



INSTITUTO POLITÉCNICO NACIONAL

**Unidad Profesional Interdisciplinaria en
Ingeniería y Tecnologías Avanzadas**

PROYECTO TERMINAL II

**“Módulo de consultas federadas geoespaciales en el
contexto de la Web de Linked Data para el triple store
Apache Marmotta”**

Que para obtener el título de
“Ingeniero en Telemática”

Presenta:

Páez Ortega Oswaldo Emmanuel

Asesores:

Dr. Luis Manuel Vilches Blázquez

Dra. Cyntia Eugenia Enríquez Ortiz



ÍNDICE GENERAL

1	CAPÍTULO III: MARCO TEÓRICO	5
1.1	WEB SEMÁNTICA	5
1.2	LINKED DATA	6
1.3	RDF (RESOURCE DESCRIPTION FRAMEWORK)	7
1.4	URI (<i>UNIFORM RESOURCE IDENTIFIER</i>)	7
1.5	RDF <i>TRIPLE STORE</i>	8
1.6	SPARQL	8
1.7	SPARQL ENDPOINT	9
1.8	GEOSPARQL	9
1.9	ARQUITECTURA SOA	10
1.10	PROTOCOLO HTTP	11
1.11	REST	11
1.12	JSON	12
1.13	TRADUCTOR	13
1.14	COMPILADOR	13
1.15	INTÉRPRETE	13
1.16	JERARQUÍA DE LENGUAJES:	13
1.17	PARTES DE UN COMPILADOR	14
2	CAPÍTULO IV: ANÁLISIS DEL SISTEMA	15
2.1	DESCRIPCIÓN GENERAL	15
2.2	PERSPECTIVA DEL PRODUCTO	15
2.3	CARACTERÍSTICAS DE LOS USUARIOS	15
2.4	RESTRICCIONES	15
2.5	SUPOSICIONES Y DEPENDENCIAS	16
2.6	REQUISITOS ESPECÍFICOS	16
2.7	REQUERIMIENTOS FUNCIONALES	16
2.8	REQUERIMIENTOS NO FUNCIONALES	24
2.9	INTERFACES DE HARDWARE	25
2.10	INTERFACES DE SOFTWARE	25
2.11	INTERFACES DE COMUNICACIÓN	25
3	CAPÍTULO V: DISEÑO DEL SISTEMA	26
3.1	CASO DE USO	26
3.2	DIAGRAMAS DE CLASE	27
3.3	DIAGRAMAS DE ESTADO	28
3.3.1	<i>Diagrama de estado para la aplicación web – Usuario</i>	<i>28</i>
3.3.2	<i>Diagrama de estado para la aplicación web – Administrador</i>	<i>29</i>
3.3.3	<i>Diagrama de estado para módulo de consultas en Apache Marmotta</i>	<i>30</i>
3.4	DIAGRAMAS DE SECUENCIA	31
3.4.1	<i>Diagrama de secuencia para aplicación web (Usuario).</i>	<i>31</i>
3.4.2	<i>Diagrama de secuencia para aplicación Web (Administrador)</i>	<i>33</i>
3.4.3	<i>Diagrama de secuencia para el módulo.</i>	<i>34</i>
3.5	SECUENCIA DE INTERFACES	37
4	REFERENCIAS	38

Índice de figuras

FIGURA 1 DIAGRAMA PROPUESTO POR TIM BERNERS-LEE MOSTRANDO LAS TECNOLOGÍAS QUE CONFORMA LA WEB SEMÁNTICA.....	5
FIGURA 2 LA NUBE DE DATOS LINKED DATA.	6
FIGURA 3 GRAFO DE RDF.....	7
FIGURA 4 DIAGRAMA DE CÓMO ESTÁ COMPUESTO UN URI.....	7
FIGURA 6 RESULTADO DE LA CONSULTA EN SPARQL.....	8
FIGURA 7 REPRESENTACIÓN DE DATOS ESPACIALES.....	9
FIGURA 8 RESULTADO DE LA CONSULTA EN GEOSPARQL.....	10
FIGURA 9 CASO DE USO DEL DISEÑO DEL SISTEMA.	26
FIGURA 10 DIAGRAMA DE CLASES.	27
FIGURA 11 DIAGRAMA DE ESTADOS - APLICACIÓN WEB – USUARIO.....	28
FIGURA 12 DIAGRAMA DE ESTADOS - APLICACIÓN WEB ADMINISTRADOR.....	29
FIGURA 21 DIAGRAMA DE ESTADOS - MÓDULO DE CONSULTAS FEDERADAS.....	30
FIGURA 22 DIAGRAMA DE SECUENCIA - APLICACIÓN WEB – USUARIO COMÚN.....	31
FIGURA 15 DIAGRAMA DE SECUENCIA - APLICACIÓN WEB - USUARIO COMÚN.....	32
FIGURA 16 DIAGRAMA DE SECUENCIA - APLICACIÓN WEB - USUARIO COMÚN, CONTINUACIÓN.	32
FIGURA 17 DIAGRAMA SECUENCIA APLICACIÓN WEB - ADMINISTRADOR.....	33
FIGURA 18 DIAGRAMA DE SECUENCIA PARA MÓDULO DE CONSULTAS FEDERADAS GEOESPACIALES.....	34
FIGURA 19 DIAGRAMA SECUENCIA INICIO SESIÓN.....	35
FIGURA 20 DIAGRAMA DE SECUENCIA - SELECCIÓN DE MODO.....	35
FIGURA 21 DIAGRAMA DE SECUENCIA - VALIDACIÓN CONSULTA.....	36

Índice de tablas

TABLA 1 CARACTERÍSTICA DE USUARIO NORMAL.	15
TABLA 2 CARACTERÍSTICAS DE USUARIO ADMINISTRADOR.	15
TABLA 3 REQUERIMIENTO FUNCIONAL ESTABLECER COMUNICACIÓN.....	16
TABLA 4 REQUERIMIENTO FUNCIONAL VALIDAR CONEXIÓN.	16
TABLA 5 REQUERIMIENTO FUNCIONAL SELECCIÓN MODO OPERACIÓN.	17
TABLA 6 REQUERIMIENTO FUNCIONAL MODO DATASET.	17
TABLA 7 REQUERIMIENTO FUNCIONAL SELECCIÓN DATASET.....	17
TABLA 8 REQUERIMIENTO FUNCIONAL CARGAR DATOS.	18
TABLA 9 REQUERIMIENTO FUNCIONAL MODO CONSULTA.....	18
TABLA 10 REQUERIMIENTO FUNCIONAL ENVIAR CONSULTA.....	18
TABLA 11 REQUERIMIENTO FUNCIONAL VALIDAR CONSULTA.	19
TABLA 12 REQUERIMIENTO FUNCIONAL RECIBIR RESULTADOS.	19
TABLA 13 REQUERIMIENTO FUNCIONAL VISUALIZAR DATOS.	19
TABLA 14 REQUERIMIENTO FUNCIONAL EXPLORAR DATOS.	20
TABLA 15 REQUERIMIENTO FUNCIONAL INICIAR DE SESIÓN.	20
TABLA 16 REQUERIMIENTO FUNCIONAL INICIAR DE SESIÓN.	20
TABLA 17 REQUERIMIENTO FUNCIONAL VALIDAR USUARIO.	21
TABLA 18 REQUERIMIENTO FUNCIONAL VALIDAR NUEVO USUARIO.....	21
TABLA 19 REQUERIMIENTO FUNCIONAL DAR DE BAJA A USUARIO.	21
TABLA 20 REQUERIMIENTO FUNCIONAL CARGAR CONSULTA FEDERADA.....	22
TABLA 21 REQUERIMIENTO FUNCIONAL EXTRACCIÓN URI Y ARGUMENTOS.	22
TABLA 22 REQUERIMIENTO FUNCIONAL CONSULTA TRIPLE STORE.	22
TABLA 23 REQUERIMIENTO FUNCIONAL PROCESAR RESULTADOS.....	23
TABLA 24 REQUERIMIENTO FUNCIONAL GUARDAR DATOS.	23
TABLA 25 REQUERIMIENTO NO FUNCIONAL DISPONIBILIDAD.....	24
TABLA 26 REQUERIMIENTO NO FUNCIONAL USABILIDAD.	24
TABLA 27 REQUERIMIENTO NO FUNCIONAL INTERFAZ DE LA APLICACIÓN.	24
TABLA 28. REQUERIMIENTO NO FUNCIONAL CONFIDENCIALIDAD	25

1 Capítulo III: Marco teórico

Los conceptos descritos en esta sección fueron fundamentales en el presente proyecto terminal para mostrar los elementos que lo componen. RDF, URI, *triple store* y *SPARQL endpoint* son conceptos fundamentales que se trabajaron a lo largo del desarrollo del módulo; tales conceptos explican los objetos con los que se tuvieron que abordar para este proyecto terminal.

En lo que respecta a *Linked Data* y Web Semántica, estos son conceptos que se tuvieron que contemplar, ya que no se debe de perder de vista sobre qué contexto se estuvo desarrollando el módulo, porque bien se pudo hablar de datos cualquiera pero no fue así, el proyecto terminal está construido bajo estos conceptos.

Con respecto a *SPARQL* y *GeoSPARQL*, estos fueron los estándares que se siguieron y cumplieron. Si no hubiera sido así, simplemente no se hubiera podido nada que funcione con el *triple store* Apache Marmotta o cualquier otra plataforma que esté basada en cualquiera de los dos estándares.

1.1 Web semántica

La Web semántica es el grupo de actividades que el consorcio de la *World Wide Web* con la intención de construir tecnologías que permitan publicar datos para que puedan ser legibles por computadoras a través de conceptos que definan los objetos de la Web, semántica que permita a las máquinas interpretar los conceptos de los objetos, metadatos (RDF) como recurso para describir a los objetos y ontologías para establecer la relación existente entre los objetos de la Web. En la figura 9 se muestra la pila que Tim Berners-Lee propone para describir la Web Semántica en cuanto a elementos que lo conforman.

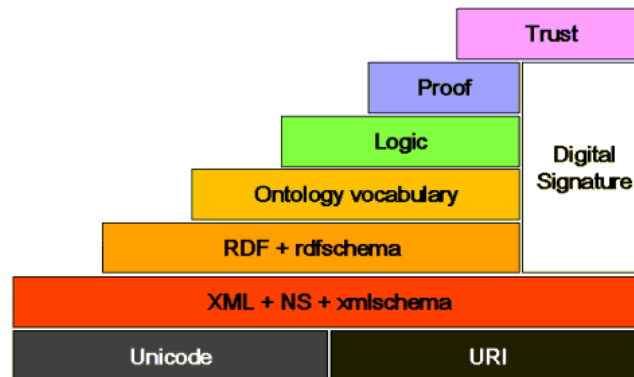


Figura 1 Diagrama propuesto por Tim Berners-Lee mostrando las tecnologías que conforma la Web Semántica.

Describiendo de abajo hacia arriba, en el primer nivel se encuentran las tecnologías que permiten la identificación de cada entidad (Unicode y URI). En el segundo nivel se encuentran las tecnologías que le dan estructura, pero no significado, al documento publicado en la web (XML, NS, XML schema). A partir del tercer nivel al sexto, la pila muestra una firma digital (columna a la derecha) para garantizar la autenticidad de cada entidad. En el tercer nivel están las tecnologías que describen el contenido del documento, tal que permiten a la mayoría de las computadoras entender el significado de la entidad en cuestión (RDF y RDF schema). En el cuarto nivel, si bien no es una tecnología, se encuentra un documento o archivo que define las relaciones entre entidades mediante ontologías y reglas de inferencia (comúnmente se implementa el lenguaje OWL para lograrlo). En el quinto nivel se encuentra la lógica, la cual reúne las diversas ontologías usadas al igual que las reglas de los lenguajes que se usaron para describir y estructurar la entidad; en este nivel se llevan a cabo las inferencias y se les da un significado a los datos. En el sexto nivel, se realiza

la prueba, es decir, el cómo se hicieron las inferencias y el origen de los datos. Por último, se encuentra la confianza, la confianza de que el sistema es capaz de funcionar correctamente, de que el sistema pueda explicar qué hace, del origen de las fuentes de datos y servicios, así como la tecnología e interfaz de usuario.

La Web Semántica se puede ver como un todo y el *Linked Data* como los elementos que la conforman puesto que proporciona sustento y las bases para que la Web Semántica sea construida correctamente [1].

1.2 Linked Data

El *Linked Data* es un conjunto de buenas prácticas para publicar y conectar datos estructurados en la web. Los principios [2] que tiene asociado son los siguientes:

- URI para identificar entidades en el mundo.
- HTTP que es el mecanismo con el que se recuperan recursos o descripciones de recursos.
- Protocolo SPARQL y archivos RDF para estructurar, consultar y enlazar los objetos presentes en la nube del *Linked Data*.

Ya que el *Linked Data* está basado en web, la diferencia entre sitios de Internet comunes y los basados en *Linked Data* es que mientras que los HTML simples en la web son conectados mediante hipervínculos comunes, *Linked Data* se basa en documentos que albergan datos en formato RDF.

