



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**SISTEMA GEOREFERENCIAL PARA UN ESTUDIO DE AVES EN EL
DEPARTAMENTO DE IZABAL APLICANDO UN SISTEMA DE GESTIÓN DEL
CONOCIMIENTO**

Ana Luisa Muñoz Tello

Evelyn Silvana Recinos Herrera

Asesorado por el Ing. Javier Gramajo

Guatemala, mayo de 2011

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**SISTEMA GEOREFERENCIAL PARA UN ESTUDIO DE AVES EN EL
DEPARTAMENTO DE IZABAL APLICANDO UN SISTEMA DE GESTIÓN DEL
CONOCIMIENTO**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

ANA LUISA MUÑOZ TELLO

EVELYN SILVANA RECINOS HERRERA

ASESORADO POR EL ING. JAVIER GRAMAJO

AL CONFERÍRSELES EL TÍTULO DE

INGENIERAS EN CIENCIAS Y SISTEMAS

GUATEMALA, MAYO DE 2011

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

| | |
|------------|-------------------------------------|
| DECANO | Ing. Murphy Olympo Paiz Recinos |
| VOCAL I | Ing. Alfredo Enrique Beber Aceituno |
| VOCAL II | Ing. Pedro Antonio Aguilar Polanco |
| VOCAL III | Ing. Miguel Ángel Dávila Calderón |
| VOCAL IV | Br. Luis Pedro Ortiz de León |
| VOCAL V | P.A. José Alfredo Ortiz Herincx |
| SECRETARIO | Ing. Hugo Humberto Rivera Pérez |

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

| | |
|------------|---------------------------------|
| DECANO | Ing. Murphy Olympo Paiz Recinos |
| EXAMINADOR | Ing. Juan Álvaro Díaz Ardavin |
| EXAMINADOR | Ing. César Fernández Cáceres |
| EXAMINADOR | Ing. Pedro Pablo Hernández |
| SECRETARIO | Ing. Hugo Humberto Rivera Pérez |

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presentamos a su consideración nuestro trabajo de graduación titulado:

SISTEMA GEOREFERENCIAL PARA UN ESTUDIO DE AVES EN EL DEPARTAMENTO DE IZABAL APLICANDO UN SISTEMA DE GESTIÓN DEL CONOCIMIENTO

Tema que nos fuera asignado por la coordinación de la Carrera de Ingeniería en Ciencias y Sistemas, con fecha enero de 2011.

Ana Luisa Muñoz Tello

Evelyn Silvana Recinos Herrera

AGRADECIMIENTOS A:

| | |
|---|---|
| Universidad de San Carlos de Guatemala | Por ser la casa de estudios que nos permitió tener acceso a la educación superior. |
| Facultad de Ingeniería | Por fomentar en sus aulas valores y brindar los sabios conocimientos que nos forman como profesionales. |
| Ing. Javier Gramajo | Por su apoyo y asesoría durante el desarrollo de este proyecto. |
| Dr. Jean Luc | Por el valioso aporte para la implementación de este trabajo. |

ACTO QUE DEDICO A:

- Dios** Por haberme permitido culminar mis estudios y ser fuente de regocijo.
- Mi madre** Hasmin Tello por el esfuerzo y dedicación que me ha dado a lo largo de mi vida.
- Mis hermanos** Luis, Evelyn y Olga por estar incondicionalmente en todo momento.
- Mi sobrino** José Carlos por su paciencia y amor.
- Mis amigos** Evelyn y Amilcar.
- En especial** Oscar Valdez, por ser parte de este triunfo y haberme apoyado siempre.

Ana Luisa Muñoz Tello

ACTO QUE DEDICO A:

Dios Por ser mi fuerza, mi guía y mi inspiración en todo momento.

Mi señora madre Rubí Herrera, quien por medio de su ejemplo de amor, valor y perseverancia, me enseñó lo importante que es desarrollarme tanto espiritual como profesionalmente.

Mi abuelita Olga Cristina, por todo su apoyo y amor.

Mis hermanos José y Paola, por tener su apoyo incondicional.

Otto Raúl Padilla Por apoyarme a lo largo de la carrera.

Mis tíos y primos Quienes siempre me han brindado sus consejos y apoyo.

Mis amigos Ana Luisa, Deisy, Nancy, Cindy, William, Melvin, Gabriel, Fredy y Saulo, por todo su apoyo.

Evelyn Silvana Recinos Herrera

ÍNDICE GENERAL

| | |
|---|-----|
| ÍNDICE DE ILUSTRACIONES | I |
| GLOSARIO | III |
| RESUMEN | V |
| OBJETIVOS Y/O HIPÓTESIS | VII |
| INTRODUCCIÓN | IX |
| | |
| 1. <i>WEB SEMÁNTICA</i> | 1 |
| 1.1 <i>Arquitectura de la web semántica</i> | 3 |
| 1.1.1 <i>Unicode</i> | 3 |
| 1.1.2 <i>URI</i> | 3 |
| 1.1.3 <i>XML + NS + XML Schema</i> | 3 |
| 1.1.4 <i>RDF + RDF Schema</i> | 3 |
| 1.2 <i>Lenguaje de ontologías</i> | 4 |
| 1.2.1 <i>Lógica</i> | 4 |
| 1.2.2 <i>Pruebas, confianza y firma digital</i> | 4 |
| 1.3 <i>Ontologías</i> | 4 |
| 1.3.1 <i>Usos de la ontología</i> | 5 |
| 1.3.1.1 <i>Comunicación entre personas,</i> <i>organizaciones y aplicaciones</i> | 5 |
| 1.3.1.2 <i>Interoperabilidad de sistema</i> | 5 |
| 1.3.1.3 <i>Razonamiento automático</i> | 6 |
| 1.3.2 <i>Componentes de una ontología</i> | 6 |

| | | |
|----------|---------------------------------|----|
| 2. | ANÁLISIS Y DISEÑO | 9 |
| 2.1 | Descripción caso de uso general | 9 |
| 2.2 | Especificación casos de uso | 11 |
| 2.2.1. | Realizar consulta por especie | 11 |
| 2.2.1.1. | Descripción | 11 |
| 2.2.1.2. | Objetivo | 11 |
| 2.2.1.3. | Flujo básico | 12 |
| 2.2.1.4. | Flujo alternativo | 12 |
| 2.2.1.6 | Prerrequisito | 12 |
| 2.2.1.7 | Riesgo | 12 |
| 2.2.2. | Realizar consulta por género | 13 |
| 2.2.2.1 | Descripción | 13 |
| 2.2.2.2 | Objetivo | 13 |
| 2.2.2.3 | Flujo básico | 14 |
| 2.2.2.4 | Flujo alternativo | 14 |
| 2.2.2.5 | Prerrequisito | 14 |
| 2.2.2.6 | Riesgo | 14 |
| 2.2.3. | Realizar consulta por estado | 15 |
| 2.2.3.1. | Descripción | 15 |
| 2.2.3.2. | Objetivo | 15 |
| 2.2.3.3. | Flujo básico | 16 |
| 2.2.3.4. | Flujo alternativo | 16 |
| 2.2.3.5. | Prerrequisito | 16 |
| 2.2.3.6. | Riesgo | 16 |
| 2.2.4. | Realizar consulta por estatus | 17 |
| 2.2.4.1. | Descripción | 17 |
| 2.2.4.2. | Objetivo | 17 |
| 2.2.4.3. | Flujo básico | 18 |
| 2.2.4.4. | Flujo alternativo | 18 |

| | | |
|----------|--------------------------------|----|
| 2.2.4.5. | Prerrequisito | 18 |
| 2.2.4.6. | Riesgo | 18 |
| 2.2.5. | Ingresar tipo de consulta | 19 |
| 2.2.5.1. | Descripción | 19 |
| 2.2.5.2. | Objetivo | 19 |
| 2.2.5.3. | Flujo básico | 20 |
| 2.2.5.4. | Flujo alternativo | 20 |
| 2.2.5.5. | Postrequisito | 20 |
| 2.2.5.6. | Riesgo | 20 |
| 2.3. | Diagramas | 21 |
| 3. | CONSULTAS SPARQL | 27 |
| 3.1 | Definición | 27 |
| 3.2 | Sintaxis | 27 |
| 3.3 | Especificación SPARQL | 28 |
| 4. | PROTOTIPO DE LA APLICACIÓN WEB | 31 |
| | CONCLUSIONES | 33 |
| | RECOMENDACIONES | 35 |
| | BIBLIOGRAFÍA | 37 |

