



Universidad del Azuay

Facultad de Ciencias de la Administración

Escuela de Ingeniería en Sistemas y Telemática

**RECOMENDADOR MÓVIL SENSIBLE AL  
CONTEXTO DE NOTICIAS Y EVENTOS  
UNIVERSITARIOS EMPLEANDO  
TECNOLOGÍAS SEMÁNTICAS**

Trabajo de graduación previo a la obtención del título de Ingeniero en  
Sistemas y Telemática

Autores:

**Nicolás Oswaldo Andrade Hidalgo.**

**Juan Diego Torres Argudo.**

Director:

**Mgt. Lenin Erazo Garzón.**

**Cuenca – Ecuador**

**2018**

## **DEDICATORIA**

Este trabajo de tesis va dedicado a nuestros padres que nos apoyaron con sus consejos y han estado a nuestro lado desde que comenzamos nuestras carreras como estudiantes, y sin su apoyo no podríamos cumplir con esta meta de nuestras vidas.

## **AGRADECIMIENTO**

Queremos brindar un especial agradecimiento a nuestro director de tesis el Ing. Lenin Erazo por ser la persona que confió en nosotros para realizar esta tesis de grado, también al Ing. Andrés Patiño quien nos apoyó durante las investigaciones del tema y el desarrollo del trabajo final.

## **RESUMEN**

En la actualidad el avance de las aplicaciones móviles permite a los usuarios realizar una infinidad de funciones en todo momento y lugar, siendo no la excepción el dominio universitario, sin embargo aún es incipiente el uso del contexto, como un aspecto para mejorar la experiencia del usuario. El propósito del presente trabajo es desarrollar un recomendador móvil sensible al contexto de noticias y eventos académicos, empleando ontologías y tecnologías semánticas. Para ello se realizó una revisión bibliográfica del dominio de estudio, se especificaron los requerimientos del software, se construyó la ontología y se diseñó, codificó e implantó el recomendador. Como resultado se obtuvo una aplicación web para mantener la información a recomendar, y una aplicación móvil para su visualización.

## ABSTRACT

The development of mobile applications allows users to perform an infinity of functions at any time and place, without being the exception the university domain. However, the use of context as an aspect to improve the user experience has been incipient. The purpose of this work was to develop a mobile recommender sensitive to the context of news and academic events, using ontologies and semantic technologies. For this, a bibliographic review of the study domain was carried out. The software requirements were specified and the ontology was built. Finally, the recommender was designed, codified and implemented. As results, a web application was obtained to maintain the information to be recommended along with a mobile application for display.

A handwritten signature in blue ink, which appears to be "Paul Arpi".

Translated by

Ing. Paul Arpi

## INTRODUCCIÓN

El uso de las aplicaciones móviles en la actualidad gracias al avance tecnológico se ha vuelto imprescindible para las personas. En este sentido, esta investigación pretende hacer uso de estas aplicaciones móviles integrando la sensibilidad al contexto para brindar información sobre eventos y noticias oportunos de acuerdo con la situación del entorno.

El uso de las aplicaciones móviles en el dominio universitario, no ha sido la excepción, se ha extendido notablemente a fin de brindar servicios de calidad e información oportuna a sus principales actores (estudiantes, profesores); sin embargo, en este tipo de aplicaciones resulta esencial considerar la información contextual para enriquecer la experiencia del usuario, mediante la provisión de información relevantes con su situación actual.

La elección de la presente investigación se ha visto motivada desde dos perspectivas: primero, apoyar al ente universitario a través de una herramienta tecnológica “inteligente” para la difusión y recomendación personalizada de noticias y/o eventos universitarios basada en información contextual disponible, y dirigida a estudiantes y profesores, como por ejemplo: perfil del usuario, carrera y nivel de estudio, preferencias, localización, temporalidad; y el segundo punto, el interés propio sobre los aplicativos móviles con sensibilidad al contexto y las tecnologías semánticas, debido a que son un campo de investigación en pleno desarrollo.

Particularmente, en el ámbito de la difusión de noticias y eventos universitarios a través de portales web, aplicaciones móviles, SMS, entre otros, los usuarios se ven abrumados por la gran cantidad de información de carácter general que reciben, generando insatisfacción en su uso y disminuyendo el impacto de este tipo de medios de difusión. De ahí la necesidad de un sistema móvil sensible al contexto que apoye en la difusión de noticias y eventos que sean plenamente del interés personal para los estudiantes y profesores de conformidad con su contexto actual.

Este trabajo de investigación está conformado por siete capítulos. El capítulo 1 profundiza los conceptos de computación ubicua, contexto y sus tipos, así como también un estudio de la computación sensible al contexto, y por último se describen varias propuestas de modelos ontológicos para representar el contexto.

En el capítulo 2 se realiza el estudio de las metodologías existentes en la literatura para el modelado de ontologías, además de las tecnologías semánticas para la implementación de ontologías, tales como: XML, RDF, OWL, SPARQL, FUSEKI, entre otros.

El capítulo 3 incluye la especificación de requisitos del software tanto de la aplicación web cuyo propósito es la administración de la información básica para el funcionamiento del recomendador móvil, como de la aplicación móvil encargada de presentar las recomendaciones sensibles al contexto sobre noticias y eventos académicos.

El capítulo 4 tiene relación con la construcción de una red de ontologías que permite representar el contexto, específicamente describe los requisitos que debe cumplir la ontología para el sistema recomendador y el proceso de modelado de la ontología.

El capítulo 5 describe todo lo relacionado con el diseño del software, es decir, diseño arquitectónico (modelo-vista-controlador), diseño de la base de datos relacional, el diagrama de clases, diagramas de secuencia y el diseño de las interfaces de la aplicación web y móvil.

El capítulo 6 detalla la configuración del servidor SPARQL (FUSEKI) y del motor de inferencia encargados de gestionar y almacenar la ontología, así como de inferir conocimiento a partir de ella, respectivamente. Luego se describe la implementación de los servicios web tipo REST que incluyen los algoritmos de consulta de las recomendaciones de noticias y/o eventos mediante la ontología. Por último se describe la evaluación del software en un escenario real universitario.

Finalmente, se incluyen las conclusiones de la investigación y los posibles trabajos futuros.

# ÍNDICE

## Índice de contenido.

1.	Estado del arte .....	14
1.1	Introducción .....	14
1.2	Computación ubicua .....	15
1.3	Contexto .....	15
1.4	Tipos de contexto .....	16
1.5	Computación Sensible al contexto .....	17
1.6	Ontología.....	19
1.7	Modelos ontológicos para representar el contexto.....	20
1.7.1	COBRA-ONT .....	20
1.7.2	SOUPA.....	21
1.7.3	CONON .....	24
1.7.4	CoDAMoS .....	25
1.7.5	CACOnt .....	26
1.7.6	mIO!.....	30
1.7.7	Análisis comparativo de los modelos ontológicos estudiados .....	32
1.8	Otras ontologías relacionadas con el dominio de estudio .....	33
1.8.1	FOAF (Friend of a Friend).....	33
1.8.2	AIISO (Academic Institution Internal Structure Ontology) .....	34
1.8.3	OWL-Time.....	34
1.8.4	WGS84 Geo Positioning.....	37
2.	Métodos y Herramientas .....	38
2.1	Metodología NeOn para construcción de ontologías .....	38
2.1.1	Introducción .....	38
2.1.2	Escenarios .....	38
2.1.3	Actividades.....	39
2.1.4	Conclusión .....	43
2.2	Otras Metodologías .....	43
2.2.1	Metodología Methontology.....	43
2.2.2	Metodología On-To-Knowledge.....	44
2.2.3	Metodología Diligent .....	45
2.3	Tecnologías semánticas.....	46



2.3.1	XML.....	46
2.3.2	RDF.....	49
2.3.3	OWL.....	53
2.3.4	SPARQL .....	58
2.3.5	Apache Marmotta.....	59
2.3.6	API Jena y FUSEKI.....	60
3.	Análisis del software.....	61
3.1	Especificación de requerimientos de software (ERS).....	61
3.1.1	Propósito .....	61
3.1.2	Ámbito del Software .....	61
3.1.3	Definiciones, acrónimos y abreviaturas .....	61
3.1.4	Referencias.....	62
3.2	Descripción general.....	62
3.2.1	Características de los usuarios .....	62
3.2.2	Restricciones .....	63
3.2.3	Suposiciones y dependencias .....	63
3.3	Requerimientos funcionales.....	63
3.3.1	Diagrama de casos de uso de control de acceso.....	63
3.3.2	Descripción de casos de uso de control de acceso .....	64
3.3.3	Diagrama de casos de uso de la aplicación web .....	65
3.3.4	Descripción de casos de uso generales de la aplicación web.....	66
3.3.5	Descripción de casos de uso específicos de la aplicación web.....	69
3.3.6	Diagrama de caso de usos de la aplicación móvil.....	70
3.3.7	Descripción de casos de uso de la aplicación móvil .....	70
4.	Construcción de una red de ontologías para representar contexto.....	77
4.1	Especificación de requerimientos de la ontología.....	77
4.2	Modelamiento de la ontología.....	80
4.2.1	Definir los escenarios.....	80
4.2.2	Búsqueda de recursos ontológicos .....	80
4.2.3	Comparación y selección de ontologías .....	81
4.2.4	Fusión de ontologías .....	82
4.2.5	Búsqueda de recursos no ontológicos .....	82
4.2.6	Transformación de recursos no ontológicos .....	82
4.2.7	Ontología Resultante.....	82

5.	Diseño del software.....	86
5.1	Diseño de arquitectónico.....	86
5.1.1	Aplicación móvil.....	87
5.1.2	Aplicación web .....	87
5.2	Diseño de la base de datos relacional.....	88
5.3	Diseño orientado a objetos .....	94
5.3.1	Diagramas de clases del sistema administrativo .....	94
5.3.2	Diagramas de secuencia de sistema administrativo .....	95
5.3.3	Diagrama de clases de la aplicación móvil .....	98
5.3.4	Diagramas de secuencia de la aplicación móvil.....	99
5.4	Diseño de interfaces de usuario de la aplicación web.....	104
5.5	Diseño de interfaces de la aplicación móvil.....	107
6.	Implementación y pruebas del software.....	112
6.1	Configuración servidores .....	112
6.1.1	Configuración de razonador y FUSEKI.....	112
6.2	Servicios de consulta de información en la red ontológica.....	115
6.2.1	Descripción procedimental de los servicios web para consulta de información.....	115
6.3	Evaluación del sistema recomendador móvil de eventos y noticias académicas.....	117
6.3.1	Pruebas de unidad .....	117
6.3.2	Pruebas de integración .....	118
7.	Conclusiones y trabajo futuro .....	120
	Bibliografía.....	122

## Índice de tablas y figuras

### Tablas

Tabla 1 Lista de clases y propiedades CORBA-ONT 0.2	20
Tabla 2 Comparativa de Ontologías	32
Tabla 3 Definición de requerimientos de la ontología	40
Tabla 4 Comparación entre XML y Web Semántica a varios niveles	49
Tabla 5 Definiciones, acrónimos y abreviaturas.	61
Tabla 6 Referencias de especificación de requerimientos	62
Tabla 7 Usuarios que intervienen en el sistema	62
Tabla 8 Caso de uso de iniciar sesión	64
Tabla 9 Caso de uso de cerrar sesión	64
Tabla 10 Caso de uso de agregar nuevos registros	66
Tabla 11 Caso de uso de modificar registros existentes	67
Tabla 12 Caso de uso de eliminar registros existentes	68
Tabla 13 Caso de uso de consultar registros	68
Tabla 14 Caso de uso de gestionar instituciones	69
Tabla 15 Caso de uso de configuración de parámetros de la aplicación	70
Tabla 16 Caso de uso de proporcionar coordenadas de la ubicación del usuario	71
Tabla 17 Caso de uso de recomendar eventos o noticias por áreas de conocimiento	72
Tabla 18 Caso de uso de recomendar eventos o noticias por tiempo	73
Tabla 19 Caso de uso de recomendar eventos por tipo	74
Tabla 20 Caso de uso de recomendar eventos por lugar	75
Tabla 21 Caso de uso de mostrar ubicación de evento	75
Tabla 22 Caso de uso de mostrar noticia detallada	76
Tabla 23 Especificación de requerimientos de la ontología sección 1 a 5	77
Tabla 24 Especificación de requerimientos de la ontología sección 6	78
Tabla 25 Especificación de requerimientos de la ontología sección 7	79
Tabla 26 Ontologías encontradas organizadas en categorías	81
Tabla 27 Clases de la ontología	83
Tabla 28 Object Property de la ontología	84
Tabla 29 Data Property de la Ontología	85
Tabla 30 Diccionario de datos de la base de datos	90
Tabla 31 Resultados de prueba de muestra	119
Tabla 32 Caso de uso de gestionar facultades	127
Tabla 33 Caso de uso de gestionar escuelas	127
Tabla 34 Caso de uso de gestionar departamentos	128
Tabla 35 Caso de uso de gestionar lugares	129
Tabla 36 Caso de uso de gestionar personas	130
Tabla 37 Caso de uso de gestionar asignaturas	130
Tabla 38 Caso de uso de gestionar cursos	131
Tabla 39 Caso de uso de gestionar distributivos académicos	132
Tabla 40 Caso de uso de gestionar escuelas	133
Tabla 41 Caso de uso de gestionar campos amplios	134
Tabla 42 Caso de uso de gestionar campos específicos	134
Tabla 43 Gestionar campos detallados	135
Tabla 44 Gestionar tipos de usuarios	136
Tabla 45 Caso de uso de gestionar usuarios	137

Tabla 46 Caso de uso de gestionar eventos	137
Tabla 47 Caso de uso de gestionar tipos de evento	138
Tabla 48 Caso de uso de gestionar noticias	139
Tabla 49 Caso de uso de gestionar horarios	140
Tabla 50 Caso de uso de gestionar intereses de escuelas	140
Tabla 51 Caso de uso de gestionar intereses de personas	141
Tabla 52 Caso de uso de gestionar intereses de eventos	142
Tabla 53 Caso de uso de gestionar intereses de noticias	143

## **Figuras**

Figura 1 Ontología SOAP	22
Figura 2 Ontología CONON	24
Figura 3 Ontología CoDAMoS	26
Figura 4 Ontología CACOnt	27
Figura 5 Modelo User CACOnt	27
Figura 6 Modelo Device CACOnt	28
Figura 7 Modelo Service CACOnt	29
Figura 8 Modelo Space CACOnt	29
Figura 9 Modelo Environment CACOnt	30
Figura 10 Ontología mio!	31
Figura 11 Clase TemporalEntity	35
Figura 12 Clase TemporalPosition	36
Figura 13 Clase Duration	37
Figura 14 Pasos para definir requisitos de una ontología con la metodología NeOn	41
Figura 15 Pasos para realizar el plan y modelado de la ontología con la metodología NeOn	42
Figura 16 Conjunto de RDF statements (RDF dataset)	50
Figura 17 Propiedad inversa entre profesor-curso	56
Figura 18 Clasificación de actores	62
Figura 19 Diagrama de casos de uso de control de acceso	63
Figura 20 Diagrama de casos de uso generales de la aplicación web	65
Figura 21 Diagrama de casos de uso específicos de la aplicación web	66
Figura 22. Diagrama de casos de uso de la aplicación móvil.	70
Figura 23 Ontología resultante	83
Figura 24 Modelo Vista Controlador	86
Figura 25 Diagrama entidad relación de la base de datos	89
Figura 26 Diagrama de clases del sistema administrativo	94
Figura 27 Diagrama de secuencia para consultar eventos	95
Figura 28 Diagrama de secuencia para ingresar eventos	96
Figura 29 Diagrama de secuencia para modificar eventos	97
Figura 30 Diagrama de secuencia para eliminar eventos	98
Figura 31 Diagrama de clases de la aplicación móvil	99
Figura 32 Diagrama de secuencia de inicio de sesión	99
Figura 33 Diagrama de secuencia de configuración de parámetros	100
Figura 34 Diagrama de secuencia para proporcionar ubicación	100
Figura 35 Diagrama de secuencia para obtener eventos por área de conocimiento	101
Figura 36 Diagrama de secuencia para obtener noticias por área de conocimiento	101

Figura 37 Diagrama de secuencia para obtener eventos por tiempo	102
Figura 38 Diagrama de secuencia para obtener noticias por tiempo	102
Figura 39 Diagrama de secuencia para obtener eventos por tipo	103
Figura 40 Diagrama de secuencia para obtener eventos por lugar	103
Figura 41 Diagrama de secuencia para abrir un evento	103
Figura 42 Diagrama de secuencia para abrir una noticia	104
Figura 43 Diseño y resultado final de la pantalla principal	105
Figura 44 Diseño y resultado final de pantalla de mantenimiento	106
Figura 45 Pantalla de inicio de sesión	107
Figura 46 Pantalla principal	108
Figura 47 Menú de la aplicación	108
Figura 48 Pantalla de tipos de Eventos	109
Figura 49 Pantalla de mapa de evento e información completa de la noticia	110
Figura 50 Pantalla de preferencias	111
Figura 51 Ubicación del usuario para realizar la prueba	119

## Índice de anexos

Anexo1 Casos de uso específicos de aplicación web	127
---	-----

# CAPÍTULO 1

## 1. ESTADO DEL ARTE

### 1.1 Introducción

Con la gran evolución que han tenido los dispositivos móviles, el concepto de aplicaciones sensibles al contexto ha crecido en los últimos años. Según (Schilit, Adams, & Want, 1994) el contexto tiene varias definiciones dependiendo de la rama o ciencia que se trate, en los sistemas sensibles al contexto está caracterizado por hacer referencia a la ubicación, las personas que se encuentran en el entorno, servidores, y todos los dispositivos a los que se puede acceder en cualquier momento.

Los sistemas sensibles al contexto tienen el reto de explotar los cambios que se dan en el entorno, analizando todo lo que está a su alrededor teniendo en cuenta tres aspectos importantes como son: en donde te encuentras, con quien te encuentras y los dispositivos cercanos con los que puedes interactuar incluyendo sensores de luminosidad, de ruido, GPS, conexiones a la red, etc. y reaccionar mediante la información obtenida sin necesidad de la intervención del usuario. (Schilit, Adams, & Want, 1994)

Una gran ventaja que se tiene en la actualidad es el uso de los teléfonos móviles inteligentes ya que gracias a que cuentan con varios sensores como el GPS, acelerómetros, giroscopio que proporcionan información en todo momento colaboran con la implementación y mejora de los sistemas sensibles al contexto. Otras ventajas que se tienen son los accesos a la red como WIFI o GSM que permiten a los dispositivos estar conectados a la red todo el tiempo e intercambiar información entre ellos. (Nalepa & Bobek, 2014)

Las principales herramientas para la implementación de sistemas sensibles al contexto son las ontologías las cuales según (Asensio, 2008). Son la representación de conceptos u objetos en un dominio, en el que se toman en cuenta todas sus relaciones y propiedades. Y el Framework de descripción de recursos (RDF) que es un etiquetado gráfico de formato de datos con el objetivo de representar información en la web, la web semántica, y se utiliza para organizar el conocimiento, desempeñando un papel importante en las ontologías y la representación de conocimientos. Un modelo RDF consiste de tripletas que incluyen tres partes: sujeto, predicado y un objeto (Jamgade & Karale, 2015)

## 1.2 Computación ubicua

El uso intensivo de los dispositivos móviles, sean tabletas o teléfonos inteligentes, por parte de la sociedad para apoyar cada vez a más tareas sin importar los escenarios, en gran medida se debe a la convergencia tecnológica resultante de la mejora de los sistemas de comunicaciones, la disminución del tamaño y el aumento del rendimiento de los dispositivos móviles, la aparición de nuevos tipos de sensores y la presencia de software con mejores funcionalidades (Erazo, Illescas, & Espinoza, 2016), como consecuencia de lo anotado, tal como lo había presagiado Weiser (1991), se está siendo testigo de un periodo de transición, en la que el Internet y la computación distribuida, nos acercan a una nueva era de computación en todos lados, denominada computación ubicua.

Según (Cano, Calafate, Malumbres, & Manzoni, 2008) el propósito de la computación ubicua es básicamente la capacidad de interactuar con diferentes dispositivos “*inteligentes*” distribuidos alrededor de todos los escenarios donde se encuentre localizado el ser humano, sin que estos sean percibidos, es decir realizando diferentes procesos de manera transparente y natural para el usuario.

Por lo acotado la integración de la computación al entorno de las personas, conlleva a una de las ramas de investigación más importante de la actualidad, como es la computación sensible al contexto. El contexto o entorno es un elemento que hay que explotar a fin de enriquecer la experiencia de los usuarios con respecto a las cosas y dispositivos de uso cotidiano, y en particular a los dispositivos móviles.

## 1.3 Contexto

El significado general de contexto según el diccionario de Oxford indica que es un conjunto de circunstancias que rodean una situación y sin las cuales no se puede comprender correctamente (Oxford, s.f.).

El término contexto en la rama de las ciencias de la computación fue introducido por primera vez por (Schilit, Adams, & Want, 1994), que lo definen en base a tres aspectos: la ubicación, las personas que se encuentran cerca, y todos los dispositivos a los que se puede acceder en cualquier momento, así como los cambios sobre estos objetos.

En la literatura científica se han propuesto innumerables definiciones de contexto (Krumm, 2010), desde aquellas que enumeran las diversas dimensiones que pueden formar parte del contexto (ubicación, identidad del usuario, temporalidad, objetos y personas en el entorno del usuario, etc.) hasta aquellas definiciones sinónimas que lo consideran como el entorno del usuario o el medio ambiente de la aplicación. En la práctica tanto las definiciones enumerativas como las sinónimas resultan difíciles de aplicar al momento de incorporar nuevos aspectos del contexto a las aplicaciones.

En la comunidad científica una de las definiciones más aceptadas es la de Dey (2000) que define al contexto como: *“Cualquier información que puede ser usada para caracterizar la situación de una entidad. Una entidad puede ser una persona, un lugar o un objeto que es considerado relevante para la interacción entre el usuario y la aplicación misma”*.

## **1.4 Tipos de Contexto**

El contexto es un concepto multidimensional formado por innumerables entidades, de ahí que se han presentado varias clasificaciones en la literatura, sin embargo tomando como base las clasificaciones de (Morse, Dey, & Armstrong, 2000), (Poslad, 2006) y (Erazo, Illescas, & Espinoza, 2016), los tipos de contexto más importantes son los siguientes:

**Contexto físico:** la información de este contexto es proporcionada por sensores desplegados en un entorno particular, es decir se considera como el contexto de más bajo nivel que proporciona las propiedades físicas de usuarios, edificios o cualquier objeto, tales como: temperatura, nivel de ruido, presión arterial, humedad, entre otros.

**Contexto geográfico:** hace referencia a la localización del dispositivo o del usuario, tanto en lugares externos como en el interior de las edificaciones. La ubicación puede ser lógica representada mediante datos simbólicos, tales como: campus, biblioteca, auditorio, aula, entre otros, o física que es la que se obtiene a través de sistemas de posicionamiento como el GPS.

**Contexto temporal:** describe cualquier información de carácter temporal, tales como hora, fecha, horario de clases, calendarios de eventos, etc.



Contexto de tecnologías de información (TI): engloba la infraestructura de TI existente en el entorno, a nivel de hardware y software, tales como: dispositivos, aplicaciones, servicios informáticos, redes, etc.

Contexto de usuario: denota la identidad o perfil del usuario, cargo, funciones, responsabilidades, gustos, preferencias, entre otros.

Contexto sobre actividades/eventos: describe la actividad actual que realiza un usuario, así como las actividades planificadas a futuro. Por ejemplo: los estudiantes asisten a una conferencia de Ingeniería de Software.

Contexto social: describe a las personas e instituciones con las que el usuario tiene relación como: familia, amigos, compañeros de clase, redes sociales, etc.

## **1.5 Computación Sensible al contexto**

El estudio y desarrollo de la computación sensible al contexto inició en la década de los 90s con la introducción del concepto de computación ubicua por parte de Mark Weiser en su artículo titulado *“The Computer for the 210st Century”*. La computación sensible al contexto ha evolucionado a través del tiempo en todos los ámbitos en los que se aplica, desde aplicaciones de escritorio, aplicaciones web, hasta las tecnologías actuales como dispositivos móviles e internet de las cosas. (Perera, Zaslavsky, Christen, & Georgakopoulos, 2014)

El concepto introducido por Schilit, Adams y Want (1994) referente a sistema sensible al contexto indica que “Un sistema es sensible al contexto cuando utiliza el entorno y el ambiente en el que se encuentra para proporcionar información y/o servicios al usuario tomando como prioridad las preferencias del mismo”. El concepto que se utiliza en la actualidad, lo define como un sistema distribuido cuyo objetivo principal es utilizar toda la información que se encuentra alrededor del usuario y los cambios que esta pueda tener, aprovechando las conexiones con dispositivos como sensores ya sean internos o externos, servidores, teléfonos inteligentes, etc., a fin de que la aplicación pueda tomar decisiones y realizar diferentes tipos de tareas de conformidad con la situación y estado actual del usuario.

El reto de la computación sensible al contexto es explotar los cambios que se den en el entorno, analizando todo lo que está a su alrededor teniendo en cuenta tres aspectos importantes como son: en donde te encuentras, con quien te encuentras y los dispositivos cercanos con los que puedes interactuar incluyendo sensores de luminosidad, de ruido, GPS, conexiones a la red, entre otros, y de esta forma reaccionar mediante la información obtenida. (Schilit, Adams, & Want, 1994)

Existen tres niveles o características principales de la sensibilidad al contexto definidas por Barkhuus y Dey (2003):

**Personalización:** permite definir al usuario sus gustos, configuraciones, análisis, procesos e ingresar toda la información que utiliza el sistema de forma manual. Un ejemplo es el ingreso de información del perfil del usuario como: nombres, profesión, sexo, edad, etc.

**Sensibilidad al contexto pasiva:** es cuando el sistema muestra diferentes opciones obtenidas por el análisis y monitoreo constante del contexto, para que a partir de estas opciones el usuario escoja la acción que desea realizar. Un ejemplo es cuando el dispositivo muestra diferentes rutas en un mapa obtenidas a partir del análisis de los diferentes factores como tráfico o distancia y el usuario escoge la ruta que desea seguir.

**Sensibilidad al contexto activa:** el sistema es capaz de recopilar toda la información posible, analizarla, tomar decisiones y realizar procesos de manera automática sin la intervención del usuario. Un ejemplo son los sistemas de alarmas de los edificios que dependiendo de la situación que esté sucediendo el sistema decida si sonar, arrojar agua, llamar a emergencias, etc.

La función más significativa de los sistemas sensibles al contexto es que tienen la capacidad de adaptarse al comportamiento del contexto actual sin una intervención del usuario. Dicho contexto se debe modelar y una de las maneras de lograrlo es mediante ontologías (Kuutti, Dvoryanchikova, Lobov, Martinez Lastra, & Vahtera, 2012).

## 1.6 Ontología

Según Grubber (1995) Una ontología es una especificación compartida de conceptos con características similares de una manera formal y explícita. Por concepto se refiere a algún objeto o fenómeno existente en el mundo y las características referentes a este; explícito porque se identifican los conceptos y restricciones del uso; formal porque debe ser representado en un lenguaje que pueda ser entendido por la máquina; y compartido porque es consensuado y aceptado por un grupo de personas.

Existen varios tipos de formalismos para su representación, pero los más comunes son los siguientes:

**Clases:** representa los conceptos amplios, es decir que puede representar a otros conceptos específicos o menores, se deben organizar de tal manera que la herencia sea aplicable.

**Relaciones:** representan las diferentes relaciones que tiene un concepto con otros, generalmente se utilizan relaciones binarias en las cuales el primer argumento es el dominio de la relación y el segundo el rango. También se pueden utilizar las relaciones binarias para establecer los atributos de un concepto.

**Instancia:** se utilizan para representar a los elementos de una ontología.

El OWL (Ontology Web Language) es un lenguaje propuesto por la W3C (Consortio World Wide Web) para representar ontologías con el cual se puede representar de forma explícita todos los componentes de la misma, este lenguaje está compuesto por otros sublenguajes siendo el más utilizado el OWL Lite debido a que con su terminología se puede representar una gran cantidad de ontologías.

La terminología del OWL propuesta por la W3C es la siguiente.

`owl:Class`.- permite definir una clase con sus instancias y propiedades.

`rdfs:subClassOf`.- define las jerarquías de las clases.

rdf:Property.- se utilizan para definir relaciones entre instancias o *ObjectProperty*, y para definir relaciones desde instancias a valores de tipo dato o *DatatypeProperty*.

rdfs:subPropertyOf.- define jerarquías de propiedades.

rdfs:domain.- limita las instancias en la que una propiedad puede ser aplicada.

rdfs:range.- limita las instancias que una propiedad puede tener como valor.

## 1.7 Modelos ontológicos para representar el contexto

### 1.7.1 COBRA-ONT

Tabla 1  
Lista de clases y propiedades CORBA-ONT 0.2

Clases de la Ontología CoBra		Propiedades de la Ontología CoBra	
Relación "Lugar"	Agentes en Contexto de Ubicación	Relación "Lugar"	Agentes en Contexto de Ubicación
Place		latitude	locatedIn
AtomicPlace	ThingInBuilding	longitude	locatedInAtomicPlace
CompoundPlace	SoftwareAgentInBuilding	hasPrettyname	locatedInRoom
Campus	PersonInBuilding	isSpatiallySubsumedBy	locatedInRestroom
Building	ThingNotInBuilding	SpatiallySubsumes	locatedInParkingLot
AtomicPlaceInBuilding	SoftwareAgentNotInBuilding	accesRestricteToGender	locatedInCompoundPlace
AtomicPlaceNotInBuilding	ThingInRoom	lotNumber	locatedInBuilding
Room	SoftwareAgentInRoom		locatedInCampus
Hallway	PersonInRoom	<b>Relación "Agente"</b>	
Stairway			
OtherPlaceInBuilding		hasContactInformation	
Restroom	<b>Agentes en Contexto de Actividad</b>	hasFullName	<b>Agentes en Contexto de Actividad</b>
Gender		hasEmail	participatesIn
LadiesRoom	PresentationSchedule	hasHomepage	startTime
MensRoom		hasAgentAddress	endTime
ParkingLot	EventHappeningNow	fillsRole	location
<b>Relación "Agente"</b>	PresentationHappeningNow	isFilledBy	hasEventHappeningNow
Agente	RoomHasPresentationHappeningNow	intendsToPerform	invitedSpeaker
Person	ParticipantOfPresentationHappeningNow	desiresSomeoneToAchieve	expectedAudience
SoftwareAgent	SpeakerOfPresentationHappeningNow		presentationTitle
Role	AudienceOfPresentationHappeningNow		presentationAbstract
SpeakerRole			presentation
Audience	PersonFillsRoleInPresentation		eventDescription
IntentionalAction	PersonFillsSpeakerRole		eventSchedule
ActionFoundInPresentation	PersonFillsRoleAudienceRole		

Fuente: Chen, Finin y Joshi (2003)

Chen, Finin y Joshi (2003) han modelado una serie de ontologías que comprenden la versión 0.2 de COBRA-ONT mostrada anteriormente en la Tabla 1 y abarca 4 temáticas:

Ontología sobre lugares físicos: es una ontología que sirve para representar lugares físicos e incluso sirve para representar jerarquías, es decir cuando un lugar contiene otros lugares. Su

clase de más alto nivel es *Place* que representa un lugar físico y esta consta de varias propiedades como la latitud, longitud, nombre, etc. También consta de la clase *CompoundPlace* que puede contener otros lugares y la clase *AtomicPlace* que no puede contener otros lugares.

Ontología sobre agentes: es una ontología que representa los perfiles de agentes software y agentes humanos, definiendo sus respectivas características y roles. Su clase de más alto nivel es *Agent* que contiene a las subclases de *Person* y *SoftwareAgent*, entre algunas de las propiedades que se tienen para definir el perfil, tenemos: *HasContactInformation*, *HasFullName*, *HasHomePage*, *HasAgentAddress* y *FillsRole*.

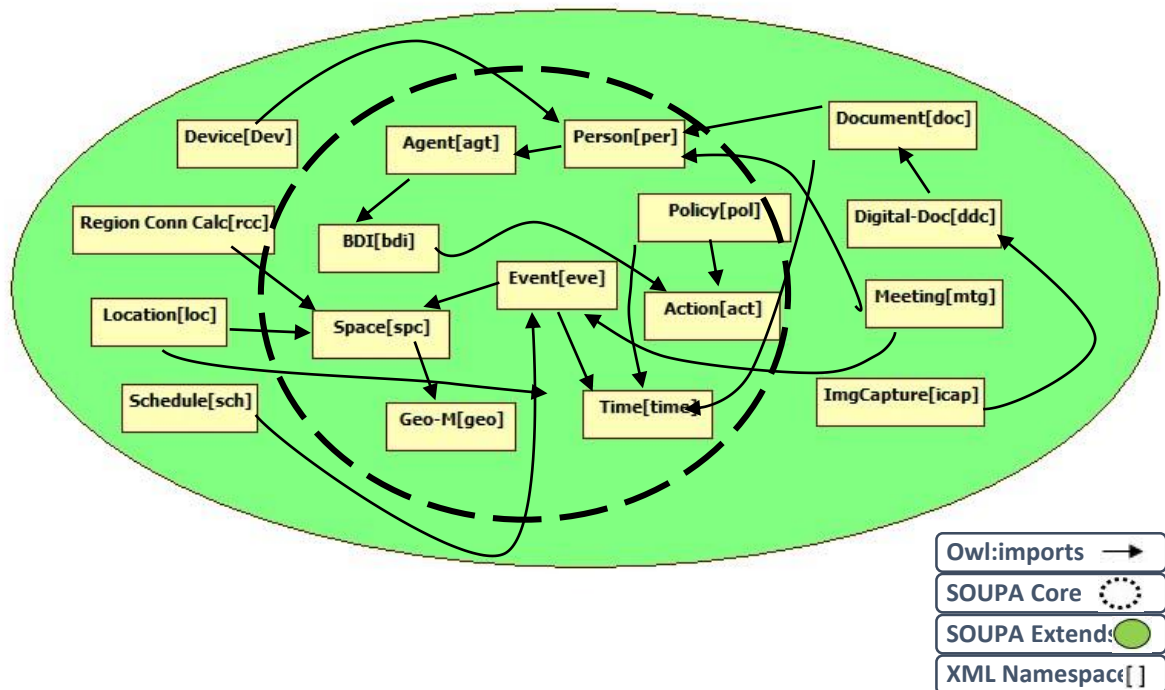
Ontología sobre el contexto de ubicación de los agentes: es una ontología que describe la ubicación de los diferentes agentes en el tiempo, la propiedad para determinar la ubicación del agente se denomina *LocatedIn*, esta a su vez contiene dos subpropiedades *LocatedInAtomicPlace* y *LocatedInCompoundPlace* que sirven para identificar si el agente está en un *Atomicplace* o un *CompoundPlace*.

Ontología sobre el contexto de la actividad de los agentes: esta ontología describe los eventos en los que los diferentes agentes participan, contiene varias propiedades que definen al evento como: *StartTime*, *EndTime*, *PresentationTitle*, *PresentationAbstract*, etc. Esta ontología contiene otras clases que sirven para saber si un evento se está dando actualmente y toda la información de dicho evento, estas clases son: *PresentationEventHappeningNow*, *RoomHasPresentationEventHappeningNow*, *SpeakerOfPresentationHappeningNow*, *AudienceOfPresentationHappeningNow*.

### 1.7.2 SOUPA

Está conformado por dos subconjuntos de ontologías que son la SOUPA CORE y la SOUPA EXTENTION, es una ontología propuesta por Chen, Perich, Finin y Joshi en (2004) y construida en OWL.

Figura 1  
Ontología SOAP



Fuente: Chen, Perich, Finin y Joshi (2004)

SOUPA CORE: como se presenta en la Figura 1 sirve para modelar el contexto relacionado con los conceptos de personas, agentes, acciones, políticas de seguridad, tiempo, eventos y espacio. A continuación se explican algunas clases contenidas en la ontología:

Person: Consta de propiedades que sirven para definir a una persona como por ejemplo:

Información de perfil como: nombre, género, edad, etc.

Información de contacto como: correo electrónico, dirección, teléfono, etc.

Perfil social y profesional que contiene información de sus círculos sociales y organizaciones a las que pertenece.

Policy & Action: Sirve para representar las políticas de seguridad y de privacidad sobre las diferentes acciones que se realicen, en SOUPA las políticas de seguridad y privacidad son reglas que permiten o restringir realizar cualquier acción. Esta ontología incluso ayuda con un

vocabulario para definir a los autores de las políticas, fecha de creación, hora de creación, y toda la información adicional que pueda incluir la política.

Agent & BDI: Brinda un vocabulario para representar a diferentes tipos de agentes humanos o computacionales, entendiendo que para SOUPA un agente es todo lo que tiene o puede generar conocimiento, tener creencias y ciertos deseos o intenciones. La clase principal es la clase Agent y consta de las propiedades *agt:believes*, *agt:desires* y *agt:intends* que son las propiedades principales que ayudan a definir a los agentes.

Time: Provee un vocabulario para representar el tiempo y los eventos que ocurren en un contexto, sus clases principales son:

a) *tme:TemporalThing* que sirve para representar a todo lo que es temporal y se asocia con la clase *tme:InstantThing* que representa cosas temporales en un instante de tiempo y la clase *tme:IntervalThing* que representa cosas temporales en un intervalo de tiempo.

b) *tme:TimeInstant* que representa a un instante en el tiempo, contiene la propiedad *tme:at* que le permite asociarse con la clase *tme:InstantThing* y las propiedades *tme:to* y *tme:from* que permiten asociarse con la clase *tme:IntervalThing*.

c) *tme:TimeInterval* que representa a un intervalo de tiempo.

