

DISEÑO DE UNA RED SOCIAL DE EGRESADOS DE LA UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR BASADA EN WEB SEMÁNTICA E IMPLEMENTACIÓN DE ONTOLOGÍAS

JEISON FOX, GERMAN HERNANDEZ, TILSON ALTAMAR
Universidad Tecnológica de Bolívar



**DISEÑO DE UNA RED SOCIAL DE EGRESADOS DE LA
UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR BASADA EN
WEB SEMÁNTICA E IMPLEMENTACIÓN DE ONTOLOGÍAS**

TILSON ALTAMAR COGOLLO

JEISON ALEXANDER FOX PÉREZ

GERMÁN DANILO HERNÁNDEZ SÁNCHEZ

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR
PROGRAMA DE INGENIERÍA DE SISTEMAS
FACULTAD DE INGENIERÍA
CARTAGENA DE INDIAS**

2012

**DISEÑO DE UNA RED SOCIAL DE EGRESADOS DE LA
UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR BASADA EN
WEB SEMÁNTICA E IMPLEMENTACIÓN DE ONTOLOGÍAS**

TILSON ALTAMAR COGOLLO

JEISON ALEXANDER FOX PÉREZ

GERMÁN DANILO HERNÁNDEZ SÁNCHEZ

**Trabajo de tesis para optar al título de Ingeniero de
Sistemas**

MOISES QUINTANA ALVAREZ

Magister en Informática Aplicada

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR
PROGRAMA DE INGENIERÍA DE SISTEMAS
FACULTAD DE INGENIERÍA
CARTAGENA DE INDIAS
2012**

CONTENIDO

INTRODUCCION	1
1. ONTOLOGÍA	3
1.1 DEFINICIÓN DE ONTOLOGÍA	3
1.1.1 Componentes de una Ontología	5
1.1.2 Clasificación de las Ontologías	5
1.2 CREACIÓN DE LA ONTOLOGÍA	7
1.2.1 Metodología para la creación de la ontología	7
1.2.2 Aplicación de METHONTOLGY para la creación de una ontología de una red social	11
1.2.3 Herramientas utilizadas para la creación de la ontología	16
2. REDES SOCIALES	19
2.1 DEFINICION DE REDES SOCIALES	19
2.2 HISTORIA DE LAS REDES SOCIALES	19
2.2.1 Historia de las redes sociales en internet	22
2.3 CREACION DE UNA RED SOCIAL	23
2.3.1 Arquitectura de una red social	24
2.3.2 Herramientas utilizadas para la creación de una red social	26
3. APLICACIONES SEMÁNTICAS	31
3.1 ANTECEDENTES DE APLICACIONES SEMANTICAS	31
3.2 DIFERENCIA DE APLICACIONES TRADICIONALES CON APLICACIONES SEMÁNTICAS	34

3.3 PROCESO DE DESARROLLO DE APLICACIONES SEMÁNTICAS	36
3.3.1 Tecnologías para el desarrollo de aplicaciones semánticas	40
3.3.2 Persistencia de ontologías con Jena	43
4. CREACION DE SOCIAL UTB	53
4.1 METODOLOGIA PARA LA CREACION DE SOCIAL UTB	53
4.2 DESCRIPCION DE REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES	54
4.2.1 Definiciones, acrónimos y abreviaturas	54
4.2.2 Especificaciones funcionales	56
4.2.3 Especificaciones suplementarias	56
4.3 DESARROLLO DEL MODELO CONCEPTUAL	59
4.4 DESARROLLO DE CASOS DE USO	60
4.5 DESARROLLO DE LAS ACTIVIDADES	61
4.6 ARQUITECTURA DE SOCIAL UTB	61
4.7 DIAGRAMA DE CLASES	63
4.8 IMPLEMENTACION DE SOCIAL UTB	64
4.8.1 Empire	65
4.8.2 Empire y SocialUTB	65
5. CONCLUSIONES	70
6. RECOMENDACIONES	72
BIBLIOGRAFÍA	74
ANEXOS	77

LISTA DE TABLAS

Tabla 1: Framework para web semántica	39
Tabla 2: Framework para web semántica	43
Tabla 3: Cuadro comparativos SESAME vs Jena	51

LISTA DE FIGURAS

Figura 1. Representación grafica de una ontología	7
Figura 2. Diagrama de taxonomías de conceptos	13
Figura 3. Diagrama de relaciones binarias	14
Figura 4. Ejemplo de un grafo de una parte de una red social	21
Figura 5. Pilares de una red social	24
Figura 6. Componentes de una aplicación social	26
Figura 7. Modelo de datos de ELGG	27
Figura 8. Ejemplo de red social creada con NING	29
Figura 9. Estructura de OPEN SOCIAL	30
Figura 10. Evolución de la web	33
Figura 11. Estado de las tecnologías de la web semántica	34
Figura 12: Comparación entre web tradicional y web semántica	36
Figura 13: Arquitectura Jena	42
Figura 14: Creación de un modelo relacional persistente de una ontología	46
Figura 15: Carga del driver JDBC para MySQL al inicio de una aplicación que trabaje con conexiones	47
Figura 16: Arquitectura SESAME	48
Figura 17: Esquema de persistencia SESAME en MySQL	50
Figura 18: Modelo Conceptual de SocialUTB	59
Figura 19: Diagrama general de casos de uso de SocialUTB	60
Figura 20: Arquitectura física de SocialUTB	61
Figura 21: Diagrama de clases de SocialUTB	63
Figura 22: Ejemplo de uso de Empire	65

Figura 23: Modificación de la clase JenaConfig	66
Figura 24: Archivo de configuración SDB.TTL	67
Figura 25: Creación del modelo SDB	68
Figura 26: Persistencia mediante Empire	69

GLOSARIO

APACHE: es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual

API (*APPLICATION PROGRAMMING INTERFACE*): es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas (también denominadas vulgarmente "librerías").

DBMS (*DATA BASE MANAGER SYSTEM*): son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

GUID (*GLOBALLY UNIQUE IDENTIFIER*): es un número pseudoaleatorio empleado en aplicaciones de software. Aunque no se puede garantizar que cada GUID generado sea único, el número total de claves únicas es tan grande que la posibilidad de que se genere un mismo número dos veces puede considerarse nula en la práctica.

ISBN (*INTERNATIONAL STANDARD BOOK NUMBER*): es un identificador único para libros, previsto para uso comercial.

J2EE: es una plataforma de programación —parte de la plataforma Java— para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

JDBC (*JAVA DATABASE CONNECTIVITY*): es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos

a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

Jena: es *framework* semántico *open source* para Java. Provee un API para extraer datos de y escribir a grafos RDF.

JPA (*Java Persistence API*): es un framework del lenguaje de programación Java que maneja datos relacionales en aplicaciones usando la Plataforma Java en sus ediciones Standard (Java SE) y Enterprise (Java EE).

JSF: es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE.

MVC (*MODEL–VIEW–CONTROLLER*): es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos.

MYSQL: es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

OWL (*WEB ONTOLOGY LANGUAGE*): es un lenguaje de marcado para publicar y compartir datos usando ontologías en la WWW. OWL tiene como objetivo facilitar un modelo de marcado construido sobre RDF y codificado en XML.

PHP (*PHP HYPERTEXT PRE-PROCESSOR*): es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

POA (*PROGRAMACIÓN ORIENTADA A ASPECTOS*): es un paradigma de programación relativamente reciente cuya intención es permitir una adecuada modularización de las aplicaciones y posibilitar una mejor separación de incumbencias.

POO (PROGRAMACIÓN ORIENTADA A OBJETOS): es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento.

RDF (*RESOURCE DESCRIPTION FRAMEWORK*): es un framework para metadatos en la World Wide Web (WWW), desarrollado por el *World Wide Web Consortium* (W3C).

RUP (*RATIONAL UNIFIED PROCESS*): es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

SOA (*SERVICE ORIENTED ARCHITECTURE*): es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio.

SPARQL (*SPARQL PROTOCOL AND RDF QUERY LANGUAGE*): se trata de un lenguaje estandarizado para la consulta de grafos RDF, normalizado por el RDF Data Access Working Group (DAWG) del World Wide Web Consortium (W3C). Es una tecnología clave en el desarrollo de la Web Semántica que se constituyó como Recomendación oficial del W3C el 15 de Enero de 2008.

URL (*UNIFORM RESOURCE LOCATOR*): es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación, como por ejemplo documentos textuales, imágenes, vídeos, presentaciones digitales, etc.

W3C (*WORLD WIDE WEB CONSORTIUM*): es un consorcio internacional que produce recomendaciones para la World Wide Web.

XML (*EXTENSIBLE MARKUP LANGUAGE*): es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C). Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma

manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes.

PLANTEAMIENTO DEL PROBLEMA

DESCRIPCIÓN DEL PROBLEMA

Actualmente la Universidad Tecnológica de Bolívar almacena la información de sus egresados en una base de datos recopilada en hojas electrónicas y en muchos casos esa información se encuentra incompleta.

Otro inconveniente que se presenta es la duplicidad de la información. En la base de datos actual, existen registros repetidos pertenecientes a la misma persona.

También existen problemas de integridad ya que muchos de los egresados no poseen toda la información completa y no todos los egresados poseen la información en la misma base de datos. Adicionalmente algunos de los egresados tienen información en el sistema Sirius y otros no. Por esta razón el envío de correspondencia física o por cualquier otro medio se dificulta porque la información no es confiable.

Todo esto hace que la relación egresado-universidad no sea tan estrecha y productiva como se desearía que fuese. Muchas veces la Universidad Tecnológica de Bolívar tiene una oferta de trabajo y debido al hecho que no tiene una forma confiable, directa y rápida de contactarse con sus egresados, termina ofreciendo dicha oferta a personas ajenas a ella misma. Actualmente a la Universidad Tecnológica de Bolívar, no le es posible aprovechar la base de conocimientos que podrían generar sus egresados.

Cabe mencionar que tampoco existe una forma en que sus egresados estén en constante interacción entre ellos

FORMULACIÓN DEL PROBLEMA

La Universidad Tecnológica de Bolívar con más de 30 años de funcionamiento ha conseguido posicionarse como una de las mejores universidades de la región Caribe; al convertirse y ubicarse como una de las mejores instituciones educativas para la educación superior. La Universidad Tecnología de Bolívar gana adeptos de una manera muy fácil y con el paso de los años la juventud mira en esta, la mejor oportunidad de formarse como profesional.

Actualmente la Universidad cuenta con más de 5.000 egresados los cuales dan muestra de la excelente labor que realiza la Universidad Tecnológica de Bolívar. Esta cifra de egresados es un valor que con el paso de los años tiende a aumentar a un ritmo acelerado. Dado el alto número de egresados con los cuales cuenta la Universidad, es normal preguntarse lo siguiente: ¿La Universidad tiene forma de saber que está sucediendo con cada uno de sus egresados? ¿Es correcta y completa la información que la Universidad posee acerca de cada uno de sus egresados? ¿Existe un mecanismo confiable que le haga llegar al egresado información que le pueda ser beneficiosa para su situación económica, social, profesional?

Estas son algunas de las problemáticas que presenta la Universidad Tecnológica de Bolívar con respecto al manejo de la información de sus egresados.

JUSTIFICACIÓN

Actualmente la Universidad Tecnológica de Bolívar cuenta con una base de conocimiento conformada por más de cinco mil estudiantes egresados. Esta extensa cantidad de conocimiento, es sumamente envidiable y deseada. Solo existe un problema; el cual es que por más que se cuente con tal base de conocimientos, no existe una forma en la cual poder consultarla en su totalidad ni mucho menos aprovecharla al máximo. Esto es por que, no existe un recurso con el cual la Universidad Tecnológica de Bolívar pueda ponerse en contacto con sus egresados de una manera limpia, instantánea y segura, sin intermediarios, solo la universidad y sus egresados. En estos momentos, no existe una relación universidad-egresados de forma directa y constante.

Y es esa brecha la que SocialUTB viene a cerrar. Su finalidad es la de generar un medio en el cual la Universidad Tecnológica de Bolívar pueda estar en contacto inmediato y directo con sus egresados, todo esto mediante el espacio que la red virtual social ofrece. En los últimos años, las redes sociales virtuales se han convertido en una parte muy importante en la vida del ser humano promedio y todo apunta a que esta tendencia se mantendrá en el futuro. Las personas hoy en día invierten mucho de su tiempo en ellas, ya sea por recreación, negocios, investigación o simplemente por ocio. El hecho de que SocialUTB se haya planteado como una red social virtual no es cuestión de azar. Este planteamiento fue elegido después de una investigación, que demostró que la forma mas interactiva y menos aburrida para que los usuarios estuviesen en constante interacción con la institución, fue mediante una red social virtual. Pero no se quería simplemente crear una red social virtual sin innovación, se deseaba una red social virtual que fuese única, que constara de significado para las máquinas y así de esta manera se pudiesen automatizar procesos que de otra forma tendrían que realizar las personas. Y es por esto que se decide dar el paso a la web 3.0 y crear la primera red social virtual basada en la web semántica que existe en Colombia.

Para generar un vínculo más estrecho entre la universidad y sus egresados que permita el aprovechamiento de sus capacidades y que a la vez se beneficien de las ofertas que la universidad tiene para ofrecerles no es únicamente el objeto de la creación de la SocialUTB; también se busca que los egresados con el tiempo y la interacción entre sí, comiencen a generar nuevos conocimientos, convirtiéndose así a SocialUTB, en un espacio para la generación y discusión de nuevos conocimientos.

OBJETIVOS

Objetivo General

- Diseñar e implementar una red social basada en web semántica, para los egresados de la Universidad Tecnológica de Bolívar.

Objetivos Específicos

- Comprender los principios que rigen el concepto de web semántica y redes sociales.
- Identificar y manejar las diferentes tecnologías más importantes que soportan a la web semántica y las redes sociales.
- Reconocer y estudiar las ontologías existentes que puedan servir de base para el diseño y creación de la red social.
- Establecer cada uno de los agentes que componen la red social y describir detalladamente sus atributos y relaciones.
- Diseñar y crear las ontologías necesarias para poder describir los agentes que componen la red social y las relaciones entre ellos.
- Diseñar la interfaz de la red social y establecer sus políticas de seguridad.
- Diseñar y crear la base de datos de la red social.
- Codificar la red social mediante un lenguaje de programación previamente definido.

INTRODUCCION

Desde sus inicios el hombre se ha caracterizado por ser un ser social con necesidad de informar y a su vez de ser informado, pasando desde la tradición oral para poder mantener vivas antiguas costumbres y leyendas hasta la aparición de la escritura que da origen oficial a la historia como tal. Con la aparición de la escritura surgen nuevos medios de comunicación los cuales permiten acortar distancia logrando satisfacer esa necesidad de información.

A medida que evoluciona el hombre nace y crece la tecnología permitiendo suplir necesidades que antes era inimaginables y con esta evolución de tecnología aparece el computador uno de los inventos que revolucionó la historia permitiendo agilizar muchos procesos de la vida diaria.

Después de la aparición del computador muchos creían que ya no vendría nada mas; estaban totalmente equivocados con la aparición de Internet el mundo entero se convirtió en una sola red y esa necesidad de información logra satisfacerse. Con Internet la información abunda por doquier pero surge un inconveniente y es el exceso de esta, la información es tanta que al momento de buscar algo muchos resultados de las búsquedas que hechas no tienen nada que ver con tema que interesa.

Como se observa con el paso del tiempo la tecnología evoluciona a pasos agigantados y con esta nace una nueva posibilidad que permitirá solventar el problema de exceso de información y esas son las Ontologías. Estas como se observará durante el trabajo permite dar una descripción semántica al contenido de internet haciendo que el navegador no interprete si no que entienda cual es el contenido que busca y con esto el resultado será certero. Ahora bien el hombre como ser social encontró la posibilidad con internet de expandir sus círculos sociales y crear comunidades virtuales, aquí es donde nacen las redes sociales virtuales. Imaginemos entonces si aprovechamos esas redes sociales y la dotamos de contenido semántico de tal forma que si necesitamos encontrar a una

persona al momento de buscar en esa red nos muestre esa persona que queremos y no información que no interesa como sucede muchas veces, aquí es donde comienza nuestra investigación en la cual se tendrá la oportunidad de conocer nuevas tecnologías y observar como surge una nueva red social totalmente semántica Social UTB.

1. ONTOLOGÍA

1.1. DEFINICIÓN DE ONTOLOGÍA

Para llevar la web semántica a la realidad es necesario que los datos se puedan expresar de una forma, tal que puedan ser interpretados por los computadores y además puedan ser reutilizables. La ontología es el medio que permite que esto pueda suceder.

Las ontologías en la informática provienen de la inteligencia artificial, John McCarthy introdujo el término en 1980; son vocabularios comunes para las personas y aplicaciones que trabajan en un dominio - un área específica de interés o un área de conocimiento - Según el grupo de trabajo en ontologías del consorcio W3C, una ontología define los términos que se usan para describir y representar un cierto dominio. Toda ontología representa cierta visión del mundo con respecto a un dominio. Por ejemplo, una ontología que defina "ser humano" como "espécimen vivo o muerto correspondiente a la especie Homo sapiens; primate bípedo que pertenece a la familia de los homínidos, como los chimpancés, gorilas y orangutanes" expresa una visión del mundo totalmente distinta a la de una ontología que lo defina como "sujeto consciente y libre, centro y vértice de todo lo que existe; todos tienen la misma dignidad, pues han sido creados a imagen y semejanza de Dios"[1].

En el campo de la ciencia de la computación existen muchas definiciones de ontología, tales como la que plantea Gruber, quien define una ontología como "una especificación explícita de una conceptualización" [2]. Aunque en el marco de este proyecto, la definición de ontología más acertada y la cual se empleara en este proyecto, es la planteada por Studer, Benjamins, y Fensel, quienes describen a las ontologías como:

Una especificación explícita y formal de una conceptualización compartida. Conceptualización se refiere a un modelo abstracto de algún fenómeno en el mundo, habiendo identificado los conceptos relevantes de dicho fenómeno. Explícita significa que el tipo de conceptos usados, y las restricciones de su uso están explícitamente definidas. Formal se refiere al hecho de que la ontología debe ser leíble para las máquinas. Compartida refleja la noción de que una ontología captura conocimiento colectivo, lo cual no la priva también de capturar conocimiento individual, pero el cual es aceptado por un grupo [3].

Cualquier persona tiene en su cabeza ontologías mediante las que representa y entiende el mundo que lo rodea. Estas ontologías no son explícitas, en el sentido de que no se detallan en un documento ni se organizan de forma jerárquica o matemática pero desafortunadamente las máquinas no poseen estas ontologías y es por eso que no pueden percibir lo que está alrededor y necesitan de unas ontologías que estén explícitas

Un ejemplo ya centrado en el campo de la web semántica sería: En la web semántica se manejan las denominadas bases de conocimiento que bien se pueden considerar sistemas de información, ahora bien a la hora que dos sistemas de información intentan comunicarse se producirán problemas semánticos que dificultarán la comunicación entre ellos. Entonces se entra a los problemas semánticos.

Estos problemas son de dos tipos: de dominio y de nombre. Los conflictos de dominio aparecen cuando conceptos similares en cuanto a significado, pero no idénticos, se representan en distintos dominios. Por ejemplo, el concepto representado por trabajador en una base de datos puede corresponder a un trabajador cualificado, mientras que otra, puede usar trabajador para cualquier trabajador, sea o no cualificado. Ambos conceptos están muy vinculados, pero no son equivalentes ni deberían mezclarse. Es aquí donde las ontologías juegan su papel y haciendo uso de ellas se podría especificarse que el primer concepto

corresponde a una especialización del segundo; y un sistema de razonamiento automático basado en ontologías impediría, por ejemplo, que se contratara para tareas calificadas a trabajadores no calificados.

El segundo de los problemas semántico es el problema de nombre. Este conflicto generalmente se presenta de dos formas: sinónimos y homónimos. Los sinónimos ocurren cuando los sistemas usan distintos nombres para referirse al mismo concepto. Por ejemplo, una base de datos puede usar trabajador para el mismo concepto que otra usa empleado. En ese caso, se podría usar una ontología que definiera como idénticos los dos términos. Así, las aplicaciones que manejaran esas bases de datos sabrían como llevar datos de una a otra.

1.1.1. Componentes de una Ontología.

Las ontologías tienen muchas similitudes estructurales sin importar el lenguaje en el cual estén hechas, algunas poseen componentes que otras no poseen, pero todas las ontologías comparten cuatro componentes bases que son lo mínimo que debe poseer una ontología, dichos componentes son:

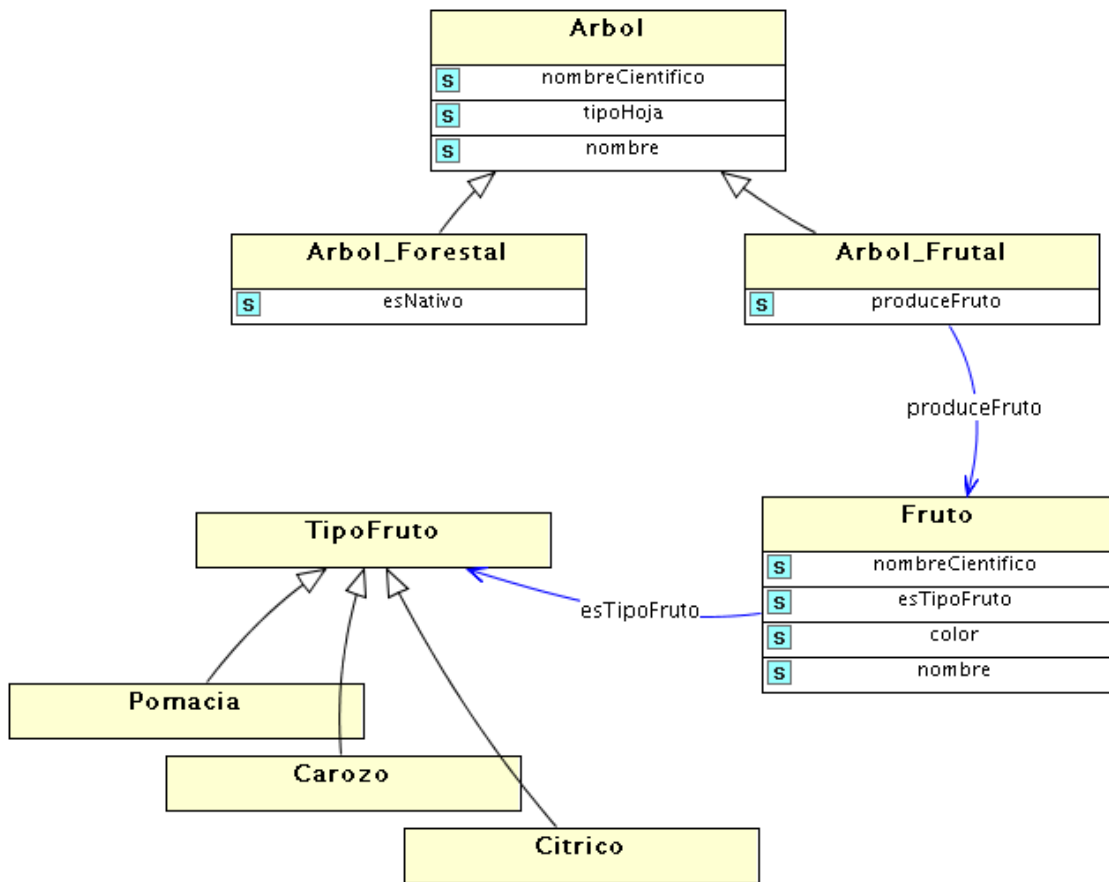
- Individuos: son instancias u objetos, son los objetos básicos de toda ontología.
- Clases: conjunto, tipo, colección, conceptos de objetos o cosas.
- Atributos: aspectos, propiedades, características o parámetros que los objetos y las clases pueden tener.
- Relaciones: formas en las cuales los objetos y clases pueden estar relacionados los unos con los otros [4].

1.1.2. Clasificación de las Ontologías.

Dependiendo del grado de formalidad, las ontologías explícitas se clasifican en informales, semi-informales, semi-formales y formales. Las primeras se expresan

directamente en cualquier lenguaje natural. Las segundas se expresan en una forma estructurada y restringida de algún lenguaje natural. Las terceras se expresan en lenguajes estructurados, como RDF. Por último, las ontologías formales definen los términos mediante lenguajes lógico-matemáticos cuyos símbolos se definen exactamente y sin ambigüedades; en consecuencia, estas ontologías permiten emplear teoremas y demostraciones. Los dos últimos tipos de ontologías permiten que las aplicaciones puedan usar las definiciones de los conceptos del dominio y sus relaciones y son estas las que se utilizara para el debido progreso de la web semántica. Los tres primeros tipos de ontologías pueden contener términos ambiguos o inconsistentes, las ontologías formales no los permiten. Preguntas tan capciosas como "El único barbero de una ciudad afeita a todos los hombres que no se afeitan a sí mismos ¿Quién lo afeita a él?" ó "¿A qué conjunto pertenece el conjunto de los conjuntos que no se incluyen a sí mismos?" no pueden representarse en una ontología formal [5].

Figura 1. Representación grafica de una ontología.



Fuente: <http://www.dcc.uchile.cl/~ekrsulov/slides/titulo/slide3-0.html>

1.2. CREACIÓN DE LA ONTOLOGÍA

1.2.1. Metodología para la creación de la ontología.

Con el motivo de partir de una base sólida para poder realizar un correcto diseño de la ontología a usar durante el proyecto, se investigaron las metodologías de diseño y creación de ontologías más usadas y conocidas. Las cinco metodologías estudiadas y analizadas fueron:

- **DILIGENCE:** Metodología de desarrollo de ontologías basada en la colaboración de múltiples participantes, de manera que la creación de una ontología se concibe como un proceso social, distribuido y muy poco controlado. Se hace hincapié en el consenso, facilitando un marco para la discusión de diferentes propuestas y el intercambio de argumentos a favor de las diferentes posibilidades hasta llegar a un acuerdo final. Se asume también que toda ontología debe evolucionar constantemente a lo largo del tiempo, de forma que permite introducir nuevas etapas en ellas, en las cuales los usuarios pueden introducir sucesivas modificaciones [6].

- **On-To-Knowledge:** Desarrollado por la Universidad de Karlsruhe, hace hincapié en las aplicaciones futuras de la ontología a la hora de diseñarla. Los pasos esenciales de que consta son:
 - **Estudio de viabilidad:** Previo al desarrollo de la ontología y base para el proceso siguiente.
 - **Inicio:** Proceso de delimitación del dominio y objetivo de la ontología, extracción de las fuentes de conocimiento (libros, revistas, documentos, etc.), junto con la descripción de sus usuarios y aplicaciones futuras.
 - **Refinamiento:** Producción de una ontología orientada a sus aplicaciones, conforme las especificaciones extraídas del proceso anterior.
 - **Evaluación:** Prueba de la utilidad de la ontología y del entorno de software asociado a ella en la aplicación para la que fue diseñada.
 - **Mantenimiento:** Decisión sobre el responsable de esta tarea y de los procesos posibles en ella [7].

- **Competency Questions:** La metodología de las “Preguntas Relevantes” ó “Preguntas de Verificación” es una de las más sencillas de aplicar. Consiste en determinar el dominio y el alcance de la ontología mediante la

lista de preguntas que el sistema debería ser capaz de contestar. Las respuestas a estas preguntas sugieren lo que podrían ser las instancias de la ontología, a partir de las cuales se deducirían (generalizando) las clases de la misma. Al mismo tiempo, estas preguntas servirán como factor decisivo a la hora de evaluar la propia ontología, permitiendo comprobar si se ha representado la suficiente información como para poder responder a dichas cuestiones, especialmente relevantes.

- **NeOn:** La Metodología NeOn para la construcción de redes de ontologías es una metodología basada en escenarios que se apoya en los aspectos de colaboración de desarrollo de ontologías y la reutilización, así como en la evolución dinámica de las redes de ontologías en entornos distribuidos. Las claves de la Metodología NeOn son:
 - Un conjunto de nueve escenarios para la construcción de ontologías y redes de ontologías, haciendo hincapié en la reutilización de los recursos ontológicos y no ontológicos, la reingeniería y la fusión, y teniendo en cuenta la colaboración y el dinamismo.
 - El glosario de procesos y actividades identifica y define aquellos procesos y actividades involucrados en el desarrollo de redes de ontologías.
 - Directrices metodológicas para diferentes procesos y actividades del proceso de desarrollo de la ontología de la red, tales como la reutilización y la reingeniería de los recursos ontológicos y no ontológicos, la especificación de los requisitos de la ontología, la localización de la ontología, la programación, etc. Todos los procesos y actividades se describen con una tarjeta llena, un flujo de trabajo, y ejemplos [8].

- **METHONTOLOGY:** Desarrollado en la Universidad Politécnica de Madrid. Propone un ciclo de vida de construcción de la ontología basado en

prototipos evolutivos, porque esto permite agregar, cambiar y renovar términos en cada nueva versión (prototipo). Para cada prototipo, el proceso consta de los siguientes pasos esenciales:

- **Especificación:** Consiste en delimitar los objetivos de su creación (compartir información entre personas o por agentes software; permitir la reutilización del conocimiento de un dominio; hacer explícitas las suposiciones que se efectúan en un dominio; separar el conocimiento del dominio del conocimiento sobre su fabricación o forma de operación; ó simplemente analizar el conocimiento del dominio), decidir el dominio de actuación de la ontología (para no modelizar objetos poco relevantes en perjuicio de otros más importantes), quién la usará y para qué, las preguntas a las que deberá responder (ayuda al establecimiento de las dos anteriores), y quién se encargará de su mantenimiento (decidiendo si se limitará a introducir nuevas instancias, se permitirá la modificación de conceptos o atributos, etc.)
- **Conceptualización:** Consiste en crear un glosario de términos que pertenecen al dominio, definirlos y crear una taxonomía (estableciendo una clasificación o jerarquía entre los conceptos, sus niveles, las relaciones entre ellos, sus instancias, sus propiedades o atributos, e igualmente los axiomas o reglas).
- **Formalización:** Proceso consistente en convertir el modelo anterior en un modelo formal ó semi computable. Se puede emplear en este paso una herramienta como Protégé.
- **Implementación:** Convierte el modelo formalizado en un modelo computable mediante un lenguaje para construcción de ontologías. Se puede emplear en este paso una herramienta como Protégé.
- **Mantenimiento:** Labor que puede acarrear desde el borrado de instancias ya inútiles o la incorporación de nuevas instancias que se han ido produciendo con el tiempo, hasta las tareas de introducción

de cambios en el contenido de la información, ya sea redefiniendo atributos, relaciones o incluso conceptos [9].

Para el diseño de la ontología se tomó como base la metodología creada dentro del grupo de ingeniería ontológica de la universidad politécnica de Madrid, llamada METHONTOLOGY, como ya se mencionó anteriormente. El motivo por el cual se decidió utilizar esta metodología como base para la creación de la ontología a usar por la red social, fue debido a las siguientes razones:

- La buena documentación con la cual cuenta. Esta fue una de las principales razones por la cual se selecciono a esta metodología sobre las demás.
- La variedad de proyectos desarrollados con esta metodología que sirvieron de guía para la aplicación de esta en el proyecto.
- Su gran nivel detalle, el cual hace más fácil la elaboración de la ontología en el software a usar para su creación.
- Su fácil implementación.

A continuación se ilustra el proceso de creación de la ontología SocialUTB usando la metodología seleccionada.

1.2.2. Aplicación de METHONTOLGY para la creación de una ontología de una red social.

Paso 1.

Definición del dominio: Determinar el dominio y el alcance de la ontología.

¿Cuál es el dominio que la ontología cubrirá?

La representación de una red social de estudiantes egresados de la Universidad Tecnológica de Bolívar.

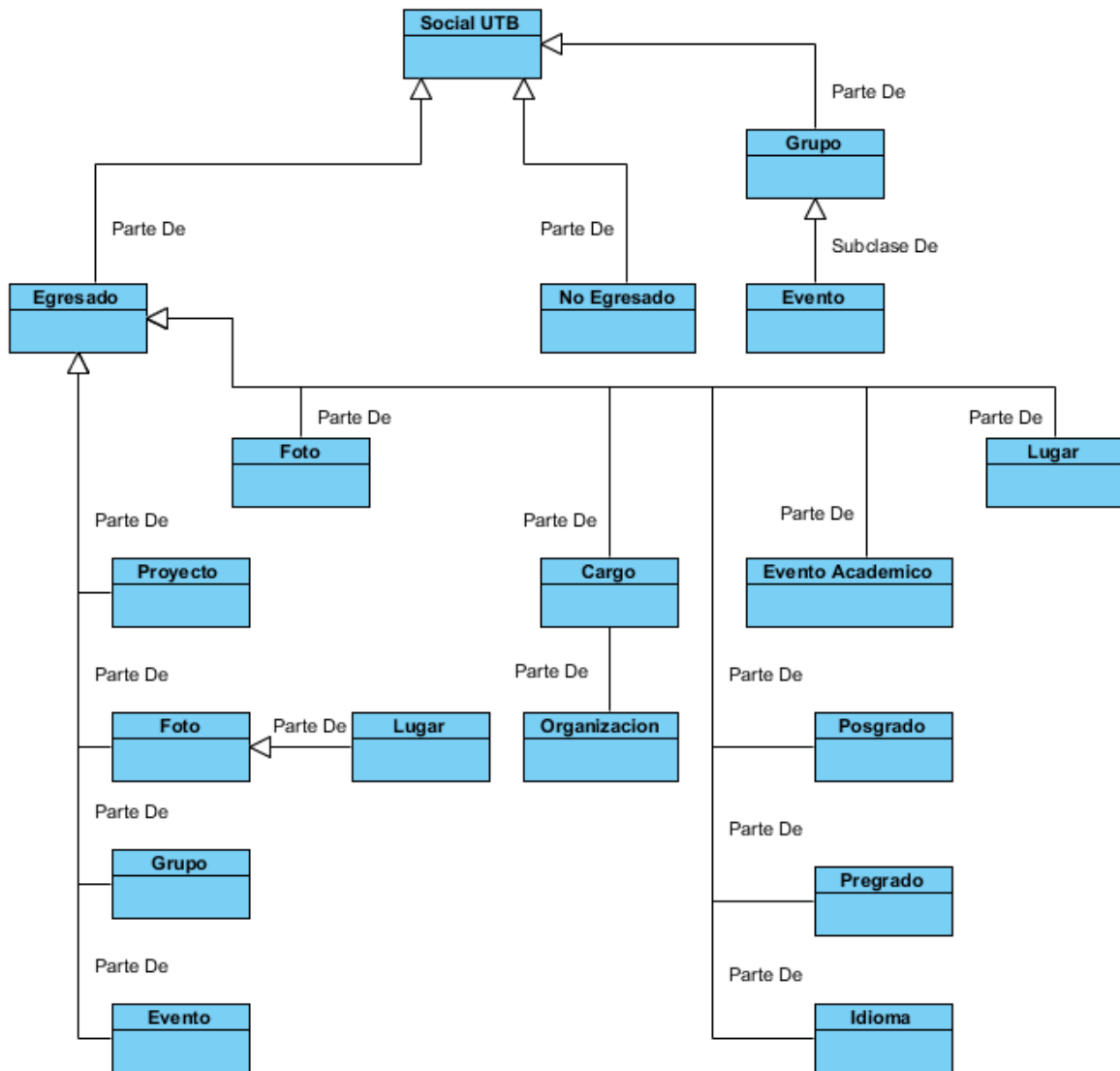
Paso 2.