ÍNDICE DE ILUSTRACIONES

FIGURAS

WEB SEMÁNTICA

- | | | |
|----|---|---|
| 1. | Arquitectura de la <i>Web Semántica</i> | 2 |
|----|---|---|

ANÁLISIS Y DISEÑO

- | | | |
|-----|-------------------------------|----|
| 2. | Diagrama casos de uso general | 9 |
| 3. | Diagrama caso de uso No. 1 | 11 |
| 4. | Diagrama caso de uso No. 2 | 13 |
| 5. | Diagrama caso de uso No. 3 | 15 |
| 6. | Diagrama caso de uso No. 4 | 17 |
| 7. | Diagrama caso de uso No. 5 | 19 |
| 8. | Diagrama de clases | 21 |
| 9. | Diagrama de componentes | 22 |
| 10. | Diagrama de <i>deployment</i> | 24 |
| 11. | Diagrama de secuencia | 25 |
| 12. | Modelo conceptual ontología | 26 |

PROTOTIPO DE LA APLICACIÓN *WEB*

- | | | |
|-----|---------------------|----|
| 13. | Página principal | 31 |
| 14. | Ejemplo de consulta | 32 |

GLOSARIO

| | |
|--------------------------------------|---|
| Actor | Entidad que utiliza los casos de uso del sistema. |
| Casos de Uso | Técnica utilizada para la captura de requisitos potenciales del nuevo sistema. |
| Diagrama de Casos de uso | Representación gráfica de parte o el total de los actores y casos de uso del sistema, incluyendo sus interacciones. |
| Diagrama de Clases | Tipo de diagrama estático que describe la estructura del sistema mostrando sus clases, atributos y las relaciones entre ellos. |
| Diagrama de Componentes | Muestra las organizaciones y dependencias lógicas entre componentes de <i>software</i> , sean éstos componentes de código fuente, binarios o ejecutables. |
| Diagrama de <i>Deployment</i> | Describe la arquitectura física del sistema durante la ejecución en términos de servidores, dispositivos y componentes de <i>software</i> . |

| | |
|------------------------------|---|
| Diagrama de Secuencia | Muestra las interacciones entre objetos ordenadas en secuencia temporal. Muestra los objetos que se encuentran en el escenario y la secuencia de mensajes intercambiados entre los objetos para llevar a cabo la funcionalidad descrita por el escenario. |
| <i>Exclude</i> | Relación de dependencia entre dos casos de uso que denota que un caso de uso es una especialización de otro. |
| <i>Include</i> | Relación de dependencia entre dos casos de uso que denota la inclusión del comportamiento de un escenario en otro. |
| Razonador | Herramienta que aprovecha el gran poder semántico de la ontología, de forma que, aplicándose sobre ésta el razonador o máquinas de inferencia pueda extraer conocimiento no expresado de forma explícita en su construcción permitiendo además realizar consultas sobre ella. |

RESUMEN

Este trabajo de graduación es una descripción formal del desarrollo de una aplicación *web* utilizando un sistema de gestión del conocimiento conocido como ontología, el cual permite al usuario realizar búsquedas inteligentes.

El uso de ontologías se lleva a cabo por medio de la definición de clases y las relaciones que existen entre ellas, permitiendo al usuario que al momento de realizar una consulta, le retorne las posibilidades más cercanas a lo que él desea encontrar. Para ello se utilizó *Protégé* que es una herramienta de desarrollo de ontologías que permite generar clases y establecer la relación de las mismas y ser exportada por medio de *OWL* para luego mapear a una base de datos la información, en este caso *MYSQL*.

El desarrollo de la aplicación se llevó a cabo por medio de *Java Netbeans*, en el cual se agregó la librería de *Protégé* conocida como *Jena* que permite la manipulación de los datos desde la aplicación, así como la generación de las consultas por medio del lenguaje conocido como *Sparql* y el uso de las librerías que proporciona *Google Maps* para georeferenciación.

Este trabajo de graduación presenta el análisis y diseño que se realizó para poder cumplir con los requerimientos del usuario, así como ejemplos del uso de las herramientas.

OBJETIVOS

General

Desarrollar una aplicación que sea capaz de georeferenciar los lugares en los cuales se ha observado presencia de las aves indicadas por el usuario.

Específicos

1. Diseñar una herramienta *web* que facilite la interoperabilidad entre los programas de *software* requeridos para el desarrollo de esta aplicación, como: *Protégé, Mysql, Sparql, Netbeans* y *Google Maps*.
2. Georeferenciar los sitios en los cuales se ha observado presencia de individuos, según la especie y género. Georeferenciar los sitios en los cuales se ha observado presencia de individuos, según la vulnerabilidad de la especie. Georeferenciar los sitios en los cuales se ha observado presencia de individuos, según el estatus y/o capacidad de migración de la especie.
3. Permitir al usuario interactuar con una aplicación amigable y fácil de utilizar.

INTRODUCCIÓN

Con el avance de la tecnología y la necesidad de los científicos e investigadores de obtener resultados exactos y un análisis detallado y especializado del área de estudio, surge la necesidad de obtener una herramienta que permita observar la distribución geoespacial de su información y el impacto que los resultados de un análisis específico pueda brindar en el avance de las investigaciones.

Por tal motivo se implementa un sistema georeferencial basado en una ontología dirigida a un estudio de aves que habitan en una región determinada, que permite al usuario por medio de búsquedas especializadas obtener la ubicación y distribución geoespacial según especie, estatus, estado y género de la muestra considerada para el desarrollo del prototipo de la aplicación.

La aplicación de conocimientos sobre *web* semántica y georeferenciación son necesarios para el desarrollo de una aplicación de este tipo, utilizando herramientas que brinden la exactitud y eficacia necesaria que permite cumplir con los requerimientos de los usuarios y generar conocimiento. Para ellos se utilizan herramientas como *Protégé* y *Google Maps*, aplicado a una plataforma orientada a objetos como *Java*, utilizando como gestor de base de datos *MySQL* y lenguaje de consultas *Sparql*.

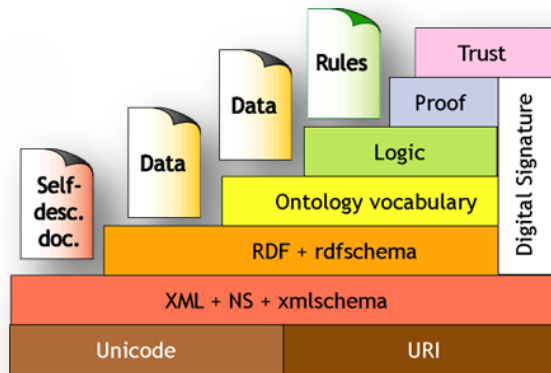
1. WEB SEMÁNTICA

Actualmente, la mayoría de buscadores trabajan por medio de palabras claves que aparecen en el código *HTML* de los diferentes sitios *web*. Esto genera sobrecarga de información, ya que muchas veces retorna al usuario información que no corresponde a la búsqueda.

Para que las búsquedas sean óptimas, se requiere que los ordenadores sean capaces de comprender y entender el contenido que almacena la red y relacionarlo con lo que el usuario le solicita de manera más natural. Con este propósito da inicio la *Web Semántica*, la cual permite a los usuarios encontrar respuestas rápidas y óptimas a sus preguntas.