Consta también de propiedades que comparan instancias de tiempo como: *tme:before*, *tme:after*, *tme:beforeOrAt*, *tme:afterOrAt*, y *tme:sameTimeAs*.

Space & Geo Measurement: Sirve para representar información de las diferentes regiones geográficas como sus relaciones, coordenadas geoespaciales, representación en el mapa y organización política. La clase principal es *spc:SpatialThing* que representa la ubicación de todas las cosas con características especiales y consta de la propiedad *spc:hasCoordinates* que sirve para describir a las coordenadas obtenidas por GPS.

Consta de una clase que le permite describir regiones controladas políticamente y esta se denomina *spc:GeographicalSpace*, la clase que describe el ente político se denomina *spc:GeopoliticalEntity*. La clase *spc:GeographicalSpace* puede contenerse así misma por jerarquías con la ayuda de las propiedades *spc:spatiallySubsumes* y *spc:spatiallySubsumedBy*.

Event: Proporciona un vocabulario para describir todo tipo de eventos ya sean temporales o espaciales. Sus clases principales son:

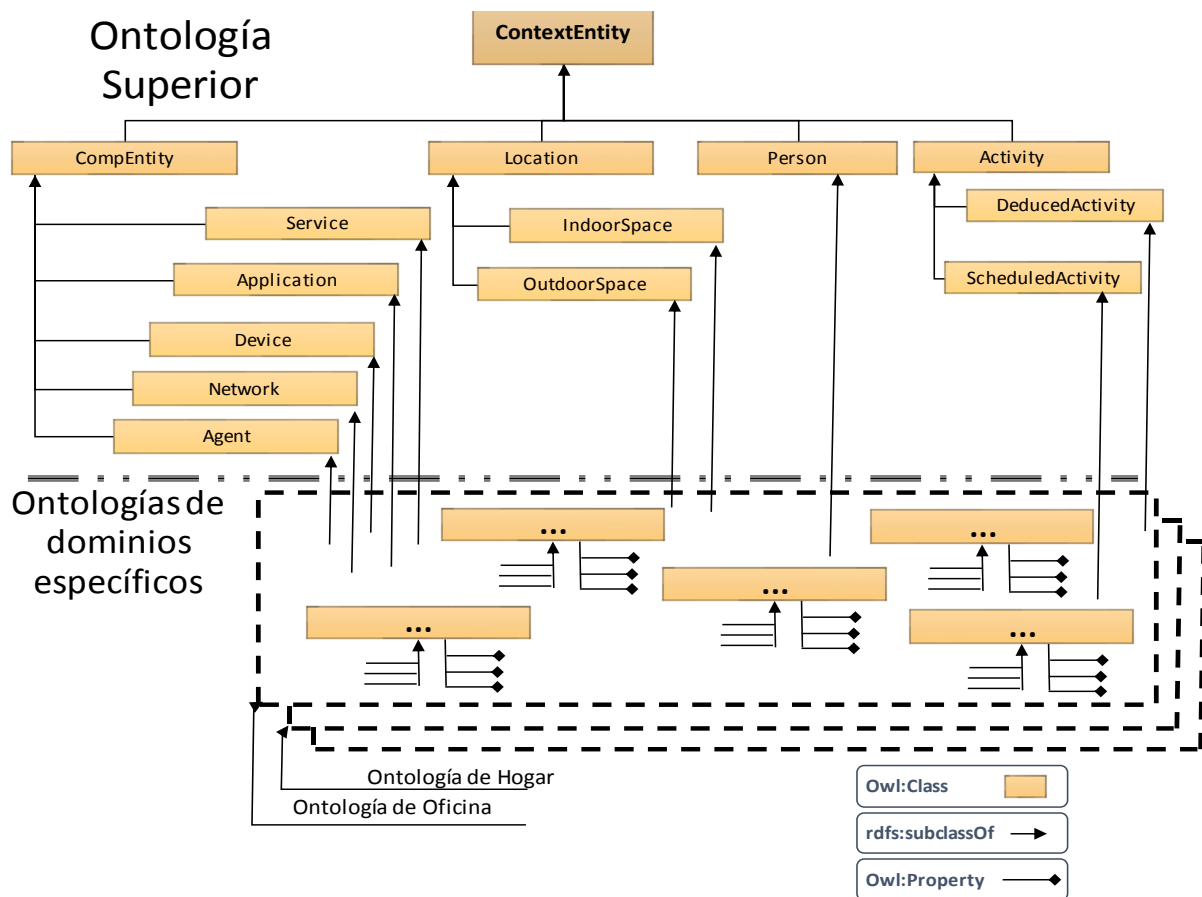
a) La clase `eve:Event` que representa a todos los eventos del dominio.

b) La clase `eve:SpatialTemporalThing` que representa a las cosas con extensión temporal y espacial. La clase `eve:SpatialTemporalThing` es una intersección de las clases `me:TemporalThing` y `spc:SpatialThing`.

c) La clase `eve:SpatialTemporalEvent` que representa a los eventos que tienen extensión temporal y espacial. La clase `eve:SpatialTemporalEvent` es una intersección entre las clase `eve:SpatialTemporalThing` y `eve:Event`.

### 1.7.3 CONON

Figura 2  
Ontología CONON



Fuente: (Wang, Zhang, Gu, & Pung, 2004)



Como se puede apreciar en la Figura 2, la ontología CONON (Wang, Zhang, Gu, & Pung, 2004) está definida por cuatro entidades contextuales: personas, localización, entidades computacionales y actividades. La ontología CONON sirve para capturar conceptos generales en un contexto básico a partir del cual se pueden agregar nuevos conceptos propios de dominios como: hogar, oficina, vehículo, etc.

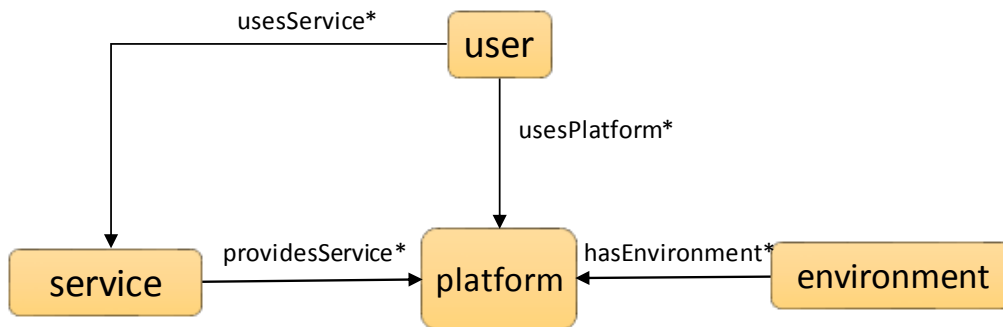
La clase principal se denomina *ContextEntity* de la que se derivan las siguientes clases:

- **CompEntity**: esta clase sirve para representar entidades computacionales y consta de cinco subclases.
  - Service.
  - Application.
  - Device.
  - Network.
  - Agent.
- **Location**: esta clase sirve para describir lugares físicos la conforman las subclases *IndoorSpace* (aula, pasillo, edificio, etc.) y *OutdoorSpace* (jardín, cancha, parqueadero, etc.). También consta de propiedades como *longitude*, *latitude*, *temperature*, *altitude*, etc.
- **Person**: es una clase que sirve para representar a usuarios humanos con propiedades como: *name*, *age*, *homeAddress*, etc.
- **Activity**: se divide en las subclases *DeducedActivity* y *ScheduledActivity* y sirve para representar actividades que se realizan en un contexto. También consta de las propiedades *startTime* y *endTime*.

#### 1.7.4 CoDAMoS

Tomando como referencia el estudio realizado por Martins y Silva (2012). La ontología CoDAMoS fue diseñada para crear aplicaciones o sistemas sensibles al contexto, con el fin de dar solución a los problemas que se pueden presentar en los entornos que utilizan sistemas inteligentes como: dispositivos móviles, generadores de códigos, adaptación de aplicaciones al entorno, etc. Además la ontología CoDAMoS tiene la capacidad de extenderse y adaptarse a nuestras necesidades permitiendo agregar nuevos conceptos y eliminar conceptos innecesarios.

Figura 3  
Ontología CoDAMoS



Fuente: Martins y Silva (2012)

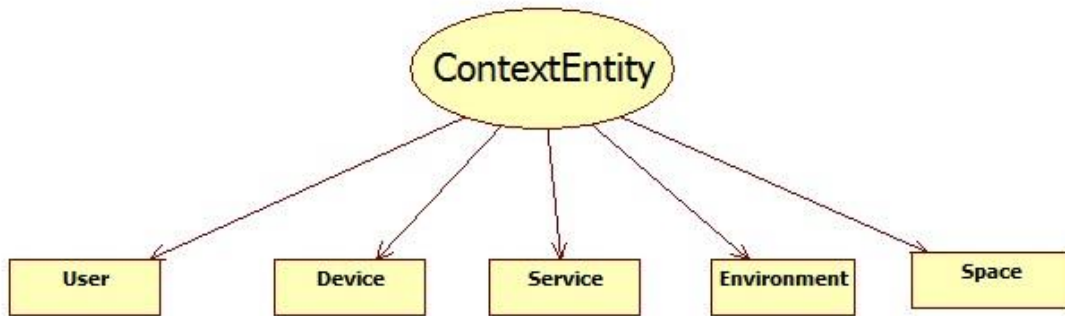
Como se puede observar en la Figura 3, la ontología CoDAMoS consta de 4 conceptos iniciales, considerados como los más importantes en los entornos inteligentes:

- user: captura la información sobre perfiles, gustos, estados de ánimo y todo tipo de información relacionada con el usuario.
- environment: captura la información sobre el entorno como: clima, ubicación, temperatura, hora, etc.
- platform: contiene información del hardware y software del dispositivo como: memoria disponible, procesador, ancho de banda, etc.
- service: provee información sobre los servicios planeados o prestados.

### 1.7.5 CACOnt

CACOnt (Xu, Zhang, Yang, Zhang, & Xing, 2013) es una ontología sensible al contexto que ayuda a modelar el contexto y proveer mecanismos de inferencia de información, esta ontología no solamente permite capturar información sobre los conceptos básicos de contexto sino que también es extensible y permite la integración fácil con otras ontologías. La Figura 4 es una representación de la ontología CACOnt en alto nivel.

Figura 4  
Ontología CACOnt

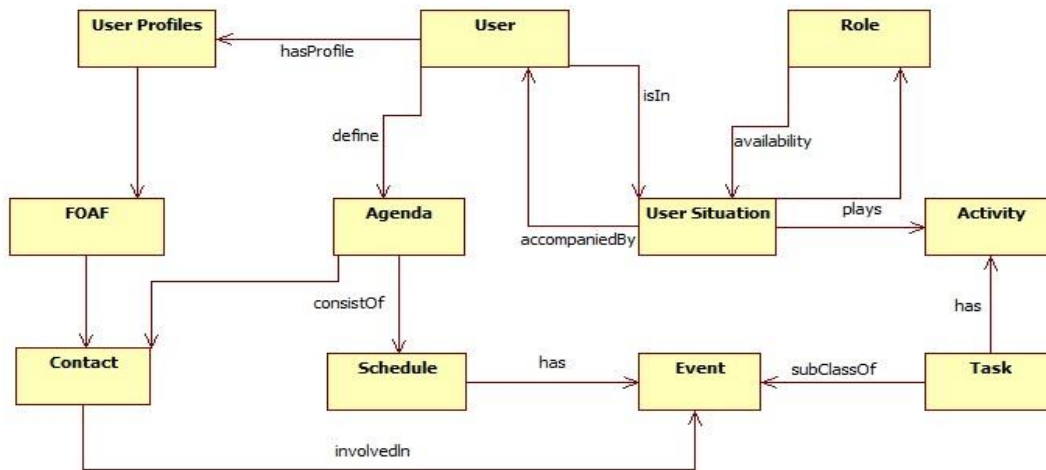


Fuente (Xu, Zhang, Yang, Zhang, & Xing, 2013)

En la ontología CACOnt se identifican los siguientes elementos: *User*, *Space*, *Environment*, *Device* y *Service* los cuales son descritos a continuación.

*User*: contiene toda la información relacionada a los usuarios humanos como: perfil, agenda, eventos, actividades, etc. La Figura 5 identifica todas las clases y relaciones del modelo *User*.

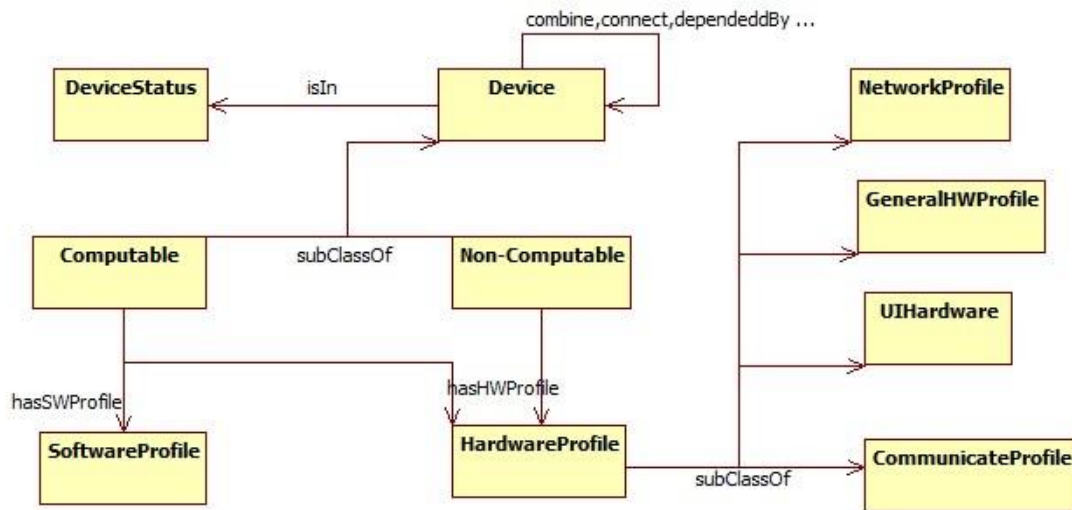
Figura 5  
Modelo User CACOnt



Fuente (Xu, Zhang, Yang, Zhang, & Xing, 2013)

*Device*: ayuda con la representación de la información relacionada con perfiles no humanos es decir de hardware o software. La Figura 6 identifica todas las clases y relaciones del modelo *Device*.

Figura 6  
Modelo Device CACOnt



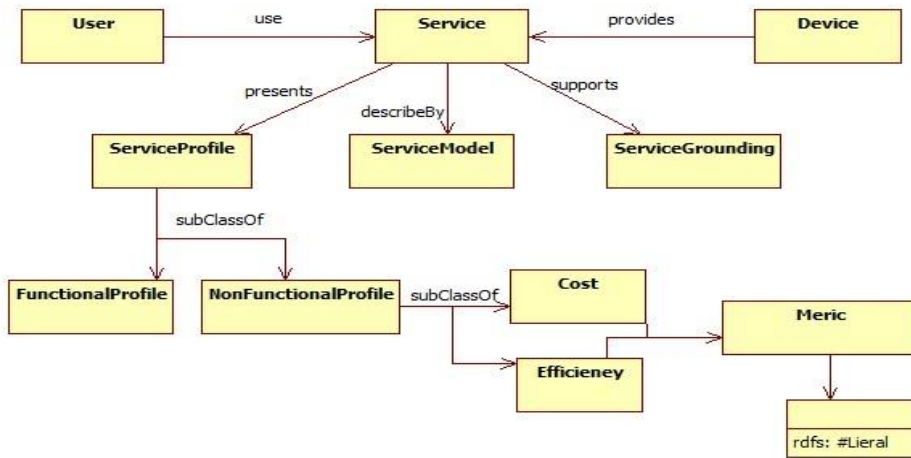
Fuente (Xu, Zhang, Yang, Zhang, & Xing, 2013)

Service: se divide en tres subclases:

- ServiceProfile: provee una descripción de las características funcionales y no funcionales del servicio.
- ServiceModel: especifica cómo hacer peticiones al servicio y cómo este responde, además de los flujos de control y datos en la interacción con los usuarios, para que de esta manera se decida si el servicio es bueno o malo.
- ServiceGrounding: especifica información detallada sobre las interacciones, accesos, protocolos, mensajes y detalles específicos del servicio.

La Figura 7 identifica todas las clases y relaciones del modelo Service.

Figura 7  
Modelo Service CACOnt

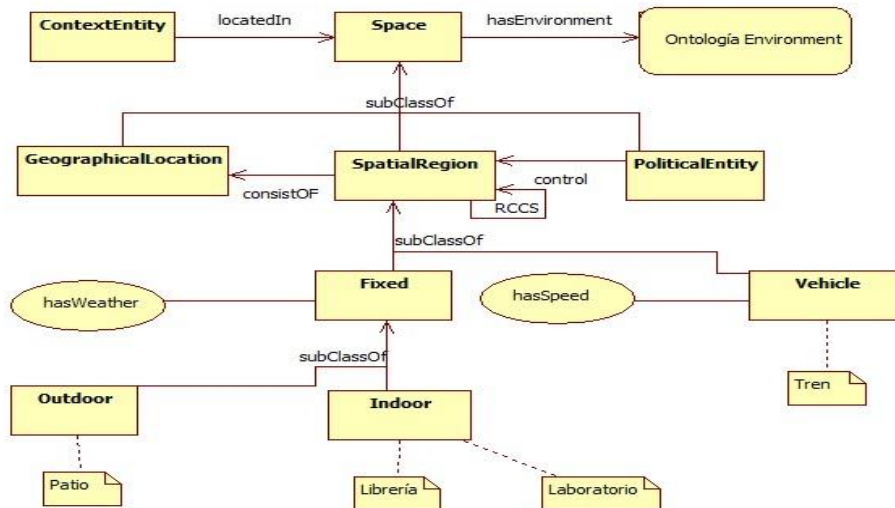


Fuente (Xu, Zhang, Yang, Zhang, & Xing, 2013)

Space: hace referencia a la información sobre la localización del usuario, basado en tres aspectos principales como: la ubicación geográfica, las regiones espaciales y las entidades políticas. Además permite especificar vehículos y definir si el espacio es en el exterior o interior de una edificación.

La Figura 8 identifica todas las clases y relaciones del modelo Space.

Figura 8  
Modelo Space CACOnt

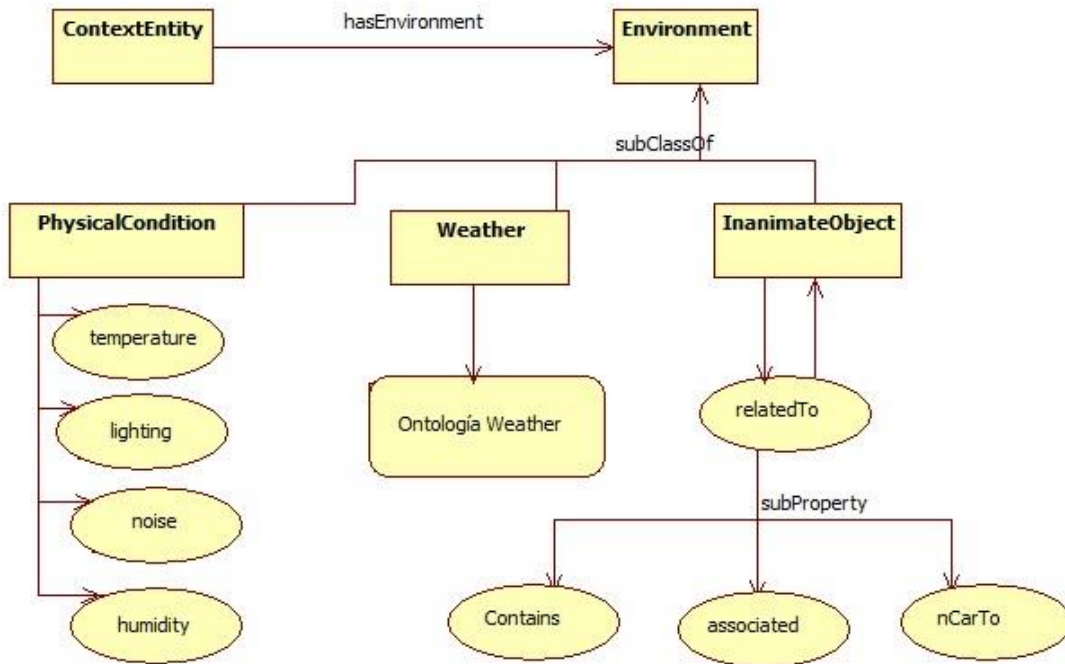


Fuente (Xu, Zhang, Yang, Zhang, & Xing, 2013)

Environment: esta clase ofrece al usuario información para tomar decisiones o influir en su comportamiento. En este caso se tienen en cuenta tres subclases que hacen referencia a la condición física, a los objetos con los que se interactúa y al clima.

La Figura 9 identifica todas las clases y relaciones del modelo Environment.

Figura 9  
Modelo Environment CACOnt

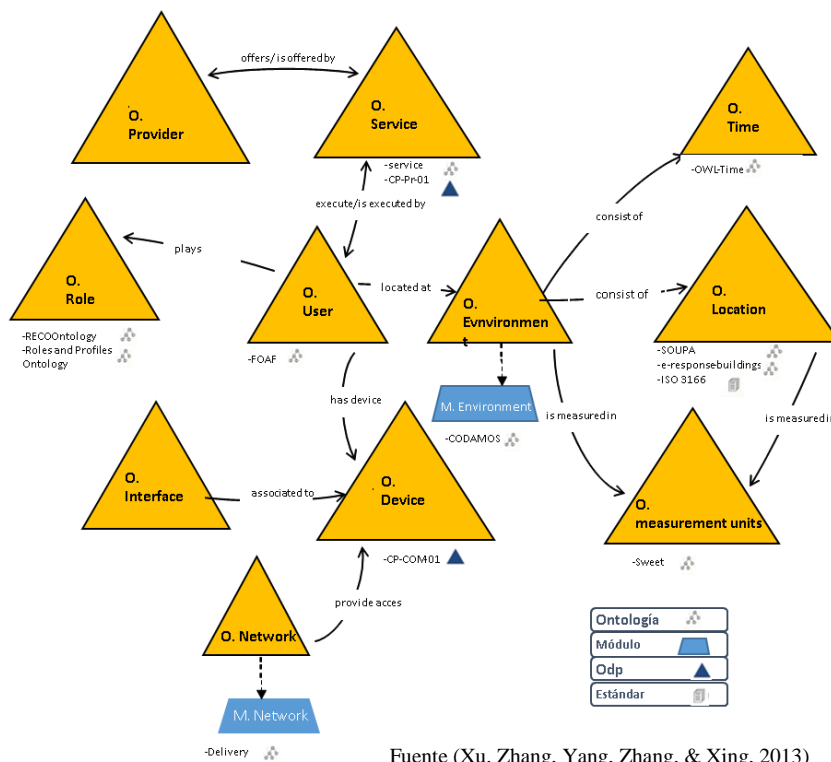


Fuente (Xu, Zhang, Yang, Zhang, & Xing, 2013)

### 1.7.6 mIO!

Según el estudio realizado en (Cadenas, y otros, 2009), la ontología mIO! consiste en una ontología núcleo que interconecta varias ontologías específicas como: CODAMOS, SOUPA, entre otras. Esta red de ontologías se utiliza para todo lo referente al conocimiento sobre el contexto, facilitando un modelo base para su representación.

Figura 10  
Ontología mio!



Esta red de ontología tiene un diseño modular, el cual permite agregar o quitar ontologías según la cantidad o el tipo de información que se requiera en los diferentes casos de uso. En la Figura 10 se puede observar la red de ontologías y sus conexiones representada en un gráfico de alto nivel:

- User: esta ontología reutiliza conceptos de las ontologías FOAF y CoDAMoS para representar usuarios, grupos, organizaciones, etc.
- Role: modela lo referente al conocimiento sobre perfiles, preferencias, roles, etc.
- Environment: reutiliza los conceptos de la ontología CoDAMoS sobre el entorno como luminosidad, humedad, etc.
- Location: reutiliza conceptos de la ontología SOUPA sobre la ubicación, coordenadas, distancias, etc.
- Time: utiliza conceptos sobre la ontología OWL-Time que sirve para modelar información sobre el tiempo como unidades de tiempo, intervalos de tiempo, instantes, etc.

- Role: reutiliza conceptos de la ontología RECOOntology y sirve para definir el rol o perfil del usuario.
- Measurementunits: esta clase utiliza conceptos que sirven para definir unidades de medida por ejemplo metros, kilómetros, etc.
- Service: esta ontología modela el conocimiento acerca de cualquier tipo de servicio. Por ejemplo: un servicio común que se presta en las empresas como: Restaurantes, servicios tecnológicos, Proveedores de internet, etc.
- Provider: esta ontología modela el conocimiento sobre los proveedores de los diferentes servicios.
- Device: reutiliza la ontología CoDAMoS para representar el conocimiento sobre dispositivos como información de hardware, software, plataformas, etc.
- Interface: esta ontología modela el conocimiento acerca de las diferentes interfaces para conectar los dispositivos.
- Network: esta ontología modela el conocimiento acerca de las conexiones de red como: topologías, operadores, administradores, precios, etc.

### 1.7.7 Análisis comparativo de los modelos ontológicos estudiados

En la Tabla 2 se presenta el detalle del tipo de información contextual que soporta cada una de las ontologías estudiadas con la finalidad de facilitar su elección dependiendo del alcance del dominio a implementar.

Tabla 2  
*Comparativa de Ontologías*

Conceptos	COBRA-ONT	SOUPA	CONON	CoDAMoS	CACOnt	mIO!
Personas	X	X	X	X	X	X
Lugares(localización)	X	X	X		X	X
Intereses/Deseos/Necesidades	X	X			X	X
Entorno	X			X	X	X
Servicios				X	X	X
Actividades/Eventos	X	X	X		X	
Políticas		X				
Tiempo		X			X	X
Unidades de Medidas						X
Dispositivos		X	X		X	X
Redes			X		X	X
Roles	X			X	X	X

Fuente: Elaboración propia (2017).



## 1.8 Otras ontologías relacionadas con el dominio de estudio

### 1.8.1 FOAF (Friend of a Friend)

La guía de especificación del vocabulario FOAF (Brickley & Miller, 2014) indica que el lenguaje de representación FOAF fue diseñado especialmente para sistemas de gran escala, tiene una gran capacidad de adaptabilidad e integración con otros modelos ontológicos, inclusive puede ajustarse según las necesidades de un dominio específico. Este lenguaje está basado en las reglas y estándares de la W3C, por lo que su representación se realiza generalmente en el lenguaje RDF.

FOAF es una ontología que consta actualmente de 19 clases y 44 propiedades que ayudan a describir a las personas y sus relaciones sociales, especializándose en las personas que realizan actividades en la web. A continuación se describen las clases principales:

- **Agent:** esta clase representa a las entidades u objetos que realizan actividades, generalmente son las personas pero también se puede asociar a software como por ejemplo los bots.
- **Document:** hace referencia a todo lo que es considerado documentos.
- **Group:** representa a grupos de agentes es decir, a una colección de agentes individuales. Por ejemplo grupos de trabajo, grupos informales, etc.
- **Organization:** representa agentes relacionados con instituciones, empresas, organizaciones, etc.
- **Person:** es una subclase de *agent* hace referencia a los agentes como personas debido a que todas las personas son consideradas agentes.
- **Image:** es una subclase de *document* y hace referencia a todo lo referente a imágenes.
- **OnlineAccount:** representa la prestación de algún servicio en línea a un agente.
- **PersonalProfileDocument:** representa la información de las personas que se encargaron de realizar los documentos.
- **Project:** representa a las clases que son referentes a proyectos, pueden ser individuales o grupales, públicos o privados, etc.

### 1.8.2 AIISO (Academic Institution Internal Structure Ontology)

Los creadores de esta ontología son Rob Styles y Nadeem Shamir (2008) quienes la describen como una ontología con clases y propiedades para representar organizaciones internas o instituciones académicas. Algunas clases se definen a continuación:

- Center: representa a un grupo de personas u organización reconocido como centro.
- College: representa a un grupo de personas u organización reconocida como universidad.
- Course: representa el material educativo de una organización.
- Department: representa a un grupo de personas u organización reconocida como departamento.
- Division: representa a un grupo de personas u organización reconocida como división.
- Faculty: representa a un grupo de personas u organización reconocida como facultad.
- Institute: representa a un grupo de personas u organización reconocida como instituto.
- Institution: es el nivel superior de una institución académica.
- KnowledgeGroup: representa materiales, recursos, objetivos de aprendizaje, etc.

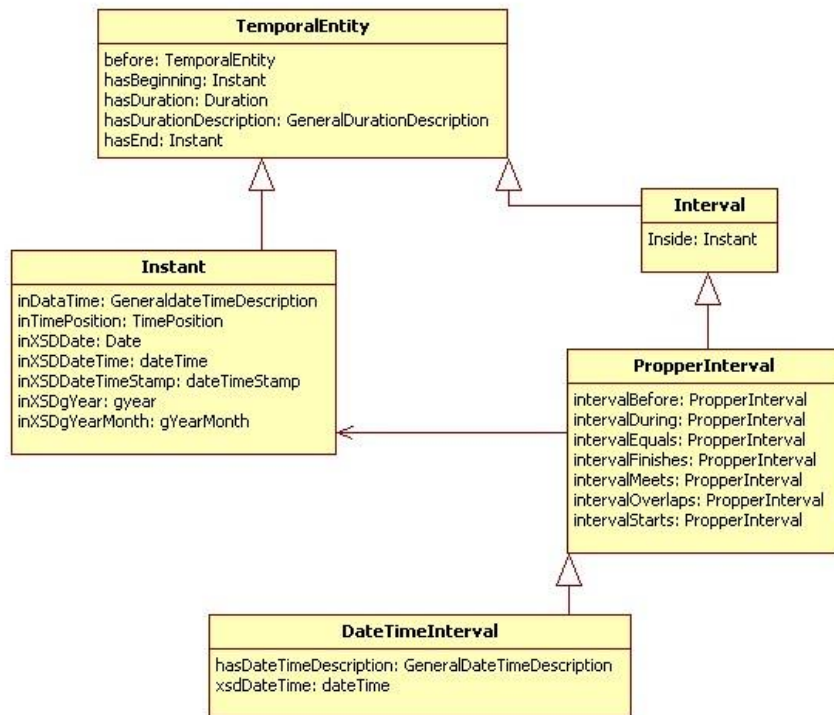
### 1.8.3 OWL-Time

Según W3C (Hobbs & Pan , 2017) que fue la organización que desarrolló la ontología, esta consta de una serie de recursos que ayudan a la representación del tiempo y de propiedades temporales como intervalos, instantes, duraciones, fecha, hora, etc. Además se adapta fácilmente a otras ontologías y se pueden agregar o quitar propiedades según las necesidades que se presenten.

La ontología OWL-Time consta de tres clases principales:

- TemporalEntity: esta clase consta de una serie de subclase que permiten la representación de instantes, intervalos, tiempo de inicio, tiempo de fin, etc. Las propiedades y subclases se presentan en la siguiente figura.

Figura 11  
*Clase TemporalEntity*

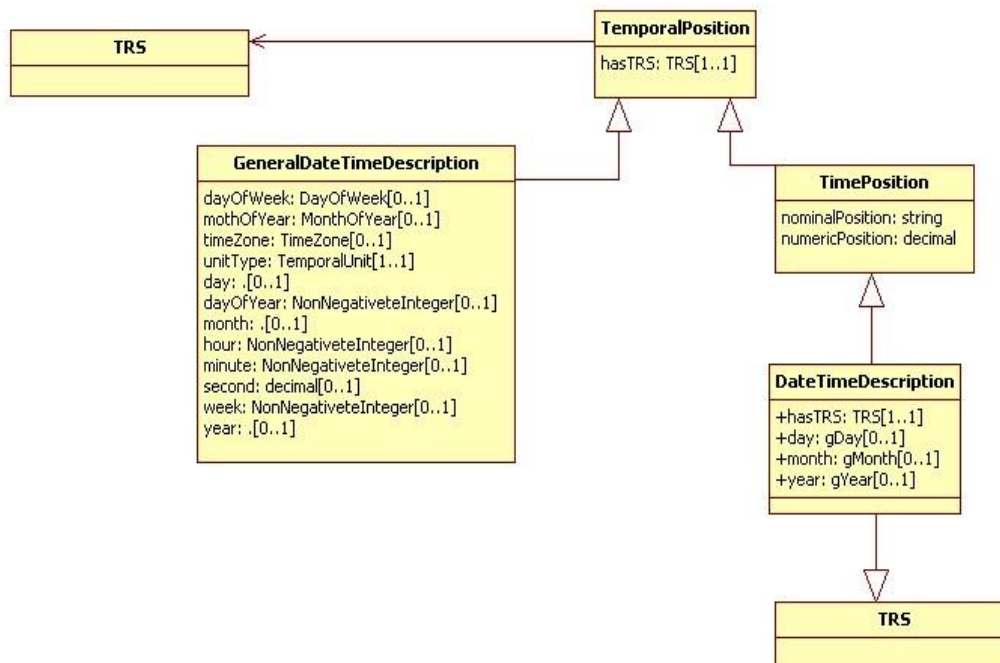


Fuente: W3C (Hobbs & Pan , 2017)

- **TemporalReferenceSystem:** se refiere al sistema que estamos utilizando para referenciar el tiempo como por ejemplo: un reloj, un calendario, horarios, etc. En la ontología esta es una clase vacía por lo que no tiene propiedades ni subclases esto quiere decir, que el usuario que va a implementar la ontología decide las propiedades y subclases que contendrá esta clase.
- **TimePosition:** esta clase va relacionada con la clase *TemporalReferenceSystem* y ayuda a identificar la posición en el tiempo en la que estamos por ejemplo: día de la semana, mes del año, hora en formato numérico o formato nominal, etc.

A continuación en la Figura 12 se presenta la clase *TimePosition* con sus subclases y propiedades.

Figura 12  
Clase *TemporalPosition*

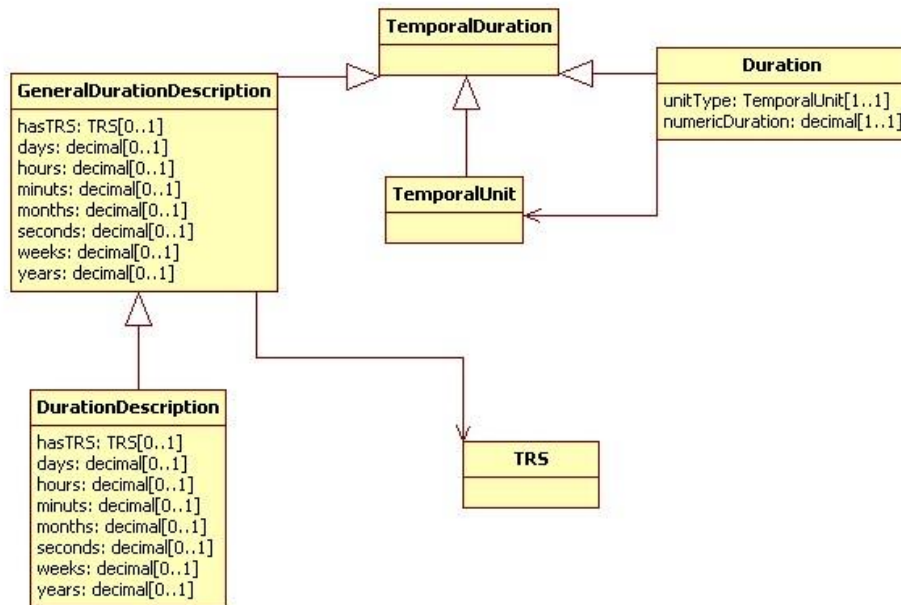


Fuente: W3C (Hobbs & Pan , 2017)

- **Duration:** esta clase ayuda a representar la duración de los intervalos, es decir si un intervalo dura dos días, tres horas o cualquier duración. Sus subclases ayudan a describir la duración del intervalo utilizando números, propiedades de calendarios, propiedades de reloj, etc.

En la Figura 13 se presenta las relaciones, subclases y propiedades de la clase *Duration*.

Figura 13  
Clase Duration



Fuente: W3C (Hobbs & Pan , 2017)

#### 1.8.4 WGS84 Geo Positioning

La ontología WGS84 Geo Positioning (W3C, 2005) es un vocabulario descrito en formato RDF para definir información geoespacial básica como latitud, longitud y altitud de un objeto. Esta información se representa por medio de las propiedades *latitude*, *longitude* y *altitude* que pertenecen al rango o al tipo de dato *float*.

## **CAPÍTULO 2**

### **2. MÉTODOS Y HERRAMIENTAS**

#### **2.1 Metodología NeOn para construcción de ontologías**

##### **2.1.1 Introducción**

La metodología NeOn permite a los ingenieros y desarrolladores crear redes ontológicas cubriendo todas las carencias de las metodologías antecedentes como la Dilligent, Methontology, On-To-Knowledge, etc. Dichas carencias se centran en la falta de guías detalladas para la realización de los diferentes procesos de las ontologías, no manejan el concepto de red de ontologías y no permiten la reutilización que es un factor clave de la metodología NeOn.

