

Universidad de las Ciencias Informáticas

“Facultad 6”



**“ALINEAMIENTO DE INFORMACIÓN EN BASES DE DATOS
ESPACIALES”**

**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autor: Carlos Enrique Luis De Armas

Tutores: MsC. Yanet Espinal Martín

Ing. Carlos Enrique Ramírez Martín

Co-tutor: Ing. Liester Cruz Castro

La Habana, 2014

“Año 56 de la Revolución”

**SI QUIERES CONSTRUIR UN BARCO,
NO EMPIECES POR BUSCAR MADERA, CORTAR TABLAS O DISTRIBUIR EL
TRABAJO;
PRIMERO HAZ DE EVOCAR EN LOS HOMBRES EL ANHELO DEL MAR LIBRE
Y ANCHO.**

ANTOINE DE SAINT-EXUPERY

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la LPS Aplicativos SIG de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2014.

Carlos Enrique Luis De Armas

MsC. Yanet Espinal Martín

Ing. Carlos Enrique Ramírez Martín

DATOS DE CONTACTO

Tutor: MsC. Yanet Espinal Martín

yanete@uci.cu

Licenciada en Ciencias de la Computación en la Universidad de Oriente (UO), graduada en el año 2005.

Profesor de la Universidad de las Ciencias Informáticas.

Tutor: Ing. Carlos E. Ramírez Martín

cmartin@uci.cu

Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) graduado en el año 2009.

Profesor de la Universidad de las Ciencias Informáticas.

AGRADECIMIENTOS

A todas aquellas personas que han aportado su granito de arena para que este momento se hiciera realidad. A mis abuelos, los que están y los que no están. A mi mamá, a mi papá, a mi tía, a mis hermanas, porque sin ellos, llegar hasta aquí hubiese sido imposible. A mi novia, aquella que me dio fuerzas cuando las cosas no salieron del todo bien. A mis compañeros y profesores, a mis amigos, con los que compartí cinco años maravillosos. A aquellos que fueron mi familia, en momentos de alegría o de tristeza, cuando por motivos de distancia, otros no se encontraban. A mis tutores, aquellos que supieron guiarme y contribuyeron en gran medida a que este sueño se hiciera realidad. A todos, Gracias.

DEDICATORIA

Dedico este trabajo a toda mi familia, a los que se encuentran más cercanos o más distantes; a todos aquellos que de una forma u otra han estado siempre a mi lado.

RESUMEN

En la Línea de Productos de Software (LPS) Aplicativos SIG, el uso de Bases de Datos (BD) se hace fundamental en la creación de un nuevo software. En las cartografías relacionadas a las BD se tiene almacenada información, la cual se encuentra representada de diferentes maneras en cada cartografía, provocando que no exista una uniformidad en dicha información. Para lograr una integridad en el software a desarrollar en la LPS, y en la información manejada, se hace necesario disminuir las irregularidades existentes en las BD. Reconocida la necesidad explicada anteriormente, se propuso el desarrollo de un sistema para alinear información en las BD, con el objetivo de eliminar dichas irregularidades. Para la construcción del sistema se hace uso de tecnologías libres, tales como: NetBeans 7.3, como Entorno Integrado de Desarrollo (IDE), Visual Paradigm Enterprise Edition, como herramienta de modelado, así como los lenguajes de programación y modelado, Java y UML 2.0 respectivamente. Las herramientas mencionadas se integran en un proceso de desarrollo de software con un enfoque ágil que utiliza las prácticas definidas en la metodología Agile Unified Process (AUP). Las pruebas realizadas a la aplicación tuvieron resultados satisfactorios, lo que garantiza el correcto funcionamiento de la aplicación y la satisfacción del cliente.

Palabras claves: Cartografías, información, irregularidades.

ABSTRACT

In Software Product Line (LPS) Aplicativos SIG, the use of databases (DB) is fundamental in the creation of new software. In the maps related to the BD, information is stored which is represented in different ways in each mapping, so that there is no uniformity in the information. To achieve the integrity of software to be developed in the LPS, and on the information managed, it is necessary to diminish irregularities in the BD. Recognized the need discussed above, the development of a system is proposed, with the goal of diminish such irregularities. For the construction of such a system using free technologies is made, such as NetBeans 7.3, as Integrated Development Environment (IDE), Visual Paradigm Enterprise Edition, as a tool for modelling and programming languages and modelling, Java and UML 2.0 respectively. These tools are integrated into a software development process to an agile approach using the practices defined in the methodology Agile Unified Process (AUP). Testing of the application were successful, ensuring the proper functioning of the application and satisfaction client.

Key Words: Maps, information, irregularities.

ÍNDICE DE CONTENIDOS

Introducción	1
Capítulo 1: “Fundamentación teórica del proceso de alineamiento de la información en bases de datos geoespaciales”	8
1.1. Conceptos asociados al dominio de la investigación.....	8
1.2. Descripción del dominio del problema	10
1.3. Análisis de soluciones similares.....	12
1.3.1. Ontologías	12
1.3.2. Sistemas Gestores de Bases de Datos	16
1.3.3. Procesamiento de Lenguaje Natural	17
1.3.3.1. Minería de texto	18
1.3.3.2. Algoritmos de similitud de cadena	19
1.4. Metodología de desarrollo	21
1.5. Lenguaje de Modelado. UML v2.0.	23
1.6. Herramienta CASE: Visual Paradigm for UML 8.0 Enterprise Edition	24
1.7. Lenguaje de programación Java.	24
1.8. IDE NetBeans 7.3	25
1.9. Conclusiones del capítulo	25
Capítulo 2: El proceso de alineamiento de información en bases de datos espaciales.	27
2.1. Diagrama del dominio	27
2.2. Requisitos Funcionales	29
2.3. Requisitos no Funcionales	31
2.3.1. RNF 1. Usabilidad e Interfaces.....	31
2.3.2. RNF 2. Seguridad	31
2.3.3. RNF 3. Rendimiento.....	32
2.3.4. RNF 4. Software.....	32
2.3.5. RNF 5. Hardware	32
2.4. Casos de Uso	33
2.4.1. Descripción de Casos de Uso arquitectónicamente significativos	34
2.5. Arquitectura.....	40

ÍNDICE DE CONTENIDOS

2.6.	Algoritmo de similitud de cadenas de Levenshtein.....	41
2.7.	Conclusiones del capítulo	43
Capítulo 3: Implementación y validación del sistema de alineamiento de nomencladores en bases de datos geoespaciales.....		44
3.1.	Patrones de diseño	44
3.2.	Diagrama de clases de diseño	46
3.3.	Diagrama de componentes	48
3.4.	Diagrama de Despliegue.....	49
3.5.	Pruebas de software	49
3.5.1.	Pruebas de caja negra	49
3.5.2.	Encuesta de aceptación al interesado.....	59
3.6.	Conclusiones del Capítulo.....	60
Conclusiones		61
Recomendaciones		62
Referencias.....		63
Anexos.....		68

ÍNDICE DE TABLAS

Tabla 2.1 Nomencladores predefinidos para el alineamiento de información.	28
Tabla 2.2 Definición de Requisitos Funcionales del Sistema.....	29
Tabla 2.3 Descripción CU Calcular Errores.....	34
Tabla 2.4 Descripción CU Corregir Errores.....	36
Tabla 2.5 Descripción CU Alinear Información.....	37
Tabla 3.1 Secciones a probar en el caso de uso Gestionar Nomenclador.	50
Tabla 3.2 Tabla de variables.	55
Tabla 3.3 Matriz de datos SC Adicionar Nomenclador.	56
Tabla 3.4 Matriz de datos SC Modificar Nomenclador.	57
Tabla 3.5 Matriz de datos SC Eliminar Nomenclador.	57

ÍNDICE DE FIGURAS

Fig. 1.1 Instancias de las Tablas “Provincia” en dos BD diferentes.	12
Fig. 1.2 Ciclo de vida de AUP (Universidad Nacional de Costa Rica, 2006).....	23
Fig. 2.1 Diagrama de dominio.	27
Fig. 2.2 Diagrama de Casos de Uso del Sistema.	34
Fig. 2.3 Capas del patrón arquitectónico del sistema.....	41
Fig. 3.1 Diagrama de clases del diseño.	47
Fig. 3.2 Diagrama de componentes.	48
Fig. 3.3 Diagrama de despliegue.....	49
Fig. 3.4 Gráfico de no conformidades encontradas. Caso de uso Gestionar Nomenclador.	59

INTRODUCCIÓN

Desde los inicios de la humanidad ha existido una necesidad vital de almacenar información que valiera, bien como evidencias, o para realizar consultas futuras. La misma era recopilada, registrada y acumulada según su contenido pero, como es lógico, a lo largo de los años su volumen se fue haciendo cada vez más grande, requiriendo de mayor cantidad de espacio y de tiempo para efectuar cualquier tipo de investigación.

Estas dificultades obligaron al hombre a buscar una alternativa viable que le permitiera depositar toda la información obtenida. La solución a los problemas existentes con la acumulación de grandes volúmenes de información surge con la aparición de la informática: nacen las llamadas Bases de Datos (BD).