La *Web Semántica* fue introducida por el creador de la *WWW*, Tim Berners Lee en septiembre de 1998, con apoyo de otras personas del *W3C* (*World Wide Web Consortium*). Propone que la *Web Semántica* debe poseer una Arquitectura como la que se muestra en la Figura 1.

Figura 1. **Arquitectura de la Web Semántica**



Fuente: Tim Berners-Lee. *Semantic Web -XML2000. Architecture*
<http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide11-0.html>

Esta arquitectura, mostrada en la figura 1, tiene como fin principal la estandarización de los lenguajes utilizados para el desarrollo *web* para crear agentes *web* que puedan comprender la información de una forma semiautomática. Estos agentes deben ser capaces de convertir los datos que tienen en conocimiento. Esto por medio del cumplimiento de ciertas características considerando que sea capaz de transmitir información del entorno mediante ontologías; proporcionar información y, además, prestar un servicio que sea autónomo en sus acciones y aprenda de otros agentes.

Con esto se puede decir que un agente es capaz de entender qué se le solicita, comprender el contenido, verificar si cumple con los requerimientos y concluir de lo obtenido. Esto se logra gracias a la arquitectura de la *Web Semántica*, ya que estandariza las funciones por medio de las capas que posee.

1.1. Arquitectura de la *web* semántica

1.1.1 *Unicode*

Es el alfabeto entre la computadora y el usuario. Esta es la estructura básica que permite se codifique texto sin restricción de símbolos e idiomas.

1.1.2 *URI*

Las referencias. *URI* (Identificador Uniforme de Recursos) permite localizar un recurso que puede ser utilizado en la *WEB*.

1.1.3 *XML + NS + XML Schema*

Capa técnica que permite el marcado estructurado de los datos por medio de las etiquetas que utiliza el lenguaje *XML*, así como, *NS* que relaciona los atributos con los nombres asociados a los *URI* y *XML Schema*, que permite crear un estándar de los documentos.

1.1.4 *RDF + RDF Schema*

RDF es un lenguaje simple mediante el cual se definen las sentencias en el formato de una triple relación, la cual tiene sujeto, predicado y objeto. *RDF Schema* que brinda un vocabulario definido sobre *RDF* que permite el modelo de objetos con una semántica claramente definida.

1.2. Lenguaje de ontologías

Estos brindan un lenguaje que permite catalogar y clasificar la información, describiendo los entes y las relaciones que existen entre ellos.

1.2.1. Lógica

Además de ontologías, se precisan también reglas de inferencia, como por ejemplo: si Juan es hijo de Pedro y Pedro es papá de María, entonces María y Juan son hermanos. Es decir, la información se manipula de manera que la computadora entienda lo que sucede, como si fuera un humano quien lo realiza.

1.2.2. Pruebas, confianza y firma digital

Estos tienen como finalidad comprobar los resultados, las fuentes de la información y, por último, encriptar el contenido.

1.3. Ontologías

Las ontologías pertenecen al área de inteligencia artificial. Estas permiten definir las áreas importantes o vocabulario de un dominio, es decir, catalogan las propiedades y crean las relaciones entre los elementos para así generar una definición adecuada.

Las ontologías por medio de la definición de las tripletas y las relaciones entre los objetos, permiten obtener la información, procesarla y así realizar conclusiones de esta, a través de la generación de un modelo relacional en una base de datos. En otras palabras, tiene un enfoque con el cual pretende hacer de la computadora un ente que entienda o que sea inteligible, es decir, que al

momento de tener un conjunto de datos, entienda y relacione los mismos, que no represente solamente un conjunto de datos sin sentido. Se trata que la computadora, procese los datos y esto brinde como resultado la transmisión de conocimiento al usuario.

Las ontologías están definidas por un sujeto, predicado y un objeto, así también permite definir sinónimos y homónimos, con el fin de ampliar el vocabulario y aumentar el conocimiento que pueda ser capaz de transmitir.

1.3.1. Usos de la ontología

1.3.1.1. Comunicación entre personas, organizaciones y aplicaciones

Esto se logra al tener una conceptualización y comprensión común del dominio; por ejemplo, se puede decir que una *Notebook* es parte de computadoras, entonces al escribir computadora debe mostrar *Notebook*.

1.3.1.2. Interoperabilidad de sistemas

Se tiene en el momento en que dos sistemas son capaces de trabajar de manera conjunta, sin la intervención humana y de una forma automática.

1.3.1.3. Razonamiento automático

Por medio de las reglas de inferencia se puede hacer que la computadora tome decisiones automáticamente. Por ejemplo, al decir Juan vive en la calle de Los Pasos, la calle de Los Pasos está ubicada en la Antigua Guatemala, por tanto, Juan vive en la Antigua Guatemala.

1.3.2. Componentes de una ontología

Una ontología se integra por una serie de componentes que permiten hacer una representación del conocimiento que se intenta transmitir de un dominio:

- **Conceptos:** son los objetos que se intentan formalizar.
- **Relaciones:** representan la interacción entre los conceptos que se tienen. Por ejemplo, es parte de, tiene un, cerca de, pertenece a, etc.
- **Funciones:** es una relación de alto nivel, que busca identificar un elemento mediante acciones sobre varios elementos de una ontología.
- **Instancia:** representa un objeto específico de un concepto.
- **Axiomas:** son las reglas que se deben cumplir sobre las relaciones definidas para los elementos de la ontología. Por ejemplo, si se tiene que Juan y Pedro son mexicanos, entonces Juan no es de Guatemala.

Para el almacenamiento de la información o conocimiento de lo que se desea crear la ontología, existen herramientas de *software* capaces de facilitar esta tarea.

Protégé es una herramienta gratuita que permite el almacenamiento de dicha información utilizando un archivo de texto *OWL* y *RDF*.

OWL es el acrónimo de *Ontology Web Language*, es un lenguaje que se utiliza para la creación de ontologías con orientación a la *web* semántica.

Consta de tres elementos: individuos, propiedades y clases, los cuales están directamente relacionados con los componentes de una ontología.

En *Protégé* es posible la creación de clases y subclases, así como las relaciones entre las mismas y aplicar los usos de una ontología que fueron mencionados con anterioridad. Los elementos de esta aplicación son casos, *slots* y clases que tienen el mismo significado que los elementos de *OWL*. Cuando se desarrollan aplicaciones basadas en éstas ontologías es necesario exportar la información a modelos relacionales, es decir, un almacenamiento persistente como una base de datos. Para tales fines es posible la utilización de un *plugin* de *Protégé* llamado *Protégé2Jena* que es una herramienta de fácil instalación. De igual forma que *Protégé*, su funcionamiento es bastante sencillo. Con esta aplicación se mapea la información a una base de datos relacional que permitirá realizar consultas sobre la misma.

MySql es el gestor de bases de datos que se utiliza en el desarrollo del proyecto. Este puede descargarse de la *Web* de una forma gratuita y brinda una herramienta eficaz y robusta que permite almacenar el modelo relacional que generó el mapeo de la ontología por medio de *Jena*.