Esta metodología proporciona nueve escenarios y guías metodológicas que sirven de orientación para la construcción de redes ontológicas a través de distintas posibilidades de desarrollo como por ejemplo: la reutilización de ontologías o redes ontológicas existentes, reingeniería de redes ontológicas, etc. (Suarez Figueroa, 2010).

La metodología NeOn es la metodología más utilizada a partir de su creación debido a la facilidad, completitud, detalle y estandarización que proporciona a los ingenieros para facilitar la creación de ontologías o redes ontológicas.

##### **2.1.2 Escenarios**

La metodología NeOn (Suarez Figueroa, 2010) consta de nueve escenarios puesto que se ha demostrado que no siempre se sigue el mismo camino para la construcción de la ontologías o de las redes ontológicas. Los cuales son flexibles ya que se los puede combinar y adaptar según las necesidades para obtener la mejor solución posible.

Los escenarios son los siguientes:

Escenario 1. Desarrollo de las redes ontológicas desde la especificación hasta la implementación:

Escenario 2. Desarrollo de redes ontológicas mediante reutilización y reingeniería de recursos no ontológicos.

Escenario 3. Desarrollo de redes ontológicas mediante reutilización de recursos ontológicos.

Escenario 4. Desarrollo de redes ontológicas mediante reutilización y reingeniería de recursos ontológicos.

Escenario 5. Desarrollo de redes ontológicas mediante reutilización y mezcla de recursos ontológicos.

Escenario 6. Desarrollo de redes ontológicas mediante reutilización, mezcla y reingeniería de recursos ontológicos.

Escenario 7. Desarrollo de redes ontológicas mediante reutilización de patrones de diseño ontológicos.

Escenario 8. Desarrollo de redes ontológicas mediante la reestructuración de recursos ontológicos.

Escenario 9. Desarrollo de redes ontológicas mediante la localización de recursos ontológicos.

### **2.1.3 Actividades**

Las diferentes actividades que propone la metodología NeOn (Poveda Villalón, 2010) son flexibles lo que permite seleccionar solamente las actividades necesarias para la construcción de la ontología de manera que al final se desarrolle solamente lo necesario sin pérdidas de tiempo.

Las actividades se han agrupado en dos grupos que son la especificación de requisitos de la ontología y el modelamiento de la ontología los cuales se describen a continuación.

#### **A. Especificación de requisitos de la ontología.**

Se definen los principales requisitos para el desarrollo de la ontología, se utiliza la plantilla presentada en la Tabla 3 para su representación.

Tabla 3

*Definición de requerimientos de la ontología*

<b>1. Propósito</b>
Aquí se define el objetivo principal de la ontología, función o rol a cumplir.
<b>2. Alcance</b>
La cobertura y el detalle de la ontología.
<b>3. Lenguaje de implementación.</b>
Aquí se define el lenguaje en el que se va a implementar la ontología.
<b>4. Usuarios finales previstos.</b>
Aquí se definen los usuarios finales previstos de la ontología.
<b>5. Usos previstos.</b>
Los usos previstos de la ontología.
<b>6. Requisitos.</b>
<b>6.1. Requisitos no funcionales.</b>
Aquí se definen los requisitos o aspectos que la ontología debe satisfacer, incluyendo las prioridades para cada requisito.
<b>6.2. Requisitos funcionales: Preguntas de competencia.</b>
Aquí se definen las preguntas que la ontología debe responder junto con la respuesta que se busca obtener. Una vez definidos las preguntas y respuestas éstas se deben agrupar en categorías establecidas según los dominios que estas abarquen. Esto sirve para priorizar requisitos y dominios.
<b>7. Pre - glosario de términos</b>
<b>7.1. Términos de las preguntas de competencia.</b>
Aquí se definen la lista de términos utilizados en las preguntas junto con la frecuencia con la que aparecen para poder definir prioridades.
<b>7.2. Términos de las respuestas a las preguntas.</b>
Aquí se definen la lista de términos utilizados en las respuestas junto con la frecuencia con la que aparecen para poder definir prioridades.
<b>7.3 Objetos</b>
Aquí se definen los objetos que aparecen junto con su frecuencia tanto en preguntas como en respuestas.

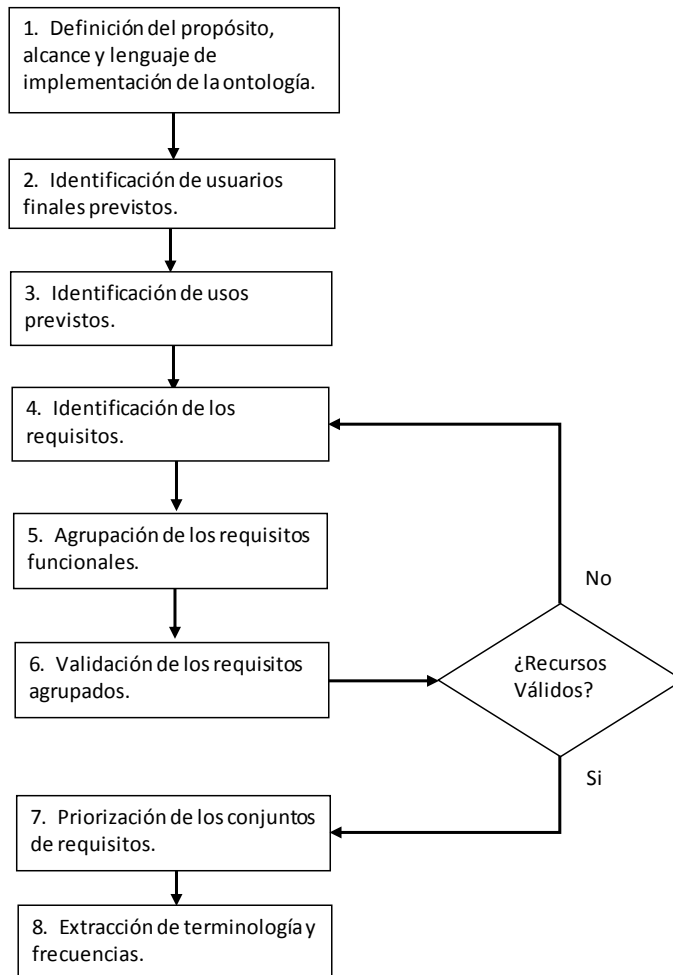
Fuente: (Poveda Villalón, 2010)



El proceso recomendado para desarrollar este procedimiento se muestra en la Figura 14.

Figura 14

*Pasos para definir requisitos de una ontología con la metodología NeOn*



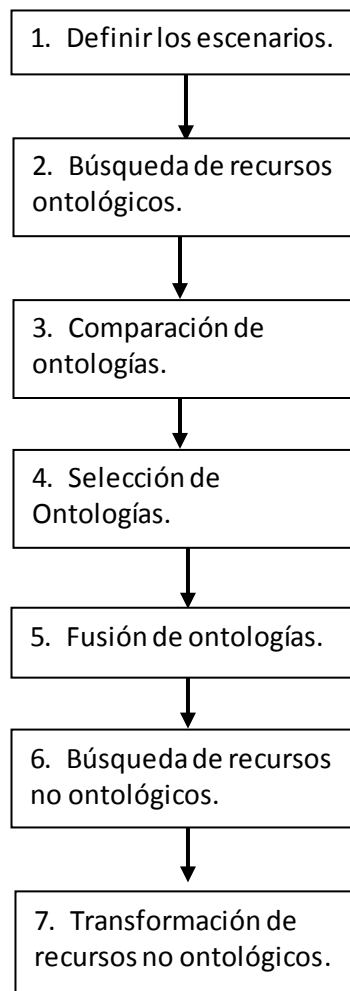
Fuente: (Poveda Villalón, 2010)

## **B. Plan y Modelado de la ontología**

En esta actividad se realiza todo con respecto al desarrollo y creación de la ontología, los procesos que conforman esta actividad se presentan en la Figura 15.

Figura 15

*Pasos para realizar el plan y modelado de la ontología con la metodología NeOn*



Fuente: (Poveda Villalón, 2010)

- 1) Definir escenarios: definir cuál o cuáles escenarios van a ser los que se van a utilizar para el desarrollo: se escogen los escenarios que se van a utilizar entre los 9 que propone la metodología NeOn.
- 2) Búsqueda de recursos ontológicos: se realiza la indagación o investigación de los recursos ontológicos que ya existen para poder evaluarlos y verificar si se pueden adaptar e implementar a la nueva ontología.
- 3) Comparación de ontologías: se realiza una comparación entre todas las ontologías seleccionadas anteriormente estableciendo criterios de comparación y categorías.
- 4) Seleccionar ontologías: se emplean métodos de valoración para poder obtener las ontologías que son más aptas y aplicables en el proyecto de desarrollo.

- 5) Adaptación y fusión de las ontologías seleccionadas: se realiza una reingeniería de todas las ontologías seleccionadas de manera que se puedan relacionar entre ellas y con los nuevos conceptos que se pueden agregar a la nueva red de ontologías.
- 6) Búsqueda de recursos no ontológicos: se realiza la búsqueda de los recursos no ontológicos que son necesarios para la nueva ontología que no se encontraron en los recursos ontológicos encontrados anteriormente.
- 7) Transformar los recursos no ontológicos en ontologías: se toma los recursos no ontológicos obtenidos en el paso anterior, se los estudia y analiza a cada uno para definir las propiedades, relaciones y conceptos que se necesitan para completar la nueva ontología.

#### **2.1.4 Conclusión**

La metodología NeOn contiene una secuencia de pasos que facilitan la creación y diseño de ontologías, cubriendo de manera muy eficiente todas las carencias de las metodologías predecesoras. Una de las ventajas de esta ontología es que implementa la reutilización por lo que en lugar de diseñar las ontologías desde cero se pueden utilizar ontologías existentes.

Debido a la fácil implementación, reutilización, estandarización y detalle en sus procesos, la metodología NeOn es la más utilizada en la actualidad por los ingenieros y por las personas que desarrollan sistemas inteligentes o sistemas relacionados con la semántica.

## **2.2 Otras Metodologías**

### **2.2.1 Metodología Methontology**

Methontology (Cárdenas Ruíz, 2017) es una metodología estructurada que proporciona una serie de guías para desarrollar ontologías de una manera correcta, fue desarrollada por el grupo de ingeniería ontológica de la Universidad Politécnica de Madrid el cual ha propuesto una serie de diferentes metodologías y ontologías en el ámbito de la ingeniería de conocimientos.

Consta de una guía bien detallada y su aplicación es sencilla por lo que es una de las metodologías que más se utilizan, y además consta de actividades que permiten el desarrollo, gestión y soporte de la ontología las cuales son descritas a continuación:

### **A. Desarrollo:**

1. Especificación: determina los usuarios, objetivos y la razón de la ontología.
2. Conceptualización: utiliza notaciones tabulares y gráficas para transformar las percepciones informales en percepciones semi-formales, aquí se obtiene el modelo conceptual de la ontología.
3. Formalización: convierte el modelo anterior a un modelo de lenguaje formal.
4. Implementación: transforma el modelo formal a un modelo computable en cualquier lenguaje de representación de ontologías como OWL, RDF, etc.
5. Mantenimiento: se realiza el constante mantenimiento y actualización de la ontología.

### **B. Gestión:**

6. Planificación.
7. Control.
8. Aseguramiento de la calidad.

### **C. Soporte:**

9. Adquisición de conocimientos.
10. Integración.
11. Evaluación.
12. Documentación.
13. Gestión de la configuración.

## **2.2.2 Metodología On-To-Knowledge**

La metodología On-To-Knowledge (Poveda Villalón, 2010) es una metodología creada con el objetivo de generar conocimiento a partir de información almacenada, orientado a grandes empresas y sistemas distribuidos.

Una de los requerimientos principales para la construcción de una ontología empleando la metodología On-To-Knowledge es conocer el objetivo de la ontología y cómo la ontología va a ser utilizada o servir en el futuro. Las actividades de esta ontología son:

- 1) Estudio de viabilidad: determinar si es posible la construcción de la ontología y si va a cumplir con los requerimientos planteados.
- 2) Lanzamiento: se identifican conceptos, relaciones y se construye la ontología.

- 3) Perfeccionamiento: se revisa la ontología construida anteriormente para verificar si contiene todos los conceptos y relaciones necesarias para cumplir con el objetivo.
- 4) Evaluación: se realizan diferentes pruebas para verificar que la ontología responde a todas las preguntas y proporciona toda la información que se requiere.
- 5) Mantenimiento: dar mantenimiento y actualizaciones a la ontología en caso de que existan cambios de requerimientos o adición de nuevos conceptos y relaciones.

### **2.2.3 Metodología Diligent**

La metodología Diligent (Andrade Pinto, Staab, & Tempich, 2004) fue creada con objetivo de que los expertos e ingenieros puedan crear ontologías de manera distribuida, es decir, no es necesario que las partes que están diseñando la ontología se encuentren reunidas siempre para la construcción.

Lo que busca es que la mayoría de partes que interactúen con el sistema se involucren en la creación y actualización de la ontología, por lo que se han dado casos en los que los propios usuarios del sistema son los que han proporcionado ideas y cambios importantes en la ontología.

Los procesos de la ontología son los siguientes:

- 1) Construcción: implica la construcción de toda la ontología con sus respectivas distribuciones.
- 2) Adaptación local: cada parte realiza las modificaciones correspondientes para que la ontología se adapte al sistema local.
- 3) Análisis: se analiza la ontología completa con todas sus distribuciones para poder determinar si consta de todos los requerimientos planteados local y externamente.
- 4) Revisión: se realiza la revisión de la ontología para determinar si esta contiene errores y responde a las preguntas planteadas, además verificar si proporciona toda la información requerida
- 5) Actualizaciones locales: en caso de que se den cambios o actualizaciones éstos se realizarán de manera local.

## 2.3 Tecnologías semánticas

### 2.3.1 XML

El *Extensible Markup Language* (Lenguaje de marcado extensible en español) es un lenguaje diseñado para documentos estructurados que se basa en la creación de etiquetas para generar semántica. Consta de un metalenguaje que funciona como un mecanismo para representar otros lenguajes de una manera estandarizada, lo que hace que pueda ser aplicado universalmente ya que puede definir etiquetas personalizadas para una ilimitada cantidad de documentos (Klein, 2001).

Un documento XML consiste en un árbol ordenado de etiquetas que puede contener los siguientes elementos: nodos de documento, elementos, atributos, nodos de texto, comentarios, instrucciones de procesamiento y espacio de nombres. (Bikakis, Christodoulakis, Gioldasis, Stavrakantonakis & Tsinarak, 2013), por ejemplo:

```
<?xml version="1.0"?>
<empleados>
<nombre>John</nombre>
<telefono>47782</telefono>
</empleados>
```

Un documento en XML se estructura mediante un modelo en árbol que consta de las siguientes características:

- Tiene una raíz.
- Los nodos que no son raíces tienen un nodo padre.
- Cada nodo tiene una etiqueta.
- Los elementos no forman ciclos entre sí (conservan una organización jerárquica).
- Principales métodos para estructurar documentos XML:
- DTD

El DTD (*Document type Definition*) es un mecanismo usado para definir la estructura de un documento XML, determinando los elementos que pueden presentarse, su orden, cardinalidad, etc. En el ejemplo anterior los elementos: empleados, nombre y telefono se pueden representar en DTD de la siguiente manera:

```
<!ELEMENT empleados (nombre,telefono)>
  <!ELEMENT nombre (#PCDATA)>
    <!ELEMENT telefono (#PCDATA)>
```

Los elementos nombre y telefono pertenecen al elemento empleados en ese orden y #PCDATA indica que el elemento contiene un texto que será validado por un analizador sintáctico.

## XML Schema

Es un lenguaje de definición de esquemas desarrollado por el *World Wide Web Consortium* (W3C) con el propósito de estructurar y restringir el contenido de los documentos XML. Con XML Schema puede especificar exactamente la jerarquía y los tipos de elementos utilizados en un documento XML. (Bikakis, Christodoulakis, Gioldasis,Stavrakantonakis & Tsinarak , 2013)

XML Schema soporta una variedad de tipos de datos como: *string, integer, time, etc.* Adicionalmente permite gestionar los tipos de datos definidos por el usuario como:

- Tipos de datos complejos (*Complex data types*): son definidos por tipos de datos ya existentes pero estructurando algunos atributos y usando:
  - Una secuencia de tipos de datos existentes donde el orden es importante.
  - Una colección de elementos que deben aparecer (orden no es importante).
  - Una selección de elementos que se deben elegir.
- Extensión de tipos de datos: a los tipos de datos existentes se les puede expandir con nuevos elementos o atributos.
- Tipos de dato de restricción: un tipo de dato es restringido añadiendo ciertos valores.

```
<complexType name = "restriccionTipoLectura">
  <restriction base="TipoLectura">
    <sequence>
      <element name="Nombre" type="string"
        minOccurs="1" maxOccurs="2"/>
    </sequence>
    <attribute name = "tirulo" type = "string"
      Use="required"/>
  </restriction> </complexType>
```

Ejemplo de un XML Schema para el documento de XML de empleados:

```
<complexType name="empleados">
  <attribute name="nombre" type="string" use="required"/>
  <attribute name="telefono" type="string" use="optional"/>
</complexType>
```

## Espacio de nombres XML

Los espacios de nombres (*Namespaces*) permiten diferenciar varios DTDs o XML schemas aplicados a un mismo documento XML, para lo cual se les proporciona un prefijo. Cada elemento en un namespaces puede ser utilizado con su prefijo.

`xmlns:prefix="location"` (location es la ubicación del DTDs o Schema)

## Transformaciones XLS (XLST)

XLS es un lenguaje de estilos extensible, como css, pero aplicado a transformaciones y formatos de documentos XML. XLST especifica las reglas de como un documento de XML puede ser transformado en otro documento XML mediante un documento HTML o en un texto plano.

- XLST mueve los metadatos de una representación XML a otra.
- Se utiliza cuando una aplicación tiene diferentes DTDs o XML schemas y necesita comunicarse.
- Es usado para el procesamiento de contenido y muestra la información en un formato legible para las personas
- El documento transformado puede utilizar el mismo DTD, XML schema, o uno completamente diferente.

## XPath y XQuery

XML Path Language (XPath) es una recomendación de W3C que proporciona un lenguaje capaz de describir la ruta de expresiones utilizadas en el modelo de datos de árbol de XML. Es un subconjunto de XQuery (Bikakis, Christodoulakis, Gioldasis,Stavrakantonakis & Tsinarak , 2013).



El XML Query Language (XQuery) se basa en el modelo del árbol XML y utiliza las rutas proporcionadas por XPath para redireccionar partes específicas de un documento XML. La estructura básica de estas consultas se basa en una expresión FLWOR que representa el *for*, *where*, *order by* para las cláusulas de XQUERY (Bikakis, Christodoulakis, Gioldasis, Stavrakantonakis & Tsinarak, 2013).

### XML y la web Semántica

La principal limitación de XML es que sólo describe reglas gramaticales. No hay manera de reconocer una semántica en un dominio en particular porque un documento de XML no impone una interpretación común de los datos contenidos en el documento.

Para intercambiar documentos XML y representar conocimiento, las asignaciones de un dominio deben traducirse mediante procedimientos de asignación. Esta tarea representa bastante esfuerzo ya que traducir un modelo requiere de una reingeniería que genere procedimientos de mapeo para documentos XML.

Una comparación realizada por Bikakis, Christodoulakis, Gioldasis, Stavrakantonakis & Tsinarak (2013) entre XML y la web semántica se representa en la siguiente tabla:

Tabla 4  
*Comparación entre XML y Web Semántica a varios niveles*

Nivel	XML	Web Semántica
Datos	XML	RDF
Esquema	XML Schema	RDF Schema
Consulta	XQuery-XPath	SPARQL

Fuente: Elaboración propia (2017).

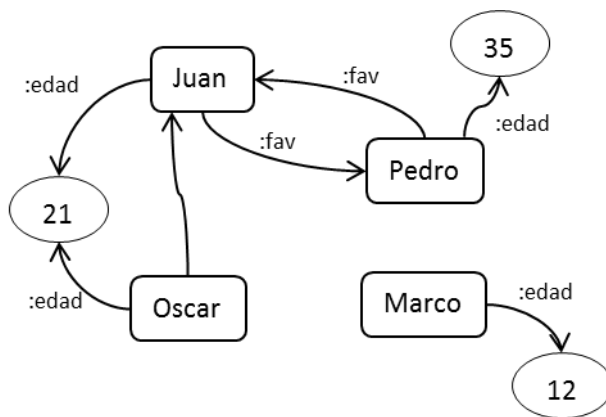
### 2.3.2 RDF

El Framework de descripción de recursos, o RDF, es un lenguaje ontológico ligero que tiene el propósito general de representar los recursos en la Web. RDF tiene un modelado de datos sencillo y flexible que se basa en un *RDF statement*, el mismo que consiste en elaborar triplas que contienen sujetos, predicados y objetos, siendo la propiedad del recurso el

predicado, el valor de la propiedad el objeto y el recurso u objeto el sujeto (Bikakis, Christodoulakis, Gioldasis,Stavrakantonakis & Tsinarak , 2013).

La colección de estas declaraciones (RDF statements) pueden ser comprendidas directamente como un gráfico de etiquetas donde los recursos son nodos y las declaraciones son arcos que conectan estos nodos. Un conjunto de RDF statements se denominan RDF Dataset. Un ejemplo de representación de edades y amigos favoritos se presenta en la Figura 16.

Figura 16  
*Conjunto de RDF statements (RDF dataset)*



Fuente: (Bikakis, Christodoulakis, Gioldasis,Stavrakantonakis & Tsinarak , 2013).

Para representar las sentencias de RDF de una manera procesable por máquina, RDF define una sintaxis específica de lenguaje de marcado extensible (XML), denominada RDF / XML (Staab & Studer, 2013). Los recursos anotados por RDF (es decir, los sujetos) suelen ser nombrados por Referencias de Identificador de Recurso Uniforme (URI). Los URIs son URIs pero con un fragmento de identificación al final. Los recursos sin URIs se llaman nodos en blanco que indica la existencia de un recurso, sin mencionar explícitamente el URIref de ese recurso y puede utilizarse para permitir que varias sentencias RDF hagan referencia al mismo (Staab & Studer, 2013), por ejemplo:

```

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix ex: <http://example.org/#>
@prefix elp: <http://example.org/Animal#>

elp:Ganesh ex:mytitle "Recurso llamado Ganesh" ;
           ex:mycreator "Juan".

           elp:name "Elefante" .

```

## RDF Schema

El RDF Schema, o RDFS intenta expresar ontologías simples con sintaxis RDF. Los recursos Web predefinidos *rdfs:Class* se utilizan para definir clases (conceptos), *rdfs:Resource* y *rdf:Property* para definir recursos y propiedades (roles).

RDFS predispone de un conjunto de meta-propiedades que se pueden utilizar para representar suposiciones de fondo en las ontologías según Staab & Studer (2013):

- *rdf:type*.- la instancia de la relación.
- *rdfs:subClassOf*.- la propiedad que modela la jerarquía entre las clases
- *rdfs:subPropertyOf*.- la propiedad que modela la jerarquía entre propiedades
- *rdfs:range*.- la propiedad que restringe todas las instancias de una determinada propiedad para tener valores que son instancias de una clase particular
- *rdfs:domain*.- la propiedad que restringe todas las instancias de una particular propiedad para describir instancias de una clase particular

Por Ejemplo:

```

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
@prefix elp <http://example.org/Animal#>
elp:Animal rdf:type rdfs:Class . -> Indica al esquema que Animal
es una clase
elep:Habitat rdf:type rdfs:Class .
elep:Elephant rdf:type rdfs:Class ; rdfs:subClassOf elp:Animal .
                                -> Indica que elefante es una subclase de animal

elp:liveIn rdf:type rdf:Property;
           rdfs:domain elp:Animal ; rdfs:range elp:Habitat .

```

->Indica que elefante tiene la propiedad liveIn(vive en) y en el rango de Animal y hábitat (elefante vive en un hábitat o en un hábitat vive un elefante).

```
Elp:south-sahara rdf:type elp:Habitat . -> Indica que sahara del sur es un hábitat
```

En un dominio que define el modelo entre objetos y relaciones, RDF se representa de forma natural y no hay que implementar pasos de traducción como en XML. Además el modelo RDF se puede utilizar incluso si la sintaxis XML desaparece o cambia, ya que RDF es una capa independiente de XML (Stefan, y otros, 2011).

### **Representación del conocimiento**

Según (Stefan, y otros, 2011), la representación de conocimiento se compone de tres capas:

- Nivel de implementación: consiste en estructuras de datos para una implementación particular.
- Nivel lógico: define de forma abstracta las inferencias que son realizadas por el sistema.
- Nivel epistemológico: define primitivas de representación adecuadas para expresar el conocimiento de una manera conveniente.

El nivel epistemológico suele definirse por una gramática que define el lenguaje de interés, pero la representación primitiva también pueden considerarse una Ontología. Y según (Stefan, y otros, 2011), “definir una ontología en RDF significa definir un RDFS donde se especifica los conceptos y relaciones del lenguaje en particular”.

Entonces debido a que cada RDFS (ontología) utiliza un propio espacio de nombres, se utiliza un prefijo OIL, para poder mezclar diferentes ontologías en un documento RDF sin que exista confusión (Stefan, y otros, 2011). RDF define un objeto, por lo que es posible hacer afirmaciones con un lenguaje sobre un objeto definido en términos de otro. Por ejemplo, podríamos mezclar la ontología con declaraciones de comportamiento sobre diferentes clases de animales. Esto no es posible en XML porque el significado de la etiqueta no se define, y no se puede asumir nada sobre la estructura del objeto.

### 2.3.3 OWL

El W3C *Web Ontology Working Group* ha identificado un numero de características, casos-uso para la Web Semántica, que requiere mucha más expresividad que lo que ofrece RDF y RDF Schema. Esto llevó a una iniciativa conjunta para definir un lenguaje más rico, llamado DAML + OIL2.

DAML + OIL (Version previa de DAML+OIL@) fue tomado como el punto de partida por el W3C *Web Ontology Working Group* en la definición del *Web Ontology Language* (OWL), el lenguaje que está destinado a ser el estándar para la Web Semántica (Antoniou & van Harmelen, 2004).

Los principales requerimientos según Antoniou & van Harmelen (2004) para un lenguaje ontológico son:

- Una sintaxis correctamente definida.
- Una semántica formal.
- Apoyo al razonamiento eficiente.
- Suficiente poder expresivo.

Uso de una semántica formal permite a una ontología razonar:

- Membresía de clase. Si  $x$  es una instancia de una clase  $C$ , y  $C$  es una subclase de  $D$ , entonces podemos inferir que  $x$  es una instancia de  $D$ .
- Equivalencia de clases. Si la clase  $A$  es equivalente a la clase  $B$ , y la clase  $B$  es equivalente a la clase  $C$ , entonces  $A$  es equivalente a  $C$ .
- Coherencia. Si se declara  $x$  como una instancia de la clase  $A$  que es una subclase de  $D$  y  $A$  también es una subclase de  $B \cap C$ , pero  $B$  y  $D$  son disjuntas. Entonces tenemos una inconsistencia porque  $A$  debe estar vacía, pero tiene la instancia  $x$ . Esto es una indicación de un error en la ontología.
- Clasificación. Si hemos declarado que ciertas parejas de propiedad-valor son una condición para pertenecer a una clase  $A$ , entonces si un individuo  $x$  satisface estas condiciones, podemos concluir que  $x$  debe ser una instancia de  $A$ .

## Tipos de OWL

- OWL Full: utiliza todos los lenguajes OWL primitivos y permite la combinación de estos primitivos de forma arbitraria con RDF y RDF Esquema. Esto incluye la posibilidad (también presente en RDF) de cambiar el significado de las primitivas predefinidas (RDF u OWL). Por ejemplo, en OWL Full, podríamos imponer una restricción de cardinalidad en la clase de todas las clases, limitando esencialmente el número de clases que se pueden describir en cualquier ontología. La ventaja de OWL Full es que tanto semántica como sintáctica es completamente compatible con RDF. Y la desventaja es que el lenguaje es muy poderoso por lo que no proporciona un apoyo de razonamiento eficiente
- OWL DL: abreviatura de *Description logic* o descripción lógica en español, OWL DL es un sub lenguaje del lenguaje anterior que restringe la forma en que los constructores de RDF y OWL pueden ser utilizados. La ventaja es que permite un apoyo de razonamiento eficiente pero se pierde la compatibilidad total con RDF.
- OWL Lite: subconjunto de OWL DL que excluye clases enumeradas, declaraciones y cardinalidad arbitraria, lo que conlleva a un lenguaje más fácil de entender (para los usuarios) y de implementar, pero con una expresividad restringida.

Todos los tipos de OWL utilizan RDF y RDFS para:

- Todos los tipos de OWL utilizan RDF para su sintaxis.
- En todos los tipos de OWL las instancias se declaran como en RDF, utilizando descripciones RDF e información de escritura.
- En todos los constructores OWL de cualquier tipo como *owl: Class* y *owl: DatatypeProperty* y *Owl: ObjectProperty* son especializaciones de sus contrapartes RDF.

Para el presente proyecto se utilizará el Lenguaje OWL lite por su facilidad a la hora de implementación.

## Cabecera de OWL

El elemento raíz, de una ontología OWL o documento OWL, es un elemento `rdf:RDF` que especifica un número de nombres de espacio (*namespaces*), y luego empieza con una colección de afirmaciones que se agrupan bajo el elemento `owl:Ontology` que contiene la versión de control que lleva la ontología, así como comentarios que servirán para ubicarse dentro del documento. Con el elemento `owl:imports` enumera otras ontologías cuyo contenido forma parte de la ontología actual, por ejemplo:

```
<rdf:RDF
  xmlns:owl = "http://www.w3.org/2002/07/owl#"
  xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#" >
```

La cabecera indica los prefijos que se van utiliza para este ejemplo son: owl,rdf,rdfs y xsd.

```
<owl:Ontology rdf:about="">
  <rdfs:comment>Ejemplo de una ontology OWL</rdfs:comment>
  <owl:priorVersion
    rdf:resource="http://www.mydominio.org/uni-ns-old" />
  <owl:imports
    Rdf:resource="http://www.mydominio.org/persons" />
  <rdfs:label>Universidad de Ontologia </rdfs:label>
</owl:Ontology>
```

Para indicar la versión utiliza `owl:priorVersion`, para indicar los comentarios utiliza `<rdfs:coment>` y para importar un recurso a la ontología con `owl:imports`.

## Propiedades

Objeto Propiedad (*ObjectProperty*) para relacionar objetos con otros objetos. Como *isTaughtBy* y *supervises*.

Ejemplo: El objeto propiedad llamado *isTaughtBy* (es impartido por) tiene un dominio de cursos y un rango de miembros académicos.

El curso es impartido por un miembro académico.

```

<owl:ObjectProperty rdf:ID="isTaughtBy">
  <rdfs:domain rdf:resource="#course" />
  <rdfs:range rdf:resource="#academicStaffMember" />
</owl:ObjectProperty>

```

Propiedades de tipos de datos para relacionar objetos con valores de tipos de datos. Como *phone*, *title*, *age*, etc.

```

  <owl:DatatypeProperty rdfs:ID="age">
    <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"/>
  </owl:DatatypeProperty>

```

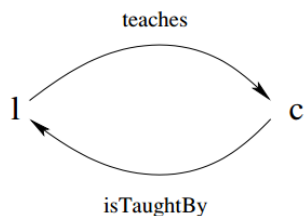
En OWL permite relacionar "propiedades inversas". Un ejemplo típico es el par *IsTaughtBy* y *teaches* (es enseñado y enseña) que se muestra en el siguiente ejemplo:

```

<owl:ObjectProperty rdf:ID="teaches">
  <rdfs:range rdf:resource="#course"/>
  <rdfs:domain rdf:resource="#academicStaffMember"/>
  <owl:inverseOf rdf:resource="#isTaughtBy"/>
</owl:ObjectProperty>

```

Figura 17  
*Propiedad inversa entre profesor-curso*



Fuente Antoniou & van Harmelen (2004)

En realidad el dominio y el rango se pueden heredar de la propiedad inversa. Según la Figura 17 el elemento l enseña el elemento c entonces el elemento c es enseñado por l.

Con *owl:equivalentProperty* se puede definir equivalentes de una propiedad a un elemento:

```

<owl:ObjectProperty rdf:ID="lecturesIn">
  <owl:equivalentProperty rdf:resource="#teaches"/>
</owl:ObjectProperty>

```



## Restricciones en las propiedades

Para que un elemento obtenga ciertos valores o propiedades se da ciertas restricciones al elemento con: *owl:allValuesFrom* y *owl:onProperty*, donde la primera propiedad (*allValuesFrom*) especifica los posibles valores que puede contener la propiedad especificada en *onProperty*. En el siguiente ejemplo, el elemento necesita los cursos de primer año que son impartidos por profesores. Para aplicar una restricción a la propiedad *isTaughtBy* se utiliza *onProperty* y *allValuesFrom* para definir como recurso, es decir de donde puede obtener los datos, la clase *Professor* con todos sus valores.

```
<owl:Class rdf:about="#firsYearCourse">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#isTaughtBy"/>
      <owl:allValuesFrom rdf:resource="#Professor"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Con la propiedad de restricción *owl: onProperty* se pueden utilizar otras restricciones como:

- *owl:hasValue*, existe un valor específico para la propiedad.
- *owl:someValuesFrom*, si existe al menos uno de los valores en la propiedad.
- Restricciones de cardinalidad, se puede especificar el número de coincidencias que se requiere de la propiedad, por ejemplo:

```
<owl:Class rdf:about="#department">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasMember"/>
      <owl:minCardinality
        Rdf:datatype="$xsd;nonNegativeInteger">
        10
      </owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasMember"/>
      <owl:maxCardinality
        Rdf:datatype="$xsd;nonNegativeInteger">
        30
      </owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

```

        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

```

Donde la clase departamento debe tener entre 10 miembros (*owl:minCardinality*) y 30 miembros (*owl:masCardinality*) . Dentro de las restricciones de cardinalidad se utiliza *\$xsd* para referirse al esquema XML y especificar que no debe ser un numero negativo (*nonNegativeInteger*).

### Propiedades Especiales de OWL

- *Owl: TransitiveProperty*, define una propiedad transitiva, como "es más alto que", o "es antepasado de". Por ejemplo la propiedad se puede utilizar entre clases de padre e hijo.
- *Owl: SymmetricProperty*, define una propiedad simétrica, como "mismo grado que" o "es hermano de". Se puede utilizar para explicar que dos clases son casi idénticas.
- *Owl: FunctionalProperty*, define una propiedad que tiene como máximo un valor para cada objeto, como "edad", "altura" o "directSupervisor".

### 2.3.4 SPARQL

Es un lenguaje que permite consultar las tripletas RDF especificando algunas plantillas para el proceso. Una plantilla contiene componentes de tripletas como sujeto, predicado u objetos, así los datos que coinciden con la plantilla son los devueltos por la consulta. El motor de SPARQL devuelve una lista de los componentes de la tripleta o tripletas que contengan información de la plantilla, por ejemplo si queremos saber los sujetos que tienen la propiedad de 21 años:

```
?person <example:age> "21" ^^example:age
```

Se utiliza una consulta de ejemplo llamadas QBE, donde el usuario define un patrón de ejemplo que el motor de consulta intenta combinar utilizando componentes del almacén de

datos. Este proceso es razonablemente intuitivo, y similar a los enfoques QBE aplicados a los datos relacionales como expresiones regulares o SQL. SPARQL se implementa en Jena a través del paquete ARQ, y las consultas se pueden hacer desde scripts Java o mediante un SPARQL cliente distribuido con Jena.

Un subconjunto de la sintaxis básica de una consulta de selección SPARQL (Grobe, 2009):

```
PREFIX   abreviatura_del_prefijo: <alguna_URI>

Select
    Algunas_variables
From
    <URI_algunas_fuentes_RDF>
Where
{
    { una_tripleta .
      otra_tripleta .
        ..... .      } .
}
```

En el *Select* se indican variables para guardar la información que se desea obtener de la consulta. En el *from* se indica los URL de donde se van obtener los datos y finalmente en el *where* se utilizan triplas con las propiedades de la ontología que nos ayuden a obtener el resultado esperado y todas las variables descritas en el *select*.

### 2.3.5 Apache Marmotta

Apache Marmotta es una plataforma abierta para *Linked Data* (Web Semántica), su objetivo es proporcionar una implementación abierta de una plataforma de *Linked Data* que se pueda utilizar de forma extendida y fácilmente desplegada por las organizaciones que desean publicar los datos vinculados o construir aplicaciones personalizadas en *Linked Data* (Apache, 2017).

Apache Marmotta permite:

- Lectura y escritura en un servidor de datos vinculados a la pila de Java EE.
- Almacenes conectados de triplas RDF basado en Eclipse RDF4J.
- Consultas SPARQL y LDPATH.

- Transparente el almacenamiento en caché de los datos vinculados.
- Básicos mecanismos integrados de seguridad.

### 2.3.6 API Jena y FUSEKI

La API de Jena tiene como objetivo proporcionar una interfaz de programación coherente para el desarrollo de aplicaciones con ontologías, independientemente de qué lenguaje de ontología esté utilizando en sus programas.