En la figura 10 se observa la nube de datos *Linked Data*, la cual existe en que existe en *DBpedia*, donde cada burbuja representa un *triple store*. Cada color implica un dominio de datos distinto tales como gobierno, ciencias, multimedia entre otros. El dominio que compete al actual proyecto son los geoespaciales.

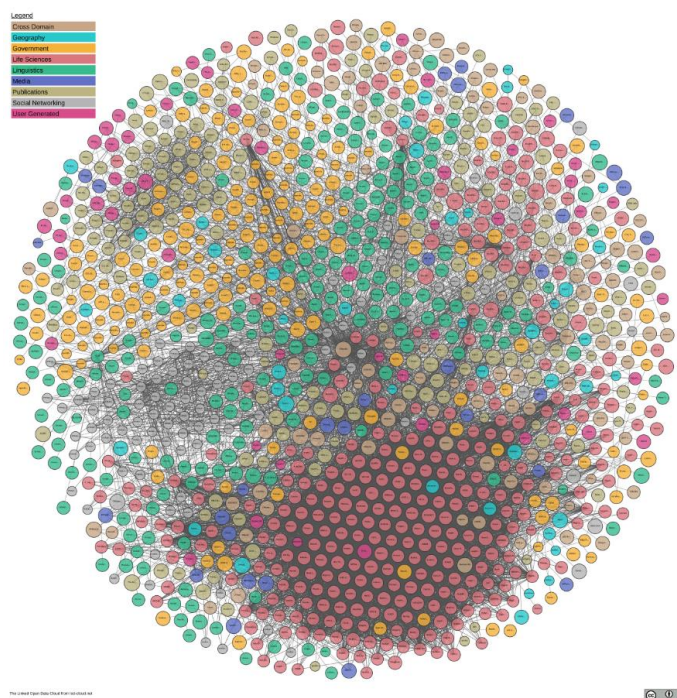


Figura 2 La nube de datos *Linked Data*.

1.3 RDF (Resource Description Framework)

El *framework* de descripción de recurso, RDF por sus siglas en inglés, es un modelo estándar para el intercambio de datos en la web. La característica de RDF es que extiende las estructuras de enlaces de la web, al usar URI tanto para nombrar relaciones entre cosas como para los puntos finales de las relaciones, a veces referido como “*triple*”. El modelo RDF permite representar los datos y la relación existente entre ellos mediante ontologías a través de relaciones semánticas. La relación semántica que guardan los *RDF* es: sujeto, predicado y objeto. El sujeto y el predicado de un *triple* son URI que identifican a cada uno. El predicado especifica como el sujeto y el objeto están relacionados, y también es representado por un URI. Esta característica provee un modelo de datos basado en grafos [3]. La figura 11 muestra un ejemplo de cómo diferentes sitios se enlazan entre sí mediante sus respectivas URI. Los vértices son los objetos y sujetos mientras que las aristas son los predicados, cada una asociada a una URI.

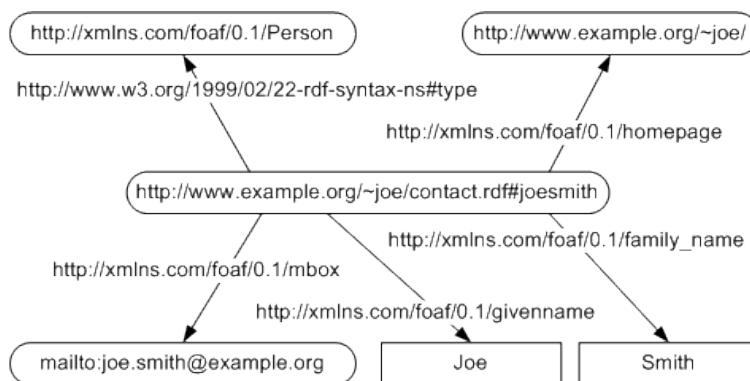


Figura 3 Grafo de RDF.

1.4 URI (Uniform Resource Identifier)

Identificador de recursos uniforme, URI por sus siglas en inglés, es una cadena ASCII que identifica recursos de información en la Web Semántica.

Tal y como se observa en la figura 12, una URI puede estar compuesto de un localizador de recursos uniforme (URL, por sus siglas en inglés), de un nombre de recursos uniforme (URN, por sus siglas en inglés) o de ambos [4].

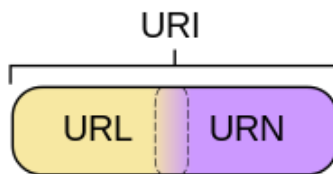


Figura 4 Diagrama de cómo está compuesto un URI.

Un ejemplo que se puede mostrar sobre la composición de una URI es la figura 13 en la que las letras en negritas denotan la ubicación donde el recurso está albergado como URL, y en letra normal, el nombre del recurso como URN.

http://sitiointernet.com/autor/bibliografia.html#posts

Figura 5 Ejemplo URI

1.5 RDF triple store

Es un tipo de base de datos basada en grafos de tripletas RDF [5], por lo que, al ser una base de datos basada en grafos, el *triple store* puede ser vista como una red de objetos enlazados. El *triple store* al ser una herramienta de la Web Semántica, las entidades que conforman a la base de datos, tripletas RDF, son representadas como sujeto, predicado y objeto o también puede ser considerada como sujeto, predicado y etiqueta.

1.6 SPARQL

SPARQL es un acrónimo para el Protocolo *SPARQL* y Lenguaje de Consultas RDF, por sus siglas en inglés, y es un protocolo y lenguaje de consultas para *Linked Data* en la web o bases de datos semánticas basadas en grafos (*RDF triple stores*) [6]. *SPARQL* está diseñado y respaldado por el consorcio de la web (W3C).

El siguiente código muestra cómo hacer una consulta *SPARQL* de los músicos mexicanos famosos que ya han muerto y que están sobre el *triple store* de Dbpedia.

```
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbp: <http://dbpedia.org/ontology/>
SELECT ?musico ?nombreMusico ?fechaFallecimiento
WHERE {
    ?musico dcterms:subject
    <http://dbpedia.org/resource/Category:Mexican_musicians>;
    rdfs:label ?nombreMusico ;
    dbp:birthDate ?fechaNacimiento ;
    dbp:deathDate ?fechaFallecimiento .
    FILTER (LANG(?nombreMusico) = "es")
}
```

La figura 14 muestra el resultado de la consulta

musico	nombreMusico	fechaFallecimiento
http://dbpedia.org/resource/Juan_García_Esquivel	"Juan García Esquivel"@es	2002-01-03
http://dbpedia.org/resource/Juan_García_Esquivel	"Juan García Esquivel"@es	"2002-1-3"^^<http://www.w3.org/2001/XMLSchema#date>
http://dbpedia.org/resource/Cornelio_Reyna	"Cornelio Reyna"@es	1997-01-22
http://dbpedia.org/resource/Cornelio_Reyna	"Cornelio Reyna"@es	"1997-1-22"^^<http://www.w3.org/2001/XMLSchema#date>
http://dbpedia.org/resource/Manolo_Muñoz	"Manolo Muñoz"@es	"2000-3-14"^^<http://www.w3.org/2001/XMLSchema#date>
http://dbpedia.org/resource/Carlos_Gómez_Barrera	"Carlos Gómez Barrera"@es	"1996-1-1"^^<http://www.w3.org/2001/XMLSchema#date>
http://dbpedia.org/resource/Tito_Guizar	"Tito Guizar"@es	1999-12-24
http://dbpedia.org/resource/Lorenzo_Barcelata	"Lorenzo Barcelata"@es	"1943-7-13"^^<http://www.w3.org/2001/XMLSchema#date>
http://dbpedia.org/resource/Consuelo_Velázquez	"Consuelo Velázquez"@es	"2005-1-22"^^<http://www.w3.org/2001/XMLSchema#date>
http://dbpedia.org/resource/Chavela_Vargas	"Chavela Vargas"@es	2012-08-05
http://dbpedia.org/resource/Chavela_Vargas	"Chavela Vargas"@es	"2012-8-5"^^<http://www.w3.org/2001/XMLSchema#date>
http://dbpedia.org/resource/José_Mojica	"José Mojica"@es	1974-09-20
http://dbpedia.org/resource/José_Mojica	"José Mojica"@es	"1974-9-20"^^<http://www.w3.org/2001/XMLSchema#date>
http://dbpedia.org/resource/Rafael_Méndez	"Rafael Méndez (trompetista)"@es	"1981-9-15"^^<http://www.w3.org/2001/XMLSchema#date>
http://dbpedia.org/resource/Chico_Che	"Chico Che"@es	"1989-3-29"^^<http://www.w3.org/2001/XMLSchema#date>
http://dbpedia.org/resource/Macedonio_Alcalá	"Macedonio Alcalá"@es	1869-08-24
http://dbpedia.org/resource/Macedonio_Alcalá	"Macedonio Alcalá"@es	"1869-8-24"^^<http://www.w3.org/2001/XMLSchema#date>
http://dbpedia.org/resource/Fernando_Valdés	"Fernando Valdés"@es	1978-12-14
http://dbpedia.org/resource/Ángel_Tavira	"Ángel Tavira"@es	"2008-6-30"^^<http://www.w3.org/2001/XMLSchema#date>
http://dbpedia.org/resource/Enrique_Ballesté	"Enrique Ballesté"@es	"2015-9-19"^^<http://www.w3.org/2001/XMLSchema#date>
http://dbpedia.org/resource/José_Agustín_Ramírez_Altamirano	"José Agustín Ramírez Altamirano"@es	"1957-9-12"^^<http://www.w3.org/2001/XMLSchema#date>
http://dbpedia.org/resource/Zacarias_Salmerón	"Zacarias Salmerón"@es	"2011-1-29"^^<http://www.w3.org/2001/XMLSchema#date>
http://dbpedia.org/resource/Silvestre_Vargas	"Silvestre Vargas"@es	"1985-10-7"^^<http://www.w3.org/2001/XMLSchema#date>
http://dbpedia.org/resource/Cirilo_Marmolejo	"Cirilo Marmolejo"@es	"1960-1-1"^^<http://www.w3.org/2001/XMLSchema#date>

Figura 5 Resultado de la consulta en SPARQL.

Actualmente se encuentra en su segunda versión, *SPARQL 1.1* [6], y ya describe una extensión para explícitamente delegar subconsultas a diferentes *SPARQL endpoint*. Esta característica es conocida como consulta federada.

1.7 SPARQL endpoint

Se le denominar *SPARQL endpoint* al identificador único de recursos, URI por sus siglas en inglés, asociado al servidor HTTP que ofrece y devuelve peticiones HTTP para peticiones provenientes de clientes que usan el protocolo SPARQL [7]. Un ejemplo de *SPARQL endpoint* es DBpedia [8] el cual es uno de los *endpoints* más famoso ya que en ella se albergan aproximadamente 4.58 millones de objetos en su base de datos de conocimiento.

1.8 GeoSPARQL

Es un lenguaje de consultas geográficas para datos RDF en la Web Semántica estandarizado por el *Open Geospatial Consortium* (OGC) [9]. *GeoSPARQL* define un vocabulario para la representación de datos en RDF y también define una extensión para el lenguaje de consultas geoespaciales *SPARQL*. Esta extensión de *SPARQL* para geo datos es útil en la solución de problemas de logística, hidrología y turismo [10].

GeoSPARQL busca relaciones topológicas entre objetos que posean datos asociados a una ubicación geográfica. Dicha búsqueda lo realiza mediante tres componentes principales:

- Definición de un vocabulario para representar características, geometrías y sus relaciones (ontologías).
- Conjunto de funciones espaciales para llevar a cabo consultas en *SPARQL*.
- Conjunto de reglas de transformación de consultas.

Un ejemplo de consulta es el siguiente: Determinar los objetos que estén completamente contenidos en la figura A de la figura 15.

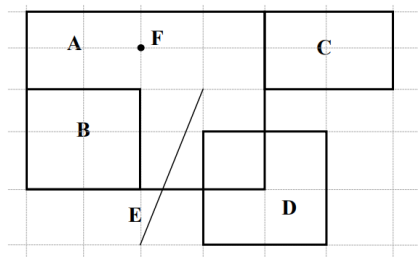


Figura 6 Representación de datos espaciales.

A continuación, la consulta *SPARQL* asociada para resolver ese interrogante:

```
PREFIX my: <http://example.org/ApplicationSchema#>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
SELECT ?f
WHERE { my:A my:hasExactGeometry ?aGeom .
        ?aGeom geo:asWKT ?aWKT .
```

```

    ?f my:hasExactGeometry ?fGeom .
    ?fGeom geo:asWKT ?fWKT .
    FILTER (geof:sfContains(?aWKT, ?fWKT) &&
            !sameTerm(?aGeom, ?fGeom))
}

```

El resultado se muestra en la figura 16.

?f
my:B
my:F

Figura 7 Resultado de la consulta en GeoSPARQL.