A lo largo de los años las BD se han ido modernizando y haciéndose cada vez más grandes y necesarias para la humanidad. La utilización de las BD se extiende a numerosas áreas de trabajo. Se pueden poner varios ejemplos de centros donde son utilizadas, en algunos con más complejidad que en otros, como pueden ser instituciones militares, industrias, empresas, en el ámbito de la salud, la educación, el turismo, pero al final su utilidad es imprescindible en lugares donde se almacene información. Su importancia puede ser fácilmente apreciable en instituciones que se encarguen de la predicción sismográfica en un país; es necesario saber cuándo y dónde ocurrieron los sismos para determinar las áreas de mayor riesgo y poder predecir con una mayor certeza su ocurrencia en un futuro.

La Universidad de las Ciencias Informáticas (UCI) es una de estas instituciones donde se utilizan en gran medida las BD. En la UCI existen varios centros de desarrollo asociados a las facultades docentes. Cada centro se especializa en desarrollar productos vinculados a las diferentes ramas de la economía y la sociedad en el país. Uno de los centros de desarrollo de la UCI es el Centro de Geoinformáticas y Señales Digitales (GEYSED), perteneciente a la facultad 6. Dicho centro es el encargado de desarrollar Sistemas de Información Geográfica (SIG) y aplicaciones asociadas al tratamiento de las Señales Digitales.

Dentro del centro GEYSED se encuentra la Línea de Productos de Software (LPS) Aplicativos SIG, donde se trabaja en la construcción de diferentes SIG, la cual cuenta con varios profesionales y estudiantes asignados para la realización de sus diferentes actividades. Su personal se encuentra distribuido en las

INTRODUCCIÓN

siguientes factorías: Interfaz de usuario; Integración de componentes; Gestión; Ingeniería y Base de datos espaciales y cartografías.

Los integrantes de la factoría *Interfaz de usuario* son los encargados de elaborar la interfaz del sistema que se está desarrollando. Estos deben definir todas las interfaces del producto, teniendo siempre en cuenta que debe ser de fácil manejo y entendimiento para el personal que va a interactuar con la aplicación.

El grupo de desarrolladores de la factoría *Integración de componentes* son los encargados del diseño, construcción y ensamblaje de los componentes, en la misma se une el trabajo desarrollado por la factoría de BD con el de Interfaz de usuario. El grupo de desarrollo de *Gestión*, tiene la tarea de controlar la calidad interna de los procesos de desarrollo de software en la LPS, además, deben realizar la planificación del proyecto, así como llevar el control de los estudiantes, velar por la asistencia y la correcta realización de las tareas asignadas e impartir los cursos de capacitación.

Los miembros del grupo de *Ingeniería*, tienen bajo su responsabilidad, llevar a cabo toda la documentación técnica del proceso de desarrollo de software. A su vez, estos deben elaborar los casos de prueba, el manual de usuario de la aplicación, así como, hacer un correcto levantamiento de todos los requisitos del sistema. Por último se encuentran los desarrolladores de la factoría de *Base de Datos espaciales y cartografías*, los cuales deben diseñar y elaborar las bases de datos para los nuevos sistemas a desarrollar, en estas va a estar guardada toda la información que los sistemas necesiten.

Cada factoría juega un papel fundamental en el desarrollo de un nuevo software al tener tareas y responsabilidades diferentes, a cumplir en un tiempo determinado ya que en ocasiones, una factoría depende del trabajo de otra. Esta estructura funciona como una línea de ensamblaje, donde cada factoría elabora una parte específica del software, para finalmente, todas juntas componer el producto requerido por el cliente.

La LPS Aplicativos SIG, para el desarrollo de sus productos, hace uso de herramientas libres tales como PostgreSQL para el almacenamiento y gestión de la información geoespacial asociada a las diferentes soluciones creadas. Las BD existentes en el proyecto guardan la información geoespacial de los diversos mapas, junto con la información socioeconómica asociada a cada elemento.

Se encuentran almacenadas además diversas cartografías pertenecientes a distintos tipos de elementos tanto de nuestro país, como de otros países de los que se han realizado SIG, dígase municipios, provincias, calles, viales, poblados. Cada una posee solamente los datos necesarios del sistema para la que fueron creadas, teniéndose varias versiones de estas con elementos distintivos y diferentes niveles de calidad. También existen cartografías referentes a un mismo elemento, contenidas en BD diferentes y elementos geoespaciales que son necesarios en una cartografía y se encuentran en otra.

Para el desarrollo de un nuevo producto de software se escoge la cartografía que más se adecue a las necesidades del cliente. Para ello se recurre a la factoría de BD, la cual está encargada de la gestión y el mantenimiento de la información asociada a las cartografías, y es quien, una vez establecido el modelo del nuevo software a desarrollar, debe mostrar una propuesta de diseño de BD para este producto.

La propuesta elaborada es discutida con el resto de las factorías y si es aprobada, es mostrada al cliente. Posteriormente se procede a la definición de la estructura final de la BD del software a desarrollar. Este proceso ha permitido la reutilización, en varias ocasiones, de los diferentes mapas existentes en el proyecto. Asimismo la información se ha actualizado, se han mejorado los mapas, obteniéndose nuevos mapas con un mayor nivel de detalles.

En las diferentes BD se pueden encontrar elementos que a su vez pueden estar contenidos en otras BD, pero estos, aunque hacen referencia a la misma información, no necesariamente están identificados de la misma manera, existiendo irregularidades en la información¹ que se encuentra almacenada. De esta forma se puede encontrar un mismo elemento geoespacial con diferentes nombres en varias BDs. Por ejemplo, basándose en una cartografía perteneciente a la provincia Sancti Spíritus, la misma puede aparecer en varias BD con diferentes nombres tales como Sancty Spíritus, Sanctispíritus, Santi spíritus, entre otros. Esto provoca que en el momento de hacer la integración, se reconozcan como elementos diferentes.

¹ Es importante definir que en el marco de la presente investigación, el término *irregularidades en la información* hace referencia solamente a las diferencia en la representación de los valores de nomencladores similares en diferentes bases de datos.

Otros de los problemas a tener en cuenta, es cuando el cliente presenta su propia BD y es necesario unificar² la misma con la elaborada por la LPS. Este problema trae consigo transformaciones necesarias al integrar los datos, como la inserción de nuevas columnas a las tablas o unir tablas por un determinado parámetro con el fin de lograr un acoplamiento entre ambas BDs. Asimismo puede ocurrir que el cliente tenga la información almacenada en la BD, representada de forma diferente a como se representa en la LPS. En dicho caso, también se haría necesario eliminar las irregularidades existentes en la información.

Para la LPS Aplicativos SIG, el solucionar los problemas de irregularidades en la información, de forma manual, resulta un consumo innecesario de recursos y tiempo, ya que en el momento de corregir los errores o hacer cambios en el sistema, los desarrolladores y administradores de BD deben hacerlo manualmente o en el propio código, teniendo que revisar cada cartografía, con el fin de eliminar todas la irregularidades en la información, lo cual se traduce en demoras y puede derivar situaciones tales como mal funcionamiento del sistema, incongruencias³ en la base de datos, desbordamientos de memoria y errores en las funcionalidades.

Todos estos aspectos atentan contra la realización del proyecto debido a que se producen retrasos en el cronograma de desarrollo, incidiendo negativamente tanto en el prestigio del proyecto, como en sus realizadores y en la satisfacción propia de los clientes. A su vez se compromete el prestigio de la Universidad, que puede conllevar a la pérdida de clientes, tanto nacionales como internacionales, los cuales pudieran reportar ingresos considerables a la economía del país.

El análisis de la problemática anterior origina el siguiente **problema de la investigación** a resolver: *Irregularidades en la información de las bases de datos geoespaciales de la LPS Aplicativos SIG.*

Con el propósito de facilitar la resolución del problema a resolver, se decide dividir este en subproblemas, obteniendo así las siguientes **preguntas científicas**:

² En el contexto de la presente investigación, *unificar BD* hace referencia a unir la información de diferentes BDs en una misma.

³ “*Falta de acuerdo, relación o correspondencia de una cosa con otra*” (RAE, 2005). En el contexto de la presente investigación, el término *incongruencias*, refiere a la representación de la misma información, de manera diferente en varias BD.

- ¿Cómo se soluciona el problema de irregularidades en la información en las BD espaciales en la factoría Base de Datos Espaciales y Cartografías de la LPS Aplicativos SIG?
- ¿Se ajusta esta vía de solución a las necesidades de los desarrolladores?
- ¿Existe alguna otra vía para solucionar el problema?
- ¿De qué manera pudiera solucionarse dicho problema, de forma que se emplee menos tiempo en lograr una integridad en la información y una mayor eficacia?

Como **objeto de estudio** se identificó al *procesamiento del lenguaje natural*.

El objeto de estudio planteado anteriormente se enfocó en el **campo de acción** de los *algoritmos de similitud de cadenas*.

Para dar solución al problema planteado se propone el siguiente **objetivo general**: *Desarrollar un sistema que permita alinear la información de las bases de datos espaciales en la LPS Aplicativos SIG*.