Jena es neutral en el lenguaje: la clase *OntClass* de Java puede representar una clase OWL o una clase RDFS. Para representar las diferencias entre las distintas representaciones, cada uno de los lenguajes de ontología tiene un *profile* (perfil), que enumera los constructores permitidos, los nombres de las clases y propiedades. Cuando se trabaja con una ontología en Jena, toda la información permanece codificada como tripletas de RDF (indexada como sentencias de Jena) almacenadas en el modelo RDF (Apache, Apache Jena, 2011–2017). La API de ontología no cambia la representación RDF de ontologías. Lo que sí hace es añadir un conjunto de clases y métodos de conveniencia que facilitan la escritura de programas que manipulan las tripletas subyacentes de RDF.

FUSEKI es una parte del proyecto de Apache Jena (Apache, Apache Jena, 2011–2017) pero es un paquete que se distribuye por separado, y permite hacer consultas y actualizaciones, usando simples solicitudes http, y obteniendo respuestas en varios formatos (JSON, XML y CSV). FUSEKI proporciona una interacción de tipo REST con los datos RDF.

## CAPÍTULO 3

### 3. ANÁLISIS DEL SOFTWARE

#### 3.1 Especificación de requerimientos de software (ERS)

##### 3.1.1 Propósito

El propósito es recopilar, especificar y validar los requerimientos y funcionalidades que debe cumplir el sistema de recomendaciones, y que tienen relación con la entrega de notificaciones a los estudiantes y profesores sobre los eventos y noticias que ocurren dentro de la universidad de conformidad con su contexto. Los requerimientos del software están estructurados en base a las directrices dadas por el estándar IEEE 830.

##### 3.1.2 Ámbito del Software

El sistema de recomendaciones móvil sensible al contexto de noticias y eventos académico está conformado por dos módulos:

Una aplicación web encargada de la administración de la información básica para el funcionamiento del recomendador móvil. Particularmente, realiza los mantenimientos de eventos, noticias, usuarios, cursos, estudiantes, profesores, áreas de conocimiento, materias, lugares (Auditorio, IERSE, etc.), entre otros.

Una aplicación móvil que presenta las recomendaciones de los eventos o noticias en función del perfil de usuario, áreas de conocimiento de interés, localización y tiempo.

##### 3.1.3 Definiciones, acrónimos y abreviaturas

Tabla 5

*Definiciones, acrónimos y abreviaturas.*

Nombre	Descripción
<b>Computación ubicua</b>	Integración de la computación en el contexto de la persona, de forma omnipresente y no percibida como productos u objetos informáticos diferenciados.
<b>Sensibilidad al contexto</b>	Referencia a que las computadoras (en el sentido más amplio de la palabra) pueden sentir y reaccionar en base al entorno que las rodea.
<b>Sensores</b>	Hardware de teléfonos inteligentes como giroscopio, sensor de proximidad, GPS.
<b>ERS</b>	Especificación de requisitos de software

Fuente: Elaboración propia (2017)

### 3.1.4 Referencias

Tabla 6  
*Referencias de especificación de requerimientos*

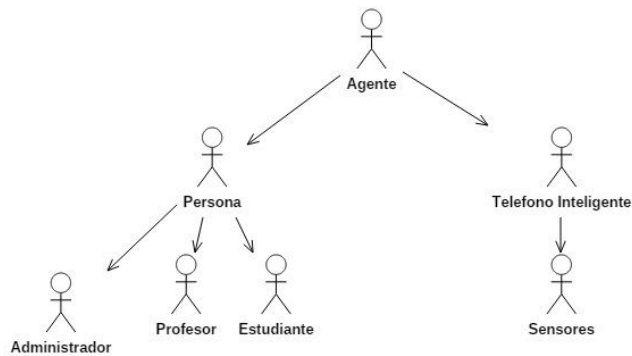
Título del Documento	Referencia
Standard IEEE 830 - 1998	IEEE

Fuente: Elaboración propia (2017)

## 3.2 Descripción general

### 3.2.1 Características de los usuarios

Figura 18  
*Clasificación de actores*



Fuente: Elaboración propia (2017)

La Tabla 7 muestra los usuarios que intervienen en el sistema.

Tabla 7  
*Usuarios que intervienen en el sistema*

<b>Tipo de usuario</b>	Administrador
<b>Actividades</b>	Control y manejo del sistema en general. Administración de la base de datos relacional. Encargado de que la información se sincronice con la ontología.
<b>Tipo de usuario</b>	Profesor
<b>Actividades</b>	Buscar recomendaciones sobre eventos o noticias de interés. Obtener notificaciones sobre eventos o noticias. Especificar parámetros de configuración de sus preferencias.
<b>Tipo de usuario</b>	Estudiante
<b>Formación</b>	Estudiante Universitario
<b>Actividades</b>	Buscar recomendaciones sobre eventos o noticias de interés. Obtener notificaciones sobre eventos o noticias. Especificar parámetros de configuración de sus preferencias.

Fuente: Elaboración propia (2017)

### 3.2.2 Restricciones

- Las Herramientas de desarrollo son: JENA, FUSEKI, RDF, OWL, SPARQL, GEOASPARQL, JAVA, PRIMEFACES Y MYSQL.
- Los servidores deben ser capaces de atender consultas concurrentemente.
- El sistema deberá generar conocimiento sobre los usuarios.
- El sistema debe comprender su entorno para dar información en tiempo real.
- El teléfono inteligente debe ser de gama media en adelante.
- El sistema deberá tener un diseño e implementación sencilla, independiente de la plataforma o del lenguaje de programación.
- Mantener una conexión permanente con la red.

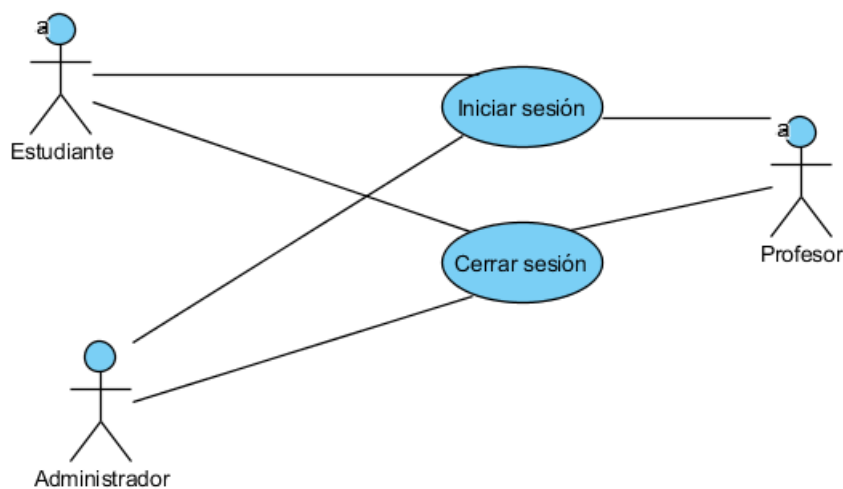
### 3.2.3 Suposiciones y dependencias

- Se asume que los requisitos aquí descritos son estables.
- Los equipos en los que se va a ejecutar el sistema deben cumplir con los requisitos indicados en el apartado 3.2.2 para garantizar una ejecución correcta de la misma.

## 3.3 Requerimientos funcionales

### 3.3.1 Diagrama de casos de uso de control de acceso

Figura 19  
*Diagrama de casos de uso de control de acceso*



Fuente: Elaboración propia (2017)

### 3.3.2 Descripción de casos de uso de control de acceso

Tabla 8

*Caso de uso de iniciar sesión*

<b>Identificador</b>	SE-01	
<b>Nombre</b>	Iniciar sesión	
<b>Descripción</b>	Estudiante y profesor deben iniciar sesión en la aplicación móvil para tener acceso a sus funciones. El administrador debe iniciar sesión en la aplicación web para tener acceso a las funciones de la misma.	
<b>Precondición</b>	Contar con la aplicación móvil y web. Disponer de una cuenta de usuario en la universidad.	
<b>Postcondición</b>	Usuario autenticado y mostrar página principal con recomendaciones de eventos por área de conocimientos en caso de la aplicación móvil. Para la aplicación web mostrar la página principal con el menú de opciones para gestionar la información académica.	
<b>Actores</b>	Estudiante, Profesor, administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema presenta una interfaz para ingresar el nombre de usuario y la contraseña.
	2	El estudiante, profesor o administrador ingresa su información de usuario y contraseña
	3	El sistema verifica que los datos sean correctos. En caso de error continua en la secuencia de error 1.
	4	El sistema presenta la página principal con el listado de eventos recomendados para el caso de la aplicación móvil y para la aplicación web la página principal para la gestión académica.
<b>Secuencia de error 1</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema informa sobre los datos incompletos para que sean ingresados.
	2	El sistema marca con alerta los campos incompletos y da el enfoque al primero de ellos para que sea corregido.
	3	Vuelve al paso 2 de la secuencia normal.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 9

*Caso de uso de cerrar sesión*

<b>Identificador</b>	SE-02	
<b>Nombre</b>	Cerrar sesión.	
<b>Descripción</b>	Permite al usuario cerrar la sesión.	
<b>Precondición</b>	Iniciar sesión.	
<b>Postcondición</b>	Permitir al usuario abandonar su cuenta.	
<b>Actores</b>	Estudiantes, profesores, Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El estudiante, profesor o administrador selecciona la opción de cerrar sesión.



	2	El sistema pregunta al usuario si está seguro continuar. Si el usuario responde que no continúa en la secuencia alternativa 1.
	3	El sistema termina la sesión. Si es la aplicación móvil el sistema borra cualquier preferencia que se haya guardado. (el archivo <i>Shared Preferences</i> )
<b>Secuencia alternativa 1</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema continúa en la pantalla principal con la posibilidad de seguir eligiendo las opciones que se brinda a los estudiantes, profesores o administrador.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Requerida	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

### 3.3.3 Diagrama de casos de uso de la aplicación web

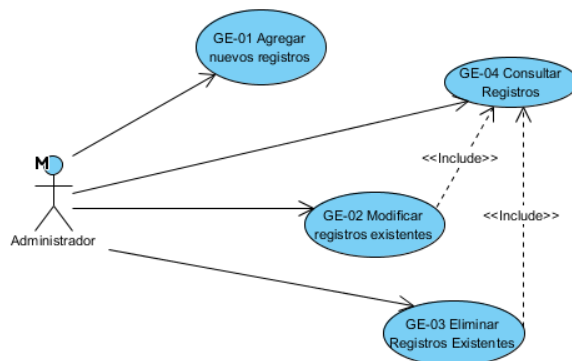
La aplicación web se encarga del mantenimiento de las tablas básicas de la base de datos relacional que son utilizadas para alimentar la base de datos triplestore de la ontología. Esta base de datos triplestore proporciona la información a la aplicación móvil para presentar y recomendar eventos y/o noticias académicas sensibles al contexto al usuario final.

Debido a que los procesos de ingresar, eliminar, modificar y consultar registros en las tablas de la base de datos son idénticos para todos los mantenimientos del sistema, se decidió dividir los casos de uso en dos partes: diagramas de caso de uso generales de la aplicación web y diagramas de casos de uso específicos de la aplicación web.

1. Diagrama de casos de uso generales de la aplicación web: describen procesos que son idénticos para todos los mantenimientos del sistema.

Figura 20

*Diagrama de casos de uso generales de la aplicación web*

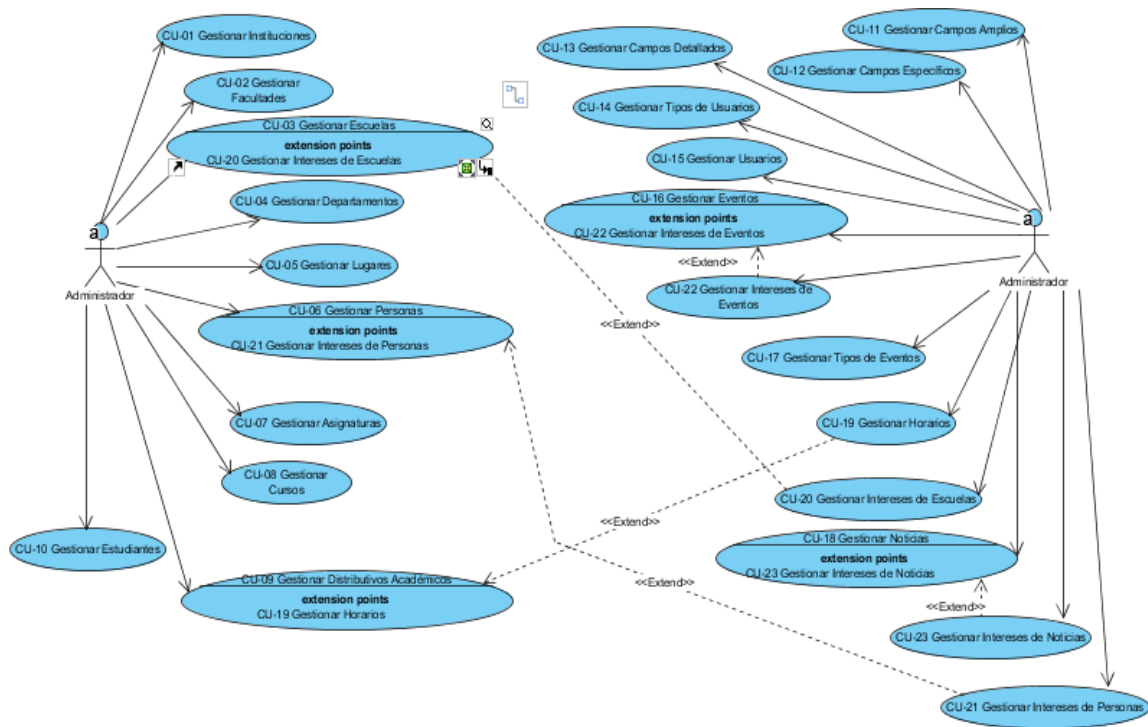


Fuente: Elaboración propia (2017)

2. Diagrama de casos de uso específicos de la aplicación web: este diagrama contiene todas las gestiones del mantenimiento del sistema, estos procesos utilizan los diagramas generales especificados anteriormente.

Figura 21

Diagrama de casos de uso específicos de la aplicación web



Fuente: Elaboración propia (2017)

### 3.3.4 Descripción de casos de uso generales de la aplicación web

A continuación se presenta la descripción de los casos de uso generales de la aplicación web.

Tabla 10

Caso de uso de agregar nuevos registros

<b>Identificador</b>	GE-01	
<b>Nombre</b>	Agregar nuevos registros	
<b>Descripción</b>	Ingresa registros nuevos a la base de datos del sistema	
<b>Precondición</b>	Seleccionar en la pantalla principal una de las opciones de mantenimientos.	
<b>Postcondición</b>	Visualizar los registros que se ingresaron a la base de datos.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador da clic en el botón “Crear”.

	2	El sistema muestra una pantalla con todos los campos requeridos en blanco.
	3	El Administrador debe llenar los campos que se muestran en la pantalla.
	4	El Administrador debe presionar el botón “Guardar” para ingresar el nuevo registro o en el botón “Cancelar” para cancelar el ingreso.
	5	En caso de que la información ingresada por el Administrador sea incorrecta o que un campo requerido se encuentre en blanco el sistema pasa a la secuencia de error 1.
	6	Se muestra un mensaje que indica que el ingreso se realizó con éxito.
<b>Secuencia de error 1</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema muestra un mensaje de error y marca los campos incompletos o incorrectos.
	2	Vuelve al paso 3 de la secuencia normal.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>	Este caso de uso es general, por lo que se utiliza en todos los casos de uso de gestión del sistema.	

Fuente: Elaboración propia (2017)

Tabla 11

*Caso de uso de modificar registros existentes*

<b>Identificador</b>	GE-02.	
<b>Nombre</b>	Modificar registros existentes.	
<b>Descripción</b>	Modifica los registros que se encuentran almacenados en la base de datos.	
<b>Precondición</b>	Seleccionar en la pantalla principal una de las opciones de mantenimientos.	
<b>Postcondición</b>	Visualizar los cambios realizados en los registros de la base de datos.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador debe realizar la búsqueda del registro ejecutando el caso de uso GE-04 de consulta.
	2	El Administrador debe dar clic en el registro que se va a modificar.
	3	El Administrador debe dar clic en el botón “Modificar”.
	4	El sistema muestra una pantalla con los campos que contienen la información antigua del registro.
	5	El Administrador debe llenar o cambiar la información de los campos que se muestran en la pantalla.
	6	El Administrador debe dar clic en el botón “Guardar” para modificar la información del registro o dar clic en el botón “Cancelar” para cancelar la transacción.
	7	En caso de que la información ingresada por el Administrador sea incorrecta o que los campos requeridos se encuentren en blanco, el sistema pasa a la secuencia de error 1.
	8	Se muestra un mensaje que indica que el ingreso se realizó con éxito.
<b>Secuencia de error 1</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema muestra un mensaje de error y marca los campos incompletos o incorrectos.
	2	Vuelve al paso 4 de la secuencia normal.

<b>Importancia</b>	Vital
<b>Urgencia</b>	Inmediatamente
<b>Observaciones</b>	Este caso de uso es general por lo que se utiliza en todos los casos de uso de gestión del sistema.

Fuente: Elaboración propia (2017)

Tabla 12  
*Caso de uso de eliminar registros existentes*

<b>Identificador</b>	GE-03.	
<b>Nombre</b>	Eliminar registros existentes.	
<b>Descripción</b>	Elimina los registros que se encuentran almacenados en la base de datos.	
<b>Precondición</b>	Seleccionar en la pantalla principal una de las opciones de mantenimientos.	
<b>Postcondición</b>	Visualizar los cambios realizados en los registros de la base de datos.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario debe realizar la búsqueda del registro ejecutando el caso de uso GE-04 de consulta.
	2	El Administrado debe dar clic en el registro que se va a eliminar.
	3	El Administrador debe dar clic en el botón “Eliminar”.
	4	El Administrador debe dar clic en el botón “Aceptar” para eliminar el registro o dar clic en el botón “Cancelar” para cancelar la transacción.
	5	El Sistema muestra un mensaje de confirmación, si el Administrador desea continuar debe dar clic en el botón “Si”, en caso de cancelar dar clic en el botón “No” y se regresa al paso 1.
6	Se muestra un mensaje que indica que se eliminó el registro.	
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>	Este caso de uso es general por lo que se utiliza en todos los casos de uso de gestión del sistema. Se debe realizar un buen análisis de la información antes de eliminar un registro de la base de datos	

Fuente: Elaboración propia (2017)

Tabla 13  
*Caso de uso de consultar registros*

<b>Identificador</b>	GE-04.	
<b>Nombre</b>	Consultar Registros.	
<b>Descripción</b>	Muestra la información existente en la base de datos.	
<b>Precondición</b>	Seleccionar en la pantalla principal una de las opciones de mantenimientos.	
<b>Postcondición</b>	Mostrar la información del registro solicitado.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>

	1	El sistema muestra una pantalla con los primeros 10 registros de la base de datos, en caso de no existir ninguno se muestra un mensaje que indica que no hay información almacenada.
	2	El sistema muestra cuadros de texto en las cabeceras de las columnas en los que se ingresará la información del registro quiere encontrar
	3	El Administrador debe llenar uno o varios de los cuadros de texto.
	4	El sistema muestra el o los registros encontrados en la base de datos según la información ingresada por el Administrador.
	5	El Administrador puede ordenar la información dando clic en las flechas que se encuentran en las cabeceras de las columnas.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>	Este caso de uso es general por lo que se utiliza en todos los casos de uso de gestión del sistema. La información de los registros siempre se muestra en las pantallas de mantenimientos por lo que no hay una secuencia de pasos específica.	

Fuente: Elaboración propia (2017)

### 3.3.5 Descripción de casos de uso específicos de la aplicación web

Debido a que todos los casos de uso de mantenimiento de las entidades de la base de datos son similares, a continuación en la Tabla 14 se presenta un ejemplo del caso de uso para gestionar instituciones. Los demás casos de uso se presentan en el Anexo 1.

Tabla 14

#### *Caso de uso de gestionar instituciones*

<b>Identificador</b>	CU-01	
<b>Nombre</b>	Gestionar Instituciones	
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar instituciones en la base de datos. El único campo que puede modificarse es el nombre de la institución.	
<b>Precondición</b>	El lugar en el que se encuentra la institución debe ser previamente ingresado en la base de datos.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción “Instituciones” de la pantalla de mantenimientos. Una vez seleccionado se puede realizar las siguientes acciones.
<b>Secuencia alternativa de Ingresar Institución</b>	2	Se ejecuta el caso de uso GE-01.
<b>Secuencia alternativa de Modificar Institución</b>	2	Se ejecuta el caso de uso GE-02.

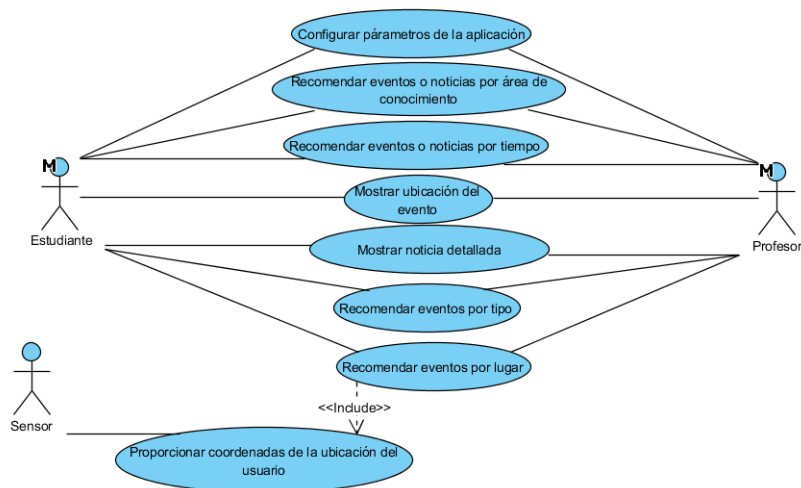
<b>Secuencia alternativa de Eliminar Institución</b>	2	Se ejecuta el caso de uso GE-03.
<b>Secuencia alternativa de Consultar Institución</b>	2	Se ejecuta el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>	En caso de que no se considere importante la información del lugar en el que se encuentra la institución solamente se debe seleccionar la opción NINGUNO en el campo de lugar.	

Fuente: Elaboración propia (2017)

### 3.3.6 Diagrama de caso de usos de la aplicación móvil

Figura 22.

*Diagrama de casos de uso de la aplicación móvil.*



Fuente: Elaboración propia (2017)

### 3.3.7 Descripción de casos de uso de la aplicación móvil

A continuación se describen los casos de uso de la aplicación móvil.

Tabla 15

*Caso de uso de configuración de parámetros de la aplicación*

<b>Identificador</b>	CU-01
<b>Nombre</b>	Configurar parámetros de aplicación.

<b>Descripción</b>	El sistema permite al usuario configurar parámetros como el tiempo de actualización de las recomendaciones, distancia en la que prefiere buscar los eventos, elegir si desea noticias o eventos que sean del día, de la semana o del mes	
<b>Precondición</b>	Se realizó el inicio de sesión correctamente.	
<b>Postcondición</b>	Guardar configuraciones	
<b>Actores</b>	Estudiante, Profesor	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema presenta una opción en el menú para ingresar a una pantalla de configuración.
	2	El estudiante o profesor selecciona la opción “preferencias”.
	3	La aplicación móvil presenta un formulario en el que el usuario ingresa la siguiente información: Si desea recomendaciones por Día, Semana o Mes. Ingresa una distancia y la unidad (20 metros, 1 kilómetro, etc.) El tiempo de actualización del aplicativo en minutos.
	4	El estudiante o profesor ingresa sus preferencias y guarda los cambios. En caso de error continua en la secuencia de error 1.
	5	La aplicación móvil guardará las configuraciones realizadas por el estudiante o profesor en un archivo del dispositivo móvil llamado “Shared Preferences”. La información se guardará con un apuntador (espacio de Nombre) para leer los datos de las preferencias del usuario.
	6	El sistema regresa a la pantalla principal.
<b>Secuencia de error 1</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema presenta un mensaje de que los datos son incorrectos (en el caso de ingresar números, si el usuario ingreso números negativos)
	2	Vuelve al paso 3.
<b>Importancia</b>	Opcional	
<b>Urgencia</b>	Requerida	

Fuente: Elaboración propia (2017)

Tabla 16

*Caso de uso de proporcionar coordenadas de la ubicación del usuario*

<b>Identificador</b>	CU-02	
<b>Nombre</b>	Proporcionar coordenadas de la ubicación del usuario.	
<b>Descripción</b>	Se obtiene las coordenadas del dispositivo móvil (la ubicación del usuario).	
<b>Precondición</b>	Permisos en la aplicación móvil para obtener la localización.	
<b>Postcondición</b>	Guardar las coordenadas del usuario, para que estas sean utilizadas en las solicitudes de la aplicación móvil a los servicios web.	
<b>Actores</b>	Sensor	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema verifica que haya permisos para utilizar la ubicación del dispositivo móvil. Si se produce un error continúa en la secuencia de error 1.
	2	Los sensores brindan información sobre la localización (GPS) del usuario al sistema.
	3	El sistema guarda esta información.

<b>Secuencia de error 1</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema no puede obtener la localización del usuario.
	2	El sistema solicita al usuario que permita a la aplicación móvil el uso del GPS.
	3	El usuario acepta los permisos de localización.
	4	Vuelve al paso 2 de la secuencia normal.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	

Fuente: Elaboración propia (2017)

Tabla 17

*Caso de uso de recomendar eventos o noticias por áreas de conocimiento*

<b>Identificador</b>	CU-03	
<b>Nombre</b>	Recomendar eventos o noticias por área de conocimiento	
<b>Descripción</b>	Mostrar al usuario la información recuperada sobre eventos o noticias por área de conocimiento en la aplicación móvil. La información que se presenta al usuario es los eventos o noticias que pertenezcan a su carrera y a sus preferencias.	
<b>Precondición</b>	Iniciar sesión. Información almacenada en la ontología.	
<b>Postcondición</b>	Presentar los diferentes eventos o noticias al usuario.	
<b>Actores</b>	Sistema, Estudiantes y Profesores	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El estudiante o profesor selecciona la opción “evento por área de conocimiento” o la opción “noticia por área de conocimiento”.
	2	El sistema mediante un servicio web tipo REST solicita al servidor Glassfish la información según la opción seleccionada.
	3	El servidor Glassfish construye la consulta SPARQL y la ejecuta en el servidor FUSEKI.
	4	El servidor Glassfish obtiene los resultados, los representa en formato JSON y los envía a la aplicación móvil. En caso de error continua en la secuencia de error 1.
	5	La aplicación móvil obtiene el objeto en formato JSON para dividir entre la información que se muestra al usuario y la que no.
	6	Para el caso de eventos la aplicación móvil obtiene : Nombre del evento. Descripción del evento. Fecha y lugar del evento. Coordenadas de la ubicación del evento. (No se muestra)  Para el caso de noticias la aplicación obtiene: Título de la noticia. Breve descripción de la noticia. Fecha de la noticia. Toda la descripción de la noticia. (No se muestra)
	7	La aplicación móvil muestra al estudiante o profesor una lista con las recomendaciones encontradas.
<b>Secuencia error 1</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema no encuentra respuestas para la opción seleccionada.



	2	El sistema prepara un objeto JSON que tiene un título de no recomendaciones y el aplicativo mostrará esa respuesta al usuario.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Requerida	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 18

*Caso de uso de recomendar eventos o noticias por tiempo*

<b>Identificador</b>	CU-04	
<b>Nombre</b>	Recomendar eventos o noticias por tiempo	
<b>Descripción</b>	Mostrar al usuario la información recuperada sobre eventos o noticias por tiempo (Día, Semana o Mes).	
<b>Precondición</b>	Iniciar sesión. Información almacenada en la ontología	
<b>Postcondición</b>	Presentar los diferentes eventos o noticias al usuario	
<b>Actores</b>	Estudiantes, profesores	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El estudiante o profesor selecciona la opción “eventos por tiempo” o la opción “noticias por tiempo”.
	2	El sistema mediante un servicio web tipo REST solicita al servidor Glassfish la información según la opción seleccionada. Y envía al servicio REST el período de tiempo(día, semana, mes) que seleccionó previamente en las preferencias CU-01
	3	El servidor Glassfish construye la consulta SPARQL y la ejecuta en el servidor FUSEKI.
	4	El servidor Glassfish obtiene los resultados, los representa en formato JSON y los envía a la aplicación móvil. En caso de error continua en la secuencia de error 1
	6	La aplicación móvil obtendrá cada objeto en formato JSON para dividir entre la información que se muestra al usuario y la que no.
	7	Para el caso de eventos la aplicación móvil obtiene : Nombre del evento. Descripción del evento. Fecha y lugar del evento. Coordenadas de la ubicación del evento. (No se muestra)  Para el caso de noticias la aplicación móvil obtiene: Título de la noticia. Breve descripción de la noticia. Fecha de la noticia. Toda la descripción de la noticia. (No se muestra)
	8	La aplicación móvil muestra al estudiante o profesor una lista con todas las recomendaciones encontradas.
<b>Secuencia error 1</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema no encuentra respuestas para la opción seleccionada.
	2	El sistema prepara un objeto JSON que tiene un título de no recomendaciones y el aplicativo mostrará esa respuesta al usuario.

<b>Importancia</b>	Vital
<b>Urgencia</b>	Requerido
<b>Observaciones</b>	

Fuente: Elaboración propia (2017)

Tabla 19  
*Caso de uso de recomendar eventos por tipo*

<b>Identificador</b>	CU-05	
<b>Nombre</b>	Recomendar eventos por tipo.	
<b>Descripción</b>	Mostrar al usuario una pantalla en la que puede visualizar los diferentes eventos de manera organizada según su tipo.	
<b>Precondición</b>	Iniciar sesión. Información almacenada en la ontología	
<b>Postcondición</b>	Visualizar los tipos de eventos según la opción elegida por el usuario	
<b>Actores</b>	Estudiantes, Profesores	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El estudiante o profesor selecciona la opción “evento por tipo”.
	2	La aplicación móvil presentará una nueva pantalla donde se podrá seleccionar los diferente tipos de evento entre ellos: Congreso Simposio Foros Convenciones Workshops Debate Otros
	3	El estudiante o profesor selecciona un tipo de evento.
	4	La aplicación móvil mediante un servicio web de tipo REST solicita los datos al servidor Glassfish.
	5	El servidor Glassfish se encarga de construir la consulta SPARQL y de ejecutar la consulta en el servidor de FUSEKI.
	6	El sistema recuperara los datos de contexto mediante un objeto en formato JSON según el tipo de evento. En caso de error continúa en la secuencia de error número 1.
	7	La aplicación móvil muestra al usuario una lista con los eventos encontrados de manera ordenada según el tipo de evento seleccionado.
<b>Secuencia error 1</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema no encuentra respuestas para la opción seleccionada.
	2	El sistema prepara un objeto en formato JSON que tiene un título de no recomendaciones y el aplicativo mostrará esa respuesta al usuario.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Requerido	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 20

*Caso de uso de recomendar eventos por lugar*

<b>Identificador</b>	CU-06	
<b>Nombre</b>	Recomendar eventos por lugar.	
<b>Descripción</b>	Mostrar al usuario la información sobre eventos de interés que se encuentran cerca del estudiante o profesor.	
<b>Precondición</b>	Iniciar sesión. Información almacenada en la ontología. Coordenadas de los lugares ingresados en la ontología.	
<b>Postcondición</b>	Visualizar eventos por lugar	
<b>Actores</b>	Estudiantes, Profesores	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El estudiante o profesor selecciona la opción “eventos por lugar”.
	2	La aplicación móvil obtendrá las coordenadas del usuario en ese momento.
	3	La aplicación móvil enviará la solicitud del usuario mediante un servicio web de tipo REST con las coordenadas, distancia y unidad de medida al servidor Glassfish. Para recuperar datos de contexto.
	4	El servidor Glassfish realiza una consulta GEOSPARQL a FUSEKI para obtener los lugares cercanos a las coordenadas del usuario.
	5	El servidor Glassfish obtiene los lugares cercanos y con estos realiza una consulta SPARQL al servidor FUSEKI para obtener las recomendaciones. En caso de error continuar en la secuencia de error 1.
	6	El servidor Glassfish obtiene los resultados, los representa en formato JSON y los envía a la aplicación móvil.
	7	La aplicación móvil obtendrá cada objeto en formato JSON para dividir entre la información que se muestra al usuario y la que no.
	8	La aplicación móvil obtiene: Nombre del evento. Descripción del evento. Fecha y lugar del evento. Coordenadas del lugar del evento. (No se muestra)
9	La aplicación móvil muestra al estudiante o profesor una lista con las diferentes recomendaciones encontradas.	
<b>Secuencia error 1</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema no encuentra respuestas para la opción seleccionada.
	2	El servidor Glassfish prepara un objeto JSON que tiene un título de no recomendaciones. Y muestra el mensaje en la aplicación móvil.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Requerido	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 21

*Caso de uso de mostrar ubicación de evento*

<b>Identificador</b>	CU-07
<b>Nombre</b>	Mostrar ubicación del evento

<b>Descripción</b>	Mostrar al usuario la ubicación exacta de un evento seleccionado, utilizando la aplicación Google Maps.	
<b>Precondición</b>	Iniciar sesión. Cualquiera de los casos de uso: CU-04, CU-05, CU-06, CU-07.	
<b>Postcondición</b>	Visualizar en el mapa la ubicación del evento.	
<b>Actores</b>	Estudiantes, Profesores	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El estudiante o profesor selecciona una recomendación de un evento.
	2	El sistema obtendrá las coordenadas almacenadas en la ontología del evento seleccionado.
	3	La aplicación móvil envía las coordenadas a la aplicación “Google Maps” para que esta muestre la ubicación del evento seleccionado.
4	El estudiante o profesor vuelve atrás y regresa a la pantalla principal de las recomendaciones.	
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Requerido	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 22  
*Caso de uso de mostrar noticia detallada*

<b>Identificador</b>	CU-08	
<b>Nombre</b>	Mostrar noticia detallada.	
<b>Descripción</b>	Mostrar al usuario todos los detalles de una noticia seleccionada	
<b>Precondición</b>	Iniciar sesión. Cualquiera de los casos de uso: CU-04, CU-05	
<b>Postcondición</b>	Visualizar el contenido detallado de la noticia	
<b>Actores</b>	Estudiantes, Profesores	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El estudiante o profesor selecciona una recomendación de una noticia
	2	El sistema obtendrá toda la información almacenada en la ontología acerca de la noticia seleccionada.
	3	El sistema muestra una nueva pantalla en la aplicación móvil con el título de la noticia y toda la información de la misma.
4	El estudiante o profesor vuelve atrás y regresa a la pantalla principal de las recomendaciones.	
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Requerido	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

## CAPÍTULO 4

### 4. CONSTRUCCIÓN DE UNA RED DE ONTOLOGÍAS PARA REPRESENTAR CONTEXTO

En este capítulo se realizará el proceso para definir la ontología que se utiliza en el sistema, la metodología que se decidió utilizar debido a su fácil aplicación es NeOn.

#### 4.1 Especificación de requerimientos de la ontología

Primeramente, se necesita realizar una lluvia de ideas que ayude a determinar el propósito, el alcance y el lenguaje de implementación de la ontología. Luego se analiza las aplicaciones y las necesidades de contexto que puede tener la ontología en el ámbito académico universitario, y a partir de esto se determinan los usuarios y los usos previstos de la ontología. La Tabla 23 muestra los resultados del análisis inicial.

Tabla 23

*Especificación de requerimientos de la ontología sección 1 a 5*

<b>1. Propósito</b>
El objetivo de la ontología es proporcionar información sobre los eventos y noticias, tomando en cuenta el contexto en el que se encuentra el usuario. Los contextos a considerarse son: preferencias, ubicación y tiempo.
<b>2. Alcance</b>
La ontología se centra en la recomendación de eventos y noticias en entornos universitarios, por lo que la información que se necesita almacenar en la ontología es : Organización de la universidad (facultades, escuelas, departamentos, asignaturas, etc.). Preferencias del Usuario. Ubicación, tiempo y detalle de los eventos y noticias en la universidad. Ubicación y descripción de los lugares. Información de usuarios (nombre, apellido, nombre usuario, etc.).
<b>3. Lenguaje de implementación.</b>
El lenguaje que se utilizará para la implementación de la ontología es OWL/XML y RDF
<b>4. Usuarios finales previstos.</b>
Los Usuarios finales son los que utilizarán la aplicación para consultar los eventos que se dan en las instalaciones de la universidad, así como las noticias que acontecen en la misma. Tomando en cuenta que la información es personalizada, dependiendo del perfil del usuario, el cual puede ser: Docente: corresponde a los docentes de la universidad. Estudiante: corresponde a los estudiantes que pertenecen a la universidad.
<b>5. Usos previstos.</b>
Consultas de: Eventos Académicos de la universidad. Noticias acerca de la universidad y de los intereses del usuario. Ubicación y tiempo de los eventos. Eventos cercanos al Usuario.

Fuente: Elaboración propia (2017)

Los requerimientos funcionales se determinan mediante la técnica especificada por Gruninger & Fox (1995), que consiste en una serie de preguntas de competencia escritas en lenguaje natural, las cuales deben ser respondidas por la ontología. La Tabla 24 muestra una lista ordenada de requisitos funcionales y no funcionales de la ontología.