Como era de esperarse, el resultado dice que 'A' contiene tanto a 'B' como al punto 'F'

1.9 Arquitectura SOA

La arquitectura orientada a servicios, SOA por sus siglas en inglés, es un concepto en la ingeniería de software cuya intención es reducir costos de implementación, servicios para clientes innovadores, agilidad para adaptar cambios del sistema [11]. Sus características son las siguientes:

- Objetivos de negocio ligados a la infraestructura de tecnologías de información (TI).
- Orientada a la arquitectura de sistemas. Esto busca que la lógica de procesos de negocio no intervenga con la lógica del software de un sistema.
- Separación de objetivos: Dividir objetivos primarios en diferentes características con funcionalidades estrechas tan pequeñas como sean posible.
- Modularidad: La aplicación estará dividida en piezas distinguibles las cuales, desempeñarán una función en específico en el sistema.
- Bajo acoplamiento: Los atributos de los componentes de un sistema no tienen y/o no hacen uso del conocimiento de otros componentes independientes.
- Encapsulación: El acceso a datos con sus respectivas instrucciones de manipulación, las cuales estarán dentro de un paquete, es posible mediante una interfaz independiente.
- Interfaces: Implementación de un pequeño conjunto de interfaces que son mantenidas de manera separada.
- Mensajes: Uso de mensaje que contengan información a ser intercambiada a través de las interfaces mediante una estructura y vocabulario delimitado por un esquema.
- Reutilización: Es la acción de reutilizar un componente múltiples veces.
- Composabilidad: Capacidad de seleccionar componentes y ensamblarlos de diversas maneras que puedan cumplir el objetivo de la aplicación.

1.10 Protocolo HTTP

El protocolo HTTP es un protocolo a nivel de aplicación para colaborar, distribuir sistemas de información de tipo hipermedia (texto, imagen, audio, video, mapas). La comunicación entre sistemas, según el protocolo, está basado en respuestas y peticiones [12].

- El cliente hace una petición HTTP.
- El servidor recibe la petición.
- El servidor procesa la petición.
- El servidor lleva a cabo una respuesta HTTP con la información solicitada o con un mensaje de error.
- El cliente recibe la respuesta.

Así mismo, el protocolo HTTP establece un grupo de métodos de petición, los cuales indican que acción se quiere llevar a cabo sobre un recurso específico. Los métodos son los siguientes [13].

- GET: Pide una representación de un recurso específico. Este método solo debe ser capaz de recuperar datos.
- POST: Crea un nuevo recurso en el servidor.
- PUT: Las representaciones actuales de un recurso son sustituidas por la información que lleva el mensaje, y en caso de no existir, lo crea.
- DELETE: Elimina un recurso determinado.
- HEAD: Petición similar a la de GET, pero sin cuerpo de la respuesta.
- CONNECT: Establece comunicación en 2 vías con el servidor del recurso solicitado. Comúnmente es usado para llevar a cabo una comunicación túnel.
- TRACE: Lleva a cabo un mensaje de prueba de ida y vuelta en toda la ruta hasta el recurso objetivo con el fin de ser un mecanismo de depuración.
- OPTIONS: Método que permite describir las opciones de comunicación para un recurso en específico.
- PATCH: Permite modificar un recurso de manera parcial, a diferencia del método PUT que lo hace sobre todo el recurso.

1.11 REST

La transferencia de estado representacional, REST por sus siglas en inglés, es un tipo de arquitectura que define reglas de comunicación entre sistemas computacionales en la Web [14].

A pesar de que *REST* no es un estándar, si hace uso de ellos

- URL.
- HTTP.
- XML, GIF, JPG, etc. (representación de recursos).
- Extensiones multipropósito de correo de Internet (MIME).

Las características de *REST* son:

- Basado en cliente servidor: Consumo de componentes mediante peticiones.

- Sistemas independientes: Los sistemas computacionales no deben de estar desarrollados en el mismo lenguaje o paradigma de programación mientras estos cumplan el estilo de arquitectura *REST* se puede establecer comunicación entre los sistemas.
- Uso de memoria caché: Con el fin de mejorar la eficiencia de respuestas, los sistemas deben de ser capaces de decidir si las respuestas son o no parte de la memoria caché.
- Recursos etiquetados: Los sistemas deben de identificar a sus recursos con una URL.
- Interfaz uniforme: Cualquier recurso del sistema puede accederse mediante una interfaz genérica. Ejemplo: métodos HTTP.
- Representación de recursos enlazados: Las representaciones de los recursos deben de estar conectados entre sí para que el usuario sea capaz de ir de una representación a otra.
- Capas entre los componentes: Con el fin de otorgar servicios al usuario como seguridad, privacidad, eficiencia de servicio, entre otros, pueden usarse sistemas intermediarios como *gateways*, servidores *proxy* o servidores *cache* por mencionar algunos.

1.12 JSON

La notación de Objetos de JavaScript, *JSON* por sus siglas en inglés, es un formato de intercambio de información basado en texto e independiente de lenguaje. El formato fue un derivado del estándar del lenguaje de programación *ECMAScript* [15].

JSON está conformado por las siguientes 2 estructuras:

- Colección de pares nombre/valor: En diversos lenguajes de programación implementan esta estructura como un objeto, diccionario, registro, tabla *hash*, arreglos asociativos o lista de claves.
- Lista ordenada de valores: Se implementa en los lenguajes de programación como vectores, arreglos, secuencias o listas.

Ejemplo de un objeto *JSON* es

```
{
  "Image": {
    "Width": 800,
    "Height": 600,
    "Title": "View from 15th Floor",
    "Thumbnail": {
      "Url": "http://www.example.com/image/481989943",
      "Height": 125,
      "Width": 100
    },
    "Animated" : false,
    "IDs": [116, 943, 234, 38793]
  }
}
```

1.13 Traductor

Es un programa que convierte un programa fuente en un programa objeto.

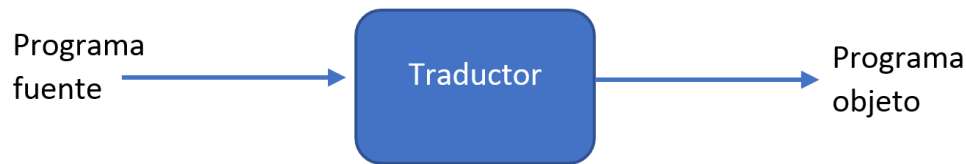


Fig. 9 Traductor.

1.14 Compilador

Es un traductor que convierte lenguaje de medio o alto nivel a lenguaje de máquina o ensamblador. Por ejemplo: C, Java, Pascal, Fortran, entre otros.

1.15 Intérprete

Es un programa que traduce y ejecuta al mismo tiempo lenguaje de alto nivel. Por ejemplo: LISP, Basic, Logo entre otros.

1.16 Jerarquía de lenguajes:

La jerarquía de lenguajes se refiere al nivel de dependencia que tiene un cierto lenguaje con la máquina o procesador. Los lenguajes se clasifican de la siguiente forma:

- Lenguajes de máquina: Son los nativos de cada computadora y describen la arquitectura de esta. Las instrucciones en lenguajes de máquina se representan mediante códigos numéricos. La comunicación entre los lenguajes de máquina y la máquina no son necesarios.
- Lenguajes ensambladores: Son conjuntos de símbolos que representan unívocamente a cada código numérico del lenguaje de máquina asociado. El nombre que reciben las instrucciones de este tipo de lenguajes es nemónico.
- Lenguajes de nivel medio: Son lenguajes que permiten el uso eficiente de lenguajes ensambladores y también sobre las estructuras de control de flujo, de iteración y el orden de un lenguaje de alto nivel.
- Lenguajes de alto nivel de uso general: Este tipo de lenguajes permiten abstraerse de la estructura interna de la computadora al manipular las estructuras de control de flujo, de iteración y secuencia.
- Lenguajes de propósito específico: Son lenguajes de alto nivel que solucionan problemas especiales en determinadas tareas como PostgreSQL para manejar bases de datos.
- Lenguajes de inteligencia artificial: Estos lenguajes permiten declarar explícitamente conocimiento y las estructuras de control de flujo, de iteración y secuencia no son expresadas en el programa por lo que está abstraída la lógica del control de la computadora.

1.17 Partes de un compilador

Si bien los compiladores pueden clasificarse en función a su construcción como de una, dos, múltiples pasadas, en función de su construcción o de alguna característica especial. Sin embargo, generalmente se dividen en 2 partes: análisis y síntesis.

La parte de análisis está compuesta por:

- Analizador léxico o *scanner*: Encargado de analizar el código fuente y separar las palabras mediante token los cuales son un conjunto de caracteres como pueden ser palabras reservadas, variables, símbolos aritméticos, de relación entre otros. Elimina comentarios y los tokens son almacenados en una tabla de símbolos.
- Analizador sintáctico *parser*: Encargado de analizar la correcta agrupación de los tokens en frases gramaticales las cuales pueden ser representadas mediante un *parse tree*.
- Analizador semántico: Encargado de verificar la consistencia de las expresiones y/o significado del código mediante la localización de errores.

La parte de síntesis está compuesta por:

- Generador de código intermedio: Encargado de llevar acabo equivalencias entre el código fuente y código máquina. Usualmente se usan como representación la notación de postfijos, árbol de direcciones de código o un árbol de sintaxis.
- Optimizador de código: Encargado de reducir el código generado por el generador de código intermedio para reducir el número de recursos y mejora la velocidad de ejecución.
- Generador de código: Genera código ensamblador o código ejecutable de tal forma que la máquina pueda interpretar el código fuente [16].

2 Capítulo IV: Análisis del sistema

El análisis de requerimientos llevado a cabo en este proyecto está basado en el estándar IEEE 830-1998 [17]. En este capítulo se presenta la descripción del proyecto especificando qué es lo que debe de hacer.

2.1 Descripción general

Se presenta una perspectiva de lo que debe de hacer el producto, las características de los usuarios del proyecto, qué restricciones existen, así como las suposiciones y dependencias que se asumen para que el proyecto funcione.

2.2 Perspectiva del producto

El módulo de consultas que se desarrolló ha permitido que el software Apache Marmotta tenga la capacidad de realizar consultas federadas con el objetivo de devolver una respuesta de datos geoespaciales unificada de los diversos *triple store* que se especifiquen en la consulta ingresada.

La aplicación Web permite llevar a cabo las consultas federadas geoespaciales y usa *GeoYASGUI* como herramienta externa para visualizar e interactuar con los resultados en un mapa.

2.3 Características de los usuarios

A continuación, se presentan las características que los 2 tipos de usuarios que pueden hacer uso del sistema.

Tabla 1 Característica de usuario normal.

Tipo de usuario	Normal
Formación	Estudiante, profesor y/o investigador
Habilidades	Conocimiento <i>SPARQL</i> y <i>GeoSPARQL</i>
Actividades	Iniciar sesión, consultar, visualizar datos, filtrar datos.

Tabla 2 Características de usuario administrador.

Tipo de usuario	Administrador
Formación	Estudiante, profesor y/o investigador
Habilidades	Uso de aplicación Web
Actividades	Iniciar sesión, administrar usuarios.

2.4 Restricciones

- Conexión a Internet
- Consultas basadas en los protocolos *SPARQL 1.1* y *GeoSPARQL*
- Las consultas pueden ser federadas
- Consultas a la nube *Linked Data*
- El lenguaje de programación para el desarrollo del módulo que fue implementado en Apache Marmotta es Java.
- Los lenguajes de programación y de marcado para la plataforma fueron
 - HTML
 - CSS
 - JavaScript
 - Java

- La aplicación está basada en métodos HTTP, en REST y JSON.
- Los resultados de las consultas poseen un LIMIT que permite limitar el número de resultados por recuperar que a su vez están asociados a la cantidad de memoria RAM disponible en el sistema. Si no se contara con este mecanismo, es probable que el sistema no pudiera procesar toda la información que Apache Marmotta devolviera al momento de hacer una consulta.

2.5 Suposiciones y dependencias

- Se consideró que la herramienta *GeoYASGUI* al momento de hacer el presente proyecto terminal seguiría usando las tecnologías mencionadas para la aplicación Web.
- Se asumió que el módulo de consultas federadas estuviera ya desarrollado cuando la aplicación Web estuviera lista.

2.6 Requisitos específicos

En esta sección se muestran los requerimientos funcionales y no funcionales que contiene el tanto el módulo de consultas como el de la aplicación Web.

2.7 Requerimientos funcionales

Los requerimientos funcionales para la aplicación Web son los siguientes:

Tabla 3 Requerimiento funcional establecer comunicación

Número de requisito	RF01
Nombre de requisito	Establecer comunicación
Tipo	Obligatorio
Características	El usuario debe de inicializar la aplicación Web para poder usarla.
Descripción del requerimiento	Se debe de establecer comunicación entre el software Apache Marmotta e Internet con la aplicación Web ya que, sin ellos la aplicación Web no puede funcionar.
Requerimiento no funcional	RNF01 RNF02 RNF03
Prioridad del requisito	Alta/Esencial

Tabla 4 Requerimiento funcional validar conexión.

Número de requisito	RF02
Nombre de requisito	Validar conexión
Tipo	Obligatorio
Características	La aplicación Web debe validar la conexión.
Descripción del requerimiento	Para usar la aplicación Web, se debe de conectar la aplicación con Apache Marmotta e Internet con el fin de avanzar o mostrar un mensaje de error de conexión en la aplicación.
Requerimiento no funcional	RNF01
Prioridad del requisito	Alta/Esencial

Tabla 5 Requerimiento funcional selección modo operación.

