'
a5	isa	'city'	a4	place	r2	a5	name	'Central City'
r1	isr	'friendship'	a1	inhabitant	r3			
r2	isr	'lives-in'	a5	place	r3			
r3	isr	'lives-in'	a3	inhabitant	r4			
r4	isr	'lives-in'	a5	place	r4			

Tabla 4.1: Representación en triples de la red social de la figura 4.2

4.2. Herramientas Elegidas

A continuación se presentan las herramientas elegidas para el desarrollo de la aplicación y los fundamentos de estas elecciones:

4.2.1. Ruby on Rails

Para el desarrollo del backend de la aplicación se eligió el framework de desarrollo web Ruby on Rails[9], comúnmente conocido simplemente como Rails, en el cual se desarrolla sobre el lenguaje Ruby. Es un framework maduro con 10 años de existencia, el cual es una de las plataformas más populares para este tipo de desarrollo actualmente en Silicon Valley.

Dentro de las características destacables de rails se pueden destacar: sus principios de privilegiar las convenciones sobre las configuraciones, es decir, el framework entrega mucha funcionalidad hecha mientras se cumplan sus convenciones, las cuales de ser necesario se

pueden sobre escribir; aplicaciones que pueden ser ejecutadas en diversos ambientes independientemente configurados como producción, testing o desarrollo.

Además de aspectos técnicos, dentro de las motivaciones al elegir rails, se encuentra la presencia de una gran comunidad de desarrolladores que crean muchas librerías, o en terminología ruby, 'gemas' las cuales hacen el desarrollo mucho más rápido reusando estas librerías que resuelven múltiples problemas frecuentes. Junto con lo anterior el alumno memorista posee vasta experiencia laboral en esta tecnología.

4.2.2. EmberJS

EmberJS es un framework de desarrollo web javascript en el lado del cliente, fue creado por Yehuda Katz y Tom Dale cuando ellos hacían la segunda versión del framework web Sproutcore.

EmberJS se destaca debido a que se pueden escribir aplicaciones completas que son ejecutadas en el lado del cliente, con clases especializadas como *modelos*, *vistas*, *controladores*, *templates* en una versión un poco distinta del modelo MVC. Una característica principal de emberjs es la actualización automática de templates cuando la información el sistema cambia; la presencia de observadores y la capacidad de adaptar emberjs a diversos modelos de persistencia como vía API JSON o el almacenamiento local integrado en los navegadores web de última generación.

Para el desarrollo de la aplicación de esta memoria, debido a que se trata de una aplicación que permite editar grafos interactivamente y es vía web, es necesaria mucha integración entre el javascript de las vistas y los datos de los nodos de las redes sociales, razón por la cual fue un mejor enfoque desarrollar la aplicación con emberJS, que resultó en pruebas iniciales mucho mejor que la otra alternativa evaluada, BackboneJS[2].

Uno de los aspectos útiles de EmberJS es que uno de sus creadores, Yehuda Katz, es integrante de los equipos principales de desarrollo de Ruby on Rails y la librería de javascript jQuery. De esta forma, EmberJS está pensado para tener una muy buena integración con Rails y jQuery, potenciando su atractivo como herramienta de desarrollo en el lado del cliente.

4.2.3. D3.js

Dentro de las herramientas principales se incluye D3.js[4], que es una librería javascript para el despliegue y manipulación de datos como información visual vía SVG en la web. Además de lo anterior D3.js es una de las opciones más utilizadas en lo que a gráficos SVG en la web se refiere.

El funcionamiento básico de D3 es el siguiente: se parte por asociar un arreglo de elementos que contienen los datos con los que trabajamos y son asociados con elementos SVG, como círculos por ejemplo y para el caso del arreglo de datos, D3 tiene 3 estados: el de entrada, es decir, cuando un elemento nuevo es agregado al arreglo, D3 se encarga de crear un nuevo elemento SVG para asociarlo a este nuevo dato; el estado de actualización es el cual que a todos los datos disponibles actualiza su posición u otras propiedades y finalmente el estado de salida es cuando un elemento del arreglo es eliminado y por consiguiente D3.js se encarga de remover su elemento SVG respectivo.

Conclusión

Conclusión content

Bibliografía

- [1] Angularjs home page. <http://angularjs.org/>, June 2013.
- [2] Backbonejs home page. <http://backbonejs.org/>, June 2013.
- [3] Cakephp home page. <http://cakephp.org/>, June 2013.
- [4] D3.js, data driven documents. <http://d3js.org/>, June 2013.
- [5] Django home page. <https://www.djangoproject.com/>, June 2013.
- [6] Meteorjs home page. <http://meteor.com/>, June 2013.
- [7] Pylons home page. <http://www.pylonsproject.org/>, June 2013.
- [8] Rails guides, the mvc architecture. http://guides.rubyonrails.org/getting_started.html#the-mvc-architecture, June 2013.
- [9] Ruby on rails, home page. <http://rubyonrails.org/>, June 2013.
- [10] Sinatra home page. <http://www.sinatrarb.com/>, June 2013.
- [11] Spring home page. <http://www.springsource.org/>, June 2013.
- [12] Manuel Bahamonde. Aplicación para la creación y administración de redes sociales, 2009. Memoria para optar al título de ingeniero civil en computación.
- [13] Yehuda Katz. Emberjs home page. <http://emberjs.com/>, June 2013.
- [14] Mauro San Martín. A model for social networks data management, 2012. Tesis para optar al grado de doctor en ciencias mención computación.
- [15] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, first edition, structural analysis in the social sciences edition, 1994.