

Universidad Central "Marta Abreu" de Las Villas
Facultad de Matemática, Física y Computación



**TOCORORO, SERVICIO DE DATOS TEMÁTICOS,
GEOREFERENCIADOS EN LA NUBE
BASADO EN SOFTWARE LIBRE**

**TESIS PRESENTADA EN OPCIÓN AL TÍTULO ACADÉMICO DE MÁSTER EN
INFORMÁTICA PARA LA GESTIÓN MEDIOAMBIENTAL**

Autor: Rebeca Pentón Montejo

Tutor: Dr. Carlos Pérez Risquet

MSc. Hermes Lorenzo Méndez

Santa Clara

2013

PENSAMIENTO

*He sido un artista toda mi vida, pero la educación, el medio ambiente y mi
disposición me han hecho lo que soy.*

Franz Marc

DEDICATORIA

“Con todo amor a mi madre.”

AGRADECIMIENTOS

Quiero agradecer a las personas que me brindaron su apoyo para desarrollar este trabajo:

A mis tutores, Dr. Carlos Pérez Risquet y Msc. Hermes Lorenzo Méndez, por sus conocimientos invaluableles que me brindaron para llevar a cabo esta investigación.

Agradezco a mi madre, a mis padres, a mis abuelos por todo su apoyo cuando más abatida me sentía.

Agradezco a mis compañeros de trabajo en especial a mi jefe por su comprensión, ayuda y dedicación que en todo momento me brindaron.

Agradezco a mis amigas del hotelito por toda la ayuda, ánimo y dedicación que me brindaron.

Agradezco a los miembros del jurado, por las valiosas contribuciones que hicieron al trabajo final y por el tiempo que dedicaron para revisarlo.

Agradezco a aquellas grandes personas, a los excelentes profesores del programa de maestría. A mis compañeros, por todos los buenos y malos momentos que pasamos juntos.

En fin a quienes sin escatimar esfuerzo alguno han sacrificado gran parte de su tiempo, que me han formado y educado. A quienes la ilusión de su existencia ha sido verme convertido en persona de provecho. A quienes nunca podré pagar con las riquezas más grandes de mundo. A ellos, los seres universalmente más queridos sinceramente

Gracias.

RESUMEN

Se propone la infraestructura de una Plataforma Colaborativa en la Nube (cloud computing) haciendo uso de tecnologías Web que permita publicar indicadores, información estadística o medioambiental que pueda ser georeferenciada así como la generación de mapas temáticos como complemento para la toma de decisiones en una arquitectura orientada a servicios. En la actualidad se pueden encontrar muchos ejemplos existentes de sistemas de información geográficos (SIG) así como por su parte se encuentra la creciente tendencia en la Internet a los blogs, las wikis y últimamente a las redes sociales. Este trabajo está orientado a la creación de una plataforma que unifique las tecnologías de los SIG con las crecientes tendencias de la Web 2.0, donde se pueda compartir e intercambiar la información gracias a las tecnologías de la computación en la nube. La presente investigación sin dudas se traduciría en beneficios para los usuarios o empresas en materia de obtención, procesamiento y distribución de datos.

ABSTRACT

The work is about the Collaborative Platform infrastructure in the cloud (cloud computing) using Web technologies that allow publishing indicators or statistical information that can be georeferenced environmental and thematic map generation in addition to decision making oriented architecture services.

Today you can find many existing examples of geographic information systems (GIS) and meanwhile there is the growing trend in Internet blogs, wikis and social networks lately. This work is aimed at creating a platform that unifies GIS technologies with the growing trends of Web 2.0, where you can share and exchange information through technology cloud computing. This research undoubtedly would result in benefits for users or companies in the procurement, processing and distribution of data.

ÍNDICE

Introducción	1
I. Situación problemática	4
II. Problema científico	5
III. Objetivo general	5
IV. Objeto de estudio y campo de acción	5
V. Objetivos específicos	5
VI. Preguntas científicas	6
VII. Métodos científicos	6
VIII. Aportes	7
IX. Estructura del trabajo de tesis	7
CAPÍTULO 1 Aspectos teóricos	8
1.1 Introducción	8
1.2 ¿Qué es una aplicación Web y Computación en la Nube?	8
1.2.1 ¿Computación en la nube?	8
1.2.1.1 Ventajas	9
1.2.1.2 Tipos de nubes	10
1.3 ¿Qué es un GIS o SIG?	10
1.4 ¿Qué es un Software Colaborativo?	11
1.4.1 Tipos de colaboración	11
1.4.2 Sistemas de Información Colaborativos	12
1.5 ¿Qué son mapas temáticos?	12
1.5.1 Tipos de mapas temáticos	13
1.6 ¿Qué es Espacio de Datos?	13
1.7 ¿Qué es Fuente de Datos?	14
1.8 ¿Qué son Campos Temáticos?	14
1.9 ¿Qué es información georeferenciada?	14
1.9.1 Diferencia entre Información TABULAR y ESPACIAL	14
1.10 ¿Quién es GEOMIX?	15
1.11 Misión y visión de GEOMIX	16
1.12 Descripción de los sistemas existentes	16
1.13 ¿Qué es un servidor de Mapas?	17
1.13.1 GeoServer	17

1.13.1.1	Características principales	18
1.14	¿Qué es Openlayers?	18
1.15	Sistemas Gestores de Bases de Datos. (SGBD)	19
1.15.1	¿Por qué PostgreSQL?	19
1.16	Apache Tomcat	20
1.17	¿Qué es Java?	21
1.17.1	Ventajas de Java.	21
1.17.2	Desventajas de Java.	22
1.17.3	¿Por qué Java y no otros lenguajes de programación?	22
1.18	Modelo-Vista-Presentador (MVP).	22
1.19	¿Qué son los marcos de trabajo (frameworks)?	24
1.20	¿Qué es Object Relational Mapping (Mapeo relacional de Objetos)?	24
1.20.1	Hibernate.....	25
1.20.1.1	Características	25
1.20.2	Hibernate Search	26
1.21	¿Qué es Inversión de Control?	26
1.21.1	¿Qué es Spring?	27
1.21.2	¿Qué es Spring Security?	27
1.22	AJAX como tecnología	27
1.23	El término AJAX Framework	28
1.24	Ext JS Framework y RichFaces.	28
1.25	¿Qué es GWT?	29
1.26	¿Qué es un Plug-in?	30
1.27	RUP	31
1.28	Conclusiones.....	32
CAPÍTULO 2 Diseño y modelación de la arquitectura		33
2.1	Introducción	33
2.2	Base de Datos	33
2.2.1	Problemática.....	33
2.2.2	Criterios evaluados.	33
2.2.3	Uso de Esquemas y Metadatos.	34
2.2.4	Modelo Entidad–Relación de las principales tablas	37
2.2.5	Descripción de Tablas	38
2.3	Gestión de grandes volúmenes de datos, gestión de concurrencia, velocidad de procesado y soporte para SIG	42

2.4	Búsqueda y recuperación fácil de la información.....	42
2.5	Seguridad sobre la información del usuario	43
2.6	Tendencias, metodologías y/o tecnologías empleadas en la investigación.	44
2.6.1	Arquitectura de N capas	44
2.6.2	Arquitectura orientada a Módulos	47
2.7	Descripción del modelo de sistema.	49
2.7.1	Visualización de la información georeferenciada y la información temática.	50
2.8	Extensibilidad mediante Plug-ins (Complementos).....	51
2.9	Internacionalización	51
2.10	Uso de Tecnologías de Software Librey Open Source.	52
2.10.1	Restricciones de la Arquitectura.....	52
2.10.2	Estándares de codificación	53
2.10.3	Patrones de diseño	53
2.10.4	Marcos de Trabajo (Frameworks).....	53
2.10.5	Plataforma de desarrollo	53
2.11	Modelación dela Plataforma.	54
2.11.1	Requerimientos funcionales	54
2.11.2	Requerimientos no funcionales.....	55
2.12	Modelo de casos de uso del sistema	58
2.12.1	Actores del sistema a automatizar	58
2.12.2	Paquetes y sus relaciones.	59
2.13	Descripción de los casos de uso	63
2.14	Conclusiones	64
CAPITULO 3 Implementación de la plataforma Web colaborativa en la nube		66
3.1	Introducción.	66
3.2	Implementación de la extensión importar y publicar cartografía	66
3.2.1	Modelo de Datos	66
3.2.1	Condiciones mínimas.....	66
3.3	Implementación de la extensión permitir campos tipo expresión autocalculada.....	67
3.3.1	Modelo de Datos	67
3.4	Implementación de la extensión de Generación de Mapas Temáticos.....	68
3.4.1	Modelo de Datos	68
3.4.2	Condiciones mínimas.....	68
3.5	Implementación de las plantillas de datos colaborativas.....	69
3.5.1	Modelo de Datos	69

3.6	Prototipos.....	70
3.6.1	Prototipo de Interfaz del Panel de Control (Usuario autenticado).....	70
3.6.2	Prototipo para definir campos auto calculado.....	74
3.6.3	Prototipo del Plug-in de Generación de Mapas Temáticos.....	79
3.6.3.1	Crear un nuevo mapa temático.....	79
3.6.3.2	Editar un mapa temático.....	80
3.6.3.3	Eliminar un mapa temático.....	80
3.6.3.4	Leyenda.....	80
3.6.4	Prototipo de la funcionalidad de subir y publicar cartografía.....	81
3.6.5	Prototipo de la Interfaz de Búsqueda (Usuario Anónimo).....	81
3.7	Configuraciones.....	83
3.7.1	Módulo Panel de Control.....	84
3.7.1.1	config.properties.....	84
3.7.1.2	jdbc.properties.....	85
3.7.1.3	Hibernate.cfg.xml.....	85
3.7.1.4	PluginsContext.xml.....	86
3.7.2	Módulo de Búsqueda.....	87
3.7.2.1	Config.properties.....	87
3.7.2.2	jdbc.properties.....	87
3.7.2.3	hibernate.cfg.xml.....	88
3.7.3	Módulo de visualización de mapas y datos temáticos (Visor).....	88
3.7.3.1	Config.properties.....	88
3.7.3.1	jdbc.properties.....	89
3.8	Conclusiones.....	90
CONCLUSIONES.....		91
RECOMENDACIONES.....		92
ANEXOS		98

ÍNDICE DE FIGURAS

Figura 1. Computación en la nube.	9
Figura 2. Modelo vista Presentador.	23
Figura 3. Organización jerárquica.	35
Figura 4. Uso de Esquemas.	35
Figura 5. Estructura del esquema "public".	36
Figura 6. Esquema de un usuario.	36
Figura 7. Modelo entidad-relación.	37
Figura 8. Diseño de la Arquitectura Tocatoro.	47
Figura 9. Arquitectura orientada a módulos.	48
Figura 10. Visor de mapas en la vista de mapas.	51
Figura 11. Diagrama de Casos de Usos por Paquetes.	60
Figura 12. Paquete web.	61
Figura 13. Diagrama de casos de uso del sistema: paquete panel de control.	62
Figura 14. Diagrama de caso de uso del sistema: paquete búsqueda.	62
Figura 15. Diagrama de caso de uso del sistema: paquete usuario.	63
Figura 16. Tablas que incluyen la funcionalidad de importar cartografía.	67
Figura 17. Tablas incorporadas a la base de datos de Tocatoro.	68
Figura 18. Cambios introducidos en la BD.	70
Figura 19. Interfaz del módulo panel de control de la plataforma Tocatoro.	71
Figura 20. Uso de internacionalización.	72
Figura 21. Menú opciones.	72
Figura 22. Filtrado del mapa.	73
Figura 23. Vista definir campo autocalculado.	75
Figura 24. Ejemplo de un campo autocalculado.	76
Figura 25. Mensajes de chequeo de la fórmula.	78

ÍNDICE DE TABLAS

Tabla 1. Definición de actores del sistema a automatizar.	59
Tabla 2. Descripción de paquetes.	60
Tabla 3. Descripción de los casos de uso y su prototipo.....	63
Tabla 4. Diagrama de actividades de los casos de uso del sistema.	64

Introducción

La primera computadora electrónica que entró en Cuba en el año 1958 fue una International Business Machines (IBM) RAMAC 650. Desde entonces hasta los años 90 en el país el desarrollo de la computación fue muy pobre, a pesar de que en 1970 se fabricó por primera vez una computadora cubana.

En la década de 1990 se comienzan a utilizar las primeras 386 y se empiezan a difundir las primeras versiones prácticas de WINDOWS. Las ventajas de las computadoras y su software asociado permitieron el uso de la informática a mayor escala. Sólo se puede hablar de generalización de la informática en la sociedad cubana a partir de 1997 a raíz del V Congreso del Partido Comunista de Cuba en que se dan orientaciones precisas para trabajar en ese sentido y el Gobierno aprueba, por primera vez, los Lineamientos Generales para la informatización de la sociedad, con objetivos generales hasta el 2000, que condujeron a la creación del Ministerio de la Informática y las Comunicaciones (MIC), con la misión de fomentar el uso masivo de las Tecnologías de la Información y las Comunicaciones (TIC) en la economía nacional, la sociedad y al servicio del ciudadano.(Lee, 2003).

La estrategia cubana de informatización está contenida en el Programa Rector de la Informatización de la Sociedad en Cuba, el que contempló en el 2005, siete áreas de acción (Barrio, 2005):

- Infraestructura, Tecnologías y Herramientas.
- Formación Digital.
- Fomento de las Tecnologías de la Información y las Comunicaciones.
- Investigación, Desarrollo y Asimilación de Tecnologías.
- Utilización de las TIC en la Dirección.
- Sistemas y Servicios Integrales para los ciudadanos.
- Utilización de las TIC en el Gobierno, la Administración y la Economía.

La disponibilidad oportuna de información confiable y significativa sobre los potenciales y las limitaciones de los recursos del medio ambiente es un requisito para la planificación y el manejo integrado de los mismos y poder potenciar el desarrollo sostenible.

En la sociedad moderna existe un número creciente de información que es considerada crítica para la toma de decisiones diarias. Cada vez se hace asequible mayor cantidad de información que se encuentra en línea (“on-line”) ya sea en blogs, wikis o redes sociales y por otra parte la posible relación de esta en algún contexto geográfico. Estas dos aristas por separadas se comportan como fuentes aisladas de información. En el entorno actual, cada día es mayor la necesidad de vincular información compartida por los usuarios con los recursos de datos geográficos siendo esto un requisito indispensable para apoyar a la sociedad en actividades como la visualización, planificación y toma de decisiones.

La tendencia en las tecnologías de la internet es a la migración hacia los servicios y la colaboración, comportándose la información como conocimiento, ejemplo claro de esto es la Web 2.0 y el concepto de la computación en la nube.

El término Web 2.0 se refiere a una segunda generación de Web basada en comunidades de usuarios y una gama especial de servicios, como las redes sociales, los blogs y los wikis, que fomentan la colaboración y el intercambio ágil de información entre los usuarios (Graells 2007). O sea, es una serie de aplicaciones y páginas de Internet que utilizan la inteligencia colectiva para proporcionar servicios interactivos en red, dando al usuario el control de sus datos.

El concepto de la computación en la nube (Cloud Computing) empezó en proveedores de servicio de Internet a gran escala, como Google, Amazon AWS, Microsoft y otros que construyeron su propia infraestructura. La **computación en la nube** conocida también como servicios en la nube, es un paradigma que permite ofrecer servicios de computación a través de Internet (Gilder 2006).

La colaboración se ha convertido en un elemento cada vez más importante en la economía de hoy, el compartir información y conocimiento son componentes vitales de una verdadera relación de colaboración.

El término de Software Colaborativo hace referencia a los métodos y herramientas de software que facilitan el trabajo en grupo, mejorando su rendimiento, y contribuyen a que personas que están localizadas en puntos geográficos diferentes puedan trabajar a la vez, ya sea directamente o de forma anónima, a través de las redes (Terán and Aldama 2008).

El estudio, análisis y exposición de las teorías, antecedentes, corrientes e investigaciones que definen o caracterizan el amplio espectro con que la ciencia de la dirección enfatiza la necesidad que tienen las organizaciones de disponer de un sistema de dirección, cuya tecnología gerencial permita su permanencia y desarrollo en correspondencia con los profundos y continuos cambios que acontecen en el entorno, requiere de un análisis crítico de las experiencias internacionales y de su adecuación y aplicación a las condiciones, no sólo de la economía nacional, sino también a las exigencias que define el objeto de estudio práctico al que se aplica.

La Dirección Empresarial (DE) ha asumido el empleo de técnicas modernas, adecuadas a las características propias y basadas en las más avanzadas prácticas contemporáneas, así como el amplio uso de todas las posibilidades de las tecnologías, servicios de información y las telecomunicaciones, con los fines de garantizar mayor eficiencia y control en la gestión de los procesos productivos.

La informática en Cuba cada día gana en desarrollo, se hacen grandes esfuerzos por llevarla a todos los niveles de la sociedad, como expresión de la voluntad para perfeccionar la labor de dirección empresarial a través de fórmulas más autónomas y ajustadas a nuestras condiciones y realidades.

De este modo se logra que el desarrollo del trabajo, en conjunto con la DE, la toma de decisiones, y el procesamiento de información sea superior y más eficiente, añadiendo el valor agregado que le reporta a cada servicio. Corresponde a los profesionales de la informática participar con implementaciones de sistemas informáticos que contribuyan a este avance. El desempeño por lograr aplicaciones más cercanas al usuario final, así como la facilidad a la hora de manipularlas, es el mayor propósito de los programadores.

Los avances técnicos significativos sin duda se traducirán en beneficios para materia de obtención, procesamiento y distribución de datos. En la esfera de la cartografía, los adelantos logrados en computadoras y en programas informáticos para la preparación de mapas ya han inducido a pasar de los métodos cartográficos tradicionales a los SIG.

Los SIG están diseñados para la administración de datos geográficos automatizados y permiten generar un sin número de alternativas y posibilidades, que en cierta forma colabora con la imaginación. Permite que las personas que utilicen esta tecnología se concentren en lo más importante: balancear dichas alternativas y tomar la decisión correcta.

I. Situación problemática

- ✓ La información estadística, medioambiental y cartográfica, crítica para la toma de decisiones, es manejada en departamentos o sectores aislados dentro de una misma organización, perdiendo tiempo, duplicando esfuerzos e inevitablemente produciendo resultados erróneos. Mapas, tablas y cartografía eran difíciles de mantener actualizados, ni siquiera con los mejores mapas, tablas o cartas, se podía imaginar cómo eran realmente las cosas, cuáles eran las mejores opciones de localización, y cuál era la decisión que se debía tomar.
- ✓ Es una necesidad la demanda de localización espacial de indicadores, datos socioeconómicos, estadísticos y medioambientales.
- ✓ La información estadística, medioambiental y cartográfica actualizada está en manos de empresas como GEOCUBA, CITMA y la Oficina Nacional de Estadísticas (ONE) pero no se encuentra estandarizada.
- ✓ La carencia de aplicaciones informáticas desarrollada con software libre (SL), en la nube, limita la colaboración al impedir compartir información y el trabajo grupal entre directivos de distintas empresas.

II. Problema científico

El **problema científico** de la presente investigación es el siguiente: ¿Cómo gestionar información estadística, medioambiental o indicadores en la nube de manera que facilite el proceso de toma de decisiones con la celeridad y precisión requerida?

III. Objetivo general

Como **objetivo general** se propone: Desarrollar una plataforma colaborativa en la nube, basada en SL, haciendo uso de tecnologías Web que permita prestar el servicio de gestionar y representar información estadística, medioambiental e indicadores geolocalizados y estandarizados.

IV. Objeto de estudio y campo de acción

Como **objeto de estudio** se define la gestión de la información estadística o medioambiental y como **campo de acción** el desarrollo de una plataforma colaborativa en la nube basada en SL y haciendo uso de tecnologías Web.

V. Objetivos específicos

- Determinar los fundamentos teóricos y prácticos referentes al desarrollo de una plataforma colaborativa en la nube.
- Definir las tecnologías de SL apropiadas para crear una plataforma colaborativa en la nube.
- Diseñar la arquitectura escalable y extensible para soportar una plataforma colaborativa en la nube.
- Implementar la versión inicial de un prototipo de arquitectura que brinde una solución al problema de investigación.

VI. Preguntas científicas

Las **preguntas científicas** que guiarán el desarrollo de la investigación, son:

- ¿Cuáles son los fundamentos teóricos y prácticos referentes al desarrollo de una plataforma colaborativa en la nube?
- ¿Cuáles son las tecnologías de SL apropiadas para crear una plataforma colaborativa en la nube?
- ¿Cómo diseñar la arquitectura escalable y extensible para soportar una plataforma colaborativa en la nube?
- ¿Cómo implementar la versión inicial de un prototipo de arquitectura que brinde una solución al problema de investigación?

VII. Métodos científicos

Los **métodos científicos** utilizados en la investigación fueron:

Métodos teóricos

1. Analítico sintético: El análisis facilitó comprender las partes integrantes desde distintas aristas (informática, ambiental, estadística y de la gestión de información) que sustenta la Web.
2. Histórico lógico: Se utilizó para el análisis de la evolución histórica en el empleo de las TIC como herramienta para creación de la plataforma colaborativa en la nube y el metodológico permitió utilizar este conocimiento en la elaboración de la arquitectura como posible solución al problema de investigación.
3. Modelación: Para diseñar una plataforma web colaborativa.

Métodos empíricos

1. Observación Científica: utilizada para obtener información acerca del problema planteado en la investigación.
2. Criterio de Expertos: utilizado para la validación de la propuesta de solución

de esta investigación.

3. Revisión de documentos: para el análisis de documentación relacionada con el tema de información estadística o medioambiental utilizada.

VIII. Aportes

El aporte práctico del trabajo es obtener una plataforma web colaborativa nacional en la nube sobre SL que permita prestar el servicio de gestionar y representar información estadística, medioambiental e indicadores geolocalizados y estandarizados.

IX. Estructura del trabajo de tesis

La tesis se estructura en tres capítulos, así como conclusiones, recomendaciones, referencias bibliográficas, bibliografía y anexos.

Capítulo 1: Fundamentación Teórica: Recoge los antecedentes teóricos que se necesitan para explicar de qué se encarga el presente trabajo. Incluye un marco conceptual sobre los SIG, la gestión de la información y una descripción de los lenguajes de programación en SL a utilizar para la implementación y diseño del sistema.

Capítulo 2: Diseño y modelación de la arquitectura: Se refiere al proceso de diseño de la Plataforma Tocaroro, partiendo de precedentes tecnológicos con aplicabilidad al diseño propuesto. Además se plantean los requisitos funcionales y no funcionales a tener en cuenta en la aplicación.

Capítulo 3: Implementación de la plataforma Web colaborativa en la nube: Se desarrolla la implementación de la solución de diseño propuesta para el desarrollo de la plataforma colaborativa Tocaroro, mostrando una versión inicial de un prototipo de arquitectura que brinde una solución al problema de investigación.

CAPÍTULO 1 Aspectos teóricos

1.1 Introducción

En este capítulo se realiza un estudio sobre los principales conceptos teóricos y prácticos referentes al desarrollo de una plataforma colaborativa. Se abordan los sistemas automatizados existentes vinculados al campo de acción, efectuando una comparación entre ellos. Además se definen las tecnologías de SL apropiadas para crear una plataforma colaborativa en la nube.

1.2 ¿Qué es una aplicación Web y Computación en la Nube?

Una Aplicación Web es aquella aplicación que los usuarios pueden utilizar accediendo de un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. La naturaleza cambiante de este tipo de página es la que le da el nombre de dinámica [1].

El término **Web 2.0** no se refiere a una tecnología o equipamiento nuevo: este fue acuñado por O'Reilly Media para referirse a una segunda generación de Web basada en comunidades de usuarios y una gama especial de servicios, como las redes sociales, los blogs y los wikis, que fomentan la colaboración y el intercambio ágil de información entre los usuarios.

1.2.1 ¿Computación en la nube?

El concepto de computación en la nube empezó en proveedores de servicio de Internet a gran escala, como Google, Amazon AWS, Microsoft que construyeron su propia infraestructura. De entre todos ellos emergió una arquitectura: un sistema de recursos distribuidos horizontalmente, introducidos como servicios virtuales de Tecnologías de la Información (TI) escalados masivamente y

manejados como recursos configurados y mancomunados de manera continua.

La **computación en la nube** concepto conocido también bajo los términos servicios en la nube, informática en la nube, nube de cómputo o nube de conceptos, del inglés "Cloud computing", es un paradigma que permite ofrecer servicios de computación a través de Internet [2].

En este tipo de computación todo lo que puede ofrecer un sistema informático se ofrece como servicio, de modo que los usuarios puedan acceder a los servicios disponibles "en la nube de Internet" sin conocimientos o al menos sin ser expertos en la gestión de los recursos que usan.

"Cloud computing" es un nuevo modelo de prestación de servicios de negocio y tecnología, que permite al usuario acceder a un catálogo de servicios estandarizados y responder a las necesidades de su negocio, de forma flexible y adaptativa, en caso de demandas no previsibles o de picos de trabajo, pagando únicamente por el consumo efectuado [3]. Es un concepto que incorpora el software como servicio, como en la Web 2.0 y otros conceptos recientes, como se aprecia en la Figura 1.

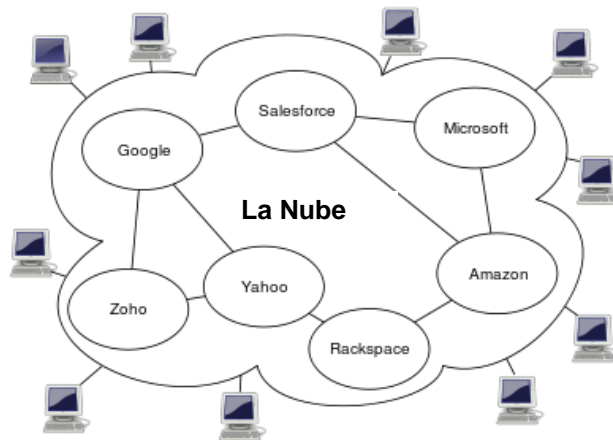


Figura 1.Computación en la nube.

1.2.1.1 **Ventajas**

- Integración probada de servicios Red: La tecnología de "Computación en la Nube" se puede integrar con mucha mayor facilidad y rapidez con el resto de sus aplicaciones empresariales.
- Prestación de servicios a nivel mundial.

- Una infraestructura 100% de "Computación en la Nube" permite al proveedor de contenidos o servicios en la nube prescindir de instalar cualquier tipo de hardware, ya que éste es provisto por el proveedor de la infraestructura o la plataforma en la nube.
- Implementación más rápida y con menos riesgos.
- Actualizaciones automáticas que no afectan negativamente a los recursos de TI.
- Contribuye al uso eficiente de la energía.

1.2.1.2 Tipos de nubes

Según Beyerlein los tipos de nubes son [4]:

- **Las nubes públicas** se manejan por terceras partes, y los trabajos de muchos clientes diferentes pueden estar mezclados en los servidores, los sistemas de almacenamiento y otras infraestructuras de la nube. Los usuarios finales no conocen qué trabajos de otros clientes pueden estar corriendo en el mismo servidor, red, discos como los suyos propios.
- **Las nubes privadas** son una buena opción para las compañías que necesitan alta protección de datos y ediciones a nivel de servicio. Las nubes privadas están en una infraestructura en demanda manejada por un solo cliente que controla qué aplicaciones debe correr y dónde. Son propietarios del servidor, red, y disco y pueden decidir qué usuarios están autorizados a utilizar la infraestructura.
- **Las nubes híbridas** combinan los modelos de nubes públicas y privadas. Usted es propietario de unas partes y comparte otras, aunque de una manera controlada.

1.3 ¿Qué es un GIS o SIG?

Un Sistema de Información Geográfica (SIG o GIS, en inglés Geographic Information System) es una integración organizada de hardware, software y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar en

todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión geográfica (Bolstad, 2005). También puede definirse como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestre y construido para satisfacer unas necesidades concretas de información. En un sentido más genérico, los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones.

La tecnología de los SIG puede ser utilizada para investigaciones científicas, la gestión de los recursos, gestión de activos, la arqueología, la evaluación del impacto ambiental, la planificación urbana, la cartografía, la sociología, la geografía histórica, el marketing, la logística (Bosque, 1992).

1.4 ¿Qué es un Software Colaborativo?

El término de Software Colaborativo hace referencia a los métodos y herramientas de software que facilitan el trabajo en grupo, mejorando su rendimiento, y contribuyen a que personas que están localizadas en puntos geográficos diferentes puedan trabajar a la vez, ya sea directamente o de forma anónima, a través de las redes [5]. Hoy en día existe un nuevo tipo de equipo "virtual", equipo formado por personas que se comunican electrónicamente.

1.4.1 Tipos de colaboración

Las **herramientas de comunicación electrónica** (Terán & Aldama, 2008) envían mensajes, archivos, datos o documentos entre personas que permiten compartir la información (colaboración asíncrona), como por ejemplo:

- Correo electrónico.
- Correo de voz.
- Publicación en web.

Las **herramientas de conferencia** facilitan compartir la información, de forma interactiva (colaboración síncrona), como por ejemplo:

- Conferencia de datos.
- Conferencias de voz.
- Conferencias de video (o audio conferencia).
- Salas de chat o mensajería instantánea.
- Sistemas para facilitar reuniones.

Las **herramientas de gestión colaborativa** facilitan las actividades del grupo, ejemplo:

- Calendarios electrónicos.
- Sistemas de gestión de proyectos.
- Sistemas de control de flujo de actividad.
- Sistemas de gestión del conocimiento.
- Sistemas de soporte a redes sociales.

1.4.2 Sistemas de Información Colaborativos

Los sistemas de información colaborativos se caracterizan por producir información y contenidos mediante la colaboración de muchas personas que suelen desconocerse entre sí [6].

Casi todos esos sistemas son el resultado de la llamada Web 2.0, por la que se ponen en marcha sistemas de información diseñados para que se pueda interactuar con ellos.

1.5 ¿Qué son mapas temáticos?

Los mapas temáticos representan cualquier fenómeno geográfico de la superficie terrestre. Persiguen objetivos bien definidos. Hacen referencia a la representación de ciertas características de distribución, relación, densidad o regionalización de objetos reales (vegetación, suelos, geología, etc.), o de conceptos abstractos (indicadores de violencia, de desarrollo económico, de calidad de vida, entre otros) (Robinson, 1982).

1.5.1 Tipos de mapas temáticos

Los diferentes mapas temáticos son [7]:

- Los **mapas de isolíneas** usan líneas curvas que unen puntos de igual valor de un fenómeno. También se llaman mapas de isopletras.
- Los **mapas de coropletras** utilizan colores o tramas para representar el área que ocupa un hecho. La intensidad del color o de la trama debe ser proporcional a la importancia del fenómeno.
- Los **mapas de diagramas** superponen un mapa diagramas de barras, de sectores, etc.
- Los **mapas de figuras (cartogramas)** utilizan figuras geométricas para localizar fenómenos puntuales.
- Los **mapas anamórficos** cambian el tamaño real de los países para hacerlo proporcional al hecho que cartografían, es decir, a la magnitud del fenómeno representado.
- Los **Mapas de flujos** son líneas de diferente espesor para representaciones dinámicas.
- Los **Mapas de relieve estadístico** son cartografías elaboradas por procedimientos informáticos de gran impacto visual; el valor de cada fenómeno se representa mediante relieves en forma de picos o depresiones, según dicho valor supere o no la media del fenómeno analizado.

1.6 ¿Qué es Espacio de Datos?

Un espacio de datos es un espacio o directorio que le permite a un usuario recoger varias fuentes de datos y recursos para trabajar con ellos como una unidad cohesiva. Este agrupa toda la información geográfica (servidor de mapas, capa base, filtros) a utilizar para la representación de la información temática (información asociada a indicadores, datos estadísticos o medio ambientales).

1.7 ¿Qué es Fuente de Datos?

Es una colección organizada de datos, es la estructura que agrupa la información (información temática) que el usuario sube al servidor y referente a un origen común. Por ejemplo: fuente de datos “Población de sexo masculino por edad” agrupa toda la información temática referente a “de 5 a 9 años”, “de 10 a 14 años”...“de 60 a 64 años”.

1.8 ¿Qué son Campos Temáticos?

Es la representación organizacional de la información que el usuario sube al servidor, del ejemplo anterior serían: “de 5 a 9 años” o “de 10 a 14 años” o “de 60 a 64 años”.

1.9 ¿Qué es información georeferenciada?

La información espacial, también conocida como información georeferenciada o espacialmente referenciada, es la información relacionada con el espacio geográfico, con la tierra; es la información que determina la ubicación de un elemento geográfico en particular [8]. Cada elemento en la tierra, tiene una ubicación única en un momento determinado, esto es, establecer las coordenadas X, Y, Z (o alguna otra forma de identificación única como por ejemplo calles y carreteras, código ZIP o algún otro tipo de identificación) que indican la posición espacial de un elemento en un determinado tiempo.

1.9.1 Diferencia entre Información tabular y espacial

La Información no **espacial o tabular**, es la que asociada a un elemento en particular, indica una o varias de sus características particulares que son importantes para un análisis en particular.

Cuando se habla de **información espacial**, se establece una ubicación única y definida para el elemento geográfico en particular, esto es, utilizar coordenadas o direcciones de calles y carreteras para identificar plenamente la ubicación exacta

del elemento.

La **información tabular** indica únicamente las características del elemento, pero combinando la información tabular con la espacial, se puede describir completamente un elemento, no sólo en sus características asociadas, sino en su posición específica en el espacio.

1.10 ¿Quién es GEOMIX?

La **Empresa Cartografía y Soluciones Geomáticas GeoSÍ** es la entidad perteneciente al Grupo Empresarial GEOCUBA responsable de la creación, mantenimiento y provisión del patrimonio cartográfico nacional de la República de Cuba, especializada en la elaboración de cartografía de pequeñas, medianas y grandes escalas, con servicios de valor agregado mediante el desarrollo de aplicaciones geoinformáticas de escritorio y en web para el gobierno, la economía y la sociedad en general. Es la encargada de garantizar tecnológicamente el desarrollo de la Infraestructura de Datos Espaciales de la República de Cuba para compartir y proveer información geográfica de GEOCUBA y de otros proveedores de datos geográficos.

Certificada por la Norma Cubana ISO 9001: 2008, con alcance a los procesos asociados a la Cartografía Digital, la Fotogrametría y el Desarrollo de Software, GeoSÍ asume el firme compromiso de actuar conforme a los requerimientos de calidad de sus clientes promoviendo la mejora continua de sus actividades y procedimientos, así como la excelencia y profesionalidad de sus servicios.

La empresa está integrada por una Dirección General, cinco direcciones funcionales, una Unidad Básica de Servicios y seis Unidades Empresariales de Base (UEB) o agencias productivas:

- Agencia de Cartografía.
- Agencia de Fotogrametría.
- Agencia de Ventas.
- Agencia de Proyectos y Aplicaciones de SIG.
- Agencia de Servicios de Control de Flota.

- Agencia GEOMIX.

1.11 Misión y visión de GEOMIX

La agencia Geomix tiene como **misión** “Proveer y comercializar cartografía y servicios asociados, así como conocimientos y soluciones geomáticas con alto nivel de integración a los sistemas informativos de sus clientes.”

Visión

- Los productos y servicios que brinda la empresa, incluida la provisión de sus datos, están al nivel del estado del arte y debidamente certificados, logrando la plena satisfacción de sus clientes y un impacto creciente en la sociedad.
- Se extienden servicios hacia el exterior y se logra la inserción de expertos de la empresa en proyectos y grupos de trabajo internacionales.
- Se satisfacen las necesidades de la Defensa, de la Economía y en particular del Sistema de Gestión y Control de Flota del país.
- Se multiplica la capacidad de la empresa como resultado de alianzas con universidades y empresas afines. Se mantienen los niveles crecientes de utilidades que garantizan la salud financiera y el mejoramiento de las condiciones de vida y trabajo, incluida la modernización tecnológica. Se ha alcanzado la categoría de empresa eficiente.
- Se dispone de personal con experiencia, motivado y trabajando con profesionalidad.

1.12 Descripción de los sistemas existentes

Cada vez es más usual encontrar en Internet redes sociales verticales o especializadas en una temática concreta, tal es el caso de ArcGis.com, creada por la compañía ESRI (Environmental Systems Research Institute) con el objetivo de que profesionales y/o potenciales profesionales del sector puedan trabajar y compartir conocimientos sobre herramientas y aplicaciones SIG. Existen muchos ejemplos de aplicaciones SIG que de cierta manera contienen semejanza con la

plataforma colaborativa ejemplos de ellos se encuentra el conocidísimo Google Maps¹, Gvsig. También con la pretensión de crear comunidades en torno al mundo de los mapas, en el año 2004 se lanzó la Open Street Map², conocida como “Wiki Mapamundi Libre”, este es el más claro ejemplo de proyecto de trabajo cartográfico colaborativo. Recientemente, una startup gallega ha lanzado la versión beta de ikiMap.com, se trata de un servicio on-line que permite crear y compartir mapas de manera rápida e intuitiva, y que busca convertirse en un lugar de referencia en la búsqueda de información geográfica. En el entorno nacional se encuentran sistemas como el SisGIM (SIG que se integra a un Sistema de Gestión de Indicadores). Por su parte Tocatoro no se trata de una red social, pero la filosofía de crear, editar, compartir y publicar información a través de la red, incluyendo mapas, son su principal meta como plataforma colaborativa.

1.13 ¿Qué es un servidor de Mapas?

Un servidor de mapas (en inglés Internet Map Server, IMS) provee mapas o cartografía³ a través de Internet, el uso de la Web como medio de difundir mapas es sin duda un gran avance para la cartografía, facilitando el proceso de publicación de Información Geográfica actualizada, en tiempo real, y de forma más barata a cualquier parte del mundo [10].