Número de requisito	RF03
Nombre de requisito	Selección modo operación
Tipo	Obligatorio
Características	El usuario deberá escoger el modo de uso de la aplicación: Consultar <i>dataset</i> o realizar consulta.
Descripción del requerimiento	La aplicación Web da la opción de escoger 2 opciones de operación. En la primera, existen varios <i>datasets</i> precargados en Apache Marmotta mientras que, en la segunda opción, el usuario puede ingresar una consulta que quiera ser ejecutada en tiempo real. Los resultados de ambas opciones se pueden visualizar en la aplicación Web.
Requerimiento funcional	no RNF02 RNF03
Prioridad del requisito	Alta/Esencial

Tabla 6 Requerimiento funcional modo *dataset*.

Número de requisito	RF04
Nombre de requisito	Modo <i>dataset</i>
Tipo	Obligatorio
Características	La aplicación Web muestra los <i>datasets</i> disponibles en Marmotta.
Descripción del requerimiento	Los <i>datasets</i> disponibles en Apache Marmotta son desplegados en la aplicación Web para que usuario escoja cual quiere explorar.
Requerimiento funcional	no RNF02 RNF03
Prioridad del requisito	Alta/Esencial

Tabla 7 Requerimiento funcional selección *dataset*.

Número de requisito	RF05
Nombre de requisito	Selección <i>dataset</i>
Tipo	Obligatorio
Características	El usuario puede escoger los <i>dataset</i> disponibles en el sistema para luego ser visualizados
Descripción del requerimiento	El sistema Web en conjunto con Apache Marmotta, muestra los <i>datasets</i> disponibles de consultas federadas previas y posteriormente, con la herramienta <i>GeoYASGUI</i> , visualizar e interactuar con los resultados de la consulta asociada a dicho <i>dataset</i> .
Requerimiento funcional	no RNF01 RNF02 RNF03
Prioridad del requisito	Alta/Esencial

Tabla 8 Requerimiento funcional cargar datos.

Número de requisito	RF06
Nombre de requisito	Cargar <i>dataset</i>
Tipo	Obligatorio
Características	Se carga el <i>dataset</i> en la aplicación Web.
Descripción del requerimiento	Cuando el usuario escoja cuál <i>dataset</i> quiere explorar, la aplicación pide los datos a Apache Marmotta para que sean cargados en la aplicación Web.
Requerimiento funcional	no RNF01
Prioridad del requisito	Alta/Esencial

Tabla 9 Requerimiento funcional modo consulta.

Número de requisito	RF07
Nombre de requisito	Modo consulta
Tipo	Obligatorio
Características	La aplicación Web muestra una caja de texto donde el usuario ingresará una consulta y también ofrece ayuda que le permita al usuario menos experto construir una consulta federada.
Descripción del requerimiento	En este modo, el usuario debe ingresar una consulta que en una caja de texto cargada en la aplicación Web o en dado caso que el usuario no sea experto, el usuario puede pedir ayuda al sistema para escribir la consulta federada.
Requerimiento funcional	no RNF02 RNF03
Prioridad del requisito	Alta/Esencial

Tabla 10 Requerimiento funcional enviar consulta.

Número de requisito	RF08
Nombre de requisito	Enviar consulta
Tipo	Obligatorio
Características	Botón que le permite al usuario enviar la consulta que quiera ser ejecutada.
Descripción del requerimiento	Una vez terminada la tarea de escribir una consulta por parte del usuario, se debe enviar la consulta escrita a Apache Marmotta mediante la acción de hacer click en un botón.
Requerimiento funcional	no RNF01 RNF02 RNF03
Prioridad del requisito	Alta/Esencial

Tabla 11 Requerimiento funcional validar consulta.

Número de requisito	RF09
Nombre de requisito	Validar consulta
Tipo	Obligatorio
Características	Validará la consulta enviada a Apache Marmotta.
Descripción del requerimiento	El compilador de Apache Marmotta es capaz de identificar qué tipo de consulta fue la que se ingresó. La respuesta puede ser: federada, normal o error. La aplicación muestra un mensaje de error en caso de que la consulta enviada haya sido incorrecta.
Requerimiento funcional	no RNF01 RNF02 RNF03
Prioridad del requisito	Alta/Esencial

Tabla 12 Requerimiento funcional recibir resultados.

Número de requisito	RF10
Nombre de requisito	Recibir resultados
Tipo	Obligatorio
Características	La aplicación Web recibe los resultados de la consulta enviada por el usuario.
Descripción del requerimiento	Si la consulta enviada a Apache Marmotta no tuvo algún error, la aplicación Web recibe los datos provenientes de Apache Marmotta.
Requerimiento funcional	no RNF01
Prioridad del requisito	Alta/Esencial

Tabla 13 Requerimiento funcional visualizar datos.

Número de requisito	RF11
Nombre de requisito	Visualizar datos
Tipo	Obligatorio
Características	El usuario puede visualizar los resultados de la consulta ingresada o <i>dataset</i> selección.
Descripción del requerimiento	Una vez que Apache Marmotta haya terminado de hacer la consulta o de cargar el <i>dataset</i> , la aplicación web debe dar la opción al usuario de visualizar los datos usando la herramienta <i>GeoYASGUI</i> .
Requerimiento funcional	no RNF01 RNF02 RNF03
Prioridad del requisito	Alta/Esencial

Tabla 14 Requerimiento funcional explorar datos.

Número de requisito	RF12
Nombre de requisito	Explorar datos
Tipo	Obligatorio
Características	La aplicación Web carga la herramienta <i>GeoYASGUI</i> .
Descripción del requerimiento	Cuando se tenga el <i>dataset</i> seleccionado cargado o los resultados de la consulta, la aplicación Web carga la herramienta <i>GeoYASGUI</i> con la que el usuario puede visualizar e interactuar con los datos.
Requerimiento funcional	no RNF01 RNF02 RNF03
Prioridad del requisito	Alta/Esencial

Tabla 15 Requerimiento funcional iniciar de sesión.

Número de requisito	RF13
Nombre de requisito	Iniciar sesión
Tipo	Obligatorio
Características	El usuario debe de iniciar sesión para acceder al sistema.
Descripción del requerimiento	Para poder hacer uso del sistema de exploración de datos, el usuario debe ingresar su usuario y contraseña.
Requerimiento funcional	no RNF03 RNF04
Prioridad del requisito	Alta/Esencial

Tabla 16 Requerimiento funcional iniciar de sesión.

Número de requisito	RF14
Nombre de requisito	Validar nuevo usuario
Tipo	Obligatorio
Características	Se corrobora que el correo y contraseña sean correctos.
Descripción del requerimiento	Para dar acceso al sistema, se verifica en el <i>backend</i> de la aplicación Web, que el correo y contraseña sean correctos para brindar el acceso a la aplicación Web.
Requerimiento funcional	no RNF03 RNF04
Prioridad del requisito	Alta/Esencial

Tabla 17 Requerimiento funcional validar usuario.

Número de requisito	RF15
Nombre de requisito	Registrar usuarios
Tipo	Obligatorio
Características	El administrador de la aplicación Web debe de registrar a los usuarios en la aplicación Web para que puedan acceder a cualquier funcionalidad de esta.
Descripción del requerimiento	Para que los usuarios puedan acceder a la aplicación Web, ellos tendrán que proporcionar correo electrónico, una contraseña y nivel de experiencia en SPARQL y GeoSPARQL para que el administrador de la aplicación los registre en el sistema.
Requerimiento funcional	no RNF03 RNF04
Prioridad del requisito	Alta/Esencial

Tabla 18 Requerimiento funcional validar nuevo usuario.

Número de requisito	RF16
Nombre de requisito	Validar nuevo usuario
Tipo	Obligatorio
Características	El <i>backend</i> debe corroborar si los datos del nuevo usuario son válidos.
Descripción del requerimiento	Cada vez que el administrador de la aplicación Web ingrese datos para registrar a un nuevo usuario, el <i>backend</i> de la aplicación corrobora que el correo ingresado no exista.
Requerimiento funcional	no RNF01 RNF02 RNF03 RNF04
Prioridad del requisito	Alta/Esencial

Tabla 19 Requerimiento funcional dar de baja a usuario.

Número de requisito	RF16
Nombre de requisito	Dar de baja a usuario
Tipo	Obligatorio
Características	Mediante la aplicación Web, el administrador puede dar de baja al usuario.
Descripción del requerimiento	El administrador debe tener la posibilidad de dar de baja mediante el correo y contraseña del usuario para que posteriormente el usuario no pueda acceder de nuevo a la aplicación.
Requerimiento funcional	no RNF01 RNF02 RNF03 RNF04
Prioridad del requisito	Alta/Esencial

Los requerimientos funcionales para el módulo de consultas federadas son los siguientes:

Tabla 20 Requerimiento funcional cargar consulta federada.

Número de requisito	RF17
Nombre de requisito	Cargar consulta federada
Tipo	Obligatorio
Características	Apache Marmotta le hace llegar la consulta federada al módulo de consultas que se va a desarrollar.
Descripción del requerimiento	Para que se puedan extraer los URI y argumentos de la consulta federada, la consulta proveniente del <i>SPARQL endpoint</i> debe de llegar al módulo de consultas federadas.
Requerimiento funcional	no RNF01
Prioridad del requisito	Alta/Esencial

Tabla 21 Requerimiento funcional extracción URI y argumentos.

Número de requisito	RF18
Nombre de requisito	Extracción URI y argumentos
Tipo	Obligatorio
Características	Se extraen los URI y argumentos de la consulta federada
Descripción del requerimiento	Para poder llevar a cabo la consulta federada, se deben de extraer los URI de los <i>triple store</i> y argumentos de la consulta que se van a realizar.
Requerimiento funcional	no RNF01
Prioridad del requisito	Alta/Esencial

Tabla 22 Requerimiento funcional Consulta *triple store*.

Número de requisito	RF19
Nombre de requisito	Consulta <i>triple store</i>
Tipo	Obligatorio
Características	Se consulta el <i>triple store</i> asociado a cada URI con sus respectivos argumentos.
Descripción del requerimiento	Para llevar a cabo una consulta federada, se debe de consultar a todos los <i>triple store</i> extraídos de la consulta federada para obtener respuestas con base a los argumentos también extraídos de la consulta.
Requerimiento funcional	no RNF01
Prioridad del requisito	Alta/Esencial

Tabla 23 Requerimiento funcional procesar resultados.

Número de requisito	RF20
Nombre de requisito	Procesar resultados
Tipo	Obligatorio
Características	Los resultados retornados por los diferentes <i>triple store</i> son unificados.
Descripción del requerimiento	El módulo desarrollado procesa todas las respuestas de cada <i>triple store</i> consultado con el fin de eliminar resultados repetidos y devolver una respuesta unificada.
Requerimiento funcional	no RNF01
Prioridad del requisito	Alta/Esencial

Tabla 24 Requerimiento funcional guardar datos.

Número de requisito	RF21
Nombre de requisito	Guardar datos
Tipo	Obligatorio
Características	Los resultados de las consultas federadas se guardan en el <i>kiwi triple store</i> para que el usuario pueda usarlos posteriormente.
Descripción del requerimiento	Por cada consulta federada exitosa, Apache Marmotta guarda el resultado de la consulta en el <i>triple store</i> de Marmotta, <i>kiwi triple store</i> , para que pueda posteriormente hacer uso de los datos de su consulta.
Requerimiento funcional	no RNF01
Prioridad del requisito	Alta/Esencial

2.8 Requerimientos no funcionales

Tabla 25 Requerimiento no funcional Disponibilidad.

Número de requisito	RNF01
Nombre de requisito	Disponibilidad
Características	El software Apache Marmotta e Internet deben estar accesibles cuando un usuario de la la aplicación Web lo requiera.
Descripción del requerimiento	Cada vez que la aplicación Web necesite hacer uso de algún recurso de Apache Marmotta o de Internet, deben de estar disponibles para hacer uso de ellos.
Prioridad del requisito	Alta/Esencial

Tabla 26 Requerimiento no funcional Usabilidad.

Número de requisito	RNF02
Nombre de requisito	Usabilidad
Características	El usuario cuenta con alguna leyenda que proporcione información de cómo usar la aplicación Web o llevar a cabo una consulta.
Descripción del requerimiento	Debido a los requerimientos funcionales disponibles en la aplicación Web, se muestra información en la aplicación en donde sea pertinente de tal forma que le permita al usuario entender cómo debe de ser usada la aplicación. También ofrece ayuda para llevar a cabo una consulta federada en caso de que el usuario haya ingresado una consulta federada de manera incorrecta. De igual forma, la aplicación Web ayuda al usuario menos experto a construir una consulta federada mediante el despliegue de mensajes en la misma página que ofrezca ayuda al usuario, mejor conocido como <i>tooltips</i> .
Prioridad del requisito	Alta/Esencial

Tabla 27 Requerimiento no funcional Interfaz de la aplicación.