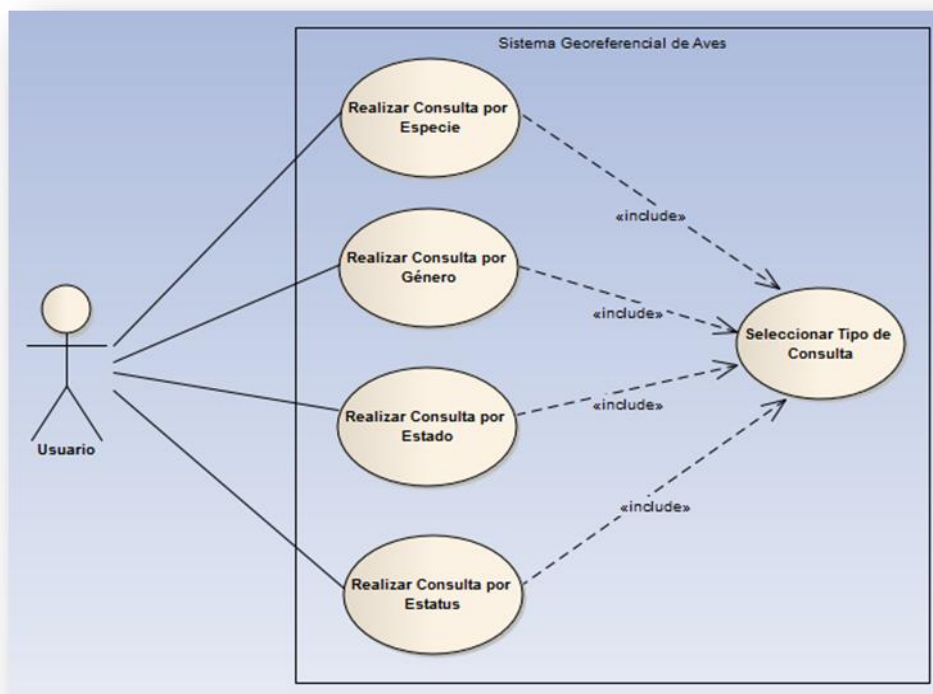
NetBeans es una plataforma de desarrollo basado en Java licenciado bajo *CDDL* y *GPLv2*.

Esta herramienta permite el desarrollo de aplicaciones por módulos y tiene la capacidad para crear aplicaciones *web*. Permitiendo la conexión a bases de datos como *MySQL*, así también importar librerías o *APIs* necesarios para el desarrollo de proyectos completos.

Finalmente, la georeferenciación es un término que en los últimos tiempos ha sido aplicado para temas de investigación, ya que permite un análisis espacial y facilita la toma de decisiones. Actualmente, *Google Inc.* cuenta con un *API* llamado *Google Maps*, el cual se puede definir como: “*Google Maps* es el nombre de un servicio gratuito de *Google*. Es un servidor de aplicaciones de mapas en *Web*. Ofrece imágenes de mapas desplazables, así como fotos satelitales del mundo entero e incluso la ruta entre diferentes ubicaciones. Es similar a *Google Earth*, una aplicación *Windows/Mac/Linux* que ofrece vistas del globo terráqueo impactantes, ofrece, así mismo, la posibilidad de que cualquier propietario de una página *Web* integre muchas de sus características a su sitio *Web*.” Fuente: Enciclopedia Libre *Wikipedia*.

2. ANÁLISIS Y DISEÑO

Figura 2. Diagrama casos de uso general



Fuente: Ana Luisa Muñoz Tello/ Evelyn Silvana Recinos Herrera.
Diagrama de casos de uso general

2.1. Descripción caso de uso general

En el diagrama se visualizan los casos de uso identificados, para poder cumplir con los requerimientos de los usuarios. En este se observa las relaciones que existen entre los casos de uso y el actor.

El único actor del diagrama de casos de uso es el Usuario, el cual puede ser cualquier persona que navegue por el sitio *web*.

Como se puede observar, se tienen cuatro casos de uso básicos:

- a. Realizar Consulta por Especie
- b. Realizar Consulta por Género
- c. Realizar Consulta por Estado
- d. Realizar Consulta por Estatus

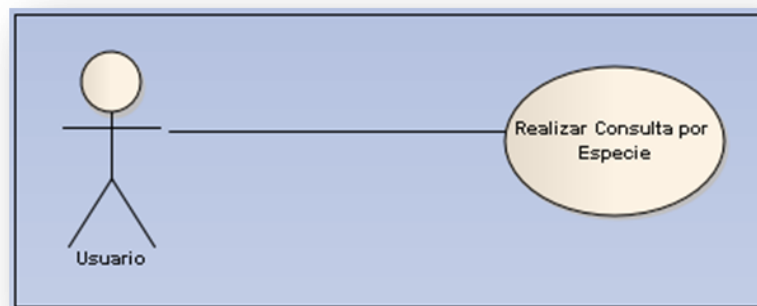
Además se tiene una relación *Include* con el caso de Uso Ingresar Tipo de Consulta, ya que es necesario que el usuario indique que tipo de consulta desea realizar, esta opción será mostrada en la página principal.

2.2. Especificación casos de uso

Caso de uso No. 1

2.2.1. Realizar Consulta por Especie

Figura 3. Diagrama caso de uso No. 1



Fuente: Ana Luisa Muñoz Tello/ Evelyn Silvana Recinos Herrera
Diagrama de caso de uso Realizar Consulta por Especie

2.2.1.1 Descripción

El usuario desea conocer la ubicación geográfica en la cual se localizan individuos de la especie ingresada en la consulta.

2.2.1.2 Objetivo

Mostrar al usuario en un mapa, utilizando la aplicación *Google Maps* los lugares en los cuales se ha observado presencia de la especie de aves indicada en la búsqueda.

2.2.1.3 Flujo básico

- a. El usuario selecciona el tipo de consulta;
- b. El usuario ingresa el nombre del ave;
- c. El usuario presiona el botón Enviar;
- d. El usuario presiona el botón Buscar;
- e. El sistema por medio de la ontología almacenada en la base de datos, realiza la búsqueda;
- f. El sistema muestra los resultados georeferenciando los mismos en un mapa.

2.2.1.4 Flujo alterno

- a.1 El usuario no ingresa una cadena válida.
- c.1 El usuario busca la información antes de enviarla.
- f.1 El sistema no muestra los resultados, ya que no encuentra información asociada al tipo de especie indicada.

2.2.1.5 Prerrequisito

Ingreso del tipo de consulta.

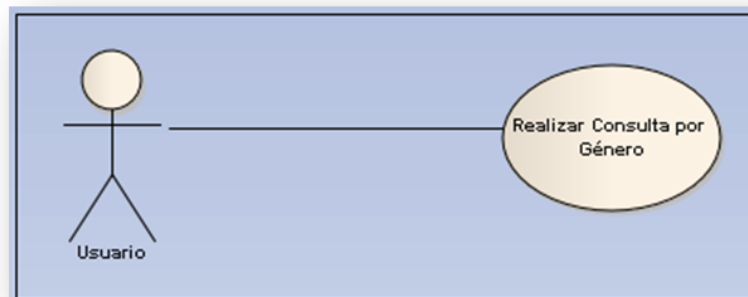
2.2.1.6 Riesgo

No retornar los resultados esperados por el usuario, de acuerdo con la consulta ingresada.

Caso de uso No. 2

2.2.2. Realizar Consulta por Género

Figura 4. Diagrama caso de uso No. 2



Fuente: Ana Luisa Muñoz Tello/ Evelyn Silvana Recinos Herrera
Diagrama de caso de uso Realizar Consulta por Género

2.2.2.1 Descripción

El usuario desea conocer la ubicación geográfica en la cual se localiza el género del ave indicado en la consulta.

2.2.2.2 Objetivo

Mostrar al usuario en un mapa utilizando la aplicación *Google Maps* los lugares en los cuales se ha observado presencia de individuos, dependiendo del género indicado en la búsqueda.

2.2.2.3 Flujo básico

- a. El usuario selecciona el tipo de consulta;
- b. El usuario ingresa el nombre del ave;
- c. El usuario presiona el botón Enviar;
- d. El usuario presiona el botón Buscar;
- e. El sistema por medio de la ontología almacenada en la base de datos, realiza la búsqueda;
- f. El sistema muestra los resultados georeferenciando los mismos en un mapa.

2.2.2.4 Flujo alterno

- a.1 El usuario no ingresa una cadena válida.
- c.1 El usuario busca la información antes de enviarla.
- f.1 El sistema no muestra los resultados, ya que no encuentra información asociada al género indicado.