Construcción de un glosario de términos: Incluye todos los términos relevantes del dominio (conceptos, instancias, atributos, relaciones entre conceptos, etc.), sus descripciones en lenguaje natural, y sus sinónimos y acrónimos. (Ver tabla en Anexo A).

Paso 3.

Construcción de taxonomías de conceptos: Seleccionar del glosario aquellos términos que son conceptos. Esta taxonomía define la jerarquía entre los conceptos del dominio.

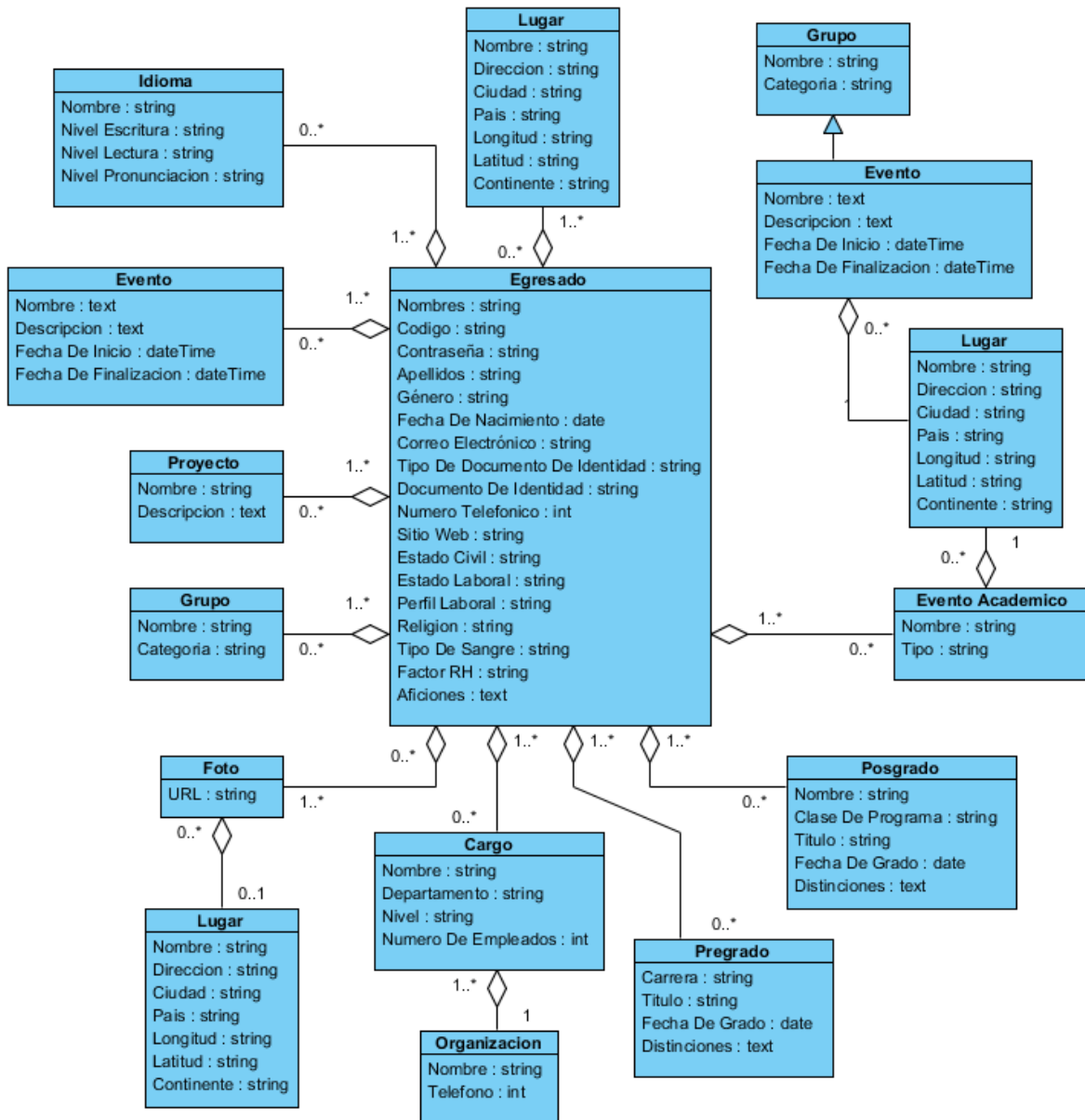
Figura 2. Diagrama de taxonomías de conceptos.



Paso 4.

Construcción de un diagrama de relaciones binarias: El objetivo de este diagrama es establecer las relaciones existentes entre conceptos de la ontología (o con otras ontologías existentes) y sus inversas.

Figura 3. Diagrama de relaciones binarias.



Paso 5.

Construcción de un diccionario de conceptos: Especificar cuáles son las propiedades que describen cada concepto de la taxonomía, las relaciones del

diagrama de relaciones binarias y las instancias de cada uno de los conceptos. (Ver tabla en Anexo B).

Paso 6.

Descripción en detalle de las relaciones binarias: Para cada relación binaria se debe especificar: nombre, nombres de sus conceptos origen y destino, cardinalidad y relación inversa, si existe. (Ver tabla en Anexo C).

Paso 7.

Descripción en detalle de los atributos de instancias: Por cada atributo de instancia se debe especificar: nombre, concepto al que pertenece, tipo de valor, rango de valores (para valores numéricos) y cardinalidad. (Ver tabla en Anexo D).

Paso 8.

Descripción en detalle de los atributos de la clase: Para cada atributo de clase se debe especificar: nombre del atributo; nombre del concepto donde el atributo se define; tipo de valor; valor(es); unidad de medida y precisión del valor (en el caso de valores numéricos); cardinalidad. Debido a que ninguno de los conceptos de la ontología posee algún atributo de clase, este paso se omite.

Paso 9.

Descripción en detalle de las constantes: Para cada constante del glosario de términos se debe especificar: nombre, tipo de valor, valor y unidad de medida (para constantes numéricas).

Debido a que ninguno de los conceptos de la ontología posee alguna constante, este paso se omite.

Paso 10.

Definición de los axiomas formales: Para realizar esta tarea se debe identificar los axiomas formales (expresiones lógicas que son siempre verdaderas y son usadas normalmente para especificar restricciones en la ontología), que son

necesarios en la ontología y describirlos de manera precisa. (Ver tabla en Anexo E).

Paso 11.

Definición de reglas: Identificar qué reglas (usadas generalmente para inferir conocimiento en la ontología) se necesitan en la ontología, y posteriormente describirlas en la tabla de reglas. METHONTOLOGY propone especificar las expresiones de las reglas utilizando el formato *si <condiciones> entonces <consecuencias o acciones>*. (Ver tabla en Anexo F).

Paso 12.

Codificación de la ontología: Llevar la ontología a un lenguaje entendible para las computadoras, ya sea RDF u OWL.

1.2.3. Herramientas utilizadas para la creación de la ontología.

Para la elaboración de la ontología en un lenguaje entendible por las computadoras se utilizó Protégé. Protégé es un ambiente extensible multiplataforma para la creación de ontologías y bases de conocimiento basado en Java. Sus principales características son:

- Posee una interfaz de usuario muy intuitiva y fácil de usar.
- Altamente escalable.
- Fácilmente extensible.

Su extensibilidad radica en el hecho de que incorpora el uso de *plugins*, los cuales extienden su funcionalidad más allá de los límites planteados en el momento de su creación. Protégé viene con unos *plugins* por defecto, pero la inmensa mayoría de los *plugins* que se pueden encontrar en la web para Protégé, son hechos por su comunidad de usuarios. La activa comunidad de usuarios que tiene a su alrededor hace que periódicamente se creen nuevos *plugins* o se mejoren los ya existentes.

Además de incorporar *plugins*, Protégé incorpora también razonadores semánticos o como se conocen comúnmente, *reasoners*. Estos *reasoners* juegan un papel muy importante en el proceso de la creación de la ontología en Protégé. Un *reasoner* es una pieza de software capaz de inferir consecuencias lógicas de un conjunto de hechos afirmados o axiomas. Básicamente los *reasoners* sirven para probar la consistencia lógica de la ontología hallando inconsistencias lógicas en esta. Una ontología que contiene inconsistencias lógicas, es una ontología que no está bien diseñada, por ende es una ontología que no sirve.

Existen muchos *reasoners* para OWL, algunos de ellos son pagos, tales como:

- RacerPro
- Bossam
- OntoBroker
- SHER

Mientras que otros son gratis, como lo son:

- FaCT++
- KAON2
- Pellet
- HermiT

Pellet es el *reasoner* que trae por defecto Protégé y es uno de los que mejor rendimiento y estabilidad tiene con él, debido a esto, fue que mayormente se utilizó durante el desarrollo del proyecto, aunque no fue el único, también se usaron los *reasoners* FaCT++ y HermiT.

Protégé tiene dos ramas, la 3.x y la 4.x. Existen varias diferencias importantes entre ambas ramas que pueden influenciar al usuario en su decisión final de cual emplear a la hora de crear una ontología, he aquí unas recomendaciones para saber cual rama es la adecuada para el proyecto a realizar:

- Si se desea trabajar con ontologías basadas en *frames*, la única elección posible es la versión 3.4 de Protégé, ya que la versión 4.x no los soporta.

- Si se desea crear aplicaciones OWL puras, se recomienda usar la versión 4.x
- Si se desea trabajar con OWL 2.0, se debe usar la versión 4.x ya que la versión 3.4 no soporta OWL 2.0.
- Existen muchos *plugins* que funcionan en la versión 3.4 que no son soportados en la versión 4.x, si el uso de uno de estos *plugins* es primordial para la creación de la ontología, se recomienda usar la versión 3.4.
- La versión 4.x es más liviana que la versión 3.4, además cuenta con una interfaz más sencilla.

En el proyecto se decidió usar ambas ramas de Protégé. Primeramente se uso la versión 3.4.7 para el proceso de creación y codificación de la ontología. Se decidió usar la versión 3.4.7 de Protégé debido a la amplia gama de *plugins* que este soporta y que hacen más fácil el proceso de creación y codificación de la ontología.

Una vez que la ontología fue creada usando la versión 3.4.7, se pasó a la versión 4.1. El uso de la versión 4.1 fue debido a que la versión 4.x soporta una gama más amplia de *reasoners* que la versión 3.4., los cuales son esenciales para el *testing* y el chequeo de inconsistencias en la ontología. Además de la razón antes expuesta, se usó la versión 4.1. Porque se deseaba crear una ontología basada en OWL 2.0 puro, el cual es solo soportado en la versión 4.x.

2. REDES SOCIALES

Después de conocer el proceso de creación de una ontología nos adentramos en el mundo de las populares redes sociales analizando sus antecedentes, su historia, y las herramientas que facilitan su creación. De esta forma se podrá observar cuál es la ruta correcta a tomar para la creación de una red social basada en web semántica.

2.1. DEFINICION DE REDES SOCIALES

Una red social es una estructura social fundamentada en la unión de nodos que por lo general son individuos u organizaciones que interactúan entre sí. Un error que se comente comúnmente es relacionar las redes sociales con la web, pero la realidad es otra totalmente diferente.

Las redes sociales no son nuevas, no son algo que va y viene como puede pensar la mayoría de las personas, estas provienen de la antigüedad y como todas las cosas existentes en este mundo han tenido su proceso de evolución hasta llegar al nivel actual donde se encuentran.

Para entender mejor el funcionamiento de las redes sociales es bueno conocer como ha sido comportamiento a través de la historia.

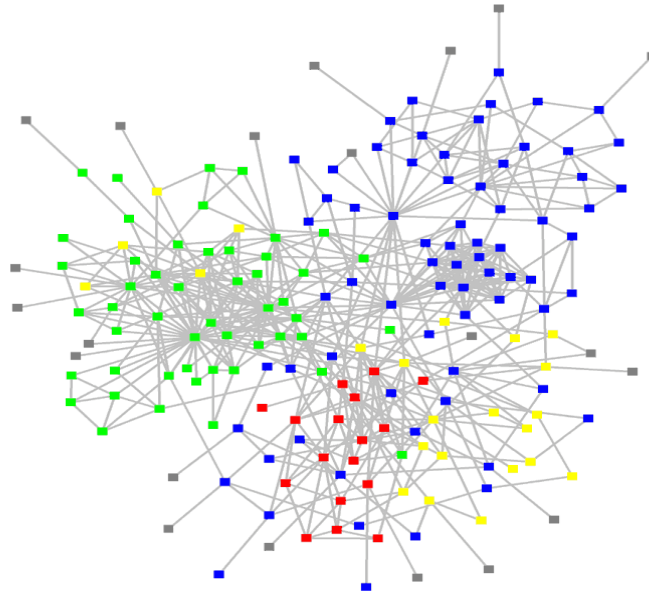
2.2. HISTORIA DE LAS REDES SOCIALES

La evolución de las redes sociales se puede describir en tres etapas.

- La primera etapa se le puede considerar como la etapa del “cara a cara” donde esta era la única forma posible de crear y mantener la red.
- La segunda etapa comienza con los avances en las tecnologías de comunicación; aparecen los correos, el teléfono, el fax, que en conjunto al sistema cara a cara, se añaden como ingredientes para darle un toque de modernización y amplían mucho más el concepto de la red social.
- La tercera y la “última” etapa llega con la aparición de Internet, permitiendo expandir aun más el concepto de redes sociales. Se puede decir que con el Internet comienza el “boom” de las redes sociales. Es en esta etapa donde nacen las redes sociales virtuales. Simón Tenzer, Olga Ferro y Nuria Palacios definen a las redes sociales virtuales como “una estructura social de relaciones entre usuarios a través de internet. Son sitios basados en la web, que permiten a los usuarios compartir contenido, interactuar y crear comunidades sobre intereses similares” [10].

Conociendo un poco mejor el concepto de red social, a continuación se analiza desde el punto de vista computacional. La filosofía de la red social continúa siendo la misma, pero se aprovecha el internet como una tecnología que permite la expansión de una manera mucho más ágil y eficiente. Ahora bien como se mencionó, una red social consta de un conjunto de nodos que representan a los actores, enseguida se puede observar una relación de complejidad la cual es directamente proporcional al número de nodos en la red, es decir entre mas actores tenga la red mas nodos tendrá esta y por ende mayor será su complejidad. Esta es la razón por la cual los grafos de las redes sociales por lo general suelen ser muy complejos.

Figura 4. Ejemplo de un grafo de una parte de una red social.



Fuente: <http://www.infovis.net/printMag.php?num=136&lang=1>

Las redes sociales en Internet han ganado su lugar de una manera acelerada convirtiéndose en promisorios negocios para empresas y sobretodo en lugares para encuentros humanos. A diferencia del sistema cara a cara mencionado en la primera etapa y los ingredientes añadidos en la segunda etapa, en la tercera etapa las redes sociales en Internet brindan la posibilidad de interactuar con otras personas aunque no se conozcan. El sistema es abierto y se va construyendo obviamente con lo que cada suscripto a la red aporta, cada nuevo miembro que ingresa transforma al grupo en otro nuevo. La red no es lo mismo si uno de sus miembros deja de ser parte de ella.

Intervenir en una red social empieza por hallar allí, a otros con quienes compartir intereses, preocupaciones o necesidades mutuas, de esa forma se rompe el aislamiento que suele aquejar a la gran mayoría de las personas, lo cual suele manifestarse en retraimiento y otras veces en excesiva vida social sin afectos comprometidos.

2.2.1. Historia de las redes sociales en internet.

Alrededor de 2001 y 2002 surgen los primeros sitios que fomentan redes de amigos. Hacia 2003 se hacen populares con la aparición de sitios tales como Friendster, Tribe y Myspace.

Rápidamente algunas empresas ingresan a las redes sociales. Google lanza en enero de 2004 Orkut, apoyando un experimento que uno de sus empleados realizaba en su tiempo libre. En 2005 ingresan Yahoo 360° y otros.

Básicamente el funcionamiento comienza cuando una vez montado el soporte técnico, un grupo de iniciadores invitan a amigos y conocidos a formar parte de la red social, cada miembro nuevo puede traer consigo muchos nuevos miembros y el crecimiento de esa red social puede ser geométrico.

Y he aquí que se transforma en un interesante negocio. Un buen ejemplo de esto es Facebook, una red social enfocada a estudiantes, muy similar a Myspace, con millones de usuarios registrados y donde ha habido una importante inversión publicitaria de parte de Microsoft.

No todo es buenaventura para las redes sociales, conforme se fue convirtiendo en buen negocio, surgieron intereses y por lo tanto, corre cierto peligro, al menos en su filosofía de funcionamiento libre, ya que la empresa Friendster.com logró que la oficina de patentes de los Estados Unidos le acepte una patente sobre redes sociales la cuál cubre:

“Todo sistema, método y aparato para conectar usuarios en sistemas computarizados en línea basándose en sus relaciones en redes sociales” [11].

Actualmente las redes sociales virtuales son parte de nuestra vida diaria, se estima que Facebook, la cual es la red social virtual con más usuarios registrados en el mundo, supera los 900 millones de usuarios, lo cual es una cantidad sumamente impresionante. Dada la enorme cantidad de personas que están vinculadas a las redes sociales, las empresas han comenzado a invertir en estas. Esta el caso de Dell, la cual hace ofertas exclusivas para los usuarios que la

siguen en Twitter, Microsoft ha invertido muchos millones en publicidad a través de Facebook. Un ejemplo de una empresa colombiana que está aprovechando las redes sociales para hacer publicidad es Avianca, la cual al igual que Dell, ofrece ofertas exclusivas para sus seguidores en Twitter.

Inclusive, se han creado redes sociales virtuales enfocadas cien por ciento al ámbito empresarial, como es el caso de LinkedIn y Xing, las cuales están enfocadas a los negocios, permitiendo así establecer contactos profesionales con personas de todo el mundo.

2.3. CREACION DE UNA RED SOCIAL

“Las personas quieren conectarse a Internet y ver que hacen sus amigos, así que, ¿Por qué no construir una página de Internet que les ofrezca eso? Amigos, fotos, perfiles y lo que puedan visitar y encontrar. Quizás es alguien que conocieron en una fiesta. Eduardo, no estoy hablando de un sitio de citas por Internet, estoy hablando de llevar por completo la experiencia social de la universidad y ponerla en línea” [12].

Con esta filosofía fue que se origino Facebook la red social mas popular en la actualidad, la cual nos lleva a preguntar como se crea una red social o que cuales son los pasos para crear una red social.

Desafortunadamente no existe una arquitectura estándar para desarrollar una red social o un patrón definido para organizar los componentes que permitirán hacer de una página web, una debida red social.

Pero lo que si se ha logrado analizar es que todas las redes sociales en la actualidad poseen una arquitectura muy similar que facilita a los usuarios la

capacidad de interactuar entre si, creando un espacio donde no solo reciban información, sino que también la trasmitan.

2.3.1. Arquitectura de una red social.

Parte de esa arquitectura similar que comparten todas las redes sociales contiene cinco pilares principales que son mencionados a continuación:

- Organización en la red.
- Pequeñas influencias.
- Métricas relevantes.
- Tipología de la red.
- Identidad virtual.

Figura 5. Pilares de una red social.



Fuente: RAMIREZ TERRY, Jose. Web 2.0 y redes sociales.

Los cuales permiten orientar el enfoque que tendrá la red social (profesional, búsqueda de amigos, parejas, académica, etc.).

Pero aparte de estos 5 pilares para que una aplicación pueda ser considerada como una red social, de tener cuatro aspectos fundamentales:

- Un componente social: este aspecto es fundamental y en este se describe cual es el aspecto social que se desea implementar.
- Tecnologías: hace referencia a las tecnologías que se utilizarán para el desarrollo de la aplicación.
- Distribución: con este aspecto definimos como nuestra aplicación podrá interactuar con otras aplicaciones externas.
- APIs de Terceros: Al momento de incluir APIs de terceros aumenta la posibilidad de convertir nuestra red social mucho más interactiva.

Como se puede notar, el componente social es el corazón de la red social, aquí se define como se comunicarán los usuarios y como estos usuarios podrán sacar beneficio a los aportes realizados por los diferentes actores en la red. Este es el momento donde se define la temática de la red. Una red social académica creada con el fin de garantizar la comunicación entre los egresados de la Universidad Tecnológica de Bolívar, eso es lo que busca SocialUTB.

Figura 6. Componentes de una aplicación social.



Fuente: RAMIREZ TERRY, Jose. Web 2.0 y redes sociales.

Conociendo la arquitectura básica que debe tener una red social se hace necesario mirar que herramientas nos se pueden emplear para el desarrollo de una red social

2.3.2. Herramientas utilizadas para la creación de una red social.

Como se mencionó anteriormente, las redes sociales no cuentan con un estándar de diseño para su desarrollo, más bien se siguen patrones de diseño como por ejemplo: Modelo, vista, controlador (MVC) y patrones de implementación que están definidos para cada lenguaje de programación con el cual se esté trabajando.

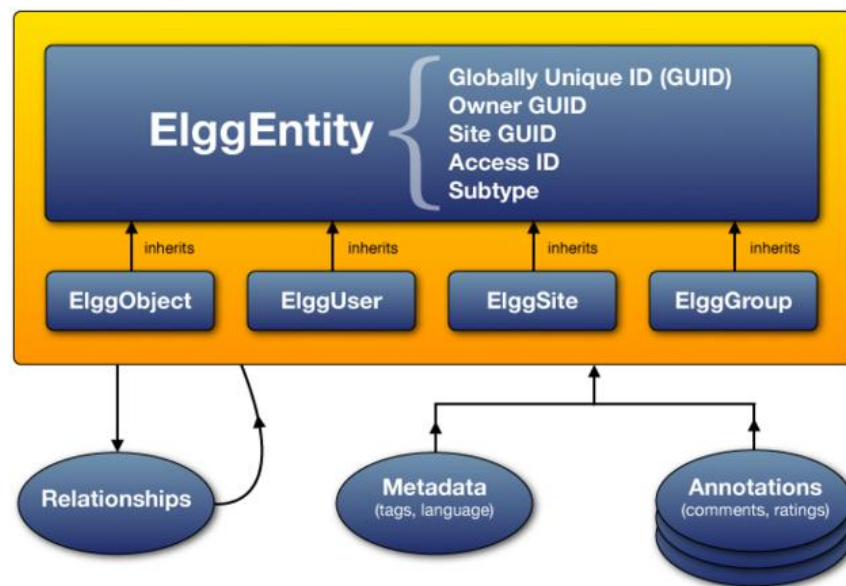
Actualmente se han desarrollado muchas APIs, *frameworks* y herramientas que incorporan estos patrones, facilitando el proceso de creación de una red social.

A continuación se listan las herramientas más populares para la creación de redes sociales.

- ELGG:

Figura 7. Modelo de datos de ELGG.

Elgg Data Model



Fuente: <http://docs.elgg.org/wiki/Engine/DataModel>

Elgg es una plataforma abierta para redes sociales personales, desarrollado con tecnología Linux, Apache, MYSQL y PHP. Lo que se aprecia en la figura siete, es el modelo conceptual de los componentes de ELGG, donde la clase **ElggEntity** es una entidad que representa cualquier elemento perteneciente al sistema, además de controlar permisos, propiedad, entre otros. **ElggObject** son objetos de entrada, es decir archivos subidos y favoritos. **Elgguser** hace referencia a cada usuario del sistema. **Elggsite** es

cada sitio creado en una instalación ELGG. Elgggroup son las llamadas comunidades. Todas estas clases al ser hijas de Elggentity heredan propiedades y comportamientos como el identificador único global (GUID), permisos de acceso, propietario o entidad a la que pertenece [13].

A las entidades se les puede añadir más información por medio de etiquetas (*tags*), número ISBN, o simplemente por la configuración de un idioma o la información de un archivo.

Si bien, ELGG captura metadatos de los datos compartidos (*metadata*), esto es un proceso manual y subjetivo, además, posee componentes bien estructurados y las aplicaciones se integran por medio de APIs, por lo que la interacción con la plataforma es rígida y hay que adaptarse a las interfaces proporcionadas, la cual genera cierta dependencia [13].

- NING.

Esta herramienta a diferencia de ELGG es totalmente virtual. Es una herramienta en línea que permite a los usuarios crear redes sociales de una manera rápida y sencilla.

Una de las ventajas que posee NING es facilitar la elección de la temática de la red brindando al usuario la posibilidad de personalizar su red sin tener conocimientos de programación.

Figura 8. Ejemplo de red social creada con NING.



Fuente: <http://www.ning.com>

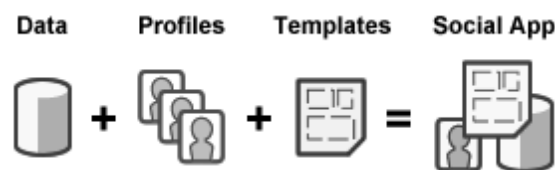
En la figura ocho, se puede apreciar una red social creada con Ning donde se puede demostrar las principales características ofrecidas por esta herramienta. Otra de las ventajas de esta herramienta es permitir la elección del tipo de red es decir, si la red será privada o publica y acordar cual será su funcionalidad.

A diferencia de ELGG, NING dejó de ser una herramienta gratuita para convertirse en una de pago lo cual ha ocasionado una gran deserción de los usuarios NING.

- OPEN SOCIAL.

A diferencia de las herramientas mencionadas anteriormente OPEN SOCIAL sería catalogado como un servicio abierto proporcionado por Google y sus socios a través de APIs para crear aplicaciones web. En la Figura nueve se puede observar el uso de las APIs para el acceso a la información de los perfiles de usuario, información de los amigos y sus actividades y *gadgets* que se pueden embeber en cualquier sitio web. En otras palabras, las fuentes de datos y el grafo social pueden estar en Orkut, Yahoo, MySpace, LinkedIn, Hi5 o NING, dotadas de *gadgets* (pequeñas aplicaciones o *plugins*) que tienen el mismo medio de comunicación: OPEN SOCIAL.

Figura 9. Estructura de OPEN SOCIAL.



Fuente: http://wiki.opensocial.org/index.php?title=Articles_%26_Tutorials

La gran ventaja de OPEN SOCIAL es la posibilidad de usar varios lenguajes de programación del lado del cliente para crear aplicaciones. A pesar de esta ventaja, aún no se provee un servicio que permita hacer anotaciones semánticas, que por otra parte, y aprovechando la estructura de esta API, podría cubrir diferentes niveles de datos como, perfiles, actividades y datos compartidos, de una manera efectiva [14]. OPEN SOCIAL se puede catalogar como un intento de Google por estandarizar el desarrollo de aplicaciones sociales.

3. APLICACIONES SEMÁNTICAS

Después de observar cómo se crea una ontología y conociendo que debe tener una red social para ser considerada una red social, se hace necesario mirar que antecedentes han tenido las aplicaciones semánticas ya que Social UTB es una aplicación de este tipo.

3.1. ANTECEDENTES DE APLICACIONES SEMANTICAS

Antes de comenzar con los antecedentes es necesario comprender que es la web semántica, Tim Berners-Lee, define la web semántica como “una extensión de la web actual en la que la información se le da un significado bien definido, logrando que computadores y personas puedan trabajar en cooperación” [15]. Lo cual indica que la finalidad, es que las computadoras bien podrán comportarse como una persona y procesar la información lo más parecido posible a como una persona lo haría, esto causaría que la comunicación hombre maquina mejore y tenga más sentido.

Desde principios del año 2000 ya se hablaba de lo que es la interacción entre diferentes entes sobre un mismo punto. Ya la web era un poco más dinámica, habían sistemas de reservas en tiempo real, contenidos dinámicos, interoperabilidad, flujo de datos, aparecen lo que son los servicios web, XML, etc. Desde los principios de la web semántica cuando Tim Berners-Lee hizo anuncio de esta, se ha venido hablando mucho sobre las mejoras que vendrían con su implementación y las grandes aplicaciones que con esta se podrían hacer y consigo han surgido varias tecnologías para fortalecer a la llamada web 3.0 (web semántica) y hacerla más cercana a los desarrolladores y contribuyentes de la web actual.

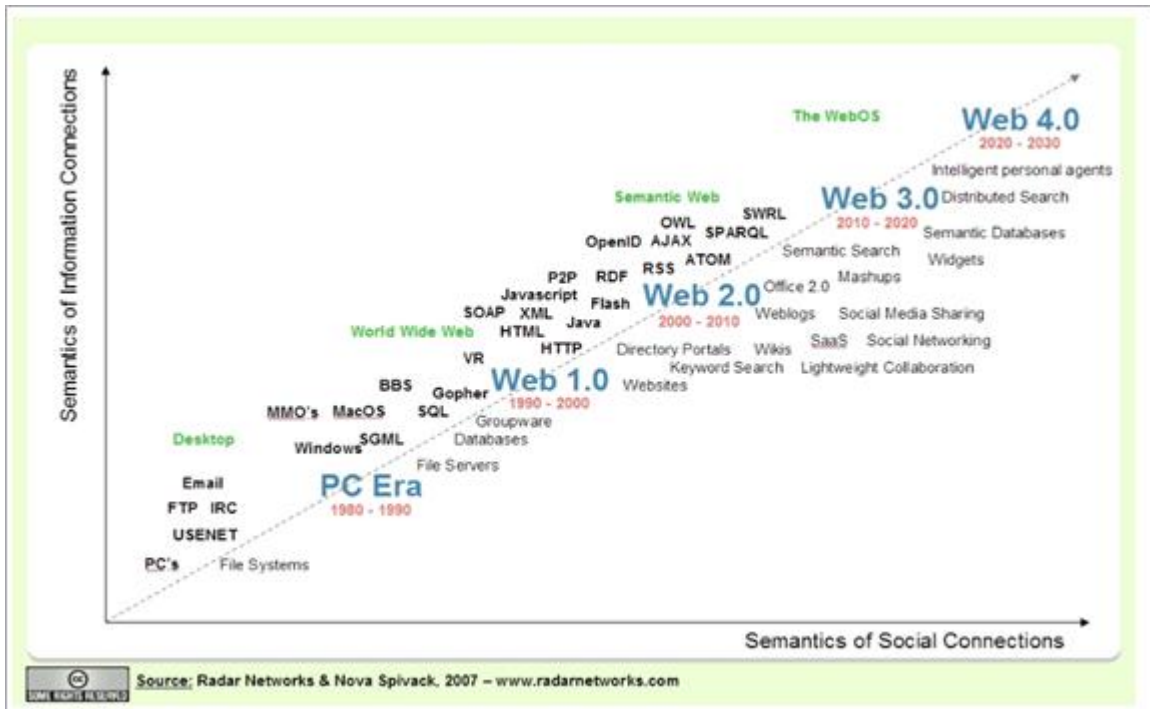
Después de esa definición que proporciona Tim Berners-lee, han surgido aplicaciones tomando como base una web de datos, esto ha causado que vayan tomando más fuerzas los términos como RDF, OWL, SPARQL entre otros, motivando esto a la aparición de las primeras aplicaciones semánticas y con ello han ido apareciendo las búsquedas semánticas ó inteligentes. Actualmente existen grandes grupos que están trabajando en la web semántica como son: Clark & Parsia, ILOG, Tenforce, HP, Agfa, SRI International, Telecom Italia, Oracle, Boeing, IBM, Chevron, Siemens, Nokia, Pfizer, Sun, Eli Lilly, Fair Isaac Corp [15].

Existen varios nichos y usos para este tipo de aplicaciones. Los principales usos que se le ha dado a la web semántica han sido para:

- Integración de datos (desde bases de datos dispersas).
- Asistencias de usuarios.
- Portales inteligentes, incluyendo buscadores sobre ontologías (Kngine, Hakia, Kosmix, DuckDuckGo, Evri, Powerset, Truevert).
- Gestión del conocimiento: soporte a decisiones.
- Integración de servicios (Business-to-business, Business-to-consumer).
- Repositorios estructurados.
- Herramientas cooperativas.
- Sistemas de evaluación de riesgo, de ayuda al cumplimiento de directivas (de seguridad financiera) [15].

En la siguiente imagen se puede observar cómo ha evolucionado la web a través del tiempo, mostrando todas sus etapas y las proyecciones de esta y cada una de las tecnologías que la acompañan en cada una de sus etapas.

Figura 10. Evolución de la web.



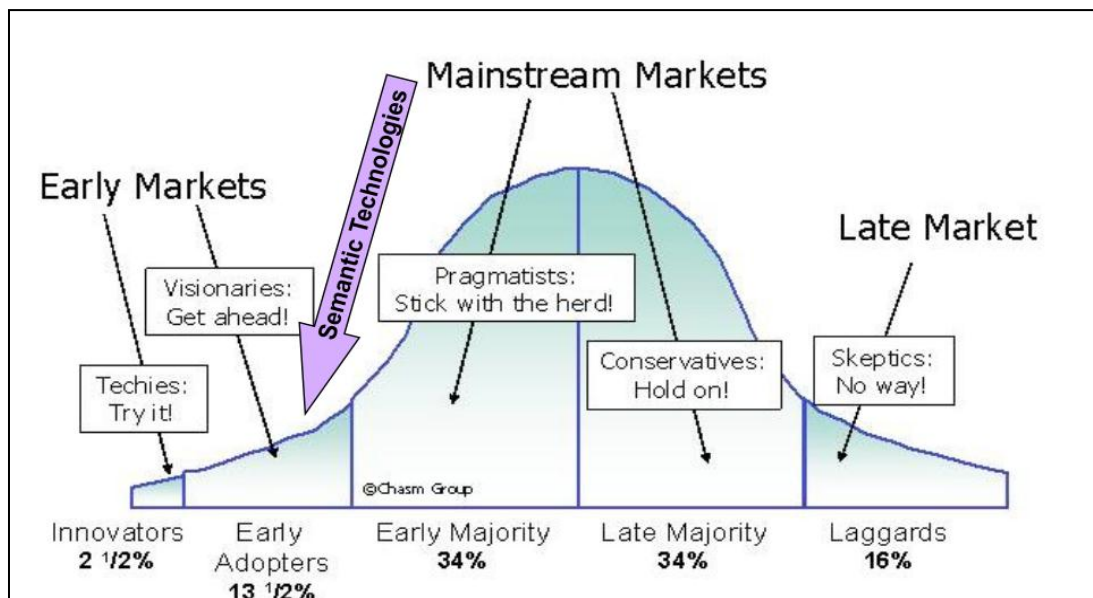
Fuente: Aplicaciones Industriales de la Web Semántica, En: Día W3C España (2008, Madrid). Presentaciones de Día W3C en España: Standards for Business, p. 9.

Los principales nichos donde se están implementando cada una de estas aplicaciones son la salud, finanzas, seguridad, turismo y cultura. Con el pasar del tiempo estos nichos se han ido incrementando ya que los datos inteligentes son más funcionales que los simples enlaces. Desde museos hasta empresas de automotriz son participes de esta nueva generación de la web inteligente donde lo importante es la comunicación entre las aplicaciones, y dado que todas manejan un mismo lenguaje, el proceso resulta más ágil y menos complejo.

Actualmente la web semántica se encuentra en una etapa de adaptación y asimilación por sus futuros consumidores y propulsores, aunque ya existen varios proyectos desarrollados de esta naturaleza.