Para dar cumplimiento al objetivo planteado se proponen las siguientes **tareas de la investigación**.

- Realización del análisis del estado del arte de los sistemas para alinear la información de las bases de datos.
- Caracterización del proceso de alineamiento de la información en bases de datos.
- Caracterización de las tecnologías y herramientas en las que se apoya la solución del sistema.
- Diseño del sistema para alinear la información de las bases de datos.
- Implementación del sistema para alinear la información de las bases de datos.
- Realización de pruebas al sistema mediante la utilización de casos de pruebas de Caja Negra.
- Validación del sistema por parte del interesado.
- Elaboración de un artículo asociado al desarrollo y utilidad de la herramienta.

Con el fin de dar un mejor enfoque a la investigación, se utilizan una serie de métodos científicos para lograr un mayor entendimiento de la situación y guiar la solución al propósito deseado. Dichos métodos pueden ser clasificados en teóricos y empíricos.

Métodos Teóricos:

Analítico – Sintético: La investigación se encaminará a partir del análisis de los conceptos y métodos existentes relacionados al tema en materia, con el fin de obtener resultados y así tenerlos en cuenta para darle un mejor enfoque al sistema. Se analizará la información y la documentación recopilada para la selección de las posibles herramientas, lenguajes, metodologías lógicas y adecuadas para el desarrollo de la aplicación, para de esta forma realizar una síntesis de esta documentación.

La investigación, se tendrán en cuenta para la selección de patrones, arquitectura, así como para la elaboración de diagramas, modelos, definición de requisitos funcionales y no funcionales a utilizar en la propuesta de solución.

Histórico – Lógico: La investigación se realizará a partir del estudio de la información existente de la problemática analizada, lo cual permitirá conocer el desarrollo de aplicaciones vinculadas al alineamiento de información en bases de datos espaciales. Teniendo en cuenta las investigaciones realizadas y los resultados obtenidos por otros autores, se establecerán similitudes entre la propuesta de solución y lo desarrollado anteriormente.

Modelación: Permitirá modelar los diagramas necesarios para obtener una idea más acertada acerca del funcionamiento del negocio, análisis, diseño e implementación del sistema.

Inductivo – Deductivo: Este método estará orientado a establecer las generalizaciones sobre la base del estudio de casos particulares. A partir de la inducción de conocimientos, inferir deducciones que nos ayuden en el desarrollo del trabajo.

Métodos Empíricos.

Observación: Permitirá la ejecución correcta del proyecto, alertando sobre posibles atrasos en el cronograma. Posibilitará conocer si este no cumple con todos los requisitos estimados o pasa por alto alguno de estos, así como también, si existen fallas en el cumplimiento de las tareas propuestas o posibles errores en la aplicación. También brindará la posibilidad de observar el trabajo en el proyecto, así como los resultados de los experimentos realizados.

Como técnica de **recolección de información** se presenta la entrevista.

Entrevista: Para la ejecución de este método se realizarán encuentros con los especialistas de las diferentes factorías, jefe de la LPS Aplicativos SIG y jefe de la Factoría de Bases de Datos, con el fin de obtener la información necesaria para comprender los flujos de trabajo en el proyecto, así como, con las bases de datos y cartografía del mismo, para poder orientar el trabajo hacia las necesidades de los especialistas. (Estas entrevistas pueden encontrarse en los anexos 1.1 y 1.2).

A raíz de la presente investigación **se espera como resultado:**

- Documentación técnica asociada al sistema para alinear la información sobre BD espaciales en la LPS Aplicativos SIG.
- Sistema para alinear la información sobre BD espaciales en la LPS Aplicativos SIG.
- Informe de la Tesis.
- Publicaciones asociadas a la investigación. Participación en foros asociados al tema.

La presente investigación se encuentra estructurada en 3 capítulos. Estos abordan, de manera completa, los aspectos fundamentales del desarrollo de la solución al problema planteado anteriormente.

En el **capítulo 1** se tratan los elementos correspondientes a la fundamentación teórica del proceso de alineación de información en bases de datos espaciales, así como, los aspectos esenciales del negocio abordado y la selección de la metodología de desarrollo y herramientas a utilizar.

En el **capítulo 2** se abordan los elementos pertenecientes a la arquitectura, requerimientos y análisis de la solución.

En el **capítulo 3** se aborda lo concerniente al diseño, implementación y prueba de la solución.

CAPÍTULO 1: “FUNDAMENTACIÓN TEÓRICA DEL PROCESO DE ALINEAMIENTO DE LA INFORMACIÓN EN BASES DE DATOS ESPACIALES”

En este capítulo se hace una descripción de los aspectos teóricos concernientes a la investigación, se realiza la descripción del proceso del dominio y se plantean las bases para el entendimiento del mismo. Se tendrá presente la terminología a utilizar en el desarrollo del proyecto, así como un análisis de las soluciones similares, utilizadas para alinear información en BD espaciales. También se plasmará una descripción del negocio y finalmente se definirá la metodología y las herramientas de desarrollo a utilizar.

1.1. Conceptos asociados al dominio de la investigación.

Para un mejor entendimiento de la investigación, a continuación se definen una serie de conceptos que son fundamentales para el correcto desarrollo de la misma.

Base de Datos

Las BD juegan un papel muy importante en la mayoría de las entidades debido al gran volumen de información que es necesario almacenar hoy en día. Asimismo, la solución propuesta en esta investigación va encaminada a resolver problemas en las mismas, haciéndose indispensable abordar con profundidad su definición.

De acuerdo con Ian Gilfillan, en su libro “La Biblia de MySQL”, una base de datos es *“una colección de archivos relacionados”* (Gilfillan, 2000). Asimismo, C. J. Date, en su libro “Introducción a los Sistemas de Bases de Datos”, define una base de datos como *“un sistema computarizado para llevar registros. Es posible considerar a la propia base de datos como una especie de armario electrónico para archivar; es decir, es un depósito o contenedor de una colección de archivos de datos computarizados”* (Date, 2001).

Con el desarrollo tecnológico y el avance de la informática, se pueden encontrar la mayoría de las bases de datos de hoy en día en formato electrónico. Esta solución ha permitido facilidades en el almacenamiento y búsqueda de los datos contenidos en las BD.

Datos Espaciales

Los datos espaciales, constituyen una parte importante de la información que se almacena en una BD geoespacial. Por esta razón es fundamental conocer específicamente, que son estos datos.

La compañía Microsoft en su página “Microsoft TechNet” propone que: *“Los datos espaciales representan información sobre la ubicación física y la forma de objetos geométricos. Estos objetos pueden ser ubicaciones de punto u objetos más complejos como países, carreteras o lagos”* (Microsoft, 2014).

De acuerdo con la empresa Española “Ager Ingenieros”: *“Los datos espaciales o geodatos presentan dos tipos de propiedades: las geométricas y las descriptivas. Estas propiedades son las que les proporcionan su utilidad, constituyendo así el núcleo de los Sistemas de Información Geográfica”* (Ager, 2003).

“Los datos espaciales describe la información relacionada con el espacio ocupado por los objetos. Los datos consisten en información geométrica y puede ser discreta o continua. Los datos discretos pueden ser un solo punto en el espacio multidimensional, sin embargo los datos espaciales discretos difieren de los datos no espaciales en que tiene un atributo de la distancia que se utiliza para localizar los datos en el espacio. Los datos continuos se extienden por una región del espacio. Estos datos pueden consistir en imágenes médicas, las regiones del mapa, o campos de estrellas” (Samet, 1994).

A continuación se abordará el concepto de **base de datos espacial** con el fin de lograr un mejor entendimiento del contexto de la investigación.

Bases de Datos Espaciales

Teniendo en cuenta los conceptos abordados hasta el momento, relacionados a “bases de datos” y “datos espaciales”, se puede proseguir a la definición del concepto de base de datos espacial.

Según R. H. Güting *“Una base de datos espacial es un sistema de bases de datos que ofrece tipos de datos espaciales en su modelo de datos y lenguaje de consulta, soporta diversos tipos de datos espaciales en su implementación, proporcionando al menos indexación espacial y métodos de unión espacial”* (Güting, 1994).

“Una base de datos espacial se define también como una colección de datos referenciados en el espacio que actúa como un modelo de la realidad” (National Center for Geographic Information and Analysis, 1990).

“Bases de datos espaciales son los sistemas de bases de datos que manejan datos espaciales. Ellos están diseñados para manejar tanto la información espacial, como los atributos no espaciales de esos datos” (Samet, 1994).

El uso de nomencladores en las BD espaciales utilizadas en la confección de SIG es fundamental, estos contienen la información geoespacial y socioeconómica de diferentes elementos. Asimismo los nomencladores son parte de la base de la presente investigación, por lo que es indispensable el conocimiento de su significado y su uso.

Nomenclador

El Diccionario Enciclopédico Vox 1, define un nomenclador como un *“Catálogo de nombres de carácter técnico u oficial* (Diccionario Enciclopédico, 2009).