En esta investigación se utiliza como servidor geográfico GeoServer. A continuación se abarcarán algunas características esenciales del mismo.

1.13.1 GeoServer

Es un servidor de código abierto escrito en Java - permite a los usuarios compartir y editar datos geoespaciales. Diseñado para la interoperabilidad y publicación de datos de cualquier fuente de datos espacial usando estándares abiertos[11].

¹ Sitio Oficial: www.maps.google.es

² Sitio Oficial: www.openstreetmap.org/

³Cartografía.- referente a mapas o ciencia que se ocupa de los mapas y su realización.

GeoServer ha evolucionado hasta llegar a ser un método sencillo de conectar información existente a servicios de mapas tales como Google Earth y NASA World Wind. Está desarrollado sobre *Geotools*, una biblioteca de clases desarrollada en Java que implementa los estándares de SIG.

1.13.1.1 Características principales

Según el artículo (OsGeoLive, 2010) de la Revista OsGeoLive entre las principales características de GeoServer se pueden citar algunas como:

- Fácil utilización a través de la herramienta de administración vía Web.
- Soporte amplio de formatos de entrada PostGIS, Shapefile, ArcSDE y Oracle. VFP, MySQL, MapInfo y WFS en cascada también están entre los formatos de entrada soportados.
- Soporte de formatos de salida tales como JPEG, GIF, PNG, SVG y GML.
- Soporte completo de SLD (Style Layer Description) para la personalización de los mapas.
- Implementación de los protocolos WFS (Web Features Service), WMS (Web Maps Service).
- Puede funcionar en cualquier contenedor de servlet.
- Diseñado para ser compatible con extensiones.
- Facilidad de escritura de nuevos formatos de datos con la interfaz de almacenamiento de datos GeoTools.

Además, GeoServer incluye un cliente integrado **OpenLayers** para previsualizar capas de datos.

1.14 ¿Qué es Openlayers?

OpenLayers es una biblioteca de JavaScript de código abierto bajo una derivación de la licencia BSD⁴ para mostrar mapas interactivos en los navegadores Web [12].

⁴BSD: Licencia de SL otorgada principalmente para los sistemas BSD (Berkeley Software Distribution).

OpenLayers ofrece un API⁵ para acceder a diferentes fuentes de información cartográfica en la red: WMS, Mapas comerciales (tipo Google Maps, Bing, Yahoo), WFS, distintos formatos vectoriales, mapas de OpenStreetMap, etc.

1.15 Sistemas Gestores de Bases de Datos. (SGBD)

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales que ha sido desarrollado de varias formas desde 1977. Comenzó como un proyecto denominado Ingres en la universidad Berkeley de California. Fue más tarde desarrollado comercialmente por la Relational Technologies/Ingres Corporation. En 1996, debido a un nuevo esfuerzo de código abierto y a la incrementada funcionalidad del software, Postgres fue renombrado a PostgreSQL. El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto. PostgreSQL está ampliamente considerado como el sistema de Bases de Datos (BD) de código abierto más avanzado del mundo [13].

1.15.1 ¿Por qué PostgreSQL?

A continuación se muestra algunas de las principales características tomadas en cuenta:

- **DBMS Objeto-Relacional:** Aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas tales como consultas SQL declarativas, control de concurrencia multiversión, soporte multiusuario, optimización de consultas, herencia, y arreglos.
- **Alta concurrencia:** Mediante el sistema denominado Acceso concurrente multiversión (MVCC) permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

⁵API: Interfaz de Programación de Aplicaciones, en inglés Application Programming Interface. Conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software.

- **GIS:** *PostGIS*, extensión que añade soporte de objetos geográficos a PostgreSQL y permite realizar análisis mediante consultas SQL espaciales o mediante conexión a aplicaciones SIG.
- **Búsqueda de texto:** En inglés full text search es incluido en el núcleo a partir de la versión 8.3. Vía Tsearch2 y OpenFTS para versiones anteriores.
- **Soporte de Esquemas (Schemas):** Tiene soporte para esquemas.
- **Tecnologías de SL y Open Source:** PostgreSQL es SL, publicado bajo la licencia BSD por una comunidad denominada PostgreSQL Global Development Group (PGDG).

1.16 Apache Tomcat

Apache Tomcat ⁶ funciona como un contenedor de servlets⁷ desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems. Tomcat **no es un servidor de aplicaciones**, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. [14].

Tomcat puede funcionar como servidor Web por sí mismo. Es usado como servidor Web autónomo en entornos con alto nivel de tráfico y alta disponibilidad. Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

⁶Apache Tomcat: también llamado Jakarta Tomcat o simplemente Tomcat.

⁷**Servlets:** son objetos que corren dentro y fuera del contexto de un contenedor de servlets (ej: Tomcat) y extienden su funcionalidad.

1.17 ¿Qué es Java?

Java originalmente fue denominado "Oak". Sus inicios datan de 1991 cuando James Gosling (en Sun Microsystems) encabezó un proyecto cuyo objetivo original era implementar una máquina virtual ampliamente portable y un lenguaje de programación ambos orientados a dispositivos "embedded" (procesadores incorporados en diversos dispositivos de consumo masivo como tostadoras, teléfonos móviles, etc.). La sintaxis del lenguaje heredó características de C y C++[15].

Con el auge de Internet, pareció natural aprovechar este lenguaje para desarrollar aplicaciones distribuidas y portables. La primera implementación de Java data de 1995 y pronto los "navegadores Web" incorporaron soporte Java para la ejecución de pequeñas aplicaciones interactivas (Applets).

En realidad, Java hace referencia a un conjunto de tecnologías entre las cuales el lenguaje Java es sólo una de ellas.

1.17.1 Ventajas de Java.

Java es un lenguaje que ha sido diseñado para producir software [16]:

- Confiable: Minimiza los errores que se escapan a la fase de prueba.
- Seguro: Applets recuperados por medio de la red no pueden causar daño a los usuarios.
- Orientado a objetos: Beneficioso tanto para el proveedor de bibliotecas de clases como para el programador de aplicaciones.
- Robusto: Los errores se detectan en el momento de producirse, lo que facilita la depuración.
- Puede integrar componentes de Internet y BD remotas.
- El uso de java no se limita a manejo de animaciones, audio o video.
- Portable: Los mismos binarios funcionan correctamente en Windows/95 y /NT 2000, XP, Unix/Motif y Power/Mac.

1.17.2 Desventajas de Java.

Así como este lenguaje tiene ventajas, también tiene desventajas, una de ellas es que puede ser un lenguaje de ejecución lenta, debido al uso de la máquina virtual de Java, a diferencia de otros lenguajes de programación de más bajo nivel como lo es "C", su velocidad de ejecución disminuye drásticamente al compararse con este lenguaje. Otra desventaja es que Java es considerado un lenguaje difícil de aprender, esto debido a su compleja sintaxis, sin embargo la estructura de este lenguaje es completa, organizada y semejante a el lenguaje de C++, por lo que si se tiene conocimiento previo en C++, Java no debería de ser un lenguaje difícil de aprender [17].

1.17.3 ¿Por qué Java y no otros lenguajes de programación?

Podemos decir que se escogió Java debido a sus principales ventajas, entre estas tenemos que Java es un lenguaje **multiplataforma**, lo que quiere decir que se ejecuta en la mayoría de los sistemas operativos, inclusive en sistemas operativos móviles. Otra ventaja es que java es un **software de distribución libre**, no es necesario pagar una licencia para poder comenzar a desarrollar en este lenguaje. Así mismo es un **lenguaje muy completo y poderoso**, se pueden realizar muchas tareas con él, pues posee una biblioteca y utilidades muy completas que facilitan la programación.

1.18 Modelo-Vista-Presentador (MVP).

El MVP es derivado del patrón de software Modelo-Vista-Controlador (MVC) [18].

El MVP es la solución a los siguientes dos grandes problemas que se presentan con gran frecuencia en el desarrollo del software:

- *Realizar pruebas automáticas sobre la interfaz de usuario (unit test):* El patrón nos permite realizar pruebas automáticas (unit test) utilizando un objeto que simula la interfaz de usuario (Vista), para esto la misma debe implementar una interfaz programática que será luego la misma que implementará el objeto simulador que se utiliza para realizar los unit test.

- *Independizar a la aplicación del tipo de interfaz de usuario elegida:* El patrón permite independizar la aplicación de la interfaz de usuario, así se puede lograr que la misma funcione tanto como una aplicación de escritorio (desktop), como una aplicación Web o como una aplicación móvil con un mínimo de impacto, también se puede lograr que la aplicación de escritorio tenga varias vistas de un mismo.

En la Figura 2 se muestra gráficamente la filosofía del modelo.

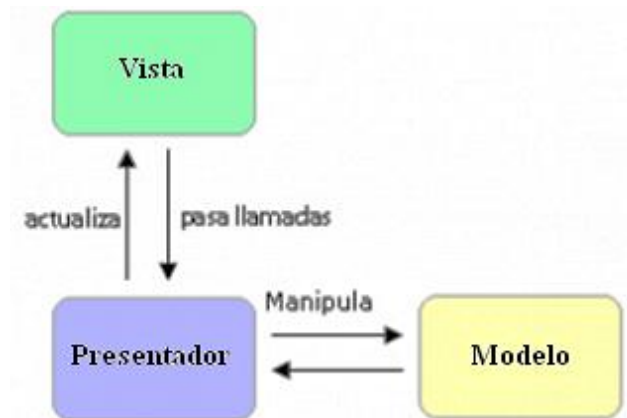


Figura 2. Modelo vista Presentador.

La idea básica del MVP es que la clase Presentador haga de intermediario entre la Vista (interfaz gráfica de usuario) y el modelo de datos.

El Presentador recibe las interfaces que deben implementar el modelo y la vista, con los métodos públicos a los que el Presentador debe llamar.

- El **modelo** funciona como una interface que define los datos que serán mostrados luego en la vista.
- La **vista** es una interface que nos muestra los datos del modelo y lanza llamadas a la capa presentación para que luego actúe con los datos.
- La **presentación** recupera los datos del modelo y actualiza la vista.

Se puede decir que el patrón MVP es una mejora del patrón MVC debido a las siguientes características:

- La vista no conoce el modelo.

- El presentador es independiente de la tecnología de interfaz de usuario.

1.19 ¿Qué son los marcos de trabajo (frameworks)?

Un marco de trabajo en términos de software es una entidad unitaria formada por códigos, scripts y clases, funciones que ofrecen una funcionalidad genérica que es especializada o extendida por el código generado por el usuario para obtener una funcionalidad específica.

Según (Sache, 2010) las ventajas de usar marcos de trabajo para el desarrollo de aplicaciones Web son:

1. Implementan patrones arquitectónicos MVC, lo cual representa una buena práctica para la modularización de la aplicación, para rehusar código y permitir el uso de varias interfaces con el usuario.
2. Posibilita la creación de aplicaciones con un punto exclusivo de entrada.
3. Limita la duración de la aplicación por el abastecimiento de algunas funciones comunes, necesarias para aplicaciones complicadas.
4. El sistema cache permite reducir los recursos usados en el servidor para generar respuestas al cliente.

1.20 ¿Qué es Object Relational Mapping (Mapeo relacional de Objetos)?

El Mapeo Relacional de Objetos (en inglés *Object-Relational Mapping*, ORM) es una técnica de programación para convertir datos entre tipos incompatibles de sistemas de BD relacionales y lenguajes de Programación Orientada a Objetos (POO). Creando en efecto una BD virtual orientada a objetos que puede ser usada desde adentro sobre la BD relacional. Esta transformación permite el uso de las bondades de la POO (El-Bakry & N, 2009).

Esta característica de los ORM les provee de ciertas ventajas para su utilización (Alvial-Cid, Saavedra-Quevedo, & Valenzuela-Parada, 2010).

- **Rapidez en el desarrollo:** La mayoría de las herramientas actuales

permiten la creación del modelo por medio del esquema de la BD, leyendo el esquema, crea el modelo adecuado.

- **Abstracción de la BD:** Al utilizar un ORM, se consigue separar totalmente del sistema de BD que se utiliza.
- **Reutilización:** Permite utilizar los métodos de un objeto de datos desde distintas zonas de la aplicación, incluso desde aplicaciones distintas.
- **Seguridad:** Los ORM suelen implementar sistemas para evitar tipos de ataques como pueden ser los SQL injections.
- **Mantenimiento del código:** Debido a la correcta ordenación de la capa de datos, haciendo que el mantenimiento del código sea mucho más sencillo.
- **Lenguaje propio para realizar las consultas:** Estos sistemas de mapeo traen su propio lenguaje para hacer las consultas, lo que hace que los usuarios dejen de utilizar las sentencias SQL para que pasen a utilizar el lenguaje propio de cada herramienta.

Los ORM facilitan la manipulación de las BD por parte de las aplicaciones disminuyendo la presencia del desarrollador al manipular los datos.

1.20.1 Hibernate

Hibernate es un framework que implementa el ORM para la plataforma Java que facilita el mapeo de atributos entre una BD relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones de las entidades que permiten establecer estas relaciones [19]. Hibernate es SL, distribuido bajo los términos de la licencia GNU/LGPL.

1.20.1.1 Características

Hibernate permite a la aplicación manipular los datos de la BD operando sobre objetos, con todas las características de la POO. Convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Además genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los

motores de BD con un ligero incremento en el tiempo de ejecución.

Está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una BD ya existente. También tiene la funcionalidad de crear la BD a partir de la información disponible.

Hibernate ofrece también un lenguaje de consulta de datos llamado *Hibernate Query Language* (HQL), al mismo tiempo que una API para construir las consultas programáticamente (conocida como "*criteria*").

1.20.2 Hibernate Search

Hibernate Search es uno de los motores de búsqueda de texto más completo, que combina la potencia de Hibernate para la gestión de BD, con la potencia de Apache Lucene como motor de búsquedas textuales basado en índices inversos. Gracias a Hibernate Search se unifican estas dos grandes herramientas unificando manejo e interfaces y resulta trivial dotar de esta funcionalidad a la aplicación.

1.21 ¿Qué es Inversión de Control?

Inversión de control (Inversion of Control en inglés, **IoC**) es un método de programación en el que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales, en los que la interacción se expresa de forma imperativa haciendo llamadas a procedimientos (procedure calls) o funciones (Jan, Aleksa, Anirvan, & Jessica, 2008).

Tradicionalmente el programador especifica la secuencia de decisiones y procedimientos que pueden darse durante el ciclo de vida de un programa mediante llamadas a funciones. En su lugar, en la inversión de control se especifican respuestas deseadas a sucesos o solicitudes de datos concretas, dejando que algún tipo de entidad o arquitectura externa lleve a cabo las acciones de control que se requieran en el orden necesario y para el conjunto de sucesos que tengan que ocurrir.

1.21.1 ¿Qué es Spring?

El **Spring Framework** (también conocido simplemente como **Spring**) es un framework de código abierto de desarrollo de aplicaciones para la plataforma Java (Rubio, 2009). A pesar de que Spring Framework no obliga a usar un modelo de programación en particular, se ha popularizado en la comunidad de programadores en Java al ser considerado como una alternativa y sustituto del modelo de Enterprise JavaBeans. Por su diseño el framework ofrece mucha libertad a los desarrolladores en Java y soluciones muy bien documentadas y fáciles de usar para las prácticas comunes en la industria.

1.21.2 ¿Qué es Spring Security?

Spring Security proporciona servicios de seguridad para aplicaciones de software empresariales basados en J2EE, enfocado particularmente sobre proyectos contruidos usando Spring Framework (Giametta, 2009).

La seguridad comprende dos operaciones: La primera operación es conocida como "autenticación", por el cual se establece si un usuario(que quiere realizar una acción en la aplicación) es quien dice ser, y la segunda operación es llamada "autorización" que se refiere al proceso de decidir si a un usuario le es permitido realizar una determinada acción en la aplicación.

En el nivel de "autenticación" Spring Security soporta muchos modelos de autenticación, proporciona su propio conjunto de características de autenticación.

1.22 AJAX como tecnología

AJAX, siglas de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo Web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla, lo que se traduce en aumento de la interactividad y velocidad (Wikimedia Foundation, Inc, 2007).

AJAX es una combinación de tecnologías ya existentes:

- **XHTML (o HTML) y hojas de estilos en cascada (CSS)** para el diseño que acompaña a la información.
- **Document Object Model (DOM)** accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- **Objeto XMLHttpRequest** para el intercambio de datos asincrónicamente con el servidor Web. (En algunos frameworks y bajo situaciones concretas, se usa un objeto “iframe” en lugar del XMLHttpRequest a la hora de realizar dichos intercambios).
- **XML** como formato usado comúnmente para la transferencia de datos entre el cliente y el servidor (aunque cualquier formato puede funcionar, incluyendo HTML pre formateado, texto plano, JSON y hasta EBML).

1.23 El término AJAX Framework

El término “**ajax framework**” no es más que una biblioteca de clases para aplicaciones Web que ayuda a desarrollar aplicaciones basadas en AJAX. La información es leída desde el servidor o enviada a éste a través de peticiones Javascript. Sin embargo, es requerido algún procesamiento del lado del servidor para manejar peticiones, por ejemplo, para buscar o guardar información.

Esto se resuelve fácilmente con el uso de un framework dedicado a procesar peticiones AJAX (Garrett, 2005). Este motor AJAX pretende reducir la espera para el usuario cuando una página trata de acceder al servidor.

1.24 Ext JS Framework y RichFaces.

Es un excelente Framework de AJAX, basado en JavaScript que puede utilizar como adaptadores (adapters) las librerías de *Yahoo User Interface* (YUI) o JQuery. Ext JS es también es una excelente biblioteca de componentes de Interfaz de Usuario (en inglés *User interface*, UI). Ext JS llega a ser una herramienta muy útil

para desarrollar aplicaciones web rica (en inglés *Rich Internet Applications*, RIA), sin dejar atrás las facilidades en la comunicación entre el servidor y los clientes (Jsakalos, 2007).

RichFaces: Es una biblioteca de código abierto basada en Java que permite crear aplicaciones Web con Ajax. Construye sobre el framework de JSF. La singularidad del planteamiento que ofrece es que la petición Ajax provoca una ejecución en el servidor y finalmente una renderización parcial o total de la página del navegador. Así, el control de lo que sucede está en el servidor. Richfaces está desarrollado y mantenido por JBoss [20].

Entre las características de RichFaces se encuentran las siguientes:

- Incluye funcionalidades Ajax, de modo que nunca se ve el JavaScript y tiene un contenedor Ajax propio.
- Contiene un conjunto de componentes visuales, los más comunes para el desarrollo de RIA, con un número bastante amplio que cubren casi todas las necesidades.
- Soporta facelets.
- Soporta Css.
- Es un proyecto Código Abierto activo y con una comunidad también activa.

1.25 ¿Qué es GWT?

GWT o **Google Web Toolkit** es un entorno de desarrollo Java, basado en SL y que permite escribir aplicaciones AJAX fácilmente. GWT permite escribir las aplicaciones en el lenguaje de programación JAVA y luego se encarga de compilarlo traduciendo la parte del cliente a lenguaje de programación JavaScript, HTML y CSS generando código JavaScript más eficiente que el escrito a mano. *Frameworks* creado por Google que permite ocultar la complejidad de varios aspectos de la tecnología AJAX [21].

Características

- Componentes gráficos dinámicos y reusables.
- Simple mecanismo RPC⁸.
- Gestión del historial del navegador web.
- Soporte para depurado de Java.
- Integración con JUnit.
- Internacionalización.
- Los desarrolladores pueden mezclar código escrito en Javascript dentro del código Java usando la Interfaz Nativa Javascript (JSNI).
- Soporte para las API de Google (inicialmente, soporte para Google Gears).
- Código abierto.
- Los desarrolladores pueden diseñar y desarrollar sus aplicaciones orientadas a objetos. Errores comunes en Javascript, como la discrepancia de tipos de datos, son controlados en tiempo de compilación.
- Existen un numeroso conjunto de bibliotecas desarrolladas por Google y terceros que amplían las funcionalidades de GWT.

Mvp4g es una gran biblioteca para desarrollar aplicaciones con el frameworks GWT, usando la arquitectura MVP. Sin utilizar Mvp4g se puede realizar una aplicación en GWT pero se necesita gestionar y crear un conjunto de clases que hacen que el código no sea tan organizado y legible, además se tendrá que gestionar el registro y la creación de instancias de las clases [22].

1.26 ¿Qué es un Plug-in?

Un Plug-in es una extensión de una aplicación que se integra para aportarle una nueva funcionalidad a esta y generalmente muy específica. Esta extensión

⁸ RPC: **Llamada a Procedimiento Remoto**, protocolo que permite a un programa de ordenador ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambos

adicional es ejecutada por la aplicación principal e interactúa por medio de la API [23].

Los Plug-in permiten:

- Que los desarrolladores externos colaboren con la aplicación principal extendiendo sus funciones.
- Reducir el tamaño de la aplicación.
- Separar el código fuente de la aplicación a causa de la incompatibilidad de las licencias de software.

En general, cualquier aplicación puede añadir soporte para complementos.

Las aplicaciones informáticas soportan plug-ins por muchas razones. Algunas de las principales razones incluyen:

- Habilitar la posibilidad que desarrolladores de terceras partes puedan crear plug-ins los cuales permitan extender una aplicación.
- Soportar fácilmente el agregar nuevas funcionalidades.
- Reducir el tamaño de una aplicación.
- Separar el código fuente de una aplicación por la existencia de incompatibilidades de licencias de software.

1.27 RUP

Rational Unified Process (RUP) es un proceso de desarrollo de software, en otras palabras, es un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Es un marco de trabajo genérico que puede especializarse, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos [24].

Fue creado por un grupo de estudiosos de la Ingeniería de *Software* formado por: Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1998. Es un proceso basado en componentes y utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software. No obstante, los verdaderos

aspectos definitorios de RUP se resumen en tres frases clave: está dirigido por casos de uso, centrado en la arquitectura y es iterativo e incremental.

UML

Unified Modeling Language (UML) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Fue creado por el mismo grupo de expertos que crearon RUP: Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1995.

UML pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. No es un lenguaje de programación sino un lenguaje de propósito general para el modelado orientado a objetos. UML es también un lenguaje de modelamiento visual que permite una abstracción del sistema y sus componentes [25].

1.28 Conclusiones

En este capítulo se realizó un estudio sobre los principales conceptos teóricos y prácticos referentes al desarrollo de una plataforma colaborativa. Se abordaron diferentes sistemas automatizados existentes efectuándose una comparación entre ellos. Además se definieron las tecnologías de SL apropiadas para crear una plataforma colaborativa en la nube.

CAPÍTULO 2 Diseño y modelación de la arquitectura

2.1 Introducción

Este capítulo presenta la descripción de la BD empleada, así como el Diagrama Entidad-Relación. Se definen cuáles son las tecnologías aplicadas para el desarrollo de la plataforma. Se analizan los requerimientos funcionales y no funcionales del sistema, que darán solución al problema planteado. Se realiza la propuesta del sistema así como los casos de uso del sistema y sus actores. Para el modelado se utilizó Rational Rose, con el objetivo de representar los diagramas de casos de uso del sistema y la especificación de los mismos.

2.2 Base de Datos

Tocororo es una plataforma desarrollada enteramente sobre las tecnologías de SL y Código Abierto, siendo este el principal motivo por el cual el Gestor de BD utilizado en la plataforma es PostgreSQL. Seguidamente se exponen algunos aspectos que justifican el por qué usarlo para dar solución a la problemática planteada.

2.2.1 Problemática

Tocororo es una plataforma que se puede desarrollar dentro de diferentes entornos: Empresarial, Intranet e Internet. Estos dos últimos además de ser un sistema orientado a colaboración y permitir el registro de usuarios, el volumen de información a manejar puede crecer exponencialmente, motivo por el cual la arquitectura debe manejar temas importantes no solo relacionados con la seguridad sino también con la eficiencia y el rendimiento a la hora de procesar la información.

2.2.2 Criterios evaluados.

1. La cantidad de usuarios puede crecer exponencialmente:
 - a) Uso de una estrategia organizacional de varias BD.
 - b) Uso de una estrategia organizacional utilizando Esquemas de BD

(Schemas).

2. Uso de metadatos para resolver la problemática de que los usuarios puedan definir datos, tipos de datos, etc.
3. Gestión de grandes volúmenes de datos, gestión de concurrencia, velocidad de procesado y soporte para SIG.
4. Búsqueda y recuperación fácil de la información.
5. Seguridad sobre la información del usuario.
6. Arquitectura de N Capas.
7. Arquitectura orientada a módulos.
8. Visualización de la información georeferenciada y la información temática.
9. Extensibilidad mediante Plug-ins.
10. Internacionalización.
11. Uso de Tecnologías de SL y Código Abierto.

2.2.3 Uso de Esquemas y Metadatos.

Debido a que PostgreSQL manipula y gestiona eficientemente los esquemas, esta fue una mejor solución para la plataforma Tocatoro que el uso de muchas BD (una por cada usuario registrado en el sistema). El uso de esquemas es la solución dada para resolver la problemática del futuro mantenimiento de la BD, teniendo un esquema de datos por cada usuario registrado (el nombre del esquema es el nombre de usuario), donde los usuarios pueden almacenar sus datos (datos asociados a las fuentes de datos, indicadores, datos estadísticos o medio ambientales entre otros) y por su parte el sistema deberá crear dinámicamente una tabla en el esquema de datos del usuario.

Los indicadores, la información estadística o la información medio ambiental (conocido en Tocatoro como los campos temáticos asociados a fuentes de datos, y entidades geográficas) se traducen en campos dinámicos en la tabla asociada a la fuente de datos. La descripción y organización de la información de las tablas de los usuarios y los campos que se van generando dinámicamente así como su manipulación se realiza mediante el uso de metadatos.

Finalmente la representación jerárquica de la información es la que se muestra en la Figura 3:

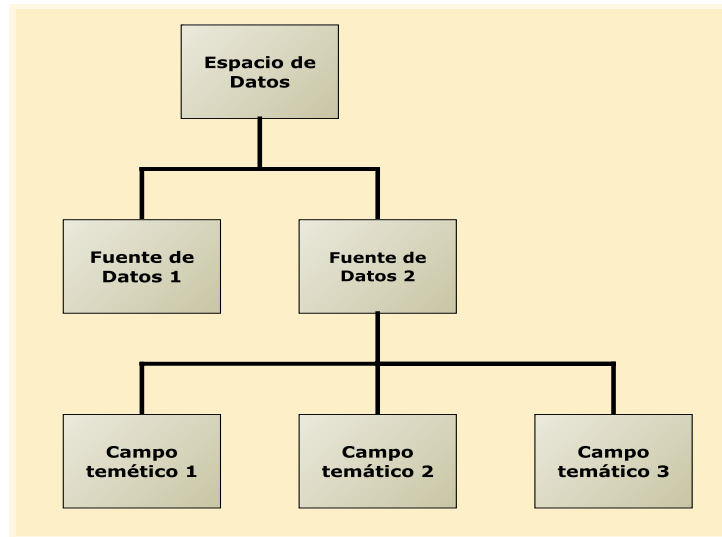


Figura 3. Organización jerárquica.

La Figura 4 muestra el uso de esquemas para dar solución a la problemática organizacional.

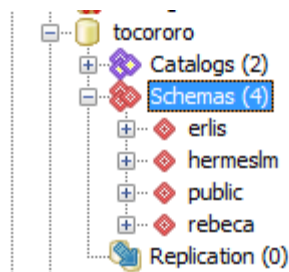


Figura 4. Uso de Esquemas.

Los metadatos son especificados en el esquema "public" que es el esquema por defecto y que contiene todas las tablas que describen la estructura de los datos de los usuarios, tanto como los espacios de datos, fuentes de datos, campos y mapas de los mismos, en la Figura 5 se muestra la estructura del esquema "public":

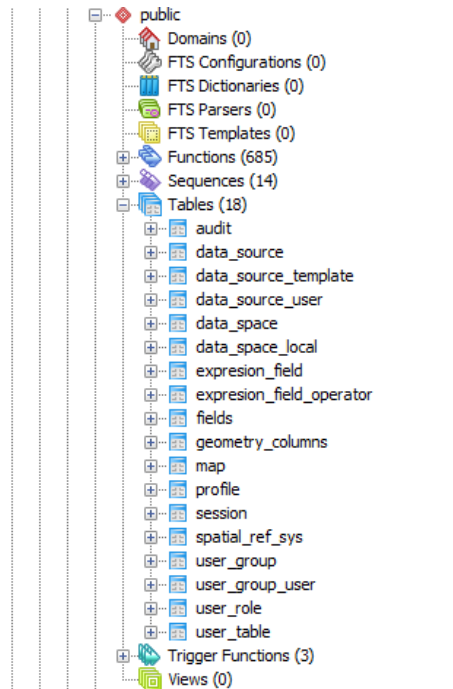


Figura 5. Estructura del esquema "public".

Por último, en la Figura 6, se muestra el esquema de un usuario que muestra las tablas de datos del usuario asociadas a cada fuente de datos, nótese que los nombres de las tablas creadas dinámicamente toman como patrón el nombre de la fuente de datos seguido de la fecha y hora de creación de la tabla (esto se define así para lograr una mejor administración de la BD).

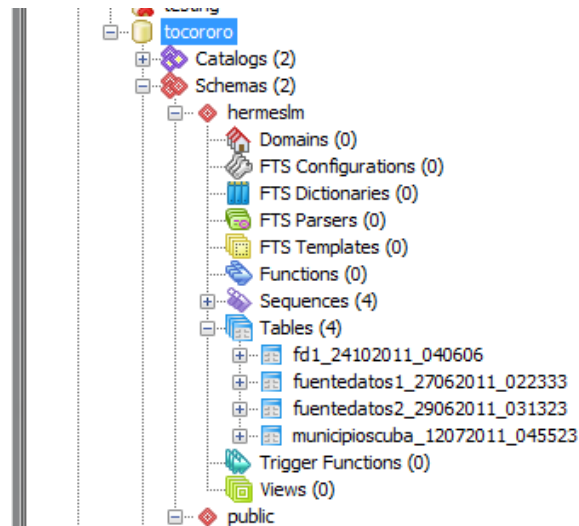


Figura 6. Esquema de un usuario.

2.2.4 Modelo Entidad-Relación de las principales tablas

El diagrama entidad relación que se muestra en Figura 7 brinda información detallada de las principales tablas de la plataforma Tocatoro.

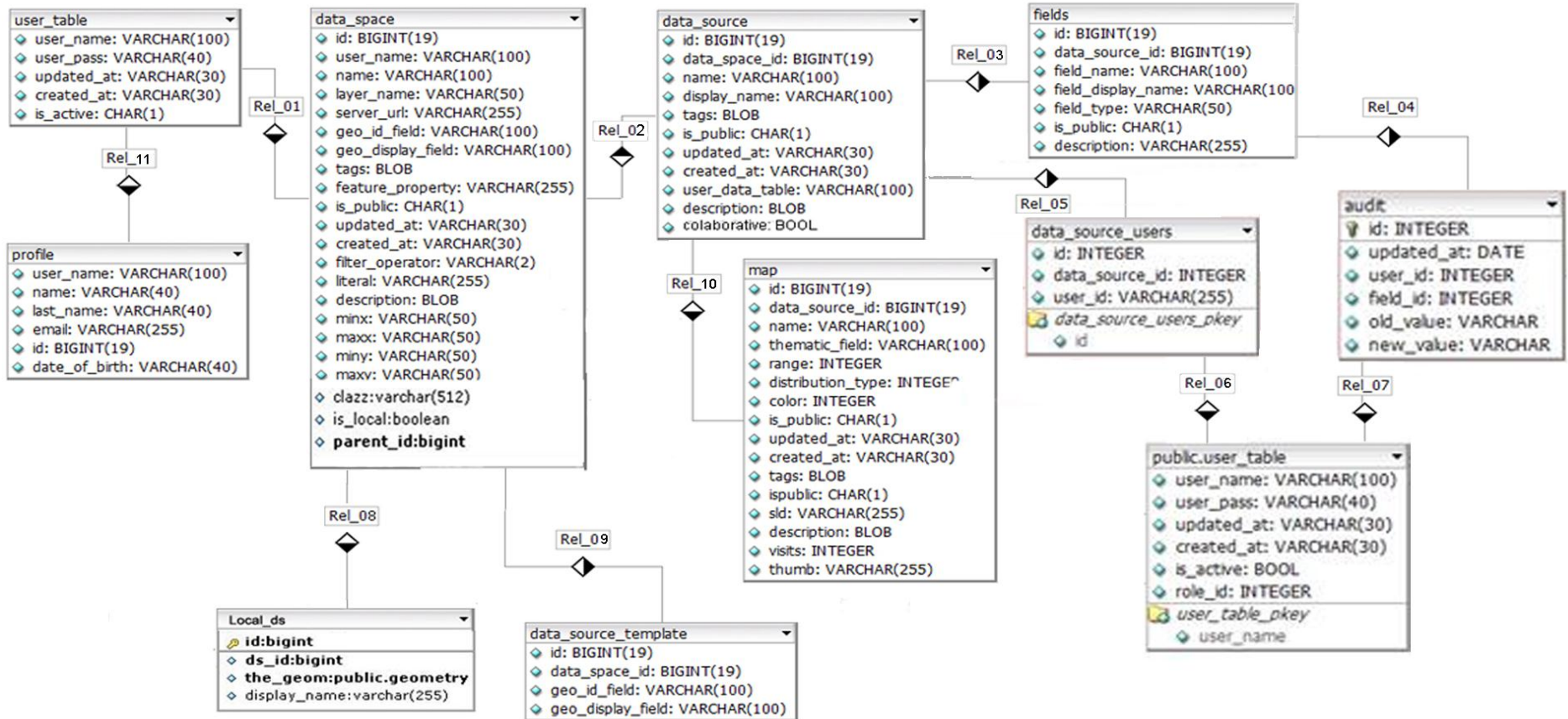


Figura 7. Modelo entidad-relación.

2.2.5 Descripción de Tablas

Tabla user_table:

Esta tabla almacena la información de los usuarios registrados.

Descripción de columnas

- user_name – Identificador de la tabla nombre de usuario.
- user_pass – Contraseña del usuario.
- updated_at – Fecha de actualización.
- created_at – Fecha de creación.
- is_active – Si está activo o no el usuario.

Tabla data_space:

Esta tabla almacena los espacios de datos.

Descripción de Tablas

- id – Identificador de la tabla.
- user_name – Llave foránea de la tabla Usuario.
- name – Nombre del espacio de datos.
- layer_name – Nombre de la capa.
- server_url – URL del servidor de mapas.
- geo_id_field – Identificador del campo geoespacial.
- geo_display_field – Nombre del campo geoespacial a mostrar.
- tags – Etiquetas para ayudar la búsqueda.
- feature_property – En el filtrado por qué campo filtrar.
- is_public – Si está pública o no.
- updated_at – Fecha de actualización.
- created_at – Fecha de creación.
- filter_operator – En el filtrado el operador.
- literal – En el filtrado el valor filtro.
- description – Descripción del espacio de datos.
- minx – Mínimo Valor de la X del boundingBox.
- maxx – Máximo Valor de la X del boundingBox.
- miny – Mínimo Valor de la Y del boundingBox.
- maxy – Máximo Valor de la X del boundingBox.
- claz- Especifica el tipo de cartografía, polígonos, líneas, puntos y multipolígonos.
- is_local- Si es un espacio de datos local o que referencia a un servidor de mapas externo.
- parent_id- Si es creado a partir de un espacio de datos ya existente.

Tabla user table:

Esta tabla almacena la información de los usuarios registrados.

Descripción de columnas

- user_name – Identificador de la tabla nombre de usuario.
- user_pass – Contraseña del usuario.
- updated_at – Fecha de actualización.
- created_at – Fecha de creación.
- is_active – Si está activo o no el usuario.

Tabla data source

Esta tabla almacena las fuentes de datos.

Descripción de columnas

- id – Identificador de la tabla.
- data_space_id – Llave foránea de la tabla espacio de datos.
- name – Nombre de la fuente de datos.
- display_name – Nombre a mostrar de la fuente de datos.
- tags – Etiquetas para ayudar la búsqueda.
- is_public – Si está pública o no.
- updated_at – Fecha de actualización.
- created_at – Fecha de creación.
- user_data_table – Nombre de la tabla de datos del usuario.
- description – Descripción de la fuente de datos.
- collaborative- Para definir si una fuente de datos es colaborativa o no.

Tabla data source users

Esta tabla donde se relacionan los usuarios con permisos sobre una plantilla.

Descripción de columnas

- id – Identificador de la tabla.
- data_source_id – Llave foránea de la tabla fuente de datos.
- user_id – Llave foránea de la tabla usuarios.

Tabla fields

Esta tabla almacena los campos creados por los usuarios.

Descripción de columnas

- id – Identificador de la tabla.

- data_source_id – Llave foránea de la tabla fuente de datos.
- field_name – Nombre del campo.
- field_display_name – Nombre a mostrar del campo.
- field_type – Tipo del campo.
- is_public – Si está pública o no.
- description – Descripción del campo.

Tabla map

Esta tabla almacena los mapas generados por los usuarios.

Descripción de columnas

- id – Identificador de la tabla.
- data_source_id – Llave foránea de la tabla fuente de datos.
- name – Nombre del mapa.
- thematic_field – Campo por el cual se genera el mapa.
- range – Rango aplicado.
- distribution_type – Tipo de distribución.
- color – Color del mapa.
- is_public – Si está pública o no.
- updated_at – Fecha de actualización.
- created_at – Fecha de creación.
- tags – Etiquetas para ayudar la búsqueda.
- sld – Nombre del mapa (SLD).
- description – Descripción del mapa.
- visits – Número de visitas.
- thumb – Nombre de la imagen en miniatura del mapa.