2.2.2.5 Prerrequisito

Ingreso del tipo de consulta

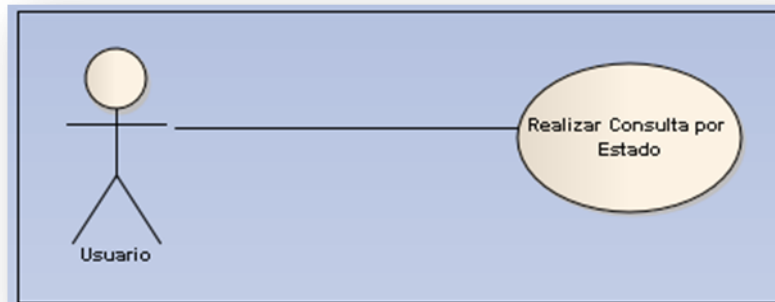
2.2.2.6 Riesgo

No retornar los resultados esperados por el usuario, de acuerdo con la consulta ingresada.

Caso de uso No. 3

2.2.3. Realizar Consulta por Estado

Figura 5. Diagrama caso de uso No. 3



Fuente: Ana Luisa Muñoz Tello/ Evelyn Silvana Recinos Herrera
Diagrama de caso de uso Realizar Consulta por Estado

2.2.3.1 Descripción

El usuario desea conocer la ubicación geográfica en la cual se localizan individuos, dependiendo del estado ingresado en la consulta.

2.2.3.2 Objetivo

Mostrar al usuario en un mapa utilizando la aplicación *Google Maps* los lugares en los cuales se ha observado presencia de individuos, dependiendo del estado indicado en la búsqueda.

2.2.3.3 Flujo básico

- a. El usuario selecciona el tipo de consulta;
- b. El usuario ingresa el nombre del ave;
- c. El usuario presiona el botón Enviar;
- d. El usuario presiona el botón Buscar;
- e. El sistema por medio de la ontología almacenada en la base de datos, realiza la búsqueda;
- f. El sistema muestra los resultados georeferenciando los mismos en un mapa.

2.2.3.4 Flujo alterno

- a.1 El usuario no ingresa una cadena válida.
- c.1 El usuario busca la información antes de enviarla.
- f.1 El sistema no muestra los resultados, ya que no encuentra información asociada al estado indicado.

2.2.3.5 Prerrequisito

Ingreso del tipo de consulta

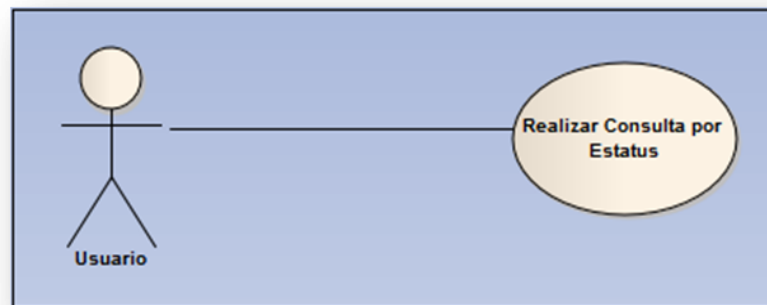
2.2.3.6 Riesgos

No retornar los resultados esperados por el usuario, de acuerdo con la consulta ingresada.

Caso de uso No.4

2.2.4. Realizar consulta por estatus

Figura 6. Diagrama caso de uso No. 4



Fuente: Ana Luisa Muñoz Tello/ Evelyn Silvana Recinos Herrera
Diagrama de caso de uso Realizar Consulta por Estatus

2.2.4.1. Descripción

El usuario desea conocer la ubicación geográfica en la cual se localizan individuos dependiendo del estatus ingresado en la consulta.

2.2.4.2. Objetivo

Mostrar al usuario utilizando *Google Maps* los lugares en los cuales se ha observado presencia de individuos, dependiendo del estatus indicado en la búsqueda.

2.2.4.3. Flujo básico

- a. El usuario selecciona el tipo de consulta;
- b. El usuario ingresa el nombre del ave;
- c. El usuario presiona el botón Enviar;
- d. El usuario presiona el botón Buscar;
- e. El sistema por medio de la ontología almacenada en la base de datos, realiza la búsqueda;
- f. El sistema muestra los resultados georeferenciando los mismos en un mapa.

2.2.4.4 Flujo alterno

- a.1 El usuario no ingresa una cadena válida.
- c.1 El usuario busca la información antes de enviarla.
- f.1 El sistema no muestra los resultados, ya que no encuentra información asociada al estatus indicado.

2.2.4.5 Prerrequisito

Ingreso del tipo de consulta

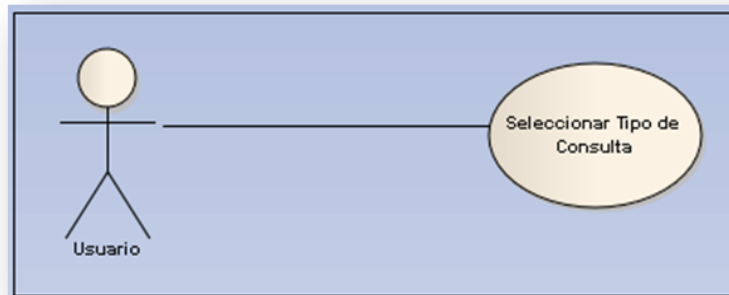
2.2.4.6 Riesgo

No retornar los resultados esperados por el usuario, de acuerdo con la consulta ingresada.

Caso de uso No. 5

2.2.5. Ingresar tipo de consulta

Figura 7. Diagrama caso de uso No. 5



Fuente: Ana Luisa Muñoz Tello/ Evelyn Silvana Recinos Herrera
Diagrama de caso de uso Ingresar Tipo de Consulta

2.2.5.1. Descripción

El usuario selecciona, entre las opciones que se le presentan, el tipo de consulta que desea realizar, por especie, género, estado o estatus.

2.2.5.2. Objetivo

Mostrar al usuario por medio de un combo de selección los tipos de consulta que puede realizar.

2.2.5.3 Flujo básico

- a. El usuario selecciona el tipo de consulta.

2.2.5.4 Flujo alternativo

- a.1 El sistema no puede realizar esta acción en este momento;
- a.2 El sistema no puede cargar la página con el contenido deseado.

2.2.5.5 Postrequisito

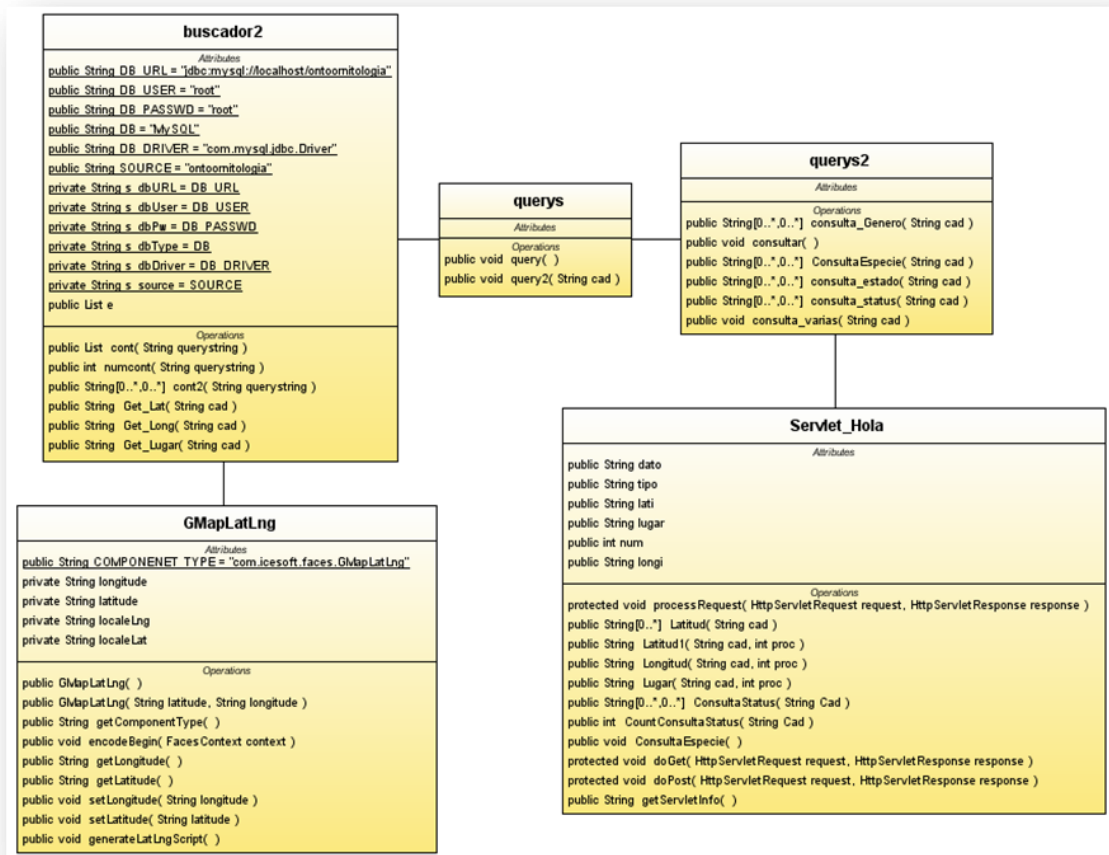
- Realizar consulta por especie
- Realizar consulta por género
- Realizar consulta por estado
- Realizar consulta por estatus