El enfoque que más ha tomado fuerza es el mejoramiento de las búsquedas con sentido semántico puesto que la web semántica ofrece la posibilidad de realizar consultas más puntuales mejorando la calidad de la información obtenida y la posibilidad de interactuar con otras bases ontológicas con facilidad.

Figura 11. Estado de las tecnologías de la web semántica.



Fuente: Aplicaciones Industriales de la Web Semántica, En: Día W3C España (2008, Madrid). Presentaciones de Día W3C en España: Standards for Business, p. 10.

3.2. DIFERENCIA DE APLICACIONES TRADICIONALES CON APLICACIONES SEMÁNTICAS

Toda evolución trae consigo características nuevas, ventajas y hasta cambios estructurales; más que diferencias entre la web actual y la web semántica, la transición a la web semántica es un cambio de paradigma que se veía venir y se hacía necesario para seguir evolucionando. Ahora se puede tener aplicaciones

que compartan información más naturalmente, todos hablando el mismo lenguaje y sobre una misma plataforma. Realmente, este es un nuevo medio de interoperabilidad y distribución de la información que beneficiara a muchas personas. Estando en el mundo de la web semántica vienen a la luz nuevas tecnologías y conceptos como: OWL, RDF, SPARQL, agentes inteligentes.

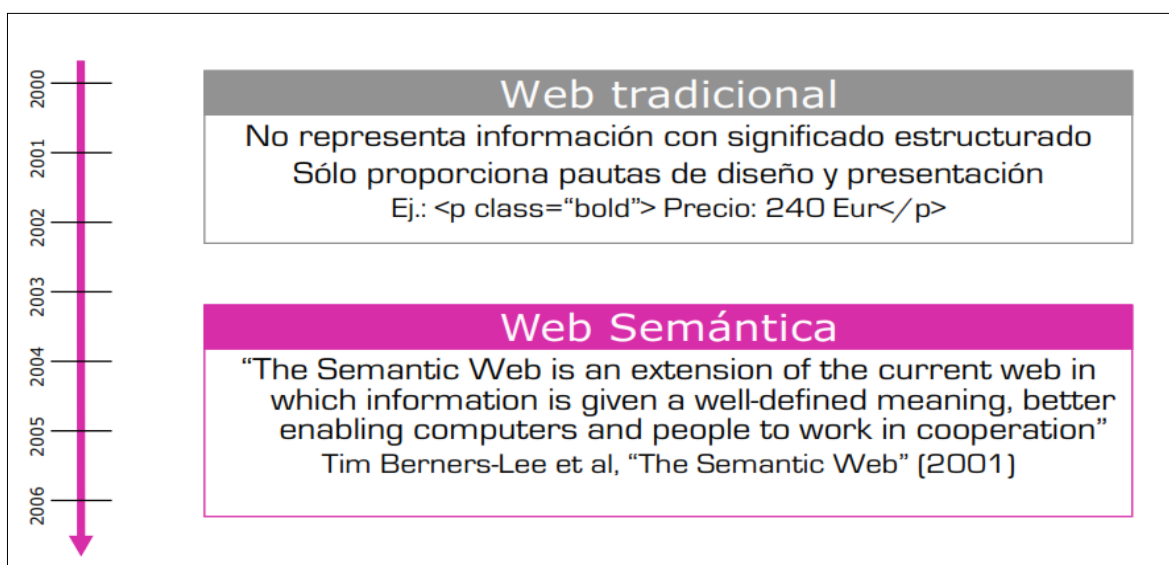
Estos nuevos conceptos y tecnologías representan un gran aporte a la web semántica, la comunicación es entre datos directamente y no entre simples textos o URL como en la web actual. Esto permite organizar la información en conceptos; garantizando búsquedas sobre éstos por significado y no por contenido textual que es lo que en la actualidad se hace.

Aunque esta transición es un poco forzada, ya que las personas actualmente se sienten muy cómodas en la web 2.0 y está claro que este cambio requiere una reconstrucción de los recursos actuales, debido a que las aplicaciones actuales carecen de respuestas exactas para usuarios más avanzados donde los contenidos tienen sentido y significancia. En la web 3.0 cada uno de estos conceptos pueden ser compartidos con facilidad porque las bases ontológicas de estos permiten tener recursos en diferentes aplicaciones en las cuales son utilizados, permitiendo saber a qué se refieren estos conceptos al momento de reutilizar los sin repetirlos, la cual es una de las problemáticas que afronta la web actual, donde existen un sin número de datos o recursos repetidos, los cuales a su vez confunden y hacen que los resultados no sean exactos y no se elija con certeza una información porque no se sabe su procedencia. La semántica y la unificación de los recursos van de la mano.

Cuando se cuente con una plataforma unificada en la cual se hable un mismo lenguaje regido por las mismas reglas, en ese momento se podría hacer uso de agentes inteligentes que podrán realizar trabajos que normalmente hace una persona, además la comunicación con otras plataformas se hará sin ningún inconveniente mientras se realice bajo las premisas planteadas anteriormente.

La diferencia entre la web 2.0 y la web 3.0 se puede resumir en la siguiente imagen.

Figura 12: Comparación entre web tradicional y web semántica.



Fuente: Aplicaciones Industriales de la Web Semántica, En: Día W3C España (2008, Madrid). Presentaciones de Día W3C en España: Standards for Business, p. 6.

3.3. PROCESO DE DESARROLLO DE APLICACIONES SEMÁNTICAS

La construcción de aplicaciones semánticas en el entorno local y nacional es un campo que aun no se ha explorado en su totalidad, hay dos caminos para llegar a la implementación de una aplicación semántica. La primera es utilizando las herramientas que nos brindan las tecnologías como Jena, Sesame, LinqToRDF, etc., bajo sus lineamientos y las representaciones que estos ofrecen de cada uno de los artefactos de la web semántica.

Al construir una aplicación con esta metodología se requiere un tiempo de desarrollo considerable y una fuerte mano de ingeniería para lograr obtener la robustez y extensibilidad que un sistema requiere, además de fácil mantenimiento, esto solo por mencionar algunas métricas que requiere un software de alta calidad. En base a todas estas variables, y después de una ardua investigación y recopilación de experiencias encontradas en distintos trabajos de grados y publicaciones sobre la temática, en los cuales hacían ciertas recomendaciones en lo que respecta la implementación de aplicaciones semántica basadas en las arquitecturas y paradigmas actualmente conocidos como: POO¹, SOA², POA³, entre otros; la idea que surgió fue la de modificar las metodologías existentes, y de esa forma conseguir que el desarrollo de SocialUTB no esté tan alejado de las metodologías mas conocidas actualmente, lo que daría como resultado, una disminución de la curva de aprendizaje de todo lo relacionado con la semántica, y así de esta forma conseguiríamos aplicaciones de buena calidad.

La otra opción es realizar lo último mencionado y tomando como referencias los *frameworks* como herramientas de trabajo, los cuales brindan ciertos mecanismos que disminuirían el tiempo de desarrollo de las aplicaciones. Mas adelante se hará mención a los *frameworks* para el desarrollo de aplicaciones semánticas que existen actualmente para cada uno de los lenguajes de programación más utilizados. Para cada lenguaje existe al menos un *framework*.

La de cual opción escoger depende del tipo de aplicación a desarrollar. Se puede afirmar que sin la ayuda de estos marcos de trabajos, el proceso seria un poco más arduo. También se encontraron muchas tecnologías orientadas a las plataformas de desarrollo que existen en la actualidad para la creación de aplicaciones semánticas. “La mayoría de las aplicaciones semánticas se construyen utilizando los mismos principios fundamentales, componentes similares, y las variaciones de una arquitectura básica.” [16] Basándose en esto, la SocialUTB será construida tomando como base la programación orientada a

¹ Programación orientada a objetos

² Service Oriented Architecture

³ Programación orientada a aspectos

objetos, la plataforma J2EE y el uso de *frameworks* para construir un sistema robusto y fácil de extender basándonos en las reglas que estos nos presentan pero logrando un alto rendimiento y logrando que la curva de aprendizaje de los nuevos incursores en el mundo de la web semántica sea más rápida, menos traumática y más productiva en materia de avances en el dominio de la nueva web. Si a las aplicaciones comunes se les integra un modelo de conocimiento y se le suma la exposición de sus los datos en modelos RDF, se convertirían inmediatamente en aplicaciones semánticas.

Los *frameworks* para el desarrollo de aplicaciones semánticas brindan plataformas robustas sobre las cuales se puede trabajar y generar códigos para las aplicaciones semánticas, que sean bastante simples y entendibles. Estos facilitan una interfaz que es más común para los desarrolladores, ya no se tendría que escribir archivos OWL, ni en ningún otro formato que podría tomar algún tiempo en realizarse. Estas plataformas nos facilitan los mecanismos necesarios para simular la creación de estos formatos, y todo lo que son sus atributos, propiedades, clases, etc. Esto no debe entenderse como si los framework le restan sentido a la semántica, ya que estos solo sirven para la declaración de las sentencias; La web semántica sigue siendo el eje central de las aplicaciones. Cabe anotar que gran parte de estas plataformas están orientadas a objetos.

En la siguiente tabla podemos ver los diferentes framework existentes para el manejo de las tecnologías que están involucradas en la web semántica.

Tabla 1: Framework para web semántica.

FRAMEWORK	LANGUAGE/INTERFACE	SEMANTIC LEVEL
Jena	Java	RDF to OWL 2
Sesame	Java & RESTful web service	RDF
OWL API	Java	OWL 2
RAP-RDF API	PHP	RDF
Redland	C, Python, Ruby, Perl, & PHP	RDF
LinqToRDF	.NET	RDF

Fuente: Semantic web programming, 1 ed. EU: Wiley, 2009. Parte III Capitulo 8, p. 267.

Explicando un poco lo que se ve en la tabla anterior, se puede afirmar que para la construcción de las aplicaciones semánticas se tienen dos áreas principales. Una es el desarrollo de los datos y la otra es la que se encarga de gestionar los datos; la primera está directamente relacionada con la manipulación y tratamiento de los datos, de esto depende la calidad de los resultados y las inferencias que se hagan sobre esta. En cuanto a la gestión de los datos, los datos en los sistemas semánticos tienen un ciclo el cual se detallara a continuación. Se describen los pasos por los cuales pasa los datos en una aplicación semántica sabiendo que estos son el centro y los responsables del éxito de la aplicación, además de la calidad de las respuestas e inferencias que se puedan hacer.

El ciclo que cumple los datos en una aplicación semántica es:

- **Almacenamiento:** los *frameworks* para las aplicaciones semánticas pueden funcionar con los diferentes tipos de persistencia de datos que existen como son en memoria y base de datos, Las aplicaciones pueden funcionar con ambos sistemas de almacenamiento eso depende de lo que se requiera en el momento.

- **Poblado:** los diferentes *frameworks* presentados pueden alimentarse de los datos semánticos provenientes de archivos, recursos en la web, bases de datos y los que se construyan directamente mediante estos.
- **Combinaciones:** estas herramientas para el desarrollo de aplicaciones semánticas tienen la capacidad de hacer referencias de múltiples orígenes para realizar operaciones entre diferentes fuentes de datos, ya sean comparaciones, mezclas, intersecciones, etc.
- **Razonamiento:** estos marcos de trabajos brindan los mecanismos para realizar inferencias por características propias o externas sobre los datos de la aplicación, para la generación de información extra, es decir para ir formando bases de conocimientos.
- **Interrogación:** las búsquedas que se hacen sobre los almacenes de datos semánticos, se hacen basadas en coincidencias simples. Además las navegaciones que se realizan sobre estos datos se rigen por las relaciones establecidas en la lógica de los datos, cada una de estas consultas se realizara bajo un lenguaje natural.
- **Exportación:** los *frameworks* proporcionan métodos para exportar los datos de la web semántica en varios formatos.
- **Des-asignación /cerrar:** estos *frameworks* contienen mecanismos para eliminar las referencias de los datos almacenados y además la liberación los recursos informáticos asignados como memorias, espacio en disco entre otros [17].

3.3.1. Tecnologías para el desarrollo de aplicaciones semánticas.

Jena Semantic Web Framework.

Es un *framework open-source* para la construcción de aplicaciones Java relacionadas con la web semántica, esta herramienta brinda una equivalencia en el lenguaje que está programado, de cada uno de los artefactos de la web semántica, como clases, objetos, interfaces, funciones y atributos. Cabe anotar

que en la web semántica, los artefactos son la representación de los datos como recursos y que en Jena son representados en objetos y sus clases [18].

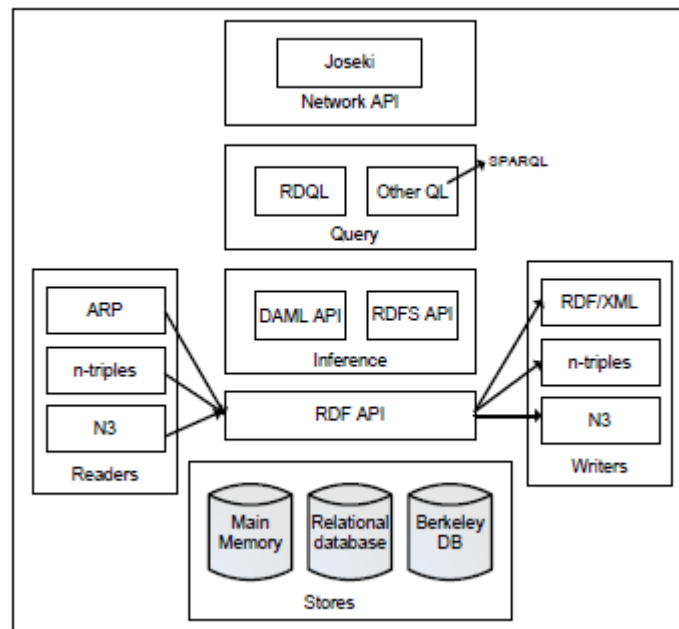
Esta herramienta incluye:

- API para RDF (*Resource Description Framework*).
- API para OWL (*Ontology Web Language*).
- Lectura y escritura RDF en formato RDF/XML, N3 y N-Triples.
- Motor de consultas SPARQL.
- Motores de inferencia y conectores a motores externos.
- Almacenamiento en memoria y almacenamiento persistente.
- Servidor HTTP de RDF [19] [20].

Arquitectura Jena.

A continuación se puede observar el esquema y jerarquía por la cual se rige Jena y que es la manera como funcionaria una aplicación semántica bajo este marco de trabajo [18].

Figura 13: Arquitectura Jena.



Fuente: Maria I. Lamas C: proyecto. Lenguajes de consulta para documentos rdf. Cataluña, 2006, p. 61.

En la siguiente tabla se puede ver un cuadro con las correspondencias de los artefactos de la web semántica y su representación en Java a través de Jena, esto ayudara a comprender con más facilidad como fue concebido Jena y de paso a recordar los conceptos que envuelven la web semántica con más facilidad permitiendo esto que los desarrolladores de la web actual comprendan como se puede extender la web actual hacia la 3.0 teniendo como bases las tecnologías actuales. Simplemente el cambio tiene que ser de mentalidad y comenzar a pensar en recursos de la web y no solo en enlaces que nos llevan a paginas con contenido, hay que llenar de sentido cada uno de estos datos y formar la web inteligente.

Tabla 2: Framework para web semántica.

ARTIFACT	SEMANTIC WEB	JENA JAVA CLASS	NOTES
Subject, predicate, object	URI	Resource, Property	A resource can be a subject, object, or predicate.
Statement	Statement	Statement	Special consideration for reified statements.
Data	Ontology and instance data	Graph and Model	Graphs are a basic building block for models. They both may contain ontology and instance data.
Query and results	SPARQL and Semantic Web data	Query and ResultSet	Analogous to relational databases.
Reasoner	Reasoner	Reasoner	Allow multiple internal and external reasoners.
Rules	SWRL	Reasoner	Rule support determined by specific reasoner.
Event notification	Not applicable	ObjectListener	Enable event-driven processing.

Fuente: Semantic web programming, 1 ed. EU: Wiley, 2009. Parte III Capitulo 8, p. 270.

3.3.2. Persistencia de ontologías con Jena.

Jena trabaja con dos subsistemas de la persistencia de datos RDF y OWL en una base de datos relacional en los principales sistemas de gestión de bases de datos como lo son MySQL, PostgreSQL, Microsoft SQL SERVER y Oracle, estos son SDB y TDB. Estos se agregan, ya que no vienen en el núcleo de Jena pero trabajan perfectamente con el.

TDB: Es un motor de almacenamiento completamente desarrollado en Java. Permite almacenar RDF y recuperarlos mediante consultas SPARQL.

SDB: Es un motor de almacenamiento que funciona como adaptador de un motor de base de datos relacional.

Jena soporta los siguientes motores, a partir de la versión indicada:

- PostgreSQL v8.
- MySQL 5.0.22.
- Apache Derby 10.2.
- Oracle 10gR2.
- Microsoft SQL Server 2005.
- IBM DB2.
- HSQLDB 1.8.
- H2 1.0.73

Permiten almacenar RDF y recuperarlos mediante consultas de SPARQL. Y además Permite analizar cómo se transforma SPARQL a SQL esto sirve para comprender un poco más este proceso dado que las consultas son totalmente diferentes a las realizadas en los lenguajes de consultas tradicionales.

Formas de trabajar con ontologías en Jena.

Almacenamiento en memoria.

Esta es la configuración que viene por defecto y con la que trabajan usualmente las aplicaciones semánticas. Es como el caso de variables que se guardan en la sesión, eso depende de la configuración que tenga el servidor. En algún momento pueden perder esta persistencia. Si la aplicación funciona de esta manera los datos estarán guardados en la memoria principal y en caso de pérdida de esta,

igualmente le pasara a los datos de las aplicaciones. Para trabajar con ontologías en Jena se puede hacer de la siguiente manera:

Declarando una variable `OntModel` e ir construyendo el modelo de la ontología con todos sus artefactos, clases, subclases, atributos, restricciones y reglas, en el tiempo de ejecución de la aplicación. Jena brinda además la posibilidad de cargar un modelo a partir de un fichero local donde este la ontología definida en el lenguaje OWL (Ver Anexo G) [19].

Almacenamiento persistente.

Otro mecanismo de almacenamiento es el persistente. Este surge ante la necesidad de almacenar los datos de la aplicación de forma duradera para su futura recuperación. El almacenamiento persistente en bases de datos relacionales tiene grandes ventajas sobre el almacenamiento en memoria mencionado anteriormente y sobre la persistencia en ficheros locales.

Cada uno de los motores de base de datos cuenta con un mecanismo para la conexión entre las aplicaciones y estas, en este caso de Java que es el lenguaje en el que está programado, por lo cual le corresponde el turno a la API JDBC, que es una colección de interfaces Java y métodos de gestión de manejadores de conexión para cada modelo específico de base de datos. En la siguiente ilustración podemos ver un ejemplo del proceso de persistir una ontología en una base de datos MySQL

Figura 14: Creación de un modelo relacional persistente de una ontología.

```
public static boolean createDatabaseSchema( String dbname,
                                           String dbuser,
                                           String dbpass,
                                           String dbengine,
                                           String dbmodelname)
{
    //Resultado a la salida del método
    boolean created = false;

    //Creación de la base de datos en caso de no existir previamente
    if (existsDatabase(dbname, dbuser, dbpass) == false)
    {
        createDatabase(dbname, dbusername, dbpass);
    }

    //Conexión a la base de datos local
    IDBConnection con = null;
    con = new DBConnection("jdbc:mysql://localhost/"+dbname, dbuser, dbpass, dbengine);

    try
    {
        //El esquema se crea partiendo de la base de datos vacia
        con.cleanDB();

        //Modelo Jena para almacenamiento persistente en la base de datos relacional
        ModelRDB RDBModel = null;

        //Creación de las tablas necesarias para almacenar el modelo 'dbmodelname'
        //El proceso utilizará la conexión existente
        RDBModel = ModelRDB.createModel(con, dbmodelname);
        System.out.println("Base de datos '" + dbname + "' preparada.");

        //Cierre del modelo y la conexión
        RDBModel.close();
        con.close();

        //La base de datos está lista
        created = true;
    }
    catch (Exception exc)
    {
        System.out.println("Se ha producido una excepción controlada.");
        exc.printStackTrace();
    }

    return created;
}
```

Fuente: Marcos Blanco Galán,

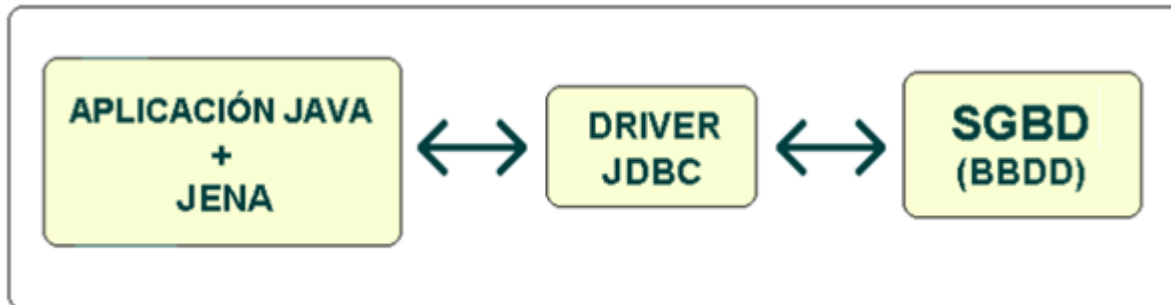
<http://marcosblanco.blogspot.com/2008/04/almacenamiento-persistente-de-ontologas.html>.

Explicando un poco la figura anterior, es un método que permite generar un esquema para el modelo de persistencia. Este método recibe como parámetros los datos de conexión de la base de datos, este proceso se hace a través de la clase

ModelRDB, una implementación de la API RDF (*Resource Description Framework*) que cuenta con la capacidad de gestionar el almacenamiento recuperación de datos en la base de datos.

A través de estos driver se pueden acceder a los datos de las ontologías guardadas en los almacenes persistentes, esto mediante el uso de los driver en este caso JDBC el proceso podemos verlo en la siguiente imagen.

Figura 15: Carga del driver JDBC para MySQL al inicio de una aplicación que trabaje con conexiones.



Fuente: Marcos Blanco Galán,

<http://marcosblanco.blogspot.com/2008/04/almacenamiento-persistente-de-ontologas.html>.

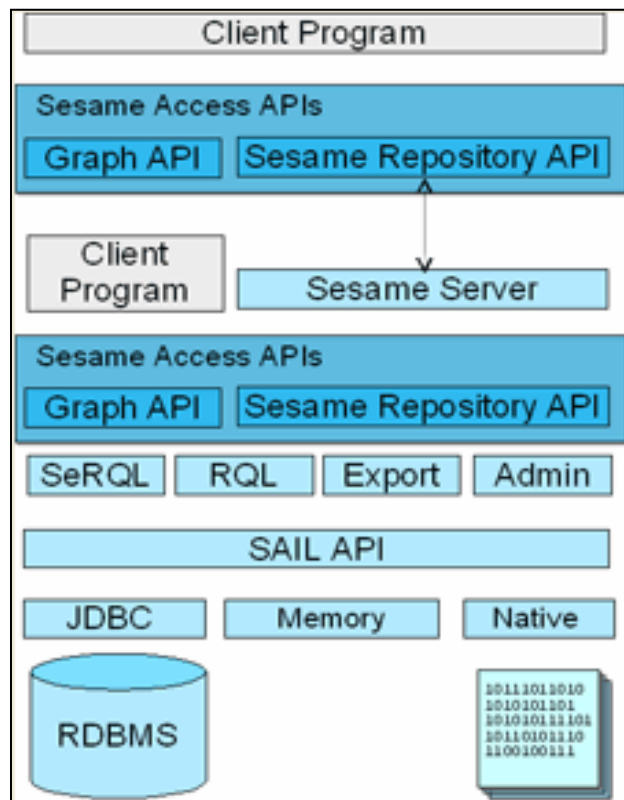
Formas de trabajar con ontologías en SESAME.

Otra tecnología para el manejo de la web semántica y creación de aplicaciones de esta naturaleza es SESAME. SESAME es una arquitectura genérica, de código abierto, que brinda la posibilidad de persistir datos y esquemas RDF y a la vez su consulta en línea e inferencia sobre la información almacenada. Los lenguajes de consulta pueden ser: RQL, RDQL y SeRQL. Actualmente también es compatible con el lenguaje de consultas SPARQL.

SESAME se ha diseñado de una manera flexible permite el uso de diferentes sistemas de almacenamiento de datos (en memoria, bases de datos relacionales, ficheros, indexaciones), y ofrece una gran escala de herramientas para los desarrolladores que les permite aprovechar el uso potencial de los RDF basado en los estándares de esta tecnología. SESAME ofrece acceso remoto transparente a los repositorios RDF utilizando la API que es exactamente la misma para el acceso local. Por último, SESAME soporta todos los principales formatos de archivos de flujo de RDF, incluyendo RDF / XML, *Turtle*, N triples-, Trig y Trix.

Arquitectura de SESAME.

Figura 16: Arquitectura SESAME



Fuente: Susana Corbera Delgado, ALMACENAMIENTO, CONSULTA Y RAZONAMIENTO: SESAME Y JENA. Madrid, p. 7.

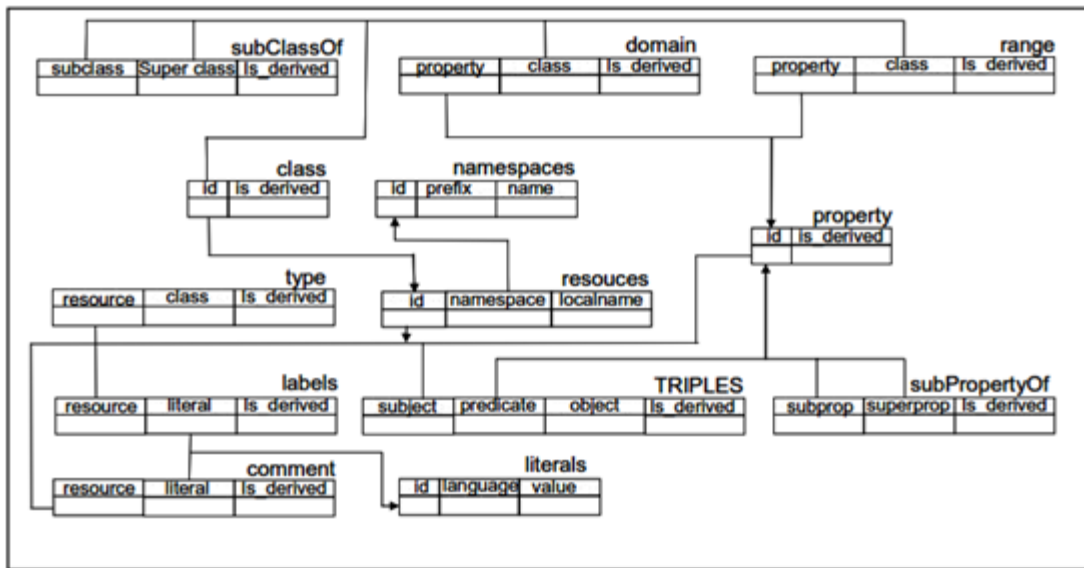
Persistencia de datos en SESAME.

Para el almacenamiento persistente de los datos, SESAME está diseñado para trabajar con un repositorio escalable. Además para que sea independiente de los DBMS (*Data Base Manager System*) usados (Oracle, PostgreSQL y MySQL) en la administración del repositorio, todo el código que hace referencia a un DBMS específico, fue concentrado en una única capa de la arquitectura SESAME: la capa SAIL (*Storage and Interface Layers*).

La capa SAIL es una API que ofrece métodos específicos RDF a sus clientes y los traduce en invocaciones a distintos métodos de los DBMS apropiados. Cada DBMS tiene su propio dialecto SQL, por ejemplo: El tipo carácter en Oracle se define como `nvarchar (2)` y en MySQL como `varchar(n)`. Esto puede significar que para cada DBMS tengamos que cambiar las sentencias SQL (o las consultas realizadas con cualquiera de los lenguajes RDF actuales). Una ventaja importante de contar con dicha capa separada (SAIL), es la posibilidad de implementar SESAME en una gran variedad de repositorios en distintos DBMS [18].

Aquí podemos observar cómo sería una estructura de una persistencia en MySQL.

Figura 17: Esquema de persistencia SESAME en MySQL



Fuente: Maria I. Lamas C: proyecto. Lenguajes de consulta para documentos rdf. Cataluña, 2006, p. 60.

SESAME vs Jena.

Para hacer un cuadro comparativo entre estos dos grandes *frameworks*, los temas más sensibles que se deben tener en cuenta es la eficiencia, que se mide con el tiempo de ejecución de una consulta en los lenguajes de consultas que estos soportan, la estabilidad que estos nos brindan al momento de crear las aplicaciones y por ultimo se debe tener en cuenta las estructuras de persistencias que estos presentan, más claramente los esquema en las bases de datos relacionales que estos utilizan. En la siguiente tabla se muestra las principales diferencias encontradas.

Tabla 3: Cuadro comparativos SESAME vs Jena

Características	SESAME	JENA
Query Language	RQL, RDQL, SPARQL	RDQL, SPARQL
Implementación	Java	Java
Almacenamiento BD	Mejor MySQL que PostgreSQL y oracle pero permite los tres	MySql, oracle, PostgreSQL
soporta actualización (ontología y datos)	Si	Si
Soporte API (querying+updating)	HTTP/SOAP	Java
Escalabilidad	Si	Si
Formato de exportacion de datos	RDF	Tripletas en ASCII
Soportan Inferencia	Si	Si
Operaciones	Union, Interseccion, Diferencia	Union, Interseccion, Diferencia

Fuente: Maria I. Lamas C: proyecto. Lenguajes de consulta para documentos rdf. Cataluña, 2006, p. 64.

Como se detallo anteriormente, para la construcción de aplicaciones semánticas se investigaron cuales eran las diferentes implementaciones con la que contaba cada uno de los lenguajes de programación actuales y se encontró una lista significativa y finalmente se abordó la del lenguaje Java, que es el caso de Jena y SESAME, de los cuales se decidió optar por Jena debido que este brindaba lo que en un principio se quería. Era sencillo y fácil de entender aunque SESAME también pero, Jena tiene un poco mas de información disponible y más casos de uso. Ambos contaban con características similares pero interpretadas de diferentes maneras para cada uno de los artefactos para el manejo de las ontologías, aunque SESAME trae consigo un mecanismo para la implementación de servicios web semánticos, con el cual Jena no cuenta pero no siendo una de las características que se necesitaban en la aplicación a construir, se decidió

escoger a Jena como herramienta base para la construcción de la aplicación debido a su fácil configuración y compatibilidad con Empire un *framework* que se abarcara en el capítulo siguiente y otras herramientas que se probaron en el camino de la construcción de la aplicación.

4. CREACION DE SOCIAL UTB

4.1. METODOLOGIA PARA LA CREACION DE SOCIAL UTB

Como se menciona en el capítulo número dos, no existe un estándar definido para la realización de una red social, mas que todo se implementan patrones de diseños que son muy comunes al momento de desarrollar cualquier aplicación web y mucho de estos patrones ya son implementados en las diferentes herramientas mencionadas en el capítulo anterior.

Dichas herramientas ofrecen muchas posibilidades para la creación de una red social convencional sobre todo ELGG y NING que brindan la posibilidad de tener una red en menos de treinta minutos y sin necesidad de ser un experto en programación. Desafortunadamente SocialUTB no es una red social convencional, es una red social semántica, donde el modelo de datos a implementar no es posible trabajarlo con el típico modelo de datos entidad relación que ofrecen las diferentes herramientas.

El API ofrecida por OPEN SOCIAL es una de las opciones mas llamativas porque esta busca ofrecer un estándar para la creación de aplicaciones sociales, es decir que si usted desarrolla una aplicación basada en OPEN SOCIAL, esta aplicación servirá sobre cualquier red social que este implementada mediante esta API social. Es un buen proyecto sobre todo si es una red abierta ya que se podría sacar provecho de las diversas aplicaciones que están creadas con OPEN SOCIAL, pero SocialUTB no es una red abierta, es una red académica con un dominio estrictamente definido el cual es los egresados de la Universidad Tecnológica de Bolívar; lo que obliga a tomar otro camino diferente a los conocidos.

Si no existe una metodología definida para la creación de una red social, mucho menos existe un estándar para la creación de redes sociales semánticas pero si

se conocen cuales son los pilares que debe tener una red social y cuales son los *frameworks* utilizados para el desarrollo de aplicaciones semánticas, es por eso que se decidió utilizar una metodología RUP combinada con ingeniería de software basada en componentes y de esa forma llevar el diseño de SocialUTB como un proceso de desarrollo similar a cualquier software empresarial pero con algunas variantes que se hacen necesarias para satisfacer una aplicación semántica. Estas variantes serán mencionadas posteriormente.

4.2. DESCRIPCION DE REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

En este apartado se realiza la especificación de requerimientos de SocialUTB. Describiendo tanto los requerimientos funcionales como los no funcionales.

4.2.1. Definiciones, acrónimos y abreviaturas.

- SRE-RF001: Requerimiento funcional Crear grupo de interés.
- SRE-RF002: Requerimiento funcional Editar grupo de interés.
- SRE-RF003: Requerimiento funcional Eliminar grupo de interés.
- SRE-RF004: Requerimiento funcional Invitar Usuario al grupo de interés.
- SRE-RF005: Requerimiento funcional Eliminar Usuario del grupo de interés.
- SRE-RF006: Requerimiento funcional Realizar publicaciones en el grupo de interés.
- SRE-RF007: Requerimiento funcional Eliminar publicaciones en el grupo de interés.
- SRE-RF008: Requerimiento funcional Crear Publicaciones.
- SRE-RF009: Requerimiento funcional Eliminar Publicaciones.
- SRE-RF010: Requerimiento funcional Comentar Publicaciones.

- SRE-RF011: Requerimiento funcional Crear mensaje.
- SRE-RF012: Requerimiento funcional Guardar mensaje.
- SRE-RF013: Requerimiento funcional Enviar mensaje.
- SRE-RF014: Requerimiento funcional Crear solicitud de amistad.
- SRE-RF015: Requerimiento funcional Enviar solicitud de amistad.
- SRE-RF016: Requerimiento funcional Aceptar solicitud de amistad.
- SRE-RF017: Requerimiento funcional Rechazar solicitud de amistad.
- SRE-RF018: Requerimiento funcional Ver información de usuario.
- SRE-RF019: Requerimiento funcional Ver información de usuarios con relación.
- SRE-RF020: Requerimiento funcional Realizar notificaciones.
- SRE-RF021: Requerimiento funcional Crear perfil de usuario.
- SRE-RF022: Requerimiento funcional Editar perfil de usuario.
- SRE-RF023: Requerimiento funcional Eliminar perfil de usuario.
- SRE-RF024: Requerimiento funcional Añadir Foto al perfil de usuario.
- SRE-RF025: Requerimiento funcional Enviar noticias a usuarios.
- SRE-RF026: Requerimiento funcional Crear Evento.
- SRE-RF027: Requerimiento funcional Cancelar Evento.
- SRE-RF028: Requerimiento funcional Editar Evento.
- SRE-RF029: Requerimiento funcional Eliminar Evento.
- SRE-RF030: Requerimiento funcional Añadir foto al Evento.
- SRE-RF031: Requerimiento funcional Invitar Usuarios al Evento.
- SRE-RF032: Requerimiento funcional Notificar cumpleaños.
- SRE-RF033: Requerimiento funcional Login.
- SRE-RF034: Requerimiento funcional mostrar Grupos de interés creados por usuario.
- SRE-RF035: Requerimiento funcional mostrar Grupos de interés en los que está vinculado el egresado.
- SRE-RF036: Requerimiento funcional mostrar eventos creados por usuario.