Tabla data_source_template

Esta tabla almacena las plantillas para la creación de fuentes de datos (datos obtenidos del servidor de mapas).

Descripción de columnas

- id – Identificador de la tabla.
- data_space_id – Llave foránea de la tabla espacio de datos.
- geo_id_field – Identificador del campo geoespacial.
- geo_display_field – Nombre del campo geoespacial a mostrar.

Tabla profile

Esta tabla almacena el perfil del usuario.

Descripción de columnas

- user_name – Nombre del usuario y llave de la tabla.
- name– Nombre.
- last_name – Apellidos.
- email – Correo electrónico.

Tabla de Datos del Usuario

Esta tabla es la que se genera automáticamente por el sistema al crearse cada fuente de datos, es importante destacar que estas tablas se crean bajo un esquema que tiene como nombre el nombre de usuario, y puede crear tantas tablas como desee.

Descripción de columnas

- id – Identificador de la tabla.
- geo_id_field – Identificador del campo geoespacial.
- geo_display_field – Nombre del campo geoespacial a mostrar.
- seguidamente todos los campos que ha creado el usuario que tendrán nombre autogenerados.

Tabla local_ds

Esta tabla almacena la cartografía importada por el sistema.

Descripción de columnas

- Id - Identificador de la tabla.
- ds_id - Llave foránea de la tabla espacio de datos.
- the_geom – Guarda la geometría.
- display_name - Nombre del campo de la cartografía importada a mostrar.

Tabla audit

Esta tabla establece una auditoría sobre los cambios ocurridos en los valores de los campos guardándose los valores anteriores para restablecer en caso de errores.

Descripción de columnas

- Id- Identificador de la tabla.
- update_at- Fecha de actualización.
- user_id- id del usuario.
- fiel_id- Llave foránea de la tabla campo.
- new_value – Nuevo valor del campo.
- old_value– Valor antiguo del campo.

2.3 Gestión de grandes volúmenes de datos, gestión de concurrencia, velocidad de procesado y soporte para SIG

PostgreSQL proporciona un gran número de características que normalmente sólo se encontraban en las BD comerciales tales como DB2 u Oracle, por lo tanto PostgreSQL soporta:

- DBMS Objeto-Relacional.
- Alta concurrencia.
- PostGIS, extensión que añade soporte de objetos geográficos a PostgreSQL.

2.4 Búsqueda y recuperación fácil de la información

PostgreSQL implementa Full Text Search que es incluido en el núcleo a partir de la versión 8.3. Por otra parte en la capa de acceso a datos la búsquedas son realizadas mediante HibernateSearch que es uno de los motores de búsqueda de texto más completo, este combina la potencia de Hibernate para la gestión de BD,

con la potencia de Apache Lucene como motor de búsquedas textuales basado en índices inversos y resuelve problemas tales como:

- Hacer la traducción entre nuestros objetos y los índices de Lucene.
- Los problemas de duplicidad, manejando los índices, manteniendo sincronizados los cambios con la base de datos, y optimizando el acceso a los índices internamente.
- Unificar las API, de manera que se pueden hacer búsquedas en el índice y recuperar los objetos pertinentes como si se tratara de una consulta normal de Hibernate.

2.5 Seguridad sobre la información del usuario

Los aspectos de seguridad en la plataforma que se tienen en cuenta son los siguientes:

Confiabilidad: la información manejada por el sistema está protegida de acceso no autorizado y divulgación. Ya que los únicos que pueden hacer cambios en la información son los propios usuarios o aquellos que el usuario designe para administrar sus datos (plantillas colaborativas) y siempre siendo autenticados en el sistema, cuya contraseña usará una encriptación mediante la función hash md5.

Integridad: la información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos. Toda la información publicada tiene que estar acorde a las condiciones y términos establecidos y en cualquiera de los casos de violación de estos los administradores podrán eliminar o deshabilitar las cuentas de estos usuarios.

Autenticación y autorización: es implementado usando **Spring Security**, framework de Java que provee estas dos operaciones incluyendo otras características de seguridad para aplicaciones empresariales.

2.6 Tendencias, metodologías y/o tecnologías empleadas en la investigación.

2.6.1 Arquitectura de N capas

Distintas arquitecturas de desarrollo han pasado hasta llegar hoy a concebir el denominado desarrollo en capas. Para la mayoría de los usuarios, una aplicación de 'n' niveles es algo dividido en distintas partes lógicas. La opción más habitual está formada por una división en tres partes (presentación, lógica de negocio y datos), aunque existen otras posibilidades. Las aplicaciones en 'n' niveles surgieron por primera vez como una forma de resolver algunos de los problemas asociados a las aplicaciones cliente/servidor tradicionales (modelo de dos capas), pero con la llegada de la Web, esta arquitectura ha llegado a dominar el nuevo desarrollo [26].

La arquitectura de aplicaciones en n-capas, se ha convertido en el estándar para el software empresarial. Se caracteriza por la descomposición de las aplicaciones [27].

- Proporciona una escalabilidad, capacidad de administración y utilización de recursos mejorados.
- Cada capa es un grupo de componentes que realiza una función específica.
- Se puede actualizar una capa sin recompilar otras capas.

Arquitectura de 3 capas:

- Capa de presentación.
- Capa de negocios.
- Capa de datos.

El uso de las tres capas es relativo, depende de la tecnología utilizada en la implementación de la arquitectura y la complejidad de la misma.

Capa de servicios de usuario o presentación.

En una aplicación de N niveles, esta capa reúne todos los aspectos del software que tiene que ver con las interfaces y la interacción con los diferentes tipos de usuarios humanos. Estos aspectos típicamente incluyen el manejo y vista de las

ventanas, la autenticación de usuarios, el formato de los reportes, menús, gráficos y elementos multimedia en general [28].

Capa de servicios de negocios.

Esta capa reúne todos los aspectos del software que automatizan o apoyan los procesos de negocio que llevan a cabo los usuarios. Estos aspectos típicamente incluyen las tareas que forman parte de los procesos, las reglas y restricciones que aplican. La lógica de negocios construida en componentes lógicos personalizados enlaza los ambientes clientes y el nivel de servicios de datos. Esta capa también recibe el nombre de capa de la Lógica de la Aplicación. Las responsabilidades de esta capa se pueden sintetizar en:

- Recibir la entrada del nivel de presentación.
- Interactuar con los servicios de datos para poder ejecutar las operaciones de negocios que la aplicación automatiza.
- Enviar el resultado procesado al nivel de presentación.

Capa de servicios de datos.

Esta capa reúne todos los aspectos del software que tiene que ver con el manejo de los datos persistentes, por lo que también se le denomina la capa de las BD.

Los principales servicios de esta capa radican en:

- Almacenar los datos.
- Recuperar los datos.
- Mantener los datos.
- La integridad de los datos.

El modelo de N capas persigue, con su arquitectura, que las aplicaciones maximicen aspectos trascendentes en el desempeño como son:

- **Autonomía:** Habilidad de una aplicación para gobernar sus recursos críticos.
- **Confiabilidad:** Habilidad de una aplicación para proporcionar resultados exactos.
- **Disponibilidad:** Cantidad de tiempo que una aplicación es capaz de dar

servicio confiablemente a las peticiones del cliente.

- **Escalabilidad:** Meta utópica del crecimiento lineal del rendimiento al agregar recursos adicionales, y es lo que le permite a una aplicación servir desde diez usuarios, hasta decenas de miles de usuarios, simplemente agregando o quitando recursos como sea necesario para "escalar" la aplicación.
- **Interoperabilidad:** Habilidad de una aplicación para acceder a las aplicaciones, los datos o los recursos en otras plataformas.
- El uso de una arquitectura de N capas permite que la potencia de cálculo recaiga en el servidor. De esta manera, los clientes son cada vez más ligeros y no necesitan ni demasiadas capacidades de cálculo ni un excesivo software instalado, porque la capa de negocio y la de datos se encuentran centralizadas en el servidor.

El diseño seleccionado para la arquitectura de Tocaroro es la que se muestra en la Figura 8:

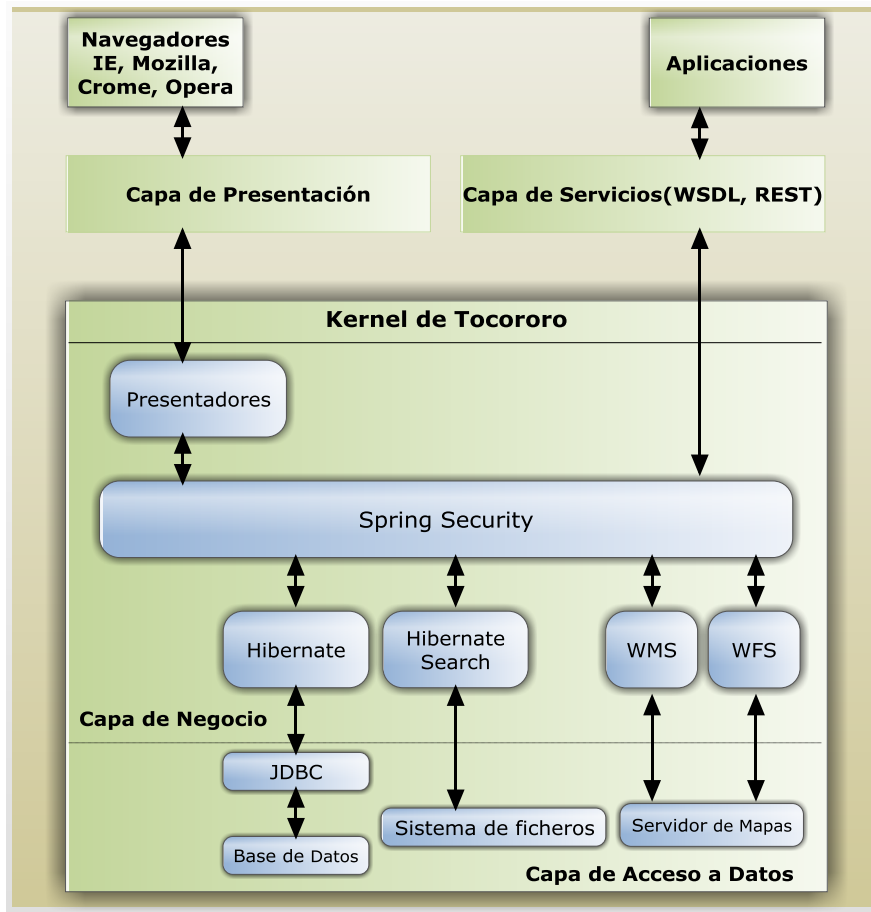


Figura 8. Diseño de la Arquitectura Tocatoro.

2.6.2 Arquitectura orientada a Módulos

Un diseño separado por componentes (módulos) es cuando estos pueden ser conectados entre ellos, donde la belleza de una arquitectura modular es que se puede reemplazar o adicionar algún componente sin afectar el resto del sistema en contraposición a una arquitectura integrada en la cual no existe una clara división entre los componentes existentes. El término de “Diseño de Software Modular”, se refiere a una estrategia de diseño en el cual un sistema está compuesto por pequeñas y autónomas rutinas que se acoplan juntas (Schilleng 2000).

En las aplicaciones grandes y complejas, el modelo de Dominio tiende a crecer extraordinariamente. El modelo llega a un punto donde es complicado hablar

sobre ello como “un todo”, y puede costar bastante entender bien todas sus relaciones e interacciones entre todas sus áreas. Por esa razón, se hace necesario organizar y dividir el modelo en diferentes módulos. Los módulos se utilizan como un método de organización de conceptos y tareas relacionadas (normalmente bloques de negocio diferenciados) para reducir la complejidad.

El concepto de módulo es realmente algo utilizado en el desarrollo de software desde sus orígenes. Es más fácil ver la foto global de un sistema completo si se subdivide en diferentes módulos verticales y después en las relaciones entre dichos módulos. Una vez que se entienden las interacciones entre dichos módulos, es más sencillo focalizarse en más detalle de cada uno de ellos. Es una forma simple y eficiente de gestionar la complejidad. El lema “divide y vencerás” es la frase que mejor lo define, ver Figura 9.

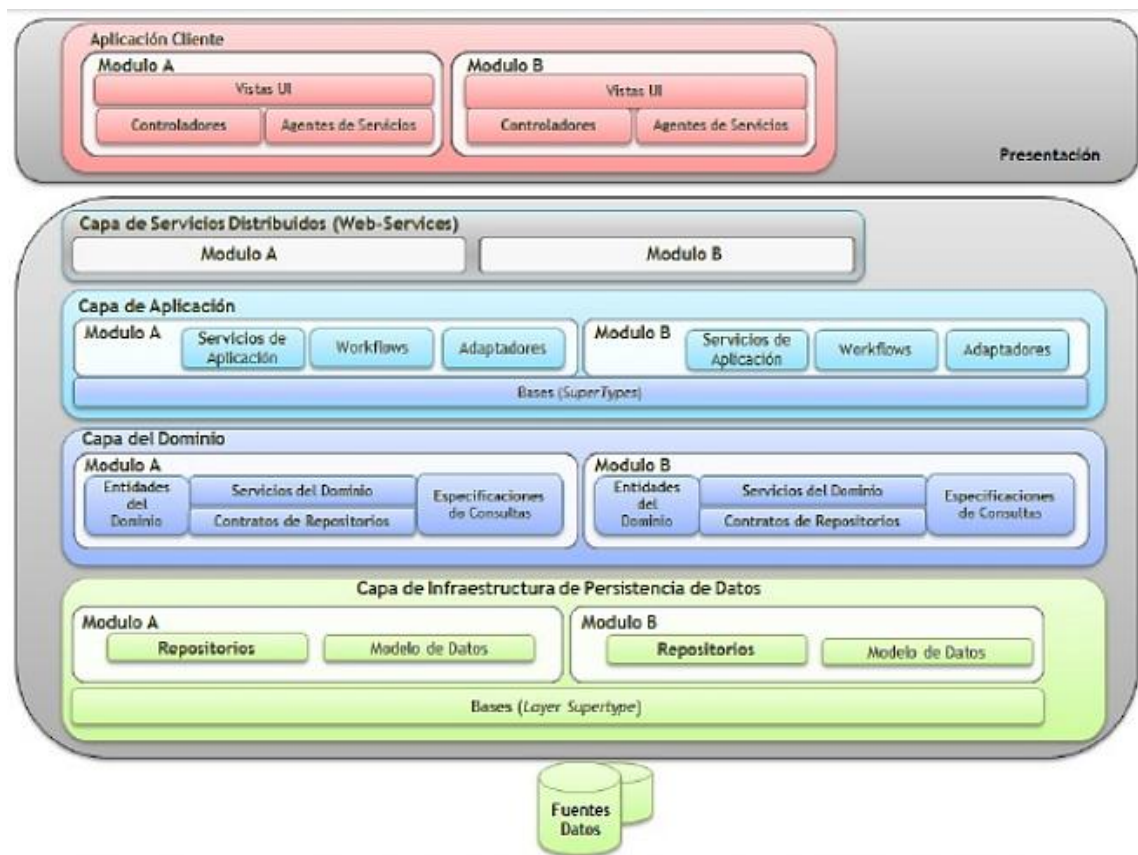


Figura 9. Arquitectura orientada a módulos.

2.7 Descripción del modelo de sistema

Tocororo es el nombre que se le asigna a la plataforma haciendo uso de nuestra ave nacional. Tocororo es una Aplicación Web colaborativa que permite la publicación de información temática y generación de mapas temáticos a partir de estos.

Está formado por varios módulos:

- Módulo “**Panel de Control**”: Permite al usuario registrado administrar la información que publica, permitiéndole las operaciones básicas de adicionar, actualizar y eliminar. Para acceder a este módulo debe estar autenticado antes y el proceso es realizado mediante un dialogo de autenticación. El Panel de Control también permite administrar los espacios de datos (datos referentes al servidor de mapas, capa, filtros), las plantillas colaborativas (información que el usuario desea que otros modifiquen y actualicen), las fuentes de datos (información temática como datos poblacionales, medioambientales o indicadores que pueda ser georeferenciada) y los mapas temáticos además definir cuál de toda esta información será pública o no.
- Módulo de “**Búsqueda**”: Permite al usuario anónimo o registrado realizar búsquedas de información así como mapas temáticos en todo el espacio de información que los propios usuarios registrados hayan subido al servidor y que compartan para el uso común en el entorno en que se desarrolla la aplicación. La interfaz mantendrá una estadística de los mapas más visitados, así como los últimos adicionados, para brindarle al usuario una visión más clara de lo que necesita buscar, de una forma muy simple para realizar las búsquedas, teniendo en cuenta los colores y una interfaz de usuario amigable donde los resultados sean orientados a brindarle al usuario una respuesta en el menor tiempo posible.
- Módulo de “**Registro de Usuarios**”: Permite el registro de usuarios al sistema, como todos los módulos de este tipo, son los encargados de crear una cuenta bajo un nombre de usuario único (dirección de correo

electrónico del usuario), además de permitirle al usuario activar la cuenta, siempre validando que la dirección de correo electrónico sea válida, el módulo también capta algunos datos más del perfil del usuario.

- Módulo de “**visualización de los mapas y los datos de los usuarios**”: Es el encargado de visualizar la información y los mapas temáticos cuando el usuario da clic en el enlace de la información recuperada mediante el módulo de búsqueda. Se muestra a través de una nueva ventana donde se visualiza el mapa y los datos asociados a estos mapas. Entre las operaciones que permite realizar este módulo se encuentra la posibilidad de exportar dicha información a diferentes formatos de intercambio tales como pdf, xls, png, jpg.

2.7.1 Visualización de la información georeferenciada y la información temática

Los mapas temáticos son mapas basados en mapas topográficos que representan cualquier fenómeno geográfico de la superficie terrestre. Persiguen objetivos bien definidos. Hacen referencia a la representación de ciertas características de distribución, relación, densidad o regionalización de objetos reales (vegetación, suelos, geología, etc.) o de conceptos abstractos (indicadores de violencia, de desarrollo económico, de calidad de vida, etc.). Para representar variables numéricas utilizan todo tipo de recursos visuales, como superficies de distintos colores o tramas (coropletas), ver Figura 10, flechas para indicar el movimiento de un fenómeno (flujos, a veces tienen un grosor proporcional a su magnitud), el trazado de líneas que unen puntos de igual valor (isolíneas), círculos o símbolos de tamaño proporcional al valor numérico, o incluso mapas deformados para que cada unidad geográfica se represente con un tamaño proporcional a su valor numérico (cartogramas o mapas anamórficos).

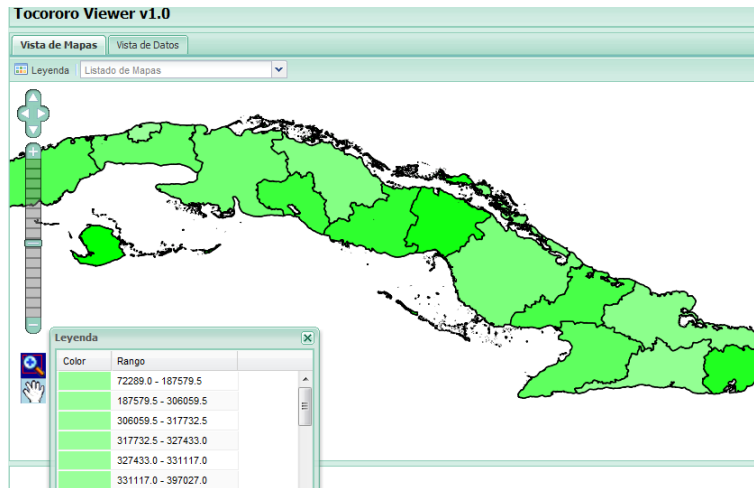


Figura 10. Visor de mapas en la vista de mapas.

2.8 Extensibilidad mediante Plug-ins (Complementos)

Tocaroro está diseñado y pensado bajo el concepto de extensibilidad, por lo tanto su arquitectura está soportada sobre plug-ins también conocido como complementos o extensiones. La arquitectura de Plug-in permite escribir cualquier extensión deseada en el ambiente y extender las funcionalidades básicas de núcleo.

El navegador es el encargado de la carga de estos componentes mediante un cargador de Plug-ins que a su vez permite la interacción de los complementos con el Núcleo de Tocaroro usando una Arquitectura dirigida por eventos(en inglés Event-driven architecture, EDA), es un patrón de arquitectura software que promueve la producción, detección, consumo de, y reacción a eventos. En el siguiente capítulo en la sección de “Configuración” se describe y se muestra el fichero de configuración de los plug-ins.

2.9 Internacionalización

La internacionalización es el proceso de diseñar software de manera tal que pueda adaptarse a diferentes idiomas y regiones sin la necesidad de realizar cambios de ingeniería ni en el código. La localización es el proceso de adaptar el software para una región específica mediante la adición de componentes específicos de un

“locale”⁹ y la traducción de los textos, por lo que también se le puede denominar regionalización. No obstante la traducción literal del inglés es la más extendida.

Es una práctica común en el idioma inglés (sobre todo en el ámbito de la computación), abreviar “internationalization” con el “numerónimo”, “i18n”. Ello se debe a que entre la primera “i” y la última “n” de dicha palabra hay 18 letras. Lo mismo sucede con localization, que se abrevia “L10n”. La “L” mayúscula se utiliza para distinguirla de la “i” minúscula de i18n.

Algunas empresas, como Microsoft e IBM, usan el término globalización para referirse a la combinación de internacionalización y localización. Globalización puede también abreviarse con el numerónimo “g11n”(Esselink y Benjamins 2000).

Tocororo es una aplicación desarrollada sobre GWT, este framework permite internacionalización.

2.10 Uso de Tecnologías de Software Libre y Open Source

Tocororo está desarrollado enteramente sobre tecnologías de SL utilizando estándares de desarrollo y las mejores prácticas de desarrollo de software por lo tanto:

2.10.1 Restricciones de la Arquitectura

El diseño de las aplicaciones se hará utilizando la POO. Encapsulación de la lógica por clases.

Se utilizará las tecnologías que brindan los frameworks definidos para cada una de las capas de la aplicación:

- **Para la capa de presentación:** JSF 1.2_15, RichFaces 3.3.2, GWT 2.3, gwt-ext 2.6, mvp4g-1.3.1 (framework que implementa el patrón de diseño MVP para GWT).

⁹ En informática, un **locale** es un conjunto de parámetros que define el idioma, país y cualquier otra preferencia especial que el usuario desee ver en su interfaz de usuario.

- **Para la capa de lógica del negocio:** Framework Spring y el módulo Spring Security de Spring para la seguridad.
- **Para la capa de Acceso a Datos:** Framework Hibernate.
- **Para la indexación de la información de los usuarios:** Hibernate Search.
- **Para la capa de datos:** Geoserver como servidor de mapas y el uso de los protocolos WMS, WFS y como gestor de BD PostgreSQL v8.4.

2.10.2 Estándares de codificación

Para el desarrollo de la aplicación se utilizará el estándar de codificación del lenguaje Java.

2.10.3 Patrones de diseño

- MVP.
- Service Locator (Localizador de servicios).
- Data Access Object (Objeto de Acceso a Datos, DAO).

2.10.4 Marcos de Trabajo

- JSF 1.2_15.
- Spring Framework 3.0.3.
- Hibernate 3.3.2.
- Spring Security 3.0.1.
- GWT 2.3.
- gwt-ext 2.6.
- mvp4g-1.3.1.

2.10.5 Plataforma de desarrollo

La plataforma de desarrollo que se utilizará será Java 2 Enterprise Edition (J2EE) usando:

- *JBossTools* y *WTP-SDK*, para desarrollar la capa de presentación y los

servicios web.

- *Spring-IDE*, para implementar el Framework Spring.
- *Hibernate Tools*, el cual facilita la ingeniería inversa y reversa de los ficheros de mapeo entre clases java y tablas de BD.
- *JUnit* el cual facilita las pruebas de unitarias.
- *SVNeclipse* el cual facilita la integración con Subversión.
- Como servidor de BD PostgreSQL.
- Como servidor de control de versiones Subversión.

2.11 Modelación de la Plataforma

2.11.1 Requerimientos funcionales

Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el mismo (Jacobson, 2000).

Los requisitos funcionales del software propuesto son los siguientes:

1. Autenticar usuarios.
2. Gestionar registro de usuarios.
3. Gestionar espacios de datos.
4. Gestionar fuentes de datos.
5. Gestionar campos.
6. Gestionar campos auto calculado (expresiones).
7. Gestionar valores de los campos.
8. Gestionar mapas.
9. Importar cartografías.
10. Importar la información georeferenciada publicada por los usuarios.
11. Exportar la información georeferenciada publicada por los usuarios.
12. Realizar búsqueda.
13. Visualizar los datos de los usuarios recuperados.
14. Visualizar los mapas de los usuarios recuperados.

15. Gestionar el registro de usuarios.
16. Gestionar perfiles de usuarios.

2.11.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener, como restricciones del entorno o de implementación, rendimiento, etc. (Jacobson, 2000).

Requerimientos de apariencia o interfaz externa

La herramienta propuesta presenta una interfaz amigable y fácil de usar, de manera que no sea una dificultad para el usuario el uso de ella.

Requerimientos de usabilidad.

Tocororo es una Aplicación Web Colaborativa que será utilizada por cualquier persona que navegue en la red, permite publicar información temática a partir de la cual se pueden generar mapas temáticos. Los mapas generados junto con su leyenda personalizable pueden guardarse y compartirse; se pueden guardar también los mapas favoritos.

Dado que Tocororo es una aplicación Cliente-Servidor es imprescindible tener conectividad teniendo en cuenta el entorno en que se desarrollará la misma.

Requerimientos de Rendimiento

Para un funcionamiento óptimo de la aplicación se seguirán las diferentes técnicas de elaboración en la Web, que faciliten el rápido acceso a sus páginas. La eficiencia del producto estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo Cliente/Servidor, y la velocidad de las consultas en la BD.

La herramienta propuesta debe ser rápida y el tiempo de respuesta debe ser el mínimo posible, adecuado a la rapidez con que el cliente requiere la respuesta a su acción.

Requerimientos de Soporte

Para garantizar el soporte a los clientes de la herramienta, se documentará, se dará la posibilidad de emitir sus quejas y sugerencias a los desarrolladores de la herramienta, por correo, realizar mantenimiento al sistema y con el aumento de la independización de las funcionalidades se necesitarán posteriores versiones.

Requerimientos de Portabilidad

La herramienta propuesta podrá ser usada por cualquier plataforma, a través de un servidor Web y servidor de BD; y será utilizado bajo los sistemas operativo Linux y Windows.

Requerimientos políticos, culturales y legales

La herramienta propuesta deberá responder a los intereses de la Constitución de la República de Cuba, así mismo no existirán prioridades en el servicio según el nivel social, cultural o étnico. Teniendo en cuenta que la información que se encuentra en la misma va dirigida principalmente a profesionales especializados.

Requerimientos de Ayudas

La herramienta contará con sistema de ayuda basado en tutoriales tales como multimedia o asistentes que guiarán al usuario a través de las diferentes opciones, donde se esclarecerán dudas sobre su uso, así como las facilidades que le brinda.

Requerimientos de Software

Servidor:

- Sistema Operativo: Windows Server 2000, 2003, 2008. GNU Linux, Solaris.
- Servidor de BD: PostgreSQL 8.4 ó superior.
- Máquina virtual de Java: JDK 1.5 ó superior.
- Servidor Web: Apache Tomcat 6 o superior. JBoss AS 4.2 o superior.

Cliente:

- Sistema Operativo: Windows XP SP2, 2003. GNU Linux, Solaris
- Navegador Web: Internet Explorer 6.0 ó superior. Mozilla Firefox 3.0 o superior.

Requerimientos de Hardware

Los requerimientos para la ejecución de Tocatoro deben ser:

Servidor:

- Memoria RAM en dependencia del entorno:
 - Empresarial: 2 GB como mínimo, 4 GB recomendado.
 - Intranet: 8 GB.
 - Internet: 16 GB a 32 GB en dependencia de las necesidades.
- Disco Duro en dependencia del entorno:
 - Empresarial: 100 GB.
 - Intranet: 1 TB.
 - Internet: En dependencia de las necesidades.
- Servidor de Aplicaciones tal como Apache Tomcat v6 recomendado v7.
- Servidor de BD Postgres v8.4.
- JRE v1.5 o superior de Java Run Time.
- Servidor de Mapas (Geoserver).

Cientes:

- 512 MB de Memoria RAM.
- Navegador (Firefox, IE, Opera, Chrome).

Requerimientos del personal

Un Analista de Pruebas.

Requerimientos de acondicionamiento

- Local provisto con aire acondicionado, energía eléctrica de 110 V, dos puntos de red funcionales e iluminación adecuada para el trabajo en las computadoras donde se efectuarán las pruebas del software.
- Discos duros externos de 500 GB como mínimo para el traslado de información.

Restricciones en el diseño y la implementación

Para ser consecuente con el planteamiento de hacer una herramienta que pueda ser usada por cualquier usuario es necesario usar para su implementación lenguajes de programación que sean multiplataforma.

Para garantizar una mejor documentación del sistema, así como el uso de última tecnología, se utiliza para realizar el análisis y el diseño del sistema UML y su extensión para el desarrollo de proyectos Web. Como herramienta de apoyo a este lenguaje de modelación se utiliza Rational Rose.

2.12 Modelo de casos de uso del sistema

2.12.1 Actores del sistema a automatizar

Un actor no es más que un conjunto de roles que los usuarios de Casos de Uso desempeñan cuando interaccionan con estos Casos de Uso. Los actores representan terceros fuera del sistema que colaboran con el mismo. Una vez que hemos identificado los actores del sistema, tenemos identificado el entorno externo del sistema (Jacobson, 2000).

En la Tabla 1 se muestran los actores que actúan en la plataforma y una breve descripción de estos.

Tabla 1. Definición de actores del sistema a automatizar.

.Nombre del actor.	Descripción
Usuario Anónimo	Este usuario interactúa de forma directa con el sistema, obteniendo una serie de información que le servirá para realizar su trabajo más rápido y eficiente, tendrá acceso a los requerimientos funcionales 12,13 y 14 del sistema.
Usuario Registrado	Este usuario interactúa de forma directa con el sistema, beneficiándose de la información que este brinda, además es el responsable de la actualización del mismo, tendrá acceso a todos los requerimientos funcionales del sistema

2.12.2 Paquetes y sus relaciones.

La forma en que interactúa cada actor del sistema con el sistema se representa con un Caso de Uso. Los Casos de Uso son “fragmentos” de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. De manera más precisa, un caso de uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia [29].

Por el número de casos de uso se introducen paquetes al modelo de casos de uso del sistema con el objetivo de disminuir el tamaño y así aumentar en comprensión. En la Tabla 2 se hace una descripción de los casos de uso por paquetes.

En la Figura 11 se muestra el diagrama de casos de uso por paquetes (paquete: Service, paquete: DAO, paquete: Web, este último paquete a su vez está dividido en tres paquetes (ver Figura 12): Panel de Control, Búsqueda y Usuarios.

Tabla 2. Descripción de paquetes.

NOMBRE PAQUETE	FUNCIONALIDAD	CRITERIO AGRUPAMIENTO	ACTORES QUE INTERVIENEN
DAO	Implementa el Acceso a Datos		
SERVICE	Implementa la capa del Negocio		
TEST	Implementa los casos de Pruebas		
WEB	Implementa las Vistas del Sistemas		Usuarios Anónimos, Usuarios Registrados

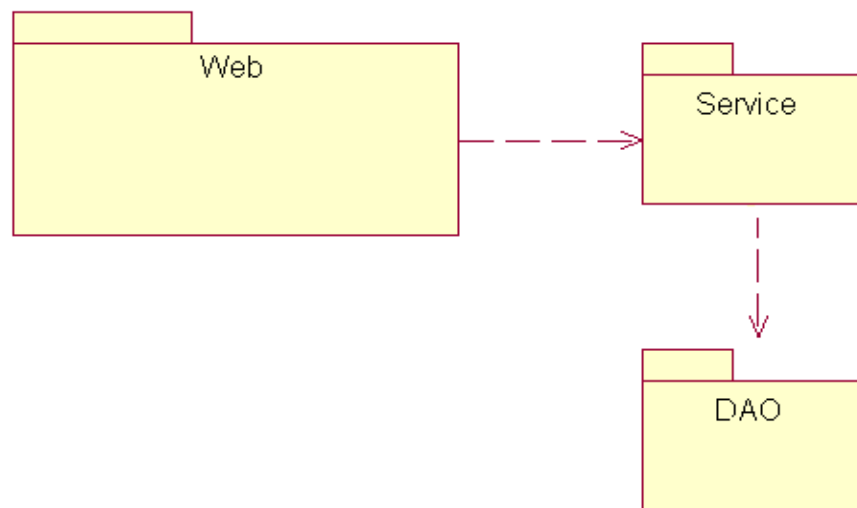


Figura 11. Diagrama de Casos de Usos por Paquetes.

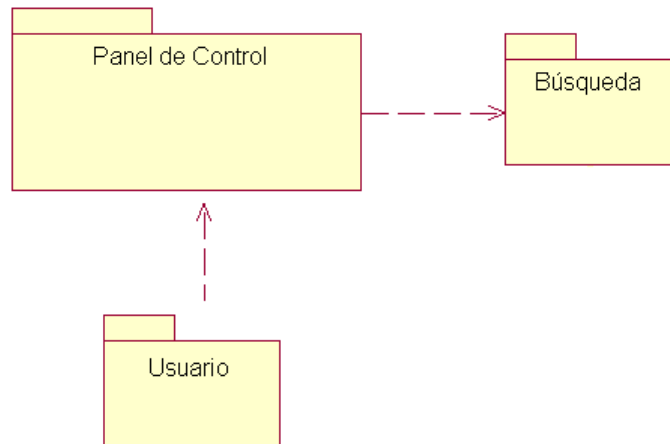


Figura 12. Paquete web.

El paquete **Panel de Control** contiene los siguientes casos de uso:

1. Autenticar usuarios.
2. Gestionar el registro de usuarios (insertar, modificar, eliminar).
3. Gestionar espacios de datos (insertar, modificar, eliminar).
4. Gestionar fuentes de datos (insertar, modificar, eliminar).
5. Gestionar campos (insertar, modificar, eliminar).
6. Gestionar campos auto calculado, expresiones (insertar, modificar, eliminar).
7. Gestionar valores de los campos (insertar, modificar, eliminar, importar, exportar).
8. Gestionar mapas (insertar, modificar, eliminar).
9. Importar cartografías.
10. Importar la información georeferenciada publicada por los usuarios.
11. Exportar la información georeferenciada publicada por los usuarios.

En la Figura 13 se muestra el diagrama de casos de uso del paquete panel de control.

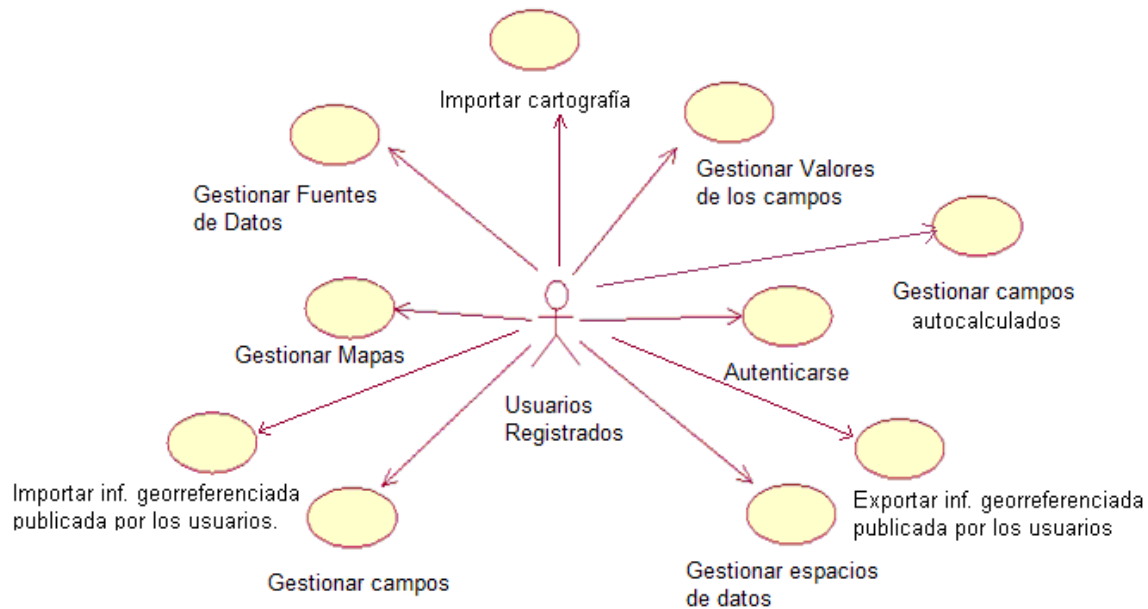


Figura 13. Diagrama de casos de uso del sistema: paquete panel de control.

El paquete **Búsqueda** contiene los siguientes casos de uso:

1. Realizar Búsqueda.
2. Visualizar los Datos de los Usuarios Recuperados.
3. Visualizar los Mapas de los Usuarios Recuperados.

En la Figura 14 se muestra el diagrama de casos de uso del paquete búsqueda.

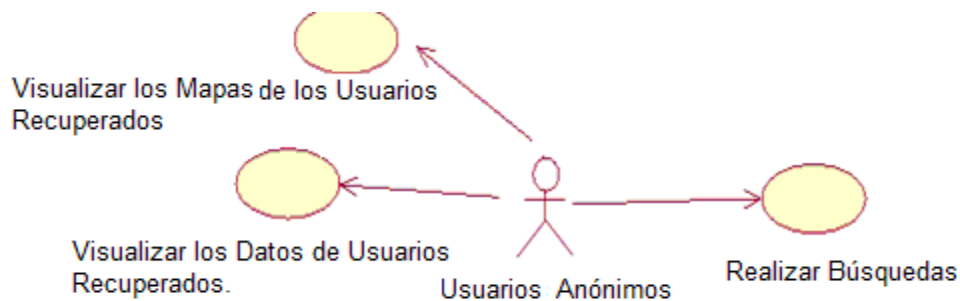


Figura 14. Diagrama de caso de uso del sistema: paquete búsqueda.