2.2.5.6 Riesgo

No retornar los resultados esperados por el usuario, de acuerdo con la consulta seleccionada.

2.3 Diagramas

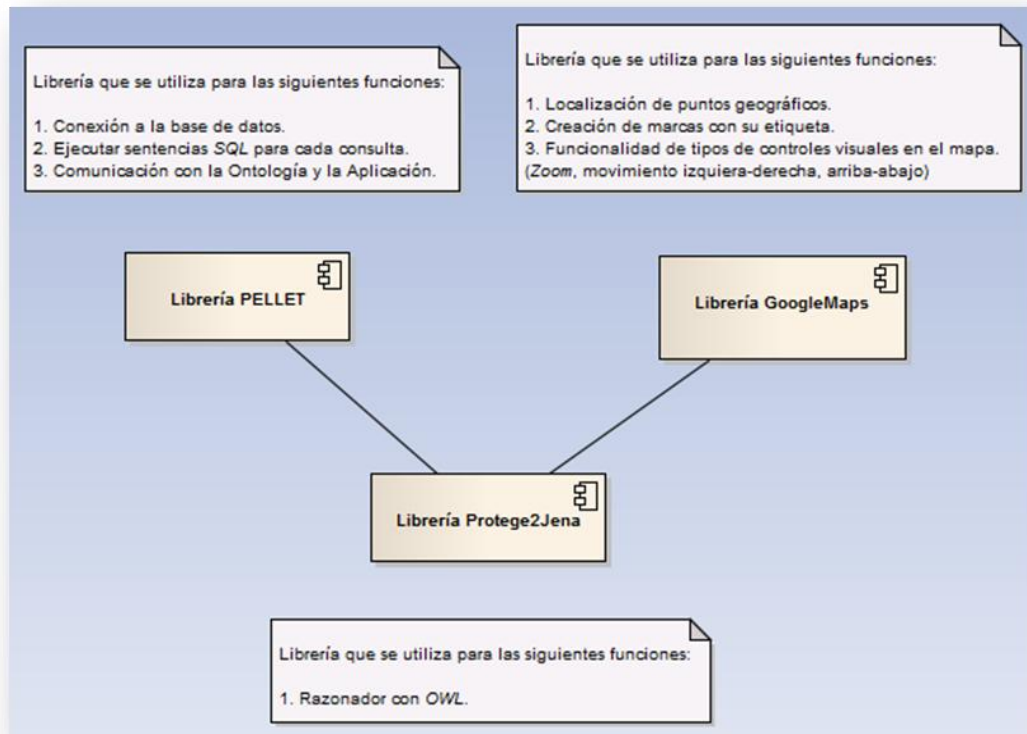
Figura 8. Diagrama de clases



Fuente: Ana Luisa Muñoz Tello/ Evelyn Silvana Recinos Herrera

Diagrama de clases de la aplicación

Figura 9. Diagrama de componentes



Fuente: Ana Luisa Muñoz Tello/ Evelyn Silvana Recinos Herrera
Diagrama de componentes.

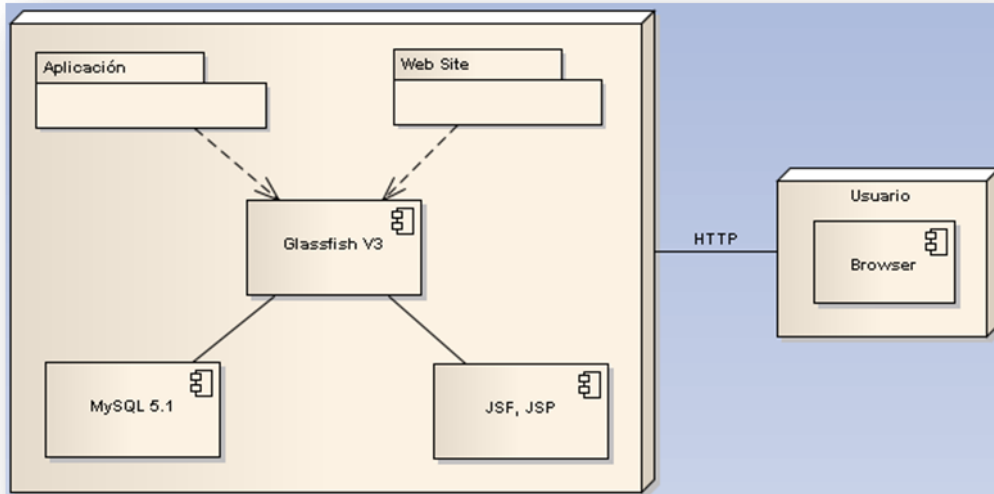
En el diagrama de componentes del sistema por desarrollar, se requiere el uso de 3 paquetes de librerías.

- a. *Google Maps*: librería que permite el enlace con la aplicación generadora de mapas georeferenciales *Google Maps*. De esta manera, se creará la referencia de los sitios a buscar en un mapa.
- b. *Protégé2Jena*: librería de *java*, que es utilizada para instanciar la ontología desarrollada en *protégé* desde la aplicación de *Netbeans*. Esta permite generar una conexión al gestor de la Base de Datos (*MySql*), que es el

lugar de almacenamiento de la información generada a partir de la ontología, posterior a su proceso de Mapeo. Así también, permite la manipulación de la información por medio de consultas que se realizan utilizando el lenguaje *Sparql*.

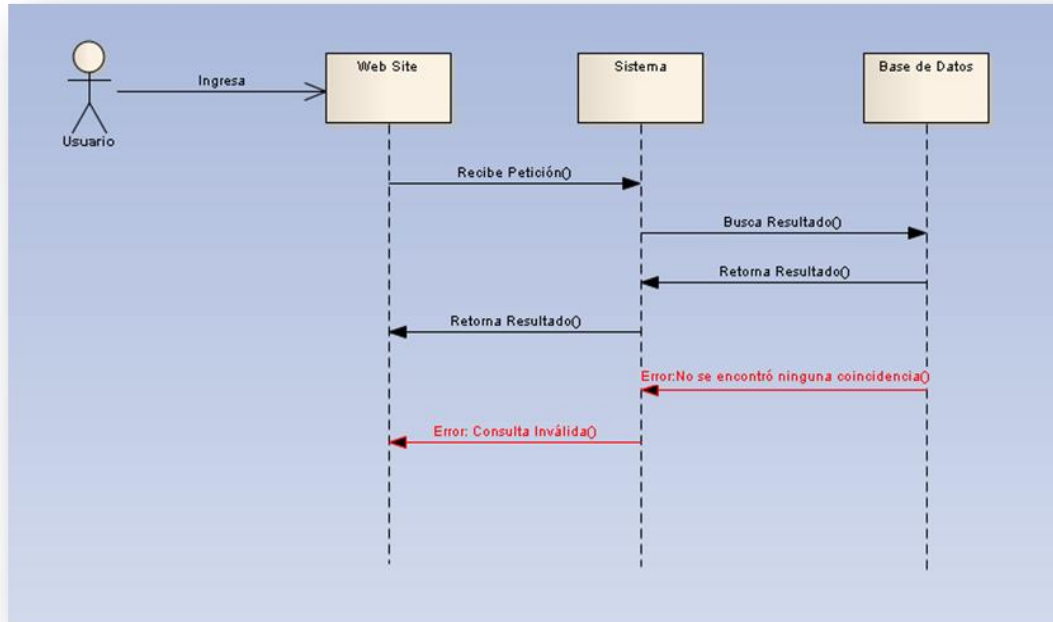
- c. *Pellet*: es el razonador de la información que fue ingresada en la ontología. Realiza la función de analizar la información que se encuentra almacenada en la ontología determinando las relaciones existentes entre las diferentes clases que están definidas.