Tabla 24  
Especificación de requerimientos de la ontología sección 6

6. Requisitos.
6.1. Requisitos no funcionales.
Las relaciones y componentes de la ontología se definen en inglés, también se reutilizan vocabularios existentes desarrollados por organizaciones dedicadas a la investigación y construcción de ontologías.
6.2. Requisitos funcionales: Preguntas de competencia.
<p><b>Personas</b></p> <p>PC1. ¿Qué actividad desempeña la persona? Estudiante, Profesor</p> <p>PC2. ¿Dónde está localizada la persona? Facultad de Filosofía</p> <p>PC3. ¿Qué carrera cursa el estudiante? Ingeniería de Sistemas y Telemática</p> <p>PC4. ¿Qué materias dicta el profesor? Ingeniería de Software</p> <p>PC5. ¿A qué Facultad pertenece la carrera? Facultad de Administración de Empresas</p> <p>PC6. ¿Cuáles son las áreas de conocimiento de su interés? Computación Tecnología de protección del medio ambiente Telecomunicaciones</p> <p>PC7. ¿En qué horario el estudiante tiene clases del curso Ingeniería de Software esta semana? Lunes 07h00 – 09h00, Miércoles 07h00 – 09h00.</p> <p><b>Eventos</b></p> <p>PC8. ¿De qué tipo es el evento? Congreso</p> <p>PC9. ¿En qué lugar y horario se realizará el evento? En el Auditorio, 11h00 – 13h00.</p> <p>PC10. ¿Cuál es la temática del evento? Marketing tecnológico.</p> <p>PC11. ¿Cuál es el área de conocimiento del evento? Terapias alternativas y complementarias.</p> <p>PC12. ¿Quién es el expositor del evento? Juan Pérez.</p> <p>PC13. ¿Qué eventos de interés del estudiante se han planificado la semana actual? Congreso de Sistemas de Información Geográfica Feria de Innovación y emprendimiento</p> <p>PC14. ¿Qué eventos cerca de la localización del profesor se han planificado el día de hoy? Feria de Innovación y emprendimiento</p> <p><b>Noticias</b></p> <p>PC15. ¿Cuáles son las noticias de interés del estudiante del día de hoy? La UDA presentó su curso de gestión de museos.</p> <p>PC16. ¿A qué lugar se refiere la noticia? Facultad de Ciencia y Tecnología</p>

Fuente: Elaboración propia (2017)

La sección 7 de la especificación de requerimientos de la ontología se divide en 3 partes que son descritas a continuación:

Términos extraídos de las preguntas de competencia: es una lista que contiene los nombres, adjetivos y verbos encontrados en las preguntas de competencia obtenidas en la sección 6.2 de la especificación de requerimientos de la ontología. En esta lista también se incluye el número de veces que se encontró cada término en las diferentes preguntas.

Términos extraídos de las respuestas a las preguntas de competencia: es una lista que contiene los nombres, adjetivos y verbos que se encuentran en las respuestas de las preguntas de competencia. En esta lista también se incluye el número de veces que se encontró cada término en las respuestas de las diferentes preguntas de competencia.

Términos que representan a nombres de entidades u objetos: es una lista que contiene los objetos encontrados tanto en las preguntas de competencia como en sus respuestas.

En la Tabla 25 se muestra la sección final de la especificación de requerimientos de la ontología.

Tabla 25  
*Especificación de requerimientos de la ontología sección 7*

7. Pre - glosario de términos	
7.1. Términos de las preguntas de competencia.	
Actividad	1
Persona	2
Desempeña	1
Localizada, lugar	4
Carrera	2
Cursar	1
Estudiante	4
Materia	1
Dictar	1
Profesor	2
Facultad	1
Pertenecer	1
Áreas de conocimiento	1
Interés	3
Horario	2
Clase	1
Curso	1
Semana	2
Evento	7
Tipo	1
Temática	1

Expositor	1
Planificar	2
Cerca	1
Noticia	2
<b>7.2. Términos de las respuestas a las preguntas.</b>	
Estudiante	1
Profesor	1
Facultad	3
Carrera	1
Curso	1
<b>7.3 Objetos</b>	
Ingeniería de Sistemas y telemática, Ingeniería de Software, Telecomunicaciones, computación, facultad de administración de empresas, Facultad de ciencia y tecnología, tecnología de protección del medio ambiente, marketing tecnológico, innovación y emprendimiento terapias alternativas y complementarias, Juan Pérez	

Fuente: Elaboración propia (2017)

## 4.2 Modelamiento de la Ontología

Una vez terminada la especificación de requisitos de la ontología se procede a realizar el modelamiento de la misma, iniciando con la definición de los escenarios, los cuales pueden ser uno o varios de los que están definidos por la metodología NeOn.

### 4.2.1 Definir los escenarios

Mediante el análisis realizado en la especificación de requisitos de la ontología se determinó que los escenarios que más se acoplan a nuestros requerimientos son los siguientes:

**Escenario 2:** Desarrollo de redes ontológicas mediante reutilización y reingeniería de recursos no ontológicos.

**Escenario 6:** Desarrollo de redes ontológicas mediante reutilización, mezcla y reingeniería de recursos ontológicos.

### 4.2.2 Búsqueda de recursos ontológicos

Debido a que existe una gran cantidad de ontologías ya desarrolladas, probadas e implementadas en otros sistemas, es necesario buscar ontologías que se adapten a los requerimientos establecidos, ahorrando tiempo en el desarrollo y pruebas de la ontología.



A continuación, se presenta una lista de ontologías encontradas organizadas por categorías.

Tabla 26  
*Ontologías encontradas organizadas en categorías*

<b>Ontologías</b>	<b>Conceptos</b>
<b>Personas</b>	
Foaf (Friend of a Friend).	Conceptos: Personas , relaciones entre personas, gustos e interés de personas, etc.
GUMO.	Conceptos: Personas.
Person.	Conceptos: información básica de personas como nombres, edad, título, género, identificación, etc.
<b>Organizaciones Académicas</b>	
AIIISO.	Conceptos: Organizaciones y universidades, facultades, escuelas, cursos, grupos, etc.
VIVO-ISF.	Conceptos: Organizaciones académicas, escuelas, facultades, organizaciones administrativas.
Univ-bench	Estudiantes, profesores, asignaturas, cursos, facultades, escuelas, etc.
<b>Tiempo</b>	
OWL-Time.	Conceptos: Tiempo, fechas, horas.
CACOnt.	Conceptos: Tiempo, fechas, horas, intervalos de tiempo, intervalos de fechas, instantes, etc.
<b>Geo localización</b>	
WGS84_Pos.	Conceptos: Geo localización, lugares, polígonos, puntos, etc.
SOUPA	Conceptos: Geo localización, coordenadas, lugares, etc.
Geo-owl	Conceptos: Geo localización, coordenadas, sistemas de coordenadas, etc.
<b>Eventos</b>	
Event.	Nombre de evento, lugar, hora, fecha, hora de inicio, hora de fin, etc.
COBRA-ONT	Nombre de evento, lugar, hora, fecha, etc.
SOUPA	Nombre de evento, lugar, hora, fecha, etc.

Fuente: Elaboración propia (2017)

### 4.2.3 Comparación y selección de ontologías

Solo se tomarán en cuenta las ontologías que cuentan con documentación completa, confiable y entendible para evitar la pérdida de tiempo en entender los conceptos y el diseño de la ontología.

Después de realizar un análisis y comparación de las ontologías encontradas se tiene que las ontologías más fáciles de implementar y que cumplen con la mayoría de requerimientos obtenidos son las siguientes:

- a) FOAF.

- b) AISO.
- c) OWL-Time.
- d) WGS84-Pos.
- e) Event.

#### **4.2.4 Fusión de Ontologías**

Una vez obtenidas las ontologías a reutilizar, lo siguiente que se realiza es una reingeniería de las ontologías para adaptar los conceptos de cada una de ellas a un contexto académico universitario. Para esto se eliminan, modifican o agregan relaciones y conceptos para obtener como resultado una ontología única y adaptada a nuestro contexto.

#### **4.2.5 Búsqueda de recursos no ontológicos**

Cuando ya se tienen todos los recursos ontológicos enlazados y adaptados lo siguiente es buscar los recursos no ontológicos, es decir, buscar la manera de incorporar a la ontología los conceptos que no se cubrieron con los pasos anteriores. Para este caso particular es necesario incorporar a la ontología la información relacionada con las áreas de conocimientos que tienen como preferencia los usuarios, y en las que se pueden enmarcar las carreras y los eventos académicos. Para incluir estos requisitos de información a la ontología se utilizó el Artículo 15 del Reglamento de Armonización de la Nomenclatura de Títulos Profesionales y Grados Académicos emitida por el Consejo de Educación Superior de la república del Ecuador que hace referencia a los campos del conocimiento tomados como referencia de la Clasificación Internacional Normalizada de la Educación (CINE) (CES, 2016).

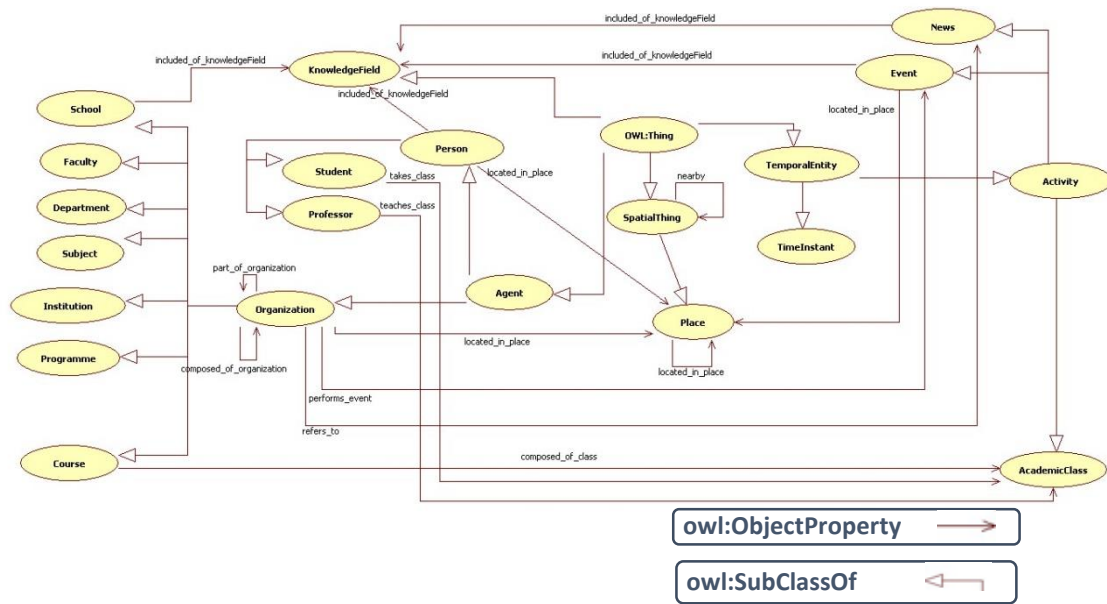
#### **4.2.6 Transformación de recursos no ontológicos**

A partir de la información obtenida en el paso anterior, lo que se hizo es convertir dicha información en conceptos y relaciones para integrarlos a la ontología ya creada, de manera que todas las necesidades sean finalmente cubiertas y la ontología quede terminada.

#### **4.2.7 Ontología Resultante**

La Figura 23 muestra la ontología obtenida a partir de los análisis y procedimientos realizados en los pasos anteriores.

Figura 23  
Ontología resultante



Fuente: Elaboración propia (2017)

A continuación en la Tabla 27, se presenta una tabla con las clases y la ontología a la que pertenecen.

Nota: las clases creadas pertenecen a la ontología llamada uda.

Tabla 27  
Clases de la ontología

Clase	Ontología
Person	FOAF
Student	Univ-bench
Professor	Univ-bench
Organization	AIISO
School	AIISO
Faculty	AIISO
Department	AIISO
Subject	AIISO
Institution	AIISO
Programme	AIISO
Course	AIISO
KnowledgeField	uda
SpatialThing	WGS84-Pos
Place	WGS84-Pos

TemporalEntity	OWL-Time
TimeInstant	OWL-Time
Activity	uda
Event	Event
News	uda
AcademicClass	uda

Fuente: Elaboración propia (2017)

Los *Object Property* se presentan en la siguiente Tabla 28.

Tabla 28  
*Object Property de la ontología*

<b>ObjectProperty</b>	<b>Dominio</b>	<b>Rango</b>
composed_of_class	Course	AcademicClass
composed_of_organization	AIISO	AIISO
included_of_knowledgefield	News,Event,School,Person	KnowledgeField
located_in_place	Place,Person,Event,Organization	Place
nearby	SpatialThing	SpatialThing
part_of_organization	AIISO	AIISO
performds_event	AIISO	Event
refers_to	News	Person,Organization
takes_class	Student	AcademicClass
teaches_class	Professor	AcademicClass

Fuente: Elaboración propia (2017)

Los *Data Property* se presentan en la siguiente Tabla 29.

Tabla 29  
*Data Property de la Ontología*

<b>DataProperty</b>	<b>Dominio</b>	<b>Rango</b>
cellphone	Person	literal
firstName	Person	literal
email	Person	literal
lastName	Person	literal
title	Person	literal
userName	Person	literal
altitude	SpatialThing	decimal
latitude	SpatialThing	decimal
longitude	SpatialThing	decimal
detaileKnowledgeField	KnowledgeField	literal
generalKnowledgeField	KnowledgeField	literal
specificKnowledgeField	KnowledgeField	literal
eventDescription	Event	literal
eventTitle	Event	literal
eventType	Event	literal
In XSD Date-time	TimeInstant	dateTime
In XSD Date-time-tamp	TimeInstant	dateTimeStamp
In XSD g-Year	TimeInstant	gYear
In XSD g-Year-Month	TimeInstant	gYearMonth
newsDescription	News	literal
newsTitle	News	literal
placeDescription	Place	literal
placeName	Place	literal
organizationName	AIISO	literal
has XSD duration	TemporalEntity	duration

Fuente: Elaboración propia (2017)

# CAPÍTULO 5

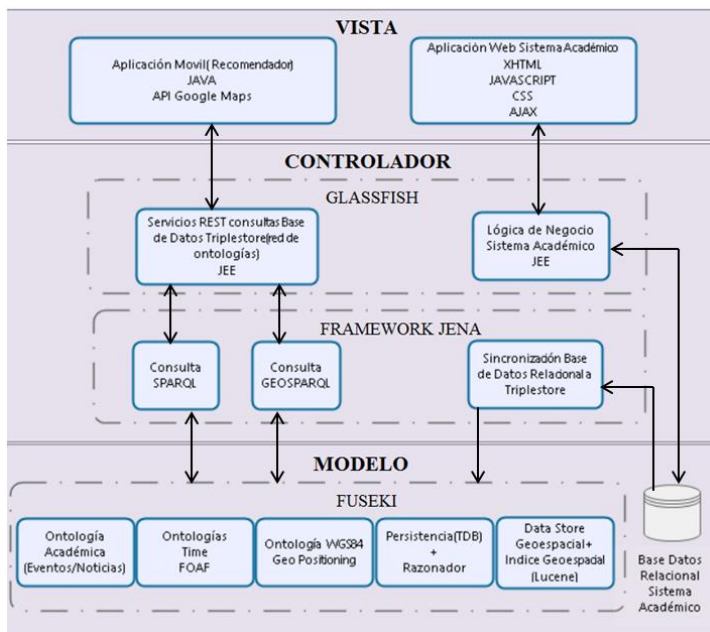
## 5. DISEÑO DEL SOFTWARE

### 5.1 Diseño de arquitectónico

Para representar el diseño se utilizó la arquitectura modelo-vista-controlador que permite diferenciar la lógica de control, la interfaz y los datos del recomendador móvil. El modelo representa los datos que se maneja, así como también la persistencia y como se accede al almacenamiento de datos, en la vista o interfaz de usuario se explica que es lo que obtiene el cliente y la interacción que realiza el mismo y en el controlador recibe los eventos que realiza el usuario, por ejemplo una actualización de datos, gestiona la información que se envía entre el modelo y la vista, es decir, es un intermediario entre modelo y vista que adapta los datos para representarlos en los dos lados.

El recomendador móvil tiene el modelo arquitectónico representado en la Figura 24.

Figura 24  
Modelo Vista Controlador



Fuente: Elaboración propia (2017)

### **5.1.1 Aplicación móvil**

Estudiantes y profesores interactúan con la aplicación móvil y eligen entre las diferentes opciones: eventos por tipo, eventos lugar, noticias por tiempo, etc. Con la posibilidad de guardar las configuraciones de preferencias (tiempo para refrescar, distancia, etc.). Cuando el usuario elija una de las opciones la aplicación móvil envía una solicitud http al controlador, según la opción seleccionada se envían diferentes parámetros y espera una respuesta por parte del servidor Glassfish.

El controlador recibe la acción solicitada por el usuario mediante servicios web de tipo REST que son gestionados por el servidor Glassfish, el mismo que gestiona que tipo de consulta SPARQL o GEOSPARQL debe realizarse al servidor FUSEKI. Luego de armar una de las consultas se comunica con el servidor FUSEKI y las ejecuta.

Se accede al modelo de la arquitectura cuando FUSEKI ejecuta la consulta y devuelve las tripletas con la información de las recomendaciones. El controlador del servidor Glassfish obtiene esta respuesta y arma un esquema JSON con las recomendaciones. Esta respuesta en formato JSON se envía a la aplicación móvil, la cual representa los datos mediante una lista en la pantalla del usuario.

El usuario obtiene la información que solicito de manera que se pueda entender y la aplicación móvil espera otra acción por parte del usuario para repetir el ciclo.

### **5.1.2 Aplicación web**

El controlador del servidor de aplicaciones Glassfish se encarga de recibir y ejecutar todas las solicitudes realizadas por los usuarios administradores.

La vista de la aplicación web presenta al usuario una serie de botones y cuadros de texto con los que puede consultar, ingresar, modificar y eliminar información de la base de datos relacional.

En el caso de las consultas, el controlador del servidor Glassfish recibe una solicitud de la aplicación web y este responde con los 30 primeros registros encontrados en la base de datos. La aplicación web obtiene la respuesta del controlador y la organiza en forma de tablas, las cuales tienen cuadros de texto en las cabeceras para filtrar la información presentada.

Cuando el usuario administrador desea ingresar, eliminar o modificar información, la aplicación web envía una solicitud al controlador Glassfish, el mismo que verifica que la información recibida es correcta y decide si la transacción se realiza o no.

El controlador aparte de realizar las funciones indicadas anteriormente, realiza otra función muy importante que es la de sincronización de los modelos base de datos relacional y base de datos triplestore, siendo esta última la que contiene la ontología, el razonador y la que responde a las consultas de la aplicación móvil. En este caso sincronizar hace referencia a tomar la información del modelo base de datos relacional e ingresarla en el modelo base de datos triplestore.

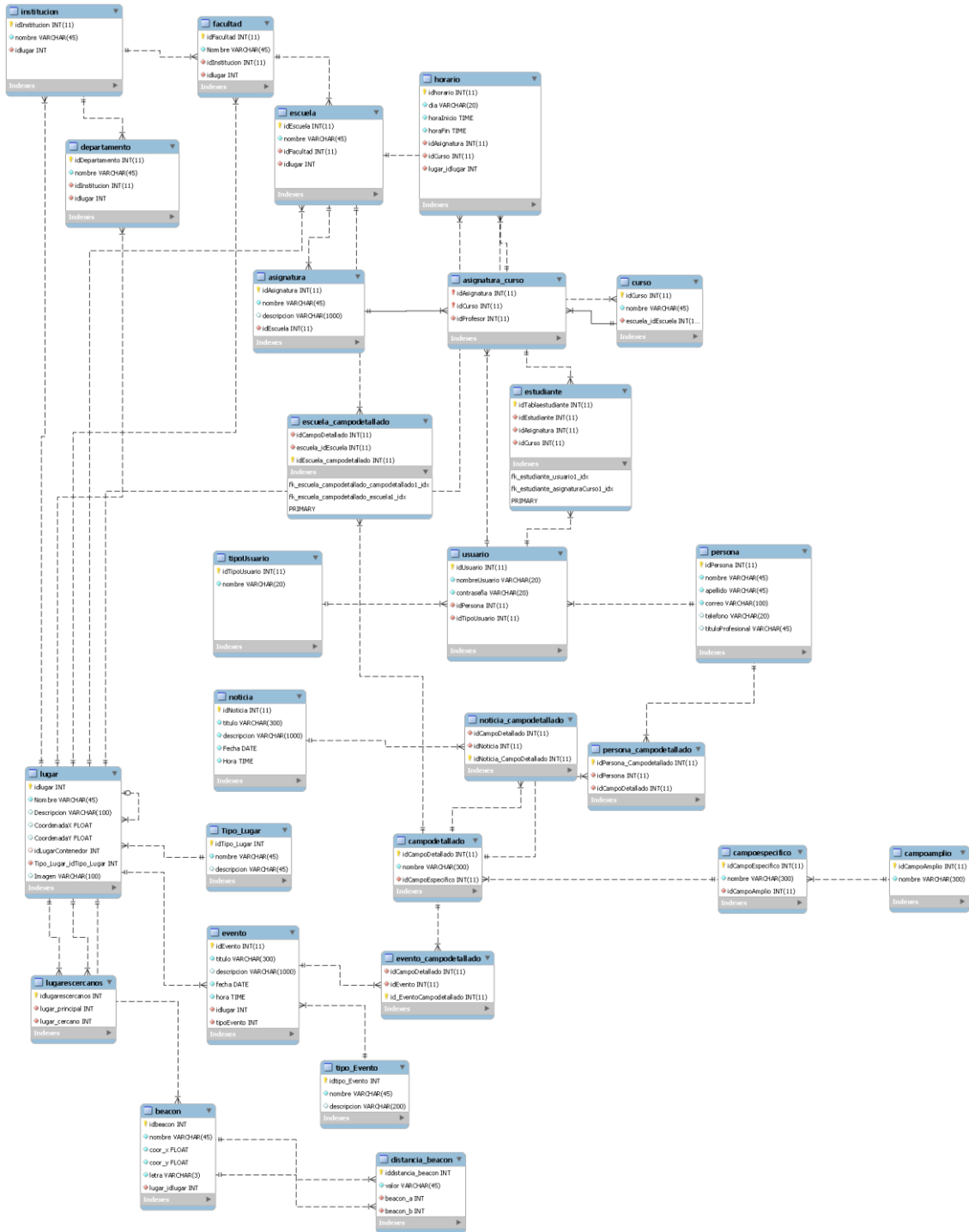
## **5.2 Diseño de la base de datos relacional**

El diseño de la base de datos se obtiene a partir de la información y los conceptos especificados en la ontología definida en el capítulo cuatro. El software que se utilizó para el diseño del modelo entidad relación es MySQL Workbench y el gestor de base de datos que se utilizó es MySQL.

En la Figura 25 se presenta el modelo entidad relación diseñado en MySQL Workbench.



Figura 25  
Diagrama entidad relación de la base de datos



Fuente: Elaboración propia (2017)

Nota: se debe tomar en cuenta que esta base de datos es utilizada también en el proyecto de localización de interiores por lo que existen algunas tablas que no se utilizan en este proyecto, tales como: beacon, distanciabeacon, etc.

En la siguiente tabla se presenta el diccionario de datos de la base de datos diseñada.

Tabla 30  
Diccionario de datos de la base de datos

Tipo_Lugar								
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI
idTipo_Lugar	INT	✓	✓					✓
nombre	VARCHAR(45)		✓					
descripcion	VARCHAR(45)							
asignatura								
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI
idAsignatura	INT(11)	✓	✓					✓
nombre	VARCHAR(45)		✓					
descripcion	VARCHAR(1000)							
idEscuela	INT(11)		✓					
asignatura_curso								
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI
idAsignatura	INT(11)	✓	✓					
idCurso	INT(11)	✓	✓					
idProfesor	INT(11)		✓					
beacon								
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI
idbeacon	INT	✓	✓					✓
nombre	VARCHAR(45)		✓					
coor_x	FLOAT		✓					
coor_y	FLOAT		✓					
letra	VARCHAR(3)		✓					
lugar_idlugar	INT		✓					
campoamplio								
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI
idCampoAmplio	INT(11)	✓	✓					✓
nombre	VARCHAR(300)		✓					
campodetallado								
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI
idCampoDetallado	INT(11)	✓	✓					✓

nombre	VARCHAR(300)		✓						
idCampoEspecifico	INT(11)		✓						
<b>campospecifico</b>									
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	
idCampoEspecifico	INT(11)	✓	✓						✓
nombre	VARCHAR(300)		✓						
idCampoAmplio	INT(11)		✓						
<b>curso</b>									
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	
idCurso	INT(11)	✓	✓						✓
nombre	VARCHAR(45)		✓						
escuela_idEscuela	INT(11)		✓						
<b>departamento</b>									
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	
idDepartamento	INT(11)	✓	✓						✓
nombre	VARCHAR(45)		✓						
idInstitucion	INT(11)		✓						
idlugar	INT		✓						
<b>distancia_beacon</b>									
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	
iddistancia_beacon	INT	✓	✓						✓
valor	VARCHAR(45)		✓						
beacon_a	INT		✓						
beacon_b	INT		✓						
<b>escuela</b>									
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	
idEscuela	INT(11)	✓	✓						✓
nombre	VARCHAR(45)		✓						
idFacultad	INT(11)		✓						
idlugar	INT		✓						
<b>escuela_campodetallado</b>									
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	
idCampoDetallado	INT(11)		✓						
escuela_idEscuela	INT(11)		✓						
idEscuela_campodetallado	INT(11)	✓	✓						✓
<b>estudiante</b>									
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	
idTablaestudiante	INT(11)	✓	✓						✓
idEstudiante	INT(11)		✓						
idAsignatura	INT(11)		✓						
idCurso	INT(11)		✓						

evento								
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI
idEvento	INT(11)	✓	✓					✓
titulo	VARCHAR(300)		✓					
descripcion	VARCHAR(1000)							
fecha	DATE		✓					
hora	TIME		✓					
idlugar	INT		✓					
tipoEvento	INT		✓					
evento_campodetallado								
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI
idCampoDetallado	INT(11)		✓					
idEvento	INT(11)		✓					
id_EventoCampodetallado	INT(11)	✓	✓					✓
facultad								
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI
idFacultad	INT(11)	✓	✓					✓
Nombre	VARCHAR(45)		✓					
idInstitucion	INT(11)		✓					
idlugar	INT		✓					
horario								
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI
idhorario	INT(11)	✓	✓					✓
dia	VARCHAR(20)		✓					
horaInicio	TIME		✓					
horaFin	TIME		✓					
idAsignatura	INT(11)		✓					
idCurso	INT(11)		✓					
lugar_idlugar	INT		✓					
institucion								
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI
idInstitucion	INT(11)	✓	✓					✓
nombre	VARCHAR(45)		✓					
idlugar	INT		✓					
lugar								
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI
idlugar	INT	✓	✓					✓
Nombre	VARCHAR(45)		✓					
Descripcion	VARCHAR(100)							
CoordenadaX	FLOAT							
CoordenadaY	FLOAT							

idLugarContenedor	INT								
Tipo_Lugar_idTipo_Lugar	INT		✓						
Imagen	VARCHAR(100)								
<b>lugarescercanos</b>									
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	
idlugarescercanos	INT	✓	✓						✓
lugar_principal	INT		✓						
lugar_cercano	INT		✓						
<b>noticia</b>									
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	
idNoticia	INT(11)	✓	✓						✓
titulo	VARCHAR(300)		✓						
descripcion	VARCHAR(1000)		✓						
Fecha	DATE		✓						
Hora	TIME		✓						
<b>noticia_campodetallado</b>									
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	
idCampoDetallado	INT(11)		✓						
idNoticia	INT(11)		✓						
idNoticia_CampoDetallado	INT(11)	✓	✓						✓
<b>persona</b>									
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	
idPersona	INT(11)	✓	✓						✓
nombre	VARCHAR(45)		✓						
apellido	VARCHAR(45)		✓						
correo	VARCHAR(100)		✓						
telefono	VARCHAR(20)								
tituloProfesional	VARCHAR(45)								
<b>persona_campodetallado</b>									
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	
idPersona_Campodetallado	INT(11)	✓	✓						✓
idPersona	INT(11)		✓						
idCampoDetallado	INT(11)		✓						
<b>tipoUsuario</b>									
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	
idTipoUsuario	INT(11)	✓	✓						✓
nombre	VARCHAR(20)		✓						
<b>tipo_Evento</b>									
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	
idtipo_Evento	INT	✓	✓						✓
nombre	VARCHAR(45)		✓						

descripcion	VARCHAR(200)								
usuario									
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	
idUsuario	INT(11)	✓	✓						✓
nombreUsuario	VARCHAR(20)		✓						
contraseña	VARCHAR(20)		✓						
idPersona	INT(11)		✓						
idTipoUsuario	INT(11)		✓						

Fuente: Elaboración propia (2017)

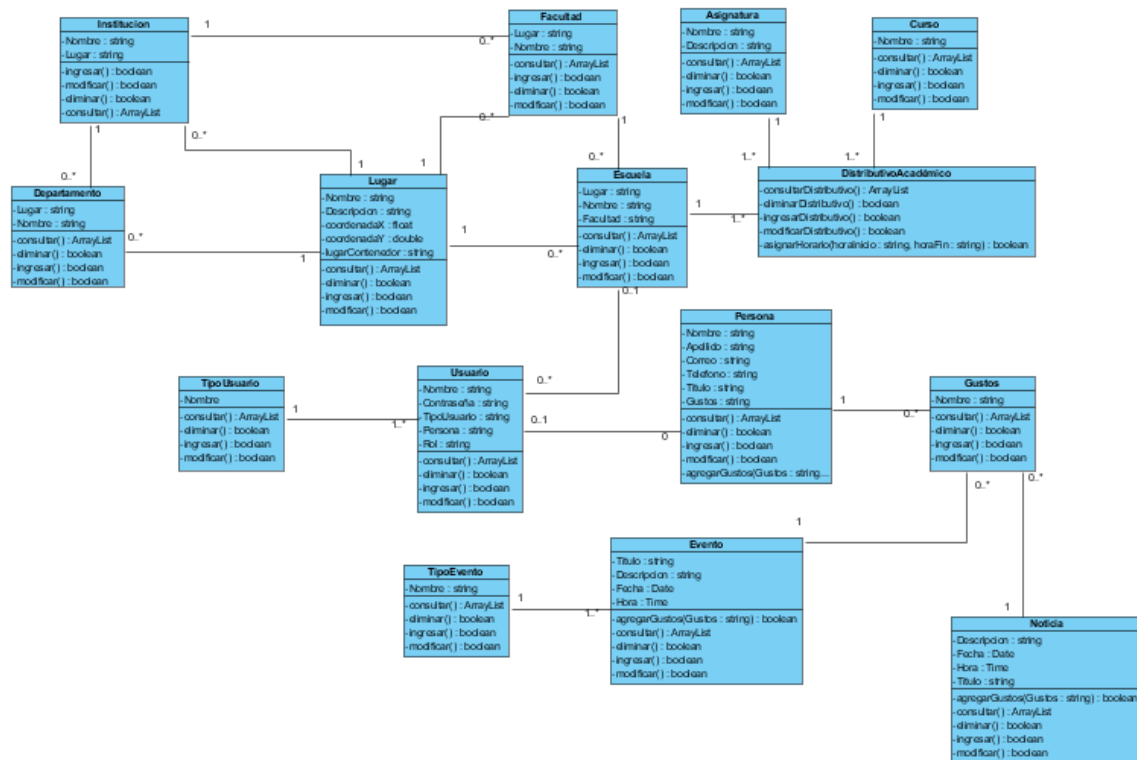
## 5.3 Diseño orientado a objetos

### 5.3.1 Diagramas de clases del sistema administrativo

El diagrama de clases del sistema administrativo se presenta en la Figura 26.

Figura 26

Diagrama de clases del sistema administrativo



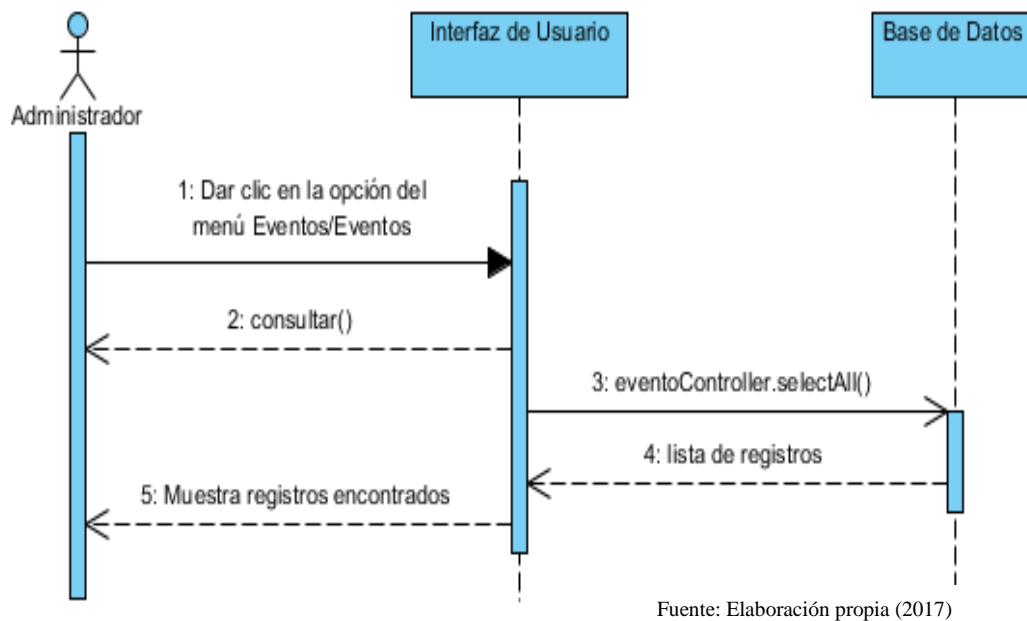
Fuente: Elaboración propia (2017)

### 5.3.2 Diagramas de secuencia de sistema administrativo

Debido a que todos los procesos de mantenimiento del sistema administrativo son idénticos, se ha decidido mostrar solamente los diagramas de secuencia del mantenimiento de eventos.

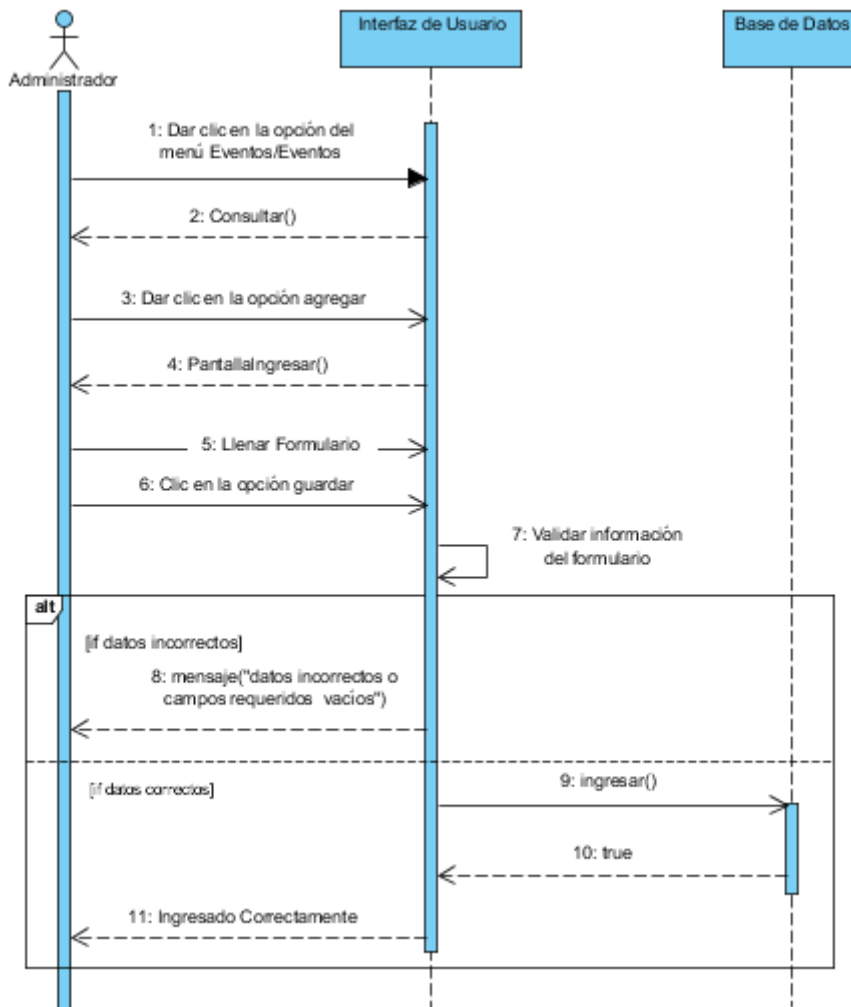
En la siguiente figura se presenta el diagrama de secuencia de consultar eventos.

Figura 27  
*Diagrama de secuencia para consultar eventos*



En la siguiente figura se presenta el diagrama de secuencia de ingresar eventos.