El paquete **Usuario** contiene los siguientes casos de uso:

1. Gestionar el Registro de Usuarios (Insertar).
2. Gestionar Perfiles de Usuarios (Modificar).

En la Figura 15 se muestra el diagrama de casos de uso del paquete usuario.

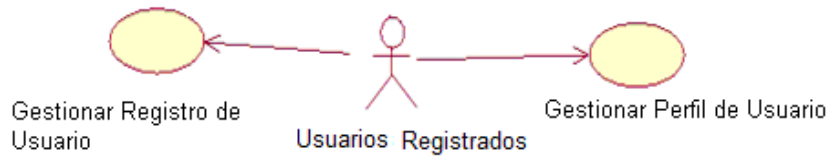


Figura 15. Diagrama de caso de uso del sistema: paquete usuario.

2.13 Descripción de los casos de uso

En la Tabla 3 se muestra una descripción detallada de los casos de uso y su prototipo.

Tabla 3. Descripción de los casos de uso y su prototipo

Casos de Uso	Descripción	Prototipo
Autenticar usuarios.	Anexo 1	Anexo 17
Gestionar el registro de usuario.	Anexo 2	
Gestionar espacios de datos.	Anexo 3	Anexo 18
Gestionar fuentes de datos.	Anexo 4	Anexo 19
Gestionar campos.	Anexo 5	Anexo 20
Gestionar campos auto calculado (expresiones).	Anexo 6	Anexo 21
Gestionar valores de los campos.	Anexo 7	Anexo 22
Gestionar mapas.	Anexo 8	Anexo 23, Anexo 24
Importar cartografías.	Anexo 9	Anexo 25
Importar la información georeferenciada publicada por los usuarios.	Anexo 10	Anexo 26
Exportar la información georeferenciada publicada por los usuarios.	Anexo 11	Anexo 27
Realizar búsqueda.	Anexo 12	Anexo 28
Visualizar los datos de los usuarios recuperados.	Anexo 13	Anexo 29
Visualizar los mapas de los usuarios recuperados.	Anexo 14	Anexo 30

En la Tabla 4 se muestra el diagrama de actividades de los casos de uso de la plataforma.

Tabla 4. Diagrama de actividades de los casos de uso del sistema.

Casos de Uso	Diagrama de Actividades
Autenticar usuarios.	Anexo 31
Gestionar espacios de datos.	Anexo 32
Gestionar fuentes de datos.	Anexo 33
Gestionar campos.	Anexo 34
Gestionar campos auto calculado (expresiones).	Anexo 35
Gestionar valores de los campos.	Anexo 36
Gestionar mapas.	Anexo 37
Importar cartografías.	Anexo 38
Importar la información georeferenciada publicada por los usuarios.	Anexo 39
Exportar la información georeferenciada publicada por los usuarios.	Anexo 40
Realizar búsqueda.	Anexo 41
Gestionar el registro de usuarios.	Anexo 42
Gestionar el perfil de usuarios.	Anexo 43

2.14 Conclusiones

A modo de conclusión se puede decir que para lograr una plataforma colaborativa basada en SL la combinación y el uso de tecnologías tales como PostgreSQL como gestor de BD, GeoServer como servidor de Mapas y lenguaje de programación Java constituyen una arquitectura robusta y segura. Además se incluye el uso de los siguientes frameworks: **GWT**, frameworks **Spring** para la Inversión de Control, el **módulo Spring Security** para la seguridad y para el Acceso a Datos: el frameworks **Hibernate**. También se realizó un análisis sobre

los requerimientos funcionales y no funcionales de la plataforma, que darán solución al problema planteado. Se realizó la propuesta del sistema así como los casos de uso del sistema y sus actores, utilizando para su modelado el Lenguaje Unificado de Modelado (UML), que permite representar el diagrama de casos de uso del sistema y la especificación de los mismos.

CAPÍTULO 3 Implementación de la plataforma Web colaborativa en la nube

3.1 Introducción

En este capítulo se desarrolla la implementación de la solución propuesta, así como las principales extensiones de la plataforma colaborativa Tocatoro para el desarrollo de la misma. Se muestra una versión inicial del prototipo de arquitectura que da solución al problema de investigación. Además se abordan temas de interés como el de los ficheros de configuración, siendo parte de la configuración inicial para el funcionamiento de la plataforma.

3.2 Implementación de la extensión importar y publicar cartografía

Esta nueva extensión tendrá como principal funcionalidad subir cartografía al servidor, la cual será publicada por la plataforma Tocatoro. Una vez importada la cartografía quedará en una estructura propia que facilitará la búsqueda, la recombinación y la publicación de las mismas.

3.2.1 Modelo de Datos

En la Figura 16 se muestran las tablas que incluyen esta nueva funcionalidad. Se crea una nueva tabla “local_ds” donde se guardan los datos obtenidos del fichero “.shp” y en la tabla “data_source” se guardan los datos del fichero “.dbf”.

3.2.1 Condiciones mínimas

Es imprescindible que la cartografía esté compactada en un fichero de tipo .Zip para su gestión y debe estar en formato shape.

3.3 Implementación de la extensión permitir campos tipo expresión autocalculada

A la hora de diseñar una fuente de datos el usuario puede adicionar campos, estos campos representan una información determinada, como puede ser cantidad de habitantes por provincia, fechas de sucesos, número de pacientes con VIH por provincia, etc. Estos campos pueden ser numéricos, de tipo texto, booleanos, fechas, etc.

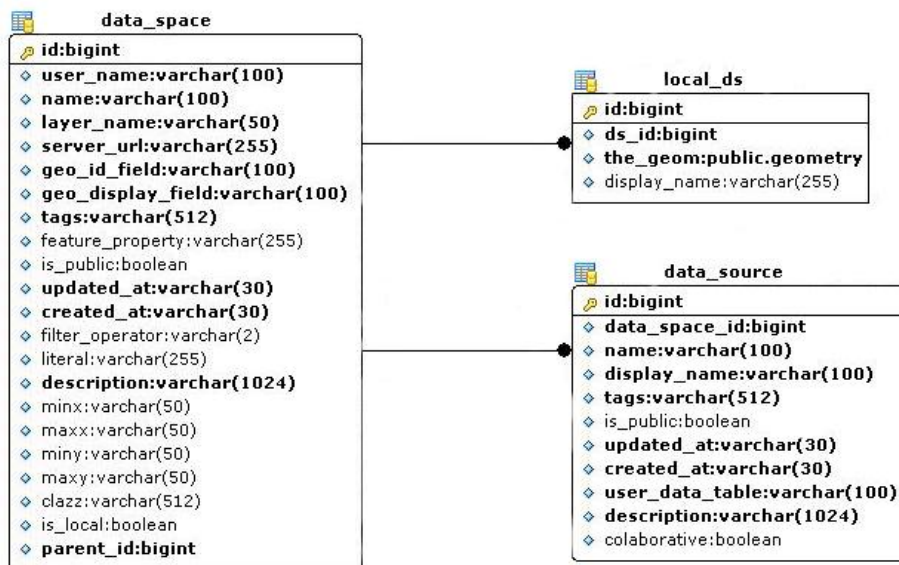


Figura 16. Tablas que incluyen la funcionalidad de importar cartografía.

El usuario podrá incorporar toda la información que desee siempre y cuando tenga los valores concretos para cada campo. Esta extensión pretende que el usuario pueda combinar algunos de estos campos con el objetivo de obtener nuevos campos autocalculados o sea que el usuario tenga un nuevo tipo de campo llamado expresión, que se calcule a partir de otros ya definidos.

3.3.1 Modelo de Datos

A la BD Tocaroro se le adicionaron dos nuevas tablas, en las cuales se almacenarán los campos autocalculados como se muestra en la

Figura 17. La tabla **expresión_field** hereda de la tabla **fields**, en la cual se almacenará la expresión matemática definida por el usuario. La otra tabla es

expresión_field_operand que representa un operando de la expresión, que no es más que la referencia a un campo de la base de datos.

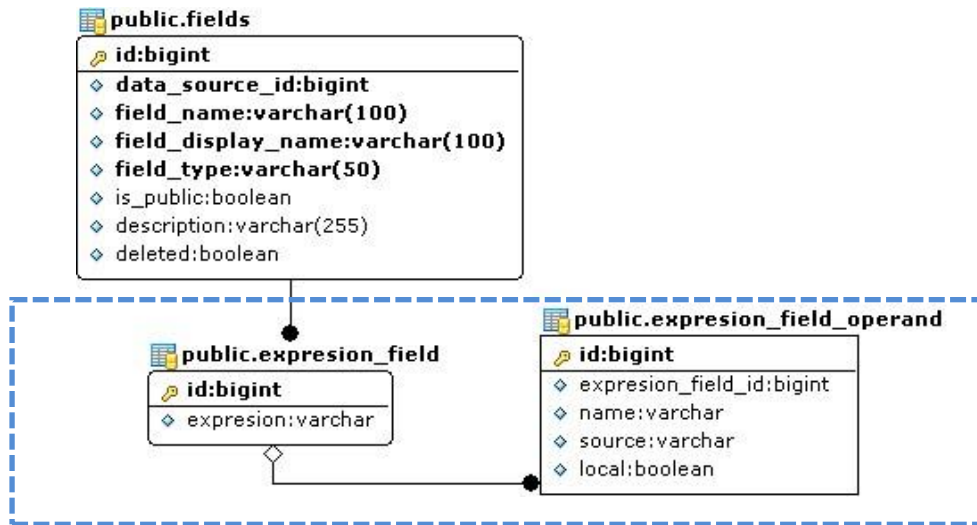


Figura 17. Tablas incorporadas a la base de datos de Tocatoro.

3.4 Implementación de la extensión de Generación de mapas temáticos

El proceso de generar mapas temáticos comienza desde el plug-in encargado de la Vista de Diseño, que es donde se crean los campos temáticos, luego en la Vista de Datos, se introducen los datos asociados al campo temático y finalmente en la Vista de Mapas (donde realmente se crean) se visualizan los mapas.

3.4.1 Modelo de Datos

Involucra la tabla data_space (servidor de mapas, capa con la que se genera el mapa), data_source (mapa que pertenece a una fuente de datos), fields (los mapas se generan a partir de uno o varios campos) y finalmente la tabla de datos asociada la fuente de datos (donde se guarda la información de cada campo).

3.4.2 Condiciones mínimas

Que exista un espacio de datos, una fuente de datos, un campo temático y datos asociados a ese campo.

3.5 Implementación de las plantillas de datos colaborativas

A partir del desarrollo del prototipo del núcleo de Tocatoro, sobre la misma plataforma, se obtendrá una extensión para definir fuentes de datos Colaborativas. Ya en sí Tocatoro, como plataforma, tiene una base colaborativa cuando los usuarios comparten los datos que actualizan y permite la remezcla de los mismos. Esta extensión va un poco más allá al permitir que un grupo de usuarios participen en la creación y actualización de una fuente de datos.

3.5.1 Modelo de Datos

La funcionalidad de plantillas de datos colaborativas necesita introducir algunos cambios a las extensiones existentes. A la BD se le agrega una tabla donde se establece la relación entre una fuente de datos y los usuarios con permiso de edición sobre la misma. También es necesaria la modificación de las interfaces visuales de los componentes de creación y edición de una fuente de datos donde al definir una de ellas como colaborativa es posible establecer los usuarios que podrán participar en la actualización de los datos.

La Figura 18 muestra las tablas del núcleo que tienen cambios con la introducción de la nueva funcionalidad.

Se puede observar que se agrega a la tabla “data_source” el campo boolean “colaborative” para definir si una fuente de datos es colaborativa o no. También se crea la tabla “data_source_users” donde se relacionan los usuarios con permisos sobre una plantilla. Otro elemento nuevo introducido es la tabla “audit” donde se establece una auditoría sobre los cambios ocurridos en los valores de los campos guardándose los valores anteriores para restablecer en caso de errores.

Como parte de la arquitectura el sistema debe ocuparse del manejo de la seguridad, solamente pueden modificar dichos datos aquellos usuarios que el propietario de la información establezca como los autorizados a insertar o modificar la información colaborativa. Además se implementan dos esquemas para la notificación al propietario de los cambios, estos son los de enviar un correo electrónico con un reporte detallado de los cambios y el otro es en el área de

notificación, donde también se le notifica al propietario de todos los cambios realizados sobre sus datos y por quién fue modificado.

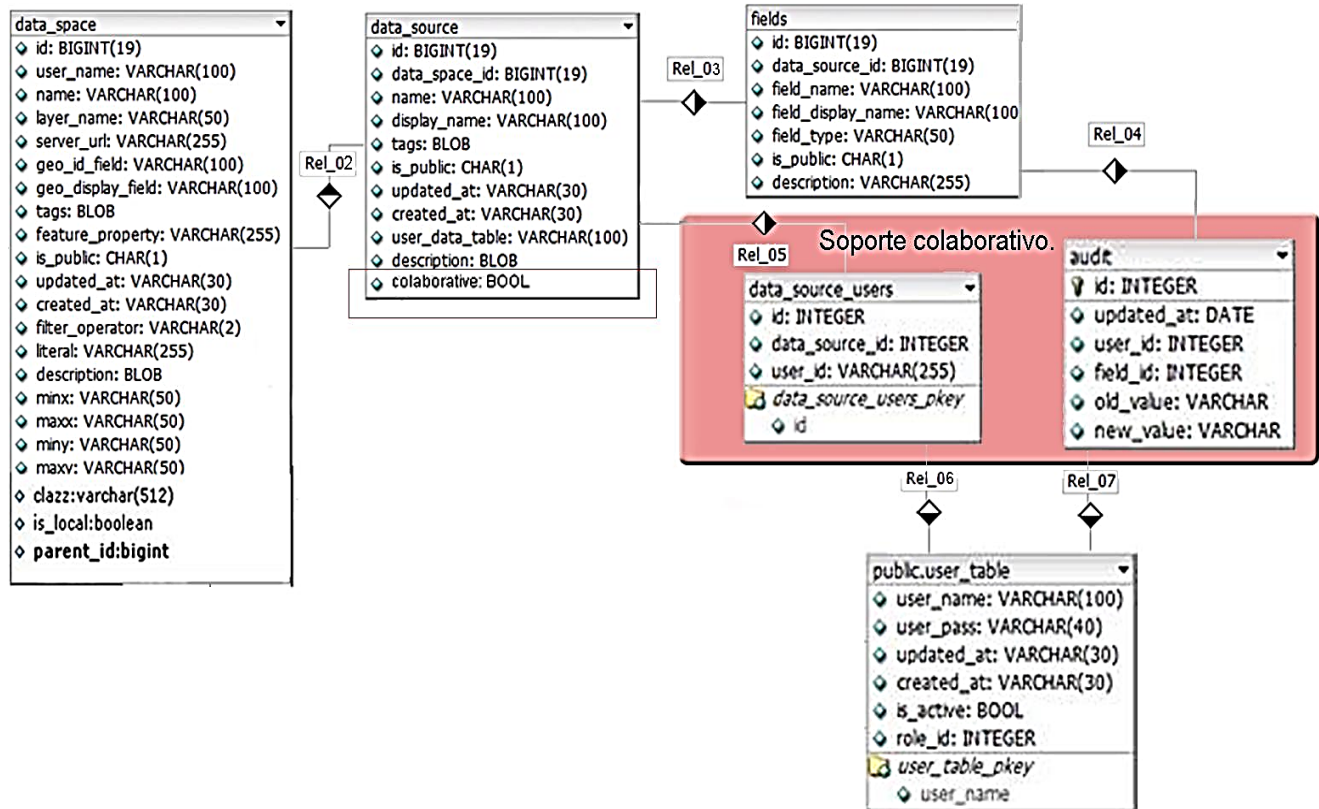


Figura 18. Cambios introducidos en la BD.

3.6 Prototipos

Los prototipos son una representación limitada de un producto, es decir, versiones incompletas del software a desarrollar, permite a las partes probarlo en situaciones reales o explorar su uso, creando así un proceso de diseño de iteración que genera calidad [30].

3.6.1 Prototipo de Interfaz del panel de control (usuario autenticado)

Módulo “Panel de Control”:

El Panel de Control de Tocatoro le permite al usuario muy fácilmente adicionar,

cambiar, actualizar eliminar sus datos (datos asociados a los espacios de datos).

Para acceder a este módulo debe estar autenticado antes, y el proceso es realizado mediante un dialogo de autenticación, donde el usuario introduce su nombre de usuario o dirección de correo electrónico y contraseña como se muestra en el Anexo 17.

Una vez autenticado, se mostrará el módulo llamado “Panel de Control” donde el usuario podrá administrar sus datos y la información que desee publicar.

Es bueno destacar que la pantalla principal de la aplicación se divide en cuatro áreas o regiones, estas se definen como: encabezado, navegador, contenido y notificaciones como se muestra en la Figura 19.

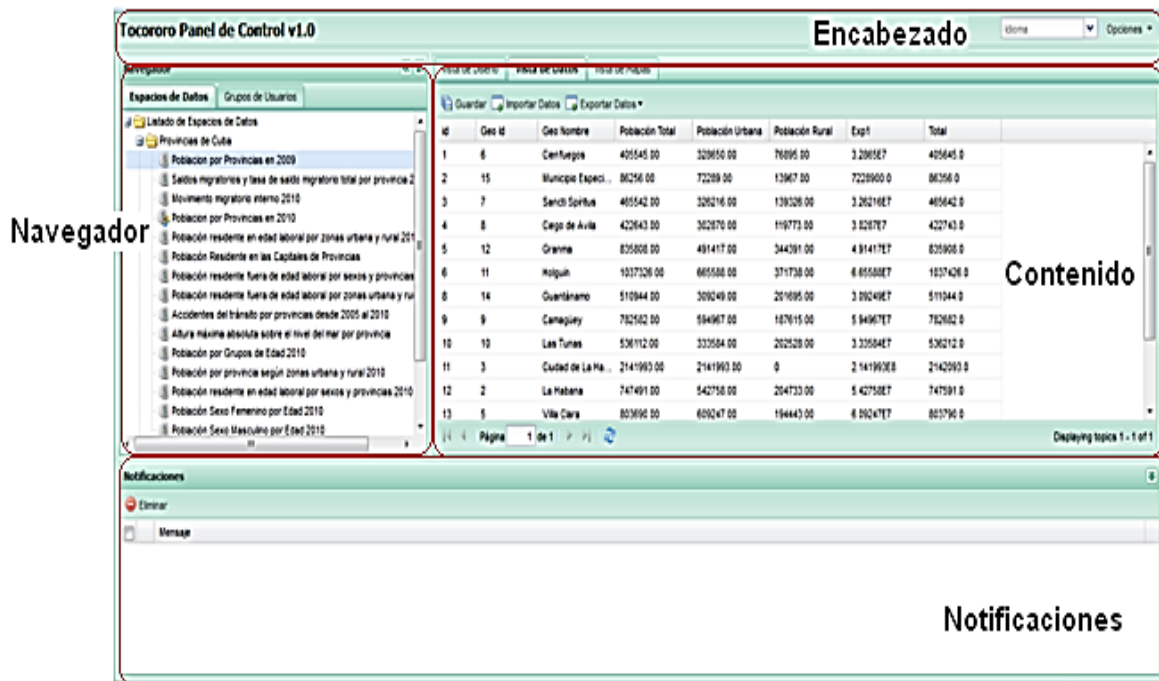


Figura 19. Interfaz del módulo panel de control de la plataforma Tocatoro.

Encabezado: Región que se localiza en la parte superior, en ella se encierra el título del módulo, contiene una lista desplegable para poder seleccionar el idioma de la plataforma como se muestra en la Figura 20 y el menú opciones donde el usuario puede cambiar de sesión, acceder al perfil del usuario y salir de la aplicación, tal y como se muestra en la Figura 21.

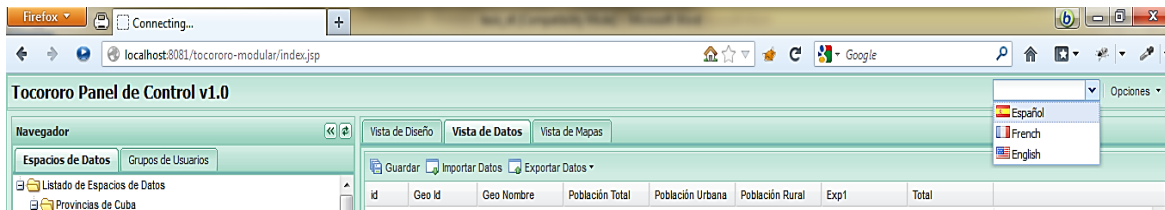


Figura 20. Uso de internacionalización.



Figura 21. Menú opciones.

El panel izquierdo (Navegador): Se presenta en forma de árbol, listando todos los espacios de datos del usuario como se muestra en la Imagen 18.1 del Anexo 18.

Por cada espacio de datos el usuario puede contener varias fuentes de datos Anexo 19. En este panel el usuario puede simplemente con el uso del clic derecho adicionar, editar, eliminar espacios de datos, además puede crear espacios de datos a partir de uno ya existente, esto es un concepto que se maneja para brindar la facilidad al usuario de crear un espacio de datos desde alguno público que ya este compartido por otro usuario del sistema, ver Imagen 18.2 del Anexo 18.

Si se selecciona *Adicionar nuevo espacio de datos* la interfaz quedaría como se muestra en la Imagen 18.3 del Anexo 18.

De esta vista es importante destacar que Tocatoro permite el uso de servidor de mapas externos o simplemente usar el que incluye el sistema, otro aspecto de interés es la opción de filtrado que permite aplicar filtros a los mapas, en la Figura 22 se visualiza la funcionalidad de este filtro.

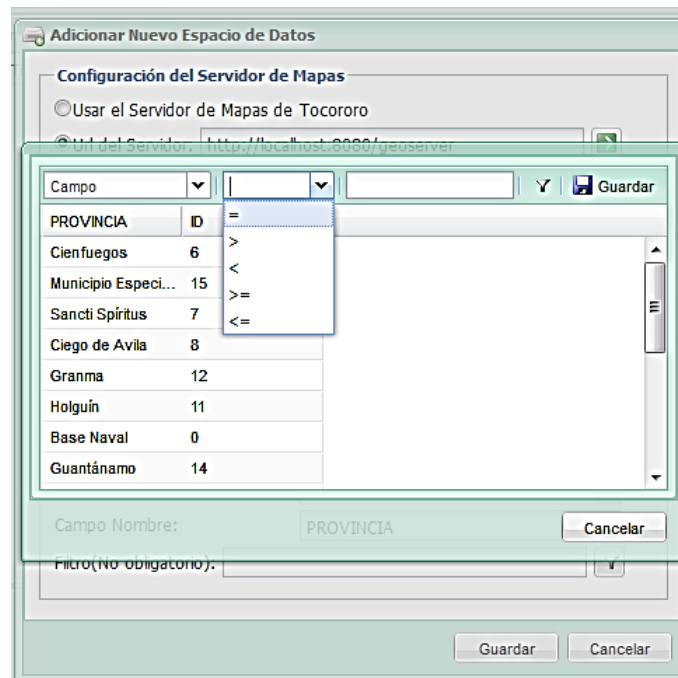


Figura 22. Filtrado del mapa.

El usuario selecciona el campo por el cual desea aplicar el filtro, el operador y el valor a filtrar, luego aplica el filtro y realiza la salva del filtro.

El usuario puede especificar si este espacio de datos va a ser público o no, en caso que sea público otros usuarios del sistema pueden hacer uso de este espacio de datos, además esto permitirá realizar búsquedas sobre el mismo. El sistema permite además crear espacios de datos a partir de uno ya existente que otro usuario haya compartido y además sea público (Imagen 18.4 del Anexo 18).

Además es posible ver las propiedades de cada objeto, esto se realiza dando clic derecho, y en dependencia del tipo (espacio de datos o fuente de datos) se mostrará una ventana con todos los datos asociados al objeto, como se muestra en la Imagen 18.5 del Anexo 18.

El Panel derecho (Contenido): Es el encargado de la carga de todos los datos asociados a los espacios de datos y fuentes de datos, esto es realizado mediante los plug-ins cargados por el núcleo de Tocatoro usando una Arquitectura dirigida por eventos,

Este panel muestra tres vistas o plug-ins: Vista de Diseño, Vista de Datos y Vista

de Mapas.

Vista de Diseño: Permite al usuario crear, editar y eliminar la información o campos a publicar (indicadores, datos estadísticos entre otros).

Vista de Datos: Muestra los valores de estos campos creados.

Vista de Mapas: Visualiza todo esta información en un mapa y se podrá crear nuevos mapas temáticos a partir de esta información.

Notificaciones: Informa al usuario si hubo modificaciones en su plantilla colaborativa, pero además notifica cuando se elimina un campo que está incluido en un campo autocalculado de otro usuario o propio, debido a que por ser un campo expresión hace referencias a otros campos.

3.6.2 Prototipo para definir campos auto calculado

En la vista diseño el usuario conforma una fuente datos, las fuentes de datos están formadas por campos. Los campos que adiciona el usuario pueden ser:

- Entero.
- Decimal.
- Booleano.
- Texto.
- Fecha.
- Enlace.
- Expresión.

El campo de tipo Expresión almacenará valores numéricos que será la combinación de otros campos, ver Imagen 20.1 del Anexo 20.

Una vez seleccionado el campo tipo *Expresión* en la vista de adicionar un nuevo campo, aparecerá el botón **Definir Expresión** y haciendo clic en él se podrá definir el campo auto calculado (Imagen 21.1 del Anexo 21).

La Figura 23 muestra la vista en cual podremos definir el campo autocalculado.

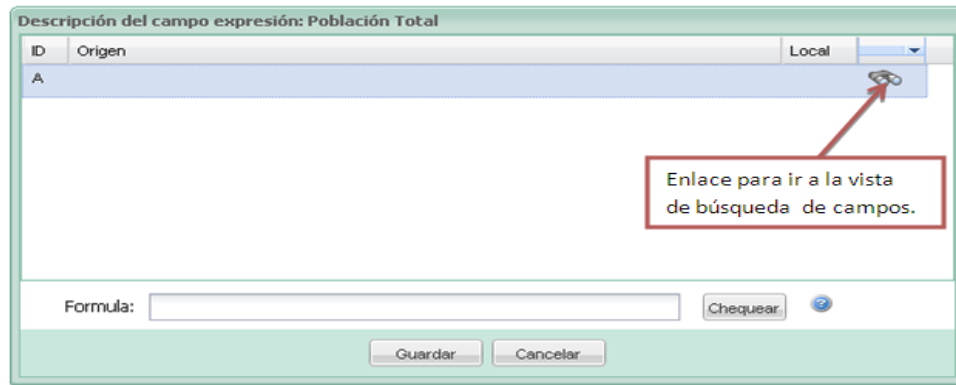


Figura 23. Vista definir campo autocalculado.

En esta ventana tenemos una tabla que inicialmente está vacía y en la cual adicionaremos los campos que van a conformar nuestra expresión.




Al hacer clic sobre la imagen que aparece a la derecha de la fila de la tabla la vista de búsqueda de campos.

La vista de búsqueda de campos le permite al usuario realizar una búsqueda, en toda la información indexada de la BD, de los campos que comparten los usuarios registrados y devuelve un resultado con todos aquellos patrones que tienen coincidencia. El prototipo de resultado de las búsquedas muestra un resultado del patrón de búsqueda que el usuario ha entrado (Imagen 21.2 del Anexo 21).

Además se ofrece la posibilidad de filtrar la búsqueda por usuario, espacio de datos, por fuente de datos y por campos locales.

De la lista de campos el usuario selecciona el que desee y hace clic en el botón **seleccionar**, con esto el campo marcado pasará a la tabla de la vista analizada en la Imagen 21.2 del Anexo 21 y es donde se define la expresión. Después de seleccionados los campos que conformará el campo autocalculado, entonces se podrá definir la expresión, como se muestra en la Figura 24.

Descripción del campo expresión: Población Total

ID	Origen	Local	
A	rebecap.Provincias de Cuba.Poblacion por Provincias en 2010.Poblacion Rural	true	
B	rebecap.Provincias de Cuba.Poblacion por Provincias en 2010.Poblacion Urbana	true	
C			

Formula:




Figura 24. Ejemplo de un campo autocalculado.

En la figura se puede observar los dos campos seleccionados. La primera columna de la tabla indica el identificador del campo para la fórmula que se va a construir. La segunda columna el origen del campo, que no es más que la concatenación del usuario, el espacio de datos, la fuente de datos y el nombre del campo. La tercera columna nos indica si el campo es local, es decir si pertenece a la fuente de datos en la cual se está creando el campo autocalculado.

En la caja de texto **fórmula** es donde se define la expresión o fórmula. La fórmula puede estar compuesta por los identificadores de los campos, por números, por los operadores (+, -, *, /, ^) y por un grupo de funciones matemáticas. Las funciones matemáticas permitidas son:

- **Ceil:** Devuelve el menor entero mayor que el argumento. Si el argumento es un entero devuelve el mismo valor.
- **Floor:** Devuelve el mayor entero menor que el argumento. Si el argumento es un entero devuelve el mismo valor.
- **Sin:** Calcula el seno trigonométrico del ángulo pasado como argumento.
- **Cos:** Calcula el coseno trigonométrico del ángulo pasado como argumento.
- **Tan:** Calcula la tangente trigonométrica del ángulo pasado como argumento.
- **ArcSin:** Calcula el arco seno del valor pasado como parámetro. El valor de retorno es un ángulo que está entre $-\pi/2$ y $\pi/2$.

- **ArcCos:** Calcula el arco coseno del valor pasado como parámetro. El valor de retorno es un ángulo que está entre 0 y pi.
- **ArcTan:** Calcula el arco tangente del valor pasado como parámetro. El valor de retorno es un ángulo que está entre $-\pi/2$ y $\pi/2$.
- **Sinh:** Calcula el seno hiperbólico del valor pasado como parámetro. El seno hiperbólico de x se define como $\frac{(e^x - e^{-x})}{2}$ donde e es el número de Euler.
- **Cosh:** Calcula el coseno hiperbólico del valor pasado como parámetro. El coseno hiperbólico de x se define como $\frac{(e^x + e^{-x})}{2}$ donde e es el número de Euler.
- **Tanh:** Calcula la tangente hiperbólica del valor pasado como parámetro. La tangente hiperbólica de x se define como $\frac{(e^x - e^{-x})}{(e^x + e^{-x})}$ donde e es el número de Euler. Esto no es más que $\frac{\text{Sinh}(x)}{\text{Cosh}(x)}$.
- **Exp:** Retorna el número de Euler elevado al valor pasado como parámetro (e^x).
- **Log:** Calcula el logaritmo natural (base Euler) del valor pasado como parámetro. Si se le pasan dos parámetros, entonces el primero de ellos indica la base y se calcula el logaritmo en base al primer parámetro del valor del segundo parámetro como argumento.
- **Round:** Retorna el valor entero más cercano al valor pasado como parámetro.
- **Random:** Retorna un valor decimal aleatorio entre 0 y 1.
- **Sqrt:** Calcula la raíz cuadrada del valor pasado como parámetro. El argumento tiene que ser mayor o igual que cero.
- **Max:** Devuelve el máximo entre los valores pasados como parámetro.
- **Min:** Devuelve el mínimo entre los valores pasados como parámetro.

- **Power:** Función de dos parámetros que devuelve el valor del primer argumento elevado a la potencia del segundo argumento. Es similar al uso del operador '^'.

Todas estas funciones con su descripción las puede encontrar el usuario haciendo clic en el icono de ayuda que está al lado del botón **Chequear**.

Una vez conformada la expresión matemática, el usuario podrá comprobar si está escrita correctamente haciendo uso del botón **Chequear**. Si la expresión está bien escrita se mostrará un mensaje indicándole que está bien conformada la expresión, de lo contrario se emitirá un mensaje de error. La Figura 25 muestra un ejemplo de expresión correcta e incorrecta con sus respectivos mensajes de chequeo.

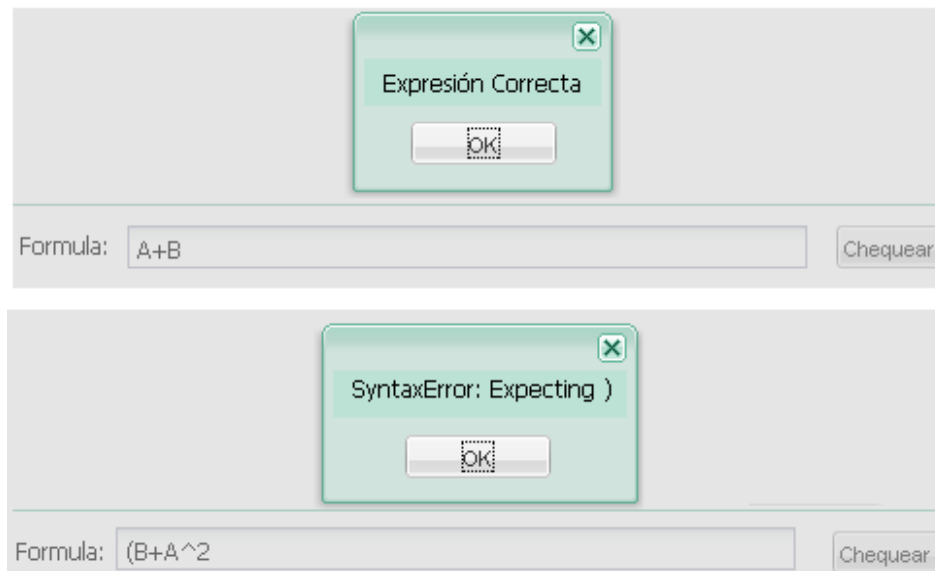


Figura 25. Mensajes de chequeo de la fórmula.

Para guardar la expresión, marcamos el botón **guardar** y se muestra la vista de adicionar un nuevo campo y finalmente se marca la opción guardar.

En la vista diseño aparecerá el nuevo campo como se muestra en la Imagen 21.4 del Anexo 21 y en la vista de datos estará el campo con sus valores ya calculados Anexo 22.

3.6.3 Prototipo del Plug-in de Generación de mapas temáticos

Este Plug-in consta con una Vista de Mapas donde se muestra la información del usuario visualizado en el mapa. Tiene una barra de herramientas donde se tienen algunas opciones tales como:

- Crear un nuevo mapa temático.
- Editar un mapa temático.
- Eliminar un mapa temático.
- Un combobox con el listado de todos los mapas generados.
- Una ventana de leyenda donde mostrar toda la información asociada al mapa temático.

Todo esto se montará sobre un Visor de Mapas (OpenLayers) de forma tal que enriquezca aún más todas las funcionalidades que brinda el visor (zoom, navegación, superposición de capas, visualización de capas desde servicios, etc.). En la Imagen 23. del Anexo 23 se muestra una imagen sobre el diseño inicial.

El proceso de generación de mapas temáticos comienza desde el Plug-in encargado de la Vista de Datos, que es donde el usuario introduce los datos asociados a la fuente de datos creada. La Imagen 23. 1 del Anexo 23 muestra información relacionada con casos de asma por provincias (Nota: los valores no son valores reales, solo son usados para pruebas). A partir de estos datos del usuario el Plug-in de generación de mapas temáticos se encargará de realizar todo el trabajo asociado a estos datos.

3.6.3.1 Crear un nuevo mapa temático

El proceso de crear un nuevo mapa comienza por hacer clic en el botón “Nuevo” y se abrirá una ventana (Imagen 23.2 del Anexo 23).

El usuario deberá especificar el nombre, descripción, etiquetas para la búsqueda, campo temático por el cual se va a generar el mapa (en este caso se escogió “casos de asma”), el tipo de distribución, la cantidad de rangos y la escala de colores en la que se quiere trabajar. Cambiando los valores de la gama de colores

se pueden obtener resultados muy variados y disímiles que le permiten al usuario la personalización de los mapas, El Anexo 23 muestra imágenes de diferentes resultados.

3.6.3.2 *Editar un mapa temático*

Este proceso le permite al usuario modificar los parámetros del mapa. Entre los parámetros a modificar están los intervalos de clasificación, también se le asigna un color para cada uno según el intervalo al que pertenezcan. Los intervalos generados a partir de los valores son configurables permitiendo al usuario ajustes propios a su situación o negocio. Para modificar los intervalos solo se cambian los valores mínimos de los intervalos, automáticamente se recalculan y se genera un nuevo mapa temático.

3.6.3.3 *Eliminar un mapa temático*

Este proceso es bien simple, solo basta con seleccionar el mapa que se desea eliminar y se presiona el botón eliminar. El sistema siempre alertará que se está eliminando un mapa y mostrará el nombre del mapa a eliminar y dará la oportunidad de aceptar o rechazar en caso de una equivocación (Imagen 23.4 del Anexo 23).

3.6.3.4 *Leyenda*

El asistente de mapas facilitará cambiar los colores de la leyenda por otros opcionales ampliando la gama de mapas que se pueden generar con la herramienta (rojo, azul, verde y gris), ver Anexo 23.

El Plug-in de generación de mapas temáticos está desarrollado sobre estándares, brindando solución mediante el uso de los diferentes frameworks líderes para Java, extendiendo nuevas funcionalidades a la plataforma colaborativa Tocatoro. Su interfaz visual muy interactiva y de fácil uso podrá permitir a los usuarios una fácil asimilación de la tecnología y sus componentes.