- SRE-RF037: Requerimiento funcional mostrar eventos en los que está vinculado el usuario.
- SRE-RF038: Requerimiento funcional Crear Evento Académico.
- SRE-RF039: Requerimiento funcional Editar Evento Académico.
- SRE-RF040: Requerimiento funcional Eliminar Evento Académico.
- SRE-RF041: Requerimiento funcional Añadir foto al Evento Académico.
- SRE-RF042: Requerimiento funcional Invitar Usuarios al Evento Académico.
- SRE-RF043: Requerimiento funcional mostrar eventos académicos creados por usuario.
- SRE-RF044: Requerimiento funcional mostrar eventos académicos en los que está vinculado el usuario.

4.2.2. Especificaciones funcionales.

Las especificaciones funcionales se pueden consultar en la sección de anexos. (Ver Anexos H-AX).

4.2.3. Especificaciones suplementarias.

Usabilidad.

Capacidad para ser entendido.

- El sistema debe contar con documentación actualizada disponible de forma inmediata.
- Deben existir guías de ayudas para el uso o entendimiento de operaciones o términos.
- Mensajes de error o de éxito, mostrados en forma clara y precisa, indicando que se debe hacer al respecto.
- Se debe simplificar la generación de informes.

- Debe permitir la navegación intuitiva.
- Debe cargarse rápidamente, el sistema debe cargar la información y mostrarla simultáneamente.

Cumplimiento de la usabilidad.

- Los títulos de las interfaces deben ser concisos y apropiados con la funcionalidad de la misma.
- El sistema debe proveer mecanismos de recuperación de claves de acceso a los usuarios y administradores.

Confiabilidad.

Tolerancia a fallas.

- El sistema debe estar disponible los 365 días del año, las 24 horas del día.
- La información no debe verse afectada por el uso o fallas ocurridas en el sistema.
- La información que actualice una función fluye automáticamente a todos los registros de las funciones relacionadas.
- Debe existir mayor precisión en la información porque se garantiza la eliminación de la redundancia de los datos.

Capacidad de recuperación.

- El sistema debe ser capaz de responder adecuadamente a la introducción de datos no válidos, o a cualquier uso no previsto en las especificaciones funcionales (sistema robusto).
- Según la complejidad de la falla los errores en el sistema deben solucionarse en cortos periodos de tiempo.

Desempeño.

Tiempo de respuesta.

- El tiempo de respuesta promedio para operaciones de actualización y consulta no debe superar los 5 segundos.

Escalabilidad.

- El sistema debe adaptarse al crecimiento del número de usuarios en la institución y al cambio de plataformas tecnológicas.

Mantenibilidad.

Capacidad para ser analizado.

- El código fuente debe estar documentado en aquellas secciones que por su complejidad así lo requieran.
- El código fuente debe estar separado y organizado.
- El sistema debe contar con la debida documentación a nivel de arquitectura, que permita la rápida ubicación de un componente, módulo o programa.

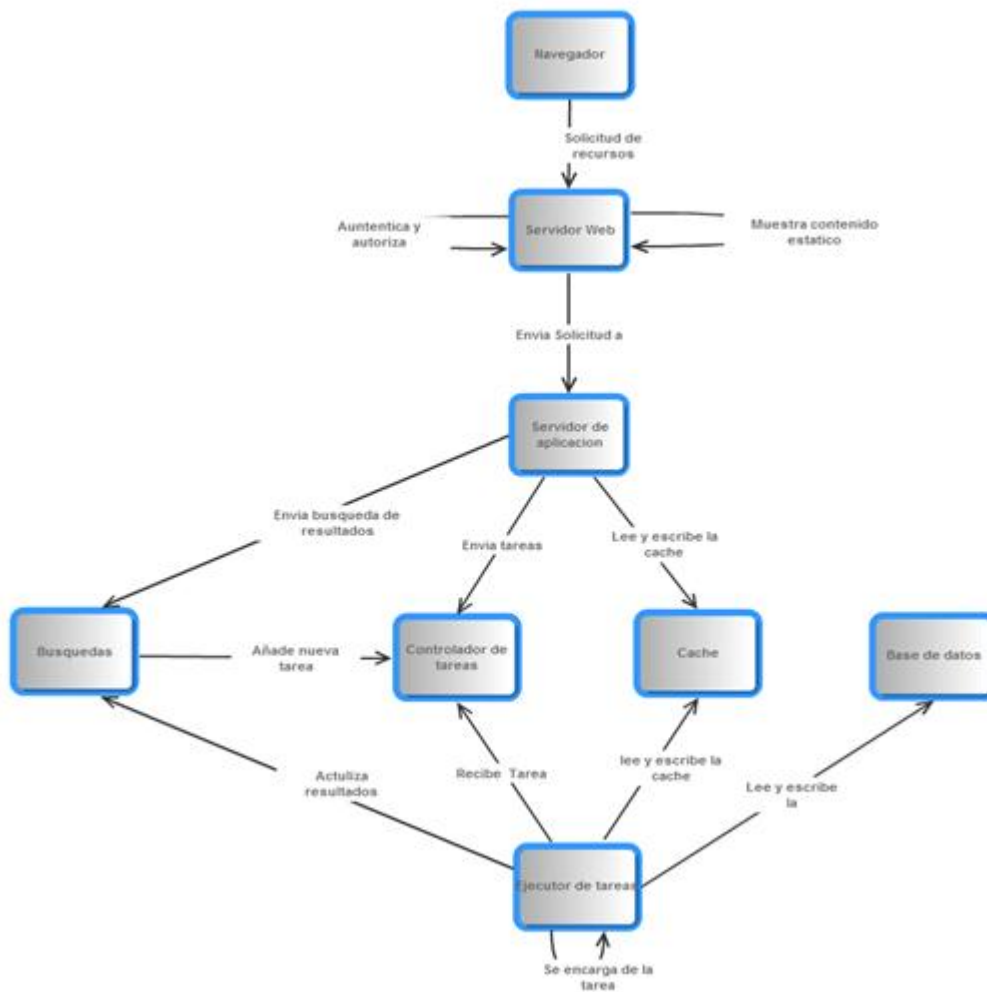
Capacidad para ser cambiado.

- El código fuente debe contener datos acerca del autor, fecha, función, historial de modificaciones, entre otros. Así mismo, el patrón de diseño debe sugerir o indicar como extender la aplicación para añadirle nuevas funcionalidades.

4.3. DESARROLLO DEL MODELO CONCEPTUAL

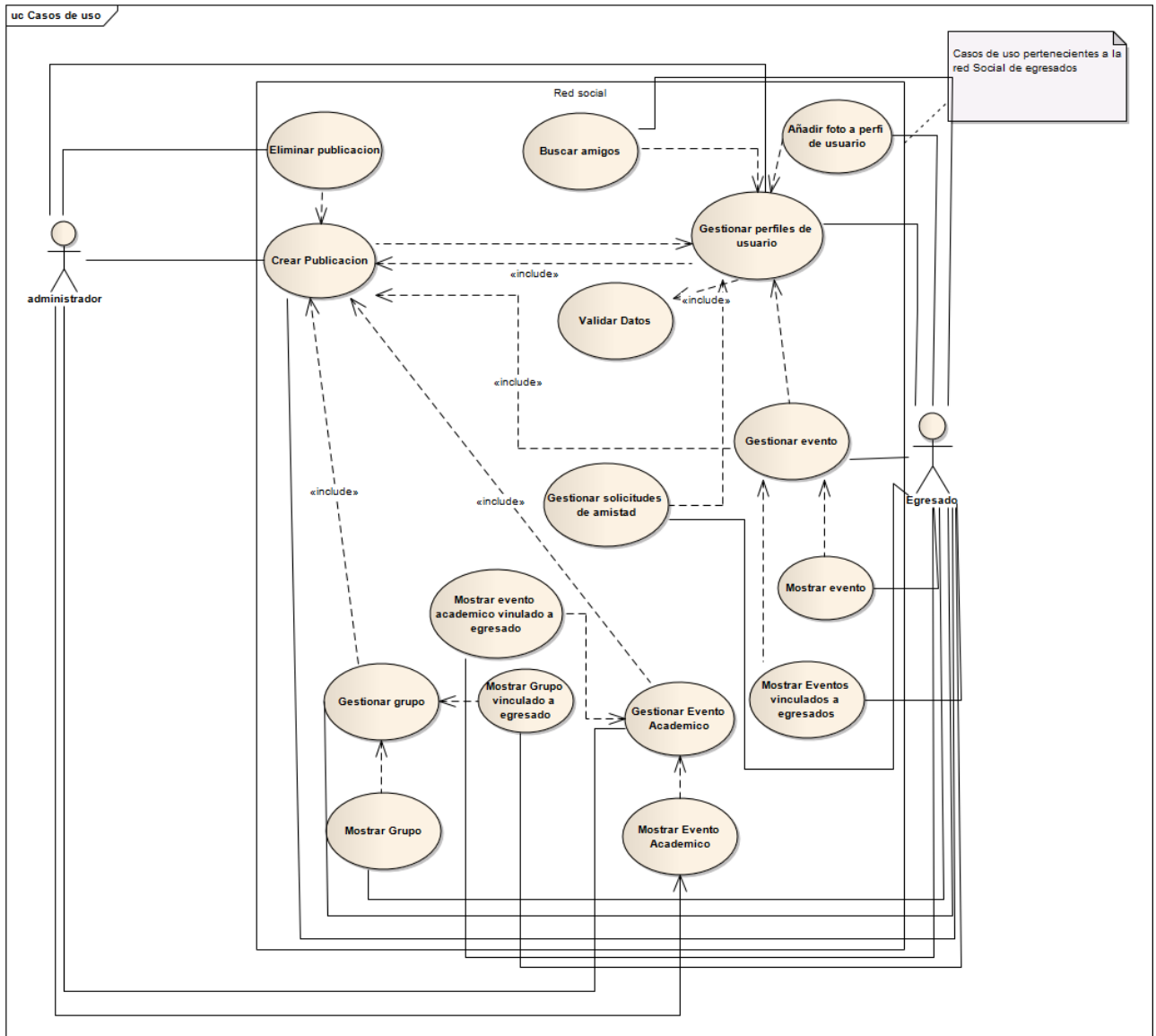
Conociendo bien cuales son los requerimientos funcionales y no funcionales se puede hacer una idea clara de lo que busca SocialUTB.

Figura 18: Modelo Conceptual de SocialUTB.



4.4. DESARROLLO DE CASOS DE USO

Figura 19: Diagrama general de casos de uso de SocialUTB



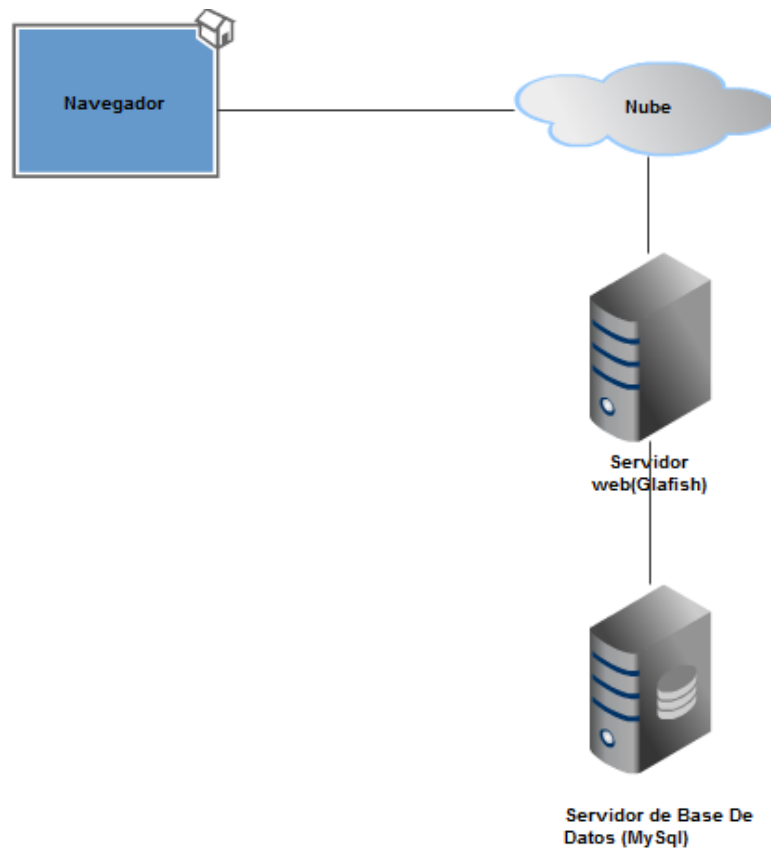
Para ver en detalle los diferentes diagramas de casos de uso acorde a los requerimientos mencionados anteriormente, ver la sección de anexos, desde el anexo AY, hasta el anexo BZ. (Ver Anexos AY-BZ).

4.5. DESARROLLO DE LAS ACTIVIDADES

Los diagramas de actividad son típicamente usados para modelar los procesos de negocio, o para modelar la lógica capturada por un caso de uso o escenario de uso. Para ver en detalle los diferentes diagramas de actividades de SocialUTB, ver la sección de anexos, desde el anexo CA, hasta el anexo CP. (Ver Anexos CA-CP).

4.6. ARQUITECTURA DE SOCIAL UTB

Figura 20: Arquitectura física de SocialUTB



En la figura veinte se muestra la arquitectura física utilizada para la red social. Es una arquitectura muy sencilla que consta de un servidor MYSQL donde estará alojada la base de datos (la base de datos contiene un modelo SDB), un servidor web que contendrá la aplicación, la cual se realizó utilizando tecnología J2EE (JSF), que será visualizada a través de un navegador.

4.8. IMPLEMENTACION DE SOCIAL UTB

Como se observa, el diagrama de clases es una extensión del diagrama de conceptos utilizado al momento de la creación de la ontología porque este solo contempla la figura del egresado más no de la red social.

Teniendo esto claro y observando las tecnologías existentes se decidió utilizar Jena para realizar un buen manejo de la ontología creada mediante Java.

El momento de implementación de la aplicación semántica ha sido uno de los puntos más críticos y con más tropiezos que ha tenido la investigación debido a que no toda la información obtenida servía cuando se ponía a prueba.

Jena ofrece un mundo de posibilidades para el trabajo con ontologías pero sigue existiendo una brecha grande entre el paradigma orientado a objetos y el desarrollo basado en ontologías, ya que la programación de estas suele ser muy compleja a medida que estas aumentan su tamaño y como se vio en el capítulo de aplicaciones semánticas, no es cuestión de crear la ontología solamente, sino también persistirla para que pueda perdurar en el tiempo.

Durante esta etapa de investigación surge una herramienta llamada Empire que precisamente busca cerrar esa brecha entre las ontologías y la programación orientada objetos facilitando así que el desarrollo de SocialUTB se pueda realizar desde un enfoque orientado a objeto como se planteo al momento del diseño y el cual muy familiar para la inmensa mayoría de programadores.

A continuación se realiza una descripción acerca de que se trata Empire y posteriormente como fue la experiencia al momento de implementar SocialUTB usando esta herramienta.

4.8.1. Empire.

Empire es una implementación de JPA (Java Persistence API) para RDF y la web semántica que como se mencionó anteriormente facilita a los desarrolladores que están familiarizados con JPA introducirse en el mundo de las ontologías y la web semántica.

Empire lo que hace es extender las anotaciones JPA con nuevas anotaciones para poder brindar soporte a datos RDF. Con Empire se pueden trabajar diferentes almacenes de datos entre los que se encuentran SESAME y Jena. A continuación se muestra un ejemplo de una clase utilizando Empire.

Figura 22: Ejemplo de uso de Empire

```
1 @Entity
2 @Namespaces({"foaf", "http://xmlns.com/foaf/0.1/"})
3 @RdfsClass("foaf:Person")
4 public class Book implements SupportsRdfId {
5
6     @RdfProperty("foaf:name")
7     private String name;
8
9     @RdfProperty("foaf:friend")
10    @OneToMany(fetch = FetchType.LAZY,
11              cascade = {CascadeType.PERSIST, CascadeType.
12                        MERGE})
13    private Collection<Person> friends = new HashSet<Person
14    > ();
15 }
```

4.8.2. Empire y SocialUTB.

Ya conociendo un poco de la filosofía de Empire se explica como fue la experiencia con este framework.

A pesar de tener una comunidad bastante activa por ser una herramienta nueva, todavía se mantenía inestable lo que hizo necesaria la modificación del framework para que funcionara de acuerdo a las necesidades deseadas.

Lo primero fue modificar la clase JenaConfig ubicada dentro de la librería Empire-Jena como se muestra en la figura veintitrés.

Figura 23: Modificación de la clase JenaConfig

```
package com.clarkparsia.empire.jena;

public interface JenaConfig extends com.clarkparsia.empire.config.ConfigKeys {

    public enum JenaModelType {

        SDB;

        private JenaModelType() {
            //compiled code
            throw new RuntimeException("Compiled Code");
        }

        public static final String SDB_FILE_URL = "c:\\configuracion\\sdb.ttl";
        public static final String TYPE = "type";
        public static final String STREAM = "stream";
        public static final String FORMAT = "format";
        public static final String FILES = "files";
        public static final String BASE = "base-uri";
        public static final String MODEL = "model";
    }
}
```

Con esta modificación se pasa la ruta para que el archivo de configuración en este caso llamado sdb.ttl sea encontrado directamente y no tenga ningún inconveniente a la hora de buscar la configuración, que fue uno de los principales inconvenientes al momento de la implementación.

En la figura veinticuatro se puede observar como es el archivo de configuración.

Figura 24: Archivo de configuración SDB.TTL

```
@prefix sdb: <http://jena.hpl.hp.com/2007/sdb#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ja: <http://jena.hpl.hp.com/2005/11/Assembler#> .
# MySQL - InnoDB
<#store> rdf:type sdb:Store;
  sdb:layout "layout2";
  sdb:connection <#conn>;
  sdb:engine "InnoDB"; # MySQL specific
.
<#conn> rdf:type sdb:SDBConnection ;
  sdb:sdbType "MySQL"; # Needed for JDBC URL
  sdb:sdbHost "localhost" ;
  sdb:sdbName "sdb2";
  sdb:sdbUser "root";
  sdb:sdbPassword "";
  sdb:driver "com.mysql.jdbc.Driver" ;
.
```

Según Empire, con solo tener este archivo de configuración al momento de crear un entity manager como se observa en la figura veinticinco automáticamente se crearía el modelo SDB, pero no fue así. Se hizo necesaria la creación de un modelo SDB utilizando código Java como se observa en la figura veinticinco.

Figura 25: Creación del modelo SDB

```
import com.hp.hpl.jena.rdf.model.Model;
import com.hp.hpl.jena.rdf.model.Statement;
import com.hp.hpl.jena.rdf.model.StmtIterator;
import com.hp.hpl.jena.sdb.SDBFactory;
import com.hp.hpl.jena.sdb.Store;
import com.hp.hpl.jena.sdb.store.StoreFactory;

/** Example of use with the Jena API for models. */

public class ExModelSDB
{
    static public void main(String...argv)
    {
        Store store = StoreFactory.create("sdb.ttl") ;
        store.getTableFormatter().create() ;
        Model model = SDBFactory.connectDefaultModel(store) ;

        StmtIterator sIter = model.listStatements() ;
        for ( ; sIter.hasNext() ; )
        {
            Statement stmt = sIter.nextStatement() ;
            System.out.println(stmt) ;
        }
        sIter.close() ;
        store.close() ;
    }
}
```

Teniendo la base de datos SDB, se hizo necesario crear el modelo y persistirlo. Como se aprecia en la figura veintiséis.

Figura 26: Persistencia mediante Empire

```
PersistenceProvider aProvider = Empire.get().persistenceProvider();  
aManager = Persistence.createEntityManagerFactory("sdb2").createEntityManager();
```

Para esto se mapearon los conceptos de la ontología desde Protégé y se adaptaron las relaciones a las anotaciones de Empire. Es en este punto donde se aprecia uno de los mayores alcances que ofrece el framework, el cual es añadir la funcionalidad de los conceptos creados previamente en una ontología.

5. CONCLUSIONES

A veces tener una idea brillante es solo una pequeña parte de lo que implica crear algo realmente revolucionario e innovador. La idea de crear una red social semántica, cuyo núcleo fuese una ontología que tuviese como dominio los estudiantes egresados de la Universidad Tecnológica de Bolívar, y cuya finalidad fuese la de crear una relación mas estrecha y productiva entre la universidad y sus egresados aprovechando todos beneficios antes mencionados que aporta la web semántica; fue algo sumamente sencillo. Pero, desarrollar esa idea y crear algo tangible, algo que se pudiese usar, fue el verdadero reto.

La gran dificultad a la hora de realizar la investigación fue que mucha de la información que existe al respecto del tema abordado, es solo teoría o información muy general de uno de los tantos tópicos de la web semántica, nada muy específico. No existe una guía estandarizada de cómo crear aplicaciones semánticas y muchos menos ejemplos prácticos.

El merito de la investigación reside en haber labrado un camino para el desarrollo de este tipo de aplicaciones, en haber dejado una guía, unos pasos, un ejemplo practico, de cómo se puede abordar el desarrollo de una aplicación web semántica. Es muy posible que no todo lo que hace parte de esta investigación sea aplicable para todas las aplicaciones web semánticas, pero al menos le dará una idea al lector de lo que hay que tener en cuenta a la hora de desarrollar dichas aplicaciones.

La investigación se centra en un dominio cerrado y específico, pero esto no quiere decir que lo aplicado en ella, no se pueda aplicar en otro dominio o en otro campo. La metodología usada para la creación de la ontología puede ser aplicada para cualquier otra ontología que se desee crear. Los pilares expuestos en la construcción de una red social, pueden servir para cualquier otra aplicación social que se desee construir. Y las herramientas y los procesos de desarrollo utilizados para la elaboración de la aplicación web semántica pueden ser usados para

cualquier otra aplicación web semántica que se desee crear. Las herramientas están dadas, lo único que falta es una idea y mucho esfuerzo para sacarla adelante.

6. RECOMENDACIONES

La primera etapa, la cual fue la construcción de la ontología que permitió modelar una red social de egresados de la Universidad Tecnológica de Bolívar. Se estudiaron los diversos conceptos que se necesitaban para entrar en el mundo de la semántica, el cual era totalmente desconocido en el inicio de la investigación.

Mucho del conocimiento adquirido fue gracias al método de prueba y error. Por eso es de suma importancia que antes de codificar la ontología en cualquiera que vaya ser el lenguaje elegido para ello, hacer un buen diseño de esta. Es esencial definir y tener muy claro el dominio de la ontología, sus conceptos, relaciones, atributos e individuos. Esto hará que se ahorren muchos problemas que puedan surgir en un futuro por un mal diseño.

Es muy posible que el primer diseño no sea el adecuado, por ende es muy recomendable que se analice en busca de errores que pueda contener y de esta forma, ir perfeccionando la ontología. Una vez se esté seguro del diseño, es importante elegir la herramienta adecuada con la cual se va a crear la ontología. Hay que familiarizarse con ella, conocer sus virtudes y sus limitantes, ver cuál es la versión más óptima para lo que se desea hacer. Elegir la herramienta incorrecta, puede significar muchas horas desperdiciadas.

Se llegó a la conclusión de que para construir una aplicación web 3.0 no era necesario inventar nuevas metodologías, ya que era más fácil emplear las ya existentes y ver si eran capaces de amoldarse a esta. Esto era lo que comentaban en algunas de las literaturas consultas, ya que la curva de aprendizaje para la construcción de estos programas orientados a lo semántico no sería tan traumática.

El contacto con los estudiantes investigadores de la Universidad de San Carlos de Guatemala fue crucial para el desarrollo de la aplicación web semántica, ellos habían abordado algo de la temática y de las tecnologías que posiblemente se

utilizarían. Se hicieron un par de preguntas, ellos respondieron y esto ayudo definir un poco más el rumbo que se debería tomar. Luego vino la hora de elegir las herramientas a usar para la construcción de la aplicación, la elección de Empire no fue algo fácil, debido a que SESAME, el cual es su competidor mas fuerte, también cumplía con muchas de las características se buscaban. Pero Empire, esta permitía ser estudiado y analizado con mas facilidad y además, contaba con una comunidad un poco amplia, aunque esta herramienta en sus primeras versiones era un poco inestable.

Pero el punto decisivo que hizo a Empire ser el ganador es que brindaba una amplia gama de utilidades, como era el caso de una implementación de JPA para RDF, que es uno de los retos que se tiene al momento de implementar una aplicación semántica. Lo importante a tener en cuenta a la hora de realizar una aplicación web semántica, es dominar el lenguaje en el cual se piensa programar, ya que herramientas para la creación de aplicaciones semánticas existen en todos los lenguajes, así que lo verdaderamente importante es el manejo que se tenga de este.

BIBLIOGRAFIA

- [1] SEGURA MAROTO, Aurelio. Desarrollo de una aplicación Web 2.0 para la captura y análisis de la valoración del usuario. Valencia, 2010. p. 56. Proyecto final de Carrera (Ingeniero Informático). Escuela Técnica Superior de Ingeniería Informática. Facultad de Ingeniería. Programa de informática.
- [2] GRUBER, T. R. A translation approach to portable ontology specifications. Knowledge Acquisition, 1993 p. 199-220.
- [3] Studer, R., Benjamins, V. R., & Fensel, D. Knowledge engineering: Principles and methods. IEEE Transactions on Data and Knowledge Engineering, 1998. p. 161-197.
- [4] Stuckenschmidt, Heiner; Van Harmelen, Frank. Information Sharing on the Semantic Web. Springer, 2003. p. 252.
- [5] RIVEROS, Roberto. Web semántica: las ontologías. 2008, <http://robertoriveros.blogspot.com/2008/11/web-semntica-las-ontologas.html>
- [6] H. Sofia Pinto, Christoph Tempich, Steen Staab, and York Sure. Diligent: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. In Ramon Lopez de Mantaras and Lorenza Saitta, editors, Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004), August 22nd - 27th, p. 393–397, Valencia, Spain, AUG 2004. IOS Press.
- [7] Fensel Dieter, Van Harmelen Frank, Klein Michel, Akkermans Hans. On-To Knowledge: Ontology-based Tools for Knowledge Management. p. 8.
- [8] M. C. Suárez-Figueroa, K. Dellschaft, E. Montiel-Ponsoda, B. Villazón-Terrazas, Z. Yufei, G. Aguado de Cea, A. García, M. Fernández-López, A. Gómez-Pérez, M. Espinoza, M. Sabou. NeOn Deliverable D5.4.1. NeOn Methodology for Building Contextualized Ontology Networks. NeOn Project. <http://www.neon-project.org>. February 2008.
- [9] Fernández, Mariano, Gómez-Pérez Asunción, Juristo Natalia. METHONTOLOGY: From ontological art towards ontological engineering. p. 8.
- [10] Tenzer Simón; Ferro Olga; Palacios Nuria. Redes Sociales Virtuales: personas, sociedad y empresa, 2009, p 4. <http://www.ccee.edu.uy/ensenian/catcomp/material/RSVa.pdf>
- [11] ZAMORA, Marcelo. Redes Sociales en Internet, 2006, <http://www.maestrosdelweb.com/editorial/redessociales/>
- [12] The Social Network <http://ataraxiasocial.blogspot.com/2011/03/la-red-social-social-network-y-la.html>

- [13] Elgg site. (2010, Febrero 1). Elgg Data Model. Developer documentation.
<http://docs.elgg.org/wiki/Engine/DataModel>
- [14] Good, R. (2007). Open Social: Google's New Social Networking Platform - What Is It and Why It Matters. Robin Good - MasterNewMedia.
http://www.masternewmedia.org/social_networking/social-networkingplatforms/Open-Social-Google-social-networking-platform-what-is-it-20071102.htm
- [15] PRIETO RODRÍGUEZ, Juan A. Aplicaciones Industriales de la Web Semántica. España. Madrid. 2008. p. 6-11.
- [16] Hebler John, Fisher Matthew, Blace Ryan, Perez-Lopez Andrew, Dean Mike. Semantic Web Programming, Part III, p. 263.
- [17] HEBELER, John. Semantic Web Programming. 1 ed. EU: Wiley, 2009. Parte III Capitulo 8, p. 267- 270.
- [18] LAMAS C, Maria I. Lenguajes de consulta para documentos rdf. Cataluña, 2006, p. 58-64, Trabajo de grado (Ingeniería Informática). Universidad Abierta de Cataluña. Facultad de Informática.
<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/640/1/38449tfc.pdf>
- [19] BLANCO GALÁN, Marcos. Almacenamiento persistente de ontologías con Jena y MySQL.
<http://marcosblanco.blogspot.com/2008/04/almacenamiento-persistente-de-ontologas.htm>.
- [20] PASQUIER, Mauricio. Jena Semantic Web Framework y SPARQL. Argentina. 2010. p. 5.

ANEXOS

Anexo A. Glosario de términos

Nombre	Sinónimos	Acrónimos	Descripción	Tipo
Conceptos				
Social UTB			Estructura social virtual de relaciones entre egresados de la Universidad Tecnológica de Bolívar	Concepto
Egresado	Graduado		Persona que se graduó, ya sea de pregrado o posgrado de la Universidad Tecnológica de Bolívar	Concepto
No Egresado			Persona que no se graduó de la Universidad Tecnológica de Bolívar, pero tiene alguna relación familiar con uno o más egresados	Concepto
Lugar	Sitio, parte, zona		Punto en específico dentro del planeta tierra	Concepto
Grupo			Grupo de egresados reunidos por un interés, característica o una situación en común. Dichos egresados no tienen por qué tener algún tipo de relación establecida	Concepto
Evento	Suceso, acontecimiento		Suceso de importancia social o personal, el cual ocurre en un momento y lugar definido y cual están asociados uno o más egresados	Concepto
Pregrado			Carrera que se escoge para	Concepto

			prepararse profesionalmente para recibir el título de grado.	
Posgrado	Postgrado		Ciclo de estudios de especialización posterior a la graduación o licenciatura	Concepto
Evento Académico			Conjunto de congresos, charlas, seminarios, cursos, capacitaciones, etc. asociadas a uno o más egresados	Concepto
Idioma	Lengua, lenguaje, habla, dialecto		Sistema de comunicación verbal o no verbal que maneje un egresado	Concepto
Cargos	Puesto		Cargos desempeñados por un egresado durante su vida laboral	Concepto
Proyecto			Conjunto de investigaciones y proyectos realizados por un egresado	Concepto
Foto			Imágenes digitales ligadas a un egresado	Concepto
Organización			Asociación de personas reguladas por un conjunto de normas en función de determinados fines	Concepto
Atributos De Instancia				
Egresado				
Nombres			Primer y segundo nombre (si se tiene) de un egresado o de un no egresado	Atributo de Instancia
Apellidos			Apellido o los apellidos de un egresado o de un no egresado	Atributo de Instancia
Género	Sexo		Género masculino o femenino de un egresado o de un no egresado	Atributo de Instancia
Fecha De Nacimiento			Año, mes y día que consta en el registro civil como fecha de nacimiento de un egresado o de un no egresado	Atributo de Instancia
Correo Electrónico	E-mail, mail		Dirección de correo electrónico	Atributo de Instancia

			principal de un egresado o de un no egresado	
Código	ID		Conjunto de caracteres que identifican exacta y únicamente a un egresado dentro del sistema	Atributo de Instancia
Contraseña	Password, clave		Conjunto de caracteres que junto con el código, le permiten a un egresado acceder al sistema	Atributo de Instancia
Tipo De Documento De Identidad			Clase o tipo de documento que utiliza un egresado para identificarse ante la sociedad	Atributo de Instancia
Documento De Identidad			Documento emitido por una autoridad administrativa competente para permitir la identificación personal de los ciudadanos	Atributo de Instancia
Número Telefónico	Teléfono		Número telefónico principal de un egresado	Atributo de Instancia
Sitio Web	Página web		Ubicación virtual de un egresado, en la cual se puede encontrar información al respecto de este	Atributo de Instancia
Estado Civil			Situación de un egresado, determinada por sus relaciones de familia, provenientes del matrimonio o del parentesco, que establece ciertos derechos y deberes	Atributo de Instancia
Estado Laboral			Situación laboral de un egresado	Atributo de Instancia
Perfil Laboral			Conjunto de perfiles laborales que posee un egresado	Atributo de Instancia
Religión	Creencia religiosa		Conjunto de creencias o preferencias espirituales que posee un egresado	Atributo de Instancia
Tipo De Sangre			Grupo sanguíneo al que pertenece un	Atributo de Instancia

			egresado	
Factor RH			Tipo de factor RH que posee un egresado	Atributo de Instancia
Aficiones	Hobbies, hobby, intereses, pasatiempos		Conjunto de intereses y afinidades que posee un egresado por algo en especial	Atributo de Instancia
No Egresado				
Nombres			Primer y segundo nombre (si se tiene) de un egresado o de un no egresado	Atributo de Instancia
Apellidos			Apellido o los apellidos de un egresado o de un no egresado	Atributo de Instancia
Género	Sexo		Género masculino o femenino de un egresado o de un no egresado	Atributo de Instancia
Fecha De Nacimiento			Año, mes y día que consta en el registro civil como fecha de nacimiento de un egresado o de un no egresado	Atributo de Instancia
Correo Electrónico	E-mail, mail		Dirección de correo electrónico principal de un egresado o de un no egresado	Atributo de Instancia
Cargo				
Nombre			Secuencia de uno o más caracteres que sirven para identificar un cargo	Atributo de Instancia
Departamento			Cada una de las partes en la cuales está dividida una organización	Atributo de Instancia
Nivel			Cada uno de los rangos o grados en los cuales está dividida una organización	Atributo de Instancia
Número De Empleados			Cantidad de empleados que un egresado tiene a su directa disposición	Atributo de Instancia
Estado			Estado del cargo (Actual o Anterior)	Atributo de Instancia

Organización				
Nombre			Secuencia de uno o más caracteres que sirven para identificar una organización	Atributo de Instancia
Teléfono	Número telefónico		Número telefónico principal de una organización	Atributo de Instancia
Lugar				
Nombre			Secuencia de uno o más caracteres que sirven para identificar un lugar	Atributo de Instancia
Dirección			Dirección en la cual se ubica el lugar referido	Atributo de Instancia
Ciudad			Ciudad en la cual se ubica el lugar referido	Atributo de Instancia
País			País en el cual se ubica el lugar referido	Atributo de Instancia
Longitud			Longitud del lugar referido	Atributo de Instancia
Latitud			Latitud del lugar referido	Atributo de Instancia
Continente			Continente en el cual se ubica el lugar referido	Atributo de Instancia
Proyecto				
Nombre			Secuencia de uno o más caracteres que sirven para identificar un proyecto	Atributo de Instancia
Descripción			Explicación detallada y ordenada acerca de un proyecto	Atributo de Instancia
Foto				
URL			La dirección global de la foto en la World Wide Web	Atributo de Instancia
Pregrado				
Carrera	Profesión		Conjunto de estudios que habilitan para el ejercicio de una profesión	Atributo de Instancia