Número de requisito	RNF03
Nombre de requisito	Interfaz de la aplicación
Características	Todas las funcionalidades que ofrece la aplicación Web se podrán acceder desde la interfaz visual.
Descripción del requerimiento	La interfaz visual en la aplicación web permitirá al usuario usar la aplicación en conjunto con Apache Marmotta de manera sencilla.
Prioridad del requisito	Alta/Esencial

Tabla 28. Requerimiento no funcional Confidencialidad

Número de requisito	RNF04
Nombre de requisito	Confidencialidad
Características	La seguridad de los datos proporcionados por los usuarios está garantizada en el sistema.
Descripción del requerimiento	La información ofrecida por los usuarios para darse de alta en el sistema está segura para que nadie más pueda hacer uso de ella.
Prioridad del requisito	Alta/Esencial

2.9 Interfaces de hardware

Para hacer uso de la aplicación Web es necesario tener equipo de cómputo con las siguientes especificaciones:

- 4 GB en memoria RAM.
- 500 MB de almacenamiento.
- Mouse.
- Teclado.
- Adaptador de Red.
- Monitor.
- Procesador doble núcleo o superior.

2.10 Interfaces de software

Para poder usar en la aplicación Web, se debe contar con las siguientes características:

- Sistema operativo Ubuntu.
- Java JDK 6 o superior.
- Servidor de aplicaciones Java (Tomcat 7.X o Jetty 6.X).
- Navegador Web (Google Chrome, Firefox, Edge, Safari u Opera).

2.11 Interfaces de comunicación

La comunicación entre la aplicación Web y Apache Marmotta es mediante métodos HTTP y el estilo de arquitectura de software es *SOA* y *REST*. Mientras que, para hacer uso de la aplicación Web, se necesita conexión a Internet.

3 Capítulo V: Diseño del sistema

En esta sección se especifican las características tanto de la aplicación Web y del módulo de consultas federadas geoespaciales a desarrollar mediante diagrama de casos de uso, diagrama de clases, diagrama de estados y diagramas de secuencia. Al final de este capítulo se muestran *mockups* de la interfaz de usuario para la aplicación Web.

3.1 Caso de uso

Los diagramas de caso de uso sirven para especificar el comportamiento esperado de un sistema sin ahondar cómo se llevará a cabo. El uso de este tipo de diagramas permite representar a nivel visual y textual el diseño del sistema desde la perspectiva del usuario final.

En los diagramas de caso de uso representa la relación existente entre sistemas, usuarios y actores, pero no muestra el orden de cómo es que será ejecutado cada caso de uso.

El siguiente es el único caso de uso, la cual es para la aplicación Web. Es el único ya que es la sección del proyecto que tiene una interacción directa con los usuarios finales. El caso de uso para la aplicación Web es la mostrada en la figura 17.

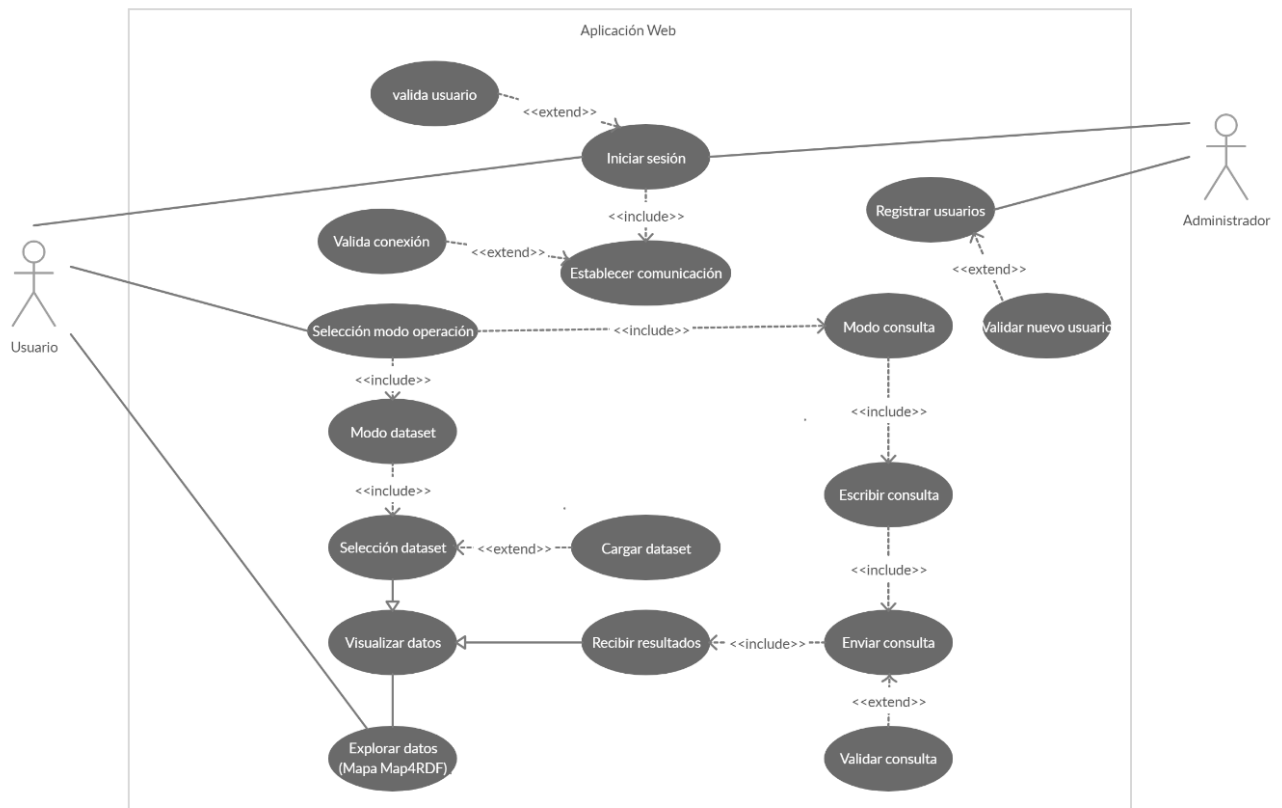


Figura 8 Caso de uso del diseño del sistema.

3.2 Diagramas de clase

Los diagramas de clase son diagramas estáticos de estructura de un sistema que permite visualizar y construir sistemas orientados a objetos el cual muestra sus clases, métodos, atributos además de la relación que existen entre los objetos.

A continuación, en la figura 18, se muestra el diagrama de clases para la aplicación Web. En ella se maneja la herencia de una clase, la cual es “Usuario” que generaliza a dos tipos de usuario: normal y administrador. Cada uno de ellos tiene sus métodos propios; el usuario “Estudiante” posee el atributo *nivel_usuario* que le permite al sistema reconocer si el usuario que esté haciendo uso del sistema es un usuario experto o no. La escala va del 0 al 5 donde el cero indica que el usuario no tiene conocimiento de cómo construir una consulta federada geoespacial y cinco, que indica que el usuario está completamente familiarizado con el desarrollo la consulta. Sin embargo, ambos comparten los atributos de correo y contraseña, y el método iniciar sesión que sirven para acceder a los elementos de la aplicación Web.

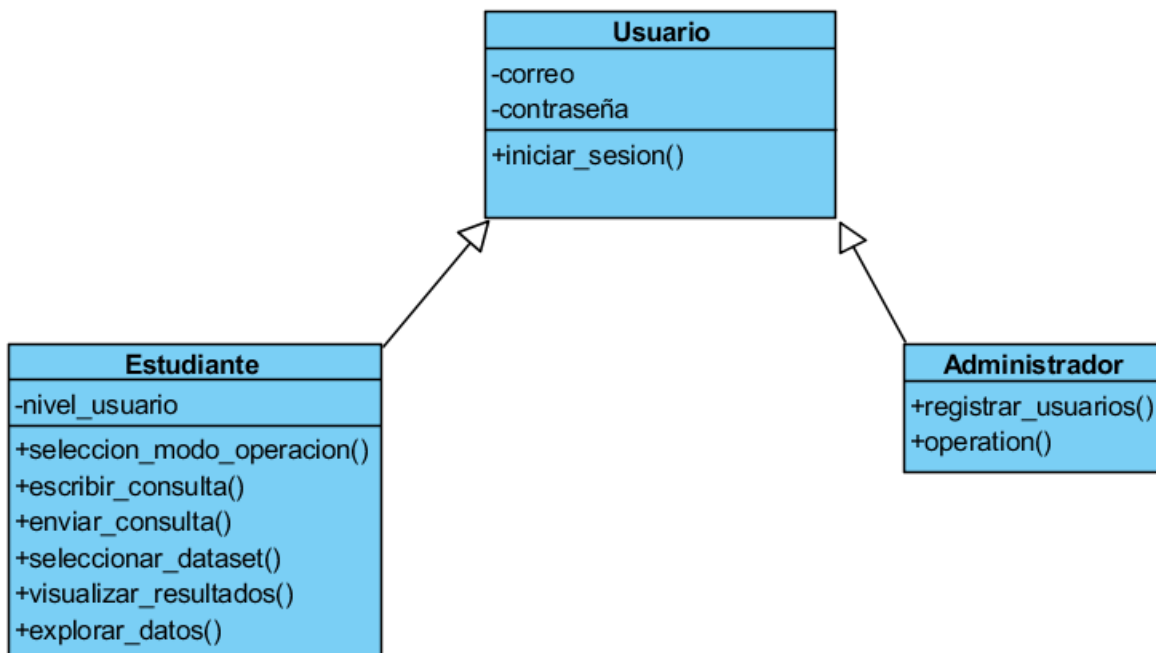


Figura 9 Diagrama de clases.

Cabe resaltar que, al igual que el diagrama de casos de uso, solo existe el diagrama de clase para la aplicación Web ya que Apache Marmotta no necesita tener información almacenada dentro de sus sistema sobre el usuario que realiza alguna operación en él.

3.3 Diagramas de estado

Los diagramas de estado son un tipo de diagrama que permite representar los diversos estados de una entidad no solo como una consecuencia de las entradas sino también de sus estados previos. De igual forma, muestra como un sistema responde a diferentes eventos dependiendo del cambio de un estado a otro.

A diferencia de los diagramas pasados, en esta sección se muestran los diagramas de estado que existen en la aplicación Web y en Apache Marmotta con el módulo de consultas federadas geoespaciales que se implementó.

3.3.1 Diagrama de estado para la aplicación web – Usuario

En la figura 19 se muestra el diagrama de estados presente en la aplicación Web. Con este diagrama se puede ver cómo el usuario común puede avanzar en la aplicación considerando el inicio de sesión, el selección de modo de operación y la visualización de datos.

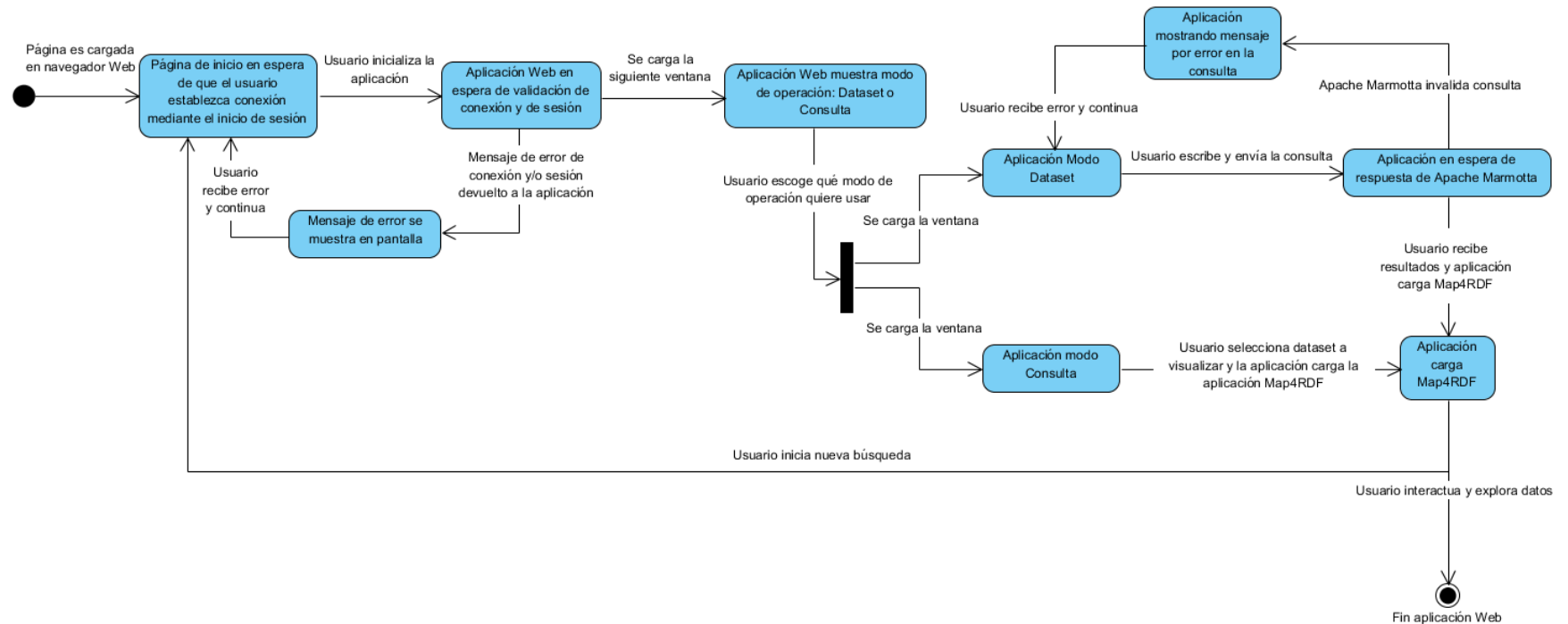


Figura 10 Diagrama de estados - aplicación Web – usuario.