La Real Academia de la Lengua Española define los nomencladores como: *“Catálogo de nombres, ya de pueblos, ya de sujetos, ya de voces técnicas de una ciencia o facultad* (Real Academia, 2014)”.

En la presente investigación, se considerará a un nomenclador como un código o un identificador, el cual está contenido dentro de una lista de códigos que fueron identificados y aprobados por los usuarios finales, para denotar elementos geoespaciales. Estos nomencladores pueden verse expresados como entidades o lugares, por ejemplo: provincias o municipios.

1.2. Descripción del dominio del problema

Dentro de la LPS Aplicativos SIG, la factoría de BD, es la encargada de la gestión y el mantenimiento de la información asociada a las cartografías almacenadas. Una vez establecida la descripción del dominio del nuevo software a desarrollar, y de haberse identificado los requisitos por parte de la factoría de ingeniería, la factoría de BD en conjunto con los analistas, desarrolla una propuesta de diseño de BD

geoespacial para este producto. Luego de elaborada la propuesta, es discutida con el resto de las factorías y de ser aprobada, se muestra al cliente. Por lo general, el comprador posee sus propias BDs, de ocurrir, el equipo debe integrar estas con la propuesta elaborada por la LPS, o relacionar los elementos similares entre las BDs geoespaciales de la LPS y el cliente. En caso de no ser posible, todo el grupo de desarrollo se reúne nuevamente para decidir la solución al problema.

Una vez tomada la decisión, se le informa al cliente y se procede a la definición de la estructura final de la BD del software a desarrollar. Luego se realiza la migración del modelo de la BD propuesta a una BD física. Cada cartografía contiene sus propios nomencladores y a su vez cada nomenclador contiene información geoespacial y socioeconómica referente al lugar o entidad que representan. Tras seleccionar los mapas que cumpla las condiciones identificadas por el cliente, se procede a corregir la información existente en dichos mapas, pues pueden encontrarse nomencladores con nombres similares, pero no exactamente iguales, lo que provoca que se reconozcan de forma diferente.

Para realizar el proceso descrito anteriormente se debe tener presente que en ocasiones el volumen de información manejado es elevado y este se realiza de forma manual. Dicha situación trae consigo atrasos en el cronograma y en algunos casos puede darse la situación de que los desarrolladores cometan errores tales como: obviar la corrección de algún nomenclador necesario o asignarle un valor no debido, que deba estar presente en la cartografía. Dichos errores se traducirían en pérdida de información geográfica o socioeconómica, afectando el resultado final del sistema.

A su vez, al migrar las BDs, puede ocurrir que sea necesario unir dos tablas que contengan información referente al mismo elemento y se encuentren en BDs diferentes. Este proceso también debe hacerse de forma manual y es necesario a la hora de lograr una comunicación entre más de una BD, así como relacionar campos similares entre estas.

A continuación en la fig. 1.1 se presenta un ejemplo de irregularidad en la información en dos tablas provincia de dos BD diferentes.

ID	Nombre
1	Pinar del Río
2	La Habana
3	Artemisa
4	Mayabeque
5	Matanzas

ID	Nombre
1	P. del Río
2	La Habana
3	Artemisa
4	Mataveke
5	Matanzas

Fig. 1.1 Instancias de las Tablas "Provincia" en dos BD diferentes.

1.3. Análisis de soluciones similares

A nivel mundial se han realizado diversas investigaciones para darle solución a la problemática tratada en este trabajo. A continuación se presentará un análisis de las soluciones similares con el objetivo de encontrar elementos en común que puedan ser útiles al desarrollo de esta investigación, o simplemente comprobar si alguna de estas es compatible para enfrentar el problema científico planteado.

1.3.1. Ontologías

El uso de ontologías, como forma de representación del conocimiento, es una de las técnicas usadas en la actualidad para lograr la unidad semántica entre diferentes sistemas. *"Los sistemas de referencia semánticos usualmente confían en el uso de ontologías, debido a que las ontologías proveen definiciones formales explícitas de entidades temáticas y sus relaciones y por lo tanto, facilitan las definiciones de métodos para proyectar, trasladar e integrar información geográfica obtenida de diferentes fuentes"* (Garea Llano, 2009).

Una de las definiciones más utilizadas en la actualidad para definir el término "Ontología" es la de T.R. Gruber, *"Una ontología es una especificación explícita de una conceptualización"* (Gruber, 1995). Asimismo el Dr. Eduardo Garea Llano plantea que: *"Una ontología es una especificación formal, explícita de una conceptualización compartida de un dominio de interés"* (Garea Llano, 2009).

El uso de ontologías se extiende hasta los SIG que usan información semántica. A este tipo de ontologías se les denomina geo-ontologías. *“Las geo-ontologías cumplen con todas la características de una ontología convencional, pero tienen además propiedades propias de este dominio, por ejemplo, un par de coordenadas (x, y) que representa la posición geográfica de un objeto. También incluye una serie de relaciones topológicas con una semántica determinada”* (Garea Llano, 2009).

A escala internacional se han realizado diversas investigaciones con el fin de darle solución al problema del alineamiento de la información a través del uso de ontologías. Uno de los ejemplos a citar es la investigación desarrollada por el Laboratorio de Interoperabilidad Semántica de Munster (MUSIL por sus siglas en inglés) ubicado en el Instituto para Geoinformáticas de la Universidad de Munster, en Alemania. Dicha investigación está dirigida a las ontologías y se basa en lograr una mejora en el uso de información geoespacial a través de la validación de interoperabilidad semántica e integración.

Sus proyectos se ven encaminados a áreas específicas como son: el planeamiento del transporte, planeamiento medio ambiental y el manejo de emergencia (Münster, 2014). Aunque esta solución está dirigida a resolver un problema semejante al de la presente investigación, la misma está encaminada a negocios con características diferentes, ya que los sistemas desarrollados por el MUSIL son Sistemas de Referencia Espacial, los cuales se utilizan para traducir los datos de un sistema de referencia a otro, por lo que por esta razón, su uso se imposibilita.

Otra de las investigaciones relacionada con la temática que se aborda, es la tesis para optar por el título de Master en Ciencias de la Computación desarrollada por Adriana López Cumplido de la Facultad de Ciencias de la Computación Benemérita Universidad Autónoma de Puebla, denominada *“Desambiguación de Topónimos para el Tratamiento de Ambigüedad en Consultas de Recuperación de Información Geográfica usando una Ontología Mixta para el idioma Español”* (López, 2011). Esta investigación propone el uso de una ontología para solucionar el problema de *desambiguación del sentido de las palabras* (WSD⁴ por sus siglas en inglés), teniendo en cuenta la proximidad geográfica

⁴*“Este término, es un mecanismo lingüístico para definir el sentido correcto de una palabra, basándose en el contexto donde se emplee, en función de sus posibles sentidos semánticos”* (López, 2011).

entre *topónimos*⁵ del mismo contexto. Para el cálculo de dicha proximidad, se hace uso de la Función de Haversine (distancia de Haversine), la cual es una función matemática que permite calcular la distancia entre dos puntos geográficos.

La investigación mencionada anteriormente aportó a la presente investigación, fundamentalmente, la idea de la necesidad de calcular la distancia entre dos elementos, con el fin de identificar cuan semejantes son. Asimismo, se pudo apreciar, que dicha distancia puede ser calculada a través de una función matemática.

En nuestro país se han realizado investigaciones y presentado soluciones para resolver el problema existente. El Dr. Eduardo Garea Llano, jefe del Departamento de Reconocimiento de Patrones del Centro de Aplicaciones de Tecnología Avanzada en Cuba (CENATAV), presentó en marzo del 2007, un reporte investigativo denominado “*Estado actual de la Interpretación Semántica de datos espaciales*”, donde expone el estado del arte acerca de los métodos de interpretación semántica de datos espaciales. Además hace un resumen de los principales resultados obtenidos en los últimos cinco años y los problemas no resueltos. Explica la necesidad de lograr una unidad semántica tanto para los SIG como para los Sistemas de Información Geográfica gobernados por ontologías (SIGGO), con el fin de que todo usuario pueda entender cualquier representación semántica (nomenclatura) hecha por cualquier entidad o por otro usuario (Garea Llano, 2007). A raíz de esta investigación se pudieron identificar los principales problemas que han aparecido en el intento de realizar una unidad semántica a través del uso de ontologías, así como cuales han sido las principales soluciones propuestas.

El propio Dr. Garea Llano y el Lic. Francisco Vera Voronisky, elaboraron un informe denominado Alineamiento de Ontologías en el dominio geoespacial. Dicho informe tiene como temática el alineamiento de geo-ontologías y expone las técnicas, así como las principales herramientas desarrolladas para alinear ontologías (Garea Llano, 2009). El principal aspecto a tener en cuenta de dicho informe para la presente investigación, radica en la explicación que realizan acerca de diferentes técnicas para el cálculo de las medidas de similitud, lo cual se hace indispensable para darle solución al problema que se presenta.

⁵ “*Nombres propios de lugares*” (RAE, 2005).

Uno de los aspectos más significativos que aborda el informe mencionado anteriormente en relación con la presente investigación radica en la definición del término de alineamiento de ontologías. A continuación se presentan las definiciones de alineamiento y alineamiento de ontologías propuestas en dicho informe.