Titulo			Calificación concedida por una institución de educación superior después de la finalización satisfactoria del programa de estudios correspondiente	Atributo de Instancia
Fecha De Grado			Año, mes y día en el cual un egresado recibe su grado de pregrado	Atributo de Instancia
Distinciones	Menciones		Conjunto de distinciones recibidas por un egresado en el desarrollo de su pregrado	Atributo de Instancia
Posgrado				
Nombre			Secuencia de uno o más caracteres que sirven para identificar un posgrado	Atributo de Instancia
Clase De Programa	Tipo de programa		Subdivisión empleada para clasificar los posgrados	Atributo de Instancia
Titulo			Calificación concedida por una institución de educación superior después de la finalización satisfactoria del programa de estudios correspondiente	Atributo de Instancia
Fecha De Grado			Año, mes y día en el cual un egresado recibe su grado de posgrado	Atributo de Instancia
Distinciones	Menciones		Conjunto de distinciones recibidas por un egresado en el desarrollo de su posgrado	Atributo de Instancia
Idioma				
Nombre			Secuencia de uno o más caracteres que sirven para identificar una cosa	Atributo de Instancia
Nivel Escritura			Nivel de escritura que posee un egresado respecto a un idioma	Atributo de Instancia
Nivel Lectura			Nivel de lectura que posee un	Atributo de Instancia

			egresado respecto a un idioma	
Nivel Pronunciación			Nivel de pronunciación que posee un egresado respecto a un idioma	Atributo de Instancia
Evento Académico				
Nombre			Secuencia de uno o más caracteres que sirven para identificar un evento académico	Atributo de Instancia
Tipo	Clase		Subdivisión empleada para clasificar los eventos académicos	Atributo de Instancia
Grupo				
Nombre			Secuencia de uno o más caracteres que sirven para identificar un grupo	Atributo de Instancia
Categoría			Subdivisión empleada para clasificar los grupos	Atributo de Instancia
Evento				
Descripción			Explicación detallada y ordenada acerca de un evento	Atributo de Instancia
Fecha De Inicio			Año, mes y día en el cual inicia un evento	Atributo de Instancia
Fecha De Finalización			Año, mes y día en el cual finaliza un evento	Atributo de Instancia
Relaciones				
Amigo (egresado, egresado)			Persona que tiene amistad como tratamiento afectuoso aunque no haya verdadera amistad	Relación
Esposos_Egresado (egresado, egresado)			Relación matrimonial entre dos egresados	Relación
Esposos_NoEgresado (egresado, no egresado)			Relación matrimonial entre un egresado y un no egresado	Relación
Padres_Egresado (egresado,			Hombre/Mujer que ha tenido	Relación

egresado)			descendencia	
Padres_NoEgresado (egresado, no egresado)			Hombre/Mujer que ha tenido descendencia	Relación
Hijos_Egresado (egresado, egresado)			Descendiente directo en primer grado de una persona	Relación
Hijos_NoEgresado (egresado, no egresado)			Descendiente directo en primer grado de una persona	Relación
Hermanos_Egresado (egresado, egresado)			Persona que con respecto a otra tiene el mismo padre y la misma madre o solamente el mismo padre o la misma madre	Relación
Hermanos_NoEgresado (egresado, no egresado)			Persona que con respecto a otra tiene el mismo padre y la misma madre o solamente el mismo padre o la misma madre	Relación
Novio(egresado, egresado)			Persona que mantiene una relación amorosa con otra sin intención de casarse y sin convivir con ella	Relación
Novia (egresado, egresado)			Persona que mantiene una relación amorosa con otra sin intención de casarse y sin convivir con ella	Relación
Vive En (egresado, lugar)			Un egresado vive en un lugar	Relación
Nació En (egresado, lugar)			Un egresado nace en un lugar	Relación
Representado Por (egresado, foto)			Un egresado es representado virtualmente por una foto	Relación
Aparece En (egresado, foto)			Un egresado es captado en una foto	Relación
Maneja (egresado, idioma)			Un egresado maneja un idioma	Relación
Integrante (egresado, proyecto)			Un egresado es integrante de un proyecto	Relación
Creador (egresado, grupo)			Un egresado es el creador de un grupo	Relación

Es Parte De (egresado, grupo)			Un egresado es parte de un grupo	Relación
Cargos (egresado, cargo)			Conjunto de cargos desempeñados por un egresado	Relación
Posee Un (egresado, pregrado)			Un egresado posee un pregrado	Relación
Posee Un (egresado, posgrado)			Un egresado posee un posgrado	Relación
Participo En (egresado, evento académico)			Un egresado participo en un evento académico	Relación
Tomada En (foto, lugar)			Una foto es tomada en un lugar	Relación
Ejercido En (cargo, organización)			Un cargo es ejercido en una organización	Relación
Celebrado En (evento académico, lugar)			Un evento académico es celebrado en un lugar	Relación
Subclase De (evento, grupo)			Evento es una subclase de grupo	Relación
Celebrado En (evento, lugar)			Un evento es celebrado en un lugar	Relación

Anexo B. Diccionario de conceptos

Nombre Del Concepto	Instancias	Atributos de Clase	Atributos de Instancias	Relaciones
Egresado			Código, contraseña, tipo de documento de identidad, documento de identidad, nombres, apellidos. genero, fecha de nacimiento, correo electrónico, número telefónico, sitio web, estado civil, estado laboral, perfil laboral, religión, tipo de sangre, factor Rh, aficiones	Amigo, esposos_Egresado, esposos_NoEgresado, padres_Egresado, padres_NoEgresado, hijos_Egresado, hijos_NoEgresado, hermanos_Egresado, hermanos_NoEgresado, novio, novia, nació en, lugar de origen, vive en, habitado por, vinculado a, maneja, es manejado por, integrante, es integrado por, creador, es creado por, es parte de, compuesto por, posee un, tiene un, tiene una, asociado a, representado por, representa a, aparece en, compuesta por, cargos, ejercido por, participo en, acogió a, publica, publicada por
No Egresado			Nombres, apellidos. genero, fecha de nacimiento, correo electrónico	Esposos_NoEgresado, padres_NoEgresado, hijos_NoEgresado, hermanos_NoEgresados

Lugar			Nombre, dirección, ciudad, país, longitud, latitud, continente	Vive en, habitado en, nació en, lugar de origen, tomada en, captado en, celebrado en, acoge un, localizada en, alberga una
Grupo			Nombre, categoría	Compuesto por, es parte de, tiene un, asociado a, creador, es creado por
Evento			Descripción, fecha de inicio, fecha de finalización	Compuesto por, es parte de, subclase de, celebrado en, acoge un, creador, es creado por, tiene un, asociado a
Pregrado			Carrera, título, fecha de grado, distinciones	Posee un, es poseído por
Posgrado			Nombre, clase de programa, título, fecha de grado, distinciones	Posee un, es poseído por
Evento Académico			Nombre, tipo	Participo en, acogió a, celebrado en, acoge un
Idioma			Nombre, nivel escritura, nivel lectura, nivel pronunciación	Maneja, es manejado por
Cargo			Nombre, departamento, nivel, número de empleados	Cargos, ejercido por, ejercido en, se ejerce en
Proyecto			Nombre, descripción	Integrante, es integrado por
Foto			URL	Representado por, representa a, aparece en, compuesta por, tomada en, captado en
Organización			Nombre, teléfono	Ejercido en, se ejerce un

Anexo C. Relaciones binarias

Nombre de la Relación	Concepto Origen	Cardinalidad Máxima	Concepto Destino	Relación Inversa
Amigo	Egresado	N	Egresado	AmigoEm
Esposos_Egresado	Egresado	1	Egresado	EsposaEfEm
Esposos_NoEgresado	Egresado	1	No Egresado	EsposaNfEm
Padres_Egresado	Egresado	N	Egresado	HijoEmEm
Padres_NoEgresado	Egresado	N	No Egresado	HijoNmEm
Hijos_Egresado	Egresado	1	Egresado	PadreEmEm
Hijos_NoEgresado	Egresado	1	No Egresado	PadreNmEm
Hermanos_Egresado	Egresado	N	Egresado	HermanoEmEm
Hermanos_NoEgresado	Egresado	N	No Egresado	HermanoNmEm
Novio	Egresado	1	Egresado	Novia
Novia	Egresado	1	Egresado	Novio
Vive En	Egresado	1	Lugar	Habitado Por
Nació En	Egresado	1	Lugar	Lugar De Origen
Maneja	Egresado	N	Idioma	Es manejado Por
Integrante	Egresado	N	Proyecto	Es Integrado Por
Creador	Egresado	N	Grupo	Es Creado Por
Es Parte De	Egresado	N	Grupo	Compuesto Por
Representado Por	Egresado	1	Foto	Representa A
Aparece En	Egresado	N	Foto	Compuesta Por
Cargos	Egresado	N	Cargo	Ejercido Por
Posee Un	Egresado	N	Pregrado	Es Poseído Por
Posee Un	Egresado	N	Posgrado	Es Poseído Por
Participo En	Egresado	N	Evento Académico	Acogió A
Tomada En	Foto	1	Lugar	Captado En

Ejercido En	Cargo	1	Organización	Se Ejerce Un
Localizada En	Organización	1	Lugar	Alberga Una
Celebrado En	Evento Académico	1	Lugar	Acoge Un
Subclase De	Evento	1	Grupo	Padre De
Celebrado En	Evento	1	Lugar	Acoge Un

Anexo D. Atributos de instancias

Nombre Del Atributo de Instancia	Concepto	Tipo de Valor	Rango de Valores	Cardinalidad
Nombres	No Egresado	String		(1:N)
Apellidos	No Egresado	String		(1:N)
Género	No Egresado	String		(1:N)
Fecha De Nacimiento	No Egresado	Date		(1:N)
Correo Electrónico	No Egresado	String		(1:1)
Nombres	Egresado	String		(1:N)
Apellidos	Egresado	String		(1:N)
Género	Egresado	String		(1:N)
Fecha De Nacimiento	Egresado	Date		(1:N)
Correo Electrónico	Egresado	String		(1:1)
Estado Laboral	Egresado	String		(1:N)
Perfil Laboral	Egresado	String		(1:N)
Cargos	Egresado	String		(N:N)
Código	Egresado	String		(1:1)
Contraseña	Egresado	String		(1:N)

Tipo De Documento De Identidad	Egresado	String		(1:N)
Documento De Identidad	Egresado	Int	0-9999999999	(1:1)
Número Telefónico	Egresado	Int	7-10	(N:M)
Sitio Web	Egresado	String		(N:1)
Estado Civil	Egresado	String		(1:N)
Religión	Egresado	String		(1:N)
Tipo De Sangre	Egresado	String		(1:N)
Factor RH	Egresado	String		(1:N)
Aficiones	Egresado	String		(1:N)
Nombre	Cargo	String		(1:1)
Departamento	Cargo	String		(N:M)
Estado	Cargo	String		(1:N)
Nivel	Cargo	String		(N:M)
Número De Empleados	Cargo	Int	0-	(1:N)
Nombre	Organización	String		(1:1)
Teléfono	Organización	Int	7-10	(N:1)
Nombre	Lugar	String		(1:N)
Dirección	Lugar	String		(1:1)
Ciudad	Lugar	String		(1:N)
País	Lugar	String		(1:N)
Longitud	Lugar	String		(1:1)
Latitud	Lugar	String		(1:1)
Continente	Lugar	String		(1:N)
Nombre	Proyecto	String		(1:1)
Descripción	Proyecto	String		(1:1)
Carrera	Pregrado	String		(1:1)
Título	Pregrado	String		(1:1)
Fecha De Grado	Pregrado	Date		(1:N)

Distinciones	Pregrado	String	(N:M)
Nombre	Posgrado	String	(1:1)
Clase De Programa	Posgrado	String	(1:N)
Título	Posgrado	String	(1:1)
Fecha De Grado	Posgrado	Date	(1:N)
Distinciones	Posgrado	String	(N:M)
Nombre	Idioma	String	(1:N)
Nivel Escritura	Idioma	String	(1:N)
Nivel Lectura	Idioma	String	(1:N)
Nivel Pronunciación	Idioma	String	(1:N)
Nombre	Evento Académico	String	(1:N)
Tipo	Evento Académico	String	(N:M)
Nombre	Grupo	String	(1:1)
Categoría	Grupo	String	(N:M)
Descripción	Evento	String	(1:N)
Fecha De Inicio	Evento	Date	(1:N)
Fecha De Finalización	Evento	Date	(1:N)

Anexo E. Axiomas formales

Nombre Axioma	Del	Descripción	Expresión	Conceptos	Relaciones	Variables
Duplicidad egresado-relación		Un mismo egresado no puede representar las dos partes de una misma relación	$\forall x \forall y (Egresado(x), Relación(x, y)) \rightarrow \neg Igual(x, y)$	Egresado	Amigo, esposos_Egresado, padres_Egresado, hijos_Egresado, hermanos_Egresado, novio, novia	x, y
Restricción relacional egresado	no	Un no egresado únicamente estará relacionado a uno o más egresados	$\forall x (NoEgresado(x), Relación(x, y)) \rightarrow Egresado(y)$	No egresado, egresado	Esposos_NoEgresado, padres_NoEgresado, hijos_NoEgresado, hermanos_NoEgresado	x
Existencia de un lugar		Un lugar debe de estar ligado por lo menos a un egresado, a un evento, a un evento académico o a una foto para poder existir	$\forall x \exists y (Lugar(x), Existe(x)) \rightarrow Ligado(x, y), (Egresado(y) \vee Evento(y) \vee EventoAcadémico(y) \vee Foto(y))$	Lugar, egresado, evento, evento académico, foto	Vive en, habitado por, nació en, lugar de origen, tomada en, captado en, celebrado en, acoge un	x, y
Vínculo lugar	evento-	Un evento siempre tendrá ligado un	$\forall x \exists y (Evento(x), Existe(x)) \rightarrow CelebradoEn(x, y), Lugar(y)$	Evento, lugar	Celebrado en, acoge un	x, y

	único lugar				
Existencia de un evento académico	Un evento académico debe de estar ligado por lo menos a un egresado para poder existir	$\forall x \exists y (EventoAcadémico(x), Existe(x)) \rightarrow Ligado(x, y), Egresado(y)$	Evento académico, egresado	Participo en, acogió a	x, y
Vínculo evento académico-lugar	Un evento académico siempre tendrá ligado un único lugar	$\forall x \exists y (EventoAcadémico(x), Existe(y)) \rightarrow CelebradoEn(x, y), Lugar(y)$	Evento académico, lugar	Celebrado en, acoge un	x, y
Existencia de un idioma	Un idioma debe de estar ligado por lo menos a un egresado para poder existir	$\forall x \exists y (Idioma(x), Existe(x)) \rightarrow Ligado(x, y), Egresado(y)$	Idioma, egresado	Maneja, es manejado por	x, y
Existencia de un posgrado	Un posgrado debe de estar ligado por lo menos a una egresado para poder existir	$\forall x \exists y (Posgrado(x), Existe(x)) \rightarrow Ligado(x, y), Egresado(y)$	Posgrado, egresado	Posee un, es poseído por	x, y
Existencia de un pregrado	Un pregrado debe de estar ligado por lo menos a una egresado para poder existir	$\forall x \exists y (Pregrado(x), Existe(x)) \rightarrow Ligado(x, y), Egresado(y)$	Pregrado, egresado	Posee un, es poseído por	x, y
Existencia de un cargo	Un cargo debe de estar ligado por lo menos a un egresado para	$\forall x \exists y (Cargo(x), Existe(x)) \rightarrow Ligado(x, y), Egresado(y)$	Cargo, egresado	Cargo actual, ejercido por, cargo anterior, ejercido por	x, y

	poder existir				
Existencia de una organización	Una organización debe de estar ligada por lo menos a un cargo para poder existir	$\forall x \exists y (Organización(x), Existe(x)) \rightarrow Ligado(x, y), Cargo(y)$	Organización, cargo	Ejercido en, se ejerce un	x, y
Existencia de un evento	Un evento debe de estar ligado por lo mínimo a un egresado para poder existir	$\forall x \exists y (Evento(x), Existe(x)) \rightarrow Ligado(x, y), Egresado(y)$	Evento, egresado	Creador, es creado por, participa en, acoge a	x, y
Existencia de un proyecto	Un proyecto debe de estar ligado por lo mínimo a un egresado para poder existir	$\forall x \exists y (Proyecto(x), Existe(x)) \rightarrow Ligado(x, y), Egresado(y)$	Proyecto, egresado	Integrante, es integrado por	x, y
Existencia de un grupo	Un grupo debe de estar ligado por lo mínimo a un egresado para poder existir	$\forall x \exists y (Grupo(x), Existe(x)) \rightarrow Ligado(x, y), Egresado(y)$	Grupo, egresado	Creador, es creado por, participa en, acoge a	x, y
Existencia de una foto	Una foto debe de estar ligada por lo mínimo a un egresado para poder existir	$\forall x \exists y (Foto(x), Existe(x)) \rightarrow Ligado(x, y), Egresado(y)$	Foto, egresado	Representado por, representa a, aparece en, compuesta por	x, y

Anexo F. Reglas

Nombre de la Regla	Descripción	Expresión	Conceptos	Atributos	Relaciones	Variables
Trabajo						
Existencia laboral	Si un trabajo existe, entonces está asociado a un egresado		Trabajo, egresado			
Relación egresado-organización-cargo	Si un egresado está asociado a un trabajo y no está desempleado, entonces está asociado a una organización y a un cargo		Egresado, trabajo, organización, cargo	Estado laboral		
Foto						
Lugar visitado	Si un egresado está asociado a una foto y la foto está asociada a un lugar, entonces el egresado estuvo en ese lugar		Egresado, foto, lugar			
Relación foto-egresado-conexión	Si una foto está asociada a un egresado, entonces hace parte de la conexión de este		Foto, egresado, conexión			
Evento						
Lugar de realización de un evento	Si un evento existe, entonces está asociado a un lugar		Evento, lugar			

Asistencia a un evento	Si un egresado está asociado a una conexión y dicha conexión está asociada a un evento, entonces el egresado asistió o asistirá a dicho evento		Egresado, conexión, evento			
Relación egresado-evento-lugar	Si un egresado asistió a un evento, entonces visito el lugar asociado a este		Egresado, evento, lugar			
Evento Académico						
Lugar de realización de un evento académico	Si un evento académico existe, entonces está asociado a un lugar		Evento académico, lugar			
Asistencia a un evento académico	Si un egresado está asociado a una formación académica y dicha formación académica está asociada a un evento académico, entonces el egresado asistió a dicho evento académico		Egresado, formación académica, evento académico			
Relación egresado-evento académico-lugar	Si un egresado asistió a un evento académico, entonces visitó el lugar asociado a este		Egresado, evento académico, lugar			
Idioma						

Manejo de idiomas	Si un egresado está asociado a una formación académica y dicha formación académica está asociada a un idioma, entonces el egresado maneja dicho idioma		Egresado, formación académica, idioma			
Posgrado						
Estudios de posgrado	Si un egresado está asociado a una formación académica y dicha formación académica está asociada a un posgrado, entonces el egresado curso o está en proceso de cursar dicho posgrado		Egresado, formación académica, posgrado			
Pregrado						
Estudios de pregrado	Si un egresado está asociado a una formación académica y dicha formación académica está asociada a un pregrado, entonces el egresado curso o está en proceso de cursar dicho pregrado		Egresado, formación académica, pregrado			

Anexo G. Cargar una ontología en memoria a partir de un fichero OWL

```
private OntModel loadOntModelFromOwlFile(String owlfile)
{
    OntModel ontmodel = null;
    InputStream is = null;

    //OWL_MEM es la especificación para modelos OWL almacenados en memoria
    ontmodel = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM);

    try
    {
        is = new FileInputStream(new File(owlfile));
        ontmodel.read(is, "");
        System.out.println("Se ha cargado una instancia OntModel.");
    }
    catch (Exception exc)
    {
        System.out.println("Se ha producido una excepción controlada.");
        exc.printStackTrace();
    }

    return ontmodel;
}
```

Anexo H. Especificación funcional - crear grupo de interés

ID REQUERIMIENTO: SRE-RF001
NOMBRE: Crear grupo de interés.
DESCRIPCION: El sistema permitirá al usuario registrado en la red social crear un grupo sobre el cual podrá debatir y comentar acerca de una temática deseada.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre, descripción, autor.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo I. Especificación funcional - editar grupo de interés

ID REQUERIMIENTO: SRE-RF002
NOMBRE: Editar grupo de interés.
DESCRIPCION: El sistema permitirá al usuario creador del grupo de interés realizar cambios concernientes a la información del grupo.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre, descripción, autor.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo J. Especificación funcional - eliminar grupo de interés

ID REQUERIMIENTO: SRE-RF003
NOMBRE: Eliminar grupo de interés.
DESCRIPCION: El sistema permitirá al usuario creador del grupo eliminar dicho grupo del sistema.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre, descripción, autor, URL.
PRIORIDAD: Alta () Media (X) Baja ()
AUTOR: Germán Hernández.

Anexo K. Especificación funcional - invitar Usuario al grupo de interés

ID REQUERIMIENTO: SRE-RF004
NOMBRE: Invitar usuario al grupo de interés.
DESCRIPCION: El sistema permitirá que usuario creador del grupo de interés pueda realizar solicitudes de invitación a los usuarios que se encuentre relacionado con el.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre, descripción, autor, invitado.
PRIORIDAD: Alta () Media (X) Baja ()
AUTOR: Germán Hernández.

Anexo L. Especificación funcional - eliminar Usuario del grupo de interés

ID REQUERIMIENTO: SRE-RF005
NOMBRE: Eliminar usuario del grupo de interés.
DESCRIPCION: El sistema permitirá al usuario creador del grupo de interés la posibilidad de eliminar a cualquier usuario que haya aceptado la invitación al grupo de interés.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre, descripción, autor, invitado.
PRIORIDAD: Alta () Media (X) Baja ()
AUTOR: Germán Hernández.

Anexo M. Especificación funcional - realiza publicaciones en el grupo de interés

ID REQUERIMIENTO: SRE-RF006
NOMBRE: Realizar publicaciones en el grupo de interés.
DESCRIPCION: El sistema permitirá a los usuarios de la red pertenecientes al grupo de interés, la posibilidad de realizar publicaciones en el grupo de interés, de la misma forma que realizan publicaciones en sus perfiles.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre, descripción, autor, invitado.
PRIORIDAD: Alta () Media (X) Baja ()
AUTOR: Germán Hernández.

Anexo N. Especificación funcional - eliminar publicación del grupo de interés

ID REQUERIMIENTO: SRE-RF007
NOMBRE: Eliminar publicación del grupo de interés.
DESCRIPCION: El sistema permitirá tanto al creador del grupo como al creador de la publicación eliminar dicha publicación del grupo de interés.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre, descripción, autor, invitado, grupo.
PRIORIDAD: Alta () Media (X) Baja ()
AUTOR: Germán Hernández.

Anexo O. Especificación funcional - crear publicaciones

ID REQUERIMIENTO: SRE-RF008
NOMBRE: Crear publicaciones
DESCRIPCION: El sistema permitirá a los usuarios realizar publicaciones en su perfil.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario, id-nota.• Publicaciones: Son los elementos que el usuario podrá añadir a su (perfil fotos, galerías, comentarios).
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo P. Especificación funcional - eliminar publicaciones

ID REQUERIMIENTO: SRE-RF009
NOMBRE: Eliminar publicaciones.
DESCRIPCION: El sistema permitirá a los usuarios y a la vez al administrador, la eliminación de las publicaciones realizadas con anterioridad.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario, id-publicación.• Publicaciones: Son los elementos que el usuario podrá añadir a su (perfil fotos, galerías, comentarios).
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo Q. Especificación funcional - comentar publicaciones

ID REQUERIMIENTO: SRE-RF010
NOMBRE: Comentar publicaciones.
DESCRIPCION: El sistema permitirá a los usuarios que tengan relación con el autor de la publicación realizar comentarios sobre esa publicación.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario, id-publicación.• Publicaciones: Son los elementos que el usuario podrá añadir a su (perfil fotos, galerías, comentarios).
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo R. Especificación funcional - crear mensajes

ID REQUERIMIENTO: SRE-RF011
NOMBRE: Crear mensajes.
DESCRIPCION: El sistema permitirá a los usuarios crear un mensaje para ser enviado posteriormente tanto a usuarios que estén relacionados como a usuarios que no estén relacionados.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta () Media (X) Baja ()
AUTOR: Germán Hernández.

Anexo S. Especificación funcional - crear borrador

ID REQUERIMIENTO: SRE-RF012
NOMBRE: Crear borrador.
DESCRIPCION: El sistema permitirá al usuario almacenar los borradores creados para ser enviados posteriormente tanto a usuarios que estén relacionados como a usuarios que no estén relacionados.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta () Media (X) Baja ()
AUTOR: Germán Hernández.

Anexo T. Especificación funcional - enviar mensajes

ID REQUERIMIENTO: SRE-RF013
NOMBRE: Enviar mensajes.
DESCRIPCION: El sistema permitirá a los usuarios enviar mensajes a diferentes usuarios así ellos no pertenezcan a su red. También permitirá que el administrador del sistema pueda enviar mensajes a los usuarios.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo U. Especificación funcional - crear solicitud de amistad

ID REQUERIMIENTO: SRE-RF014
NOMBRE: Crear solicitud de amistad.
DESCRIPCION: El sistema permitirá al usuario crear una solicitud de amistad para contactar y hacer parte de su red a los diferentes usuarios del sistema.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo V. Especificación funcional - enviar solicitud de amistad

ID REQUERIMIENTO: SRE-RF015
NOMBRE: Enviar solicitud de amistad.
DESCRIPCION: El sistema permitirá a los usuarios enviar la solicitud de amistad para contactar y hacer parte de su red a los diferentes usuarios del sistema.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario,
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo W. Especificación funcional - aceptar solicitud de amistad

ID REQUERIMIENTO: SRE-RF016
NOMBRE: Aceptar solicitud de amistad.
DESCRIPCION: El sistema permitirá a los usuarios aceptar la solicitud de amistad realizada por otro usuario.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo X. Especificación funcional - rechazar solicitud de amistad

ID REQUERIMIENTO: SRE-RF017
NOMBRE: Rechazar solicitud de amistad.
DESCRIPCION: El sistema permitirá a los usuarios rechazar la solicitud de amistad realizada por otro usuario.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo Y. Especificación funcional - ver información de usuario

ID REQUERIMIENTO: SRE-RF018
NOMBRE: Ver información de usuario.
DESCRIPCION: El sistema permitirá al administrador ver todos los usuarios del sistema.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo Z. Especificación funcional - ver información de usuario con relación

ID REQUERIMIENTO: SRE-RF019
NOMBRE: Ver información de usuario con relación.
DESCRIPCION: El sistema permitirá a los usuarios observar el perfil completo de los otros usuarios que pertenezcan a la red.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de Usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AA. Especificación funcional - realizar notificaciones

ID REQUERIMIENTO: SRE-RF020
NOMBRE: Realizar notificaciones.
DESCRIPCION: El sistema notificara automáticamente sobre alguna publicación que un tercero haga en el perfil. El sistema notificara automáticamente cuando un tercero haga un comentario en alguna publicación hecha por el usuario. El sistema notificara automáticamente, cuando haya nuevos mensajes no leídos.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AB. Especificación funcional - crear perfil de usuario

ID REQUERIMIENTO: SRE-RF021
NOMBRE: Crear perfil de usuario.
DESCRIPCION: El sistema permitirá al egresado de la Universidad Tecnología de Bolívar, la creación de su perfil de usuario en el cual introducirá los datos relevantes para posteriormente poder ser identificado.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AC. Especificación funcional - editar perfil de usuario

ID REQUERIMIENTO: SRE-RF022
NOMBRE: Editar perfil de usuario.
DESCRIPCION: El sistema permitirá al egresado de la Universidad Tecnología de Bolívar después de la creación de su perfil, poder modificar la información concerniente con este.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AD. Especificación funcional - eliminar perfil de usuario

ID REQUERIMIENTO: SRE-RF023
NOMBRE: Eliminar perfil de usuario.
DESCRIPCION: El sistema permitirá al administrador poder borrar un perfil de usuario del sistema. El administrador será el único que podrá eliminar un perfil de usuario de la red social.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AE. Especificación funcional - agregar foto al perfil de usuario

ID REQUERIMIENTO: SRE-RF024
NOMBRE: Agregar foto al perfil de usuario.
DESCRIPCION: Después de haber creado el perfil de usuario, el usuario podrá añadir su foto la cual tendrá una serie de campos que dotaran de contenido semántico a la imagen.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AF. Especificación funcional - enviar noticias a usuarios

ID REQUERIMIENTO: SRE-RF025
NOMBRE: Enviar noticias a usuarios.
DESCRIPCION: El sistema permitirá al administrador realizar publicaciones que podrán ser vistas por todos los usuarios pertenecientes a la red social de egresados.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AG. Especificación funcional - crear evento

ID REQUERIMIENTO: SRE-RF026
NOMBRE: Crear evento.
DESCRIPCION: El sistema permitirá a los usuarios la realización de eventos donde podrá disponer la fecha y lugar de realización.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AH. Especificación funcional - cancelar evento

ID REQUERIMIENTO: SRE-RF027
NOMBRE: Cancelar evento.
DESCRIPCION: El sistema permitirá a los usuarios la cancelación de eventos previamente creados.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AI. Especificación funcional - editar evento

ID REQUERIMIENTO: SRE-RF028
NOMBRE: Editar evento.
DESCRIPCION: El sistema permitirá a los usuarios creadores del evento modificar la fecha y lugar de realización.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AJ. Especificación funcional - eliminar evento

ID REQUERIMIENTO: SRE-RF029
NOMBRE: Eliminar evento.
DESCRIPCION: El sistema permitirá al usuario creador del evento eliminar el evento.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AK. Especificación funcional - añadir foto al evento

ID REQUERIMIENTO: SRE-RF030
NOMBRE: Añadir foto al evento.
DESCRIPCION: El sistema permitirá al usuario creador del evento añadir una foto de descripción para el evento.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AL. Especificación funcional - invitar usuarios al evento

ID REQUERIMIENTO: SRE-RF031
NOMBRE: Invitar usuarios al evento.
DESCRIPCION: El sistema permitirá al usuario creador del evento enviar solicitudes de invitación al evento a los usuarios que pertenezcan a su red.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AM. Especificación funcional - notificar cumpleaños

ID REQUERIMIENTO: SRE-RF032
NOMBRE: Notificar cumpleaños.
DESCRIPCION: El sistema permitirá al usuario mostrar los cumpleaños de las personas que pertenecen a su red.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AN. Especificación funcional - login

ID REQUERIMIENTO: SRE-RF033
NOMBRE: Login.
DESCRIPCION: El sistema proporcionara una zona de acceso por medio del cual se validara si el usuario esta registrado o no.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, contraseña.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AO. Especificación funcional - mostrar grupos de interés creados por el usuario

ID REQUERIMIENTO: SRE-RF034
NOMBRE: Mostrar grupos de interés creados por el usuario.
DESCRIPCION: El sistema permitirá al egresado mostrar un listado de todos los grupos de interés creados por el.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AP. Especificación funcional - mostrar grupos de interés a los que esta vinculado el egresado

ID REQUERIMIENTO: SRE-RF035
NOMBRE: Mostrar grupos de interés a los que esta vinculado el egresado.
DESCRIPCION: El sistema permitirá al egresado mostrar un listado de todos los grupos de interés a los cuales el pertenece.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AQ. Especificación funcional - mostrar eventos creados por usuario

ID REQUERIMIENTO: SRE-RF036
NOMBRE: Mostrar eventos creados por el usuario.
DESCRIPCION: El sistema permitirá al egresado mostrar un listado con todos eventos creados por el egresado.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AR. Especificación funcional - mostrar eventos a los que esta vinculado el usuario

ID REQUERIMIENTO: SRE-RF037
NOMBRE: Mostrar eventos a los que esta vinculado el usuario.
DESCRIPCION: El sistema permitirá al egresado mostrar un listado con todos eventos a los que el egresado esta vinculado.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de Usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AS. Especificación funcional - crear evento académico

ID REQUERIMIENTO: SRE-RF038
NOMBRE: Crear evento académico.
DESCRIPCION: El sistema permitirá a los usuarios la realización de eventos de tipo académicos donde podrá disponer la fecha y lugar de realización.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AT. Especificación funcional - editar evento académico

ID REQUERIMIENTO: SRE-RF039
NOMBRE: Editar evento académico.
DESCRIPCION: El sistema permitirá a los usuarios creadores del evento académico modificar la fecha y lugar de realización.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AU. Especificación funcional - eliminar evento académico

ID REQUERIMIENTO: SRE-RF040
NOMBRE: Eliminar evento académico.
DESCRIPCION: El sistema permitirá al usuario creador del evento académico eliminar dicho evento.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja()
AUTOR: Germán Hernández.

Anexo AV. Especificación funcional - añadir foto al evento académico

ID REQUERIMIENTO: SRE-RF041
NOMBRE: Añadir foto al evento académico.
DESCRIPCION: El sistema permitirá al usuario creador del evento académico añadir una foto de descripción para el evento.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

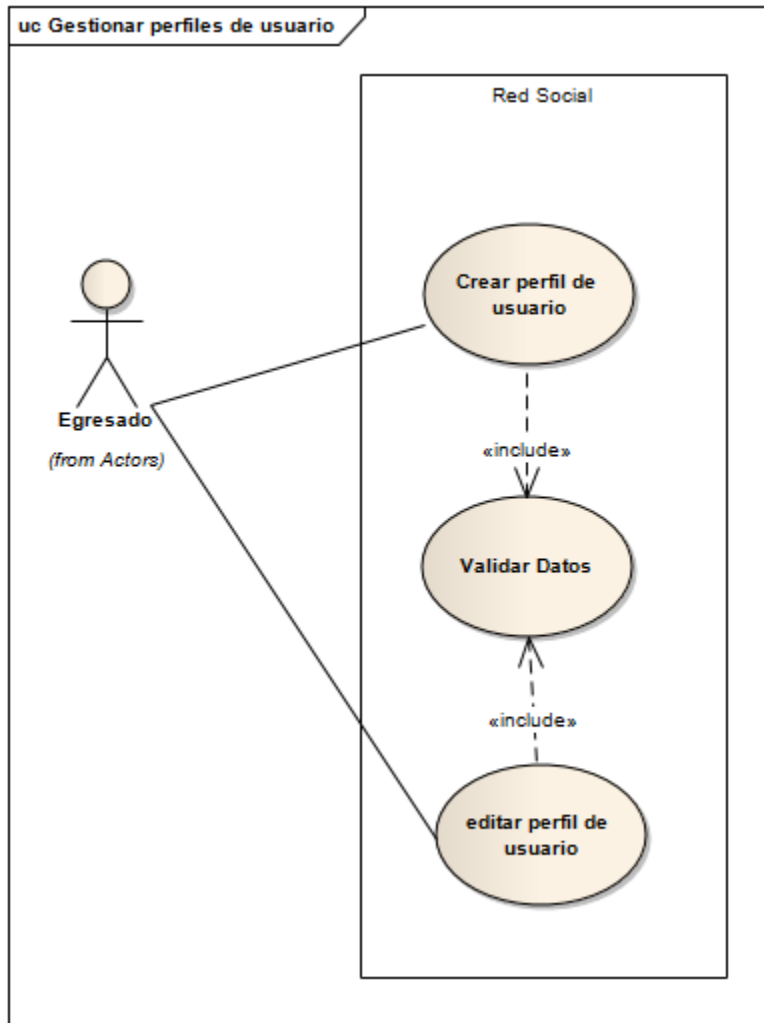
Anexo AW. Especificación funcional - invitar usuarios al evento académico

ID REQUERIMIENTO: SRE-RF043
NOMBRE: Invitar usuarios al evento académico.
DESCRIPCION: El sistema permitirá al usuario creador del evento académico enviar solicitudes de invitación a dicho evento a los usuarios que pertenezcan a su red.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AX. Especificación funcional - mostrar eventos académicos a los que esta vinculado el usuario

ID REQUERIMIENTO: SRE-RF041
NOMBRE: Mostrar eventos académicos a los que esta vinculado el usuario.
DESCRIPCION: El sistema permitirá al egresado mostrar un listado de todos los eventos académicos a los que el usuario esta vinculado.
TERMINOS: <ul style="list-style-type: none">• Datos básicos: Nombre de usuario, id-usuario.
PRIORIDAD: Alta (X) Media () Baja ()
AUTOR: Germán Hernández.