3.3.2 Diagrama de estado para la aplicación web – Administrador

El diagrama de estados para el usuario administrador, la cual se muestra en la figura 20, detalla los estados de la aplicación Web cuando el usuario administrador está usándola.

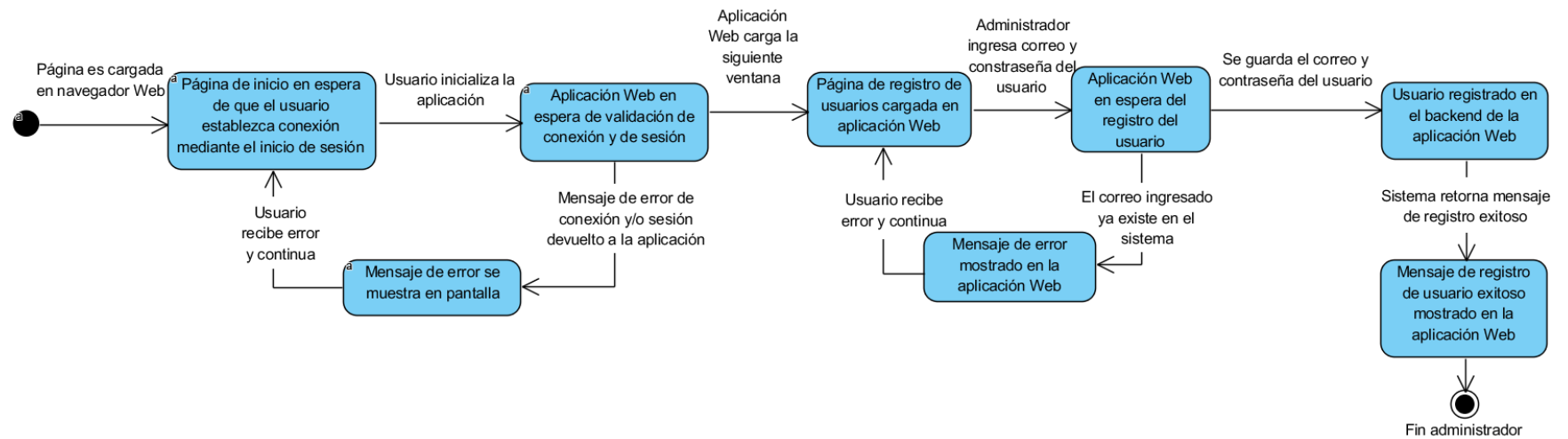


Figura 11 Diagrama de estados - Aplicación Web Administrador.

3.3.3 Diagrama de estado para módulo de consultas en Apache Marmotta

Para el módulo de consultas federadas geoespaciales que se implementó en Apache Marmotta, se diseñó el siguiente diagrama de estados que propone cómo es que el módulo reacciona a las entradas y a los estados previos. Si bien el bloque azul es el módulo que se desarrolló, los demás objetos también son elementos por considerar ya que hay que modificar el respectivo elemento de Apache Marmotta que procesa la consulta para que sea capaz de identificar si es una consulta federada o no (Figura 21).

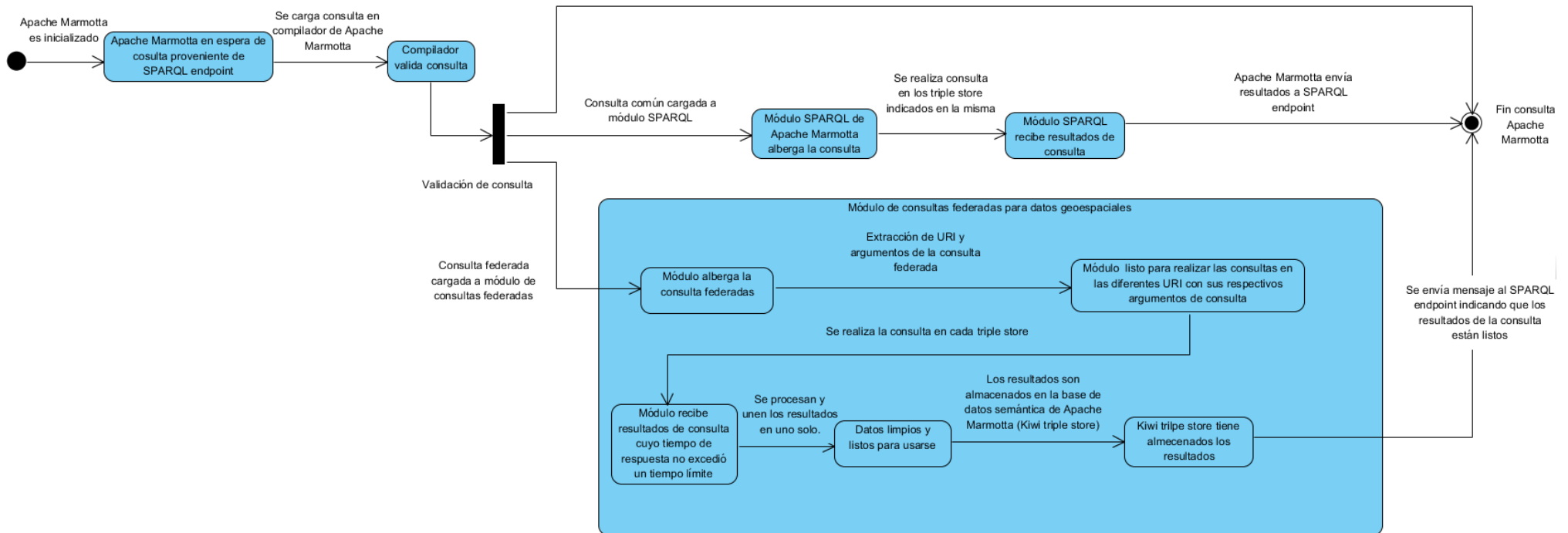


Figura 12 Diagrama de estados - Módulo de consultas federadas.

3.4 Diagramas de secuencia

Los diagramas de secuencia son un tipo de diagramas de interacción que especifica como los objetos interactúan entre sí. En estos diagramas se observan las interacciones que existen entre los objetos involucrados en el sistema mediante mensajes, que van de un objeto a otro o a sí mismos, y columnas que representan el tiempo que los objetos están presentes en el sistema. Cabe decir que a pesar de que estos diagramas involucran el tiempo, no implica la especificación de cuánto tiempo debe de existir entre mensajes.

3.4.1 Diagrama de secuencia para aplicación web (Usuario).

En el siguiente diagrama de secuencia, figura 22, muestra la interacción que existe entre el usuario común y la aplicación Web. Con la ayuda del diagrama de casos de uso, se visualiza como es que el usuario puede navegar en la aplicación Web desde el inicio de sesión hasta la visualización de datos.

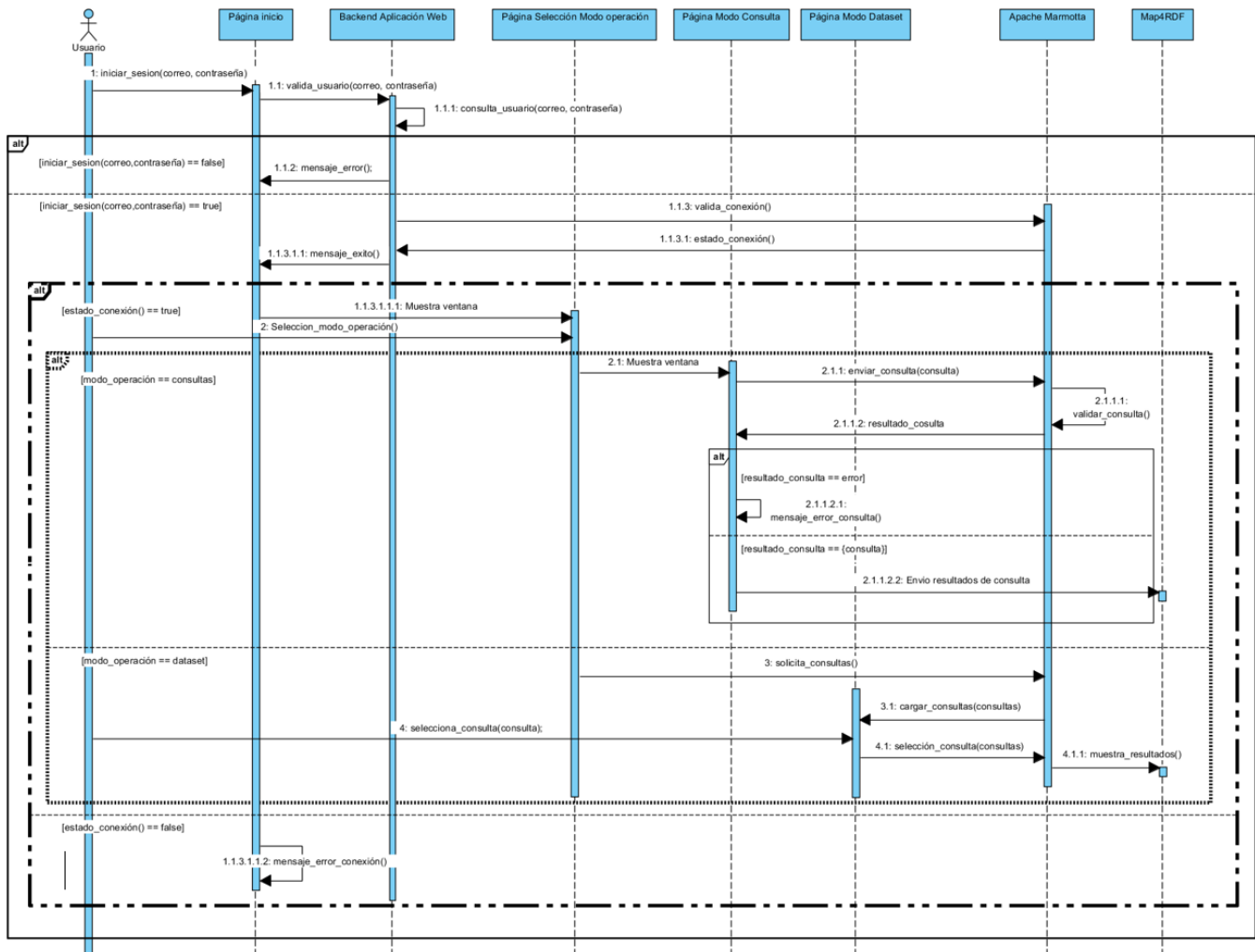


Figura 13 Diagrama de secuencia - aplicación Web – usuario común.

Con el fin de mostrar a mayor detalle el diagrama de secuencia mostrado en la figura 23, en las siguientes figuras se muestra el mismo diagrama de secuencia, pero de manera separada mostrando la referencia al siguiente diagrama.

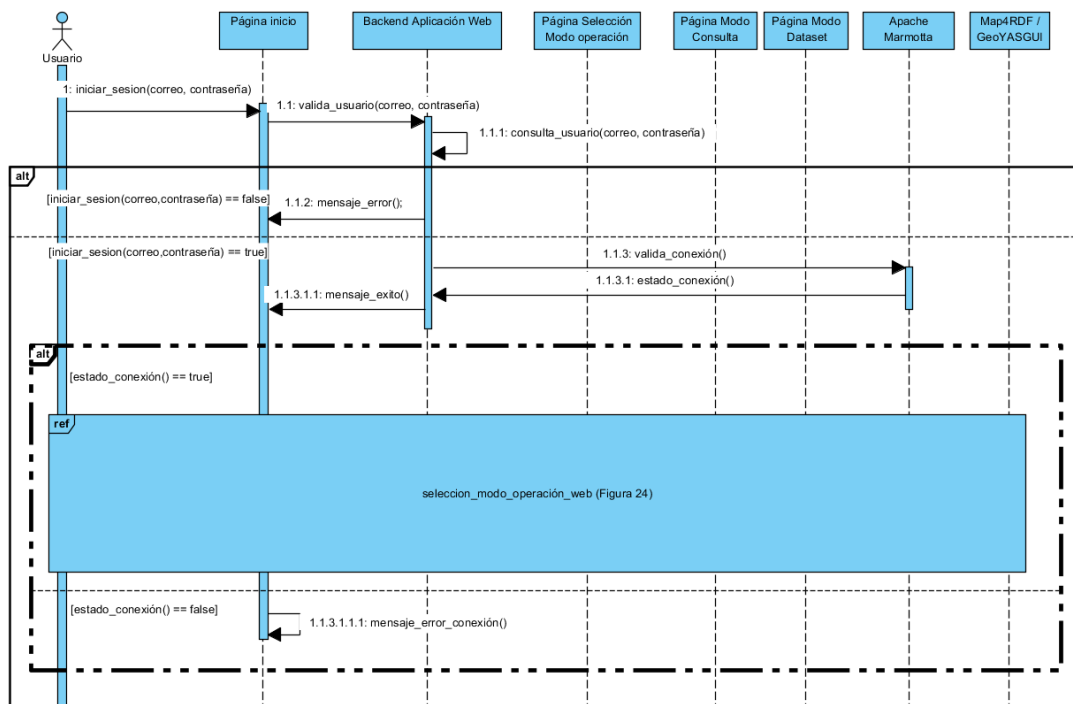


Figura 14 Diagrama de secuencia - Aplicación Web - usuario común.