“Alinear algo significa “traer en línea”. Esta es una breve definición que enfatiza que el alineamiento es una actividad en la cual después haberse realizada, los objetos involucrados están en mutua relación” (Garea Llano, 2009).

“Ehrig (Ehrig, 2007) plantea que alinear una ontología con otra significa que, por cada entidad (concepto, relación, o instancia) en la primera ontología, se trata de buscar una entidad correspondiente, la cual pretenda tener el mismo significado en la segunda ontología” (Garea Llano, 2009).

Asimismo, el Diccionario “Manual de la Lengua Española Vox”, define como alineamiento: *“Colocación de varias cosas o personas en línea recta. Conjunto de cosas alineadas o colocadas en línea recta”* (Manual de la Lengua Española, 2007).

Hans Weigand, en su libro “Multilingual Ontology-Based Lexicon for News Filtering - The TREVI Project” define una ontología como: *“Una ontología es una base de datos que describe los conceptos generales o sobre un dominio, algunas de sus propiedades y cómo los conceptos se relacionan unos con otros”* (Weigand, 1997).

Una de las soluciones presentadas es “Sistemas de Información Geográfica como herramienta para la integración e interpretación semántica de la información espacial”, desarrollado por el CENATAV. Este trabajo mediante una estructura ontológica, presenta la integración de los datos y de herramientas para el alineamiento de geo-ontologías. Los autores de dicho trabajo plantean que *“Falta la unión semántica de los mapas. Y no solo de mapas, sino de “bases geográficas” y de forma general de las representaciones geo-espaciales”* (CENATAV, 2014).

En dicha investigación se presenta una primera implementación de la propuesta de integración SIGGO para el manejo de los resultados de la zonificación agroecológica de cultivos. De la solución mencionada se puede obtener como principal referencia para el uso en la presente investigación, la manera en que

muestran los elementos geospaciales contenidos dentro de la base de datos, lo cual se hace fundamental para lograr un mayor entendimiento del funcionamiento del sistema por parte del usuario.

Como principal desventaja presentada por la investigación mencionada, es que su sistema se encuentra orientado a un negocio con características diferentes al que se encuentra orientada la presente investigación.

Las ontologías, a pesar de ser muy útiles para resolver problemas donde los elementos semánticos no varían, estas presentan una serie de desventajas. En caso de que haya alguna variación dentro de la semántica, ya sea que se modifiquen, adicionen o eliminen elementos, sería necesario redefinir toda la ontología. Asimismo su desarrollo requiere un equipo experimentado con vastos conocimientos respecto al tema. *“La confianza lógica en una ontología se basa en la adecuación, la consistencia y en cierta medida, en su completitud”* (Borrego-Días, 2005). *“La revisión puede producir una ontología consistente pero dañina o inadecuada para el procesamiento del conocimiento”* (Alonso-Jimenez, 2006). Por estas razones, se considera que la construcción de una ontología, no debe ser tomada como solución al problema que se presenta.

Teniendo en consideración los conceptos y las definiciones abordadas anteriormente de **ontología**, **alineamiento** y **alineamiento de ontologías**, el autor de la presente investigación propone que: en el marco de esta investigación, se entienda por **alineamiento de la información en una base de datos espacial**, al proceso de establecer una linealidad en la información referente a todos los elementos geospaciales contenidos en una BD espacial, con el fin de lograr una uniformidad en la información de dichos elementos.

1.3.2. Sistemas Gestores de Bases de Datos

Por las características que presentan los Sistemas Gestores de Base de Datos (SGBD), en cuanto a funcionalidades y trabajo con las BDs, se pueden establecer cierta similitud entre estos y la aplicación a desarrollar como resultado de la presente investigación. Un SGBD, es una aplicación para manipular los datos en las BDs. Algunos ejemplos de SGBD libres son PostgreSQL, MySQL, Sqlite Manager, entre otros.

Los SGBD permiten visualizar, adicionar, eliminar o modificar datos dentro de una BD. Se pueden considerar posibles soluciones ya que permiten solucionar los problemas que dan raíz a la presente investigación, pero no de forma automatizada. Su uso permitiría modificar los valores de los nomencladores representados incorrectamente, o unir tablas por determinado parámetro, pero los desarrolladores aún tendrían que realizar los cambios distintamente.

La principal ventaja que presentan estos SGBD, es que pueden ser utilizados como apoyo para el desarrollo de la solución propuesta en la presente investigación. Sus interfaces pueden ser tomadas como modelo para el desarrollo de la solución. Aunque estos cuentan con más funcionalidades y una estructura más compleja, su diseño es ideal para el sistema a desarrollar gracias a su estructura organizada.

1.3.3. Procesamiento de Lenguaje Natural

La empresa Vicomtech-IK4⁶ define que *“El **procesamiento del lenguaje natural** (PLN) es el campo que combina las tecnologías de la ciencia computacional (como la inteligencia artificial, el aprendizaje automático o la inferencia estadística) con la lingüística aplicada, con el objetivo de hacer posible la comprensión y el procesamiento asistidos por ordenador de información expresada en lenguaje humano para determinadas tareas, como la traducción automática, los sistemas de diálogo interactivos, el análisis de opiniones, etc.”* (Vicomtech-IK4, 2014).

“El PLN es de manera general, un conjunto de instrucciones que un sistema recibe en un lenguaje de programación dado, que permita comunicarse con un humano en su propio lenguaje, este procesamiento presenta diversas aplicaciones:

- *Corrección de textos.*
- *Traducción automática.*
- *Recuperación de la Información.*
- *Extracción de información y resúmenes.*
- *Búsqueda de documentos.*
- *Sistemas inteligentes para la educación y el entrenamiento.*

⁶ **Vicomtech-IK4** es un centro de investigación aplicada especializado en las tecnologías de Computer Graphics, Visual Computing y Multimedia, fundado en 2001 y localizado en el Parque Tecnológico de San Sebastián.

Además de su utilidad en el campo del procesamiento y la recuperación de información, el PLN se aplica a otros aspectos como el reconocimiento del habla o la corrección ortográfica de textos” (Benavides, 2007).

1.3.3.1. Minería de texto

Una de las técnicas de PLN más utilizadas es la minería de textos. La minería de textos es una tecnología utilizada para la búsqueda de patrones o tendencias dentro de grandes cantidades de texto. *”Se conoce con el nombre de minería de textos a aquel proceso consistente en extraer conocimiento a partir de fuentes masivas de información en forma de texto que nos permita tomar decisiones estratégicas” (Rio y Cilleruelo, 2010).* Esta permite descubrir el conocimiento que no se encuentra literalmente escrito dentro de determinado documento.

“En el caso de la minería de textos, puede decirse que es una herramienta capaz de abarcar una amplia gama de dominios, desde aquellos de la recuperación y extracción de información, presentación, resumen de multi-documentos, minería de datos aplicada a textos, etcétera” (ACIMED, 2007).

“La minería de texto es una herramienta de análisis encargada del descubrimiento de conocimiento que no existía explícitamente en ningún texto de la colección, pero que surge de relacionar el contenido de varios de ellos” (Hearst, 1999).

En las bibliografías consultadas, refieren al uso de la minería de textos fundamentalmente para (Bordón L, 2004):

- Extraer información relevante de un documento.
- Agregar y comparar información automáticamente.
- Clasificar y organizar documentos según su contenido.
- Organizar depósitos para búsqueda y recuperación.
- Clasificar textos e indizarlos en el Web.

Por las características que presenta la minería de textos, no se considera viable para solucionar el problema vigente en la presente investigación, ya que no se ajusta a las necesidades de la LPS que se intentan suplir con la presente investigación.

1.3.3.2. *Algoritmos de similitud de cadena*

Un algoritmo de similitud de cadena es un algoritmo para comparar cadenas de caracteres con el fin de conocer la similitud entre estas, muy utilizados para resolver problemas en el campo del procesamiento de textos y del lenguaje natural. El uso de estos algoritmos permite procesar cadenas de caracteres dentro de un texto determinado, con el fin de encontrar la cadena buscada, o comparar dos cadenas de caracteres con el objetivo de conocer cuán semejantes son.

A continuación se presentan al lector una serie de definiciones necesarias para lograr una mejor entendimiento del tema de los algoritmos de comparación de cadenas.

“Algoritmos de Similaridad y Distancia: Funciones que permiten ver que tan diferentes son las hileras⁷” (Ramírez, 2012).

“Al hablar de medida de similitud entre cadenas se estaría hablando de medir las diferencias que hay entre estas” (Padrón, 2006).

A continuación se presentarán tres de los principales algoritmos de similitud de cadenas estudiados.

Algoritmo Boyer-Moore

Este algoritmo ha sido el punto de referencia para la literatura de búsqueda de cadena. Fue desarrollado por Bob Boyer y J Strother Moore en 1977. Su funcionamiento se basa en identificar cadenas en un texto determinado. El algoritmo procesa la cadena que se busca de atrás hacia adelante o de derecha a izquierda buscando las coincidencias en cada posición.