3.6.4 Prototipo de la funcionalidad de subir y publicar cartografía

Esta extensión es agregada a la Plataforma Tocatoro para ayudar a cada usuario a tener un mejor control de la cartografía que necesita, haciendo de Tocatoro una plataforma más personalizada; donde se prioricen las necesidades específicas de los usuarios unido a un enriquecimiento de los orígenes de datos que otros usuarios también puedan utilizar. La principal característica de la cartografía es que debe ser en formato *shape*.

En la Imagen 25.1 del Anexo 25 se muestra la ventana para cargar la cartografía que se desea importar, la cual debe estar compactada en un fichero de tipo .Zip este a su vez se descomprime en el servidor para su procesamiento. Una vez ahí se realiza un análisis de los ficheros para obtener cuales es la información que nos proporciona y así el usuario pueda decidir qué es lo necesita.

En la Imagen 25.2 del Anexo 25 se muestra la ventana donde se visualizan los campos que se obtuvieron de los ficheros y cuáles serán importados para la BD, brinda la posibilidad al usuario de decidir cuáles campos le interesa tener y así evitar tener en la BD información innecesaria.

Cuando se seleccionan los campo se presiona en el botón “Siguiente” y automáticamente se crean los campos y se guarda la información asociada. Si el proceso se ha realizado correctamente se muestra la ventada mostrada en la Imagen 25.3 del Anexo 25.

3.6.5 Prototipo de la Interfaz de Búsqueda (Usuario anónimo)

En esta interfaz un usuario anónimo podrá realizar las búsquedas de la información temática y los mapas temáticos en todo el espacio de información que los usuarios registrados han subido al servidor y compartan para el uso común en el entorno en que se desarrolle la aplicación (Internet, Intranet o entorno Empresarial).

La interfaz mantendrá una estadística por los mapas que más visitados, así como los últimos adicionados, y siempre teniendo en cuenta que los colores así como la interfaz sea lo más amigable y los resultados sean orientados a brindarle al

usuario una respuesta en el menor tiempo posible. Desde esta interfaz el usuario podrá autenticarse y podrá además ir al Panel de Control y administrar sus datos como se muestra en el Anexo 2528.

Cuando el patrón de búsqueda o frase es entrado por el usuario y presiona el botón buscar, el sistema realiza una búsqueda en toda la información indexada de la BD y devuelve un resultado con todos aquellos patrones que tienen coincidencia.

En la recuperación de textos, la búsqueda de textos se refiere a las técnicas de búsqueda de documentos almacenados en una sola computadora o en una colección en una BD. Este tipo de búsqueda se distingue de las búsquedas basadas en metadatos o partes de los textos representados en BD (tales como títulos, resúmenes, referencias bibliográficas o secciones seleccionadas). En una búsqueda de texto completa, los motores de búsqueda examinan todas las palabras de cada documento indexado y almacenado en correspondencia con los criterios de búsqueda.

El prototipo de resultado de las búsquedas se presenta, mostrando un resultado del patrón de búsqueda que el usuario ha entrado, por ejemplo si el patrón es “enfermedades respiratorias” se devuelven resultados que coinciden no solo con la frase, sino que además se devuelven resultados que solamente se encuentra la palabra “enfermedades”, pero además se devuelven resultados que puedan coincidir con “enfermedad”. Los resultados de la búsqueda no solo se realizan con la debida rapidez de respuesta que el usuario desea sino que incluye alguna otra información que brinde al usuario, información adicional a lo que está intentando recuperar como por ejemplo una pequeña descripción, una imagen asociada, fecha de creación y de actualización o sea elementos por los cuales el usuario pueda saber el nivel de actualización de la información que está buscando, ver Anexo 29.

Cuando el usuario da clic en el enlace de la información recuperada se abre una ventana que muestra un visualizador de los mapas y los datos asociados a estos mapas, como se muestra en el Anexo 30.

La Plataforma Tocatoro como herramienta de publicación de información temática está siendo realizada bajo los estándares, brindando solución mediante el uso de los diferentes frameworks líderes para Java, permitiendo así ser una aplicación escalable y extensible. En su conjunto constituye un ejemplo de integración de diferentes. Su interfaz visual muy interactiva y de fácil uso podrá permitir a los usuarios una fácil asimilación de la tecnología y sus componentes.

3.7 Configuraciones

En comunicaciones o sistemas de computadoras, una configuración es un arreglo de unidades de funciones acordes con su naturaleza, número y características. A menudo, las configuraciones pertenecen a un hardware, software o firmware. Las configuraciones afectan las funciones y el rendimiento.

En informática, archivos de configuración constituyen los ajustes iniciales para algunos programas informáticos. Se utilizan para las aplicaciones de usuario, procesos de servidor y configuración del sistema operativo. Los archivos se escriben a menudo en ASCII (rara vez UTF-8) y orientados a línea, con líneas que terminan en una nueva línea o retorno, dependiendo del sistema operativo. Se pueden considerar una base de datos simple [31].

Algunas aplicaciones proporcionan herramientas para crear, modificar y verificar la sintaxis de sus archivos de configuración, los cuales a veces tienen interfaces gráficas. Para otros programas, los administradores del sistema pueden esperar para crear y modificar los archivos a mano usando un editor de texto. Para los procesos del servidor y la configuración del sistema operativo, a menudo no existe ninguna herramienta estándar, pero los sistemas operativos pueden proporcionar sus propias interfaces gráficas.

Algunos programas sólo leen sus archivos de configuración en el arranque. Otros revisan periódicamente los archivos de configuración para los cambios. Los usuarios pueden recargar algunos programas para volver a leer los archivos de configuración y aplicar los cambios en el proceso actual, o incluso para leer archivos arbitrarios como un archivo de configuración. No hay criterios definitivos o

convenciones fuertes.

En la plataforma colaborativa Tocatoro los archivos de configuración, forman parte de las configuraciones iniciales, configuración de BD, caminos físicos para el almacenamiento temporal o URLs de enlaces a componentes de la plataforma. Seguidamente hacemos un desglose de los ficheros de configuración de los módulos más importantes.

3.7.1 Módulo Panel de Control

3.7.1.1 *config.properties*

Este fichero almacena la configuración inicial del módulo: Panel de Control.

El siguiente segmento se refiere a la configuración del servidor de mapas incluido en Tocatoro:

```
#Url del geoserver por defecto que es el que incluye Tocatoro
geoserverUrl=http://localhost:8080/geoserver

#Capa base a utilizar
basicLayer=gmx:Cuba-ProvinciasOK

#Atributo que se desea mostrar de esta capa
geoDisplayLabel=PROVINCIA

#Boundingbox
#left
bLeft=-84.953
#bottom
bBottom=19.826
#Right
bRight=-74.131
#Top
bTop=23.276

#Proyección de los mapas
srs=EPSG:4326

#Camino físico donde se encuentra desplegado el panel de control
realPath=d:/Dev/workspace-sts/tocatoro-modular/war

#Camino donde se guardaran los SLD
path=/sld/

#Camino físico donde se guardaran las imágenes en miniatura
thumbsPath=/thumbs/

#Camino físico temporalmente utilizado también para los ficheros a exportar
tmpExportPath=/tmp/

#Camino físico donde se guarda el fichero de configuración de los plugins
pluginsConfigPath=/WEB-INF/classes/
```



```
#imagen a mostrar cuando son Fuentes de Datos y no mapas
dataSourceImage=datos.png

#Url del perfil del usuario
profileUrl=http://localhost:8081/tocororo/profile.html

#Url del módulo de Búsqueda
#serachUrl=http://localhost:8080/tocororo/search/index.jsp
```

3.7.1.2 *jdbc.properties*

Este fichero almacena la configuración de la Base de Datos del módulo.

```
#Configuracion de JDBC
jdbc.driverClassName=org.postgresql.Driver
jdbc.url=jdbc:postgresql://localhost/tocororo
jdbc.username=postgres
jdbc.password=postgres

#Configuracion de Hibernate
hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
hibernate.show_sql=true
hibernate.generate_statistics=false
hibernate.hbm2ddl.auto=update

#Cnfiguracion del pool de conexiones
dbcp.initialSize=2
dbcp.maxActive=100
dbcp.maxIdle=10
dbcp.minIdle=10
dbcp.maxWait=-1
dbcp.validationQuery=SELECT 1+1
```

3.7.1.3 *Hibernate.cfg.xml*

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration
DTD 3.0//EN" "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
<property name="hibernate.show_sql">true</property>
<property name="hibernate.hbm2ddl.auto">update</property>
<!-- HSearch settings -->
<property name="hibernate.search.default.indexBase">d:\Dev\index</property>
<property
name="hibernate.search.default.directory_provider">org.hibernate.search.store.FSD
irectoryProvider</property>
<!--<property name="hibernate.search.indexing_strategy">manual</property> -->
<mapping class="gmx.iderc.tocororo.server.entity.User"/>
<mapping class="gmx.iderc.tocororo.server.entity.DataSpace"/>
<mapping class="gmx.iderc.tocororo.server.entity.DataSource"/>
<mapping class="gmx.iderc.tocororo.server.entity.DataSourceTemplate"/>
<mapping class="gmx.iderc.tocororo.server.entity.UserField"/>
<mapping class="gmx.iderc.tocororo.server.entity.Map"/>
<mapping class="gmx.iderc.tocororo.server.entity.Profile"/>
<mapping class="gmx.iderc.tocororo.server.entity.Role"/>
<mapping class="gmx.iderc.tocororo.server.entity.Audit"/>
```

```
<mapping class="gm.x.iderc.tocororo.server.entity.DataSourceUser"/>
<mapping class="gm.x.iderc.tocororo.server.entity.Session"/>
<mapping class="gm.x.iderc.tocororo.server.entity.ExpresionField"/>
<mapping class="gm.x.iderc.tocororo.server.entity.ExpresionFieldOperand"/>
<mapping class="gm.x.iderc.tocororo.server.entity.Message"/>
<mapping class="gm.x.iderc.tocororo.server.entity.UserGroup"/>
<mapping class="gm.x.iderc.tocororo.server.entity.UserGroupUser"/>
</session-factory>
</hibernate-configuration>
```

3.7.1.4 PluginsContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<list>
  <gm.x.iderc.tocororo.client.cpanel.bean.PluginBean>
    <name>DsViewerPlugin</name>
    <title></title>
    <description></description>
    <pluginUrl>dsviewer.jsp</pluginUrl>
    <propertyFilePath>gm.x.iderc.tocororo.client.dsviewer.language.Language</propertyFilePath>
    <property>contentMapViewTabTitleText</property>
    <hidden>true</hidden>
  </gm.x.iderc.tocororo.client.cpanel.bean.PluginBean>
  <gm.x.iderc.tocororo.client.cpanel.bean.PluginBean>
    <name>DesignPlugin</name>
    <title></title>
    <description></description>
    <pluginUrl>designer.jsp</pluginUrl>
    <propertyFilePath>gm.x.iderc.tocororo.client.designer.language.Language</propertyFilePath>
    <property>contentDesignViewTabTitleText</property>
    <hidden>false</hidden>
  </gm.x.iderc.tocororo.client.cpanel.bean.PluginBean>
  <gm.x.iderc.tocororo.client.cpanel.bean.PluginBean>
    <name>DataSheetPlugin</name>
    <title></title>
    <description></description>
    <pluginUrl>dataviewer.jsp</pluginUrl>
    <propertyFilePath>gm.x.iderc.tocororo.client.dataviewer.language.Language</propertyFilePath>
    <property>contentDataSheetViewTabTitleText</property>
    <hidden>false</hidden>
  </gm.x.iderc.tocororo.client.cpanel.bean.PluginBean>
  <gm.x.iderc.tocororo.client.cpanel.bean.PluginBean>
    <name>MapPlugin</name>
    <title></title>
    <description></description>
    <pluginUrl>mapviewer.jsp</pluginUrl>
    <propertyFilePath>gm.x.iderc.tocororo.client.mapviewer.language.Language</propertyFilePath>
    <property>contentMapViewTabTitleText</property>
    <hidden>true</hidden>
  </gm.x.iderc.tocororo.client.cpanel.bean.PluginBean>
  <!--<gm.x.iderc.tocororo.client.cpanel.bean.PluginBean>
    <name>SimplePlugin</name>
    <title></title>
    <description></description>
    <pluginUrl>simple-addon.jsf</pluginUrl>
    <propertyFilePath>gm.x.iderc.tocororo.client.simpleaddon.language.Language</proper
```

```
tyFilePath>
<property>contentSimpleAddonViewTabTitleText</property>
</gmx:iderc.tocororo.client.cpanel.bean.PluginBean>-->
</list>
```

3.7.2 Módulo de Búsqueda

3.7.2.1 Config.properties

```
#Url del geoserver por defecto que es el que incluye Tocatoro
geoserverUrl=http://localhost:8080/geoserver
#Capa base
basicLayer=gmx:Cuba-ProvinciasOK
#GeoDisplaylabel
geoDisplayLabel=PROVINCIA
#Bounding
#left
bLeft=-84.953
#bottom
bBottom=19.826
#Right
bRight=-74.131
#Top
bTop=23.276
#Projection
srs=EPSG:4326
#Camino donde se guardaran los SLD
path=d:\\Dev\\workspace-sts\\tocororo-modular\\war\\sld\\
#Camino fisico donde se guardaran las imagenes en miniatura
thumbsPath=d:\\Dev\\workspace-sts\\tocororo-modular\\war\\thumbs\\
#Carpeta donde estan guarddas las imagenes en miniatura, tiene que coincidir con
el camino fisico
relativeThumbsPath=thumbs
#imagen a mostrar cuando son Fuentes de Datos y no mapas
dataSourceImage=datos.png
#imagen a mostrar cuando no tiene imagen asociada
noImage=noImage.png
#Url del modulo viewer
viewerUrl=http://localhost:8081/tocororo-viewer/viewer.html
#Url del modulo CPanel
cpanelUrl=http://localhost:8081/tocororo/index.jsf
#Url del modulo de registro
registerUrl=http://localhost:8081/register/index.jsf
#Url del modulo login
loginUrl=http://localhost:8080/tocororo/login.html
#Url del modulo de Busqueda
#serachUrl=http://localhost:8080/tocororo/search/index.jsp
```

3.7.2.2 jdbc.properties

```
jdbc.driverClassName=org.postgresql.Driver
jdbc.url=jdbc:postgresql://localhost/tocororo
jdbc.username=postgres
jdbc.password=postgres
hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
hibernate.show_sql=true
hibernate.generate_statistics=false
hibernate.hbm2ddl.auto=update
```

```
dbcp.initialSize=2
dbcp.maxActive=100
dbcp.maxIdle=10
dbcp.minIdle=10
dbcp.maxWait=-1
dbcp.validationQuery=SELECT 1+1

acquireIncrement=1
minPoolSize=2
maxPoolSize=100
maxIdleTime=300

#jdbc.driverClassName=com.mysql.jdbc.Driver
#jdbc.url=jdbc:mysql://localhost/test
#jdbc.username=root
#jdbc.password=
```

3.7.2.3 hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration
DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
<property name="hibernate.show_sql">true</property>
<property name="hibernate.hbm2ddl.auto">update</property>
<!-- HSearch settings -->
<property name="hibernate.search.default.indexBase">d:\Dev\index</property>
<property
name="hibernate.search.default.directory_provider">org.hibernate.search.store.FSD
irectoryProvider</property>
<!--<property name="hibernate.search.indexing_strategy">manual</property> -->
<mapping class="gmx.iderc.tocororo.server.entity.User"/>
<mapping class="gmx.iderc.tocororo.server.entity.DataSpace"/>
<mapping class="gmx.iderc.tocororo.server.entity.DataSource"/>
<mapping class="gmx.iderc.tocororo.server.entity.DataSourceTemplate"/>
<mapping class="gmx.iderc.tocororo.server.entity.UserField"/>
<mapping class="gmx.iderc.tocororo.server.entity.Map"/>
<mapping class="gmx.iderc.tocororo.server.entity.Profile"/>
<mapping class="gmx.iderc.tocororo.server.entity.Role"/>
<mapping class="gmx.iderc.tocororo.server.entity.Audit"/>
<mapping class="gmx.iderc.tocororo.server.entity.DataSourceUser"/>
</session-factory>
</hibernate-configuration>
```

3.7.3 Módulo de visualización de mapas y datos temáticos (Visor)

3.7.3.1 Config.properties

```
#Url del geoserver por defecto que es el que incluye Tocatoro
geoserverUrl=http://localhost:8080/geoserver
#Capa base
```

```
basicLayer=gmx:Cuba-ProvinciasOK
#GeoDisplaylabel
geoDisplayLabel=PROVINCIA
#Bounding
#left
bLeft=-84.953
#bottom
bBottom=19.826
#Right
bRight=-74.131
#Top
bTop=23.276
#Projection
srs=EPSG:4326
#Camino donde se guardaran los SLD
path=d:\\Dev\\workspace-sts\\tocororo-modular\\war\\sld\\
#Camino fisico donde se guardaran las imagenes en miniatura
thumbsPath=d:\\Dev\\workspace-sts\\tocororo-modular\\war\\thumbs\\
#Camino fisico donde se guardaran temporalmente los fichero a exportar
tmpExportPath=d:\\Dev\\workspace-sts\\tocororo-modular\\war\\tmp\\
#Camino fisico donde se guarda el fichero de configuracion de los plugins
pluginsConfigPath=d:\\Dev\\workspace-sts\\tocororo-modular\\war\\WEB-INF\\classes\\
#Carpeta donde estan guarddas las imagenes en miniatura, tiene que coincidir con
el camino fisico
relativeThumbsPath=thumbs
#imagen a mostrar cuando son Fuentes de Datos y no mapas
dataSourceImage=datos.png
#Url del servlet que genera los sld
sldUrl=http://localhost:8080/tocororo-modular/generateSldServlet
#Url del servlet que devuelve el sld
getSldUrl=http://127.0.0.1:8888/getSldServlet
#Url del modulo viewer
#viewerUrl=http://localhost:8081/tocororo/viewer.html
#Url del modulo CPanel
#cpanelUrl=http://localhost:8080/tocororo/cpanel.html
#Url del modulo login
#loginUrl=http://localhost:8080/tocororo/login.html
#Url del modulo de Busqueda
#serachUrl=http://localhost:8080/tocororo/search/index.jsp
```

3.7.3.1 *jdbc.properties*

```
jdbc.driverClassName=org.postgresql.Driver
jdbc.url=jdbc:postgresql://localhost/tocororo
jdbc.username=postgres
jdbc.password=postgres
hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
hibernate.show_sql=true
hibernate.generate_statistics=false
hibernate.hbm2ddl.auto=update

dbcp.initialSize=2
dbcp.maxActive=100
dbcp.maxIdle=10
dbcp.minIdle=10
dbcp.maxWait=-1
dbcp.validationQuery=SELECT 1+1