Figura 10. Diagrama de *deployment*



Fuente: Ana Luisa Muñoz Tello/ Evelyn Silvana Recinos Herrera
Diagrama de *deployment*

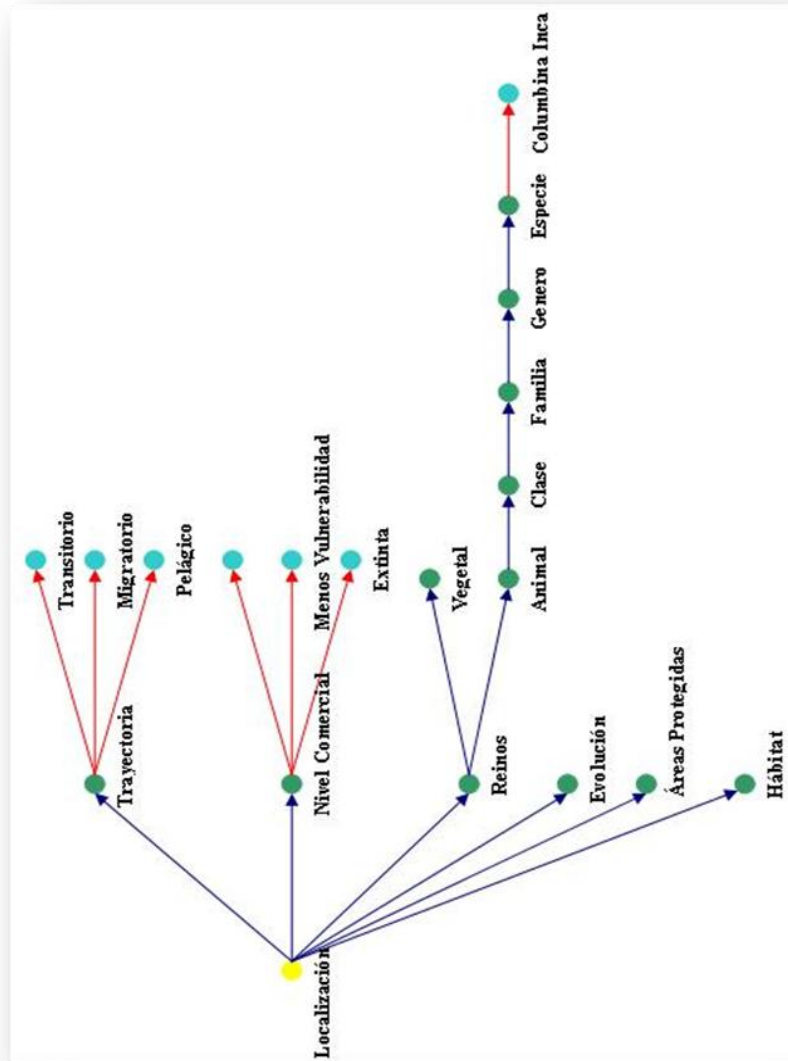
Figura 11. Diagrama de secuencia



Fuente: Ana Luisa Muñoz Tello/ Evelyn Silvana Recinos Herrera
Diagrama de secuencia

Ontología

Figura 12. Modelo Conceptual Ontología



Fuente: Ana Luisa Muñoz Tello/ Evelyn Silvana Recinos Herrera
Modelo conceptual

3. CONSULTAS SPARQL

SPARQL -Protocol and RDF Query Language-

3.1. Definición

Lenguaje utilizado para generar consultas que permitan realizar análisis semántico dentro un sitio *web*.

3.2. Sintaxis

Prefix: es el lugar en el cual se definen los *namespace*, en otras palabras se puede decir que se define el nombre con el cual se resume el nombre de la instancia a la ontología.

Select: identifica los campos que se seleccionan como resultado, es decir, se coloca el nombre de las variables que se desea obtener como resultado y el nombre con el cual se identificará.

Where: define las condiciones que se deben cumplir para que la consulta se realice de una forma óptima. La sintaxis del mismo es en forma de tripleta. La tripleta está conformada por Recurso, Propiedad y Objeto.

- Recurso: es el identificador del individuo del cual se desea cumplir la condición;

- Propiedad: es donde se indica la relación que debe cumplirse. Esto puede ser una propiedad o relación de la ontología;
- Objeto: es lo que se busca. Instancia en sí que puede ser *Filter*, etc.

3.3. Especificación *Sparql*

Las consultas se crearon con el fin de cumplir con los diferentes tipos de búsquedas que el usuario puede solicitar. Se utilizó la sintaxis básica de *SPARQL*, para la generación de las mismas.

El siguiente ejemplo permite obtener la información de la ubicación geográfica, en la cual se ha observado presencia de las aves que en su nombre tienen la palabra “ALBI”.

```
PREFIX onto:<http://www.owl-ontologies.com/Ontology.owl#>
SELECT ?n ?individuo
WHERE {?n onto:Nombre ?individuo.FILTER regex(?individuo,"ALBI") .}
```

Esta consulta obtiene el nombre de las especies que están ubicadas en el Cerro San Gil.

```
PREFIX onto:<http://www.owl-ontologies.com/Ontology1238818985.owl#>
SELECT ?ESPECIE ?individuo ?w ?Nombre_coor ?LATITUD ?LONGITUD
WHERE {?n onto:Nombre ?individuo.
?n onto:EXISTE_EN ?z.
?z onto:Nombre_coor ?Nombre_coor.
?z onto:LATITUD ?LATITUD.
?z onto:LONGITUD ?LONGITUD.
FILTER regex(?w,"CERRO SAN GIL") . }
```

Consulta que devuelve la latitud y longitud de los lugares en los cuales se ha observado presencia del nombre de la especie indicada por el usuario, en este ejemplo "COLUMBINA INCA". Para esto, el usuario ingresa la cadena "COLUMBINA INCA".

```
PREFIX onto:<http://www.owl-ontologies.com/Ontology1238818985.owl#>
SELECT ?individuo ?LOCALIZACION ?LATITUD ?LONGITUD
WHERE {?n onto:Nombre ?individuo.
?n onto:SE_ENCUESTRAN_EN ?z.
?z onto:Nombre ?LOCALIZACION.
?z onto:LATITUD ?LATITUD.
?z onto:LONGITUD ?LONGITUD.
FILTER regex(?individuo,"COLUMBINA INCA") . }
```

Consulta que permite obtener los sitios en los cuales se han observado las aves con estatus indicado por el usuario. La cadena que el usuario podría ingresar para este ejemplo es "RESIDENTE". De igual forma puede ser Migratorio, Residente-Pelágico.

```
PREFIX onto:<http://www.owl-ontologies.com/Ontology1238818985.owl#>
SELECT ?ESPECIE ?individuo ?w ?latitud ?longitud
WHERE {?n onto:Nombre ?individuo.
?n onto:EXISTE_EN ?z.
?z onto:Nombre ?w.
?w onto:Nombre?p.
FILTER regex(?w,"RESIDENTE") . }
```