El tiempo de ejecución de este algoritmo, puede tener un factor significativamente más bajo que muchos otros algoritmos de búsqueda ya que no tiene que buscar cada carácter de la cadena dada. Generalmente es más rápido cuanto más larga sea la cadena buscada. Su complejidad temporal es $O(n)$ ya que en el peor de los casos se necesitarían $3n$ comparaciones para encontrar todas las coincidencias (creativecommons, 2014).

⁷Cadenas.

Algoritmo Levenshtein

El algoritmo de Levenshtein, también conocido como Distancia de Levenshtein, es un algoritmo ideado por Vladimir Levenshtein en 1965, el cual permite medir la distancia entre dos secuencias de caracteres. Al recibir dos cadenas, este informa el número mínimo de modificaciones, ya sea de inserción, sustitución o eliminación de caracteres, necesarias para obtener la igualdad entre ambas cadenas. *“Construye una matriz con $y + 1$ filas y $x + 1$ columnas, siendo y la longitud de la cadena 1 y x la longitud de la cadena 2”* (Ramírez, 2012).

Este algoritmo cuenta con una serie de características, como son:

“Medida similaridad entre dos hileras: Hilera fuente (f) Hilera meta (m).

La similaridad entre dos palabras se calcula como el mínimo número de operaciones que transforma a una de las palabras en la otra (transformar f en m).

Las tres operaciones son: destrucción, inserción, sustitución” (Ramírez, 2012).

El algoritmo es usado en muchos correctores ortográficos. Es considerado una generalización de la distancia de Hamming⁸, aunque en el de Levenshtein no es necesario que las cadenas tengan la misma longitud. La complejidad temporal de este algoritmo es $O(m \cdot n)$, siendo m la longitud de la cadena 1 o cadena origen y n la cadena 2 o cadena destino (Consultores, 2011).

La función matemática llamada "Distancia de Levenshtein" cumple las tres propiedades básicas de una métrica (Algoritmos, 2006):

- *Reflexividad: $d(S, T) = 0$ si y sólo si $S = T$*
- *Simetría: $d(S, T) = d(T, S)$*
- *Desigualdad del triángulo: $d(A, B) \leq d(A, C) + d(C, B)$.*

⁸ **“Distancia de hamming:** Número de bits en que difieren dos palabras” (Abarca A., 2008).

El algoritmo procesa las cadenas de caracteres, convirtiéndolas en dos arreglos de caracteres. Luego compara los arreglos por posición, contando cada vez que no encuentre similitud entre los elementos de las posiciones analizadas. Finalmente, retorna el valor obtenido de las diferencias encontradas.

Algoritmo Ratcliff/Obershelp

Este algoritmo está diseñado para comparar dos cadenas y devolver el porcentaje de similitud entre estas. Este algoritmo es sensible a mayúsculas, al utilizarlo es necesaria la conversión de ambas cadenas a mayúscula o minúscula (Sinclair, 2010). Su complejidad temporal es de $O(m*n)$, siendo m la cadena 1 y n la cadena 2.

Una vez analizados los algoritmos en cuanto a complejidad temporal, características y funcionamiento, se puede determinar que: aún siendo el algoritmo de Boyer-Moore el de menor complejidad temporal, el que más se ajusta a las necesidades del sistema a desarrollar es el algoritmo de Levenshtein. Este permite comparar dos elementos con el fin de conocer cuán semejantes son, lo cual sería fundamental para identificar si se está en presencia del mismo elemento. Esta característica sería muy útil para identificar los elementos geoespaciales iguales, que se encuentran representados de manera diferente en las BD. Asimismo, la documentación referente a dicho algoritmo, es extensa en cuanto a implementación refiere, pudiéndola encontrar en distintos lenguajes de programación, lo cual representaría una ventaja en el momento de desarrollar el sistema propuesto como solución.

Al terminar con el estudio y análisis de todas las técnicas concebidas que brindan la posibilidad de resolver la problemática existente en la LPS Aplicativos SIG, donde fueron descritas sus principales características y desventajas presentadas en ellas, se decide hacer uso de los algoritmos de similitud de cadenas, específicamente el algoritmo de Levenshtein. En el próximo capítulo se abordará como será utilizado dicho algoritmo, así como sus adecuaciones para darle solución al problema planteado.

1.4. Metodología de desarrollo

El uso de una metodología es fundamental para guiar el desarrollo de un sistema informático. En dependencia de las características de un determinado proyecto, la correcta selección de la metodología

puede representar el éxito o el fracaso de dicho proyecto. A continuación se especifica la selección de la metodología de desarrollo seleccionada para guiar la presente investigación, así como su justificación.

“Una metodología de ingeniería del software es un proceso para producir software de forma organizada, empleando una colección de técnicas y convenciones de notación predefinidas” (Rumbaugh, 1999).

El sistema a implementar es un proyecto de poco volumen. Es necesario tener en cuenta que el mismo debe ser desarrollado en poco tiempo debido a la necesidad real existente en la LPS Aplicativos SIG y que el equipo de desarrollo es pequeño. Asimismo la documentación generada será solo la necesaria y los procesos fácilmente adaptables por los desarrolladores, teniendo en cuenta las condiciones y características existentes a lo largo del proceso de desarrollo. Tomando en consideración estas características es necesario seleccionar una metodología que se ajuste a las necesidades de la investigación.

Existen una serie de metodologías, diseñadas para guiar el proceso de desarrollo de un software. Podemos encontrar metodologías ágiles (MA), como XP (Extreme Programming), o tradicionales (MT) como RUP (Rational Unified Process). Las MA son apropiadas para guiar proyectos de poco volumen que requieran una rápida implementación. Por su parte, las MT son más apropiadas para proyectos grandes que por su importancia requieren una fuerte planificación.

El autor de la presente investigación decide no seleccionar la metodología de desarrollo ágil XP, pues, como principal característica presenta que el usuario final debe formar parte del equipo de desarrollo. A pesar de que los desarrolladores de la LPS se encuentran en constante interacción con el equipo de desarrollo de la solución, no forman parte de la implementación de la aplicación, no existe una completa integración entre estos. La metodología RUP también se descarta debido a que no se cuenta con el tiempo suficiente para su empleo, el ciclo de vida de esta puede extenderse hasta un año o más.

Como metodología de desarrollo se seleccionó Agile UP (AUP), la cual combina características de la metodología ágil XP con los artefactos de RUP. Esta metodología es una versión simplificada de RUP y describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP.

A continuación, en la fig. 1.2 se muestra el ciclo de vida de la metodología AUP.

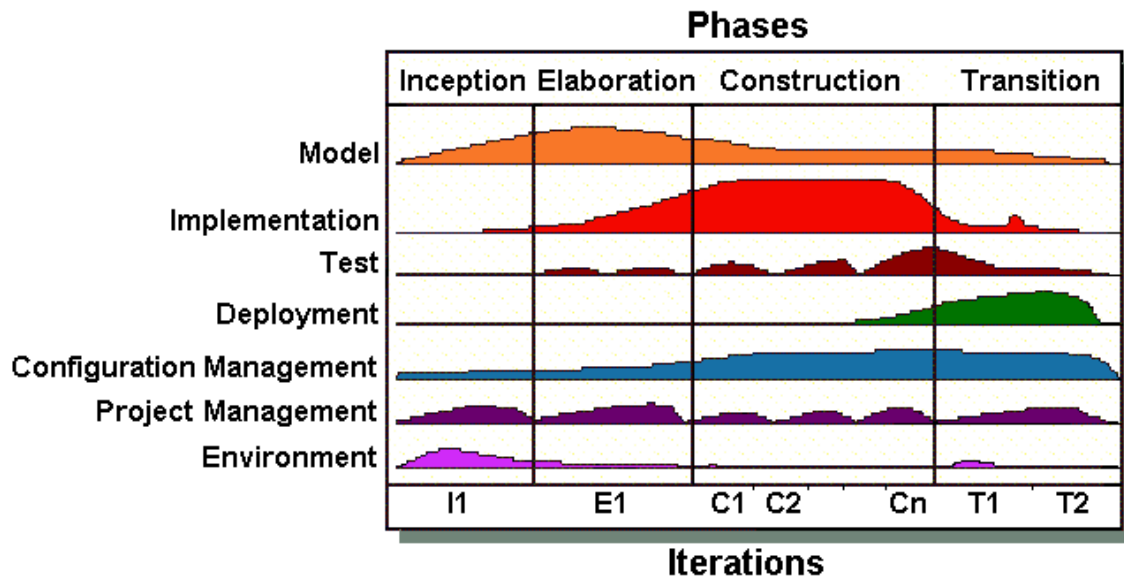


Fig. 1.2 Ciclo de vida de AUP (Universidad Nacional de Costa Rica, 2006)

1.5. Lenguaje de Modelado. UML v2.0.

Realizar el modelado de un sistema es fundamental para lograr una correcta estructuración del mismo. El modelado permitirá la visualización, especificación, construcción y documentación de los artefactos generados, creando una idea inicial de cómo quedará estructurado el proyecto. Para la realización de este es necesario el uso de un lenguaje de modelado. A continuación se muestra la selección del lenguaje de modelado y su justificación referente a la presente investigación.