acquireIncrement=1
```

```
minPoolSize=2
maxPoolSize=100
maxIdleTime=300

#jdbc.driverClassName=com.mysql.jdbc.Driver
#jdbc.url=jdbc:mysql://localhost/test
#jdbc.username=root
#jdbc.password=
```

3.8 Conclusiones

En este capítulo se mostró la implementación y los prototipos de las principales funciones de la plataforma colaborativa Tocatoro como: Importar y publicar cartografía, permitir campos de tipo expresión autocalculada, generación de mapas temáticos y plantillas colaborativas. En la plataforma colaborativa los archivos de configuración, forman parte de las configuraciones iniciales, configuración de BD, caminos físicos para el almacenamiento temporal o URLs de enlaces a componentes de la plataforma por lo que se hizo un desglose de los ficheros de configuración por cada módulo.

Conclusiones

Como resultado de la presente investigación, pudo arribarse a las conclusiones generales siguientes:

1. Para la realización del trabajo se hizo un análisis crítico de la literatura que permitió sentar las bases teóricas en materia de buenas prácticas existentes para desarrollar una plataforma colaborativa en la nube.
2. Se identificaron y seleccionaron las tecnologías de SL existentes más apropiadas para el desarrollo de la plataforma colaborativa en la nube que permitiera gestionar y representar información estadística, medioambiental e indicadores.
3. Se propuso el diseño de una arquitectura escalable y extensible que permite soportar una plataforma colaborativa en la nube.
4. Se implementó la versión inicial de un prototipo de arquitectura que brindara solución al problema de investigación.

Recomendaciones

1. Agregar la funcionalidad de control de versiones o histórico de los datos que permita gestionar los datos en el tiempo.
2. Crear del módulo de administración de la plataforma.
3. Realizar el análisis espacial de los datos, es decir que no solamente se pueden hacer análisis sobre los datos sino también sobre los mapas creados.
4. Se recomienda integrarla con un portal web por ejemplo Liveray.

REFERENCIAS

- [1] Funcionamiento de una aplicación web. Tomado de:
http://livedocs.adobe.com/dreamweaver/8_es/using/gs_12_u5.htm,
noviembre del 2007.
- [2] ¿Cómo empezó el Cómputo Cloud? Tomado de:
<http://www.itnews.ec/news/000396.aspx>, enero del 2012.
- [3] Gartner Says Cloud Computing Will Be As Influential As E-business. Tomado
De:www.gartner.com/it/page.jsp?id=707508, septiembre del 2012.
- [4] Beyerlein, M; Freedman, S.; McGee, G.; Moran, L. (2002). Beyond Teams:
Building the Collaborative Organization. The Collaborative Work Systems
series, Wiley.
- [5] Moya, Iván. Desarrollo de Software Colaborativo. Tomado de:
http://www.proveedoresempresariales.com/rpe_swcolaborativo.htm, abril
del 2012.
- [6] ¿Qué es Sistema de Información y Sistema Colaborativo? Tomado de:
<http://elmejorproyecto2010.blogspot.com/2010/04/que-es-sistema-de-informacion-y-sistema.html>, septiembre del 2012.
- [7] Mapas Temáticos. Tomado de:
<http://geohistoriaymas.wordpress.com/2011/05/22/mapas-tematicos/>,
febrero del 2012.
- [8] Darío, Julian Giraldo Ocampo (2010) Información Georeferenciada. Tomado
de: <http://juliangiraldo.wordpress.com/sig/georreferenciada/>, febrero del
2012.
- [9] Moreiro González, JE. Lenguaje natural e indización automatizada/ [et.al].--
[s.l]: Cienc Inform, 1999.--p.30.
- [10] Servidores de Mapas. Tomada de:
<http://dspace.ups.edu.ec/bitstream/123456789/56/10/Capitulo4.pdf>, marzo

del 2012.

- [11] Geoserver, Revista OsGeoLive. Tomado de:
http://live.osgeo.org/es/overview/geoserver_overview.html, Febrero del 2012.
- [12] Méndez, Xurxo, Tutorial de OpenLayers (I) – Introducción. Tomado de:
<http://blog.sonxurxo.com/2011/03/29/tutorial-de-openlayers-i-introduccion>, 29 de marzo del 2012.
- [13] PostgreSQL Práctico. Tomado de:
<http://www.sobl.org/traduccion/practical-postgres/node12.html>, noviembre del 2007.
- [14] Plakosh, Y. How to setup Apache Tomcat for remote debugging, Tomado de: <http://yplakosh.blogspot.com/2008/10/tomcat-debug.html>, octubre del 2010.
- [15] The history of Java Technology. Tomado de:
<http://www.java.com/en/javahistory/>, abril del 2012.
- [16] The Java Tutorials. Tomado de:
<http://java.sun.com/docs/books/tutorial/index.html>, abril del 2012.
- [17] Munz, Stefan. JavaScript/DOM. Tomado de:
<http://es.selfhtml.org/introduccion/tecnologias/javascript.htm>, diciembre del 2007.
- [18] Patrón MVP. Tomado de: <http://www.neleste.com/modelo-vista-controlador-y-algunas-variantes/>, 2 de febrero del 2012.
- [19] Hibernate. Tomado de: <http://www.hibernate.org/subprojects/>, 2 de febrero del 2012.
- [20] Sánchez, José M., Introducción a RichFaces. Tomado de:
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=richFacesJsflIntro>, 1 de febrero del 2010.
- [21] Google Web Toolkit, Tomado de:
http://es.wikipedia.org/wiki/Google_Web_Toolkit, 22 de Septiembre del

2012.

- [22] Ochoa, Javier, Mvp4g multiple presenters – Part 1. Tomado de:
<http://www.summa-tech.com/2011/04/27/mvp4g-multiple-presenters-part-1/>, 27 de abril del 2011.
- [23] ¿Qué es un plug-in? - Definición de plug-in. Tomado de:
<http://www.masadelante.com/faqs/plug-in>, 9 de agosto del 2012.
- [24] El Proceso Unificado de Desarrollo de Software/ Ivar Jacobson... [et.al] .--
La Habana: Editorial Félix Varela, 2004.--Tomo I.
- [25] Ferré Grau, Xavier. Desarrollo orientado a objetos con UML. Tomado de:
<http://www.clikear.com/manuales/uml/introduccion.asp>, enero del 2008.
- [26] Chappell, David. Desarrollo de aplicaciones distribuidas. Tomado de:
<http://www.microsoft.com/spanish/msdn/articulos/archivo/081102/voices/dn capas.asp>, enero del 2008.
- [27] Rodríguez Terrero, Príapo Nicolás. Aplicaciones Distribuidas 3 Capas.
Tomado de:
http://www.elguille.info/colabora/NET2005/Sagara_AplicacionesDistribuidas3Capas.htm, enero del 2010.
- [28] Muñoz, Oscar. Arquitectura de aplicaciones Web. Conferencia de
Seminarios Especiales I/ Oscar Muñoz.--Cienfuegos: UCF, 2004.-- [s.p].
- [29] El Proceso Unificado de Desarrollo de Software/ Ivar Jacobson... [et.al].--
México: Addison-Wesley, 2000.--p.116.
- [30] Ibarguengoitia Guadalupe, Oktaba Hanna. Ingeniería de Software
pragmática. Tomado de:
<http://es.scribd.com/doc/111600953/25/Prototipo-de-interfaz-de-usuario>,
febrero 2012.
- [31] Configuración (Informática). Tomado de:
http://es.wikipedia.org/wiki/Configuracion_informatica, octubre 2012.

BIBLIOGRAFÍA

1. Bolstad, P. *GIS Fundamentals: A first text on Geographic Information Systems*,.Second Edition. White Bear Lake, MN: Eider Press, 2005.
2. Bosque, Sendra J. *Sistemas de Información Geográfica*. Madrid, 1992.
3. Esselink, Bert, and John Benjamins. "A Practical Guide to Localization." 2000.
4. Garrett, J. J. "Adaptive Path." 2005.
<http://www.adaptivepath.com/ideas/essays/archives/000385.php> (accessed octubre 9, 2012).
5. Giametta, Chris. *Pro Flex on Spring*. Estados Unidos, 2009.
6. Gilder, George. " Las fábricas de información." *Wired*, 2006.
7. Graells, Pere Marqués. *La Web 2.0 y sus aplicaciones didácticas*. . 2007.
<http://www.peremarques.net/web20.htm> (accedida agosto 10, 2012).
8. Jacobson, Ivar. *El Proceso Unificado de Desarrollo de Software*. México: Addison-Wesley, 2000.
9. Jan, Machacek, Vukotic Aleksa, Chakraborty Anirvan, and Ditt Jessica. *Pro Spring 2.5*. Estados Unidos, 2008.
10. Jsakalos. "AJAX With, Resources and Information." 2007.
<http://www.ajaxwith.com/visit/Ext-JS-JavaScript-Library.html>.
11. Lee, Tenorio F. *Informe sobre la disponibilidad de computadoras y redes en los CES, en relación con la formación de profesionales curso 2002-2003*. . Ciudad de La Habana, 2003.
12. Muehrcke, Phillip, Juliana O. Muehrcke, and A. Jon Kimerling. *Map Use: Reading, Analysis and Interpretation*. JP Publications, 2001.
13. Muehrcke, Phillip, Juliana O. Muehrcke, and A. Jon Kimerling. *Map Use: Reading, Analysis, and Interpretation*. JP Publications, 2001.
14. *ORM - Object Relational Mapping* Bogotá 2010
15. OsGeoLive . *Geoserver, Web Service* . 2010.
http://live.osgeo.org/es/overview/geoserver_overview.html. (accedida

- noviembre 2, 2012).
16. Proceedings of the 9th WSEAS International Conference on Applied Informatics And Communications 383-398 2009
 17. Robinson, , Arthur H. *Early Thematic Mapping in the History of Cartography*, .Universidad de Chicago, 1982.
 18. Rubio, Daniel. *Pro Spring Dynamic, modules for OSGi™ Service Platforms*. Estados Unidos, 2009.
 19. Sache, A. "Open Source Science Journal." In *Choosing the Open-Source Framework for Web Application Development*, 5-22. 2010.
 20. Schilleng, M. A. "Towards a general modular systems theory and its application to inter-firm product modularity." (Academy of Management Review) 25:312-334 (2000).
 21. Terán, Karina, and Julio Aldama. *Sistemas Colaborativo*. 2008. http://www.publicaciones.ujat.mx/publicaciones/revista_dacb/Acervo/v4n2OL/v4n2a4-ol/index.html. (accessed septiembre 4, 2012).
 22. *Una carrera cada vez más veloz. Aplicaciones de las TICs a la educación con alcance para todos y en los diferentes sistemas y niveles de la educación* 2005 http://www.cubaminrex.cu/Sociedad_Informacion/Articulos.htm#12.

Anexos

Anexo 1. *Descripción del caso de uso autenticar usuario*

Nombre del caso de uso	Autenticar Usuario.
Actores	Usuario Registrado (inicia).
Resumen	El caso de uso se inicia cuando el usuario desea iniciar su sección para eliminar, insertar y modificar información y el sistema le brinda la posibilidad de actualizarlo con facilidades, finalizando así el caso de uso.
Precondiciones	El usuario administrador tiene que estar previamente registrado.
Post-condiciones	-
Requisitos especiales	

Anexo 2. *Descripción del caso de uso gestionar el registro de usuario*

Nombre del caso de uso	Gestionar el Registro de usuario.
Actores	Usuario Registrado (inicia).
Resumen	El caso de uso inicia cuando el usuario registrado desea entrar al sistema para adicionar, modificar o eliminar un nuevo usuario. Para ello debe ingresar su identificador y su contraseña. Una vez introducida esta información el sistema chequea si son válidos, si esto ocurre el administrador podrá acceder a las opciones del sistema que le corresponden, en caso contrario el sistema muestra un mensaje de error denegando el acceso a él, finalizando así el caso de uso.
Precondiciones	El usuario debe estar autenticado Tiene que estar almacenada la información del Usuario, en caso de eliminar y/o modificar.

Post-condiciones	<p>Se actualiza la información delos Usuarios.</p> <p><u>Si acción: insertar</u>, se inserta información de los datos: nombre de usuario, contraseña, dirección electrónica. El sistema valida que los datos sean correctos. En caso de ser correctos los datos salva la información en la BD y muestra un mensaje informando que la operación se efectuó correctamente. En caso contrario muestra un mensaje de error.</p> <p><u>Si acción: modifica</u>, varía la información de las fuentes de datos.</p> <p><u>Si acción: eliminar</u>, el sistema muestra un mensaje preguntado si el usuario está seguro que desea eliminar al usuario seleccionado, en caso de aceptar se elimina el usuario seleccionado, en caso contrario se cancela la operación.</p>
Requisitos especiales	
Excepciones	
En caso de que existan problemas con la conexión a la BD, el sistema lanzara una excepción.	
Si a la hora de insertar un espacio de datos en la BD existe otro con el mismo nombre entonces se lanzará una excepción puesto que no pueden existir dos Espacios de Datos con el mismo nombre.	

Anexo 3. *Descripción del caso de uso gestionar espacios de datos*

Nombre del caso de uso	Gestionar Espacios de Datos.
Actores	Usuario Registrado (inicia).
Resumen	El caso de uso se inicia cuando el Usuario Registrado desea actualizar información de los espacios de datos y para eso puede eliminar, insertar, modificar, y el sistema le brinda la posibilidad de actualizarlo con facilidades, finalizando así el caso de uso.
Precondiciones	El usuario debe estar autenticado Tiene que estar almacenada la información de los Espacios de Datos, en caso de eliminar y/o modificar.
Post-condiciones	Se actualiza la información de los datos de los Espacios de Datos. Si acción: insertar, se inserta información de los espacios de datos. Si acción: modifica, varía la información de los espacios de datos. Si acción: eliminar, se elimina información de los espacios de datos.
Requisitos especiales	
Excepciones	
En caso de que existan problemas con la conexión a la BD, el sistema lanzara una excepción.	

Si a la hora de insertar un espacio de datos en la BD existe otro con el mismo nombre entonces se lanzará una excepción puesto que no pueden existir dos espacios de datos con el mismo nombre.

Anexo 4. *Descripción del caso de uso gestionar fuentes de datos*

Nombre del caso de uso	Gestionar fuentes de datos
Actores	Usuario Registrado (inicia).
Resumen	El caso de uso se inicia cuando el usuario registrado desea actualizar información de las fuentes de datos y para eso puede eliminar, insertar y modificar, y el sistema muestra el listado de todas las fuentes de datos por cada espacio de datos del usuario y brinda la posibilidad de actualizarlo con facilidades, finalizando así el caso de uso.
Precondiciones	El usuario debe estar autenticado Tiene que estar almacenada la información de las fuentes de datos, en caso de eliminar y/o modificar.