Consulta que devuelve la latitud y longitud de los lugares en los cuales se ha observado presencia del género indicada por el usuario, en el ejemplo siguiente “*COLUMBIFORMES*”. Para este ejemplo, el usuario ingresa la cadena “*COLUMBIFORMES*”.

```
PREFIX onto:<http://www.owl-ontologies.com/Ontology1238818985.owl#>
SELECT ?ESPECIE ?individuo ?w
WHERE {?n onto:Nombre ?individuo.
?n onto:SE_LOCALIZAN_EN ?z.
?z onto:Nombre ?w.
FILTER regex(?individuo,"COLUMBIFORMES") .}
```

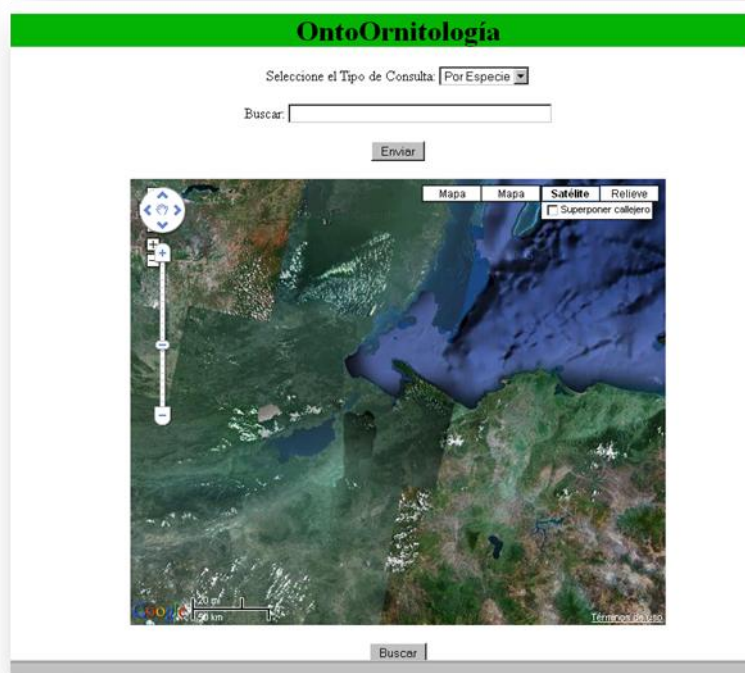
Consulta que devuelve la latitud y longitud de los lugares en los cuales se ha observado presencia de la familia de las aves indicada por el usuario, en el ejemplo siguiente “*COLUMBIDAE*”. Para este ejemplo, el usuario ingresa la cadena “*COLUMBIDAE*”.

```
PREFIX onto:<http://www.owl-ontologies.com/Ontology1238818985.owl#>
SELECT ?ESPECIE ?individuo ?w
WHERE {?n onto:Nombre ?individuo.
?n onto:SE_LOCALIZAN_EN ?z.
?z onto:Nombre ?w.
FILTER regex(?individuo,"COLUMBIDAE") .}
```

4. PROTOTIPO DE LA APLICACIÓN WEB

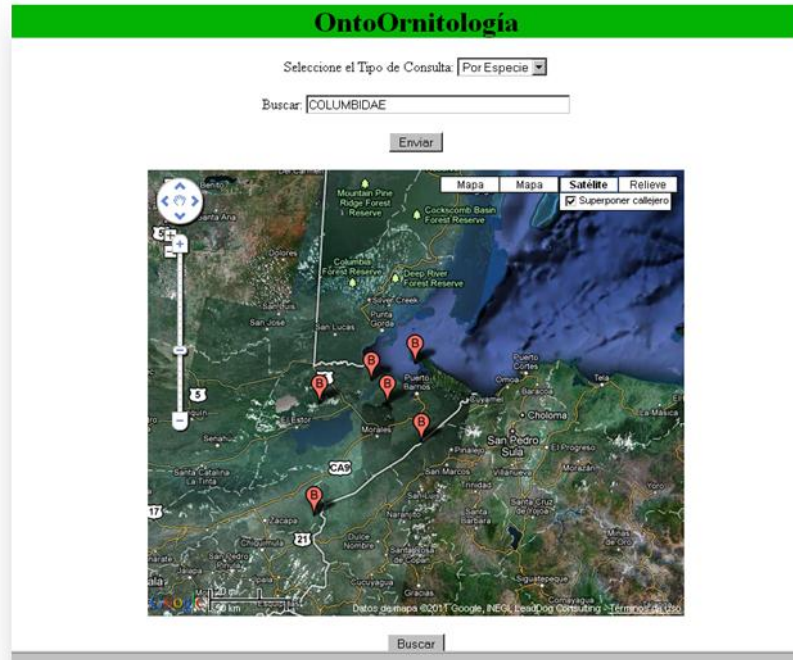
La aplicación cuenta con una página principal en la cual se debe seleccionar el tipo de consulta que se desea realizar (Por especie, estatus, estado y género). Se debe ingresar la descripción de lo que se desea buscar, luego presionar el botón enviar para que esto sea procesado en la ontología y el botón buscar para que se localicen los puntos en el mapa como se observa en las siguientes figuras.

Figura 13. **Página Principal**



Fuente: Ana Luisa Muñoz Tello/ Evelyn Silvana Recinos Herrera
Sitio Web página principal

Figura 14. Ejemplo de consulta



Fuente: Ana Luisa Muñoz Tello/ Evelyn Silvana Recinos Herrera
Sitio Web área de despliegue de la información

CONCLUSIONES

1. Por medio de la utilización de *Protégé* y *Jena*, se tiene una ontología que permite realizar diferentes tipos de consultas acorde a los requerimientos del cliente, por ejemplo: identificar qué aves se localizan en determinada región y la ubicación de las mismas.
2. Los sistemas de gestión del conocimiento, por medio del análisis semántico que brinda el desarrollo de una ontología, facilitan al usuario la personalización y captura de consultas cercanas a los que éste solicita.
3. La georeferenciación es facilitada por herramientas que actualmente existen y son proporcionadas bajo licencias gratuitas para ser implementadas de una forma fácil y eficaz en sitios *web*, tal es el caso de *Google Maps*, que brinda el servicio de visualización de mapas; permitiendo al usuario una sencilla y amigable manipulación de los mismos. Así también, la aplicación por medio de su análisis y diseño ha sido creada con el fin de que el usuario encuentre una interfaz agradable y tiempos de respuesta inmediata.

RECOMENDACIONES

1. El análisis georeferencial, acompañado de la *Web Semántica*, permite generar conocimiento que facilita la toma de decisiones y permite orientar a las empresas y/o investigadores en el análisis de sus resultados. Por ello, se recomienda que las empresas enfoquen parte de sus esfuerzos en apoyar este tipo de proyectos.

BIBLIOGRAFÍA

1. ORACLE CORPORATION (2010). *NetBeans* [Consultado en octubre 2010]. Disponible en *Web*: <http://www.netbeans.org>.
2. Pérez, Chantal (2005). *Definición de ontología como especificación del conocimiento*. [Consultado en octubre 2010]. Disponible en *Web*: <http://elies.rediris.es/elies18/531.html>.
3. Stanford Center for Biomedical Informatics Research (2011). *PROTÉGÉ*. [Consultado en enero 2011]. Disponible en *Web*: <http://protege.stanford.edu/>.
4. TECHNOLOGIES INC. ICESOFTE (2009). *Icefaces, Open Source Ajax for Java EE*. [Consultado en octubre 2010]. Disponible en *Web*: <http://www.icesoft.com>.
5. W3C (2011). *World Wide Web Consortium*. [Consultado en enero 2011]. <http://www.w3.org>
6. Wikipedia (2011). *Ontología*. [Consultado en enero 2011]. Disponible en *Web*: <http://es.wikipedia.org>.
7. _____. *Ornitología*. [Consultado en enero 2011]. Disponible en *Web*: <http://es.wikipedia.org>.

8. _____. *Protégé*. [Consultado en enero 2011]. Disponible en *Web*: <http://es.wikipedia.org>.
9. _____. *Web Semántica*. [Consultado en enero 2011]. Disponible en *Web*: <http://es.wikipedia.org>.