Lenguaje Unificado de Modelado (UML por sus siglas en inglés) “es un lenguaje de modelado. Un modelo es una simplificación de la realidad. El objetivo del modelado de un sistema es capturar las partes esenciales del sistema. Para facilitar este modelado, se realiza una abstracción y se plasma en una notación gráfica. Esto se conoce como modelado visual” (Hernández, 2002). A partir de la confección de una serie de diagramas, permite una mayor abstracción de la situación que se presenta. Como lenguaje de modelado se escogió UML 2.0. Esta versión hace el lenguaje de modelado mucho más extensible de lo que era en su versión original, además, permite la validación y ejecución de modelos creados mediante el UML.

Este lenguaje permitió el modelado de diferentes diagramas fundamentales en el desarrollo de la presente investigación, como son: el diagrama de dominio (ver epígrafe 1.2), diagrama de casos de uso del sistema (ver epígrafe 2.4), diagrama de clases del diseño (ver epígrafe 3.1), entre otros.

1.6. Herramienta CASE: Visual Paradigm for UML 8.0 Enterprise Edition

Visual Paradigm Enterprise Edition para UML es una herramienta para desarrollo de aplicaciones utilizando modelado UML. Permite modelar tanto los procesos del negocio, la BD y las clases del sistema de manera visual. Esta herramienta se utiliza para la representación del lenguaje de modelado seleccionado. La misma permitirá realizar los diferentes diagramas y diseños de forma visual, permitiendo mayor facilidad en su confección y un mayor entendimiento de los mismos.

“Visual Paradigm facilita la creación de software y sistemas que se destacan en la experiencia del usuario mediante el apoyo a la identificación el uso eficaz de los casos, los requisitos de la recopilación, el flujo de los acontecimientos, requisito generación de la especificación, etc.” (Visual Paradigm, 2014).

“Visual Paradigm Enterprise Edition es una herramienta de desarrollo de software y sistema todo-en-uno para el modelado de sistemas de tecnología de la información de extremo a extremo” (Visual Paradigm, 2014).

1.7. Lenguaje de programación Java.

Para el desarrollo de un sistema informático es necesario el uso de un lenguaje de programación. A continuación se muestra la selección y justificación del lenguaje de programación mediante el cual se desarrollará la solución de la presente investigación.

Como lenguaje de programación se seleccionó Java, el cual es un lenguaje robusto y potente. Asimismo el equipo de desarrollo lo conoce bien gracias a la experiencia acumulada con la que cuenta debido a las asignaturas curriculares de la carrera. El uso de dicho lenguaje puede ser ventajoso, ya que, si es necesario realizar en un futuro, modificaciones o acciones de mantenimiento al sistema desarrollado, el equipo perteneciente a la facultad, estará familiarizado con el lenguaje de programación. Este deriva su sintaxis de los lenguajes C y C++, y sus aplicaciones son compiladas a bytecode, que puede ejecutarse

en cualquier máquina virtual de Java sin importar la arquitectura de la computadora donde radique. En la UCI existe también un amplio grupo de desarrolladores, los cuales se encuentran especializados en este lenguaje, facilitando solventar las dudas existentes durante el desarrollo de la aplicación.

1.8. IDE NetBeans 7.3

NetBeans es un entorno de desarrollo integrado (IDE) hecho principalmente para el lenguaje de programación Java. Este IDE es libre y gratuito, sin restricciones de uso, lo cual representa una ventaja ya que no es necesario tener una licencia para su uso. El NetBeans es asimismo, el IDE con el que se imparten las asignaturas curriculares de programación en la Facultad 6, por lo que podemos encontrar gran cantidad de profesores que esten vinculados con su uso. Este IDE presenta una librería extensa, así como varias características de utilidad que facilitan el trabajo con el mismo, como son: Auto completamiento de código, visor de clases, métodos y componentes, ayuda incluida. Además permite la confección de formularios de forma visual y con una estética acorde a los estándares de diseño de software.

1.9. Conclusiones del capítulo

Del estudio realizado sobre el alineamiento de información en BD espaciales, conceptos, características, y el análisis minucioso de las diferentes técnicas que existen actualmente en el mercado, con el propósito de identificar posibles soluciones, concluimos:

El estudio y alineación de información en las BD espaciales es de vital importancia para lograr una uniformidad en las mismas, eliminando las irregularidades en la información manejada y garantizando BD más confiables para la LPS Aplicativo SIG. A partir del análisis de las posibles vías para solucionar el problema que se presenta, se demostró que el uso de una ontología, no es el camino más indicado a seguir para suplir las necesidades de la LPS. A partir del análisis de investigaciones y soluciones similares al tema tratado, se pudo constatar que ninguno da solución al problema planteado, aunque si aportan características y elementos necesarios para el desarrollo del mismo. Con el estudio de las diferentes técnicas que permiten resolver el problema presentado, se demuestra que la vía más eficiente para darle solución es a través del uso de algoritmos de comparaciones de cadena y el algoritmo que más se ajusta

a las necesidades de la LPS es el Algoritmo de Levenshtein. Dadas las características del sistema que se desea implementar, se concluye que la metodología de desarrollo más adecuada para guiar el proceso es AUP. Finalmente se determinó que los lenguajes y herramientas seleccionadas, necesarias para la confección de diagramas, clases e interfaces del sistema, fueron los más adecuados para la solución a desarrollar.

CAPÍTULO 2: EL PROCESO DE ALINEAMIENTO DE INFORMACIÓN EN BASES DE DATOS ESPACIALES.

En este capítulo se realiza una descripción del dominio, así como de los requisitos funcionales y no funcionales con los que debe cumplir la aplicación. Se define además la arquitectura con la que se estructurará el sistema. Finalmente se presenta el algoritmo de comparación de cadenas, cuya implementación será utilizada como parte de la solución. Se plantean las bases para la construcción del sistema y se presenta una propuesta de solución para lograr el alineamiento de la información de las BD espaciales de la LPS Aplicativos SIG.

2.1. Diagrama del dominio

“El Modelo de Dominio es el diagrama que describe los tipos o eventos más importantes del entorno en que se desarrolla el sistema, en un ambiente de constante cambio” (Entorno Virtual de Aprendizaje, 2007-2008).

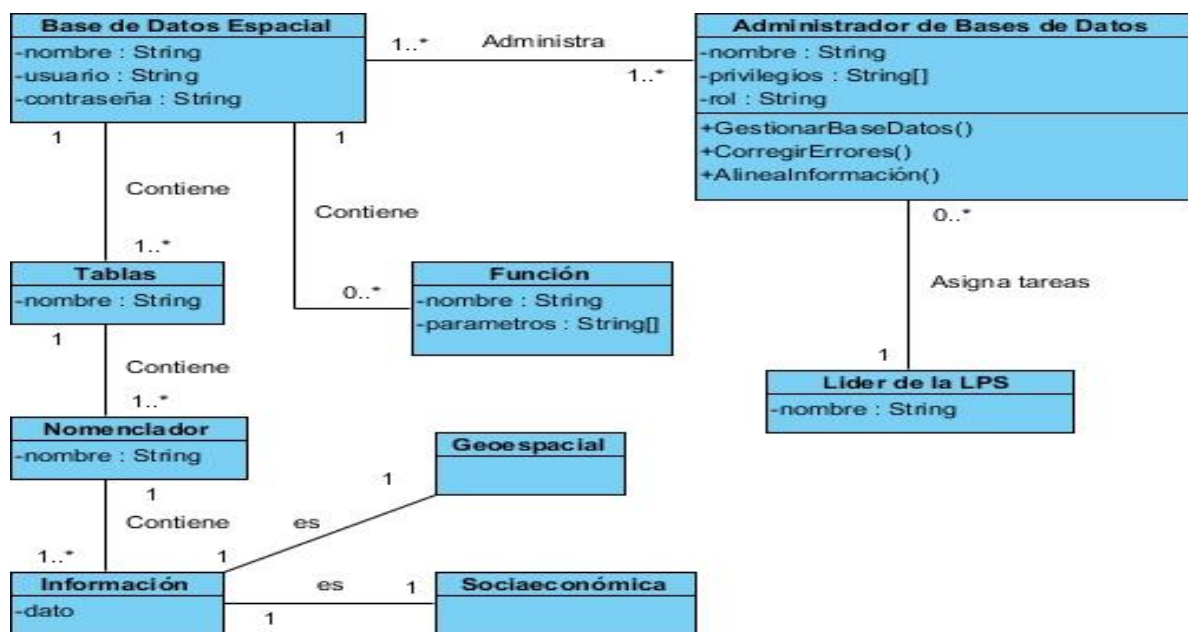


Fig. 2.1 Diagrama de dominio.