Figura 28  
*Diagrama de secuencia para ingresar eventos*

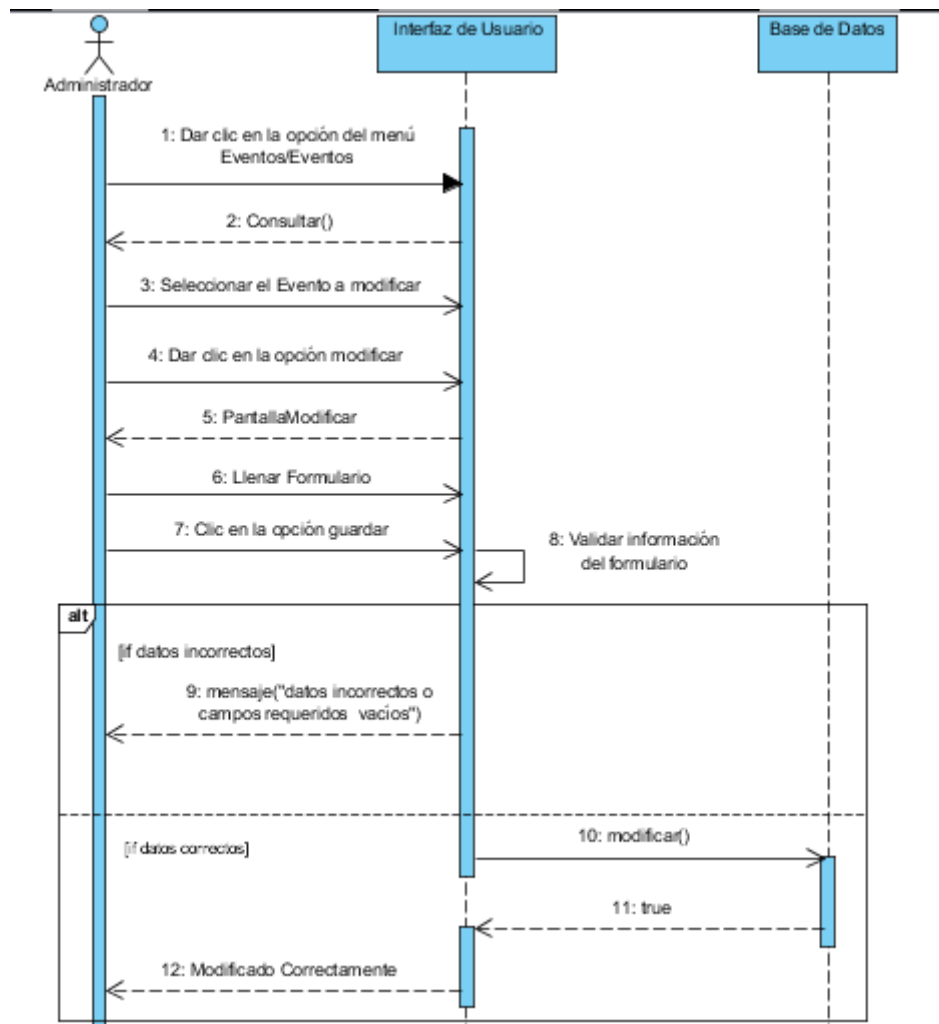


Fuente: Elaboración propia (2017)



En la siguiente figura se presenta el diagrama de secuencia de modificar eventos.

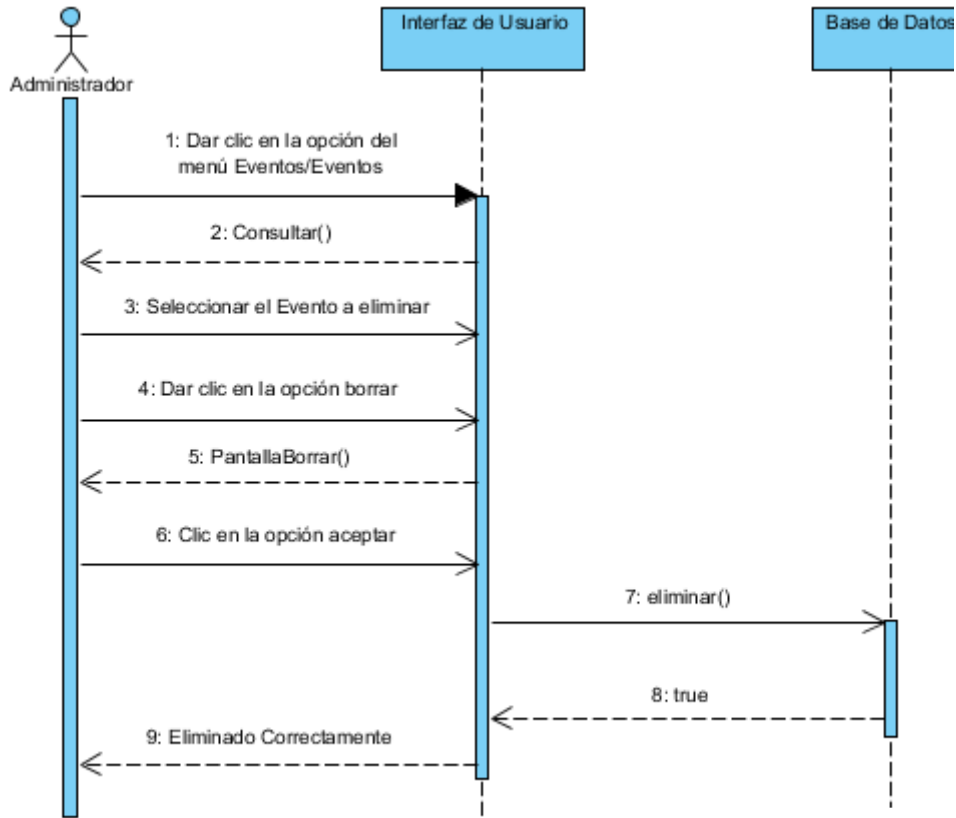
Figura 29  
*Diagrama de secuencia para modificar eventos*



Fuente: Elaboración propia (2017)

En la siguiente figura se presenta el diagrama de secuencia de eliminar eventos.

Figura 30  
*Diagrama de secuencia para eliminar eventos*

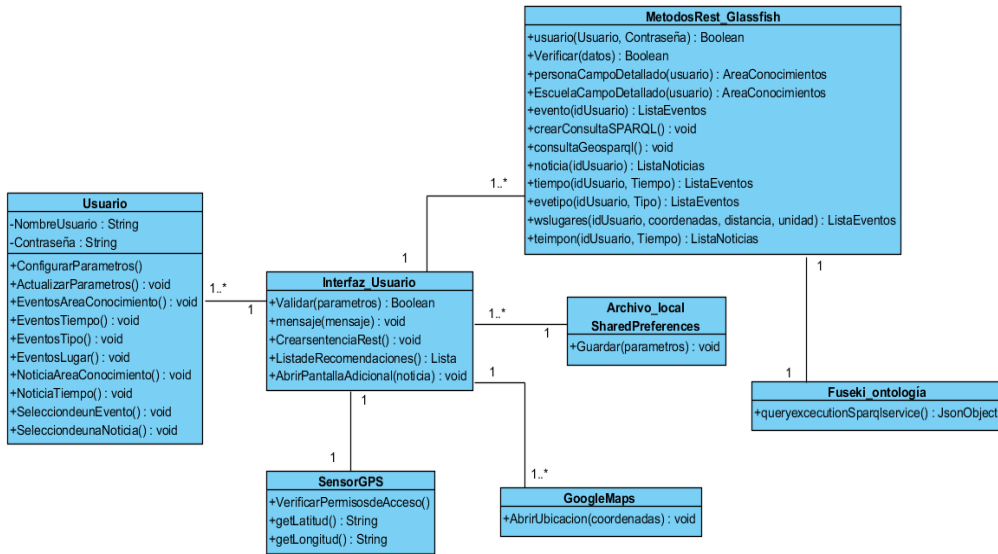


Fuente: Elaboración propia (2017)

### 5.3.3 Diagrama de clases de la aplicación móvil

En la Figura 31 se presenta el diagrama de clases de la aplicación móvil.

Figura 31  
Diagrama de clases de la aplicación móvil



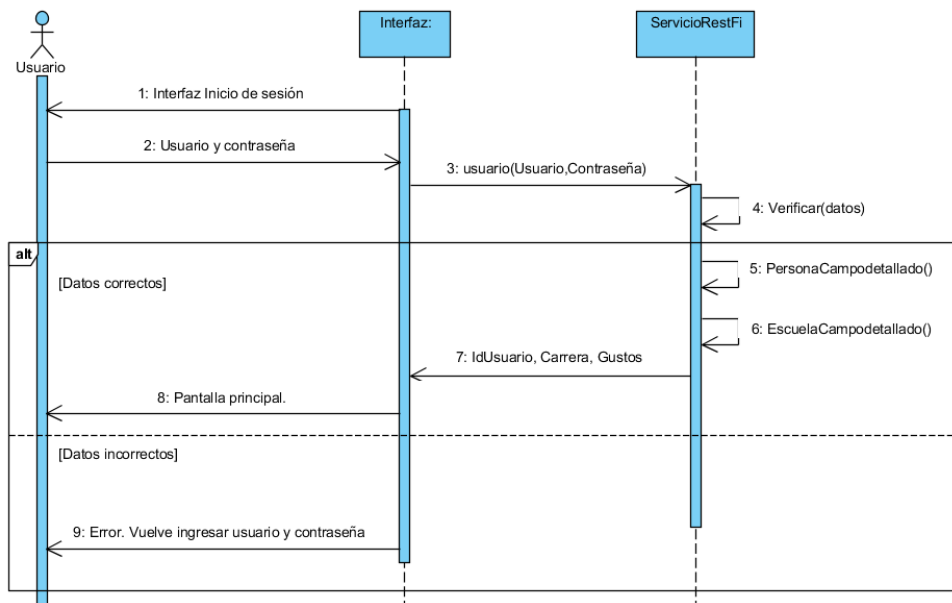
Fuente: Elaboración propia (2017)

### 5.3.4 Diagramas de secuencia de la aplicación móvil

A continuación se muestran los diagramas de secuencia de las acciones que se pueden realizar en la aplicación móvil.

En la siguiente figura se presenta el diagrama de secuencia de inicio de sesión.

Figura 32  
Diagrama de secuencia de inicio de sesión

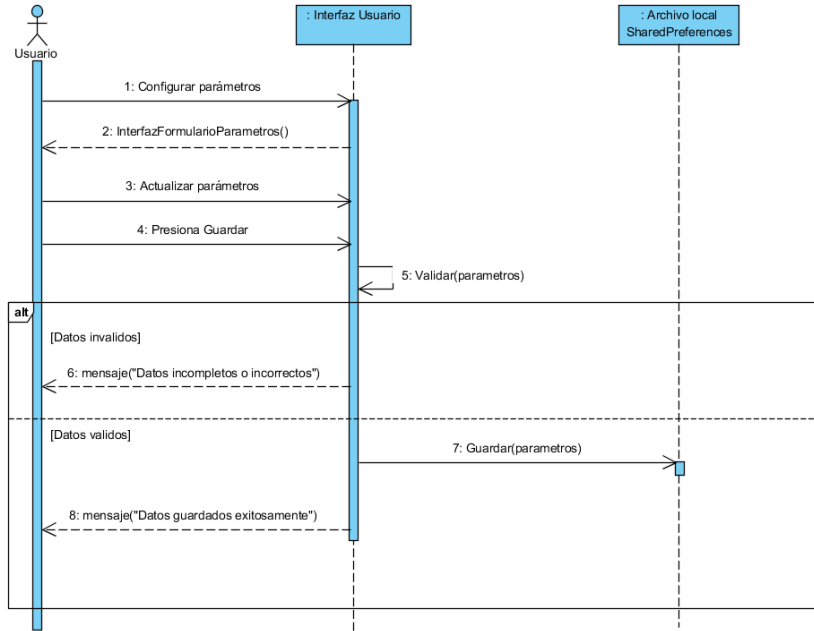


Fuente: Elaboración propia (2017)

En la siguiente figura se presenta el diagrama de secuencia de configurar parámetros.

Figura 33

Diagrama de secuencia de configuración de parámetros

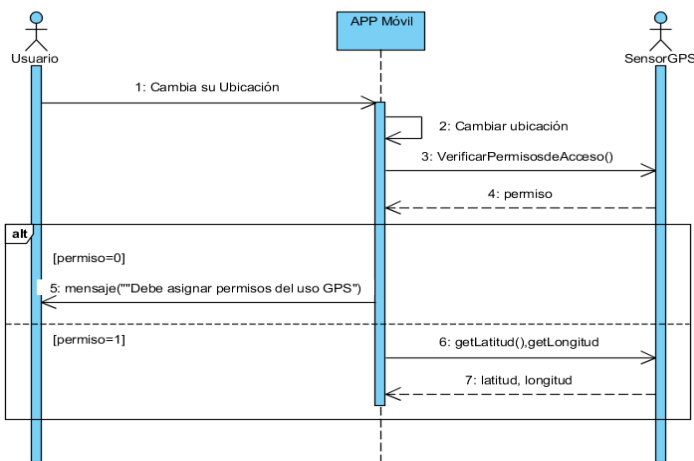


Fuente: Elaboración propia (2017)

En la siguiente figura se presenta el diagrama de secuencia de proporcionar ubicación.

Figura 34

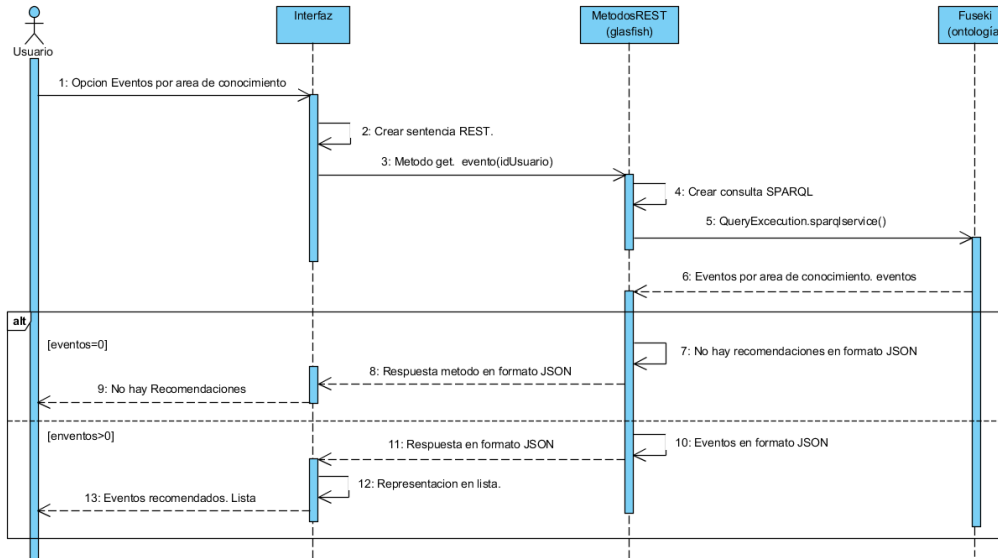
Diagrama de secuencia para proporcionar ubicación



Fuente: Elaboración propia (2017)

En la siguiente figura se presenta el diagrama de secuencia de obtener eventos por área de conocimiento.

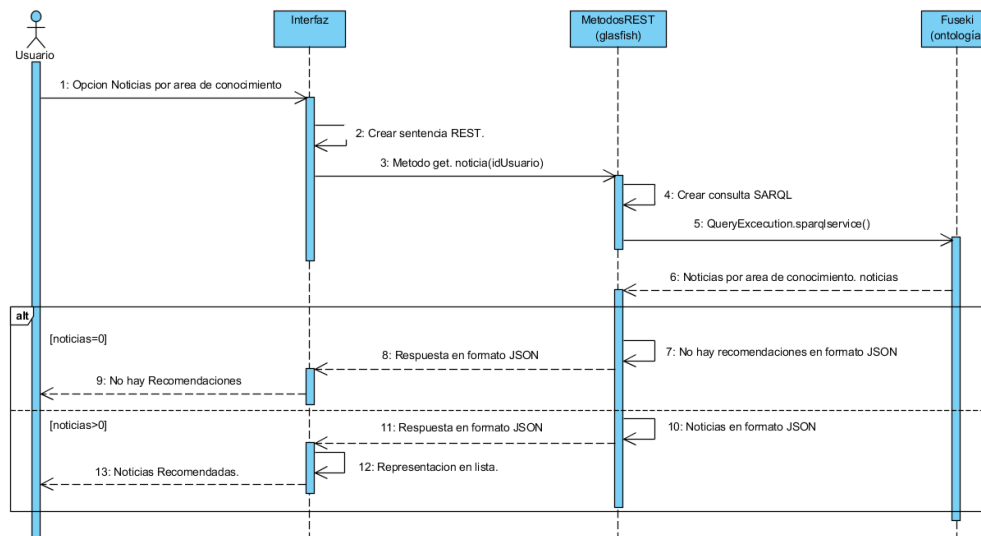
**Figura 35**  
*Diagrama de secuencia para obtener eventos por área de conocimiento*



Fuente: Elaboración propia (2017)

En la siguiente figura se presenta el diagrama de secuencia de obtener noticias área de conocimiento.

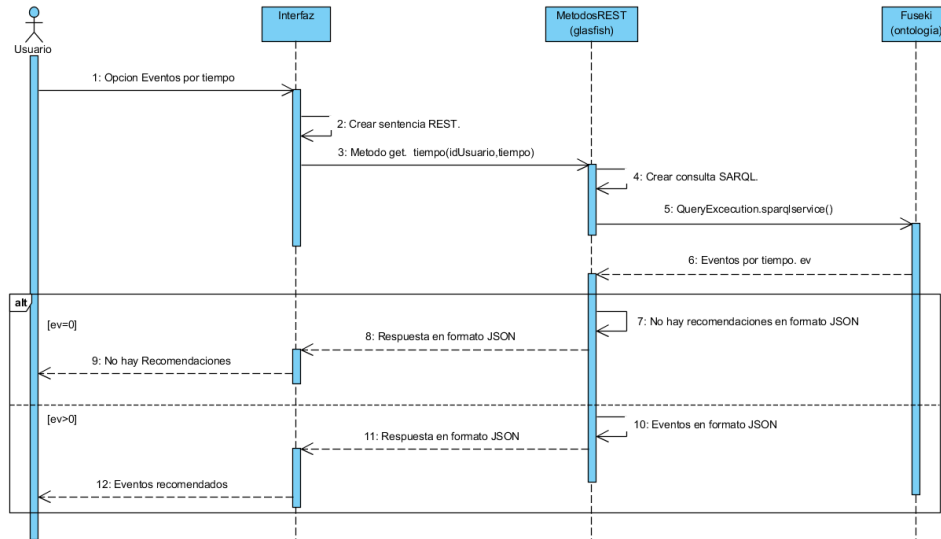
**Figura 36**  
*Diagrama de secuencia para obtener noticias por área de conocimiento*



Fuente: Elaboración propia (2017)

En la siguiente figura se presenta el diagrama de secuencia de obtener eventos por tiempo.

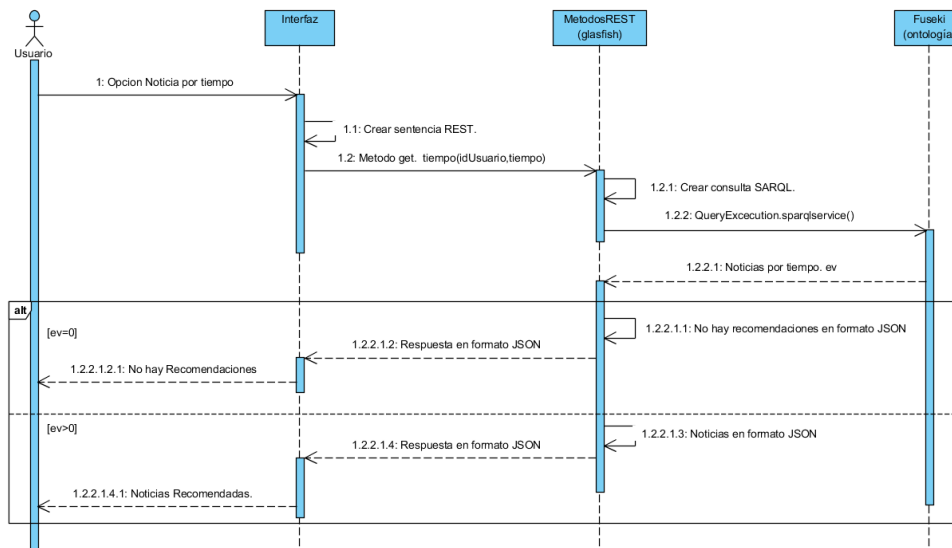
Figura 37  
Diagrama de secuencia para obtener eventos por tiempo



Fuente: Elaboración propia (2017)

En la siguiente figura se presenta el diagrama de secuencia de obtener noticias por tiempo.

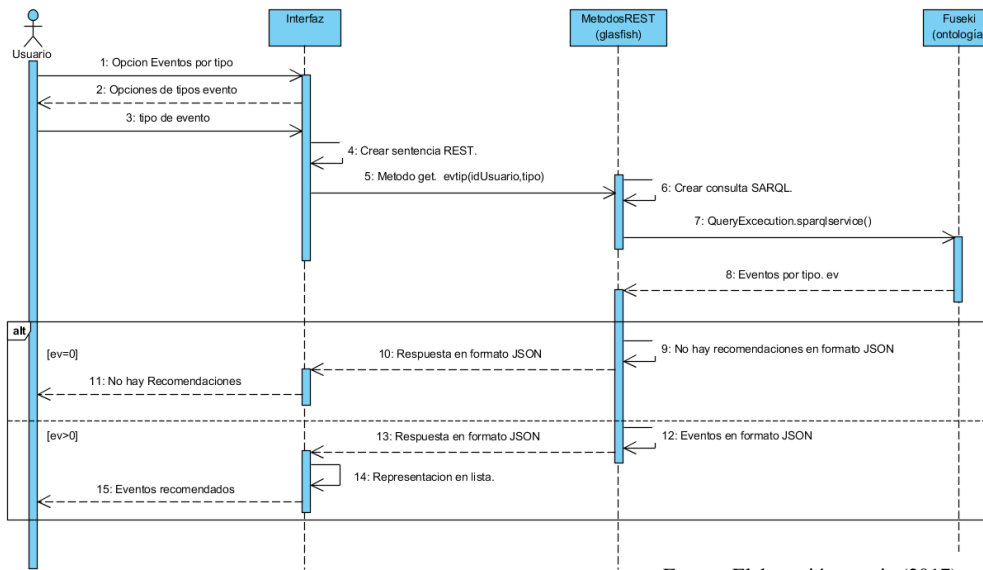
Figura 38  
Diagrama de secuencia para obtener noticias por tiempo



Fuente: Elaboración propia (2017)

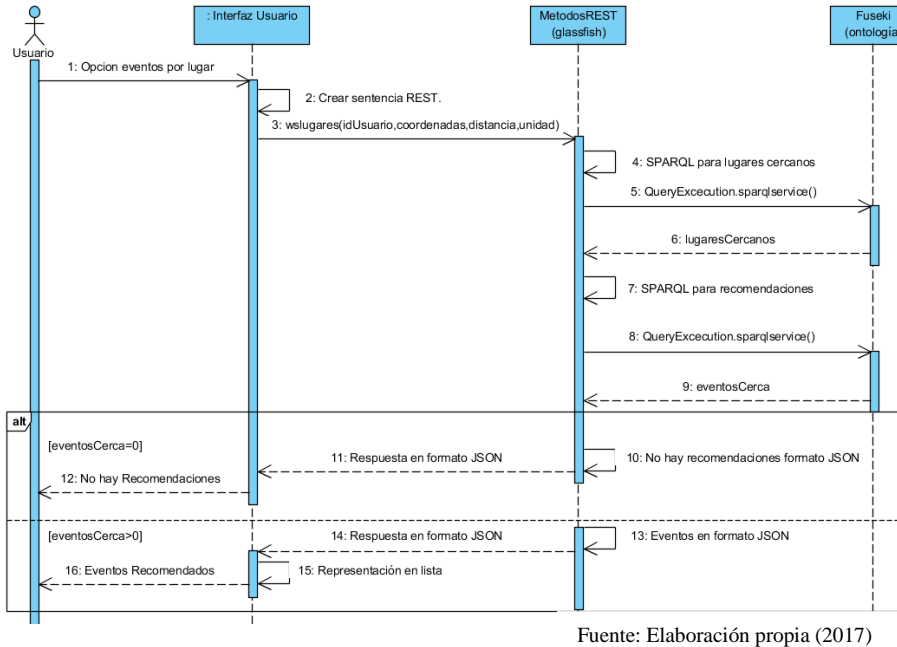
En la siguiente figura se presenta el diagrama de secuencia de obtener eventos por tipo.

Figura 39  
Diagrama de secuencia para obtener eventos por tipo



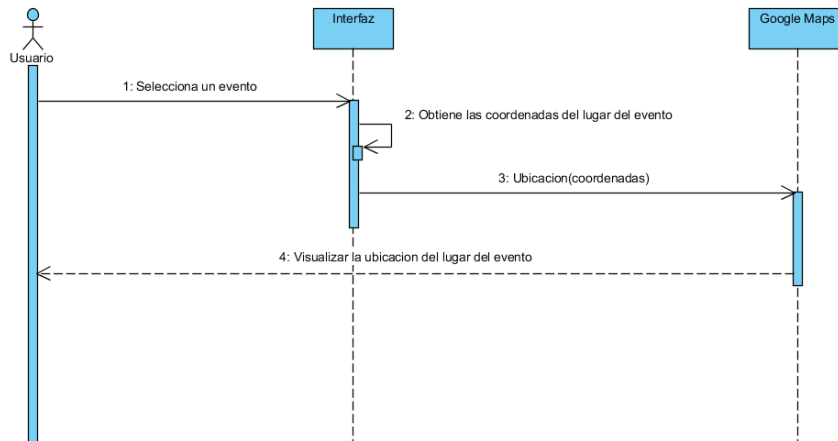
En la siguiente figura se presenta el diagrama de secuencia de obtener eventos por lugar.

Figura 40  
Diagrama de secuencia para obtener eventos por lugar



En la siguiente figura se presenta el diagrama de secuencia para visualizar la ubicación de un evento.

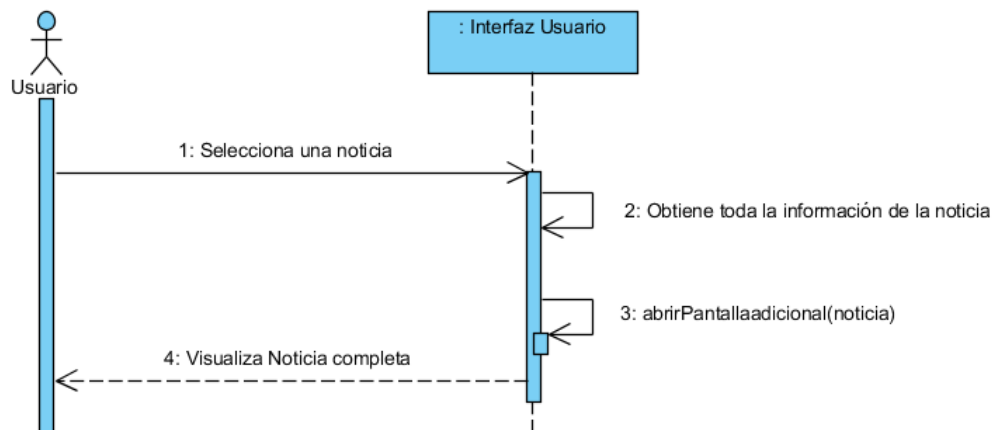
Figura 41  
 Diagrama de secuencia para visualizar la ubicación de un evento



Fuente: Elaboración propia (2017)

En la siguiente figura se presenta el diagrama de secuencia de visualizar la descripción detallada de una noticia.

Figura 42  
 Diagrama de secuencia para visualizar la descripción detallada de una noticia



Fuente: Elaboración propia (2017)

## 5.4 Diseño de interfaces de usuario de la aplicación web

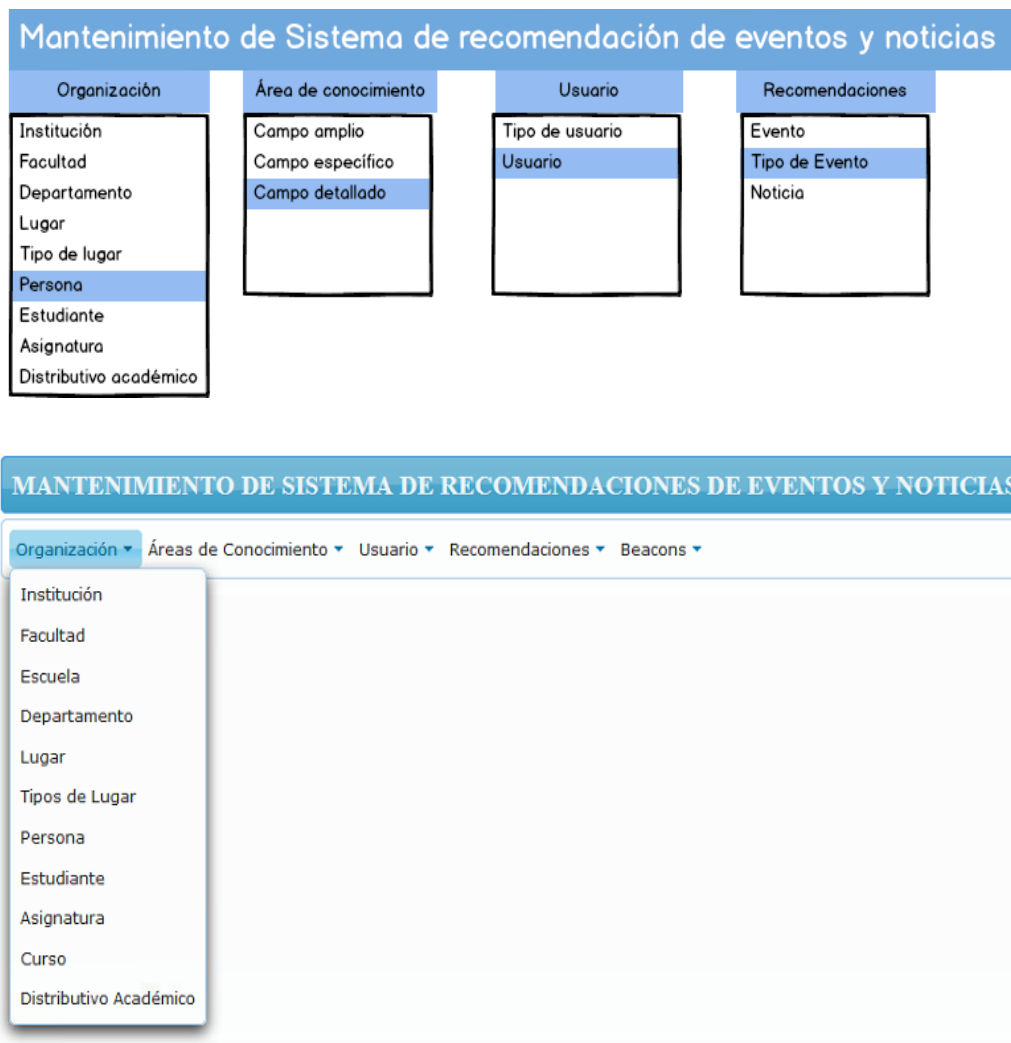
Debido a la similitud de las pantallas del sistema, las interfaces del sistema administrativo se han dividido en dos tipos: 1) pantalla principal, y 2) pantallas de mantenimientos.



Pantalla principal: el sistema presenta al usuario una pantalla que contiene un menú en el cual se han categorizado y ordenado todos los mantenimientos.

En la Figura 43 se presenta el diseño y el resultado final de la pantalla principal del sistema administrativo.

Figura 43  
Diseño y resultado final de la pantalla principal



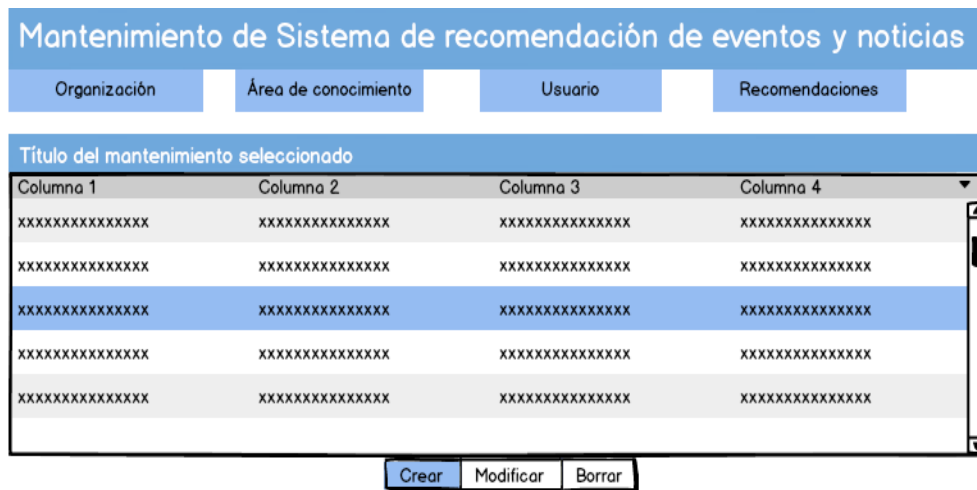
Fuente: Elaboración propia (2017)

Pantalla de mantenimiento: es una pantalla similar a la principal debido a que los menús que contienen las opciones de mantenimiento se encuentran visibles en todas las pantallas del sistema.

En la pantalla de mantenimiento se agrega una tabla en la que se muestra toda la información requerida por el usuario y en la parte inferior tres botones encargados de las funciones crear, modificar y eliminar.

En las siguientes figuras se muestra el diseño y el resultado final de la pantalla de mantenimiento.

Figura 44  
Diseño y resultado final de pantalla de mantenimiento



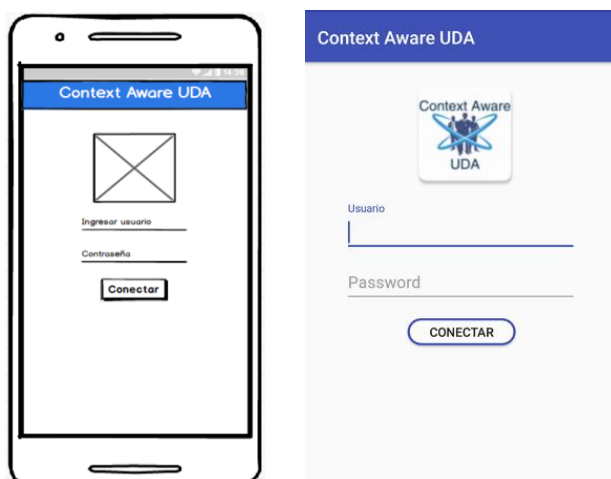
Fuente: Elaboración propia (2017)

## 5.5 Diseño de interfaces de la aplicación móvil

Se realizó el diseño de las interfaces de la aplicación móvil con el objetivo de obtener una vista preliminar de las pantallas que se presentarán al usuario final. Se utilizó el software llamado Balsamiq Mockups A continuación se describe cada una de las pantallas:

**Inicio de Sesión:** la interfaz presenta al usuario una pantalla donde se tiene una imagen del logo del recomendador móvil y dos campos de texto donde el usuario ingresa su código y la contraseña, además de un botón para conectarse. En caso de que el usuario ingrese mal los datos, aparecen mensajes de error a un lado del campo que contiene los datos incorrectos, para que estos sean ingresados nuevamente.

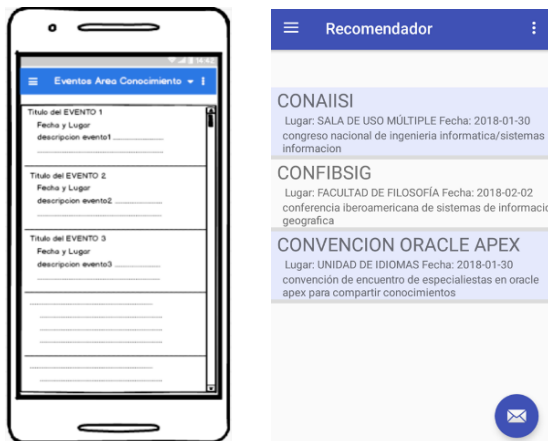
Figura 45  
*Pantalla de inicio de sesión*



Fuente: Elaboración propia (2017)

**Pantalla principal:** se muestra al usuario una vez que el mismo haya iniciado sesión. Esta pantalla presenta las recomendaciones para el usuario en forma de lista, además desde esta pantalla se puede acceder al menú principal de la aplicación móvil, ya sea por el botón con el ícono ☰, ubicado en la parte superior izquierda o simplemente deslizando la pantalla de izquierda a derecha.