Post-condiciones	<p>Se actualiza la información de las fuentes de datos.</p> <p><u>Si acción: insertar</u>, se inserta información de los datos: Nombre de la fuente de datos y nombre para mostrar. El sistema valida que los datos sean correctos. En caso de ser correctos los datos salva la información en la BD y muestra un mensaje informando que la operación se efectuó correctamente. En caso contrario muestra un mensaje de error.</p> <p><u>Si acción: modifica</u>, varía la información de las fuentes de datos.</p> <p><u>Si acción: eliminar</u>, el sistema muestra un mensaje preguntado si el actor está seguro que desea eliminar la fuente de datos seleccionada, en caso de aceptar se elimina la fuente de datos seleccionada, en caso contrario se cancela la operación.</p>
Requisitos especiales	
Excepciones	
En caso de que existan problemas con la conexión a la BD, el sistema lanzará una excepción.	
Si a la hora de insertar una fuente de datos en la BD existe otro con el mismo nombre entonces se lanzará una excepción puesto que no pueden existir dos fuentes de datos con el mismo nombre.	

Anexo 5. Descripción del caso de uso gestionar campos

Nombre del caso de uso	Gestionar campos
Actores	Usuario Registrado (inicia).
Resumen	El caso de uso se inicia cuando el usuario registrado es autenticado y da clic derecho sobre una determinada fuente de datos seleccionando la opción Vista de Diseño. El sistema muestra la pestaña de la Vista de Diseño donde el usuario podrá gestionar los campos, también se muestra el listado de todos los campos para la fuente de datos seleccionada finalizando así el caso de uso.
Precondiciones	El usuario debe estar autenticado Tiene que estar almacenada la información de los campos, en caso de eliminar y/o modificar.
Post-condiciones	Se actualiza la información de los campos. <u>Si acción: insertar</u> , se inserta información de los datos: nombre del campo, tipo y descripción. El sistema valida que los datos sean correctos. En caso de ser correctos los datos salva la información en la BD y muestra un mensaje informando que la operación se efectuó correctamente. En caso contrario muestra un mensaje de error. <u>Si acción: modificar</u> , varía la información de los campos. <u>Si acción: eliminar</u> , el sistema muestra un mensaje preguntado si el actor está seguro que desea eliminar el campo seleccionado, en caso de aceptar se elimina el campo seleccionado, en caso contrario se cancela la operación.

Requisitos especiales	
Excepciones	
En caso de que existan problemas con la conexión a la BD, el sistema lanzara una excepción.	
Si a la hora de insertar un campo en la BD existe otro con el mismo nombre entonces se lanzará una excepción puesto que no pueden existir dos campos con el mismo nombre.	

Anexo 6. *Descripción del caso de uso gestionar campos auto calculados (expresiones)*

Caso de Uso	Administrar campo tipo expresión
Actores	Usuario Registrado
Propósito	Definir un campo de tipo expresión autocalculado.
Precondición	Existencia de al menos un campo numérico.
Flujo	
Acción del Actor	Respuesta del Sistema
El caso de uso comienza cuando el actor autenticado que está dentro de una fuente de datos y hace clic en el botón nuevo campo.	Se muestra la ventana para crear el nuevo campo, se muestra una lista con los tipos de campos, así como las casillas para el nombre y descripción.

Llena los datos (nombre y descripción), selecciona el tipo expresión, y hace clic en el botón guardar.	Valida que los datos sean correctos. En caso de ser correctos los datos salva la información en la BD, muestra un mensaje informando que la operación se efectuó correctamente y muestra el nuevo campo en la lista de campos de la pestaña vista diseño y vista de datos. En caso contrario muestra un mensaje de error.
<p>Seleccionar la “Opción definir expresión” (ir al flujo alternativo Definir Campo Expresión)</p> <p>Seleccionar la “Opción modificar expresión” (ir al flujo alternativo Modificar Campo Expresión).</p>	
Flujo Alternativo: Definir campo expresión	
En la Vista de Datos hacer clic en el botón definir expresión.	Muestra la ventana del generador de expresiones donde el usuario podrá construir la expresión, en ella se muestra el listado de todos los campos válidos y los operadores que puede utilizar, así como una casilla de texto en la cual quedara construida la expresión.

Construye la expresión operando los campos que desea combinar y hace clic en aceptar.	Valida que la expresión construida sea correcta. En caso de ser correcta la expresión salva la información en la BD y muestra un mensaje informando que la operación se efectuó correctamente, si los campos que conforman la expresión tienen datos, calcula el valor del campo creado y actualiza la tabla de la aplicación con el valor. En caso contrario muestra un mensaje de error.
Flujo Alternativo: Modificar campo expresión	
En la Vista de Datos hace clic en el botón modificar expresión.	Muestra la ventana con el listado de todos los campos tipo expresión.
Se selecciona el campo que se quiere modificar (redefinir la expresión).	Muestra la ventana del generador de expresiones donde el usuario podrá construir la expresión, en ella se muestra el listado de todos los campos válidos y los operadores que puede utilizar, así como una casilla de texto en la cual quedara construida la expresión.

Construye la expresión operando los campos que desea combinar y hace clic en aceptar.	Valida que la expresión construida sea correcta. En caso de ser correcta la expresión salva la información en la BD y muestra un mensaje informando que la operación se efectuó correctamente, si los campos que conforman la expresión tienen datos, calcula el valor del campo creado y actualiza la tabla de la aplicación con el valor. En caso contrario muestra un mensaje de error.
Postcondiciones	Se adicionará o modificará un campo expresión.
Especificaciones Suplementarias	
Excepciones	
En caso de que existan problemas con la conexión a la BD, el sistema lanzara una excepción.	
Si a la hora de insertar un campo en la BD existe otro con el mismo nombre entonces se lanzará una excepción puesto que no pueden existir dos campos con el mismo nombre.	
Si se va a crear un campo tipo expresión y no existe ningún campo válido (tipo numérico), el sistema lanzará una excepción ya que no se puede definir una expresión.	

Anexo 7. *Descripción del caso de uso gestionar valores de los campos*

Nombre del caso de uso	Gestionar valores de los campos
Actores	Usuario Registrado (inicia).
Resumen	El caso de uso se inicia cuando el usuario registrado es autenticado y da clic derecho sobre una determinada fuente de datos seleccionando la opción Vista de Diseño. El sistema muestra la pestaña de la Vista de Diseño donde el usuario podrá gestionar los valores de los campos, también se muestran todos los valores, finalizando así el caso de uso.
Precondiciones	El usuario debe estar autenticado Tiene que estar almacenada la información de los valores de los campos, en caso de eliminar y/o modificar.

Post-condiciones	<p>Se actualiza la información de los Valores de los Campos.</p> <p><u>Si acción: insertar</u>, se inserta información de los datos: numéricos, textos, etc. El sistema valida que los datos sean correctos. En caso de ser correctos los datos se guardan en la BD y muestra un mensaje informando que la operación se efectuó correctamente. En caso contrario muestra un mensaje de error.</p> <p><u>Si acción: modificar</u>, varía la información de los valores de los campos.</p> <p><u>Si acción: eliminar</u>, El usuario debe seleccionar el valor del campo que desea eliminar, el sistema muestra un mensaje preguntando si el actor está seguro que desea eliminar el campo seleccionado, en caso de aceptar se elimina el campo seleccionado, en caso contrario se cancela la operación.</p>
Requisitos especiales	
Excepciones	
En caso de que existan problemas con la conexión a la BD, el sistema lanzará una excepción.	

Anexo 8. *Descripción del caso de uso gestionar mapas*

Nombre del caso de uso	Gestionar mapas
Actores	Usuario Registrado (inicia).
Resumen	El caso de uso se inicia cuando el usuario registrado es autenticado y da clic derecho sobre una determinada fuente de datos seleccionando la opción Vista de Mapas. El sistema muestra la pestaña de la Vista de Mapas donde el usuario podrá gestionar los mapas, también se muestra el listado de todos los campos para la fuente de datos seleccionada, finalizando así el caso de uso.
Precondiciones	El usuario debe estar autenticado Tiene que estar almacenada la información de los mapas, en caso de eliminar y/o modificar.

Post-condiciones	<p>Se actualiza la información de los mapas.</p> <p><u>Si acción: insertar</u>, se inserta información de los datos: nombre del mapa, tipo, descripción. El sistema valida que los datos sean correctos. En caso de ser correctos los datos salva la información en la BD y muestra un mensaje informando que la operación se efectuó correctamente. En caso contrario muestra un mensaje de error.</p> <p><u>Si acción: modificar</u>, El sistema muestra una ventana con los datos de los mapas modificables. El usuario modifica los datos del mapa y hace clic en el botón guardar.</p> <p><u>Si acción: eliminar</u>, El usuario debe seleccionar el mapa que desea eliminar, el sistema muestra un mensaje preguntado si el actor está seguro que desea eliminar el mapa seleccionado, en caso de aceptar se elimina el campo seleccionado, en caso contrario se cancela la operación.</p>
Requisitos especiales	
Excepciones	
En caso de que existan problemas con la conexión a la BD, el sistema lanzara una excepción.	
Si a la hora de insertar un mapa en la BD existe otro con el mismo nombre entonces se lanzará una excepción puesto que no pueden existir dos Mapas con el mismo nombre.	

Anexo 9. Descripción del caso de uso importar cartografía

Caso de Uso	Importar Cartografía
Actores	Usuario Registrado
Propósito	Importar cartografía
Precondición	El usuario debe estar autenticado
Flujo Básico	
Acción del Actor	Respuesta del Sistema
El caso de uso comienza cuando el actor es autenticado y da clic en la opción Importar cartografía.	Muestra una ventana con la opción de seleccionar los archivos de la cartografía a importar.
Selecciona el camino donde se encuentran los ficheros shp, dbf, etc.	Muestra un listado de los archivos de la cartografía los cuales se van a importar, además muestra la opción de poder eliminar alguno si es necesario.
Se selecciona todos los ficheros y se presiona el botón Upload.	
Postcondiciones	Se importará la cartografía seleccionada.
Especificaciones Suplementarias	
Excepciones	
En caso de que existan problemas con la conexión a la BD, el sistema lanzara una excepción.	
Si a la hora de insertar una cartografía en la BD y existe otra con el mismo nombre entonces se lanzará una excepción puesto que no pueden existir dos con el mismo nombre.	
Notas	

Anexo 10. *Descripción del caso de uso importar la información georeferenciada*

Nombre del caso de uso	Importar la información georeferenciada.
Actores	Usuario Registrado (inicia).
Resumen	El caso de uso se inicia cuando el usuario registrado decide Importar la información desde diferentes formatos de intercambio (xls,pdf,xml). El sistema muestra el asistente que lo guiará paso a paso para importar la información, si no se pudo realizar muestra un mensaje de error indicando el problema, sino muestra un mensaje de éxito, finalizando así el caso de uso.
Precondiciones	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
El caso de uso comienza cuando el actor es autenticado y da clic en la opción Importar Información.	Muestra una ventana con la opción de seleccionar los campos a importar.
Selecciona el camino donde se encuentran los ficheros.	Muestra un listado de los campos temáticos los cuales se van a importar, además muestra la opción de poder eliminar alguno si es necesario.
Postcondiciones	
Requisitos especiales	

Excepciones
En caso de que existan problemas con la conexión a la BD, el sistema lanzará una excepción.
Si a la hora de insertar una información en la BD y existe otra con el mismo nombre entonces se lanzará una excepción puesto que no pueden existir dos con el mismo nombre.

Anexo 11. *Descripción del caso de uso exportar la información georeferenciada*

Nombre del caso de uso	Exportar la información georeferenciada.
Actores	Usuario Registrado (inicia).
Resumen	El caso de uso se inicia cuando el Usuario Registrado decide Exportar la información desde diferentes formatos de intercambio (xls,pdf,xml). Si no se puede realizarla exportación muestra un mensaje de error indicando el problema, sino muestra un mensaje de éxito, finalizando así el caso de uso.
Precondiciones	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
El caso de uso comienza cuando el actor es autenticado y da clic en la opción Exportar Información.	Muestra una ventana con la opción de seleccionar los campos a Exportar.

Selecciona el camino donde se desea guardar la información temática.	Muestra un mensaje indicando que la exportación se realizó con éxito o si falló.
Postcondiciones	
Requisitos especiales	
Excepciones	
En caso de que existan problemas con la conexión a la BD, el sistema lanzará una excepción.	

Anexo 12. *Descripción del caso de uso realizar búsqueda*

Nombre del caso de uso	Realizar búsqueda
Actores	Usuario Anónimo (inicia).
Resumen	El caso de uso se inicia cuando el usuario anónimo decide realizar una búsqueda sobre la información publicada. El sistema muestra la vista del servicio de Búsqueda. El usuario teclea el patrón de búsqueda y da clic en el botón buscar, el sistema muestra el resultado del criterio de búsqueda encontrado, si la búsqueda no reporta ningún resultado muestra un mensaje indicando que no se encontraron resultados, finalizando así el caso de uso.