En la figura 24 se encuentra el diagrama de secuencia que corresponde a la interacción del usuario con la aplicación Web al seleccionar un modo de operación.

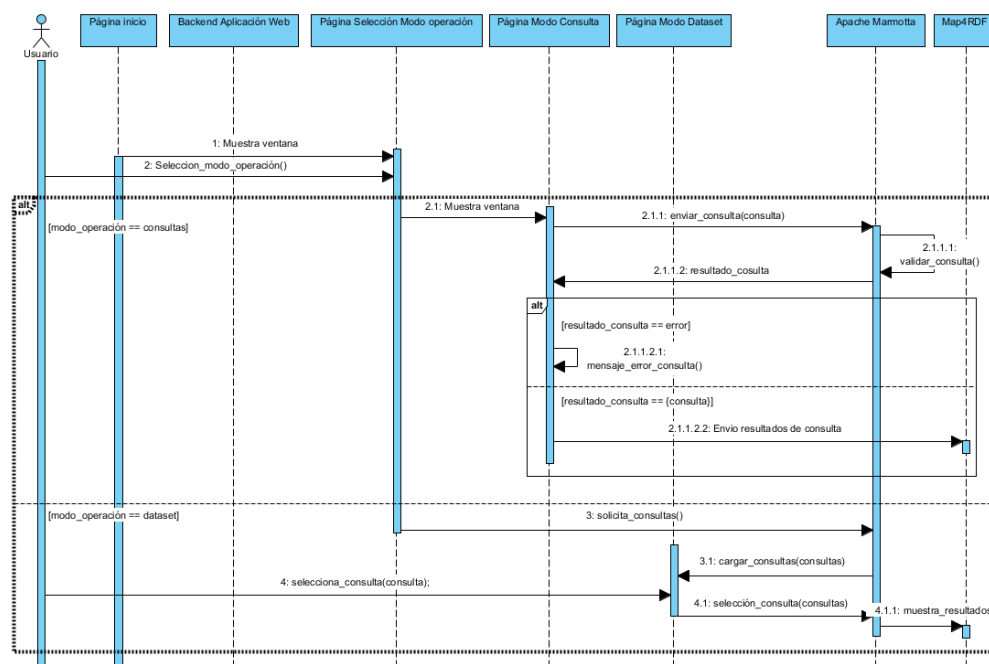


Figura 15 Diagrama de secuencia - Aplicación Web - usuario común, continuación.

3.4.2 Diagrama de secuencia para aplicación Web (Administrador)

Para el administrador de la aplicación Web se tiene el diagrama de secuencia en la figura 25. El diagrama muestra el proceso del administrador para iniciar sesión y para registrar a un nuevo estudiante.

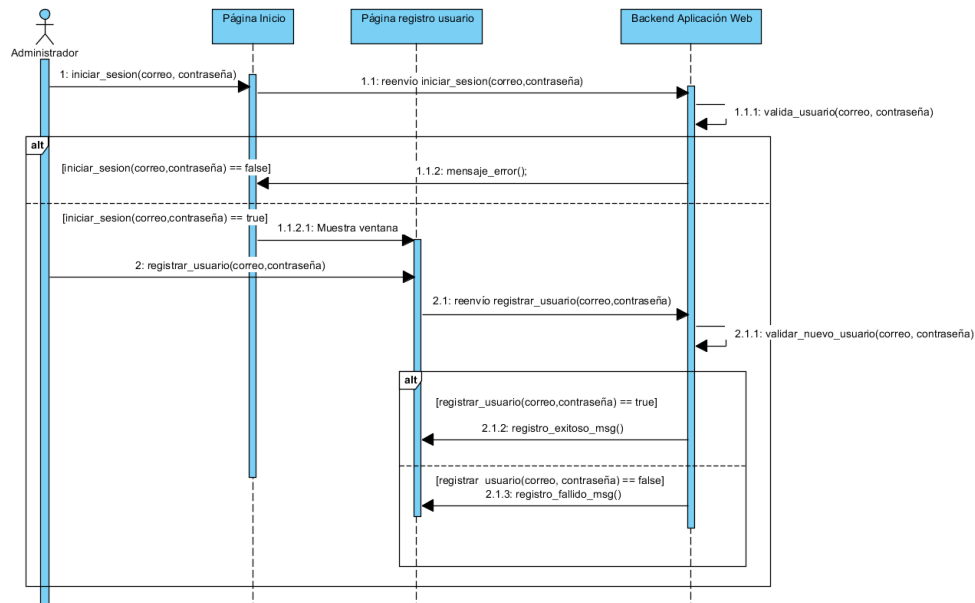


Figura 16 Diagrama secuencia aplicación Web - Administrador.

3.4.3 Diagrama de secuencia para el módulo.

La figura 26 muestra el diagrama de secuencia que el módulo de consultas federadas posee para llevar a cabo una consulta. Auxiliándose de su respectivo diagrama de estados, este diagrama detalla cómo es que se lleva a cabo el proceso.

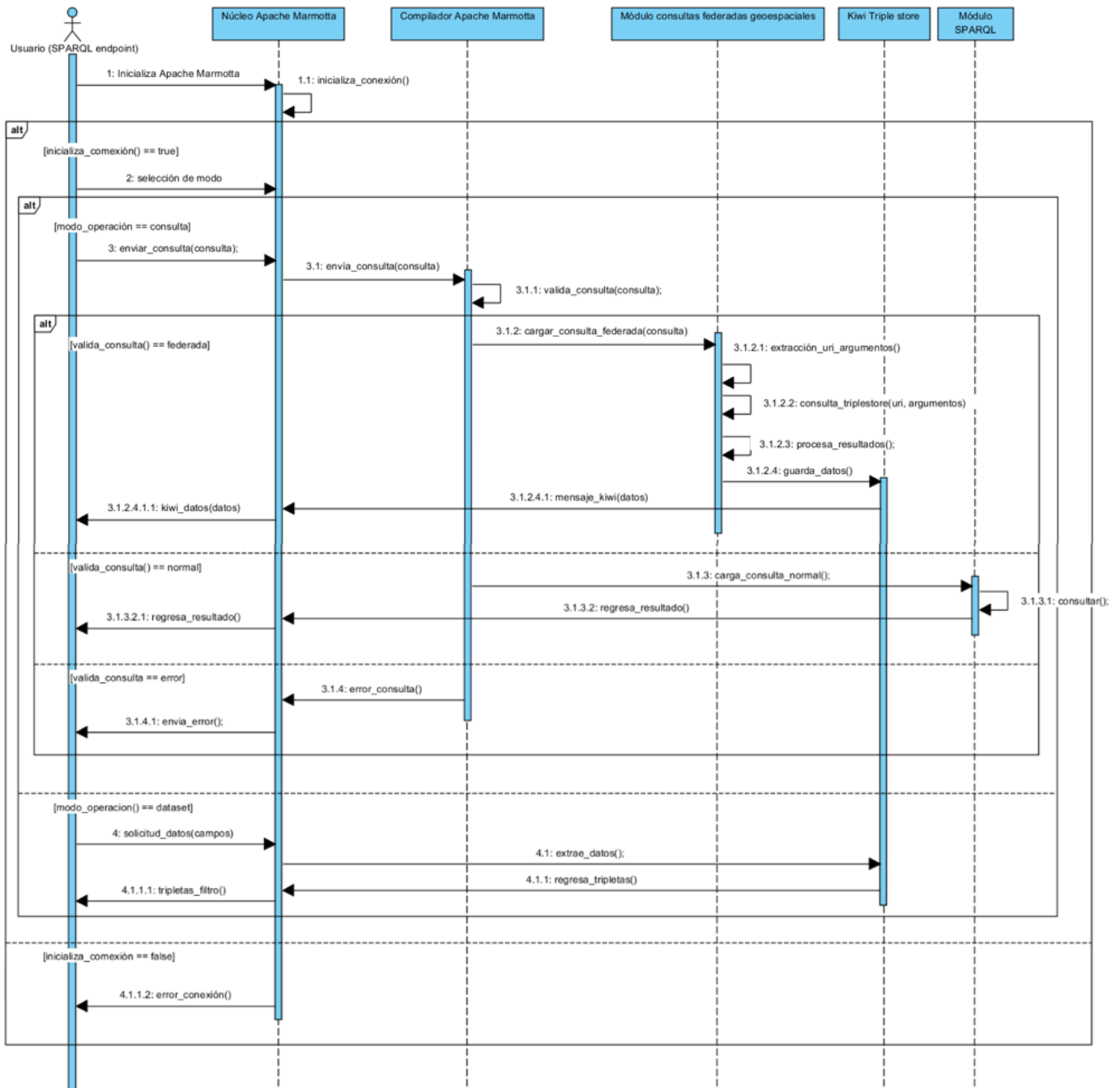


Figura 17 Diagrama de secuencia para módulo de consultas federadas geoespaciales

Con el objetivo de mostrar a detalle la figura 27, se muestran a continuación tres figuras que particularizan el diagrama de secuencias por partes.

La figura 23 muestra el diagrama de secuencia para establecer conexión con Apache Marmotta.

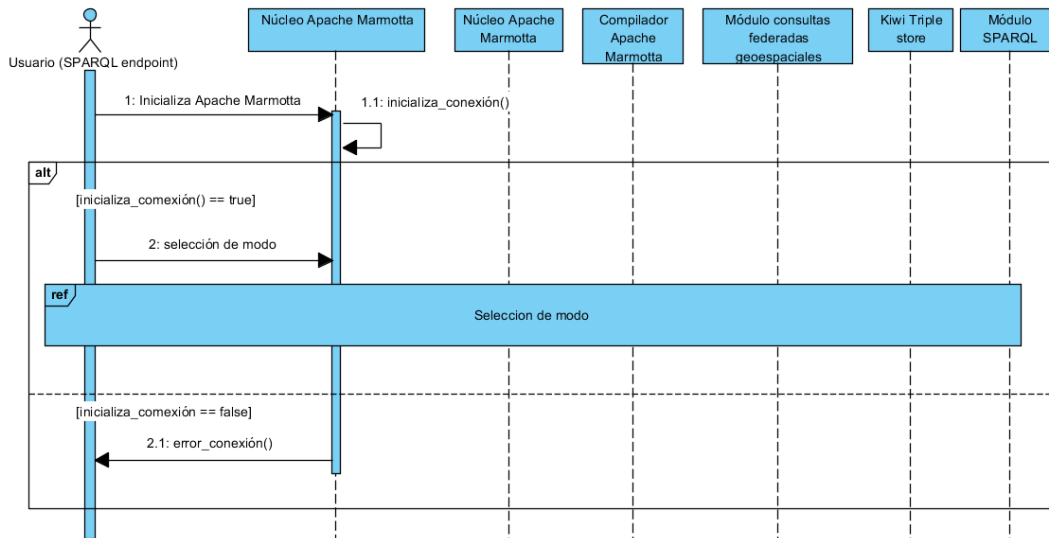


Figura 18 Diagrama secuencia inicio sesión.

La figura 28 muestra el diagrama de secuencia correspondiente a la acción de selección de modo en proveniente de la aplicación Web.

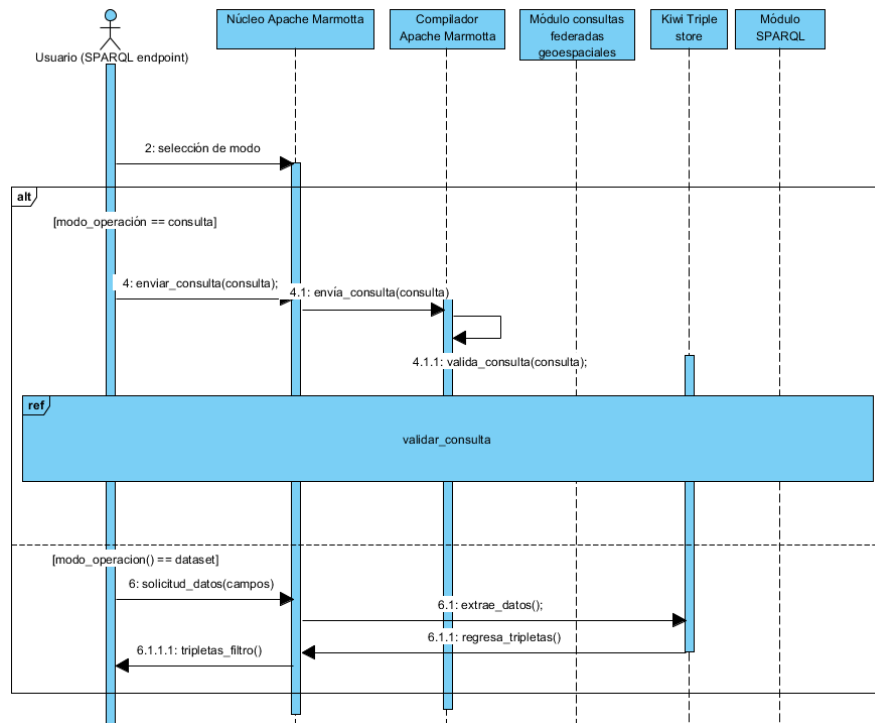


Figura 19 Diagrama de secuencia - selección de modo.