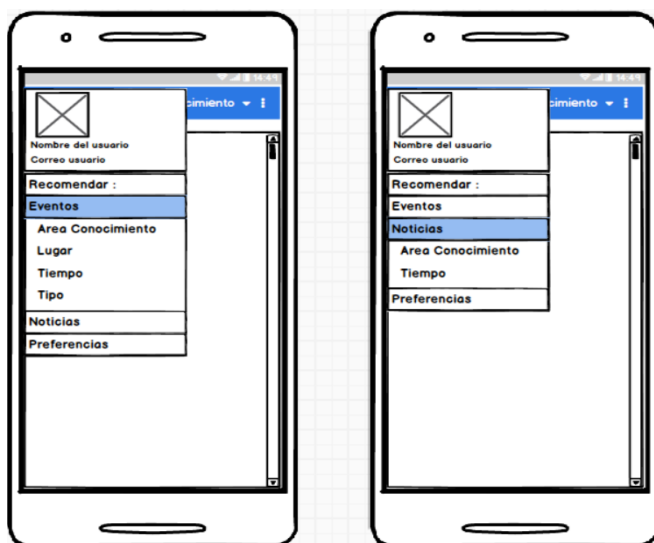
Figura 46  
Pantalla principal

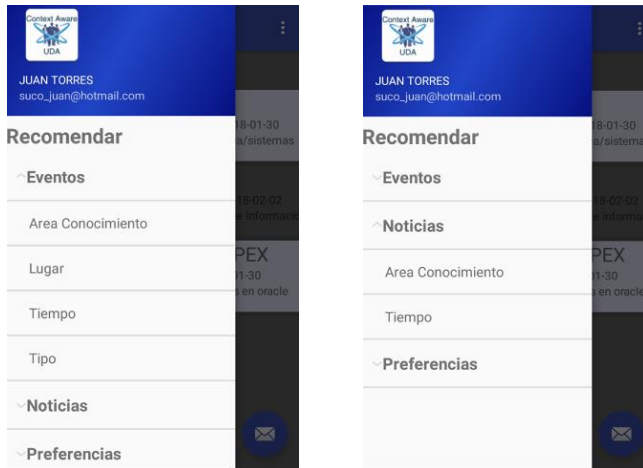


Fuente: Elaboración propia (2017)

Menú principal: el menú principal de la aplicación móvil contiene inicialmente el logo de la aplicación y el nombre del usuario junto con su correo, debajo de la información del usuario se presenta una lista con un pequeño menú con las opciones “Eventos”, “Noticias” y “Preferencias”. Al seleccionar “Eventos” o “Noticias”, se despliega una lista que contiene las opciones que tiene el usuario para obtener sus recomendaciones, tal como se muestra en la Figura 47. Al seleccionar “Preferencias” la aplicación móvil presenta la pantalla de configuración de preferencias, los detalles de esta pantalla se explican posteriormente.

Figura 47  
Menú de la aplicación





Fuente: Elaboración propia (2017)

Tipo de Eventos: al seleccionar la opción “Tipo” que se encuentra dentro del menú eventos, la aplicación presentará otra pantalla que contiene una serie de botones con los tipos de eventos existentes como se puede apreciar en la Figura 48.

Figura 48  
Pantalla de tipos de Eventos

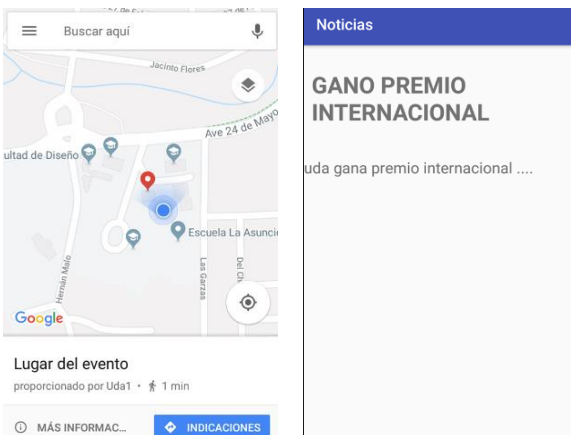


Fuente: Elaboración propia (2017)

En la pantalla principal el usuario puede seleccionar cualquier evento o noticia. Si selecciona un evento la aplicación abrirá Google Maps presentando al usuario su ubicación actual y el lugar del evento. Si selecciona una noticia la aplicación abre otra pantalla mostrando la descripción completa de la noticia.

En la Figura 49 se muestra la pantalla con el mapa del evento y la pantalla con la descripción completa de la noticia.

Figura 49  
*Pantalla de mapa de evento e información completa de la noticia*



Fuente: Elaboración propia (2017)

Y por último la pantalla de preferencias para que el usuario pueda editar el tiempo de actualización de las recomendaciones, el tiempo máximo desde cuando se desea ver los eventos o noticias (día, semana, mes) y la distancia máxima de búsqueda de las recomendaciones. Todo esto con la siguiente interfaz.

Figura 50  
Pantalla de preferencias



Fuente: Elaboración propia (2017)

## CAPITULO 6

### 6. IMPLEMENTACIÓN Y PRUEBAS DEL SOFTWARE

#### 6.1 Configuración servidores

##### 6.1.1 Configuración de razonador y FUSEKI

A continuación se detallan algunos de los pasos para habilitar el razonador dentro de Apache FUSEKI. En primer lugar se genera un archivo ttl que se publicará en

#### 1. Establecer prefijos

Se crean prefijos para los *namespaces* usados.

```
@prefix :           <#> .
@prefix fuseki:    <http://jena.apache.org/fuseki#> .
@prefix rdf:       <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:      <http://www.w3.org/2000/01/rdf-schema#> .
@prefix tdb:       <http://jena.hpl.hp.com/2008/tdb#> .
@prefix ja:        <http://jena.hpl.hp.com/2005/11/Assembler#> .
```

Los prefijos describen distintos componentes:

- FUSEKI.
- RDF: Resource description framework.
- RDFS: RDF schema.
- TDB: Base de datos no transaccional de alto desempeño que almacena datos RDF.
- JA: Ensambladores. Son mecanismos de Jena para describir objetos a ser construidos (modelos, datasets, etc.).

#### 2. Definir la sección de servidor

Se define el servidor de FUSEKI como un tipo de RDF y se nombra el servicio a utilizar.

```
[] rdf:type fuseki:Server ;
fuseki:services ( <#service1> ) .
```



### 3. Cargar clases de TDB

Se carga la clase requerida para la creación del TDB. La inicialización del TDB ocurrirá de forma automática y el sistema del ensamblador revisará dentro del archivo de configuración todas las sentencias *ja:loadClass* antes de intentar estructurar cualquier objeto (implica que la carga de clases se realizará de forma independiente a su ubicación en el archivo).

```
[ ] ja:loadClass "com.hp.hp1.jena.tdb.TDB" .
```

### 4. Configurar el servicio de FUSEKI

A continuación se procede a configurar el servicio de FUSEKI detallado anteriormente. En este punto se debe colocar el nombre del servicio principal y los *endpoints* de los servicios de consulta, actualización, etc.

```
<#service1> rdf:type fuseki:Service ;  
fuseki:name "inf" ; # http://localhost/inf  
fuseki:serviceQuery "sparql" ; # servicio de consulta SPARQL  
fuseki:serviceUpdate "update" ; # servicio de actualización  
fuseki:serviceUpload "upload" ; # servicio de carga de datos  
fuseki:serviceReadWriteGraphStore "data" ;  
fuseki:serviceReadGraphStore "get"  
fuseki:dataset <#dataset> ; .
```

Las líneas *fuseki:serviceReadWriteGraphStore* y *fuseki:serviceReadGraphStore*, definen los endpoints del protocolo de almacenamiento de gráficos de SPARQL. Este esquema se usa para la consulta y actualización distribuida del contenido de gráficos RDF usando HTTP. En el primer caso se habilita el modo de lectura-escritura, en tanto que el segundo sólo habilita la lectura.

La línea *fuseki:dataset <#dataset>* indica cuál será el dataset asociado al servicio

### 5. Definir el dataset

Se indica al ensamblador que debe crear un `RDFDataset` y se lo asocia al descrito en el punto anterior.

```
<#dataset> rdf:type ja:RDFDataset ;
```

### 6. Definir el gráfico por defecto

Se agrega la sentencia para que el ensamblador genere un gráfico por defecto y se adjunta su nombre.

```
ja:defaultGraph <#model_inf> ;
```

## 7. Configurar el modelo de inferencia

Lo siguiente es configurar el modelo de inferencia, para lo cual:

1. Se define el modelo de inferencia.

```
<#model_inf> a ja:InfModel ;
```

2. Se establece tipo de modelo como *baseModel* y se lo asocia a un gráfico TDB.

```
ja:baseModel <#tdbGraph> ;
```

3. Configuramos el ensamblador para crear una instancia del razonador proporcionándole la URL.

```
ja:reasoner [ ja:reasonerURL  
<http://jena.hpl.hp.com/2003/OWLFBRuleReasoner> ] .
```

4. Configurar data sets del modelo de inferencia

Finalmente se configuran los data sets asociados al modelo de inferencia. En primer lugar se define un data set del tipo *DatasetTDB* y su ubicación.

```
<#tdbDataset> rdf:type tdb:DatasetTDB ;  
tdb:location "DB" ;
```

Finalmente se define un data set gráfico y se lo asociamos al data set anterior.

```
<#tdbGraph> rdf:type tdb:GraphTDB ;  
tdb:dataset <#tdbDataset> .
```

## 6.2 Servicios de consulta de información en la red ontológica

### 6.2.1 Descripción procedimental de los servicios web para consulta de información

Para obtener la información se codificaron varios servicios web tipo REST en el servidor Glassfish y se utilizaron varias librerías para la comunicación con el servidor FUSEKI (contenedor de la ontología). La aplicación móvil utiliza estos servicios web de tipo REST para obtener las recomendaciones que brinda la ontología, las recomendaciones se envían a la aplicación móvil en un formato JSON. Los servicios web son similares para todas las opciones de consulta que se presentan al usuario en la aplicación móvil, por lo que se revisará el procedimiento empleado únicamente para la consulta de eventos por lugar:

1. Para recomendar los eventos según la localización del usuario se obtiene la latitud y longitud desde el sensor GPS del teléfono inteligente en el momento de la petición y se realiza la consulta al servicio web REST que se encuentra en la dirección:

```
"FusekiOntoGEOSparql/webresources/wslugares?tipo=""&x=""&y=""&dis=""=&unid=""
```

La solicitud incluye los parámetros:

**tipo:** el identificador del usuario que se obtuvo cuando inicio sesión en la aplicación. Es representado en la ontología con una cadena de texto y un número. Ejemplo estudiante1 o profesor3 (estudiante1 tiene Nombre, apellido, preferencias de áreas de conocimiento, etc.)

**x:** la latitud que se obtiene del GPS.

**y:** la longitud que se obtiene del GPS.

**dis:** es la distancia máxima de búsqueda de recomendaciones. Este dato está almacenado en el dispositivo móvil en las preferencias aplicadas por el usuario.

**unid:** la unidad de medida de la distancia máxima de búsqueda (metros o kilómetros). Este dato está almacenado en las preferencias aplicadas por el usuario.

Ejemplo final de una llamada al servicio REST:

```
FusekiOntoGEOsparql/webresources/wslugares?tipo="estudiante1"&x="-2.7051"&y="-79.008"&dis="50"=&unid="Meters"
```

2. Cuando se llama al servicio REST el servidor Glassfish realiza los siguientes pasos:
  1. Llamar al método *eventos.conlugares* con los parámetros de latitud, longitud, distancia y unidad de medida.
  2. El método mediante consultas GEOSPARQL a la ontología obtiene los lugares cercanos.

```
PREFIX spatial: <http://jena.apache.org/spatial#>
SELECT * {
?place spatial:nearby ( latitud longitude distancia
"unidad")
}
```

En el ejemplo:

```
Select * {
?place spatial:nearby (-2.7051 -79.008 50 "Meters" )
}
```

3. La respuesta a la consulta de lugares cercanos se representa en una lista en el servidor Glassfish:

lugar2, lugar3, lugar4...

3. Una vez obtenido los lugares cercanos a esos puntos, Glassfish utilizará esos lugares para realizar una nueva consulta SPARQL a la ontología empleando como criterio el identificador del usuario (estudiante1) y los lugares cercanos. De la siguiente manera:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uda:
<http://www.uazuay.edu.ec/ontologies/ContextAwareEvent#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT ?x ?y ?t ?r ?f ?lat ?lon where {
?obj <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
uda:Student .
?obj <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
owl:NamedIndividual .
?x <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
uda:Event .
uda:estudiante1 uda:has_interest ?organization .
?x uda:included_of_KnowledgeField ?organization.
?x uda:located_in_place uda:lugar2 . ->(Aqui entran
los lugares que se
```

```
encontraron en el paso anterior)
  ?x  uda:eventTitle ?y .
  ?x  uda:eventDescription ?t .
  ?x  uda:inXSDdate ?r .
  ?x  uda:located_in_place ?g.
  ?g  uda:placeName ?f.
  ?g  uda:latitude ?lat.
  ?g  uda:longitude ?lon .}
```

4. Para la respuesta obtenida se realiza una simple concatenación de caracteres para que el servicio web pueda enviar la respuesta solicitada por la aplicación móvil en formato JSON. La concatenación de los caracteres tendrá el formato:

```
[{"titulo":"tituloEvento","descripcion":"descripcionEvento","fecha":"20/01/2018","lugar":"Auditorio","latitud":"-2.50","longitud":"-78.99"}]
```

Si no se encuentra información:

```
[{"titulo":"No hay recomendaciones",
"descripcion":"","fecha":"","lugar":"","latitud":"","longitud":""}]
```

5. El recomendador móvil utilizará ese formato para separar la información y mostrar al usuario las recomendaciones que la ontología le brinda en forma de lista y de manera entendible.

## 6.3 Evaluación del sistema recomendador móvil de eventos y noticias académicas

Las pruebas del sistema recomendador móvil de eventos y noticias académicas están compuestas de dos partes: pruebas de unidad, y pruebas de integración y sistema.

### 6.3.1 Pruebas de unidad

En las pruebas de unidad se analizó cada componente del sistema por separado, en este caso primero se realizaron las pruebas de la aplicación web y luego las pruebas de la aplicación móvil.

Las pruebas de unidad fueron realizadas por los propios programadores del sistema, es decir el código implementado por un programador fue validado por el otro y viceversa, con la

finalidad de encontrar los errores más comunes que suelen presentarse en la programación, tales como: ausencia de validaciones, inconsistencias en los datos, mensajes ambiguos o incorrectos, etc.

### 6.3.2 Pruebas de integración

Las pruebas de integración tuvieron como finalidad validar que los componentes del sistema en su conjunto funcionen correctamente. Puntualmente, se verificó que todos los servicios web publicados en el servidor Glassfish sean consumidos de manera correcta por la aplicación móvil, en cuanto a los parámetros de entrada que utilizan y a la información resultante que devuelven.

En lo que respecta a las pruebas del sistema se comprobó que las aplicaciones web y móvil cumplan con todos los requerimientos funcionales descritos en el capítulo 3: Análisis del Software.

Adicionalmente, se realizaron pruebas de cumplimiento de los requerimientos no funcionales, por ejemplo se verificó la eficiencia de la aplicación móvil con relación al tiempo de respuesta al momento de obtener recomendaciones de eventos. A continuación, se presenta una muestra de las pruebas realizadas:



1. La Figura 51 presenta la ubicación del usuario desde donde se realiza la prueba, en donde el símbolo  representa la ubicación del usuario en el mapa y el símbolo  la ubicación del evento junto con su identificador.
2. La Tabla 31 está compuesta por las siguientes columnas: tiempo de respuesta o tiempo que tarda la aplicación en mostrar los resultados, el parámetro de distancia máxima de búsqueda configurado en la aplicación, la distancia real entre el usuario y el evento, y las recomendaciones obtenidas.

Figura 51  
Ubicación del usuario para realizar la prueba



Fuente: Google Maps (2018)

Tabla 31  
Resultados de las pruebas

	Parámetro de Distancia máxima de búsqueda en la aplicación (metros)	30	50	90	150
	Tiempo de respuesta.(segundos)	0.81s	0.81	0.83s	0.85s
<b>Id Eventos y distancia desde el usuario hasta el evento (metros)</b>	Evento1: 21	✓	✓	✓	✓
	Evento2: 58.9		✓	✓	✓
	Evento3: 42.76		✓	✓	✓
	Evento4: 59.8			✓	✓
	Evento5: 94.87				✓
	Evento6: 49.66		✓	✓	✓

Fuente: Elaboración Propia (2018)

## CAPITULO 7

### 7. CONCLUSIONES Y TRABAJO FUTURO

A través del presente trabajo de investigación se ha podido realizar un estado del arte sobre la computación sensible al contexto y los enfoques de representación del contexto, sobre todo acerca de los modelos ontológicos existentes en la literatura, describiendo sus principales características, y analizando las ventajas y desventajas de cada uno de ellos.

Se concluyó que la mayoría de propuestas ontológicas existentes para representar el contexto son construidas mediante una estrategia top-down, conformada por una capa superior con un nivel de abstracción alto que incluye conceptos genéricos independientes de cualquier dominio, y una capa inferior que incluye conceptos particulares a un dominio específico (entorno académico, oficinas, hogar, ciudades, etc.).

Las metodologías NeOn debido a sus procesos flexibles que fomentan la reutilización de los recursos ontológicos y no ontológicos existentes, facilitó y minimizó el esfuerzo requerido en la construcción de la red de ontologías para representar la información de los eventos y noticias académicas, así como de la información contextual.

RDF, RDF Schema y OWL son lenguajes que permiten construir ontologías, de tal forma que puedan ser procesadas por computadores, además de ser lenguajes de fácil entendimiento e implementación para los desarrolladores.

Los servidores Apache Jena y Apache Fuseki son componentes que trabajan conjuntamente para facilitar la programación e implementación de aplicaciones java que requieren almacenar y gestionar ontologías empleando consultas SPARQL.

El estándar IEEE-830 para la especificación de requerimientos de software en conjunto con los diagramas de casos de uso, clases y secuencias han permitido de manera clara, precisa y sin ambigüedades documentar las fases de análisis y diseño del software.

La adopción de una arquitectura modelo – vista – controlador para el recomendador móvil ha permitido construir un sistema altamente desacoplado, que facilita la reutilización del código, el mantenimiento, la interoperabilidad y la escalabilidad del sistema.



Para el diseño de las diferentes interfaces de usuario del sistema se utilizaron mockups, lo que ayudó a tener una vista previa de lo que sería el resultado final de las pantallas del sistema, además de facilitar los cambios sugeridos por los usuarios antes de iniciar con la programación.

Con la implementación y pruebas del sistema recomendador se pudo validar y confirmar lo señalado por varios autores en la literatura científica, acerca de que los enfoques de modelado ontológico son los más adecuados para representar el contexto, ya que brindan mayor formalidad, semántica y escalabilidad.

El resultado final del presente trabajo es una aplicación móvil que, a partir del contexto, configuraciones y preferencias de los usuarios realiza recomendaciones de eventos y noticias existentes dentro de la universidad.

Como trabajos futuros se tiene previsto crear una aplicación móvil para sugerir de forma inteligente recursos didácticos tanto físicos como digitales, teniendo en consideración la ubicación, tiempo, perfil del usuario y la programación académica de las asignaturas. Asimismo, se pueden implementar recomendadores móviles sensibles al contexto en otros dominios, tales como: anuncios publicitarios, turismo, entretenimiento, etc.

Finalmente, se deberá fortalecer el modelo ontológico construido, incluyendo nuevas estructuras conceptuales que permitan a través del razonador inferir nuevo conocimiento contextual que no esté explícitamente almacenado en la ontología mediante tripletas.

## BIBLIOGRAFÍA

- Bikakis, N., Tsinarak, C., Gioldasis, N., Stavrakantonakis, I., & Christodoulakis, S. (2013). The XML and Semantic Web Worlds: Technologies, Interoperability and Integration: A Survey of the State of the Art. En I. Anagnostopoulos, M. Bieliková, P. Mylonas, & N. Tsapatsoulis, *Semantic Hyper/Multimedia Adaptation* (págs. 319-360). New York: Springer-Verlag Berlin Heidelberg.
- Nalepa, G. J., & Bobek, S. (2014). Rule-based solution for context-aware reasoning on mobile devices. *Computer Science and Information Systems*, 171-193.
- Andrade Pinto, H. S., Staab, S., & Tempich, C. (2004). DILIGENT: Towards a fine-grained methodology for Distributed, Loosely-controlled and evolving Engineering of ontologies.
- Antoniou, G., & van Harmelen, F. (2004). *A Semantic Web Primer*. Cambridge, Massachusetts.
- Apache. (2011–2017). *Apache Jena*. Recuperado el 21 de Mayo de 2017, de <http://jena.apache.org/documentation/ontology/>
- Apache. (2017). *Apache*. Recuperado el 26 de Mayo de 2017, de <https://marmotta.apache.org/>
- Asensio, E. S. (2008). *A MDD strategy for developing context-aware pervasive systems*. Valencia.
- Barkhuus, L., & Dey, A. (2003). *Is Context-Aware computing taking control away from the user? three levels of interactivity examined*.
- Brickley, D., & Miller, L. (2014). *FOAF-Project*. Obtenido de <http://www.foaf-project.org/>
- Cadenas, A., Ruis, C., Larizgoitia, I., García-Castro, R., Lamsfus, C., Vázquez, I., . . . Poveda, M. (2009). *Context Management in Mobile Environments: a Semantic Approach*. Heraklion, Greece.
- Cano, J.-C., Calafate, C. T., Malumbres, M. P., & Manzoni, P. (2008). *Redes Inalambricas Ad Hoc como Tecnología de Soporte para la Computacion Ubicua*. Valencia.

- Cárdenas Ruíz, H. D. (2017). *Análisis, diseño y construcción de una ontología para usuario, integrada a una aplicación móvil que responda a intereses del usuario*. Quito.
- CCM. (16 de Octubre de 2008). *CCM - Comunidad Informática*. Recuperado el 26 de Abril de 2017, de Estándar GSM (Sistema global de comunicaciones móviles): <http://es.ccm.net/contents/681-estandar-gsm-sistema-global-de-comunicaciones-moviles>
- CES, C. d. (2016). *Reglamento de Armonización da la Nomenclatura de Títulos Profesionales y Académicos que Confieren las Instituciones de Educación Superior del Ecuador Codificado*. Quito.
- Chen, H., Finin, T., & Joshi, A. (2003). An Ontology for Context Aware Pervasive Computing Environments. *The Knowledge Engineering Review*, 197-207.
- Chen, H., Perich, F., Finin, T., & A., J. (2004). SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In *Mobile and Ubiquitous Systems: Networking and Services. MOBIQUITOUS*, 258-276.
- Dey, A. (2000). Providing architectural support for building context-aware applications. Tesis Doctoral. *Georgia Institute of Technology*.
- Erazo, L., Illescas, L., & Espinoza, M. (2016). Una ontología para representar el contexto en sistemas de computación académicos universitarios. *Revista Tecnológica ESPOL – RTE, Vol. 29, N. 2*, 124-152.
- Estimote. (11 de Junio de 2017). *About Estimote*. Obtenido de estimote: <https://estimote.com/about/>
- Grobe, M. (2009). *RDF, Jena, SparQL and the “Semantic Web*. Indianapolis, Indiana.
- Grubber, T. R. (1995). Toward principles for the design of ontologies used for knowledge-sharing. *International Journal of Human and computer studies*, (págs. 907-928).

- Gruninger, M., & Fox, M. S. (1995). Methodology for the design and evaluation of ontologies. *In International Joint Conference on Artificial Intelligence (IJCAI95), Workshop on Basic Ontological Issues in Knowledge Sharing.*
- Guzmán Luna, J. A., López Bonilla, M., & Durley Torres, I. (2012). Metodologías y métodos para la construcción de ontologías.
- Hightower, J., & Borriello, G. (2001). *A Survey and Taxonomy of Location Systems for Ubiquitous Computing.* Seattle: University of Washington.
- Hobbs, J., & Pan, F. (2017). W3C. Obtenido de <https://www.w3.org/TR/owl-time/>
- Jamgade, A., & Karale, S. (2015). Ontology Based Approach for Semantic Information Retrieval System. *INTERNATIONAL JOURNAL FOR TRENDS IN ENGINEERING & TECHNOLOGY.*
- Klein, M. (2001). XML, RDF, and Relatives. En *IEEE INTELLIGENT SYSTEMS* (págs. 26-28).
- Klyne, G., Reynolds, F., Woodrow, C., Ohto, H., Hjelm, J., Butler, M., & Tran, L. (2004). W3C. Obtenido de <https://www.w3.org/TR/CCPP-struct-vocab/#CCPPArchitecture>
- Krumm, J. (2010). *Ubiquitous Computing Fundamentals.* Florida: CRC Press.
- Kuutti, A., Dvoryanchikova, A., Lobov, A., Martinez Lastra, J. L., & Vahtera, T. (2012). *A Device Configuration Management Tool for Context-Aware System. THT-Control Oy.*
- Martins, H., & Silva, N. (2012). Characterization, Comparison and Systematization of Context Ontologies. *Sixth International Conference on Complex, Intelligent, and Software Intensive Systems.*
- Moore, P., Hu, B. Z., Campbell, W., & Ratcliffe, M. (2007). A Survey of Context Modeling for Pervasive Cooperative Learning. *Information Technologies and Applications in Education 2007. ISITAE '07, k5-1 - k5-6.*

- Morse, D., Dey, A., & Armstrong, S. (2000). The What, Who, Where, When, and How of Context. *Conference on Human Factors in Computing*. The Hague.
- OGC. (2011). *GeoSPARQL - A geographic query language for RDF data*.
- Oxford. (s.f.). *Español Oxford Living Dictionaries*. Obtenido de <https://es.oxforddictionaries.com/>
- Ozgun Yurur, M. I. (2013). *Context-Awareness for Mobile Sensing: A Survey*.
- Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). *Context Aware Computing for the Internet of Things: A Survey*.
- Poslad, S. (2006). *Ubiquitous Computing: Smart Devices, Environments and Interactions*. *Jhon Wiley & Sons*. Londres.
- Poveda Villalón, M. (2010). *Metodología neón aplicada a la representación del contexto*. Madrid.
- Schilit, W. N., Adams, N. I., & Want, R. (1994). Context-aware Computing Applications. *In Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications* (págs. 85-90). Santa Cruz, CA, USA: IEEE Computer Society.
- Staab, S., & Studer, R. (2013). *Handbook on Ontologies*. New York: Springer Dordrecht Heidelberg .
- Stefan, D., Melnik, S., Fensel, D., Klein, M., Frank, V., Broekstra, J., . . . Horrocks, I. (2011). The semantic web: Roles of Xml and RDF. *IEEE INTERNET COMPUTING*, 63-74.
- Strang, T., & Linnhoff Popien, C. (2004). A context modeling survey. *In 1st International Workshop on Advanced Context Modelling, Reasoning and Management. UbiCom 2004*, 34-41.
- Strang, T., Linnhoff-Popien, C., & Frank, K. (2003). *CoOL: A Context Ontology Language to Enable Contextual Interoperability*. Munich, Alemania.

- Styles, R., & Shabir, N. (2008). *Academic Institution Internal Structure Ontology (AIISO)*.  
Obtenido de <http://vocab.org/aiiso/>
- Suarez Figueroa, M. d. (2010). *NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse*. Madrid.
- W3C. (2005). *W3C*. Obtenido de <https://www.w3.org/2003/01/geo/#example>
- Wang, X., Zhang, D., Gu, T., & Pung, H. (2004). Ontology Based Context Modeling and Reasoning using OWL. In *Pervasive Computing and Communications Workshops. IEEE*, 18-22.
- Weiser, M. (1991). *The Computer for the 21st Century*.
- Weiser, M., & Brown, J. S. (1997). The coming age of calm technology. In *Beyond Calculation: the Next Fifty Years*, P. J. Denning and R. M. Metcalfe, 75-85.
- Xu, N., Zhang, W., Yang, H., Zhang, X., & Xing, X. (2013). CACOnt: A Ontology-Based Model for Context Modeling and Reasoning.

## ANEXOS

### Anexo 1. Casos de uso específicos de aplicación web.

Tabla 32

*Caso de uso de gestionar facultades*

<b>Identificador</b>	CU-02	
<b>Nombre</b>	Gestionar Facultades	
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar facultades en la base de datos. Los campos que se pueden modificar son el nombre, el lugar al que pertenecen y la institución a la que pertenecen	
<b>Precondición</b>	El lugar en el que se encuentra la facultad debe ser previamente ingresado en la base de datos. La institución a la que pertenece debe ser previamente ingresada en la base de datos.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción “Facultades” de la pantalla de mantenimientos. Una vez seleccionado se puede realizar las siguientes acciones.
<b>Secuencia alternativa de Ingresar Facultad</b>	2	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa de Modificar Facultad</b>	2	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa de Eliminar Facultad</b>	2	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa de Consultar Facultad</b>	2	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>	En caso de que no se considere importante la información del lugar en el que se encuentra la facultad solamente se debe seleccionar la opción NINGUNO en el campo de lugar.	

Fuente: Elaboración propia (2017)

Tabla 33

*Caso de uso de gestionar escuelas*

<b>Identificador</b>	CU-03
<b>Nombre</b>	Gestionar Escuelas.

<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar escuelas en la base de datos. Los campos que se pueden modificar son el nombre, la facultad a la que pertenecen y el lugar en el que se encuentra.	
<b>Precondición</b>	El lugar en el que se encuentra la escuela debe ser previamente ingresado en la base de datos. La facultad a la que pertenece debe ser previamente ingresada en la base de datos.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción "Escuelas" de la pantalla de mantenimientos. Una vez seleccionado se puede realizar las siguientes acciones.
<b>Secuencia alternativa Ingresar Escuela</b>	2	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Escuela</b>	2	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Escuela</b>	2	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Escuela</b>	2	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>	En caso de que no se considere importante la información del lugar en el que se encuentra la escuela solamente se debe seleccionar la opción NINGUNO en el campo de lugar.	

Fuente: Elaboración propia (2017)

Tabla 34  
*Caso de uso de gestionar departamentos*

<b>Identificador</b>	CU-04	
<b>Nombre</b>	Gestionar Departamentos	
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar departamentos en la base de datos. Los campos que se pueden modificar son el nombre, la facultad a la que pertenece y el lugar al que pertenece	
<b>Precondición</b>	El lugar en el que se encuentra el departamento debe ser previamente ingresado en la base de datos. La facultad a la que pertenece debe ser previamente ingresada en la base de datos.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>



	1	El Administrador selecciona la opción “Departamentos” de la pantalla de mantenimientos. Una vez seleccionado se puede realizar las siguientes acciones.
<b>Secuencia alternativa Ingresar Departamento</b>	2	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Departamento</b>	2	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Departamento</b>	2	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Departamento</b>	2	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>	En caso de que no se considere importante la información del lugar en el que se encuentra el departamento solamente se debe seleccionar la opción NINGUNO en el campo de lugar.	

Fuente: Elaboración propia (2017)

Tabla 35  
*Caso de uso de gestionar lugares*

<b>Identificador</b>	CU-05	
<b>Nombre</b>	Gestionar Lugares	
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar lugares en la base de datos.	
<b>Precondición</b>	Primero ingresar los lugares contenedores y luego los lugares que se encuentran dentro de los contenedores.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción “Lugar” de la pantalla de mantenimientos. Una vez seleccionado se puede realizar las siguientes acciones.
<b>Secuencia alternativa Ingresar Lugar</b>	2	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Lugar</b>	2	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Lugar</b>	2	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Lugar</b>	2	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	

<b>Observaciones</b>	En caso de que no se considere importante la información del lugar contenedor en el que se encuentra el lugar solamente se debe seleccionar la opción NINGUNO en el campo de lugar.
----------------------	---

Fuente: Elaboración propia (2017)

Tabla 36

*Caso de uso de gestionar personas*

<b>Identificador</b>	CU-06	
<b>Nombre</b>	Gestionar Personas	
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar personas en la base de datos. Los campos que se pueden modificar son el nombre, el apellido, el correo electrónico, el número de teléfono y el título profesional.	
<b>Precondición</b>	Las personas se pueden ingresar en cualquier momento.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción “Personas” de la pantalla de mantenimientos. Una vez seleccionado se puede realizar las siguientes acciones.
<b>Secuencia alternativa Ingresar Persona</b>	2	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Persona</b>	2	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Persona</b>	2	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Persona</b>	2	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 37

*Caso de uso de gestionar asignaturas*

<b>Identificador</b>	CU-07
<b>Nombre</b>	Gestionar Asignaturas
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar asignaturas en la base de datos. Los campos que se pueden modificar son el nombre, la descripción y la escuela a la que pertenecen.

<b>Precondición</b>	La escuela a la que pertenece debe ser previamente ingresada en la base de datos.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción “Asignatura” de la pantalla de mantenimientos. Una vez seleccionado se puede realizar las siguientes acciones.
<b>Secuencia alternativa Ingresar Asignatura</b>	2	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Asignatura</b>	2	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Asignatura</b>	2	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Asignatura</b>	2	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 38

*Caso de uso de gestionar cursos*

<b>Identificador</b>	CU-08	
<b>Nombre</b>	Gestionar Cursos	
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar cursos en la base de datos, por ejemplo primero A, segundo B, etc. lo campos que pueden modificarse son el nombre y la escuela a la que pertenece.	
<b>Precondición</b>	La escuela a la que pertenece debe ser ingresada previamente	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción “Curso” de la pantalla de mantenimientos. Una vez seleccionado se puede realizar las siguientes acciones.
<b>Secuencia alternativa Ingresar Curso</b>	2	Ejecutar el caso de uso GE-01.

<b>Secuencia alternativa Modificar Curso</b>	2	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Curso</b>	2	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Curso</b>	2	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 39

*Caso de uso de gestionar distributivos académicos*

<b>Identificador</b>	CU-09	
<b>Nombre</b>	Gestionar Distributivos Académicos	
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar registros del distributivo académico en la base de datos. Los campos que se pueden modificar son la asignatura, el curso y el profesor designado.	
<b>Precondición</b>	Deben estar ingresados previamente los usuarios en este caso de tipo docente. Deben estar ingresados previamente las asignaturas. Deben estar ingresados previamente los cursos.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción "Distributivo Académico" de la pantalla de mantenimientos. Una vez seleccionado se puede realizar las siguientes acciones.
<b>Secuencia alternativa Ingresar Distributivo Académico</b>	2	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Distributivo Académico</b>	2	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Distributivo Académico</b>	2	Ejecutar el caso de uso GE-03.

<b>Secuencia alternativa Consultar Distributivo Académico</b>	2	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 40

*Caso de uso de gestionar escuelas*

<b>Identificador</b>	CU-10	
<b>Nombre</b>	Gestionar Estudiantes	
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar registros de estudiantes de la base de datos. Los campos que se pueden modificar son el campo asignatura-curso que es el que asigna la asignatura perteneciente a cierto curso al estudiante y el campo estudiante que es el que contiene los nombres del estudiante.	
<b>Precondición</b>	Deben estar ingresados previamente los usuarios en este caso de tipo estudiante. Deben estar ingresados previamente las asignaturas. Deben estar ingresados previamente los cursos.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción “Estudiantes” de la pantalla de mantenimientos. Una vez seleccionado se puede realizar las siguientes acciones.
<b>Secuencia alternativa Ingresar Estudiantes</b>	2	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Estudiantes</b>	2	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Estudiantes</b>	2	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Estudiantes</b>	2	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 41

*Caso de uso de gestionar campos amplios*

<b>Identificador</b>	CU-11	
<b>Nombre</b>	Gestionar Campos Amplios	
<b>Descripción</b>	<p>Ingresar, modificar o actualizar y eliminar los campos amplios en la base de datos. Los campos amplios están compuestos por campos específicos y estos a su vez por campos detallados los cuales sirven para definir los intereses de las personas, de las escuelas, eventos, noticias, etc.</p> <p>El campo que puede modificarse es el nombre del campo amplio.</p>	
<b>Precondición</b>	El campo amplio puede ser ingresado en cualquier momento.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción “Campo Amplio” de la pantalla de mantenimientos. Una vez seleccionado se puede realizar las siguientes acciones.
<b>Secuencia alternativa Ingresar Campo Amplio</b>	2	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Campo Amplio</b>	2	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Campo Amplio</b>	2	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Campo Amplio</b>	2	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 42

*Caso de uso de gestionar campos específicos*

<b>Identificador</b>	CU-12	
<b>Nombre</b>	Gestionar Campos Específicos	
<b>Descripción</b>	<p>Ingresar, modificar o actualizar y eliminar los campos específicos en la base de datos. Los campos específicos están compuestos por campos detallados los cuales sirven para definir los intereses de las personas, de las escuelas, eventos, noticias, etc.</p> <p>Los campos que se pueden modificar son el nombre y el campo amplio al que pertenece.</p>	
<b>Precondición</b>	Debe encontrarse ingresado el campo amplio al que pertenece.	

<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción “Campo Específico” de la pantalla de mantenimientos. Una vez seleccionado se puede realizar las siguientes acciones.
<b>Secuencia alternativa Ingresar Campo Específico</b>	2	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Campo Específico</b>	2	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Campo Específico</b>	2	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Campo Específico</b>	2	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 43

*Gestionar campos detallados*

<b>Identificador</b>	CU-13	
<b>Nombre</b>	Gestionar Campos Detallados	
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar los campos detallados en la base de datos. Los campos detallados sirven para definir los intereses de las personas, de las escuelas, eventos, noticias, etc. Los campos que se pueden modificar son el nombre y el campo específico al que pertenece.	
<b>Precondición</b>	Debe encontrarse ingresado el campo específico al que pertenece. Visualizar datos después de haber realizado la transacción.	
<b>Postcondición</b>	Transacción realizada con éxito.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción “Campo Detallado” de la pantalla de mantenimientos. Una vez seleccionado se puede realizar las siguientes acciones.
<b>Secuencia alternativa Ingresar Campo Detallado</b>	2	Ejecutar el caso de uso GE-01.

<b>Secuencia alternativa Modificar Campo Detallado</b>	2	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Campo Detallado</b>	2	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Campo Detallado</b>	2	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 44

*Gestionar tipos de usuarios*

<b>Identificador</b>	CU-14	
<b>Nombre</b>	Gestionar Tipos de usuarios	
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar los tipos de usuarios en la base de datos. Los tipos de usuario pueden ser docente, estudiante, etc. el campo que se puede modificar es el nombre del tipo de usuario.	
<b>Precondición</b>	El tipo de usuario se puede ingresar en cualquier momento.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción “Tipo de Usuario” de la pantalla de mantenimientos. Una vez seleccionado se puede realizar las siguientes acciones.
<b>Secuencia alternativa Ingresar Tipo de Usuario</b>	2	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Tipo de Usuario</b>	2	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Tipo de Usuario</b>	2	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Tipo de Usuario</b>	2	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	



<b>Urgencia</b>	Inmediatamente
<b>Observaciones</b>	

Fuente: Elaboración propia (2017)

Tabla 45

*Caso de uso de gestionar usuarios*

<b>Identificador</b>	CU-15	
<b>Nombre</b>	Gestionar Usuarios	
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar usuarios del sistema en la base de datos. Los campos que pueden modificarse son el nombre de usuario, la contraseña, la persona a la que se le asigna el usuario y el tipo de usuario.	
<b>Precondición</b>	Deben estar ingresados previamente los tipos de usuarios	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción “Usuarios” de la pantalla de mantenimientos. Una vez seleccionado se puede realizar las siguientes acciones.
<b>Secuencia alternativa Ingresar Usuario</b>	2	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Usuario</b>	2	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Usuario</b>	2	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Usuario</b>	2	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 46

*Caso de uso de gestionar eventos*

<b>Identificador</b>	CU-16
<b>Nombre</b>	Gestionar Eventos
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar eventos en la base de datos. Los campos que se pueden modificar son el título del evento, la descripción, la fecha, la hora, el tipo de evento y el lugar en el que se va a realizar.

<b>Precondición</b>	Deben estar ingresados previamente los tipos de Eventos. Deben estar ingresados previamente los lugares.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción “Eventos” de la pantalla de mantenimientos. Una vez seleccionado se puede realizar las siguientes acciones.
<b>Secuencia alternativa Ingresar Evento</b>	2	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Evento</b>	2	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Evento</b>	2	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Evento</b>	2	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 47  
*Caso de uso de gestionar tipos de evento*

<b>Identificador</b>	CU-17	
<b>Nombre</b>	Gestionar Tipos de Eventos	
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar tipos de eventos en la base de datos. Ejemplos de tipo de eventos son: simposios, debates, etc. los campos que se pueden modificar son el nombre y la descripción.	
<b>Precondición</b>	El tipo de evento se puede ingresar en cualquier momento.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción “Tipo de Eventos” de la pantalla de mantenimientos. Una vez seleccionado se puede realizar las siguientes acciones.
<b>Secuencia alternativa Ingresar Tipo de Evento</b>	2	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Tipo de Evento</b>	2	Ejecutar el caso de uso GE-02.

<b>Secuencia alternativa Eliminar Tipo de Evento</b>	2	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Tipo de Evento</b>	2	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 48  
*Caso de uso de gestionar noticias*

<b>Identificador</b>	CU-18	
<b>Nombre</b>	Gestionar Noticias	
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar noticias en la base de datos. Los campos que se pueden modificar son el título, la descripción, la fecha y la hora.	
<b>Precondición</b>	Las noticias se pueden ingresar en cualquier momento.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción "Noticias" de la pantalla de mantenimientos. Una vez seleccionado se puede realizar las siguientes acciones.
<b>Secuencia alternativa Ingresar Noticias</b>	2	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Noticias</b>	2	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Noticias</b>	2	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Noticias</b>	2	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 49

*Caso de uso de gestionar horarios*

<b>Identificador</b>	CU-19	
<b>Nombre</b>	Gestionar Horarios	
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar horarios en la base de datos. Los campos que se pueden modificar son el día, la hora de inicio, la hora de finalización, la asignatura-curso y el lugar.	
<b>Precondición</b>	Debe ingresarse previamente la información de la gestión del distributivo académico.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción “Distributivo Académico” de la pantalla de mantenimientos.
	2	Seleccionar el registro del distributivo académico al que se le va a asignar o cambiar horarios. Una vez seleccionado se puede realizar las siguientes acciones
<b>Secuencia alternativa Ingresar Horario</b>	3	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Horario</b>	3	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Horario</b>	3	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Horario</b>	3	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	

Fuente: Elaboración propia (2017)

Tabla 50

*Caso de uso de gestionar intereses de escuelas*

<b>Identificador</b>	CU-20	
<b>Nombre</b>	Gestionar Intereses de Escuelas	
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar intereses de escuelas en la base de datos. Los campos que se pueden modificar son la escuela y el campo detallado.	
<b>Precondición</b>	Debe ingresarse previamente la información de la gestión de escuelas.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>

	1	El Administrador selecciona la opción “Escuelas” de la pantalla de mantenimientos.
	2	El Administrador selecciona el registro de escuelas al que se le va a asignar o intereses. Una vez seleccionado se puede realizar las siguientes acciones
<b>Secuencia alternativa Ingresar Interés de escuelas</b>	3	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Interés de escuela</b>	3	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Interés de escuela</b>	3	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Interés de escuela</b>	3	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 51  
*Caso de uso de gestionar intereses de personas*

<b>Identificador</b>	CU-21	
<b>Nombre</b>	Gestionar Intereses de Personas	
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar intereses de personas en la base de datos. Los campos que se pueden modificar son la persona y el campo detallado.	
<b>Precondición</b>	Debe ingresarse previamente la información de la gestión de personas.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción “Personas” de la pantalla de mantenimientos.
	2	El Administrador selecciona el registro de personas al que se le va a asignar o cambiar intereses. Una vez seleccionado se puede realizar las siguientes acciones
<b>Secuencia alternativa Ingresar Interés de persona</b>	3	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Interés de persona</b>	3	Ejecutar el caso de uso GE-02.

<b>Secuencia alternativa Eliminar Interés de persona</b>	3	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Interés de persona</b>	3	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>		

Fuente: Elaboración propia (2017)

Tabla 52  
*Caso de uso de gestionar intereses de eventos*

<b>Identificador</b>	CU-22	
<b>Nombre</b>	Gestionar Intereses de Eventos	
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar intereses de eventos en la base de datos. Los campos que se pueden modificar son el evento y el campo detallado.	
<b>Precondición</b>	Debe ingresarse previamente la información de la gestión de eventos.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción “Eventos” de la pantalla de mantenimientos.
	2	El Administrador selecciona el registro de eventos al que se le va a asignar o cambiar horarios. Una vez seleccionado se puede realizar las siguientes acciones
<b>Secuencia alternativa Ingresar Interés de evento</b>	3	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Interés de evento</b>	3	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Interés de evento</b>	3	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Interés de evento</b>	3	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>		

Tabla 53

*Caso de uso de gestionar intereses de noticias*

<b>Identificador</b>	CU-23	
<b>Nombre</b>	Gestionar Intereses de Noticias	
<b>Descripción</b>	Ingresar, modificar o actualizar y eliminar intereses asociados a noticias en la base de datos. Los campos que se pueden modificar son la noticia y el campo detallado.	
<b>Precondición</b>	Debe ingresarse previamente la información de la gestión de noticias.	
<b>Postcondición</b>	Transacción realizada con éxito. Visualizar datos después de haber realizado la transacción.	
<b>Actores</b>	Administrador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Administrador selecciona la opción “Noticias” de la pantalla de mantenimientos.
	2	El Administrador selecciona el registro de noticias al que se le va a asignar o cambiar intereses. Una vez seleccionado se puede realizar las siguientes acciones
<b>Secuencia alternativa Ingresar Interés de noticia</b>	3	Ejecutar el caso de uso GE-01.
<b>Secuencia alternativa Modificar Interés de noticia</b>	3	Ejecutar el caso de uso GE-02.
<b>Secuencia alternativa Eliminar Interés de noticia</b>	3	Ejecutar el caso de uso GE-03.
<b>Secuencia alternativa Consultar Interés de noticia</b>	3	Ejecutar el caso de uso GE-04.
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Observaciones</b>		

Doctora Jenny Ríos Coello, Secretaria de la Facultad de Ciencias de la Administración de la Universidad del Azuay

**CERTIFICA:**

Que, el Consejo de Facultad en sesión del 29 de mayo de 2017, conoció la petición de los estudiantes **NICOLÁS OSWALDO ANDRADE HIDALGO** con código **63103** y **JUAN DIEGO TORRES ARGUDO** con código **64318**, que presentan el diseño de su trabajo de titulación denominado: "**RECOMENDADOR MÓVIL SENSIBLE AL CONTEXTO DE NOTICIAS Y EVENTOS UNIVERSITARIOS EMPLEANDO TECNOLOGÍAS SEMÁNTICAS**", presentado previa a la obtención del título de Ingenieros de Sistemas y Telemática.- El Consejo de Facultad acogió el informe de la Junta Académica de Ingeniería de Sistemas y Telemática y resolvió aprobar el diseño. Designa como **Director al ingeniero Lenin Erazo Garzón** y como miembros del Tribunal Examinador a los ingenieros **Francisco Salgado Arteaga, Ph.D.** y **Chester Sellers Walden**.- En esta misma sesión el Consejo de Facultad fija como plazo para la entrega del trabajo de titulación, seis meses contados desde la fecha de su aprobación, esto es hasta el **29 de noviembre de 2017**, debiendo el Director presentar a la Junta Académica, dos informes bimensuales del desarrollo del trabajo de titulación.

Cuenca, mayo 30 de 2017



Dra. Jenny Ríos Coello  
Secretaria de la Facultad de  
Ciencias de la Administración



## CONVOCATORIA

Por disposición de la Junta Académica de Ingeniería de Sistemas y Telemática, se convoca a los Miembros del Tribunal Examinador, a la sustentación del Protocolo del Trabajo de Titulación: "RECOMENDADOR MÓVIL SENSIBLE AL CONTEXTO DE NOTICIAS Y EVENTOS UNIVERSITARIOS EMPLEANDO TECNOLOGÍAS SEMÁNTICAS", presentado por los estudiantes Nicolás Oswaldo Andrade Hidalgo y Juan Diego Torres Argudo, previa a la obtención del grado de Ingenieros en Sistemas y Telemática, para el día LUNES 15 DE MAYO DE 2017 A LAS 08h20. La sustentación se realizará en el laboratorio del IERSE.

Cuenca, 11 de mayo de 2017

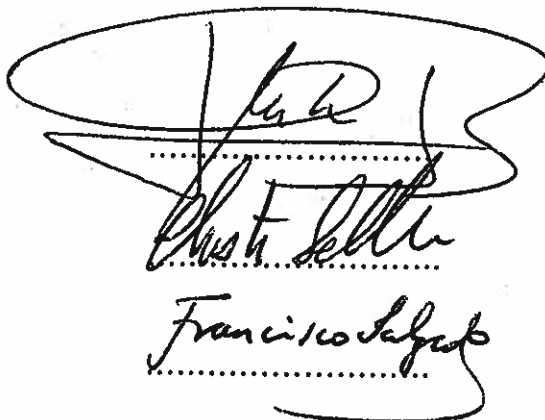


Dra. Jenny Ríos Coello  
Secretaria de la Facultad

Ing. Lenin Erazo Garzón ✓

✱ Ing. Chester Sellar Walden ✓

Dr. Francisco Salgado Arteaga



mjmr/

Comunicado K.  
12/05/2017

Oficio Nro. 063-2017-DIST-UDA

Cuenca, 12 de mayo de 2017

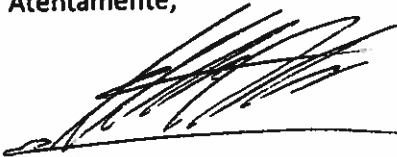
**Señor Ingeniero  
Oswaldo Merchán Manzano  
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN  
Presente.-**

De nuestras consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 12 de mayo del 2017, recibió el proyecto de tesis titulado "Recomendador móvil sensible al contexto de noticias y eventos universitarios empleando tecnologías semánticas", presentado por Nicolás Oswaldo Andrade Hidalgo y Juan Diego Torres Argudo estudiantes de la Escuela de Ingeniería de Sistemas y Telemática, y revisado por el Ing. Lenin Erazo, previo a la obtención del título de Ingeniero de Sistemas y Telemática.

Por lo expuesto, y de conformidad con el Reglamento de Graduación de la Facultad, recomendamos como director y responsable de aplicar cualquier modificación al diseño del trabajo de graduación posterior al Ing. Lenin Erazo y como miembros del Tribunal a Francisco Salgado Ph.D. e Ing. Chester Sellers.

Atentamente,



**Ing. Marcos Orellana Cordero  
Cordinador Escuela de Ingeniería de Sistemas y Telemática  
Universidad del Azuay**



ACTA

SUSTENTACIÓN DE PROTOCOLO/DENUNCIA DEL TRABAJO DE TITULACIÓN

- 1.1 Nombre del estudiante: **Nicolás Oswaldo Andrade Hidalgo y Juan Diego Torres Argudo**
- 1.2 Director sugerido: **Ing. Lenin Erazo Garzón**
- 1.3 Codirector (opcional): \_\_\_\_\_
- 1.4 Tribunal: **Ing. Chester Sellar Walden/ Dr. Francisco Salgado Arteaga**
- 1.5 Título propuesto: **“RECOMENDADOR MÓVIL SENSIBLE AL CONTEXTO DE NOTICIAS Y EVENTOS UNIVERSITARIOS EMPLEANDO TECNOLOGÍAS SEMÁNTICAS”**
- 1.6 Resolución:

1.6.1 Aceptado sin modificaciones ✓

1.6.2 Aceptado con las siguientes modificaciones:

---



---



---

1.6.3 Responsable de dar seguimiento a las modificaciones: **Ing. Lenin Erazo Garzón**

1.6.4 No aceptado

• Justificación:

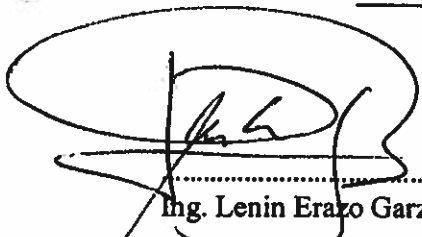
---



---

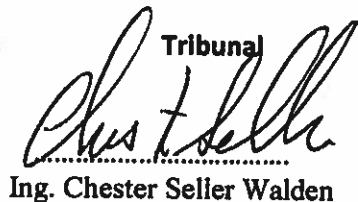


---

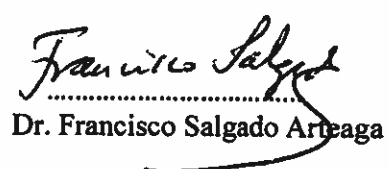


.....  
Ing. Lenin Erazo Garzón

Tribunal



.....  
Ing. Chester Sellar Walden



.....  
Dr. Francisco Salgado Arteaga



.....  
Sr. Nicolás Oswaldo Andrade Hidalgo



.....  
Sr. Juan Diego Torres Argudo



.....  
Dra. Jenny Ríos Coello  
Secretario de Facultad

Fecha de sustentación: día **LUNES 15 DE MAYO DE 2017 A LAS 08h20**



**RÚBRICA PARA LA EVALUACIÓN DEL PROTOCOLO DE TRABAJO DE TITULACIÓN**

**1.1 Nombre del estudiante:** Nicolás Oswaldo Andrade Hidalgo y Juan Diego Torres Argudo

**1.2 Director sugerido:** Ing. Lenin Erazo Garzón

**1.3 Codirector (opcional):**

**1.4 Título propuesto:** "RECOMENDADOR MÓVIL SENSIBLE AL CONTEXTO DE NOTICIAS Y EVENTOS UNIVERSITARIOS EMPLEANDO TECNOLOGÍAS SEMÁNTICAS"

**1.5 Revisores (tribunal):** Ing. Chester Seller Walden/ Dr. Francisco Salgado Arteaga

**1.6 Recomendaciones generales de la revisión:**

	Cumple totalmente	Cumple parcialmente	No cumple	Observaciones (*)
<b>Línea de investigación</b>				
1. ¿El contenido se enmarca en la línea de investigación seleccionada?	✓			
<b>Título Propuesto</b>				
2. ¿Es informativo?	✓			
3. ¿Es conciso?	✓			
<b>Estado del arte</b>				
4. ¿Identifica claramente el contexto histórico, científico, global y regional del tema del trabajo?	✓			
5. ¿Describe la teoría en la que se enmarca el trabajo	✓			
6. ¿Describe los trabajos relacionados más relevantes?	✓			
7. ¿Utiliza citas bibliográficas?	✓			
<b>Problemática y/o pregunta de investigación</b>				
8. ¿Presenta una descripción precisa y clara?	✓			
9. ¿Tiene relevancia profesional y social?	✓			
<b>Hipótesis (opcional)</b>				
10. ¿Se expresa de forma clara?	✓			
11. ¿Es factible de verificación?	✓			
<b>Objetivo general</b>				
12. ¿Concuerda con el problema formulado?	✓			
13. ¿Se encuentra redactado en tiempo	✓			



verbal infinitivo?				
<b>Objetivos específicos</b>				
14.¿Concuerdan con el objetivo general?	/			
15.¿Son comprobables cualitativa o cuantitativamente?	/			
<b>Metodología</b>				
16.¿Se encuentran disponibles los datos y materiales mencionados?	/			
17.¿Las actividades se presentan siguiendo una secuencia lógica?	/			
18.¿Las actividades permitirán la consecución de los objetivos específicos planteados?	/			
19.¿Los datos, materiales y actividades mencionadas son adecuados para resolver el problema formulado?	/			
<b>Resultados esperados</b>				
20.¿Son relevantes para resolver o contribuir con el problema formulado?	/			
21.¿Concuerdan con los objetivos específicos?	/			
22.¿Se detalla la forma de presentación de los resultados?	/			
23.¿Los resultados esperados son consecuencia, en todos los casos, de las actividades mencionadas?	/			
<b>Supuestos y riesgos</b>				
24.¿Se mencionan los supuestos y riesgos más relevantes?	/			
25.¿Es conveniente llevar a cabo el trabajo dado los supuestos y riesgos mencionados?	/			
<b>Presupuesto</b>				
26.¿El presupuesto es razonable?	/			
27.¿Se consideran los rubros más relevantes?	/			
<b>Cronograma</b>				
28.¿Los plazos para las actividades son realistas?	/			
<b>Referencias</b>				
29.¿Se siguen las recomendaciones de normas internacionales para citar?	/			
<b>Expresión escrita</b>				
30.¿La redacción es clara y fácilmente comprensible?	/			
31.¿El texto se encuentra libre de faltas ortográficas?	/			



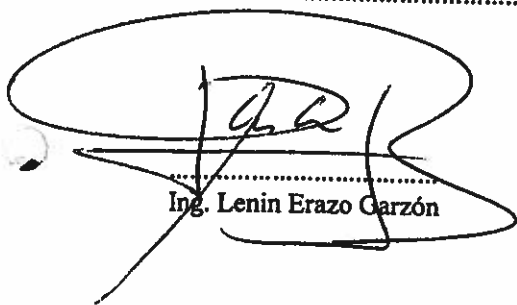
(\*) Breve justificación, explicación o recomendación.

- Opcional cuando cumple totalmente,
- Obligatorio cuando cumple parcialmente y NO cumple.

.....

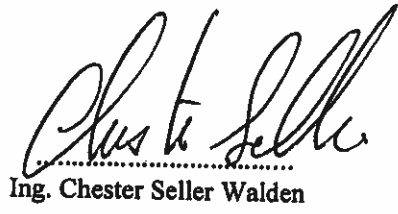
.....

.....



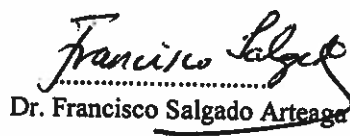
.....

Ing. Lenin Erazo Garzón



.....

Ing. Chester Seller Walden



.....

Dr. Francisco Salgado Arteaga



UNIVERSIDAD DEL  
AZUAY

DOCTORA JENNY RIOS COELLO, SECRETARIA DE LA FACULTAD DE  
CIENCIAS DE LA ADMINISTRACION DE LA UNIVERSIDAD DEL AZUAY

**CERTIFICA:**

Que, el Señor **ANDRADE HIDALGO NICOLAS OSWALDO**, con código **63103**,  
alumno de la Escuela de **INGENIERIA DE SISTEMAS Y TELEMATICA**, tiene  
aprobado el **91,11%** de créditos de su **matrícula curricular**.

Cuenca, 27 de marzo de 2017

SECRETARIA  
FACULTAD DE  
CIENCIAS DE LA ADMINISTRACION

Derecho No. 001-010-000116264

mjmr.-

DOCTORA JENNY RIOS COELLO, SECRETARIA DE LA FACULTAD DE  
CIENCIAS DE LA ADMINISTRACION DE LA UNIVERSIDAD DEL AZUAY

**CERTIFICA:**

Que, el Señor **TORRES ARGUDO JUAN DIEGO**, con código **64318**, alumno de la  
Escuela de **INGENIERIA DE SISTEMAS Y TELEMATICA**, tiene aprobado el  
**91,11%** de créditos de su malla curricular.

Cuenca, 27 de marzo de 2017



Derecho No. 001-001-000155345

mjmr.-





UNIVERSIDAD DEL AZUAY



Escuela  
Sistemas y  
Telemática

**Oficio Estudiante: Solicitud aprobación de  
Protocolo de Trabajo de Titulación**

IST-RE-EST-02  
Versión 01  
04/04/2017  
Página 1 de 1

Lugar de Almacenamiento  
F: Archivo Secretaría de la Facultad

Retención  
5 años

Disposición Final  
Almacenar en archivo pasivo de la Facultad

Cuenca, 12 de Mayo del 2017

Ingeniero,

Oswaldo Merchán Manzano

**DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN**

**UNIVERSIDAD DEL AZUAY**

De nuestra consideración,

Estimado Señor Decano, nosotros Nicolás Oswaldo Andrade Hidalgo con C.I. 0104051826, código estudiantil 63103 y Juan Diego Torres Argudo con C.I. 0105777932, código estudiantil 64318; Estudiantes de la Carrera de Sistemas y Telemática, solicitamos muy comedidamente a usted y por su intermedio al Consejo de Facultad, la aprobación del protocolo de trabajo de titulación con el tema **"RECOMENDADOR MÓVIL SENSIBLE AL CONTEXTO DE NOTICIAS Y EVENTOS UNIVERSITARIOS EMPLEANDO TECNOLOGÍAS SEMÁNTICAS."** previo a la obtención del título de Ingeniero en Sistemas y Telemática para lo cual adjuntamos la documentación respectiva.

Por la favorable acogida que brinde a la presente, anticipamos nuestro agradecimiento.

Atentamente:

Nicolás Andrade

Juan Torres

Estudiantes de la Carrera de Sistemas y Telemática



Lugar de Almacenamiento  
F: Archivo Secretaría de la Facultad

Retención  
5 años

Disposición Final  
Almacenar en archivo pasivo de la Facultad

Cuenca, 12 de mayo del 2017

Ingeniero,  
Oswaldo Merchán Manzano  
**DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN**  
UNIVERSIDAD DEL AZUAY

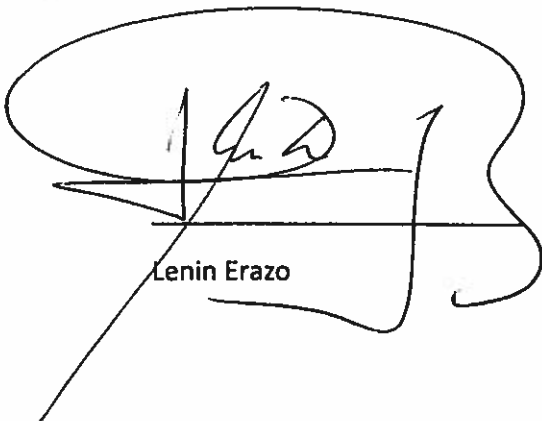
De mi consideración,

Yo, **Lenín Xavier Erazo Garzón** informo que he revisado el protocolo de trabajo de titulación previo a la obtención del título de Ingeniero/Ingeniera en Sistemas y Telemática, denominado **"RECOMENDADOR MÓVIL SENSIBLE AL CONTEXTO DE NOTICIAS Y EVENTOS UNIVERSITARIOS EMPLEANDO TECNOLOGÍAS SEMÁNTICAS"**, realizado por los estudiantes **Nicolás Oswaldo Andrade Hidalgo**, con código estudiantil 63103 y **Juan Diego Torres Argudo**, con código estudiantil 64318, protocolo que a mi criterio, cumple con los lineamientos y requerimientos establecidos por la carrera.

Por lo expuesto, me permito sugerir que sea considerado para la revisión y sustentación del mismo,

Sin otro particular, suscribo.

Atentamente



Lenín Erazo



UNIVERSIDAD DEL  
AZUAY

## 1. DATOS GENERALES

1.1 Nombre del estudiante: Andrade Hidalgo Nicolás Oswaldo

1.1.1 Código: 63103

1.1.2 Contacto:

Teléfono convencional: 2459340

Celular: 0984980528

Correo electrónico: [nicolasoah410@gmail.com](mailto:nicolasoah410@gmail.com)

1.2 Nombre del estudiante: Torres Argudo Juan Diego

1.2.1 Código: 64318

1.2.2 Contacto:

Teléfono convencional: 2830682

Celular: 0983881952

Correo electrónico: [suco\\_juan@hotmail.com](mailto:suco_juan@hotmail.com)

1.3 Director sugerido: Erazo Garzón, Lenin Ing.

1.3.1 Contacto:

Teléfono convencional: 24189687

Celular: 0991828833

Correo electrónico: [lerazo@uazuay.edu.ec](mailto:lerazo@uazuay.edu.ec)

1.4 Co-director sugerido: Sellers Walden, Chester Ing.

1.5 Asesor metodológico: Salgado Artéaga, Francisco Ing.

1.6 Tribunal designado:

Tribunal 1: Ing. Francisco Salgado

Tribunal 2: Ing. Chester Sellers

1.7 Aprobación: Junta Académica:

Consejo de Facultad:

1.8 Línea de Investigación de la carrera:

1.8.1 Código UNESCO: 1203 Informática de computadores

1203.17 Informática

1.8.2 Tipo de trabajo: Tesis en el campo formativo

1.9 Área de estudio: Computación sensible al contexto

Tecnologías semánticas

1.10 Título propuesto: Recomendador móvil sensible al contexto de noticias y eventos  
universitarios empleando tecnologías semánticas.

1.11 Subtítulo: No aplica



UNIVERSIDAD DEL  
AZUAY

1.12 Estado del proyecto: Nuevo/Multidisciplinario, incluye temas relacionados con:

Ingeniería del software, computación ubicua y móvil; computación sensible al contexto, tecnologías semánticas (ontologías).

## 2. CONTENIDO

2.1 Motivación de la investigación: El uso de las aplicaciones móviles en la actualidad

gracias al avance tecnológico se ha vuelto imprescindible para las personas. En este sentido, esta investigación pretende hacer uso de estas aplicaciones móviles integrando la sensibilidad al contexto para brindar información y/o servicios oportunos de acuerdo con la situación del entorno.

La elección de la presente investigación se ha visto motivada desde dos perspectivas:

primero, apoyar al ente universitario a través de una herramienta tecnológica "inteligente" para la difusión y recomendación personalizada de noticias y/o eventos universitarios basada en información contextual disponible y dirigida a estudiantes y profesores, como por ejemplo: perfil del usuario, carrera y nivel de estudio, preferencias, localización, temporalidad; y el segundo punto, el interés propio sobre los aplicativos móviles con sensibilidad al contexto y las tecnologías semánticas debido a que son un campo de investigación en pleno desarrollo.

### 2.2 Problemática:

En la actualidad el uso de las aplicaciones móviles en el dominio universitario se ha extendido notablemente a fin de brindar servicio de calidad e información oportuna a sus principales actores (estudiantes, profesores); sin embargo, en este tipo de aplicaciones resulta

esencial considerar la información contextual a fin de enriquecer la experiencia del usuario, mediante la provisión de información y/o servicios relevantes con su situación actual.

Particularmente, en el ámbito de la difusión de noticias y eventos universitarios a través de canales virtuales (portales web, aplicaciones móviles, SMS, etc.) los usuarios se ven abrumados por la gran cantidad de información de carácter general que reciben, generando insatisfacción en su uso y disminuyendo el impacto de este tipo de medios de difusión.

De ahí la necesidad de un sistema móvil sensible al contexto que apoye en la difusión de noticias y/o eventos que sean plenamente del interés personal para los estudiantes y profesores de conformidad con su contexto actual.

### **2.3 Pregunta de investigación:**

¿Cuál es la estructura de una red de ontologías y las tecnologías semánticas más apropiadas para implementar una aplicación móvil sensible al contexto para recomendar noticias y eventos en un dominio universitario?

### **2.4 Resumen:**

Este proyecto busca ayudar a los profesores y estudiantes a tener información personalizada en todo momento sobre las noticias y/o eventos universitarios según el perfil del usuario, carrera, calendario, lugar y otros datos contextuales que puedan ayudar a precisar de mejor forma las recomendaciones.

Este propósito se alcanzará a partir de una revisión bibliográfica que oriente el diseño de una ontología para representar contexto y la construcción de una aplicación móvil que tendrá como fuente de información una base de datos triple store accedida a través de un Framework, y



finalmente un razonador que ayude al aplicativo a inferir conocimiento contextual para recomendar al usuario final sobre eventos y noticias universitarios.

## 2.5 Indagación Exploratoria:

Con la rápida evolución de los dispositivos móviles el concepto de aplicaciones sensibles al contexto ha crecido en los últimos años. Según (Schilit, 1994) el contexto tiene varias definiciones dependiendo de la rama o ciencia que se trate, en los sistemas sensibles al contexto está caracterizado por hacer referencia a la ubicación, las personas que se encuentran en el entorno, servidores, y todos los dispositivos a los que se puede acceder en cualquier momento.

Gracias a los teléfonos inteligentes que cuentan con varios sensores como el GPS, acelerómetros, giroscopio que proporcionan información en todo momento y a los accesos a la red como WIFI o GSM, los teléfonos inteligentes son plataformas ideales para la computación ubicua. (Nalepa & Bobek, 2014)

La existencia de la sensibilidad al contexto o computación ubicua permite analizar toda la información que se puede obtener en el entorno del usuario, como el clima, altura, ubicación, eventos, gustos, etc. Para que mediante esta información el sistema responda de manera inteligente al usuario. (Ozgur Yurur, 2013)

Los sistemas sensibles al contexto tienen el reto de explotar los cambios que se dan en el entorno, analizando todo lo que está a su alrededor teniendo en cuenta tres aspectos importantes como son: en donde te encuentras, con quien te encuentras y los dispositivos cercanos con los que puedes interactuar incluyendo sensores de luminosidad, de ruido, GPS, conexiones a la red, etc. y reaccionar mediante la información obtenida. (Schilit, 1994)

Los sistemas sensibles al contexto brindan información relevante o servicios a un usuario y se adaptan al comportamiento del contexto actual en el que se encuentran sin una intervención del usuario. Dicho contexto se debe modelar y una de las maneras para lograrlo es mediante una ontología (Kuutti, Dvoryanchikova, Lobov, Martínez Lastra, & Vahtera, 2012).

Un concepto de (Asensio, 2008) para ontología indica que es la representación de conceptos u objetos en un dominio, en el que se toman en cuenta todas sus relaciones y propiedades. Las ontologías son utilizadas en varias áreas de las ciencias de la computación como inteligencia artificial, representaciones de conocimiento, webs semánticas, ingeniería del software, etc.

Para (Kuutti, Dvoryanchikova, Lobov, Martínez Lastra, & Vahtera, 2012) elegir un lenguaje ontológico es una decisión crucial para implementar una ontología. *Web Ontology Language* desarrollado por la comunidad W3C está diseñado para ser usado en situaciones donde la información que contiene un documento es para máquinas y no para humanos. OWL representa el significado de términos en vocabularios y relaciones entre esos términos, representa una semántica con tres elementos: Clases, individuales y propiedades.

El *Framework* de descripción de recursos (RDF) es un etiquetado gráfico de formato de datos con el objetivo de representar información en la web, la web semántica, y se utiliza para organizar el conocimiento, desempeñando un papel importante en las ontologías y la representación de conocimientos. Un modelo RDF consiste de tripletas que incluyen tres partes: sujeto, predicado y un objeto (Jamgade & Karale, 2015)



## 2.6 Objetivo general:

Desarrollar una aplicación móvil sensible al contexto para recomendar noticias y eventos en un dominio universitario.

## 2.7 Objetivos específicos:

1. Estudiar los modelos ontológicos de representación de contexto, las metodologías para la construcción de redes ontológicas y las tecnologías semánticas existentes para su implementación.
2. Especificar los requerimientos de software de la aplicación móvil sensible al contexto para recomendar noticias y eventos universitarios.
3. Construir una red de ontologías para representar contexto en el dominio de noticias y eventos universitarios.
4. Diseñar la arquitectura de la aplicación sensible al contexto para recomendar noticias y eventos universitarios.
5. Codificar e implementar la aplicación móvil sensible al contexto en un escenario de pruebas dentro de la Universidad del Azuay.

## 2.8 Metodología:

Se realizará una revisión sistémica de la literatura referente a la computación ubicua, computación sensible al contexto, modelos para representar el contexto y ontologías relacionadas con el dominio, así como también la revisión de tecnologías semánticas, arquitecturas y herramientas para el desarrollo de aplicaciones móviles sensibles del contexto.

Se presentará un análisis sobre los diferentes modelos para representar el contexto de las ontologías para poder elegir uno o varios modelos que ayuden a diseñar una red ontológica que permita representar el contexto en la aplicación móvil que se desarrollará.

Se utilizarán metodologías ágiles que ayuden a tener un desarrollo interactivo, permitiendo la comunicación constante entre todos los responsables del proyecto y de esta manera poder cumplir con el objetivo de desarrollar una aplicación móvil funcional que cumpla con los requerimientos de los usuarios.

Este proyecto se llevará a cabo mediante el análisis y estudio de los requerimientos que necesita la aplicación móvil sensible al contexto utilizando el estándar IEEE 830, como base para el diseño de los procesos, la ontología, la base de datos triple store, la arquitectura e interfaz gráfica de la aplicación.

## **2.9 Alcances y resultados esperados:**

Estado del arte sobre la computación sensible al contexto, modelos representación de contexto, ontologías y tecnologías semánticas que apoyen en la construcción de una red ontológica para inferir conocimiento según el contexto y que será puesta a prueba para verificar su correcto funcionamiento.

Obtener una aplicación móvil sensible al contexto para recomendar noticias y/o eventos universitarios que emplee una red de ontologías para representar contexto, base de datos triple store para su almacenamiento, tecnologías semánticas y un razonador para inferir conocimiento implícito. Se propiciará la reutilización de frameworks para la construcción de los diversos componentes de la aplicación.

## 2.10 Supuestos y riesgos:

Riesgos	Probabilidad	Alternativas de solución
Cambios en los requerimientos por desconocimiento de la temática.	media	Reuniones constantes.
Las personas clave para el avance del proyecto solo están disponibles una parte del tiempo.	media	Establecer un cronograma de revisiones.
La ontología no permite generar conocimiento implícito.	Alta	Pruebas sobre la red ontológica
El aprendizaje de la herramienta de desarrollo es más complejo de lo esperado.	Baja	Asignar un tiempo extra dentro del proyecto a la tarea de aprendizaje de la herramienta.

## 2.11 Esquema tentativo:

### CAPÍTULO I: ESTADO DEL ARTE

1. Computación ubicua y contexto
2. Computación sensible al contexto
3. Ontologías
4. Modelos para representar contexto
  - 4.1. Clasificación
  - 4.2. Modelos ontológicos para representar contexto
  - 4.3. Análisis comparativo de los modelos ontológicos estudiados
5. Otras ontologías relacionadas con el dominio de estudio

### CAPÍTULO II: MÉTODOS Y HERRAMIENTAS

1. Metodologías para la construcción de una red de ontologías
2. Tecnologías semánticas



### 2.13 Referencias

Asensio, E. S. (2008). *A MDD strategy for developing context-aware pervasive systems*. Valencia.

Erazo, L., Illescas, L., & Espinoza, M. (2016). Una ontología para representar el contexto en sistemas de computación académicos universitarios. *Revista Tecnológica-ESPOL*, 124-152.

Jamgade, A., & Karale, S. (2015). Ontology Based Approach for Semantic Information Retrieval System. *INTERNATIONAL JOURNAL FOR TRENDS IN ENGINEERING & TECHNOLOGY*.

Kuutti, A., Dvoryanchikova, A., Lobov, A., Martínez Lastra, J. L., & Vahtera, T. (2012). A Device Configuration Management Tool for Context-Aware System. *THT-Control Oy*.

Natepa, G. J., & Bobek, S. (2014). Rule-Based Solution for Context-Aware Reasoning on Mobile Devices. *Computer Science and Information Systems*, 11(1), 171-193.

Ozgur-Yurur, M. I. (2013). *Context-Awareness for Mobile Sensing: A Survey*.

Schilit, B. (1994). *Context-Aware Computing Applications*.

### 2.14 Anexos:

**2.15 Firma de responsabilidad (estudiante)**



Nicolás Andrade

**2.16 Firma de responsabilidad (estudiante)**

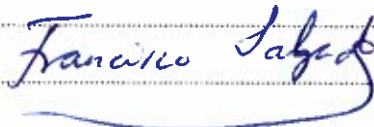


Juan Torres

**2.17 Firma de responsabilidad (director sugerido)**



Ing. Lenin Erazo Garzón



**2.18 Fecha de entrega: 10/05/2017**