Precondiciones	
Post-condiciones	Se mostrará el listado de la búsqueda encontrada.
Requisitos especiales	
Excepciones	
En caso de que existan problemas con la conexión a la BD, el sistema lanzara una excepción.	

Anexo 13. *Descripción del caso de uso visualizar los datos de los usuarios recuperados*

Nombre del caso de uso	Visualizar los datos de los usuarios recuperados.
Actores	Usuario Anónimo (inicia).
Resumen	El caso de uso comienza cuando el usuario realiza una búsqueda sobre la información publicada. El sistema muestra la vista del resultado de la Búsqueda, el usuario da clic sobre el enlace del resultado devuelto y sistema muestra la vista de visualización de los datos recuperados, también permite exportar los datos recuperados, finalizando así el caso de uso.
Precondiciones	Tienen que estar almacenados y actualizados los datos sobre los Usuarios Recuperados.
Post-condiciones	Se mostrará la vista de visualización de los datos recuperados.
Requisitos especiales	
Excepciones	
En caso de que existan problemas con la conexión a la BD, el sistema lanzara una excepción.	

Anexo 14. *Descripción del caso de uso visualizar mapas de los usuarios recuperados*

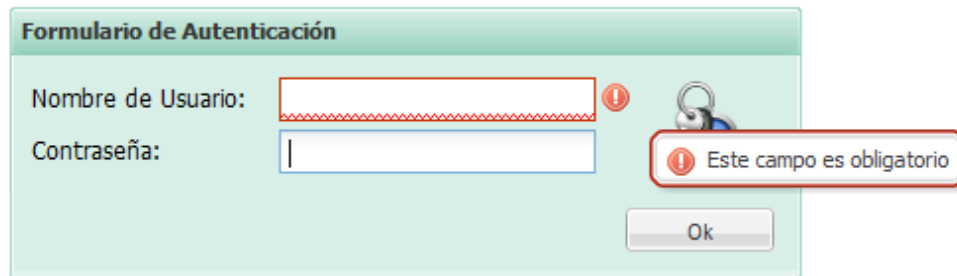
Nombre del caso de uso	Visualizar los Mapas de los usuarios recuperados.
Actores	Usuario anónimo(inicia).
Resumen	El caso de uso comienza cuando el usuario realiza una búsqueda sobre la información publicada. El sistema muestra la vista del resultado de la Búsqueda, el usuario da clic sobre el enlace del resultado devuelto y sistema muestra la vista de visualización de los datos recuperados también permite exportar los mapas recuperados de los usuarios, finalizando así el caso de uso.
Precondiciones	Tienen que estar almacenados y actualizados los datos sobre los mapas de los usuarios recuperados.
Post-condiciones	Se Mostrará la vista de visualización de los mapas recuperados.
Requisitos especiales	
Excepciones	
En caso de que existan problemas con la conexión a la BD, el sistema lanzara una excepción.	

Anexo 15. *Descripción del caso de uso gestionar registro de usuario*

Nombre del caso de uso	Gestionar registro de usuario
Actores	Usuario registrado (inicia).
Resumen	El caso de uso comienza cuando el actor desea registrar a un nuevo usuario al sistema y este muestra la pantalla de registro, donde se captan los datos del nuevo usuario a registrarse. El actor entra la información de registro: nombre de usuario, contraseña, dirección electrónica. El sistema valida que los datos sean correctos. Registra al usuario en la BD y muestra un mensaje informando que la operación se efectuó correctamente. En caso contrario muestra un mensaje de error.
Precondiciones	El usuario no debe estar registrado antes
Post-condiciones	Se registrará un nuevo usuario.
Requisitos especiales	
Excepciones	
En caso de que existan problemas con la conexión a la BD, el sistema lanzara una excepción.	
Si a la hora de registrar el usuario ya existe otro usuario con el mismo nombre de usuario en la BD entonces se lanzará una excepción puesto que no pueden existir dos nombres de usuarios repetidos.	

Anexo 16. *Descripción del caso de uso gestionar perfiles de usuarios*

Nombre del caso de uso	Gestionar perfil de usuario
Actores	usuario registrado (inicia).
Resumen	El caso de uso comienza cuando el actor desea modificar su perfil se dirige al sistema y este muestra la pantalla de modificación de perfil. El actor entra la información del perfil y el sistema valida que los datos sean correctos. Actualiza los datos en la BD y muestra un mensaje informando que la operación se efectuó correctamente. En caso contrario muestra un mensaje de error.
Precondiciones	El usuario debe estar autenticado
Post-condiciones	-
Requisitos especiales	
Excepciones	
En caso de que existan problemas con la conexión a la BD, el sistema lanzara una excepción.	

Anexo 17. *Prototipo Autenticar usuarios*

Formulario de Autenticación

Nombre de Usuario:

Contraseña:

Este campo es obligatorio

Ok

Anexo 18. *Prototipo Gestionar espacios de datos*

Imagen 18.1 Panel Izquierdo en forma de árbol lista los espacios de datos del usuario.

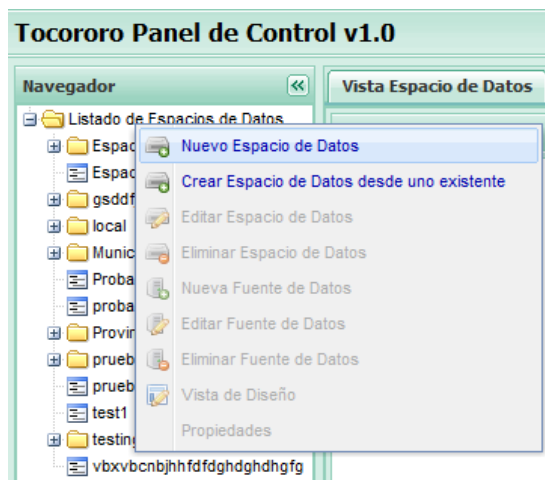


Imagen 18.2 Gestionar un espacio de datos.

Adicionar Nuevo Espacio de Datos

Configuración del Servidor de Mapas

☐ Usar el Servidor de Mapas de Tocatoro

☒ Url del Servidor:

Nombre del Espacio de Datos:

Capa:

Campo GeoId:

Campo Nombre:

Descripción:

Etiquetas(No obligatorio):

Filtro(No obligatorio):

☒ Publico

Imagen 18.3 Adicionar espacios de datos.

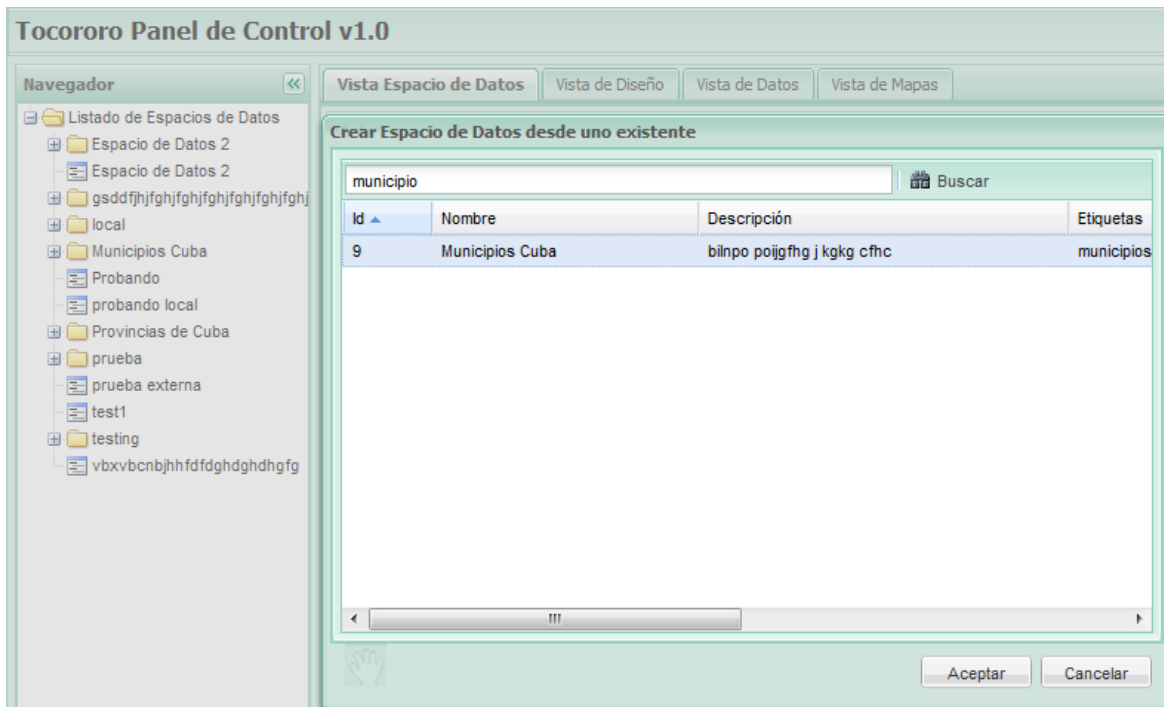


Imagen 18.4 Panel derecho de un espacio de datos.



Imagen 18.5 Propiedades de un espacio de datos.

Anexo 19. Prototipo fuentes de Datos



Imagen 19.1 Gestionar fuentes de datos

The image shows a dialog box titled 'Adicionar Nueva Fuente de Datos'. It contains the following fields and controls:

- 'Nombre de la F.D:': A text input field with a red dashed border and a red exclamation mark icon to its right.
- 'Nombre para Mostrar:': A text input field with a red dashed border and a red exclamation mark icon to its right.
- 'Etiquetas(No obligatorio):': A text input field.
- A checkbox labeled 'Publico' which is checked.
- 'Guardar' and 'Cancelar' buttons at the bottom right.

Imagen 19.2 Adicionar fuentes de datos

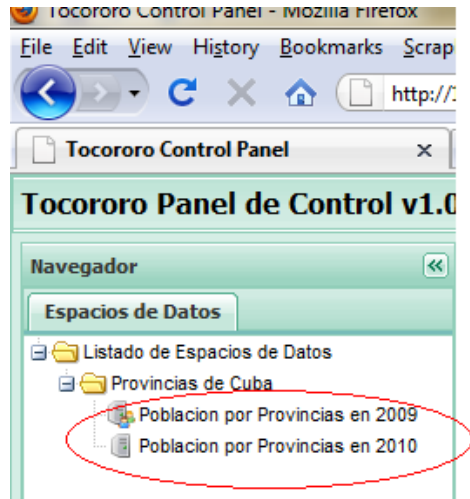


Imagen 19.3 Listado de fuentes de datos creadas.

Anexo 20. Prototipo gestionar campos

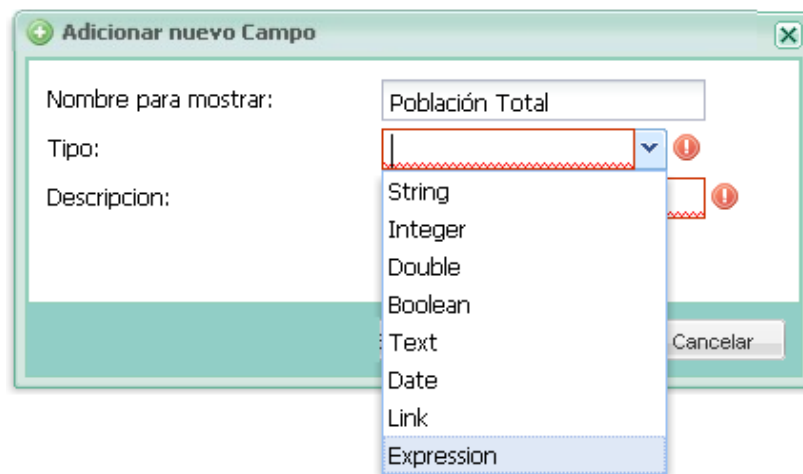


Imagen 20.1 Adicionar nuevo campo



Imagen 20.2 Listado de campos (eliminar, adicionar, modificar y guardar)

Anexo 21. Prototipo gestionar campos auto calculado (expresiones)

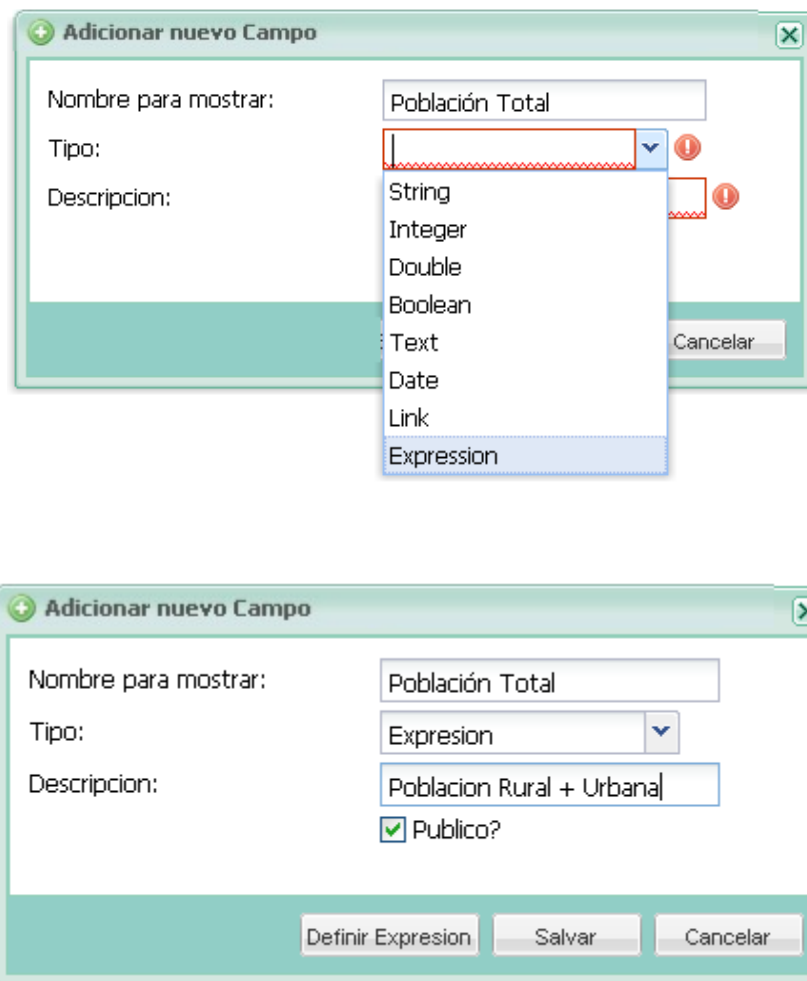



Imagen 21.1 Adicionar campo tipo *expresión*

Selección de campo

Texto a Buscar:  Buscar




Filtrar por:

☐ Campo Local ?

Nombre	Origen	Descripción
Poblacion Rural	rebecap.Provincias de Cuba.Poblacion por Provincias en 2010.Poblacion Rural	Poblacion Rural
Poblacion Urbana	rebecap.Provincias de Cuba.Poblacion por Provincias en 2010.Poblacion Urbana	Poblacion Urbana
Poblacion Total 2010	rebecap.Provincias de Cuba.Poblacion por Provincias en 2010.Poblacion Total 2010	Poblacion Total por Provincias en Cuba 20
Poblacion Total	rebecap.Provincias de Cuba.Poblacion por Provincias en 2009.Poblacion Total	Total Habitantes
Población Urbana	rebecap.Provincias de Cuba.Poblacion por Provincias en 2009.Población Urbana	Habitantes en zona urbana
Población Rural	rebecap.Provincias de Cuba.Poblacion por Provincias en 2009.Población Rural	Habitantes en zona rural

Imagen 21.2 Vista para definir el campo autocalculado.

Descripción del campo expresión: Población Total

ID	Origen	Local	
A	rebecap.Provincias de Cuba.Poblacion por Provincias en 2010.Poblacion Rural	true	
B	rebecap.Provincias de Cuba.Poblacion por Provincias en 2010.Poblacion Urbana	true	
C			


Formula: 

Imagen 21.3 Vista para adicionar los campos que van a conformar la expresión.

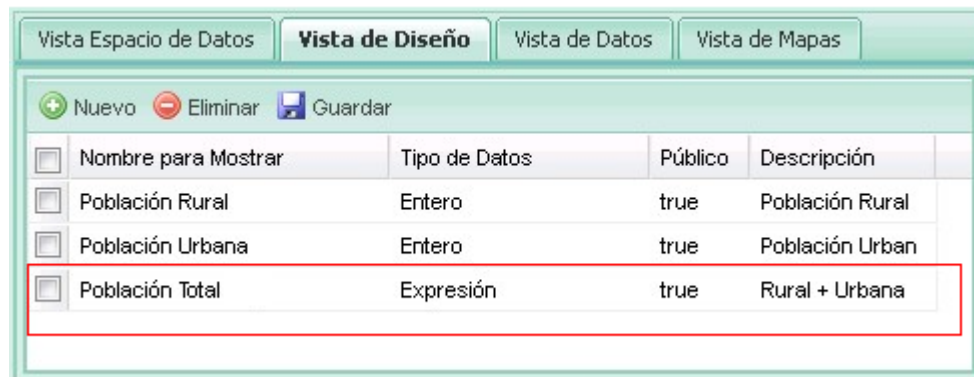


Imagen 21.4 Vista de diseño de un campo tipo Expresión

Anexo 22. Prototipo valores de los campos

id	Geo Id	Geo Nombre	Población Total	Población Urbana	Población Rural	Exp1	Total
1	6	Cienfuegos	405545.00	328650.00	76895.00	3.2865E7	405645.0
2	15	Municipio Especi...	86256.00	72289.00	13967.00	7228900.0	86356.0
3	7	Sancti Spiritus	465542.00	326216.00	139326.00	3.26216E7	465642.0
4	8	Ciego de Avila	422643.00	302870.00	119773.00	3.0287E7	422743.0
5	12	Granma	835808.00	491417.00	344391.00	4.91417E7	835908.0
6	11	Holguín	1037326.00	665588.00	371738.00	6.65588E7	1037426.0
8	14	Guantánamo	510944.00	309249.00	201695.00	3.09249E7	511044.0
9	9	Camagüey	782582.00	594967.00	187615.00	5.94967E7	782682.0
10	10	Las Tunas	536112.00	333584.00	202528.00	3.33584E7	536212.0
11	3	Ciudad de La Ha...	2141993.00	2141993.00	0	2.141993E8	2142093.0
12	2	La Habana	747491.00	542758.00	204733.00	5.42758E7	747591.0
13	5	Villa Clara	803690.00	609247.00	194443.00	6.09247E7	803790.0
14	1	Pinar del Río	729292.00	460470.00	268822.00	4.6047E7	729392.0
15	4	Matanzas	690223.00	567515.00	122708.00	5.67515E7	690323.0
16	13	Santiago de Cuba	1047181.00	727349.00	319832.00	7.27349E7	1047281.0

Imagen 22.1 Listado de diferentes valores de campos

Anexo 23. Prototipo gestionar mapas

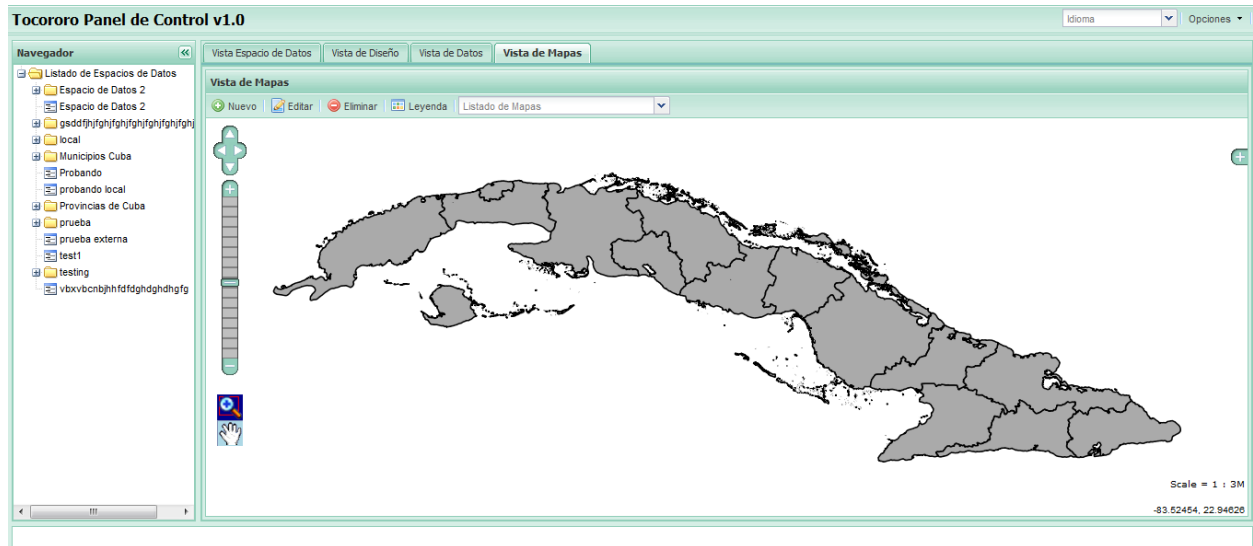


Imagen 23.1 Diseño inicial de un mapa

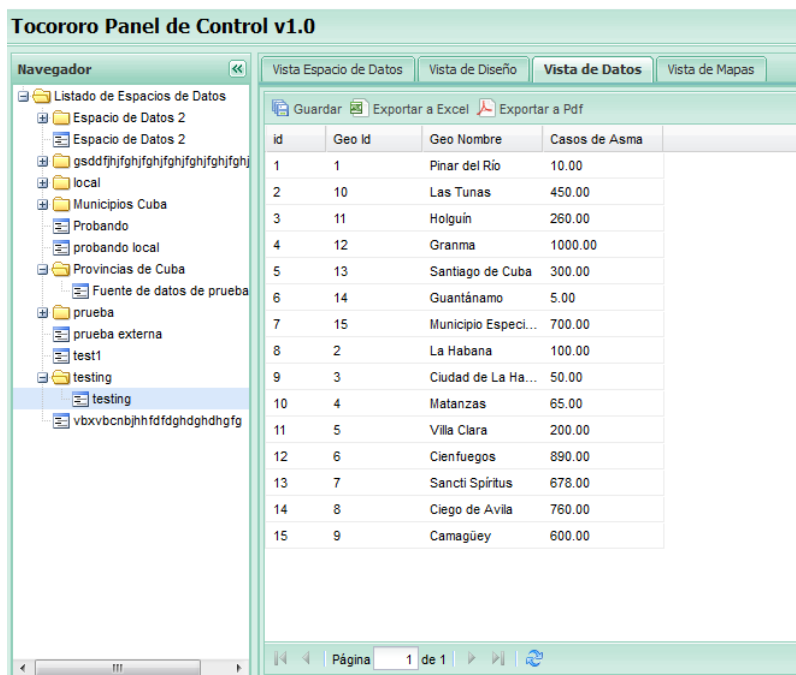


Imagen 23. 1 Vista de datos, donde el usuario introduce los datos asociados a la fuente de datos creada.

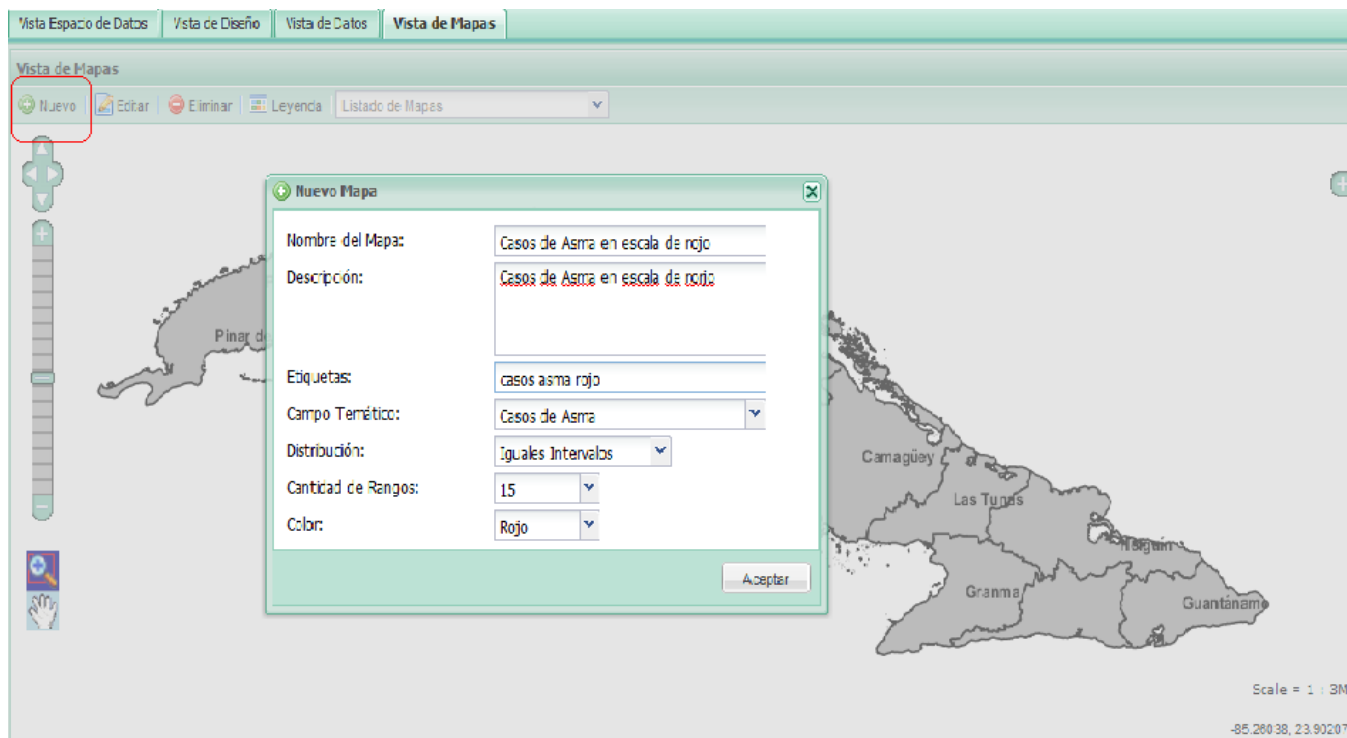


Imagen 23.2 Crear mapas temáticos.

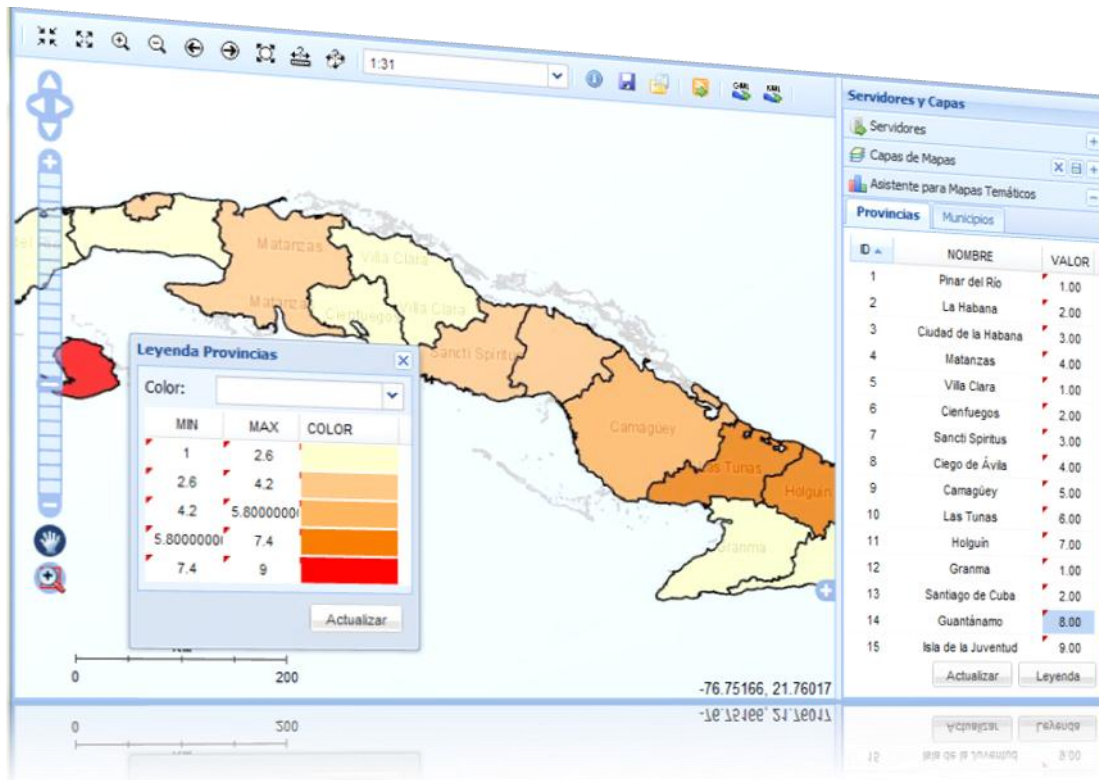


Imagen 23.3 Resultado de un mapa temático ya creado.

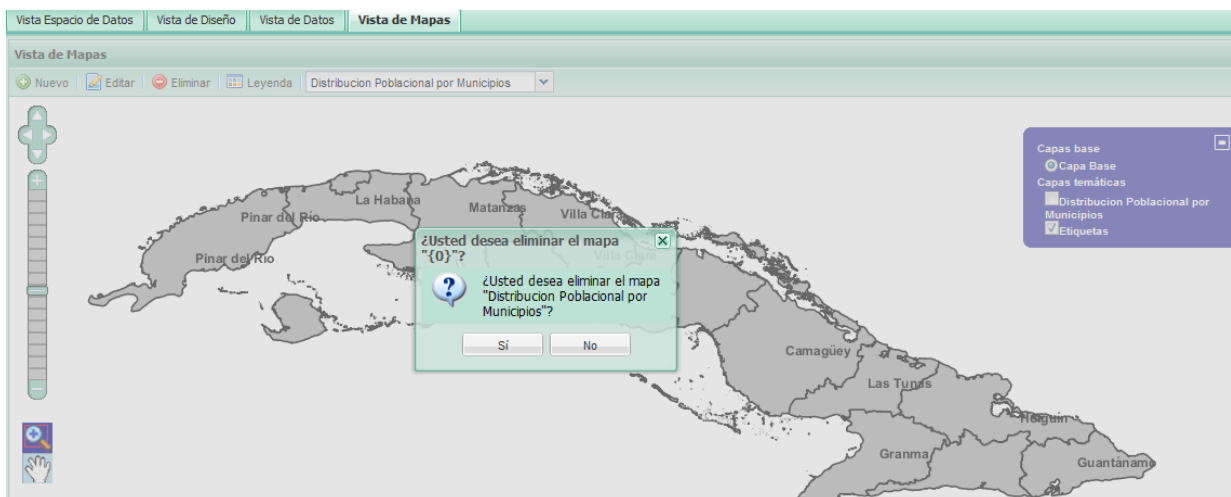


Imagen 23.4 Eliminar un mapa temático.

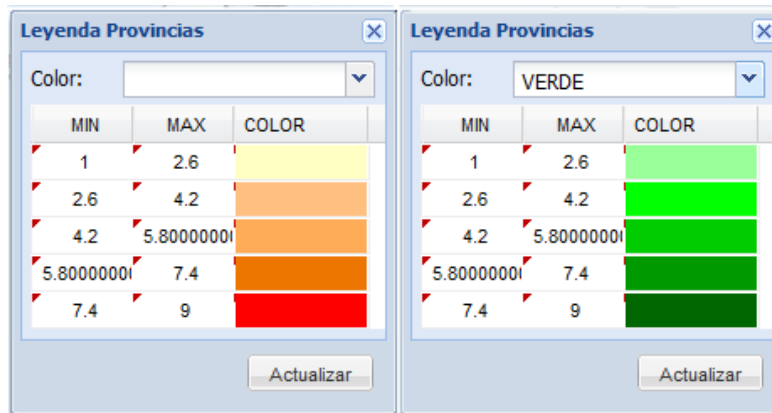
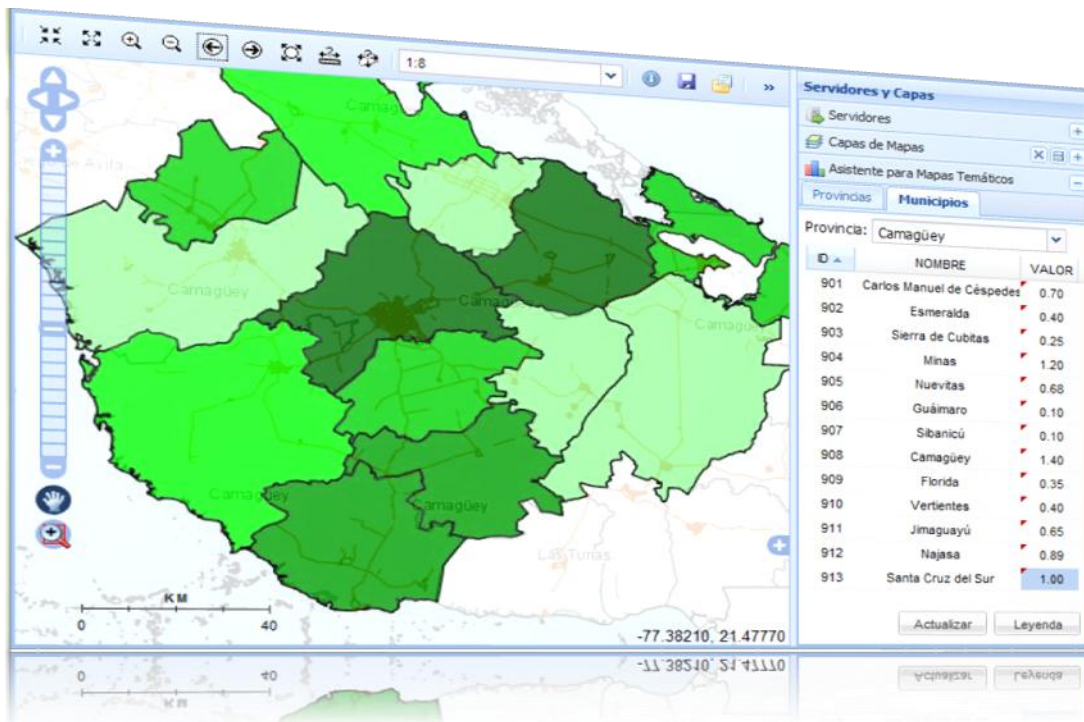
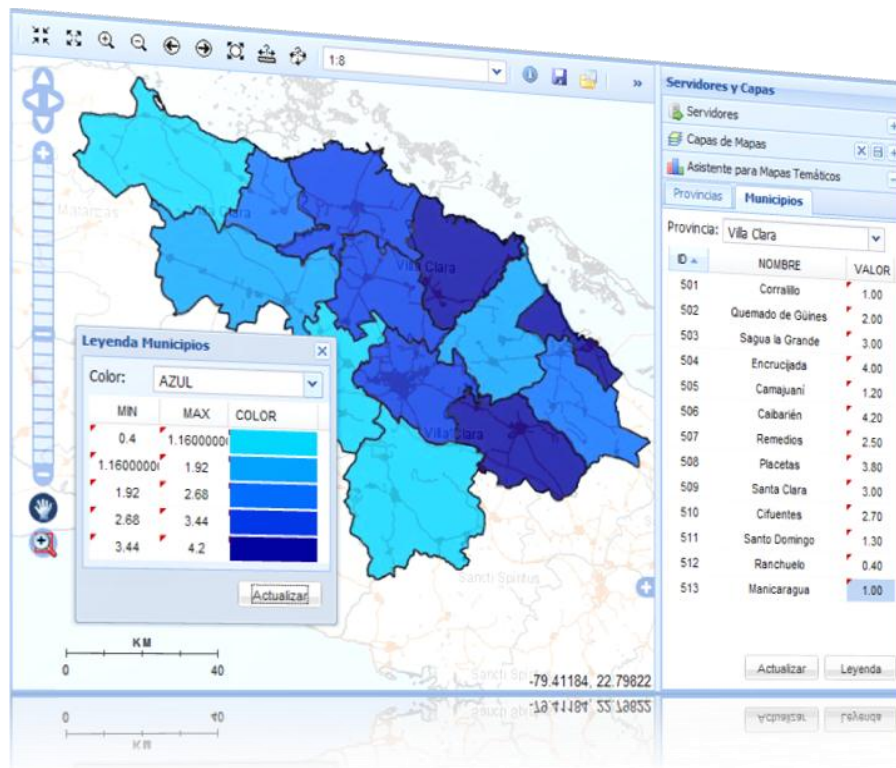
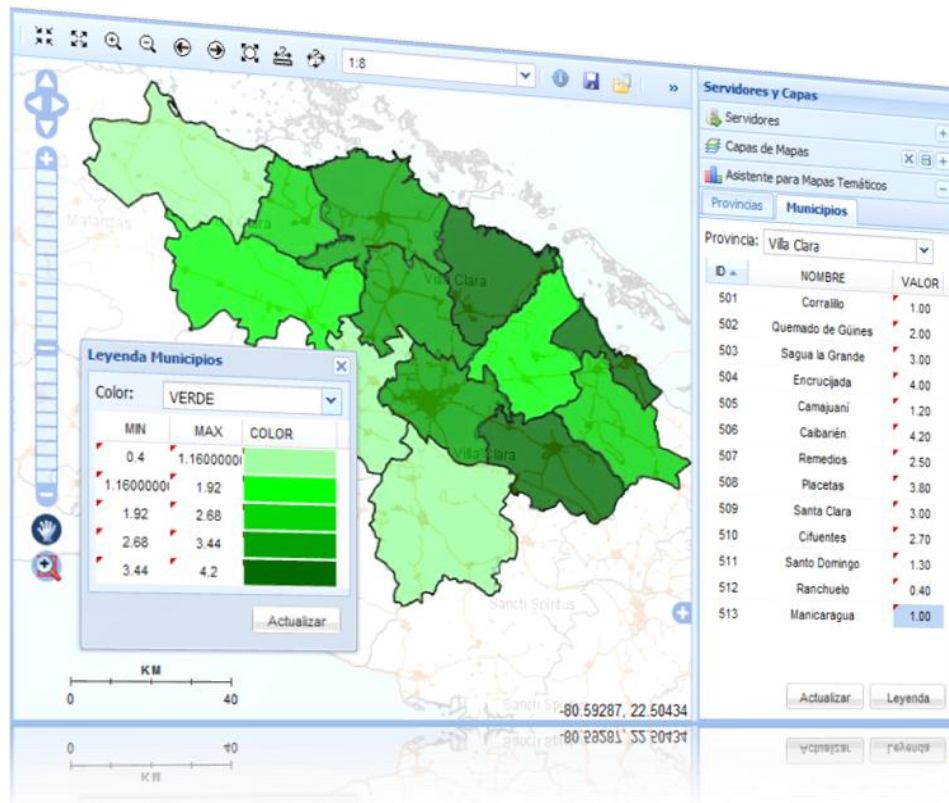
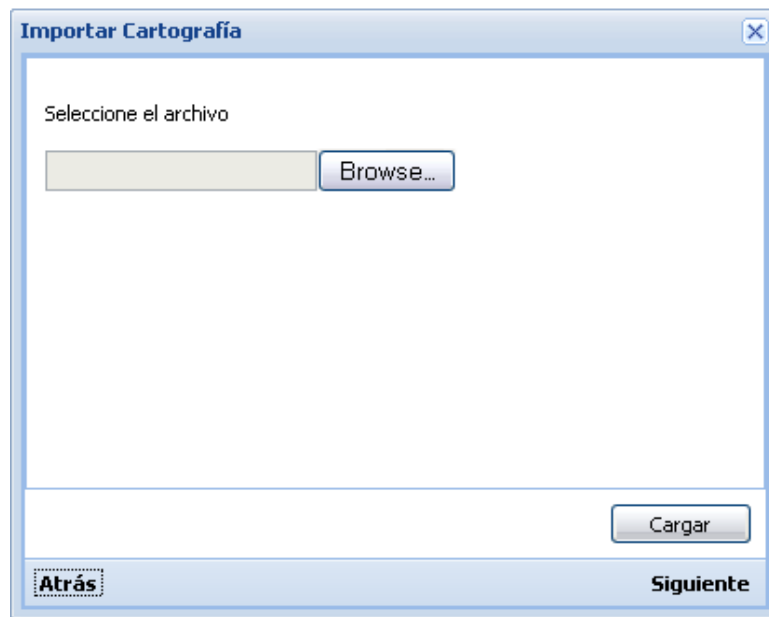


Imagen 23.5 Leyenda de un mapa.

Anexo 24. Prototipo Ejemplos de mapas temáticos





Anexo 25. Prototipo Importar cartografías**Imagen 25.1 Cargar Cartografía.****Imagen 25.2 Seleccionar campos.**

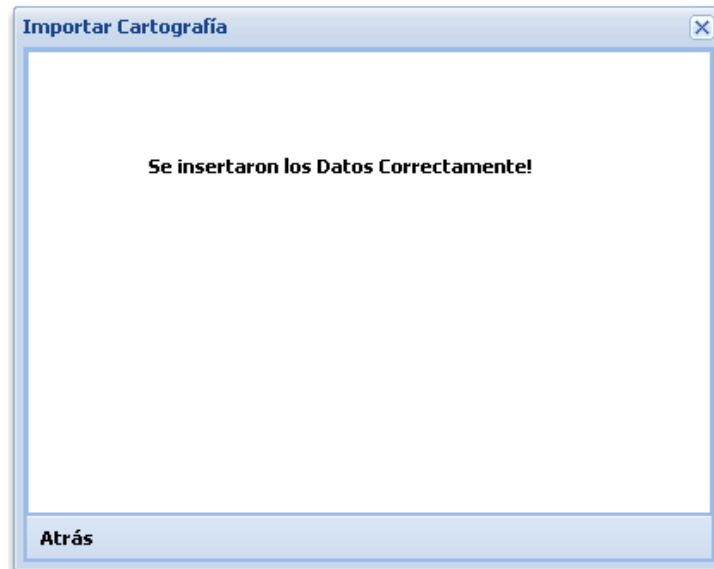
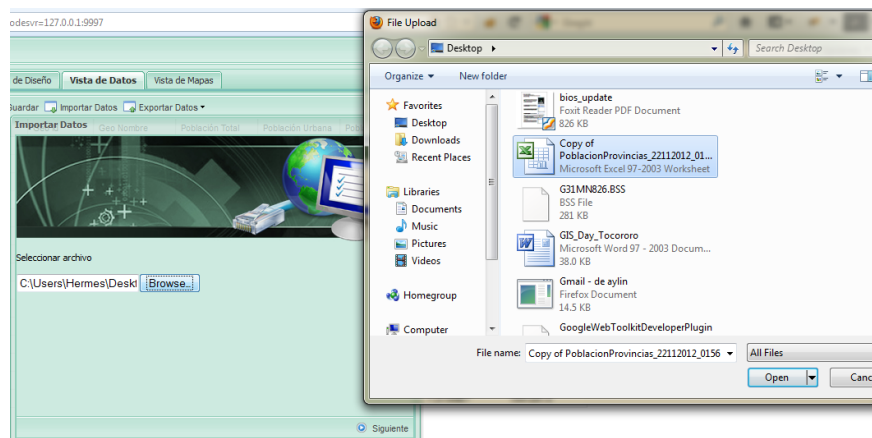


Imagen 25.3 Finalizar.

Anexo 26. Prototipo Importar la información georeferenciada publicada por los usuario



Anexo 27. Prototipo Exportar la información georeferenciada publicada por los usuario

Tocororo Panel de Control v1.0

Navegador: Espacios de Datos | Grupos de Usuarios

Lista de Espacios de Datos: Provincias de Cuba

Exportar Datos:

- Exportar a Pdf
- Exportar a Excel
- Exportar Plantilla Excel

id	Geo Id	Geo	Población Urb
1	6	Cienfuegos	328650.00
2	15	Municipio	72289.00
3	7	Sancr Spiritus	465542.00
4	8	Ciego de Avila	422643.00
5	12	Granma	635808.00
6	11	Holguin	1037326.00
8	14	Guantánamo	510944.00
9	9	Camagüey	782582.00
10	10	Las Tunas	536112.00
11	3	Ciudad de La Ha...	2141993.00
12	2	La Habana	747491.00
13	5	Villa Clara	803690.00
14	1	Pinar del Río	729292.00
15	4	Matanzas	690223.00
16	13	Santiago de Cuba	1047181.00

Anexo 28. Prototipo Realizar Búsquedas



Imagen 28.1 Realizar búsquedas en el buscador de Tocororo.

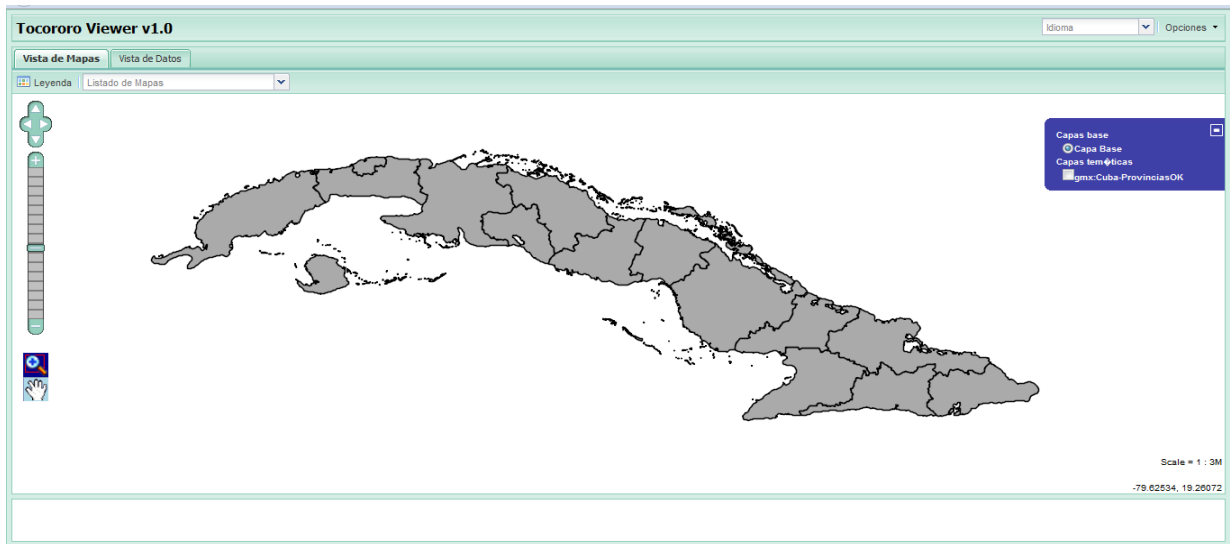


Imagen 28.2 Resultados de las búsquedas.

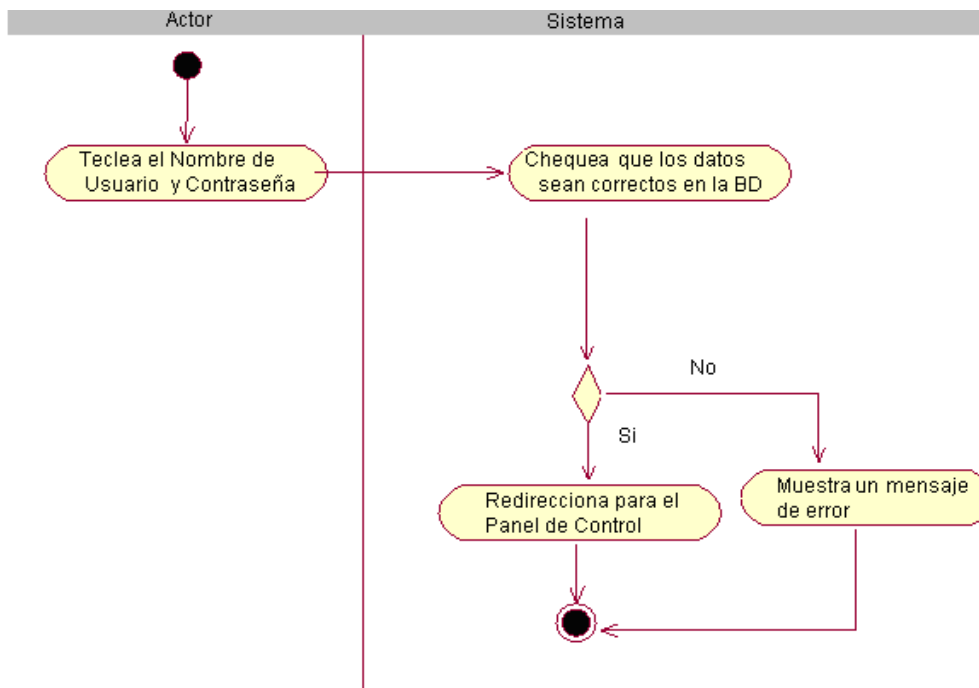
Anexo 29. *Prototipo visualizar los datos de los usuarios recuperados*

Tocatoro Viewer v1.0			
Vista de Mapas		Vista de Datos	
id	Geo Id	Geo Nombre	Enfermedades
1	11	Holguín	3234.00
2	12	Granma	5646.00
3	13	Santiago de Cuba	5675.00
4	14	Guantánamo	76776.00
5	15	Municipio Especi...	67865.00

Anexo 30. Prototipo visualizar los mapas de los usuarios recuperados

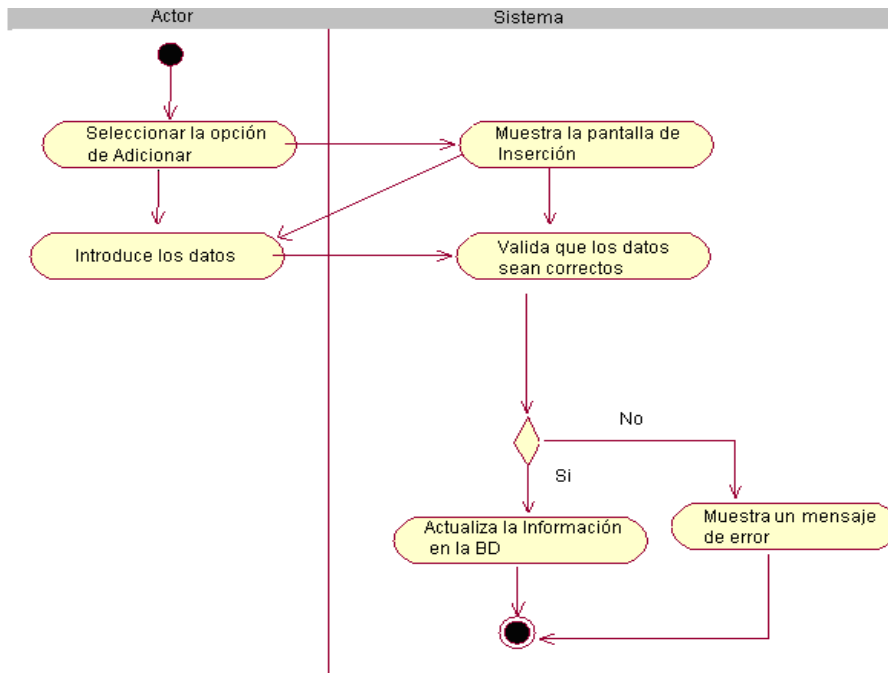


Anexo 31. Diagramas de Actividades (Flujo Alternativo Autenticar usuarios)

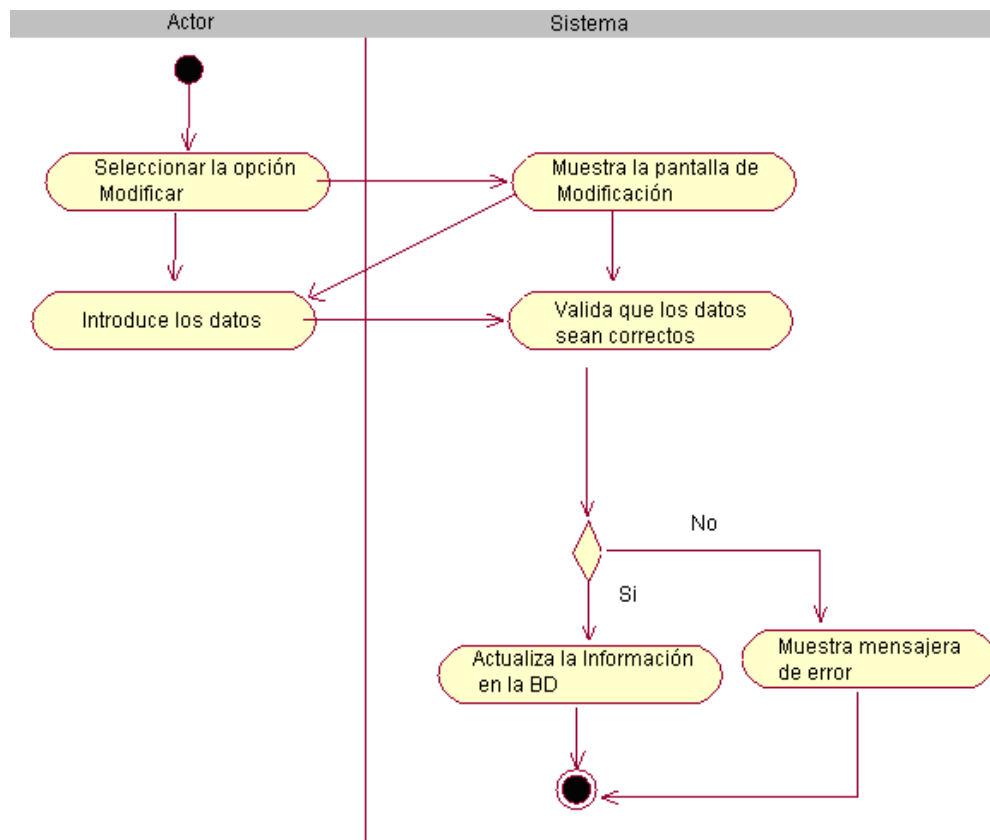


Anexo 32. Diagramas de Actividades (Flujo Alternativo: Gestionar espacio de datos)

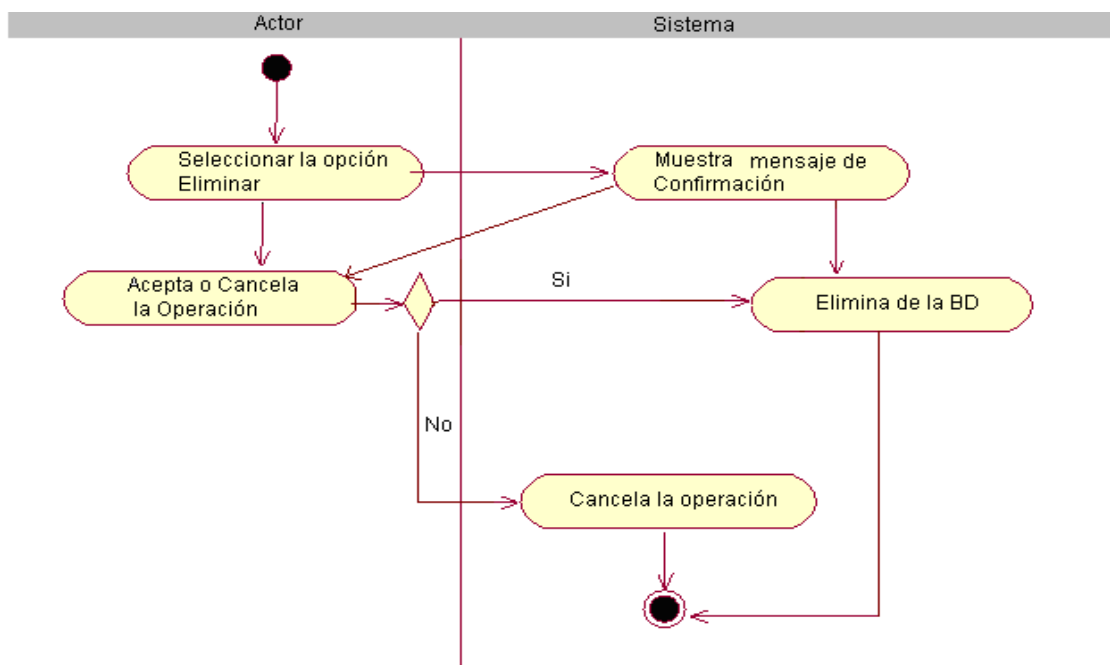
Adicionar espacio de datos.



Modificar espacio de datos

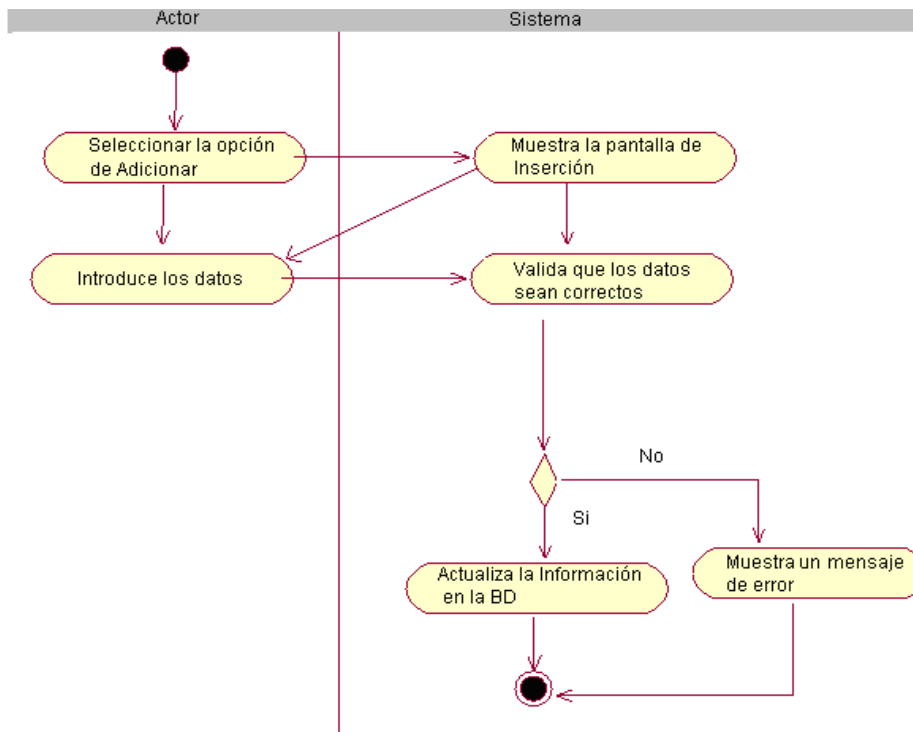


Eliminar espacio de datos

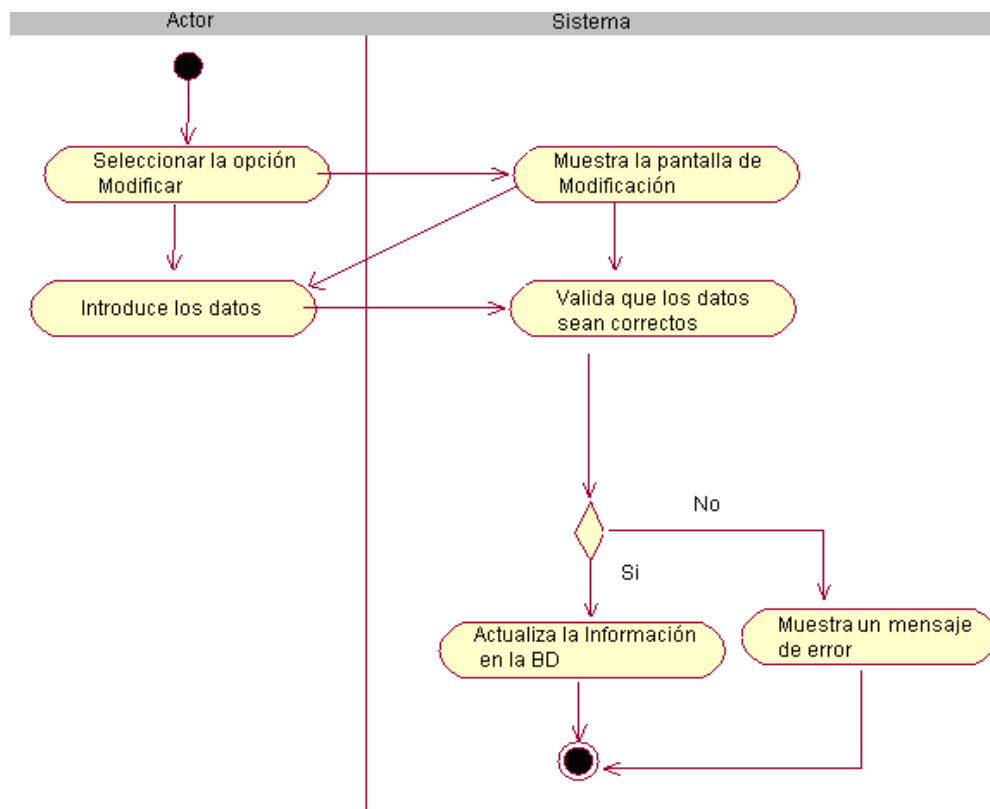


Anexo 33. Diagramas de Actividades (Flujo Alternativo: Gestionar fuente de datos)

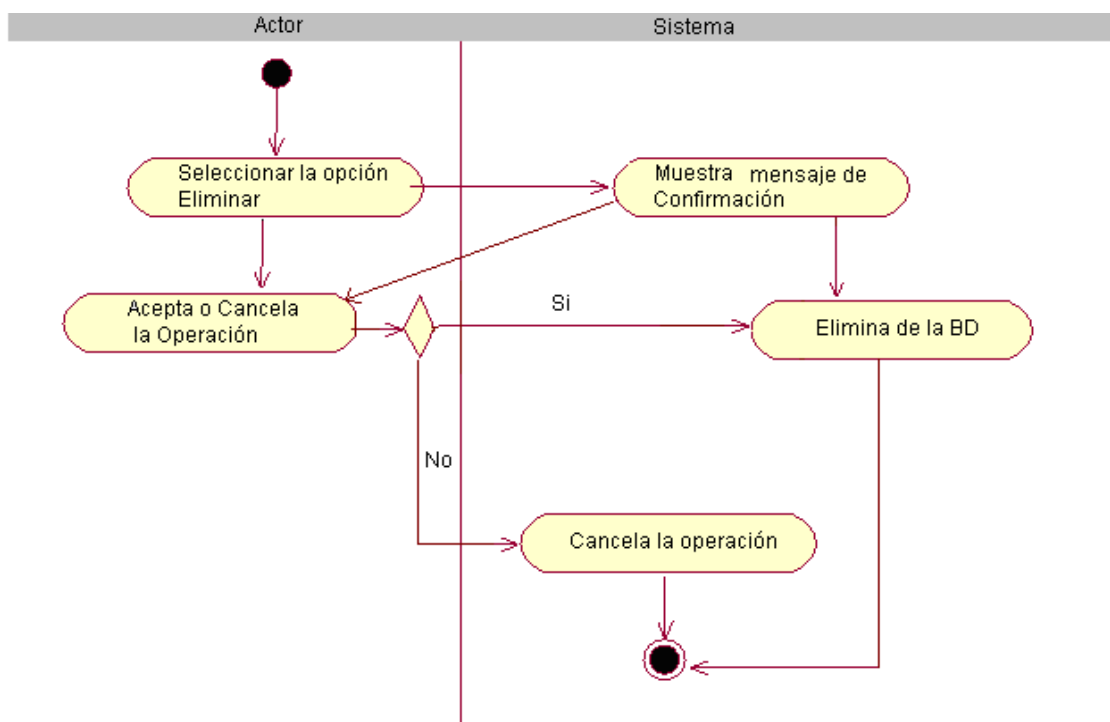
Adicionar fuente de datos



Modificar fuente de datos

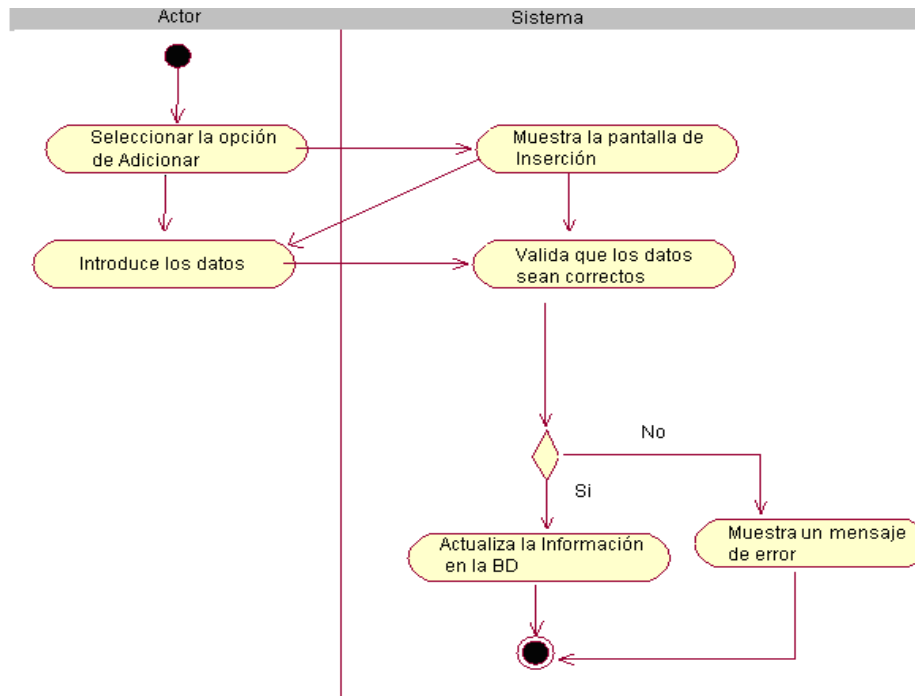


Eliminar fuente de datos

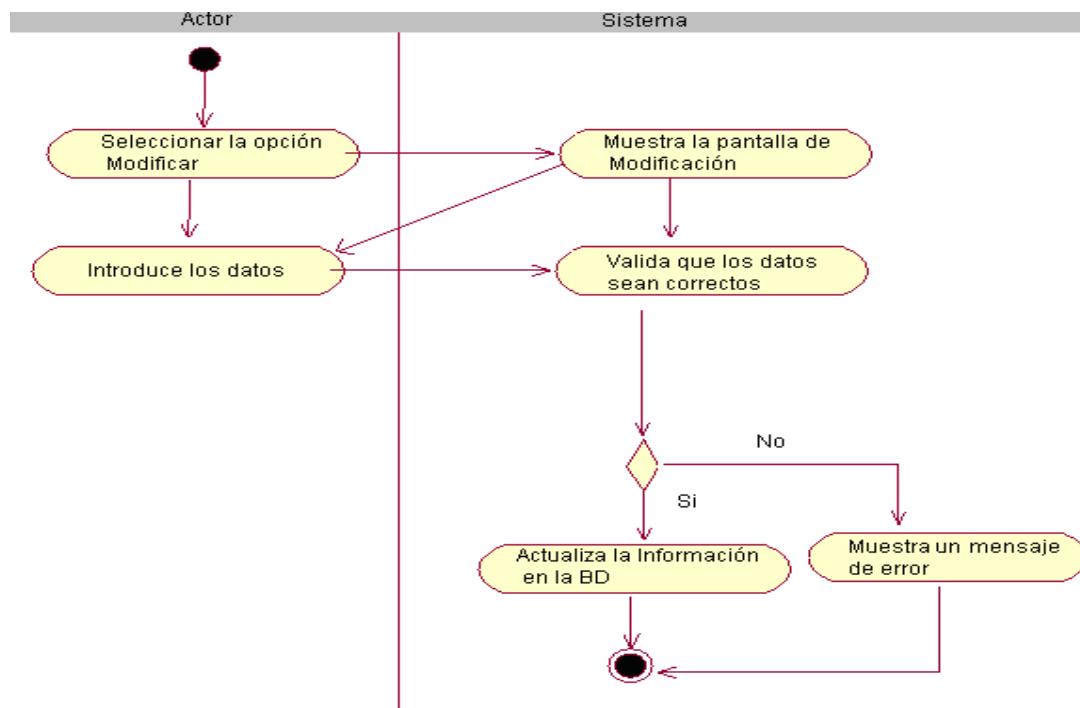


Anexo 34. *Diagrama de Actividades (Gestionar Campo)*

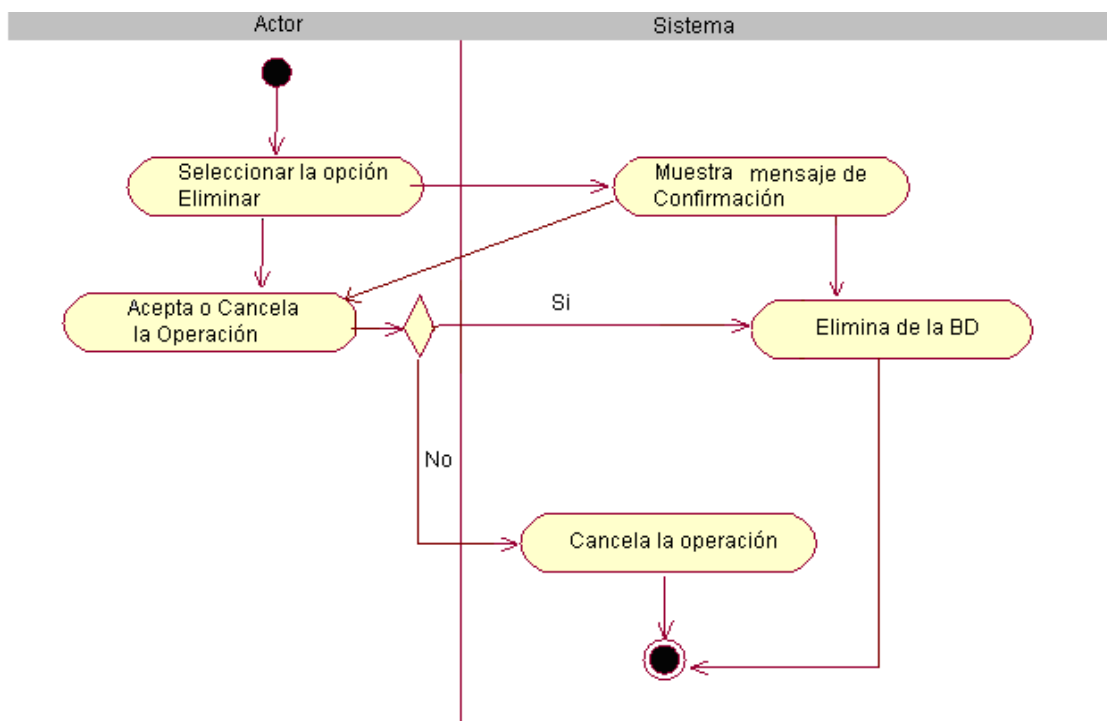
Adicionar campo.



Modificar campo

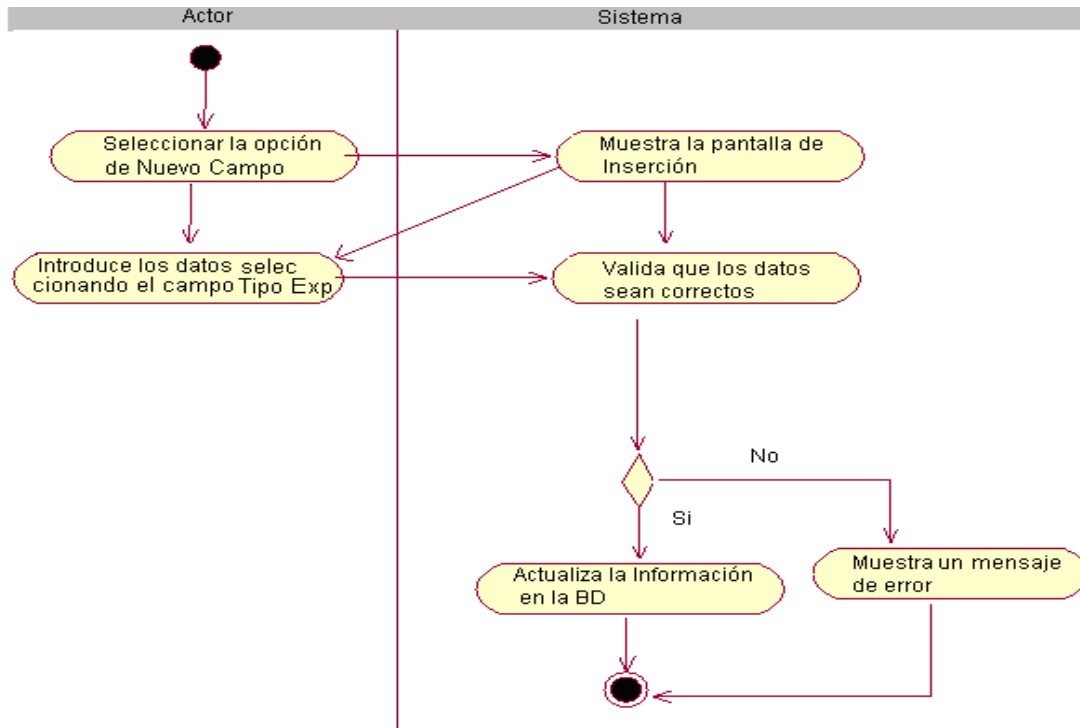


Eliminar campo

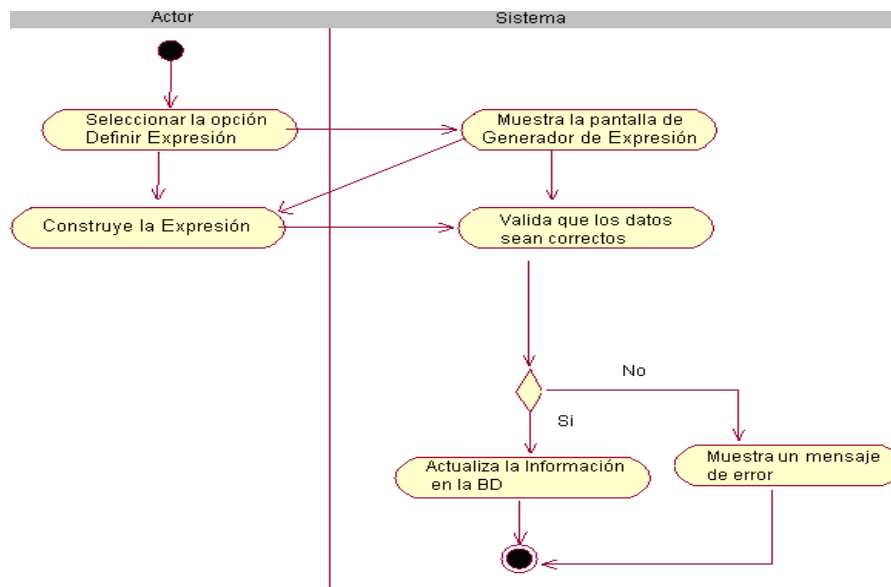


Anexo 35. Diagramas de Actividades (Gestionar campos auto calculado, expresiones))

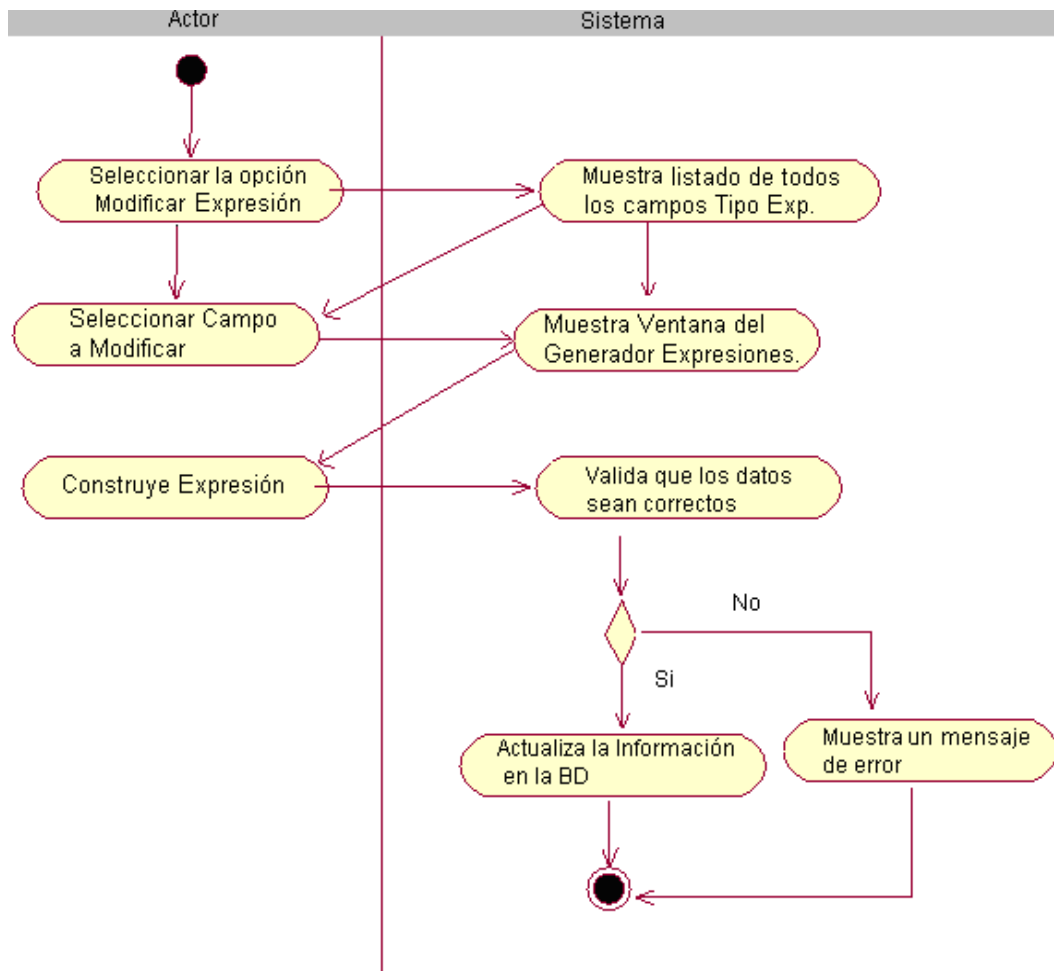
Adicionar campo auto calculado.



Definir campo autocalculado.

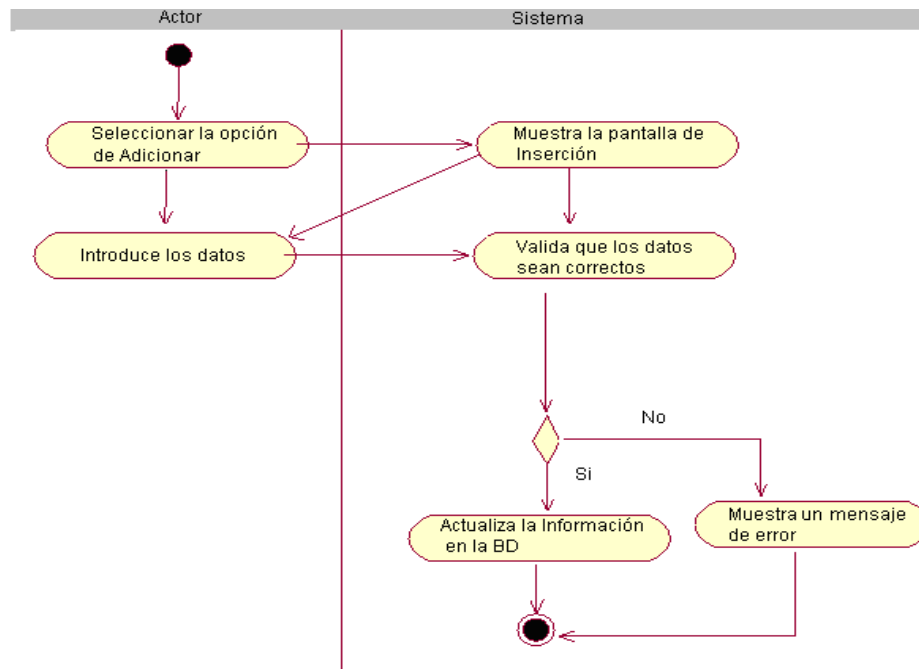


Modificar campo autocalculado.

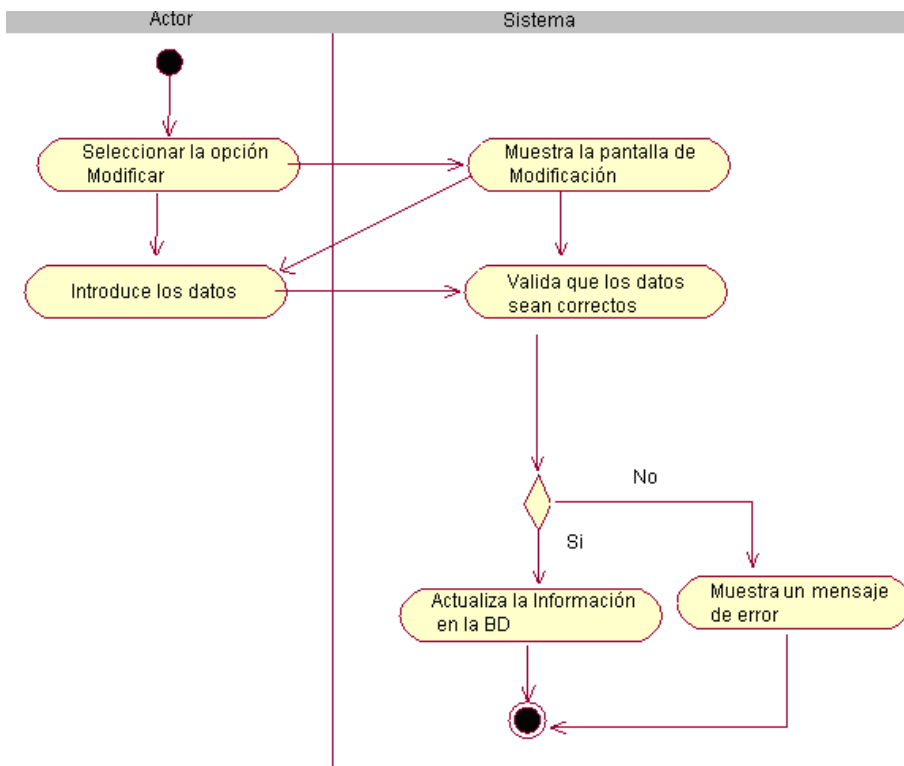


Anexo 36. Diagramas de Actividades (Flujo Alternativo: Gestionar valores de los campos)

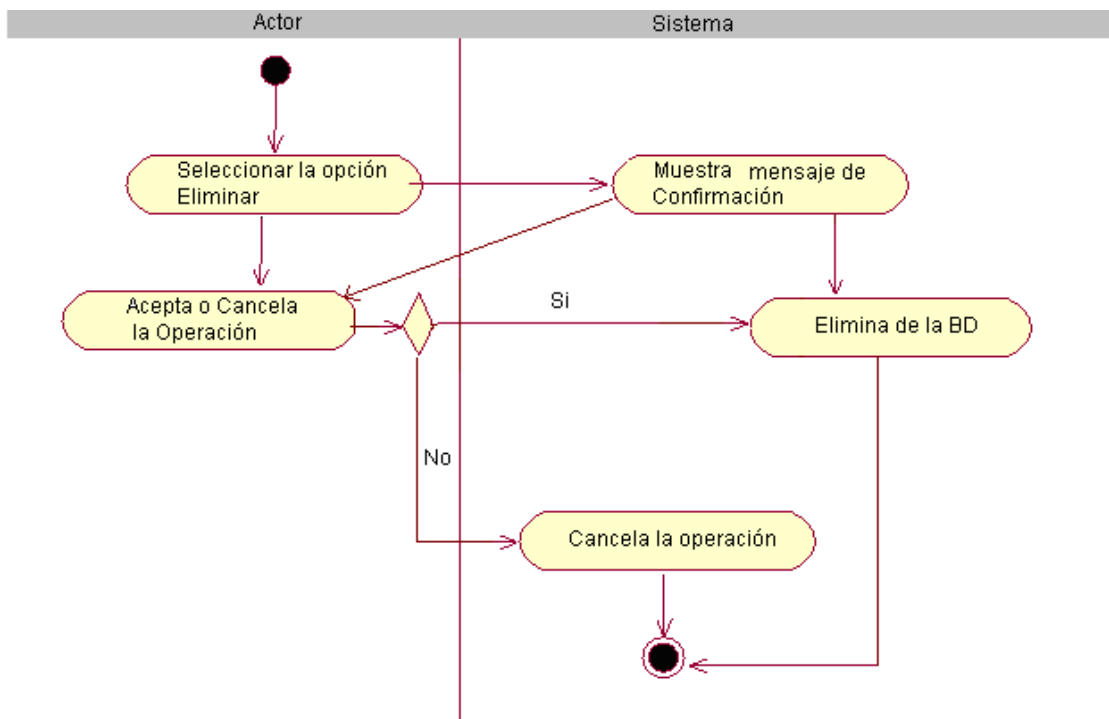
Adicionar Valores de los campos.



Modificar Valores de los campos.

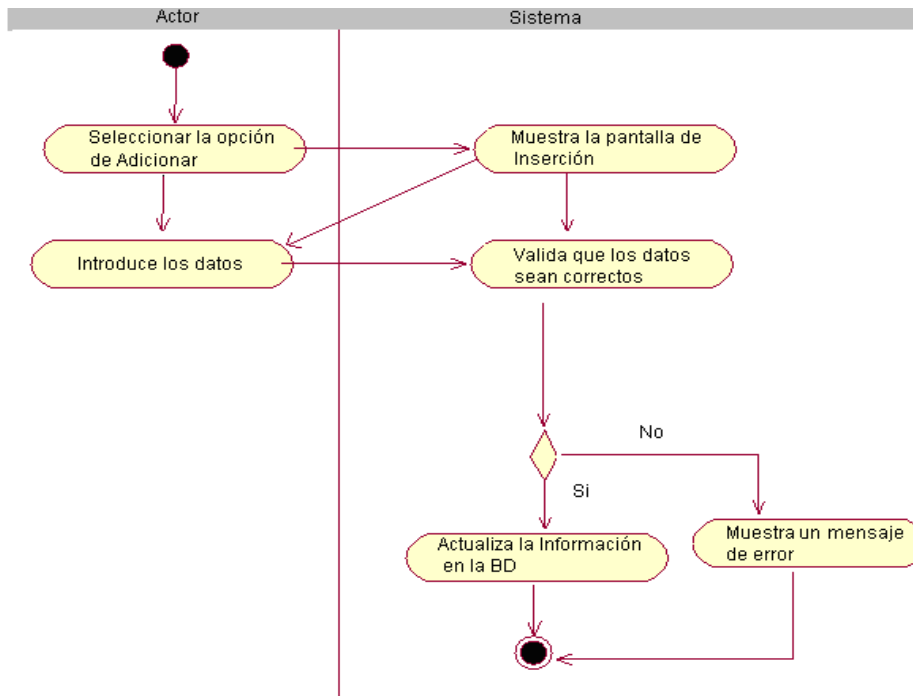


Eliminar Valores de los campos.

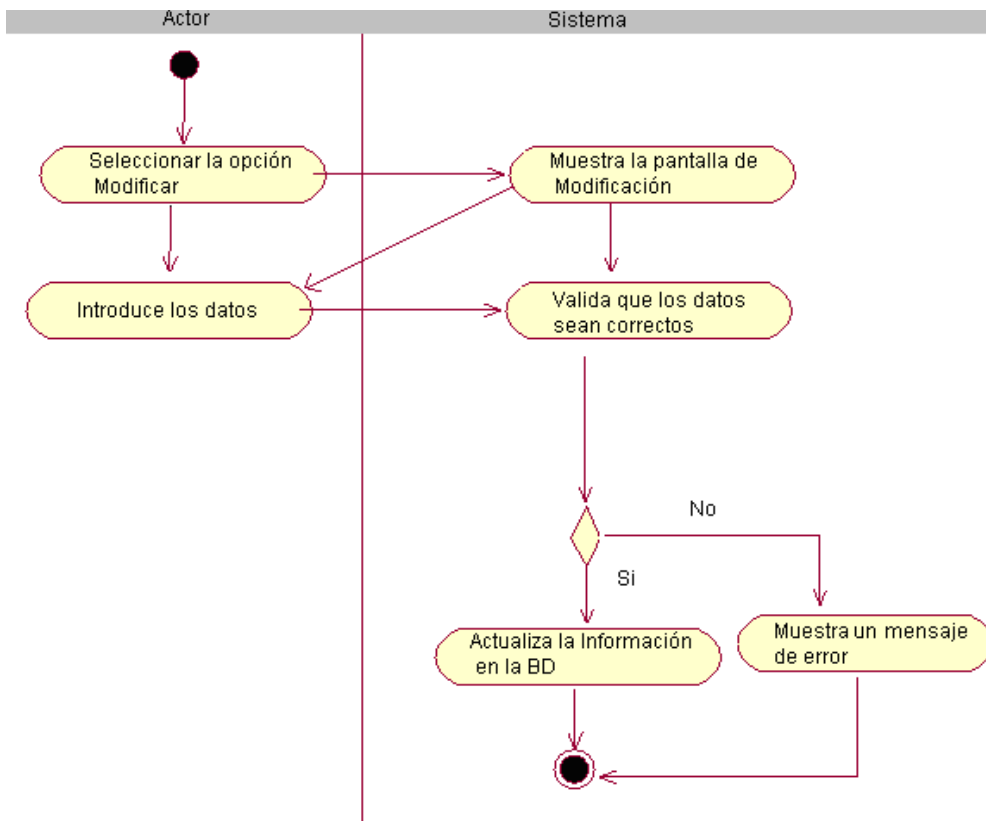


Anexo 37. Diagrama de Actividades (Flujo Alternativo: Gestionar mapas)

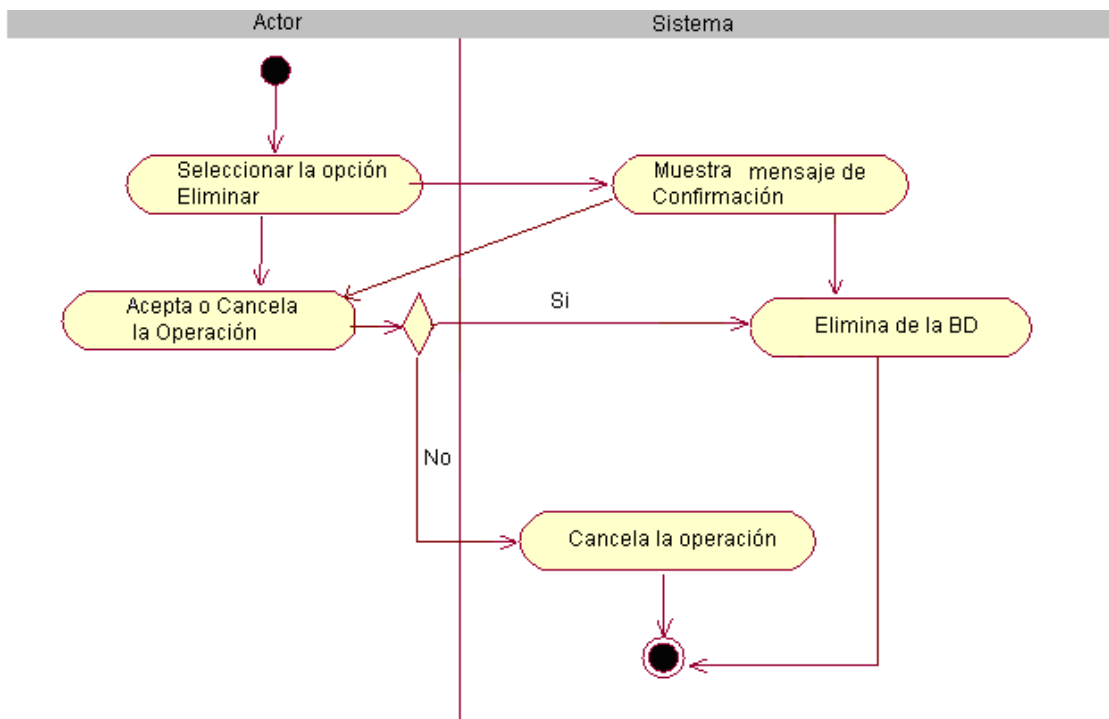
Adicionar mapa.



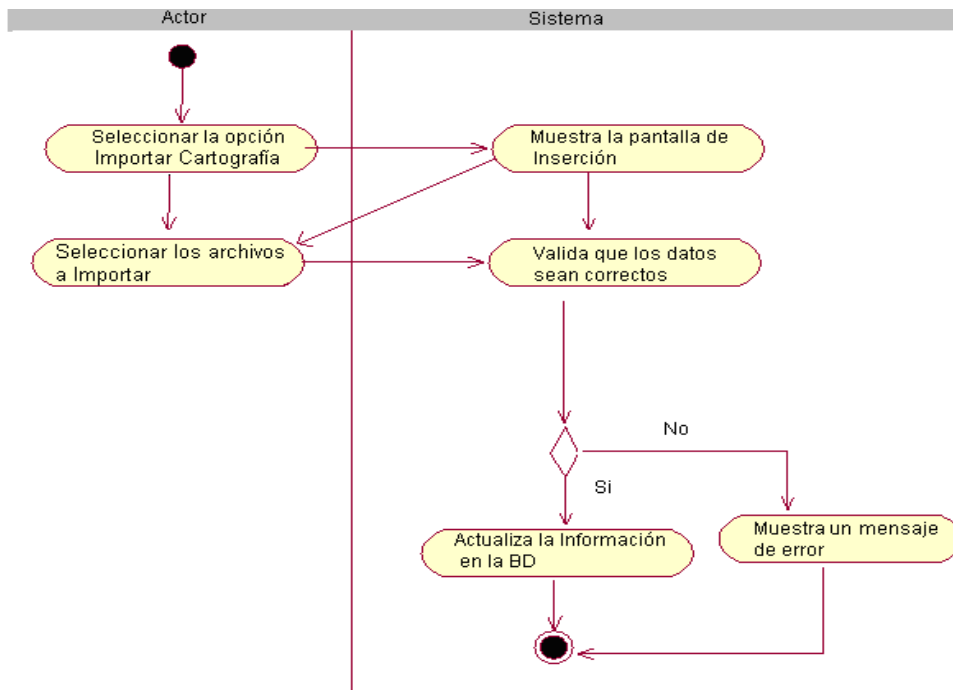
Modificar mapa.



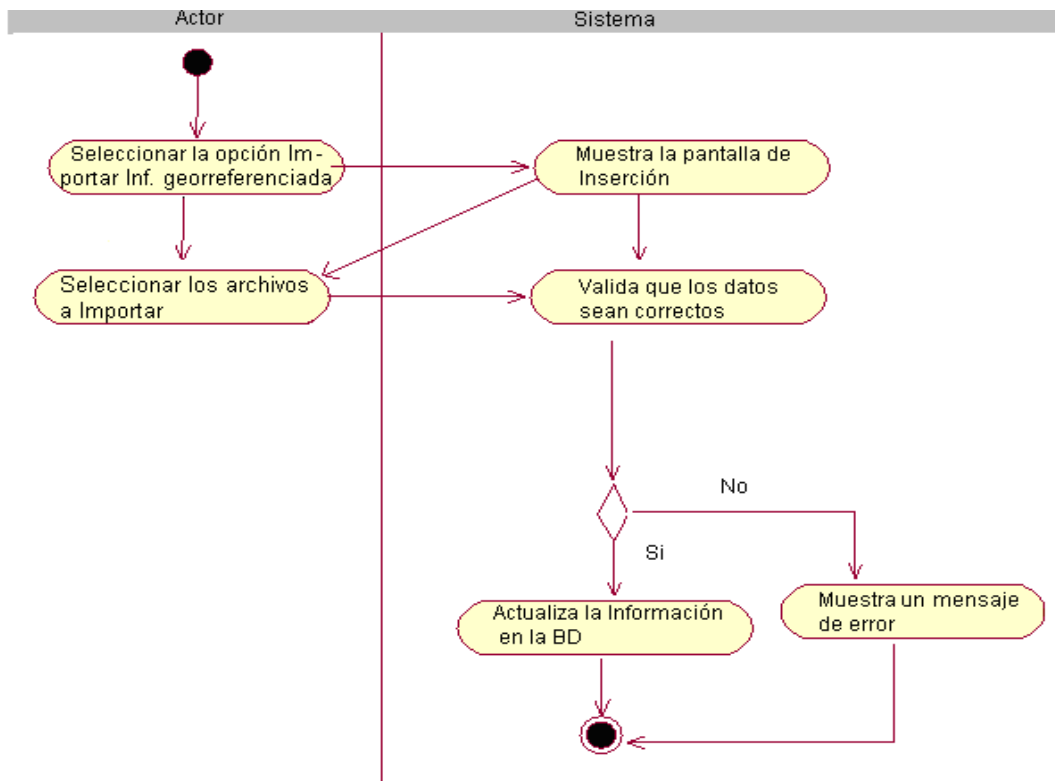
Eliminar mapa.



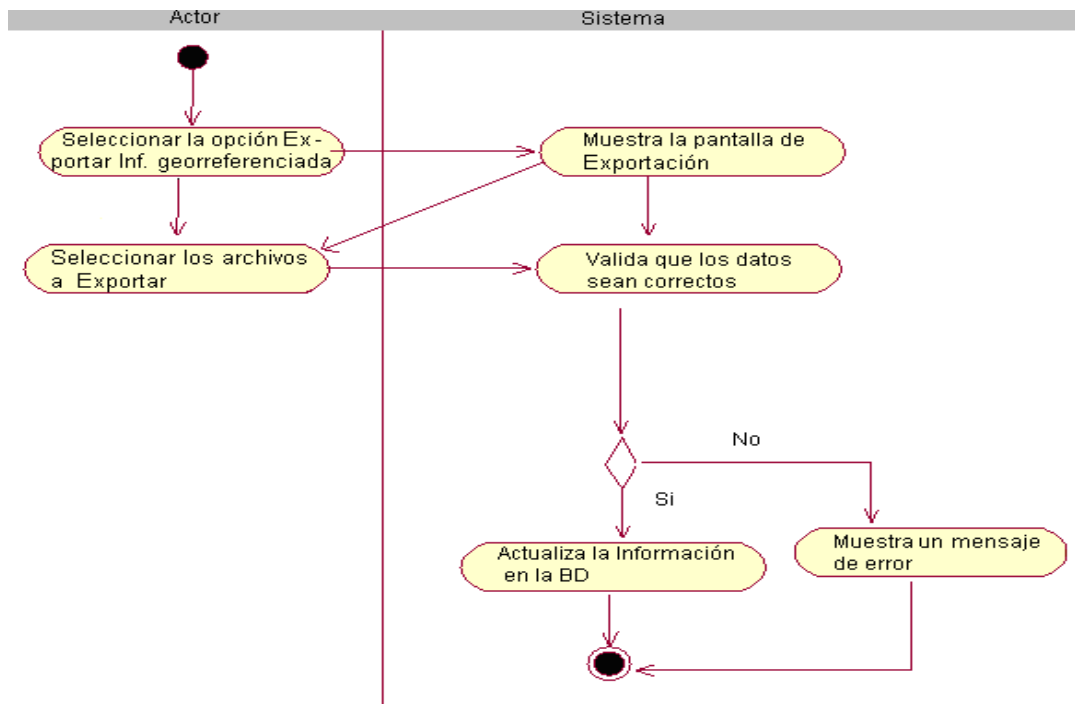
Anexo 38. Diagramas de Actividades (Flujo Alternativo: Importar cartografía)



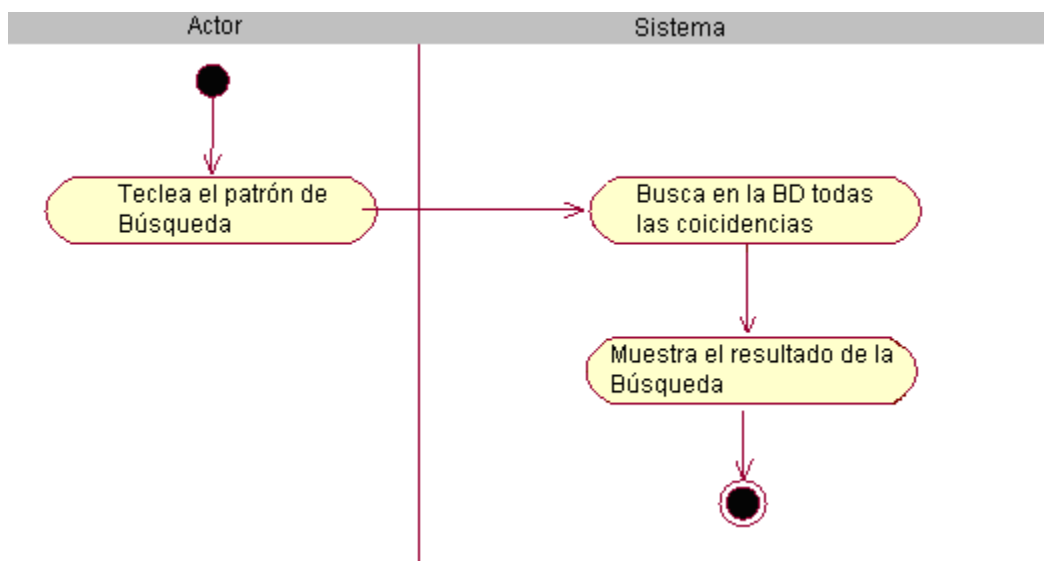
Anexo 39. *Diagramas de Actividades (Flujo Alternativo: Importar información georeferenciada publicada por los usuarios)*



Anexo 40. *Diagramas de Actividades (Flujo Alternativo: Exportar información georeferenciada publicada por los usuarios)*

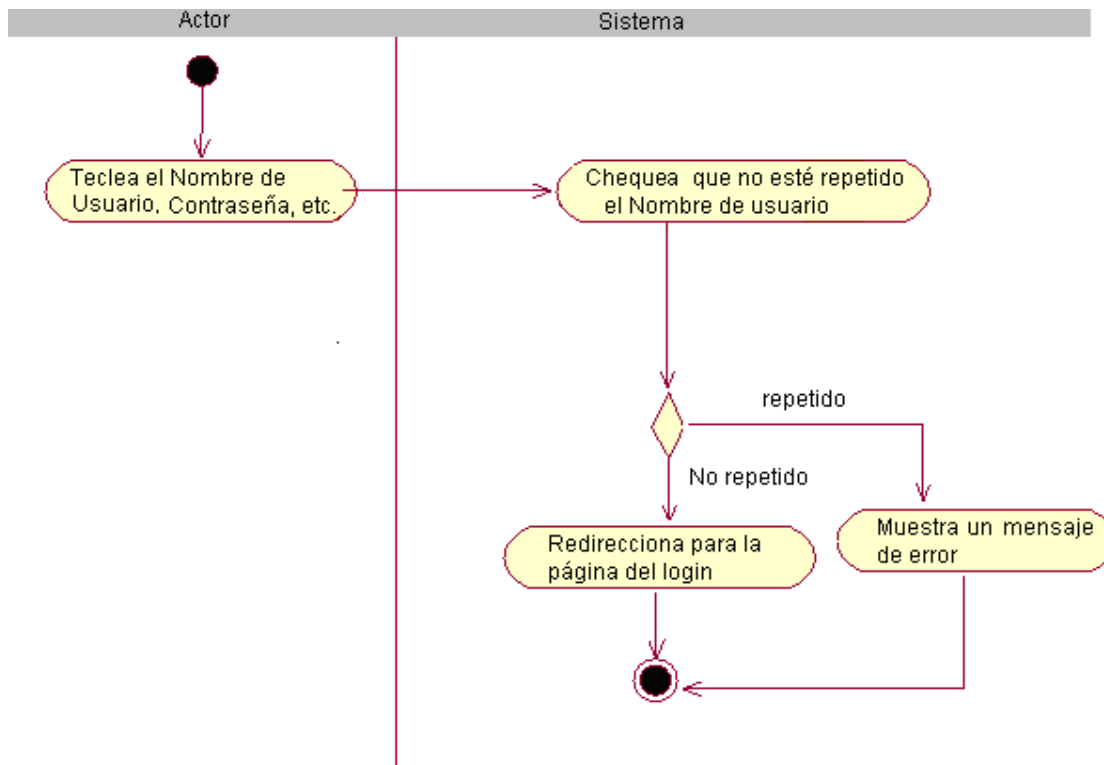


Anexo 41. *Diagramas de Actividades (Flujo Alternativo: Realizar búsquedas)*



Anexo 42. Diagramas de Actividades (Flujo Alternativo: Gestionar registro de usuarios)

Registrar usuario



Anexo 43. Diagramas de Actividades (Flujo Alternativo: Gestionar perfil de usuarios)

Modificar perfil