La figura 29 muestra el diagrama de secuencia correspondiente al flujo de la aplicación después de que Apache Marmotta reciba y valide la consulta ingresada por el usuario.

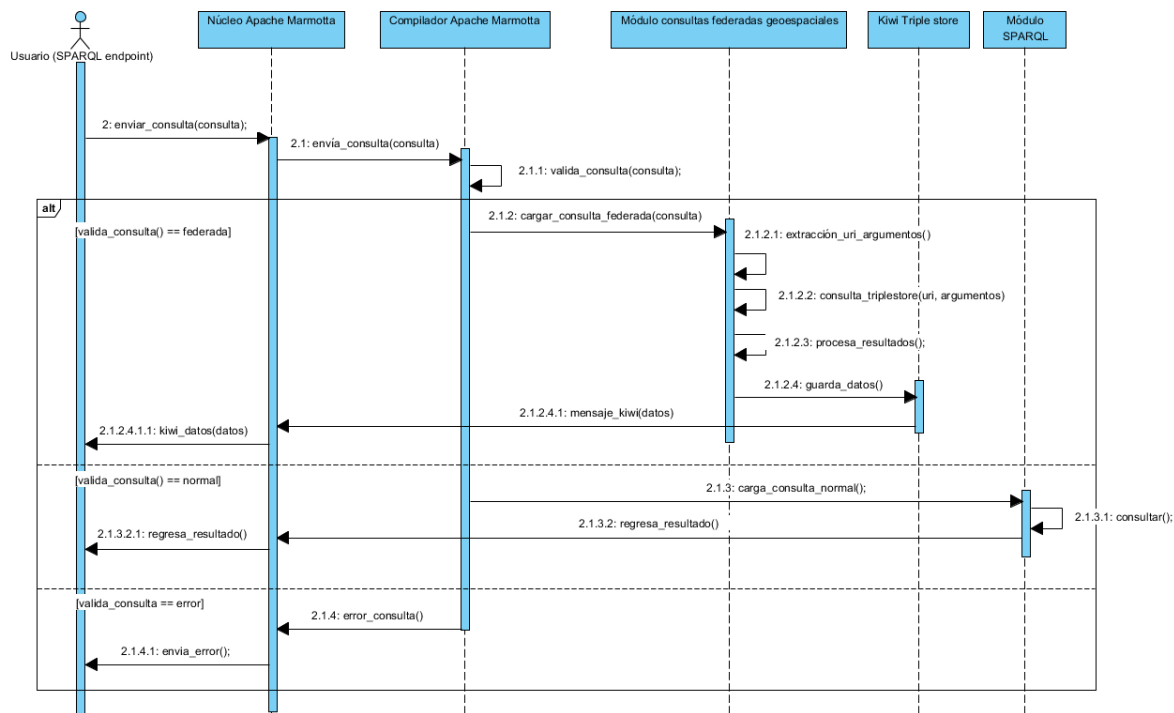


Figura 20 Diagrama de secuencia - validación consulta.

3.5 Secuencia de interfaces

A continuación, se muestra la secuencia de interfaces, *mockups*, que tiene la aplicación Web tanto para el usuario común como para el administrador. Ambos usuarios comparten una pantalla y es la de inicio de sesión, pero para el usuario administrador solo hay una pantalla y es la de registro de usuarios mientras que para el usuario común hay cuatro páginas por la que es usuario puede navegar.



4 Referencias

- [1] W3C, «Semantic Web,» 9 Marzo 2019. [En línea]. Available: <https://www.w3.org/standards/semanticweb/>. [Último acceso: 18 Marzo 2019].
- [2] W3C, «LinkedData,» 1 Agosto 2016. [En línea]. Available: <https://www.w3.org/wiki/LinkedData>. [Último acceso: 18 Marzo 2019].
- [3] W3C, «RDF,» 15 Marzo 2014. [En línea]. Available: <https://www.w3.org/RDF/>. [Último acceso: 18 Marzo 2019].
- [4] W3C, «URI,» 1 Febrero 2005. [En línea]. Available: <https://www.w3.org/wiki/URI>. [Último acceso: 18 Marzo 2019].
- [5] Ontotext, «What is RDF Triple Store?,» 21 Noviembre 2016. [En línea]. Available: <https://www.ontotext.com/knowledgehub/fundamentals/what-is-rdf-triplestore/>. [Último acceso: 20 Febrero 2019].
- [6] W3C, «SPARQL 1.1 Overview,» 21 Marzo 2013. [En línea]. Available: <https://www.w3.org/TR/sparql11-overview/>. [Último acceso: 18 Marzo 2019].
- [7] DBpedia, «Virtuoso SPARQL Query Editor,» 17 Abril 2018. [En línea]. Available: <https://dbpedia.org/sparql?help=intro>. [Último acceso: 20 Febrero 2019].
- [8] DBpedia, «Learn about DBpedia,» DBpedia, 24 Octubre 2019. [En línea]. Available: <https://wiki.dbpedia.org/about>. [Último acceso: 10 Noviembre 2019].
- [9] OGC, «GeoSPARQL - A Geographic Query Language for RDF Data,» 10 Septiembre 2012. [En línea]. Available: https://portal.opengeospatial.org/files/?artifact_id=47664. [Último acceso: 24 Abril 2019].
- [10] R. Battle y D. Kolas, «Linking Geospatial Data With GeoSPARQL,» de *Semant Web J Interoperability, Usability, Appl. Accessed*, vol. 24, Arlington, 2011.
- [11] D. K. Becker, *Information Quality & Service Oriented Architecture*, Massachusetts: MIT, 2007.
- [12] T. Berners Lee, R. Fielding, J. Gettys, J. Mogul, H. Frystyk y L. Masinter, *Hypertext Transfer Protocol -- HTTP/1.1*, W3C/MIT, 1999.
- [13] M. Developers, «MDN web docs,» 15 07 2019. [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/HTTP/Methods>. [Último acceso: 29 09 2019].
- [14] Codecademy, «Codecademy,» 12 08 2019. [En línea]. Available: <https://www.codecademy.com/articles/what-is-rest>. [Último acceso: 29 09 2019].

- [15] I. E. T. F. (IETF), The JavaScript Object Notation (JSON) Data Interchange Format, T. Bray, 2017.
- [16] J. H. Alejandro, Apuntes de compiladores, Ciudad de México: UNAM, 1980.
- [17] IEEE, Especificación de Requisitos de Software IEEE 830-1998, Standards IEEE, 1998.
- [18] T. Berners Lee, J. Hendler y O. Lassila, «The Semantic Web,» *Scientific American*, vol. 284, nº 5, pp. 34-43, 2001.
- [19] C. Bizer, T. Heath y T. Berners-Lee, «Linked Data - The Story So Far,» de *Semantic services, interoperability and web applications: emerging concepts*, USA, Information Science Reference, 2009, pp. 205-227.
- [20] L. Lupercio, F. Baculima, M. Espinoza y V. Saquicela, «Explotación de información en el dominio geo-hídrico ecuatoriano utilizando tecnología semántica,» *Maskana*, vol. 6, pp. 69-77, 2015.
- [21] L. M. Vilches Blázquez y J. Saavedra, «A framework for connecting two interoperability universes: OGC Web Feature Services and Linked Data,» *Transactions in GIS*, vol. 23, nº 1, pp. 22-47, 2018.
- [22] R. Klischewski, «Semantic Web for e-Government,» de *International Conference on Electronic Government*, Heidelberg, 2003.
- [23] Apache, «Apache Marmotta,» 15 Febrero 2019. [En línea]. Available: <http://marmotta.apache.org/>. [Último acceso: 20 Febrero 2019].
- [24] O. E. ingGroup, «MAP4RDF,» 2015. [En línea]. Available: <http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/downloads/172-map4rdf/index.html>. [Último acceso: 24 Abril 2019].
- [25] W. Beek, W. Folmer, L. Rietveld y J. Walker, «Geoyasgui: The GeoSPARQL query editor and result visualizer,» *The international Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, p. 39, 2017.
- [26] A. Marmotta, «Apache Marmotta Platform: SPARQL,» 30 Abril 2014. [En línea]. Available: <http://marmotta.apache.org/platform/sparql-module.html>. [Último acceso: 24 Abril 2019].
- [27] W3C, «Category: Triple Store,» 5 Noviembre 2013. [En línea]. Available: https://www.w3.org/2001/sw/wiki/Category:Triple_Store. [Último acceso: 20 Febrero 2019].
- [28] O. G. Consortium, «GeoSPARQL - A Geographic Query Language for RDF Data,» 7 Julio 2001. [En línea]. Available: <https://www.opengeospatial.org/standards/geosparql>. [Último acceso: 18 Marzo 2019].

- [29] M. Shmidt, O. Görlitz, P. Haase, G. Ladwig, A. Shwarte y T. Tran, «FedBench: A Benchmark Suite for Federated Semantic Data Query Processing,» de *International Semantic Web Conference*, Berlín, 2011.
- [30] M. Schmidt, T. Hornung, G. Lausen y C. Pinkel, «SP² Bench: a SPARQL performance benchmark,» de *2009 IEEE 25th International Conference on Data Engineering*, Freiburg, Alemania, 2009.
- [31] C. Bizer y A. Schultz, «The berlin sparql benchmark,» *International Journal on Semantic Web and Information Systems*, vol. 5, nº 2, pp. 1-24, 2009.
- [32] A. Fuggetta, «Open source software—an evaluation,» *Journal of Systems and Software*, vol. 66, nº 1, pp. 77-90, 2003.
- [33] N. Wiegand, R. Grove, J. Wilson y D. Kolas, «Querying Geospatial Data over the Web: a GeoSPARQL Interface,» de *Proceedings of Workshop on Managing and Mining Enriched Geo-Spatial Data*, Virginia, 2014.
- [34] R. Battle y D. Kolas, «Enabling the Geospatial Semantic Web with Parliament and GeoSPARQL,» *Semantic Web*, vol. 3, nº 4, pp. 355-370, 2012.
- [35] C. Buil-Aranda, A. Polleres y J. Umbrich, «Strategies for Executing Federated Queries in SPARQL1.1,» de *International Semantic Web Conference*, Chile, 2014.
- [36] J. Sheridan y J. Tenninson, «Linking UK Government Data,» de *Ldow*, Reino Unido, 2010.
- [37] T. Zhao, C. Zhang, L. Anselin, W. Li y K. Chen, «A parallel approach for improving Geo-SPARQL query performance,» *International Journal of Digital Earth*, vol. 8, nº 5, pp. 383 - 402, 2015.
- [38] M. Morsey, J. Lehmann, S. Auer y A. C. Ngonga Ngomo, «DBpedia SPARQL Benchmark – Performance Assessment with Real Queries on Real Data,» de *International semantic web conference*, Berlin, Heidelberg, 2011.
- [39] N. Charlampos, D. Kallirroi, K. Kostis y K. Manolis, «Sextant: Browsing and Mapping the Ocean of Linked Geospatial Data,» de *Extended Semantic Web Conference*, Grecia, 2013.
- [40] K. Bereta, G. Xiao y M. Koubarakis, «ANSWERING GEOSPARQL QUERIES OVER RELATIONAL DATA,» *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 42, 2017.
- [41] G. A. Atemezing y F. Amardeilh , «Benchmarking Commercial RDF stores with Publications Office Dataset,» de *European Semantic Web Conference*, 2018.
- [42] G. Garbis, K. Kyzirakos y M. Koubarakis, «Geographica: A Benchmark for Geospatial RDF Stores,» de *International Semantic Web Conference*, Berlin, 2013.

- [43] E. Ávila Barrientos, «Linked Open Data en la Biblioteca Digital,» de *Biblioteca Digital Académica en Bibliotecología y Estudios de la Información*, México, 2013, pp. 137-152.
- [44] A. Becerril García, R. Lozano Espinosa y J. M. Molina Espinosa, «Enfoque semántico para el descubrimiento de recursos sensible al contexto sobre contenidos académicos estructurados con OAI-PMH,» *Computación y sistemas*, vol. 20, nº 1, pp. 127-142, 2016.
- [45] R. Zárate Escobedo, *Facilitador de contenido móvil para el viajero basado en servicios de localización y web semántica*, México, 2018.
- [46] D. F. Rojas Carrasco y R. Torres Covarrubias, *Recuperación de información geográfica utilizando similitud semántica*, México, 2009.