Anexo AY. Diagrama de caso de uso – gestionar usuario



Anexo AZ. Caso de uso - crear perfil de usuario

CU-01	Crear perfil de usuario	
Resumen	Este caso de uso inicia cuando el usuario hace click en el botón crear perfil. El sistema desplegara un formulario en el cual el usuario deberá indicar un nombre y una serie de datos requerido para crear el perfil en el sistema. Si dicho nombre y descripción son validos, el sistema creara el perfil de usuario en el sistema.	
Prioridad	Alta.	
Frecuencia De Uso	Alta.	
Precondiciones	- Ser egresado de la universidad Tecnológica de Bolívar.	
Condición Final De Éxito	Se crea un nuevo perfil de usuario.	
Condición Final De Fracaso	La solicitud para la creación del perfil de usuario es rechazada.	
Actores Directos	Egresado no registrado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa el botón de crear perfil.	
Casos Incluidos	Validar datos.	
Flujo Principal	Paso	Acción
	1	El egresado hace click en el botón crear usuario.
	2	El sistema despliega un formulario.
	3	El egresado llena el formulario y lo envía.
	4 Include::Validar datos	El sistema valida los datos ingresados.
	5	El sistema crea un nuevo perfil de usuario.
	6	El sistema muestra un mensaje de éxito al usuario.
Flujo Alterno	Paso	Acción Alternativa
	4.1	Existen campos vacíos.
	4.2	El código del egresado no corresponde al registrado en la base de datos de egresados.
	4.3	El sistema rechaza la solicitud para crear un nuevo perfil de usuario.
	4.4	El sistema muestra un mensaje de error indicando los campos inválidos.

Anexo BA. Caso de uso - editar perfil de usuario

CU-02	Editar perfil de usuario	
Resumen	Este caso de uso inicia cuando el usuario hace click en el botón editar perfil. El sistema desplegara un formulario en el cual se le mostrara al usuario los campos que podrá editar. El sistema actualizara los datos del perfil de usuario en el sistema.	
Prioridad	Alta.	
Frecuencia De Uso	Alta.	
Precondiciones	- Haber creado exitosamente un perfil en la red social.	
Condición Final De Éxito	Se actualiza un perfil de usuario existente.	
Condición Final De Fracaso	La solicitud para la edición del perfil de usuario es rechazada.	
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa el botón de editar perfil.	
Casos Incluidos	Validar datos.	
Flujo Principal	Paso	Acción
	1	El usuario hace click en el botón crear usuario
	2	El sistema despliega un formulario.
	3	El usuario llena el formulario y lo envía.
	4 Include::Validar datos	El sistema valida los datos ingresados.
	5	El sistema actualiza un perfil de usuario existente
	6	El sistema muestra un mensaje de éxito al usuario.
Flujo Alternativo	Paso	Acción Alternativa
	4.1	Existen campos vacíos.
	4.2	El sistema rechaza la solicitud para crear in nuevo perfil de usuario.
	4.3	El sistema muestra un mensaje de error indicando los campos inválidos.

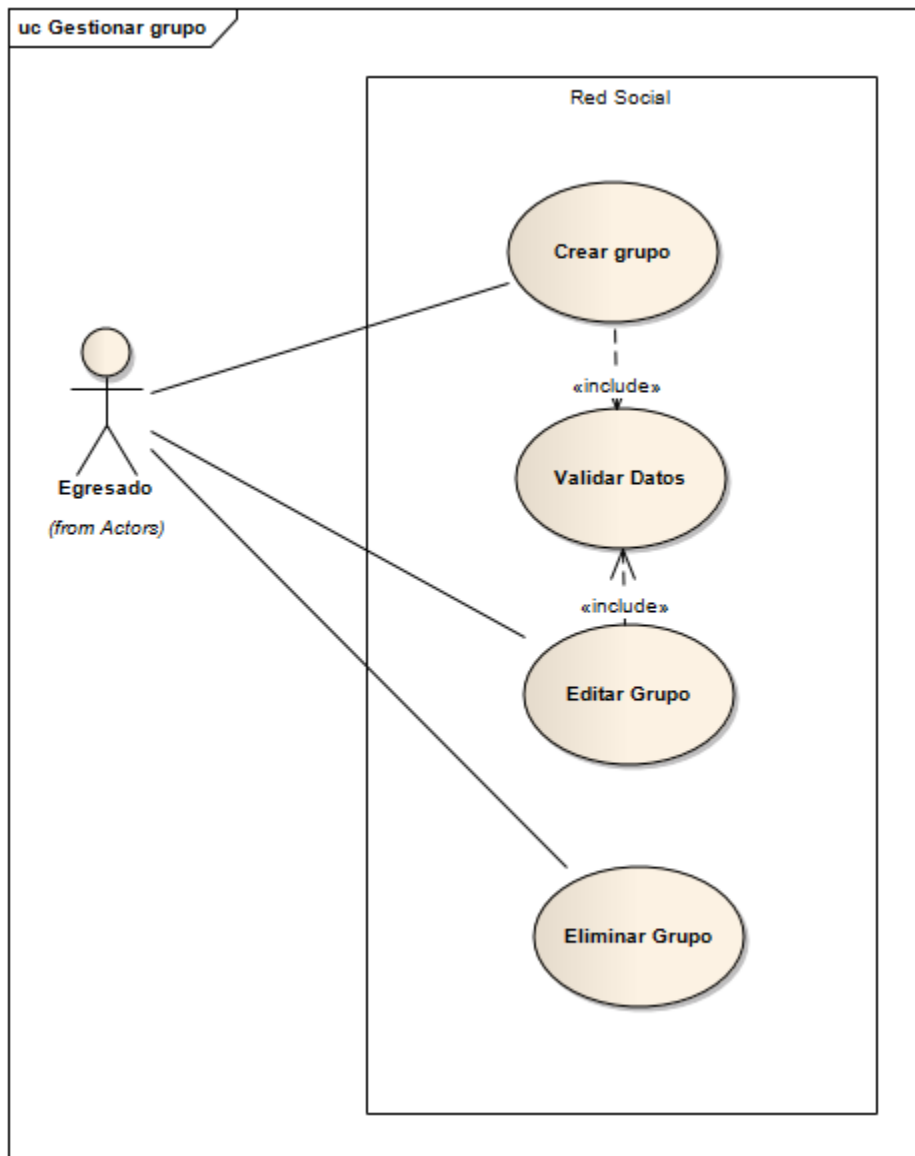
Anexo BB. Caso de uso - añadir foto al perfil de usuario

CU-03	Añadir foto al perfil de usuario	
Resumen	Este caso de uso inicia cuando el usuario hace click en añadir foto de perfil. El sistema desplegara un formulario en el cual el usuario deberá indicar un nombre y una descripción de la imagen requerida para añadir la al perfil, si dicho nombre y descripción son validos y el formato de foto es reconocida, el sistema añadirá la foto al perfil de usuario.	
Prioridad	Alta.	
Frecuencia De Uso	Alta.	
Precondiciones	<ul style="list-style-type: none"> - El perfil de usuario debe existir. - No debe existir otra foto con el mismo nombre. 	
Condición Final De Éxito	Se añade una foto al perfil de usuario.	
Condición Final De Fracaso	La foto no es añadida.	
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa el botón de añadir foto de perfil.	
Casos Incluidos	Validar datos.	
Flujo Principal	Paso	Acción
	1	El usuario hace click en el botón añadir foto de perfil.
	2	El sistema despliega un formulario con el nombre la descripción y la foto a añadir.
	3	El usuario llena el formulario y lo envía.
	4 Include::Validar datos	El sistema valida los datos ingresados.
	5	El sistema añade la foto al perfil.
	6	El sistema muestra un mensaje de éxito al usuario.
Flujo Alterno	Paso	Acción Alternativa
	4.1	Existen campos vacíos.
	4.2	La imagen no tiene un formato soportado por el sistema.
	4.3	El sistema rechaza la solicitud para añadir la foto.
	4.4	El sistema muestra un mensaje de error indicando los campos inválidos.

Anexo BC. Casos de uso – validar datos

CU-04	Validar datos	
Resumen	Este caso de uso inicia cuando el sistema necesita validar datos enviados desde un formulario. El sistema recibe los datos, aplica las validaciones necesarias y devuelve un mensaje de error si los datos no pasaron las validaciones o un mensaje de éxito si los datos fueron correctos.	
Prioridad	Alta.	
Frecuencia De Uso	Alta.	
Precondiciones	Ninguna.	
Condición Final De Éxito	Datos validados.	
Condición Final De Fracaso	Datos no validados.	
Actores Directos	Ninguno.	
Actores Secundarios	Ninguno.	
Disparador	El sistema necesita validar datos enviados desde un formulario.	
Casos Incluidos	Ninguno.	
Flujo Principal	Paso	Acción
	1	El sistema recibe los datos a validar.
	2	El sistema somete a los datos a una serie de validaciones.
	3	Los datos pasan las validaciones.
	4	El sistema envía un mensaje de éxito.
Flujo Alternativo	Paso	Acción Alternativa
	3.1	Los datos no pasan las validaciones.
	2.2	El sistema muestra un mensaje de error.

Anexo BD. Diagrama de caso de uso – gestionar grupos



Anexo BD. Caso de uso – crear grupo de interés

CU-05	Crear grupo de interés	
Resumen	Este caso de uso inicia cuando el usuario hace click en el botón crear grupo de interés. El sistema desplegara un formulario en el cual el usuario deberá indicar un nombre y una serie de datos requerido para crear del grupo de interés en el sistema. Si dicho nombre y descripción son validos, el sistema creara el grupo de interés en el sistema.	
Prioridad	Alta.	
Frecuencia De Uso	Alta.	
Precondiciones	- El perfil de usuario debe existir.	
Condición Final De Éxito	Se crea un nuevo grupo de interés.	
Condición Final De Fracaso	La solicitud para la creación del grupo de usuario es rechazada	
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa el botón de crear grupo de interés.	
Casos Incluidos	Validar datos.	
Flujo Principal	Paso	Acción
	1	El usuario hace click en el botón crear grupo de interés.
	2	El sistema despliega un formulario.
	3	El usuario llena el formulario y lo envía.
	4 Include::Validar datos	El sistema valida los datos ingresados.
	5	El sistema crea un nuevo grupo de interés
	6	El sistema muestra un mensaje de éxito al usuario.
Flujo Alterno	Paso	Acción Alternativa
	4.1	Existen campos vacíos.
	4.2	El nombre del grupo ya existe.
	4.3	El sistema rechaza la solicitud para crear un nuevo grupo.
	4.4	El sistema muestra un mensaje de error indicando los campos inválidos.

Anexo BE. Caso de uso - editar grupo de interés

CU-06	Editar grupo de interés	
Resumen	Este caso de uso inicia cuando el usuario hace click en el botón editar grupo de interés. El sistema desplegara un formulario en el que se le mostrara al usuario los campos que se podrán editar del grupo de interés en el sistema, si dicho nombre y descripción son validos, el sistema actualizara el grupo de interés en el sistema.	
Prioridad	Alta.	
Frecuencia De Uso	Alta.	
Precondiciones	<ul style="list-style-type: none"> - Se debe encontrar en el listado de grupos relacionados con el egresado (usuario). - El usuario que realiza la modificación debe ser el creador. - El grupo debe existir. 	
Condición Final De Éxito	Se actualiza un grupo de interés existente.	
Condición Final De Fracaso	La solicitud para la edición del grupo de usuario es rechazada.	
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa el botón de editar grupo de interés.	
Casos Incluidos	Validar datos, mostrar grupos creados por el egresado.	
Flujo Principal	Paso	Acción
	1	Include:: mostrar grupos creados por el egresado.
	2	El egresado hace click en el botón editar grupo de interés.
	3	El sistema despliega un formulario.
	4	El egresado llena el formulario y lo envía.
	5	El sistema valida los datos ingresados.
	Include::Validar datos	
	6	El sistema actualiza los datos del grupo de interés.
	7	El sistema muestra un mensaje de éxito al usuario.
Flujo Alterno	Paso	Acción Alternativa
	5.1	Existen campos vacíos.
	5.2	El sistema rechaza la solicitud para modificar grupo de interés.
	5.3	El sistema muestra un mensaje de error indicando los campos inválidos.

Anexo BF. Caso de uso - eliminar grupo de interés

CU-07	Eliminar grupo de interés	
Resumen	Este caso de uso inicia cuando el usuario hace click en el botón eliminar grupo de interés. El sistema mostrara al usuario un mensaje de confirmación para la eliminación del grupo; si el usuario acepta el grupo de interés será eliminado del sistema.	
Prioridad	Alta.	
Frecuencia De Uso	Media.	
Precondiciones	<ul style="list-style-type: none"> - Se debe encontrar en el listado de grupos relacionados con el usuario. - El usuario que realiza la eliminación debe ser el creador. - El grupo debe existir. 	
Condición Final De Éxito	Se actualiza un grupo de interés existente.	
Condición Final De Fracaso	La solicitud para la eliminación del grupo de usuario es rechazada.	
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa la opción eliminar grupo de interés.	
Casos Incluidos	Mostrar grupos creados por el egresado.	
Flujo Principal	Paso	Acción
	1	Include:: Mostrar grupos creados por el egresado.
	2	El usuario hace click en el botón eliminar grupo de interés.
	3	El sistema muestra un mensaje de confirmación.
	4	El grupo es eliminando del sistema guardando una copia en la base de respaldo.
	5	El sistema muestra un mensaje de éxito al usuario.
	Paso	Acción Alternativa
Flujo Alternativo	4.1	El usuario da click en cancelar durante la confirmación.
	4.2	El sistema rechaza la solicitud para eliminar el grupo de interés.
	4.3	El sistema no modifica los datos del grupo de interés.

Anexo BG. Caso de uso - invitar usuario al grupo de interés

CU-08	Invitar usuario al grupo de interés	
Resumen	Este caso de uso inicia cuando el usuario hace click en la opción invitar amigo. El sistema mostrara al usuario un listado de las personas pertenecientes a su red a las cuales se le podrá enviar una invitación al grupo de interés.	
Prioridad	Alta.	
Frecuencia De Uso	Media.	
Precondiciones	<ul style="list-style-type: none"> - El usuario debe tener al menos un amigo. - Se debe estar dentro del grupo de interés. 	
Condición Final De Éxito	Se realiza una invitación para pertenecer al grupo de interés.	
Condición Final De Fracaso	La invitación no se realiza.	
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa la opción invitar amigo.	
Casos Incluidos		
Flujo Principal	Paso	Acción
	1	El usuario hace click en invitar un amigo.
	2	El sistema muestra un listado de los amigos pertenecientes a la red.
	3	El usuario selecciona un amigo del listado.
	4	El sistema envía un mensaje de confirmación.
	5	El usuario da click en aceptar.
	6	El sistema envía la invitación.
	Paso	Acción Alternativa
Flujo Alternativo	4.1	El usuario da click en cancelar durante la confirmación.
	4.2	El sistema rechaza la solicitud para enviar la invitación.
	4.3	El sistema no envía la invitación.

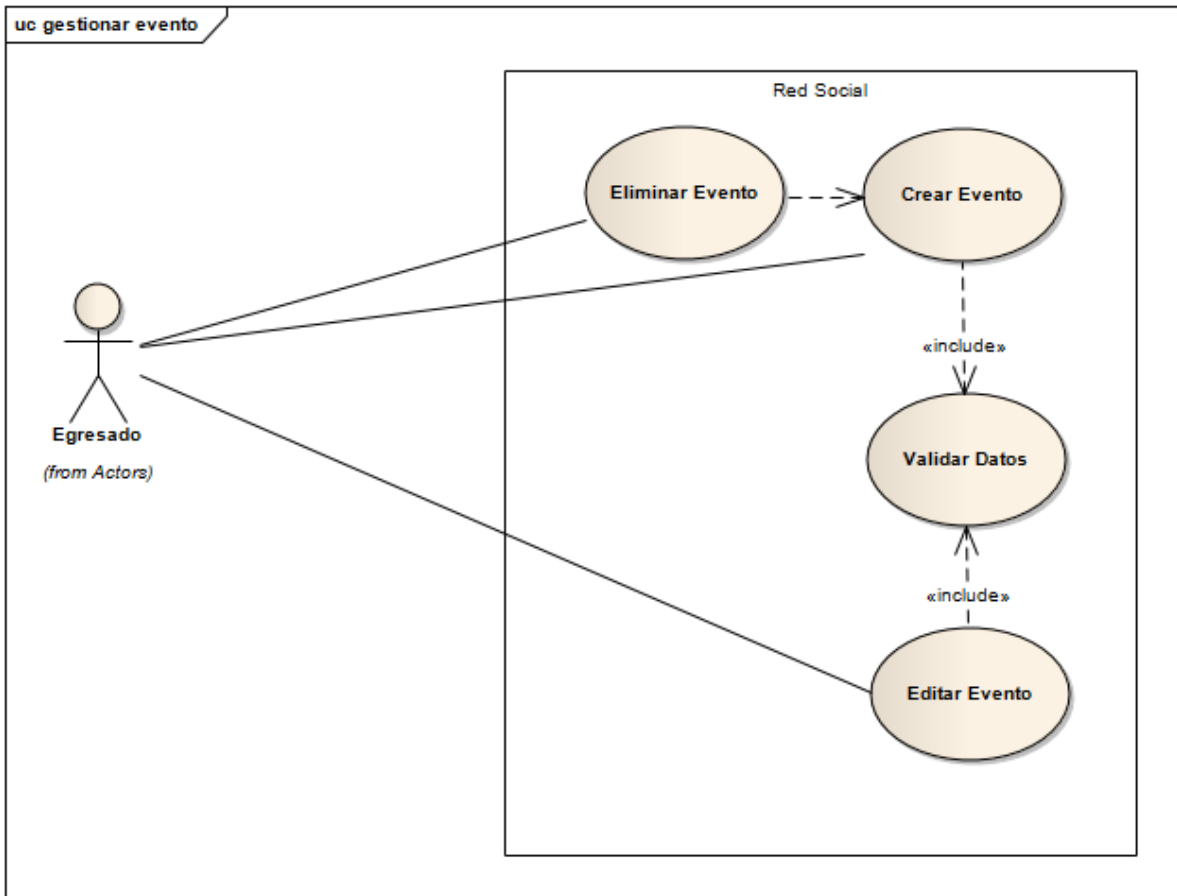
Anexo BH. Caso de uso - mostrar grupos creados por el egresado

CU-09	Mostrar grupos creados por el egresado	
Resumen	Este caso de uso inicia cuando el usuario hace click en la opción "mostrar grupos creados". El sistema desplegara un listado con los grupos creados por el usuario.	
Prioridad	Alta.	
Frecuencia De Uso	Media.	
Precondiciones	Ninguna.	
Condición Final De Éxito	Se muestra un listado con los grupos creados por el usuario	
Condición Final De Fracaso	No se muestra el listado.	
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa la opción mostrar grupos creados.	
Casos Incluidos		
Flujo Principal	Paso	Acción
	1	El usuario hace click en la opción mostrar grupos creados.
	2	El sistema verificas los grupos creados por el usuario.
	3	El sistema muestra un listado de los grupos creados por el usuario.
Flujo Alterno	Paso	Acción Alternativa
	3.1	El egresado no ha creado un grupo de interés.
	3.2	El sistema envía un mensaje advirtiéndole que el usuario no ha creado ningún grupo de interés.

Anexo BI. Caso de uso - mostrar grupos a los que pertenece el egresado

CU-10	Mostrar grupos a los que pertenece el egresado	
Resumen	Este caso de uso inicia cuando el usuario hace click en la opción "mostrar grupos a los que pertenezco". El sistema desplegara un listado de los grupos a los cuales esta vinculado el usuario.	
Prioridad	Alta.	
Frecuencia De Uso	Alta.	
Precondiciones	Ninguna.	
Condición Final De Éxito	Se muestra un listado de los grupos a los cuales esta vinculado el usuario.	
Condición Final De Fracaso	No se muestra el listado	
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa la opción mostrar grupos a los que pertenezco.	
Casos Incluidos		
Flujo Principal	Paso	Acción
	1	El usuario hace click en la opción mostrar grupos a los que pertenezco.
	2	El sistema verifica los grupos a los cuales el usuario está vinculado.
	3	El sistema muestra un listado de los grupos a los cuales el usuario está vinculado.
Flujo Alternativo	Paso	Acción Alternativa
	3.1	El usuario no esta vinculado a ningún grupo.
	3.2	El sistema envía un mensaje advirtiéndole que el usuario no esta vinculado a ningún grupo.

Anexo BJ. Diagrama de caso de uso - gestionar eventos



Anexo BK. Caso de uso – crear evento

CU-11	Crear evento	
Resumen	Este caso de uso inicia cuando el usuario hace click en el botón crear evento. El sistema desplegara un formulario en el cual el usuario deberá indicar un nombre y una serie de datos requeridos para crear el evento. Si dicho nombre y descripción son validos el sistema creara el evento.	
Prioridad	Alta.	
Frecuencia De Uso	Alta.	
Precondiciones	<ul style="list-style-type: none"> - El perfil de usuario debe existir. - No debe existir otro evento con el mismo nombre. 	
Condición Final De Éxito	Se crea un nuevo evento.	
Condición Final De Fracaso	La solicitud para la creación del evento es rechazada.	
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa el botón de crear evento.	
Casos Incluidos	Validar datos.	
Flujo Principal	Paso	Acción
	1	El usuario hace click en el botón evento.
	2	El sistema despliega un formulario.
	3	El usuario llena el formulario y lo envía.
	4 Include::Validar Datos	El sistema valida los datos ingresados.
	5	El sistema crea un nuevo evento.
	6	El sistema muestra un mensaje de éxito al usuario.
Flujo Alternativo	Paso	Acción Alternativa
	4.1	Existen campos vacíos.
	4.2	El nombre del evento ya existe.
	4.3	El sistema rechaza la solicitud para crear un nuevo evento.
	4.4	El sistema muestra un mensaje de error indicando los campos inválidos.

Anexo BL. Caso de uso - editar evento

CU-12	Editar evento	
Resumen	Este caso de uso inicia cuando el usuario hace click en el botón editar evento. El sistema desplegara un formulario en el que se le mostrara al usuario los campos que se podrán editar del evento en el sistema, si dicho nombre y descripción son validos, el sistema actualizara el evento.	
Prioridad	Alta.	
Frecuencia De Uso	Alta.	
Precondiciones	<ul style="list-style-type: none"> - El evento a editar, se debe encontrar en el listado de eventos relacionados con el usuario. - El usuario que realiza la modificación debe ser el creador. - El evento debe existir. 	
Condición Final De Éxito	Se modifica y actualiza un evento existente.	
Condición Final De Fracaso	La solicitud para la edición del evento es rechazada.	
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa el botón de editar evento.	
Casos Incluidos	Validar datos, mostrar eventos creados por el egresado.	
Flujo Principal	Paso	Acción
	1	Include:: Mostrar eventos creados por el egresado.
	2	El usuario hace click en el botón editar evento.
	3	El sistema despliega un formulario.
	4	El usuario llena el formulario y lo envía.
	5 Include::Validar datos	El sistema valida los datos ingresados.
	6	El sistema actualiza los datos del evento.
	7	El sistema muestra un mensaje de éxito al usuario.
Flujo Alternativo	Paso	Acción Alternativa
	5.1	Existen campos vacíos.
	5.2	El sistema rechaza la solicitud para modificar el evento.
	5.3	El sistema muestra un mensaje de error indicando los campos inválidos.

Anexo BM. Caso de uso - eliminar evento

CU-13	Eliminar evento	
Resumen	Este caso de uso inicia cuando el usuario hace click en el botón eliminar evento. El sistema mostrara al usuario un mensaje de confirmación para la eliminación del evento. Si el usuario acepta el evento será eliminado del sistema.	
Prioridad	Alta.	
Frecuencia De Uso	Baja.	
Precondiciones	<ul style="list-style-type: none"> - El evento a eliminar se debe encontrar en el listado de eventos relacionados con el egresado (usuario). - El egresado que realiza la eliminación debe ser el creador. - El evento debe existir. 	
Condición Final De Éxito	Se elimina el evento del sistema.	
Condición Final De Fracaso	La solicitud para la eliminación del grupo de usuario es rechazada	
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa el botón de eliminar evento.	
Casos Incluidos	Mostrar eventos creados por el egresado	
Flujo Principal	Paso	Acción
	1	Incluye:: Mostrar eventos creados por el egresado.
	2	El usuario hace click en el botón eliminar evento.
	3	El sistema muestra un mensaje de confirmación.
	4	El evento es eliminando del sistema.
	5	El sistema muestra un mensaje de éxito al usuario.
	Paso	Acción Alternativa
Flujo Alterno	4.1	El usuario da click en cancelar durante la confirmación.
	4.2	El sistema rechaza la solicitud para eliminar el evento.

Anexo BN. Caso de uso - Invitar usuario al evento

CU-14	Invitar usuario al evento	
Resumen	Este caso de uso inicia cuando el usuario hace click en la opción invitar personas. El sistema mostrara al usuario un listado de las personas pertenecientes a su red a las cuales se le podrá enviar una invitación al evento.	
Prioridad	Alta.	
Frecuencia De Uso	Alta.	
Precondiciones	<ul style="list-style-type: none"> - El usuario debe tener al menos un amigo. - Se debe estar invitado al evento. 	
Condición Final De Éxito	Se realiza una invitación para asistir al evento.	
Condición Final De Fracaso	La invitación no se realiza.	
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa la opción invitar amigo.	
Casos Incluidos		
Flujo Principal	Paso	Acción
	1	El usuario hace click en invitar un amigo.
	2	El sistema muestra un listado de los amigos pertenecientes a la red.
	3	El usuario selecciona un amigo del listado.
	4	El sistema envía un mensaje de confirmación.
	5	El usuario da click en aceptar.
	6	El sistema envía la invitación para asistir al evento.
	Paso	Acción Alternativa
Flujo Alternativo	4.1	El usuario da click en cancelar durante la confirmación.
	4.2	El sistema rechaza la solicitud para enviar la invitación de asistencia.
	4.3	El sistema no envía la invitación de asistencia al evento.

..

Anexo BO. Caso de uso - mostrar eventos creados por el egresado

CU-15	Mostrar eventos creados por el egresado	
Resumen	Este caso de uso inicia cuando el usuario hace click en la opción "eventos creados". El sistema desplegara un listado con los grupos creados por el usuario.	
Prioridad	Alta.	
Frecuencia De Uso	Media.	
Precondiciones	Ninguna.	
Condición Final De Éxito	Se muestra un listado con los eventos creados por el usuario.	
Condición Final De Fracaso	No se muestra el listado de los eventos	
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa la opción eventos creados.	
Casos Incluidos		
Flujo Principal	Paso	Acción
	1	El usuario hace click en la opción mostrar eventos creados.
	2	El sistema verificas los eventos creados por el usuario.
	3	El sistema muestra un listado de los eventos creados por el usuario.
Flujo Alternativo	Paso	Acción Alternativa
	3.1	El usuario no ha creado un evento.
	3.2	El sistema envía un mensaje advirtiéndole que el usuario no ha creado ningún evento.

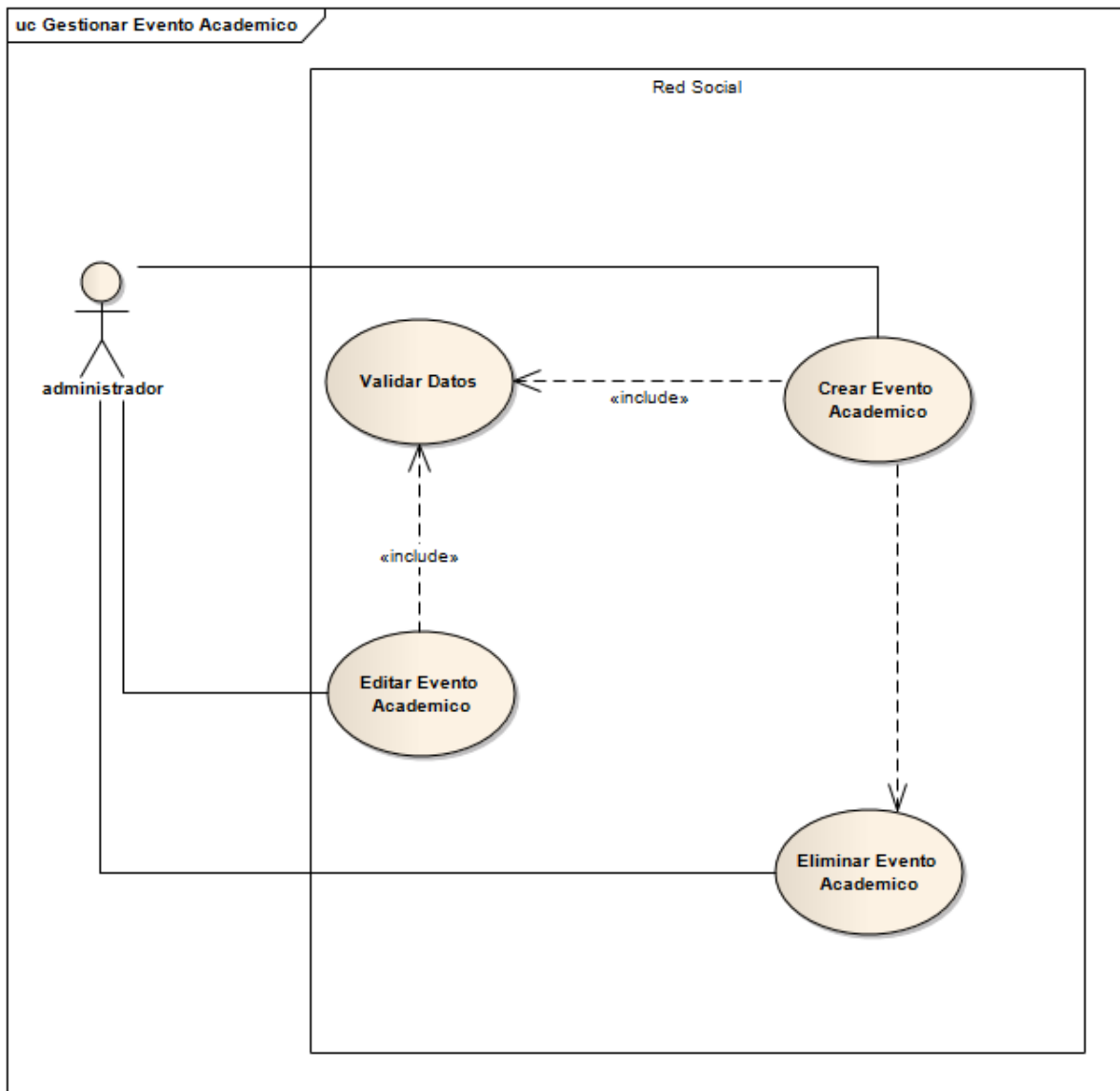
Anexo BP. Caso de uso - mostrar eventos a los que pertenece el egresado

CU-16	Mostrar eventos a los que pertenece el egresado	
Resumen	Este caso de uso inicia cuando el usuario hace click en la opción "eventos a los que estoy invitado". El sistema desplegará un listado de los eventos a los cuales está vinculado el usuario.	
Prioridad	Alta.	
Frecuencia De Uso	Alta.	
Precondiciones	Ninguna.	
Condición Final De Éxito	Se muestra un listado de los eventos a los cuales está vinculado el usuario.	
Condición Final De Fracaso	No se muestra el listado.	
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa la opción mostrar eventos a los cuales estoy invitado.	
Casos Incluidos		
Flujo Principal	Paso	Acción
	1	El usuario hace click en la opción eventos a los cuales estoy invitado.
	2	El sistema verifica los eventos en los cuales el usuario incluido en la lista de invitados
	3	El sistema muestra un listado de los eventos a los cuales el usuario está invitado.
Flujo Alternativo	Paso	Acción Alternativa
	3.1	El usuario no está invitado a ningún evento.
	3.2	El sistema envía un mensaje advirtiendo que el usuario no está invitado a ningún evento.

Anexo BQ. Caso de uso - añadir foto al evento

CU-17	Añadir foto al evento	
Resumen	Este caso de uso inicia cuando el usuario hace click en añadir foto al evento. El sistema desplegara un formulario en el cual el usuario deberá indicar un nombre y una descripción de la imagen requerida para añadir la imagen al evento. Si dicho nombre y descripción son validos y el formato de la foto es soportado por el sistema, añadirá la foto al evento.	
Prioridad	Alta.	
Frecuencia De Uso	Alta.	
Precondiciones	<ul style="list-style-type: none"> - El evento debe existir. - No debe estar añadida otra foto. - El usuario que va a añadir la foto, debe ser el creador del grupo. 	
Condición Final De Éxito	Se añade una foto al evento.	
Condición Final De Fracaso	La foto no es añadida.	
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa el botón añadir foto.	
Casos Incluidos	Validar datos.	
Flujo Principal	Paso	Acción
	1	El usuario hace click en el botón añadir foto.
	2	El sistema despliega un formulario con el nombre la descripción y la foto a añadir.
	3	El usuario llena el formulario y lo envía.
	4 Include::Validar Datos	El sistema valida los datos ingresados.
	5	El sistema añade la foto al evento.
	6	El sistema muestra un mensaje de éxito al usuario.
Flujo Alterno	Paso	Acción Alternativa
	4.1	Existen campos vacíos.
	4.2	La imagen no tiene un formato correcto.
	4.3	El sistema rechaza la solicitud para añadir la foto.
	4.4	El sistema muestra un mensaje de error indicando los campos inválidos.

Anexo BR. Diagrama de caso de uso – gestionar evento académico



Anexo BS. Caso de uso - crear evento académico

CU-18	Crear evento académico	
Resumen	Este caso de uso inicia cuando el usuario hace click en el botón crear evento académico. El sistema desplegara un formulario en el cual el usuario deberá indicar un nombre y una serie de datos requerido para crear el evento académico. Si dicho nombre y campos son validos el sistema creara el evento académico.	
Prioridad	Alta.	
Frecuencia De Uso	Alta.	
Precondiciones	<ul style="list-style-type: none"> - El perfil de usuario debe existir - El usuario a crear el evento académico debe ser un administrador. - No debe existir otro evento con el mismo nombre 	
Condición Final De Éxito	Se crea un nuevo evento	
Condición Final De Fracaso	La solicitud para la creación del evento es rechazada	
Actores Directos	Administrador.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa el botón de crear un evento académico.	
Casos Incluidos	Validar datos.	
Flujo Principal	Paso	Acción
	1	El usuario hace click en el botón crear evento académico.
	2	El sistema despliega un formulario con los campos necesarios para crear el evento.
	3	El usuario llena el formulario y lo envía.
	4 Include::Validar datos	El sistema valida los datos ingresados.
	5	El sistema crea un nuevo evento académico.
	6	El sistema muestra un mensaje de éxito al usuario.
Flujo Alterno	Paso	Acción Alternativa
	4.1	Existen campos vacíos.
	4.2	El nombre del evento académico ya existe.
	4.3	El sistema rechaza la solicitud para crear un nuevo evento académico.
	4.4	El sistema muestra un mensaje de error indicando los campos inválidos.

Anexo BT. Caso de uso - editar evento académico

CU-19	Editar evento académico	
Resumen	Este caso de uso inicia cuando el usuario hace click en el botón editar evento académico. El sistema desplegara un formulario en el que se le mostrara al usuario los campos que se podrán editar del evento en el sistema. Si dicho los nuevos valores de los campos son validos, el sistema actualizara el evento académico.	
Prioridad	Alta.	
Frecuencia De Uso	Alta.	
Precondiciones	<ul style="list-style-type: none"> - Se debe encontrar en el listado de eventos académicos relacionados con el usuario. - El usuario que realiza la modificación debe ser el creador. - El evento debe existir. 	
Condición Final De Éxito	Se modifica y actualiza un evento académico existente.	
Condición Final De Fracaso	La solicitud para la edición del evento académico es rechazada	
Actores Directos	Administrador.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa el botón de editar evento.	
Casos Incluidos	Validar datos, Mostrar eventos académicos creados por el administrador.	
Flujo Principal	Paso	Acción
	1	Include:: Mostrar eventos académicos creados por el administrador.
	2	El usuario hace click en el botón editar evento académico.
	3	El sistema despliega un formulario.
	4	El usuario llena el formulario y lo envía.
	5 Include::Validar Datos	El sistema valida los datos ingresados.
	6	El sistema actualiza los datos del evento académico.
	7	El sistema muestra un mensaje de éxito al usuario.
Flujo Alterno	Paso	Acción Alterna
	5.1	Existen campos vacíos.
	5.2	El sistema rechaza la solicitud para modificar el evento académico.
	5.3	El sistema muestra un mensaje de error indicando los campos inválidos.

Anexo BU. Caso de uso – eliminar evento académico

CU-20	Eliminar evento académico	
Resumen	Este caso de uso inicia cuando el Usuario hace click en el botón Eliminar evento. El sistema mostrara al usuario un mensaje de confirmación para la eliminación del evento; si el usuario acepta el evento será eliminado del sistema.	
Prioridad	Alta.	
Frecuencia De Uso	Casual.	
Precondiciones	<ul style="list-style-type: none"> - Se debe encontrar en el listado de eventos académicos relacionados con el Egresado(usuario) -El Egresado que realiza la eliminación debe ser el creador. -El grupo debe existir. 	
Condición Final De Éxito	Se elimina un evento académico existente.	
Condición Final De Fracaso	La solicitud para la eliminación del evento académico es rechazada	
Actores Directos	Administrador.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa el botón de eliminar evento académico.	
Casos Incluidos	Mostrar eventos académicos creados por el administrador.	
Flujo Principal	Paso	Acción
	1	Include:: Mostrar eventos académicos creados por el administrador.
	2	El usuario hace click en el botón eliminar evento.
	3	El sistema muestra un mensaje de confirmación.
	4	El evento académico es eliminando del sistema.
	5	El sistema muestra un mensaje de éxito al usuario.
	Paso	Acción Alterna
Flujo Alternativo	4.1	El usuario da click en cancelar durante la confirmación.
	4.2	El sistema rechaza la solicitud para eliminar el evento académico.

Anexo BT. Caso de uso - invitar usuario a evento académico

CU-21	Invitar usuario a evento académico	
Resumen	Este caso de uso inicia cuando el usuario hace click en la opción invitar personas. El sistema mostrara al usuario un listado de las personas pertenecientes a su red a las cuales se le podrá enviar una invitación al evento.	
Prioridad	Alta.	
Frecuencia De Uso	Media.	
Precondiciones	<ul style="list-style-type: none"> - El usuario debe tener al menos un amigo. - Se debe estar invitado al evento académico. 	
Condición Final De Éxito	Se realiza una invitación para asistir al evento académico.	
Condición Final De Fracaso	La invitación no se realiza.	
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa la opción invitar amigo.	
Casos Incluidos		
Flujo Principal	Paso	Acción
	1	El usuario hace click en invitar amigo.
	2	El sistema muestra un listado de los amigos en la red social.
	3	El usuario selecciona un amigo del listado.
	4	El sistema envía un mensaje de confirmación.
	5	El usuario da click en aceptar.
	6	El sistema envía la invitación para asistir al evento académico.
	Paso	Acción Alternativa
Flujo Alternativo	4.1	El usuario da click en cancelar durante la confirmación.
	4.2	El sistema rechaza la solicitud para enviar la invitación de asistencia.
	4.3	El sistema no envía la invitación de asistencia al evento académico.

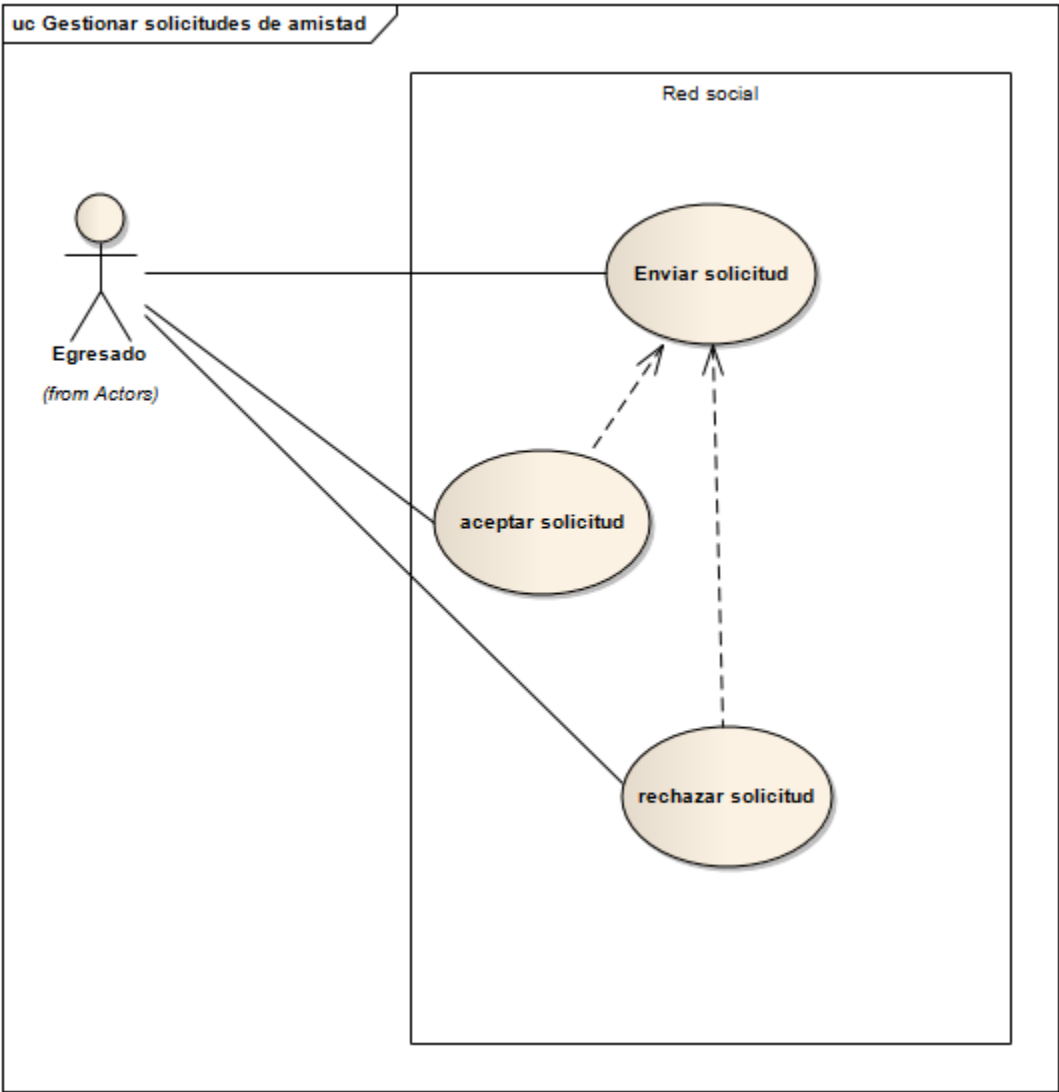
Anexo BU. Caso de uso - mostrar eventos académicos creados por el administrador

CU-22	Mostrar eventos académicos creados por el administrador	
Resumen	Este caso de uso inicia cuando el usuario hace click en la opción “mostrar eventos académico creados”. El sistema desplegara un listado con los grupos creados por el usuario.	
Prioridad	Alta.	
Frecuencia De Uso	Media.	
Precondiciones	Ninguna.	
Condición Final De Éxito	Se muestra un listado con los eventos académicos creados por el usuario.	
Condición Final De Fracaso	No se muestra el listado de los eventos académicos.	
Actores Directos	Administrador.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa la opción mostrar eventos académicos creados.	
Casos Incluidos		
Flujo Principal	Paso	Acción
	1	El usuario hace click en la opción mostrar eventos académico creados.
	2	El sistema verificas los eventos académicos creados por el usuario.
	3	El sistema muestra un listado de los eventos creados por el usuario.
Flujo Alterno	Paso	Acción Alternativa
	3.1	El usuario no ha creado un evento académico.
	3.2	El sistema envía un mensaje advirtiendo que el usuario no ha creado ningún evento académico.

Anexos BV. Casos de uso - mostrar eventos académicos a los que esta invitado el egresado

CU-23	Mostrar eventos académicos a los que esta invitado el egresado	
Resumen	Este caso de uso inicia cuando el usuario hace click en la opción “mostrar eventos académico”. El sistema desplegara un listado de los eventos académicos a los cuales está invitado el usuario.	
Prioridad	Alta.	
Frecuencia De Uso	Media.	
Precondiciones	Ninguna.	
Condición Final De Éxito	Se muestra un listado de los eventos académicos a los cuales está invitado el usuario.	
Condición Final De Fracaso	No se muestra el listado	
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa la opción mostrar eventos académicos.	
Casos Incluidos		
Flujo Principal	Paso	Acción
	1	El usuario hace click en la opción eventos académicos.
	2	El sistema verifica los eventos académicos a los cuales el usuario está invitado.
	3	El sistema muestra un listado de los eventos académicos a los cuales el usuario está invitado.
Flujo Alternativo	Paso	Acción Alternativa
	3.1	El usuario no está vinculado a ningún evento académico.
	3.2	El sistema envía un mensaje advirtiéndole que el usuario no está vinculado a ningún evento.

Anexo BW. Diagrama de caso de caso de uso - gestionar solicitud de amistad



Anexo BX. Caso de uso - enviar solicitud de amistad

CU-24	Enviar solicitud de amistad	
Resumen	Este caso de uso inicia cuando el usuario hace click en la opción enviar solicitud de amistad. El sistema mostrara un campo para redactar un mensaje introductorio. El sistema enviara una invitación a otro usuario.	
Prioridad	Alta.	
Frecuencia De Uso	Alta.	
Precondiciones	<ul style="list-style-type: none"> - Se ha encontrado previamente mediante una búsqueda al egresado que será destinatario de la solicitud de amistad. - No debe existir una invitación previa. 	
Condición Final De Éxito	Se realiza una solicitud de amistad.	
Condición Final De Fracaso	La solicitud no se realiza	
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa la opción enviar solicitud de amistad.	
Casos Incluidos		
Flujo Principal	Paso	Acción
	1	El usuario hace click en enviar solicitud de amistad.
	2	El sistema muestra un campo de texto donde el usuario agrega un mensaje introductorio y opcional para su destinatario.
	3	El usuario da click en enviar.
	4	El sistema envía la solicitud de amistad.
	5	El sistema muestra un mensaje de éxito.
	Paso	Acción Alternativa
Flujo Alternativo	3.1	El usuario da click en cancelar durante.

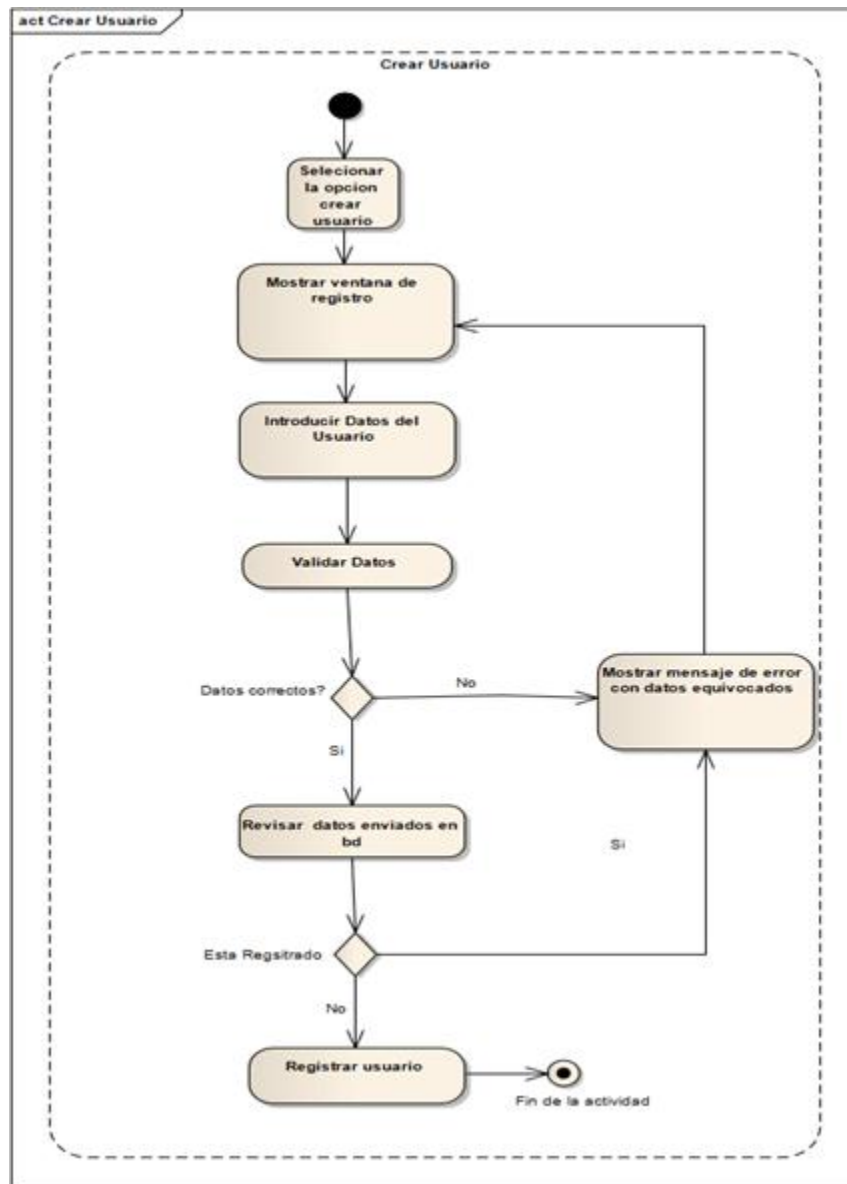
Anexo BY. Caso de uso - aceptar solicitud de amistad

CU-25	Aceptar solicitud de amistad	
Resumen	Este caso de uso inicia cuando el usuario recibe una solicitud de amistad. El sistema mostrara una solicitud de amistad de un destinatario. Junto con un mensaje introductorio. El usuario aceptara la solicitud de amistad del otro usuario y este quedara automáticamente registrado en la lista de sus amigos.	
Prioridad	Alta.	
Frecuencia De Uso	Alta.	
Precondiciones	- Se ha recibido una solicitud de amistad.	
Condición Final De Éxito	Se acepta una solicitud de amistad.	
Condición Final De Fracaso		
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa la opción aceptar solicitud.	
Casos Incluidos		
Flujo Principal	Paso	Acción
	1	El sistema muestra una solicitud de amistad.
	2	El usuario da click en aceptar.
	3	La solicitud de amistad es aceptada.
	4	El sistema muestra un mensaje de éxito de nueva amistad. (Establece una nueva relación).

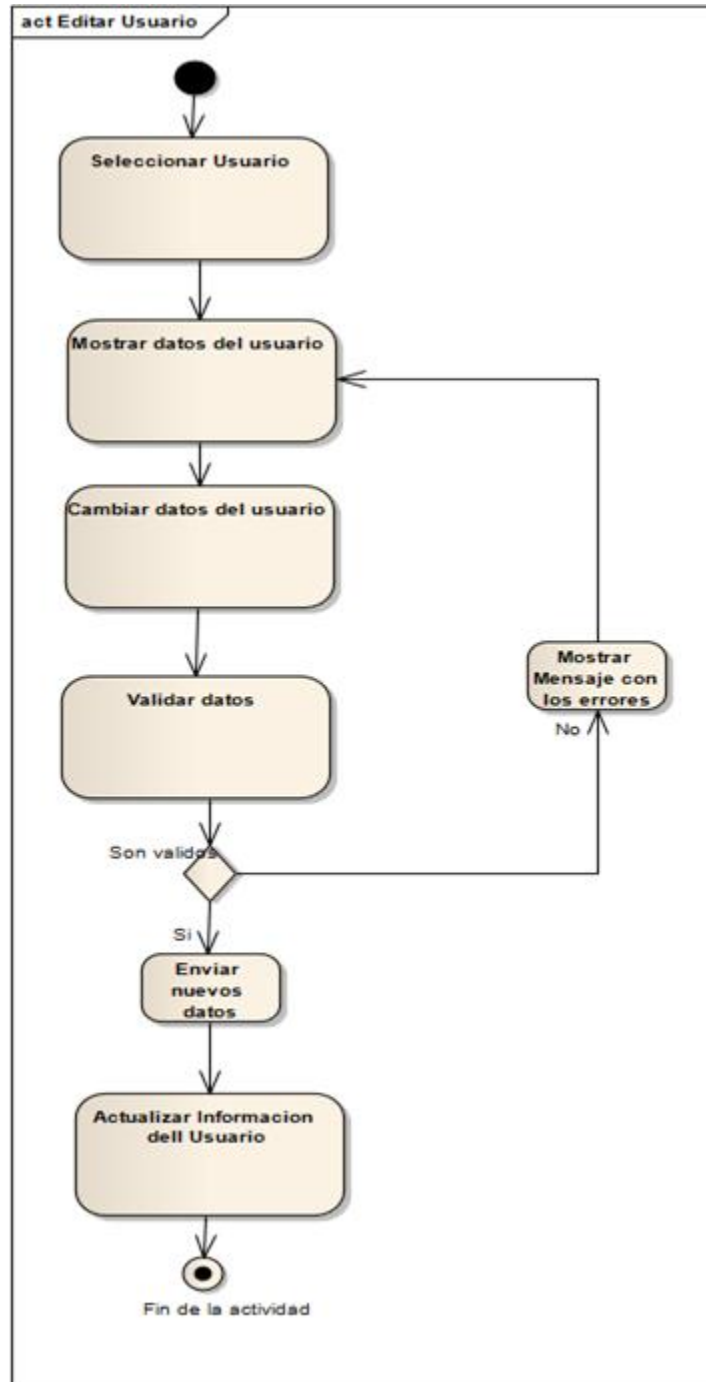
Anexo BZ. Caso de uso - rechazar solicitud de amistad

CU-26	Rechazar solicitud de amistad	
Resumen	Este caso de uso inicia cuando el Usuario Recibe una solicitud de amistad. El sistema mostrara una solicitud de amistad de un destinatario, junto con un mensaje introductorio. El usuario rechazara la solicitud de amistad del otro usuario.	
Prioridad	Alta.	
Frecuencia De Uso	Alta.	
Precondiciones	- Se ha enviado una solicitud de amistad.	
Condición Final De Éxito	Se rechaza una solicitud de amistad.	
Condición Final De Fracaso		
Actores Directos	Egresado.	
Actores Secundarios	Ninguno.	
Disparador	El usuario pulsa la opción rechazar solicitud de amistad.	
Casos Incluidos		
Flujo Principal	Paso	Acción
	1	El sistema muestra una solicitud de amistad.
	2	El usuario da click en rechazar.
	3	LA solicitud de amistad es rechazada.
	4	El sistema muestra un mensaje de solicitud rechazada.

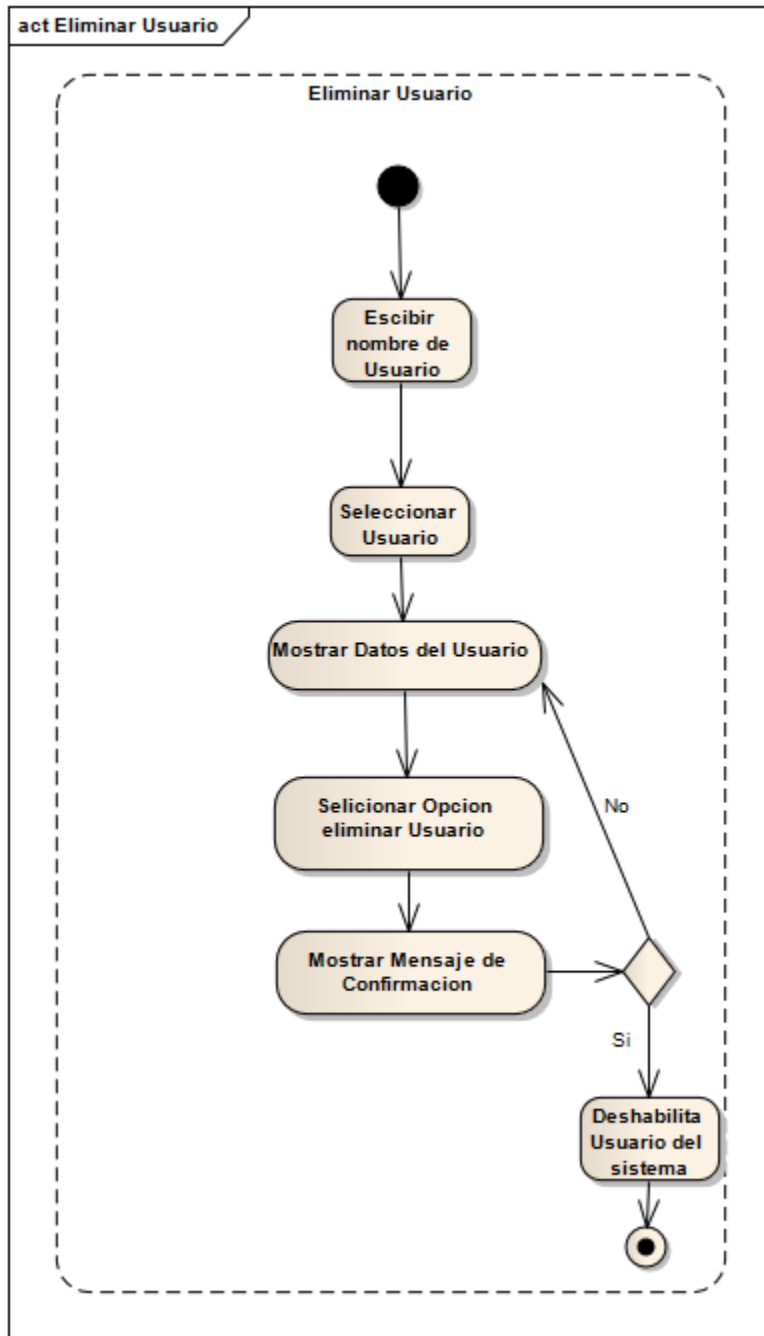
Anexo CA. Diagrama de actividad - crear usuario



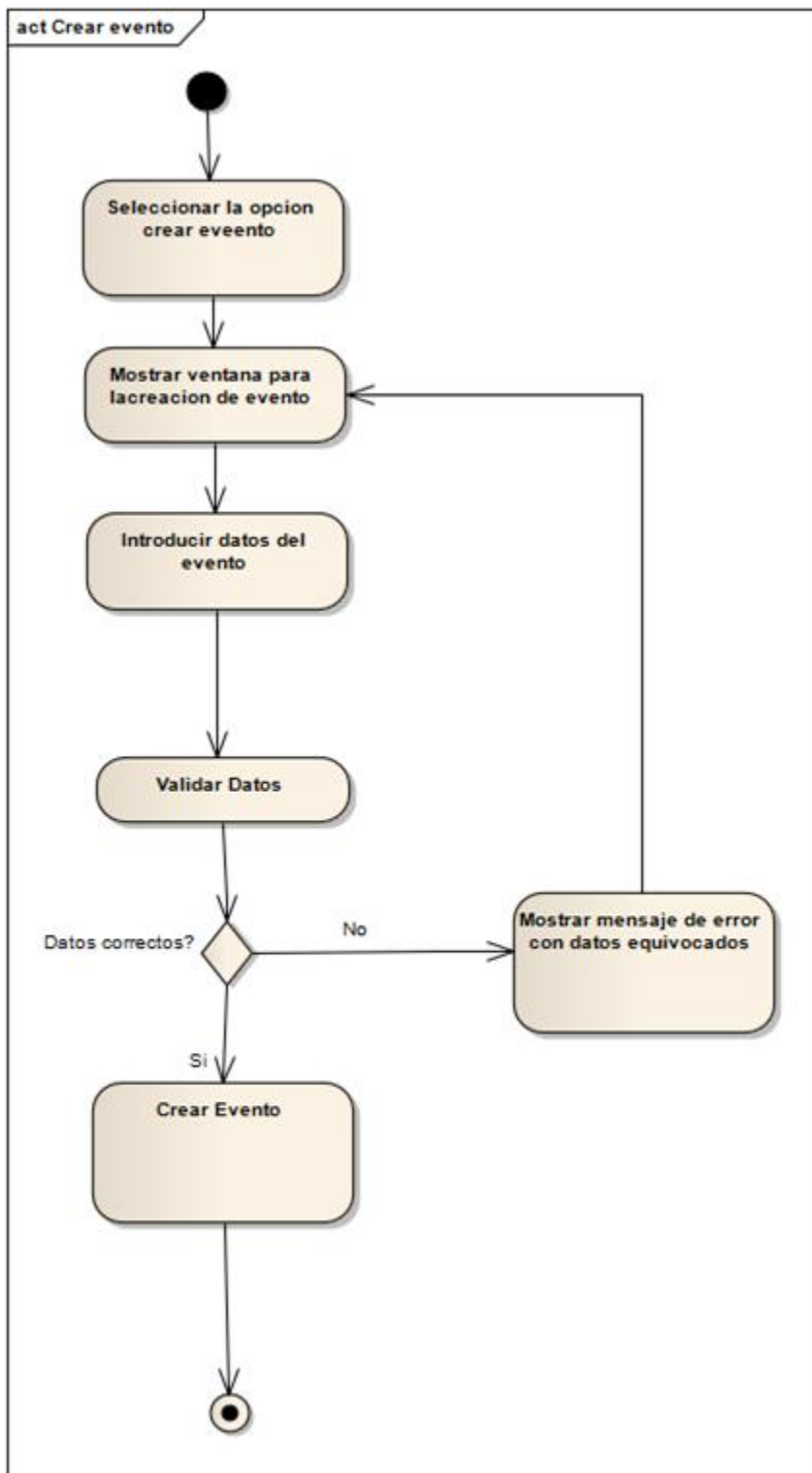
Anexo CB. Diagrama de actividad - editar usuario



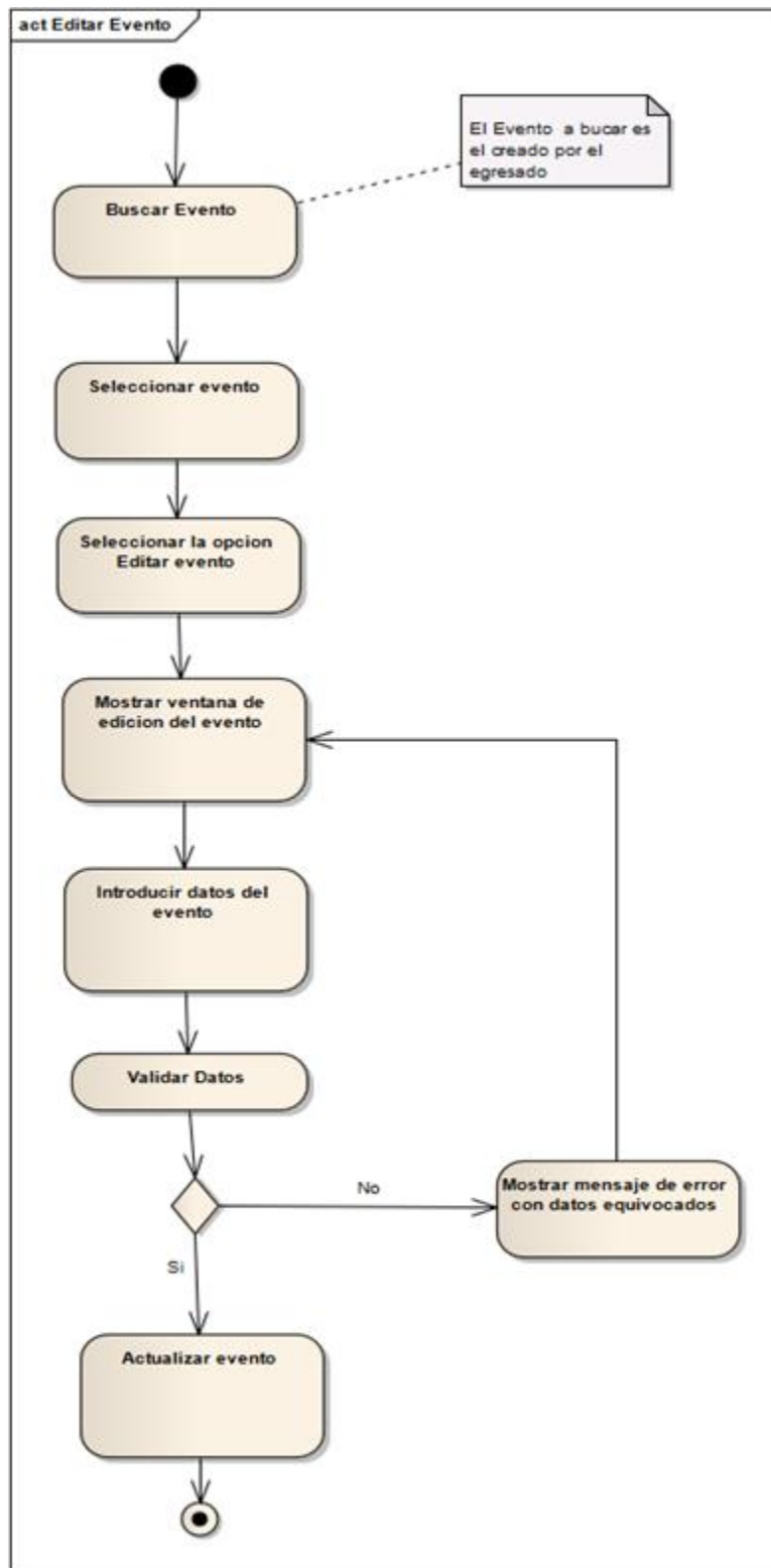
Anexo CC. Diagrama de actividad - eliminar usuario



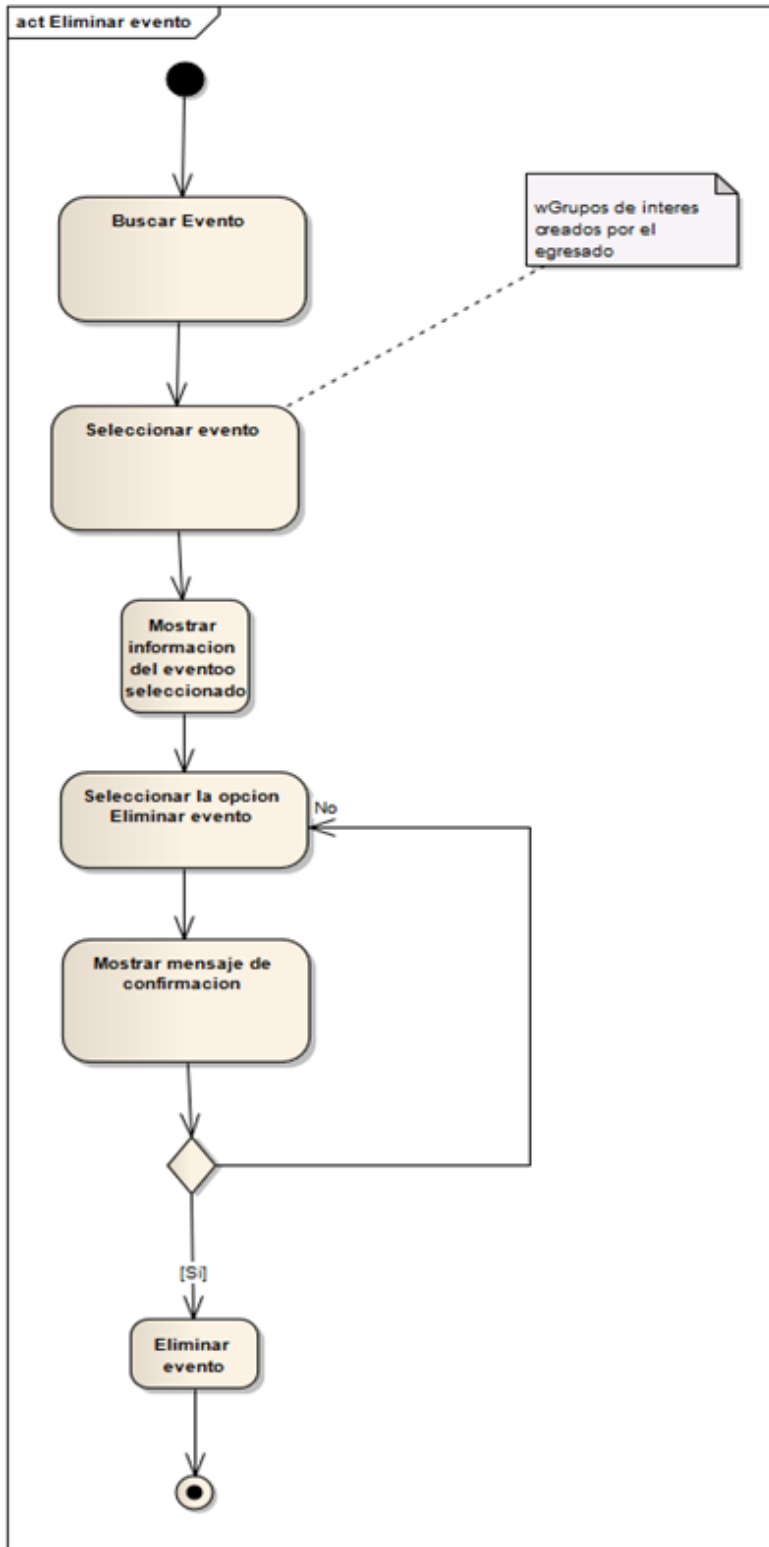
Anexo CD. Diagrama de actividad - crear evento



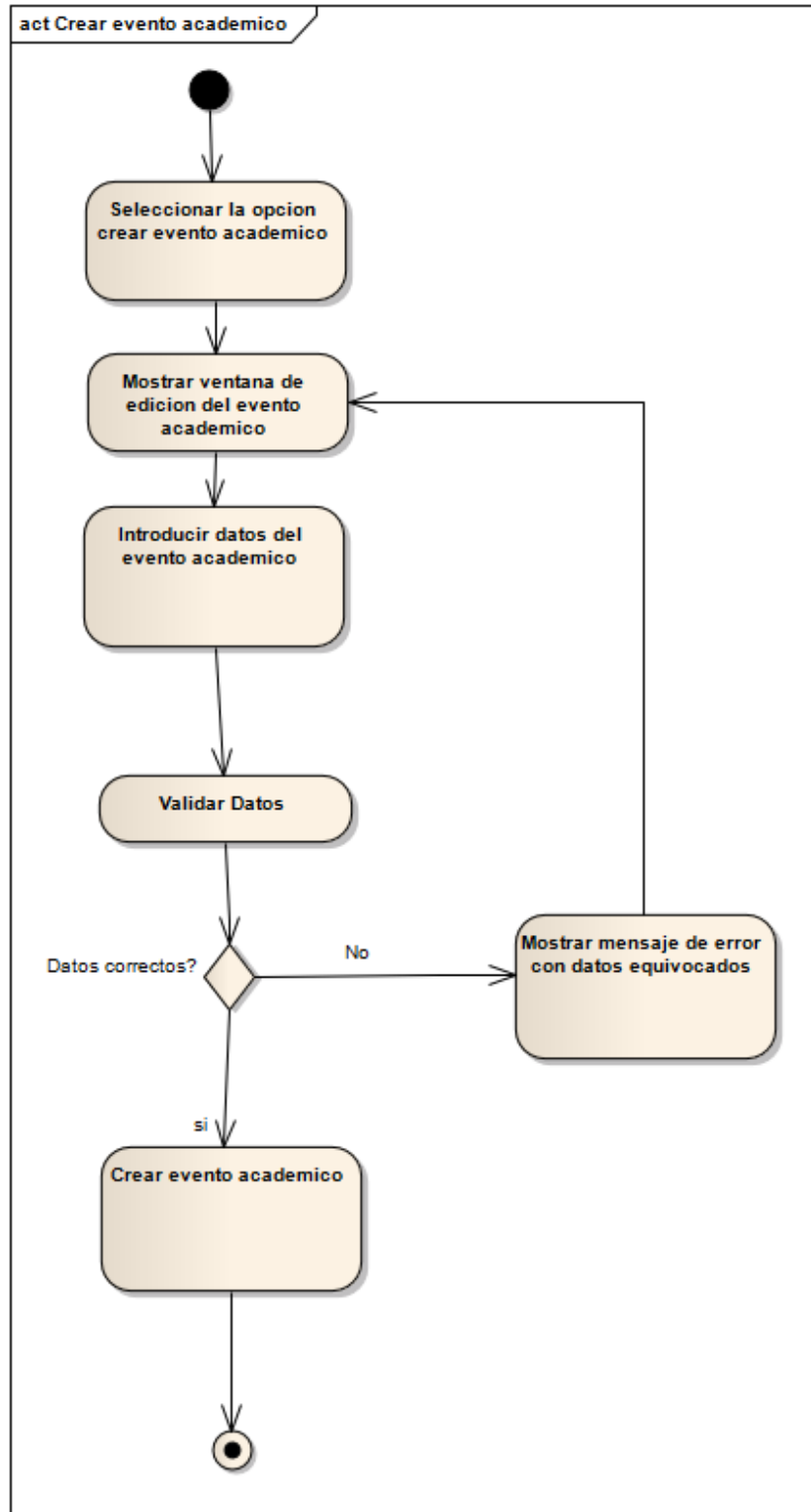
Anexo CE. Diagrama de actividades - editar evento



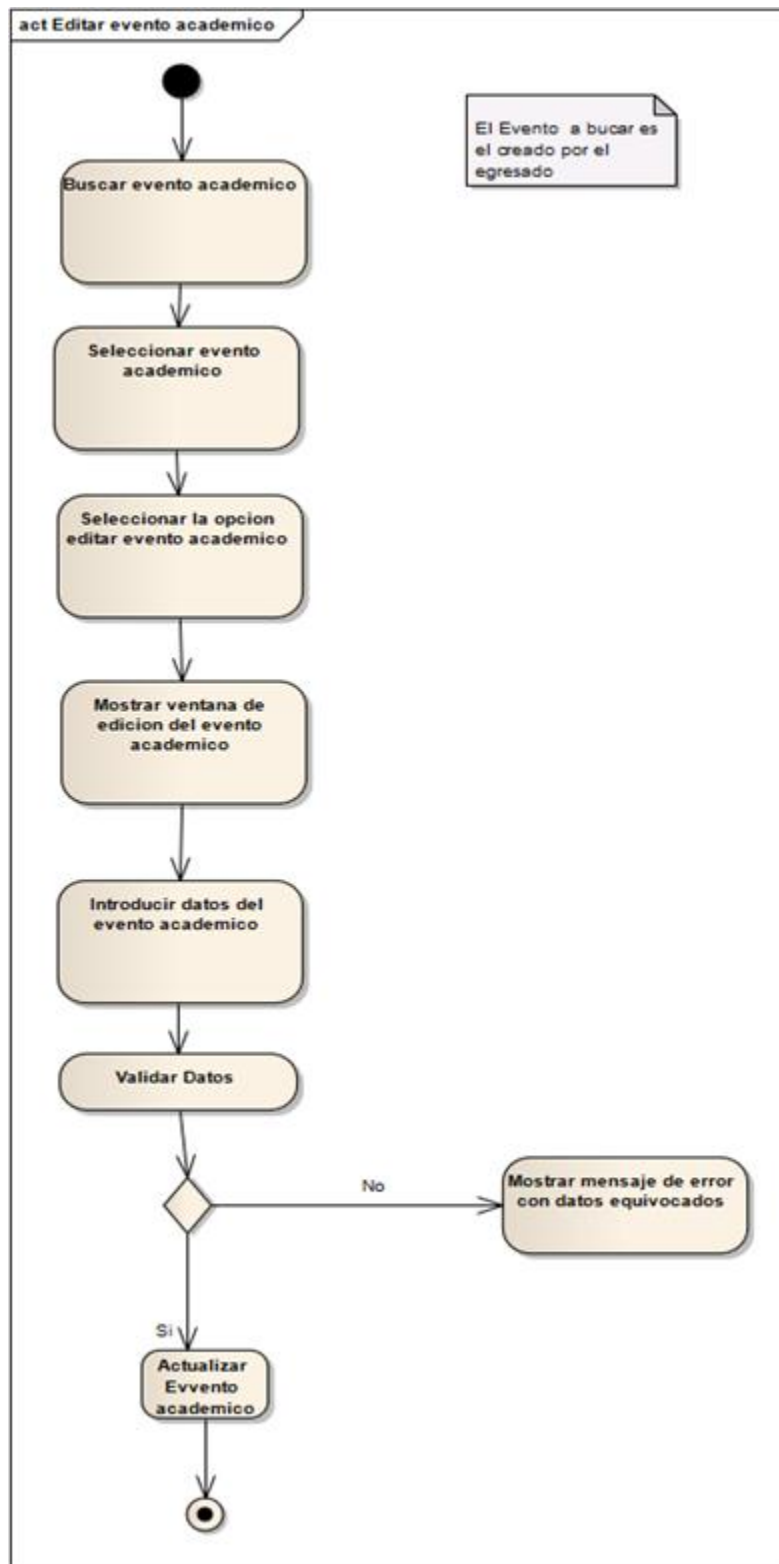
Anexo CF. Diagrama de actividad - eliminar evento



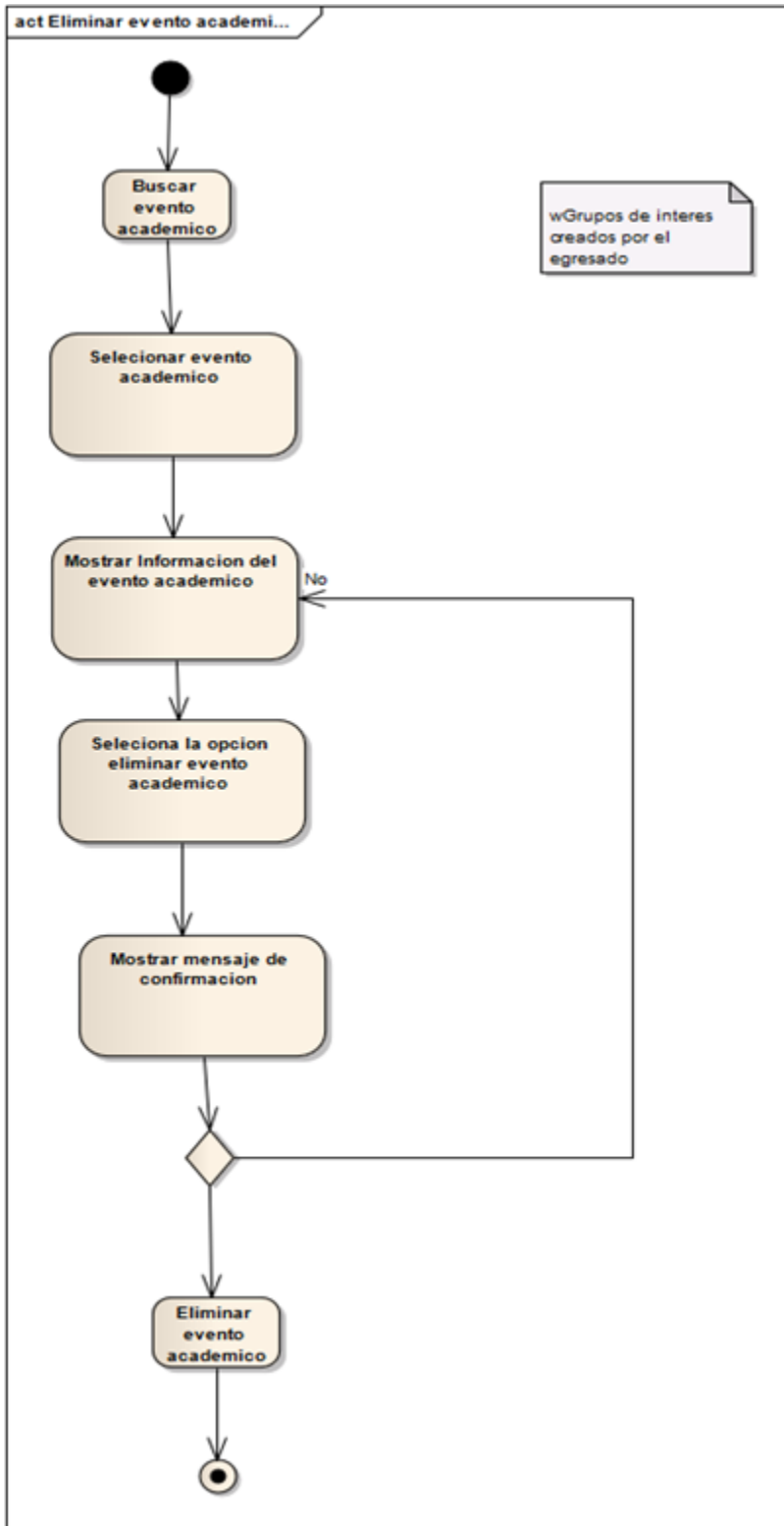
Anexo CG. Diagrama de actividad - crear evento academico



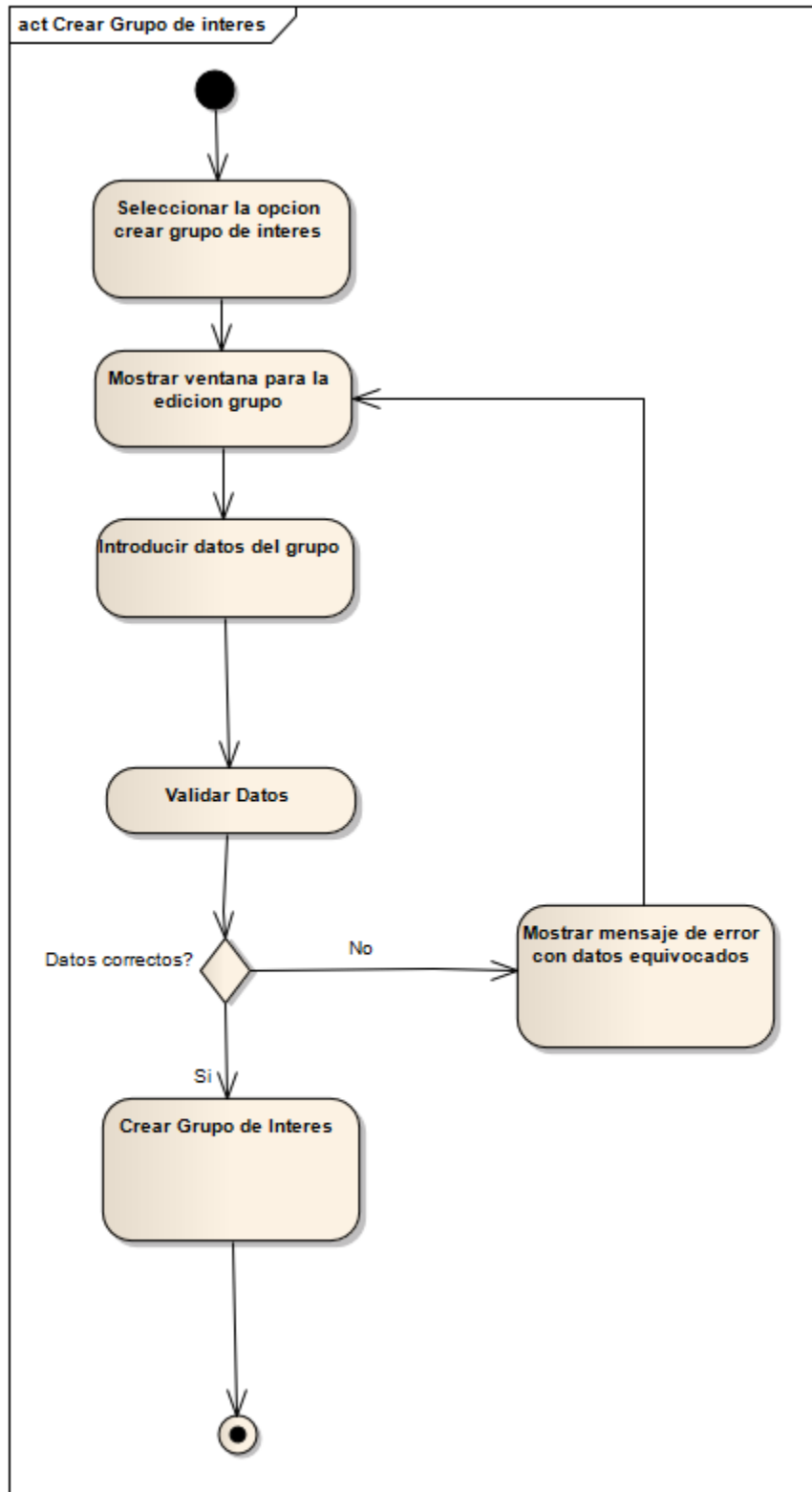
Anexo CH. Diagrama de actividad - editar evento academico



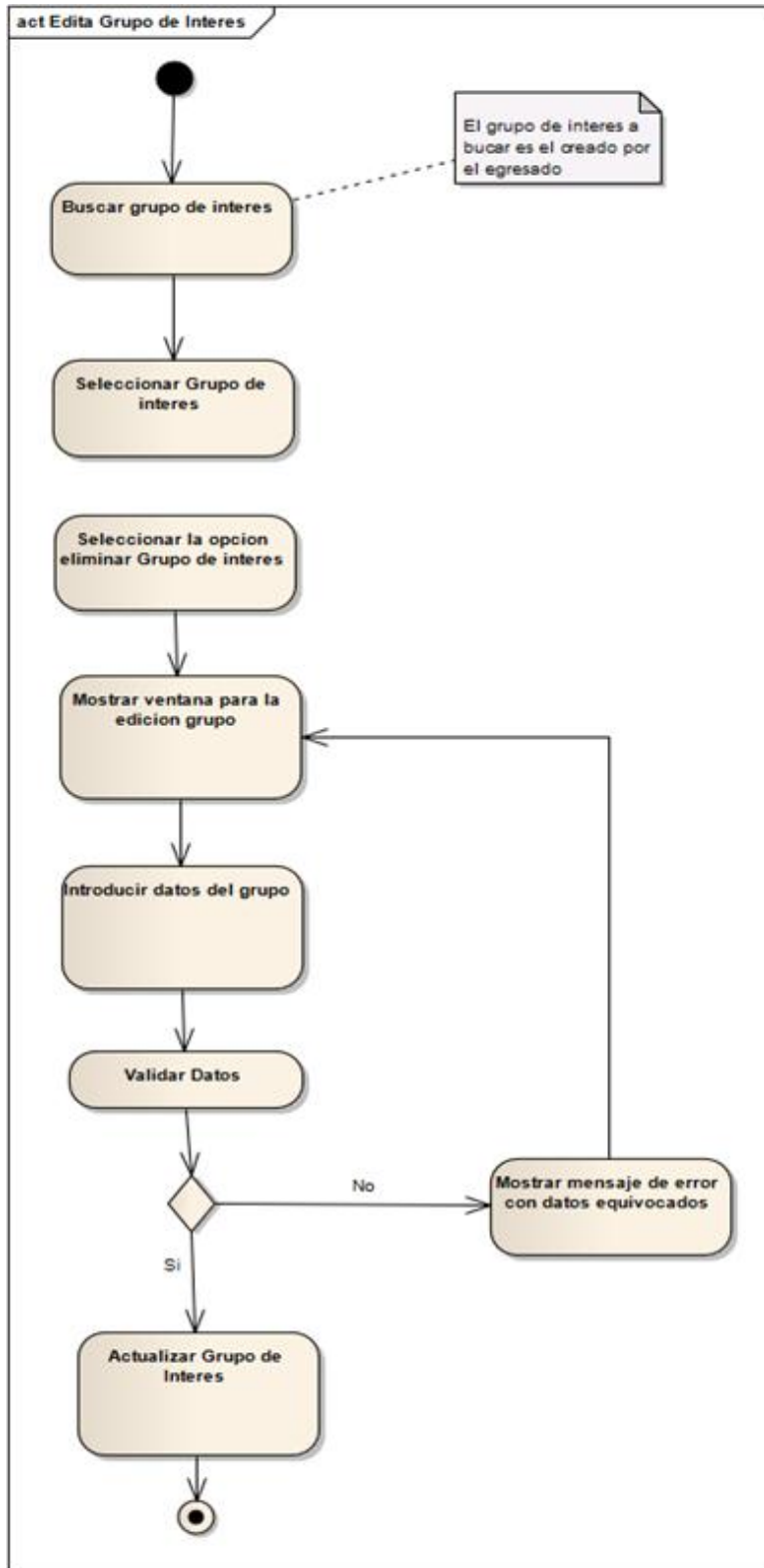
Anexo CI. Diagrama de actividad - eliminar evento academico



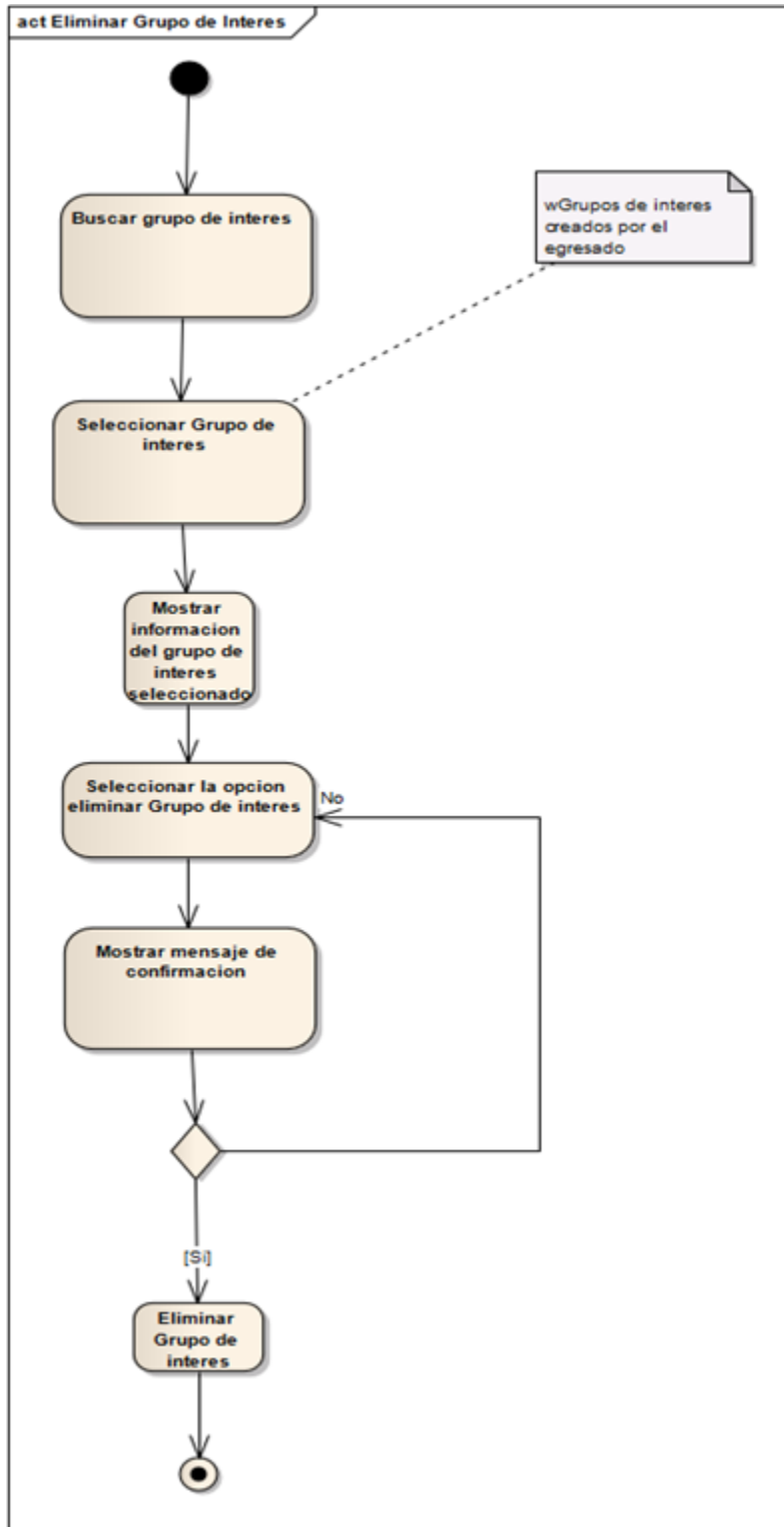
Anexo CJ. Diagrama de actividad - crear grupo de interes



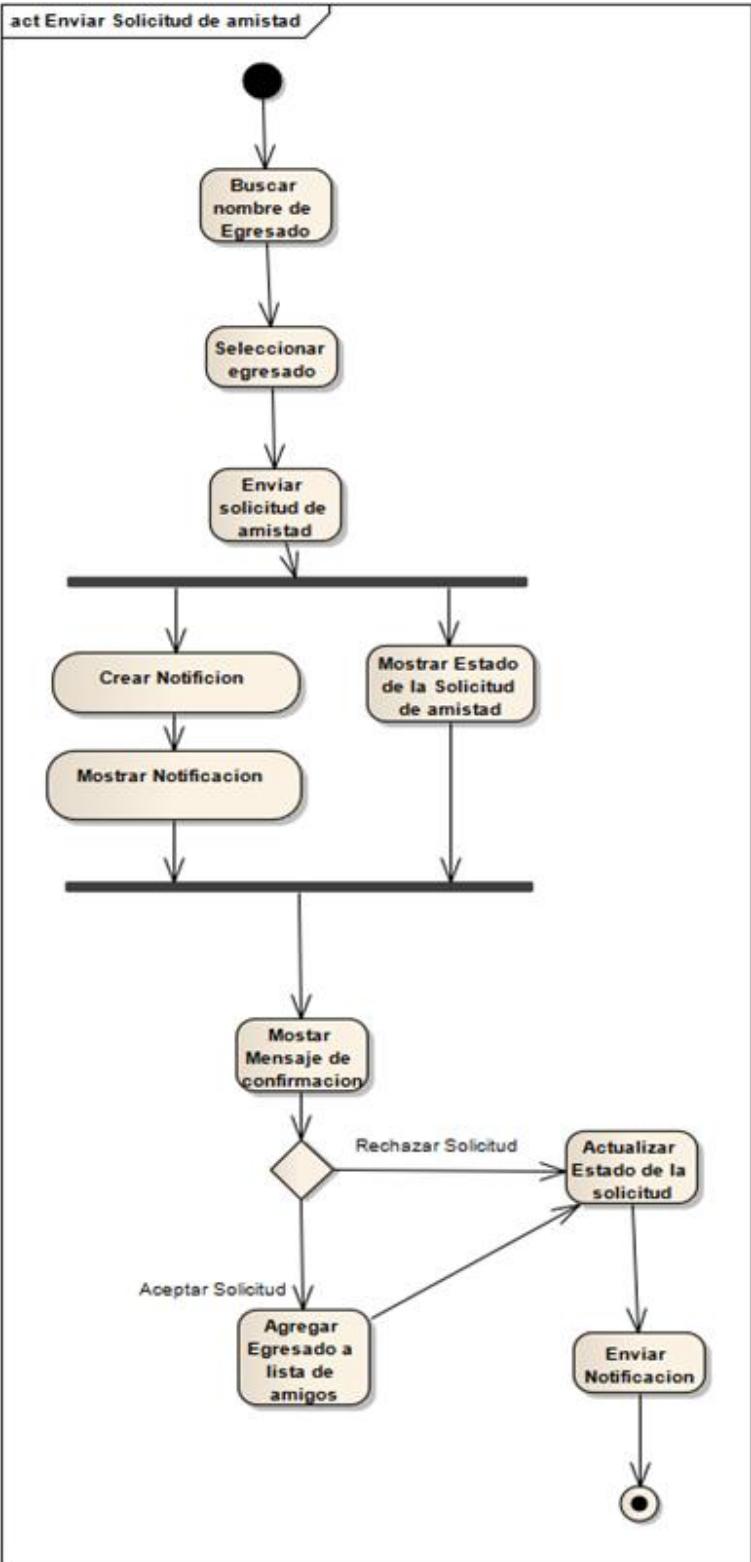
Anexo CK. Diagrama de actividad - editar grupo de interes



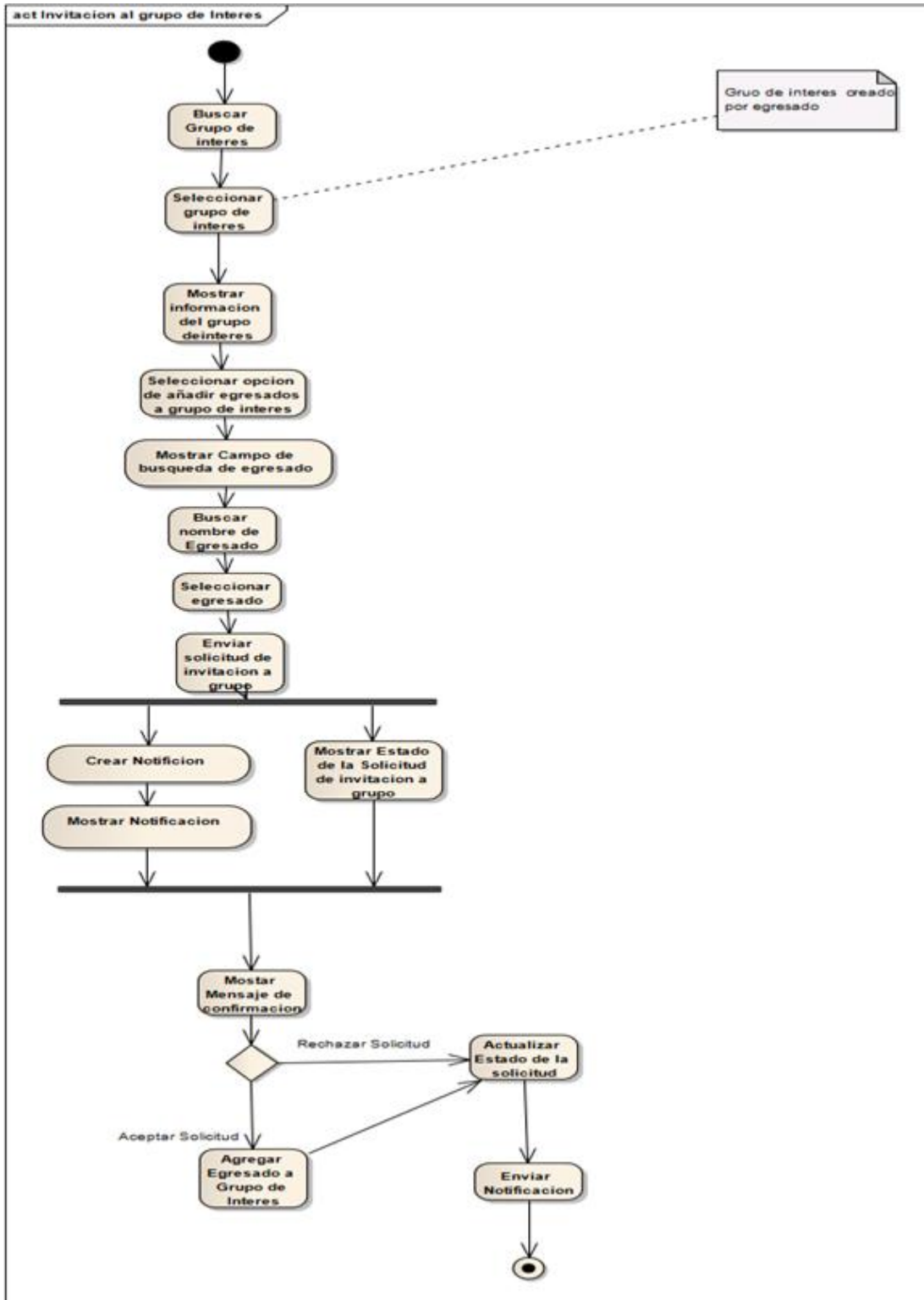
Anexo CL. Diagrama de actividad - eliminar grupo de interes



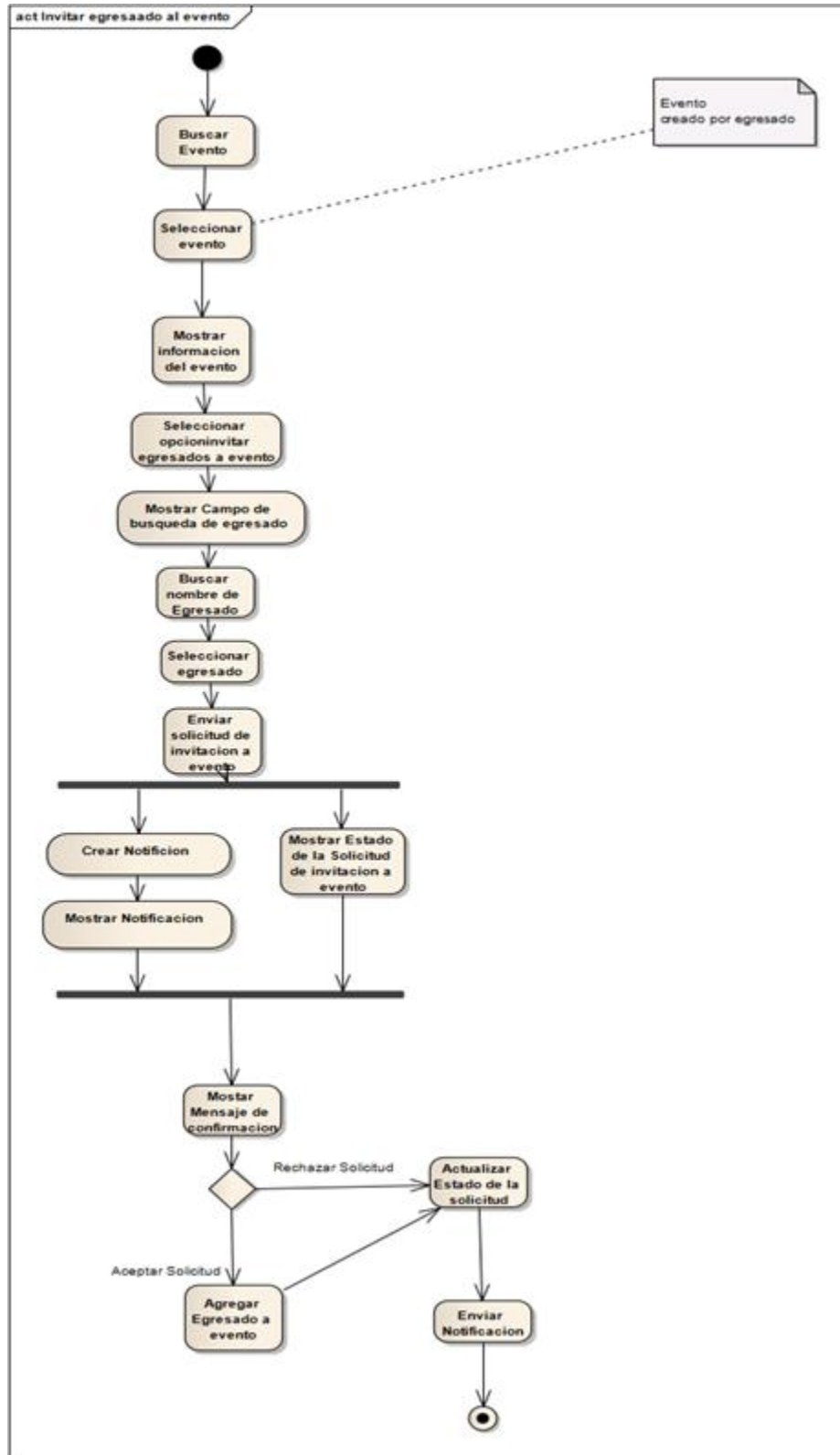
Anexo CM. Diagrama de actividad - enviar solicitud de amistad



Anexo CN. Diagrama de actividad - invitar egresado a grupo de interes



Anexo CO. Diagrama de actividad - invitar egresado a evento



Anexo CP. Diagrama de actividad invitar egresado a evento academico