En la fig. 2.1, se identifican las entidades que están presentes en el negocio y la relación entre ellas. La clase *Base de Datos Espacial*, representa una BD de la LPS Aplicativos SIG y es en sí, la entidad más general del diagrama. A su vez, las clases *Tablas* y *Función*, representan las tablas y las funciones que componen las BD. La clase *Nomenclador*, representa los nomencladores que se encuentran almacenados en las BD, los cuales hacen referencia a diferentes elementos, ya sean provincias, municipios, ciudades, entre otros. Asimismo, la clase *Información*, representa toda la información almacenada, la cual puede ser de tipo geoespacial o socioeconómica, representadas en las clases *Geoespacial* y *Socioeconómica* respectivamente. La clase *Administrador de Base de Datos*, refiere al personal encargado de administrar las BD, los cuales pertenecen a la factoría Bases de Datos Espaciales y Cartografías, de la LPS Aplicativos SIG. Asimismo, la clase *Líder de la LPS*, refiere al Líder de la LPS Aplicativos SIG.

Cada BD tiene varias tablas y funciones. Cada tabla se encuentra identificada con varios elementos que son a su vez nomencladores. Cada nomenclador contiene información, que puede ser geoespacial o socioeconómica. A su vez, las BD espaciales son administradas por el Administrador de BD, el cual ejecuta las tareas asignadas por el Líder de la LPS.

A partir del análisis realizado en las BD existentes en la LPS Aplicativos SIG, se identificaron dos elementos en común. Estos elementos a su vez fueron descritos como nomencladores predefinidos por el sistema para el alineamiento de la información. Dichos nomencladores se encuentran relacionados en la tabla 2.1.

Tabla 2.1 Nomencladores predefinidos para el alineamiento de información.

Nombre	Descripción
Provincia	Contiene todos los valores pertenecientes a las provincias de la actual división político administrativa de la República de Cuba.
Municipio	Contiene todos los valores pertenecientes a los municipios de la actual división político administrativa de la República de Cuba.

Inicialmente se definen solamente estos nomencladores. Estos fueron identificados en conjunto con los administradores de las BD. No serán los únicos existentes, el usuario podrá definir otros nomencladores con sus respectivos valores. Provincia y Municipio, así como sus valores asociados se mantendrán invariables.

2.2. Requisitos Funcionales

Para lograr una elevada calidad en el sistema es necesario dar solución a las necesidades del interesado, para ello se definen una serie de requisitos que la aplicación debe cumplir.

La corporación Synergix⁹, en su página “Tecnología y Synergix” expresa que *“Cuando hablamos de una característica requerida de la cual se sabe que va a ser satisfecha por medio de la adición de un subsistema o bloque de código en el software, entonces se dice que estamos ante un requisito funcional, por cuanto es un requisito que denota una funcionalidad del sistema”* (Synergix, 2011).

“Ingeniería de Requerimientos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software” (Pressman, 2006).

El levantamiento de requisitos es la actividad que permite definir las necesidades del negocio, así como las funcionalidades a implementar. Describen los escenarios con los cuales interactúan los usuarios. Un correcto levantamiento de requisitos permitirá posteriormente realizar un correcto levantamiento de las funcionalidades con las que debe contar el sistema.

A partir del estudio del problema y la entrevista realizada al grupo de desarrollo de base de datos espaciales y cartografías, se identificaron los siguientes requisitos funcionales con los que debe cumplir la aplicación, los cuales se muestran en la tabla 2.2:

Tabla 2.2 Definición de Requisitos Funcionales del Sistema.

No	Nombre	Descripción
----	--------	-------------

⁹ Compañía venezolana dedicada a la implantación, puesta en punto y soporte de Sistemas de Planificación de Recursos Empresariales (ERP por sus siglas en inglés) en pequeñas y grandes empresas, así como al desarrollo de soluciones de Tecnología de la Información.

RF1	Realizar conexión a BD	Permite realizar la conexión entre el sistema y las BD almacenadas en un servidor de BD.
RF2	Realizar desconexión de BD	Permite realizar la desconexión entre el sistema y las BD que se encuentren conectadas al mismo.
RF3	Guardar información de la conexión	Permite guardar todos los datos referentes a la conexión realizada.
RF4	Calcular el grado de similitud entre dos cadenas de caracteres	Permite calcular el grado de similitud entre dos cadenas de caracteres a partir de la utilización de un algoritmo de comparación de cadenas.
RF5	Mostrar información de las tablas de la BD conectadas	Permite mostrar la información de las tablas contenidas en la BD conectada.
RF6	Adicionar nuevo nomenclador	Permite adicionar un nuevo nomenclador a la lista de nomencladores existentes.
RF7	Eliminar nomenclador	Permite eliminar un nomenclador de la lista de nomencladores que han sido creados por el usuario.
RF8	Modificar nomenclador	Permite modificar tanto el nombre, como cualquiera de los valores correspondientes a dicho nomenclador.
RF9	Calcular lista de errores	Permite listar todos los errores existentes en los nomencladores de determinada tabla o campo determinado por el usuario.
RF10	Corregir errores	Permite corregir todos los nomencladores identificados previamente como incorrectos, por los seleccionados como correctos.
RF11	Guardar estado original de la BD	Permite guardar el estado original de la BD conectada.
RF12	Restaurar BD	Permite restaurar la BD conectada a su estado original, sin ningún cambio realizado.

RF13	Guardar cambios realizados	Permite modificar la BD con todos los cambios realizados por el usuario.
RF14	Calcular lista de alineación	Permite realizar una lista con el resultado de la alineación de las dos tablas y posteriormente mostrarla.
RF15	Alinear información	Permite unir dos tablas seleccionadas por el usuario, por determinado campo seleccionado por el usuario.

2.3. Requisitos no Funcionales

Los Requisitos No Funcionales juegan un papel muy importante en el éxito de un proyecto, ya que estos forman una parte significativa en la especificación del sistema. Son cualidades que la aplicación debe cumplir.

2.3.1. RNF 1. Usabilidad e Interfaces

El sistema cuenta con una apariencia profesional y un entorno gráfico sencillo. Se emplean alertas y mensajes de fácil entendimiento cada vez que sea requerido, dándole la respuesta necesaria a cada acción realizada por el usuario. El tamaño de formularios, mensajes y botones no son sobredimensionados o demasiado pequeños. El funcionamiento del sistema está orientado a guiar al usuario, de forma que las funcionalidades se realicen en el orden necesario, para de esta forma, cumplir con los estándares internacionales, específicamente el ISO 9126 para la evaluación de la calidad de software.

2.3.2. RNF 2. Seguridad

En base a la seguridad del sistema, para establecer la conexión a la BD, será necesario conocer el usuario y la contraseña de la misma. Asimismo la aplicación no permitirá insertar datos incorrectos y los nomencladores a utilizar serán definidos previamente por los usuarios administradores de las BDs y posteriormente, podrán ser ingresados nuevos nomencladores por el propio usuario. El sistema permitirá

restaurar la información de la BD conectada a su estado original permitiendo deshacer todos los cambios realizados en la misma.

2.3.3. RNF 3. Rendimiento

El tiempo de respuesta estará dado fundamentalmente por el tiempo que demore el algoritmo de comparación de cadenas (ver epígrafe 2.4) en comparar todos los nomencladores definidos en el sistema con los de las BDs. Mientras más campos tengan las tablas de las BD, mayor será el tiempo de respuesta.

2.3.4. RNF 4. Software

La estación de trabajo donde se implantará el sistema debe contar con la Máquina Virtual de Java (JVM por sus siglas en inglés) para poder ejecutar el sistema.

2.3.5. RNF 5. Hardware

De acuerdo con la página “*TecnoProgramas*” (Oracle, 2014), Oracle recomienda como requerimientos mínimos para el uso de la JVM:

- Procesador 166 MHz o superior.
- Memoria RAM física 64 MB o superior.
- Espacio libre en disco superior a 98 MB.

Teniendo en cuenta estas recomendaciones y las demás prestaciones que debe tener la estación de trabajo donde se pretende ejecutar el sistema, resultado de la presente investigación, se recomienda que esta cuente con los siguientes requerimientos:

- Memoria RAM superior o igual a los 512 MB.
- Procesador 256 MHz como mínimo.
- Espacio libre en disco de 1 Gb como mínimo.

Estas prestaciones deben estar en correspondencia con las necesidades mínimas para la configuración de la estación de trabajo, dígame: sistema operativo instalado y antivirus, así como las actualizaciones respectivas para cada uno de estos y demás configuraciones.

Una vez determinados los requisitos funcionales, estos fueron agrupados en serie de casos de uso del sistema, los cuales representan las funcionalidades con las que debe contar el sistema. A continuación se muestra la especificación de los casos de uso identificados, la descripción de cada uno de ellos y su relación con los requisitos funcionales.

2.4. Casos de Uso

“Un modelo de caso de uso (CU) describe el comportamiento del sistema ante la petición de un usuario, estos usuarios son también conocidos como actores. Los actores son las personas que utilizarán el sistema en correlación a las acciones que se describen (Pressman, 2003).”

Actor	Descripción
Administrador de la BD	Son los encargados de administrar las BD en la LPS. Son los únicos que tienen acceso a las BDs, ya que poseen usuario y contraseña para acceder a las mismas.

A continuación se muestra en la fig. 2.2, el diagrama de CU del sistema y posteriormente la descripción de los mismos.

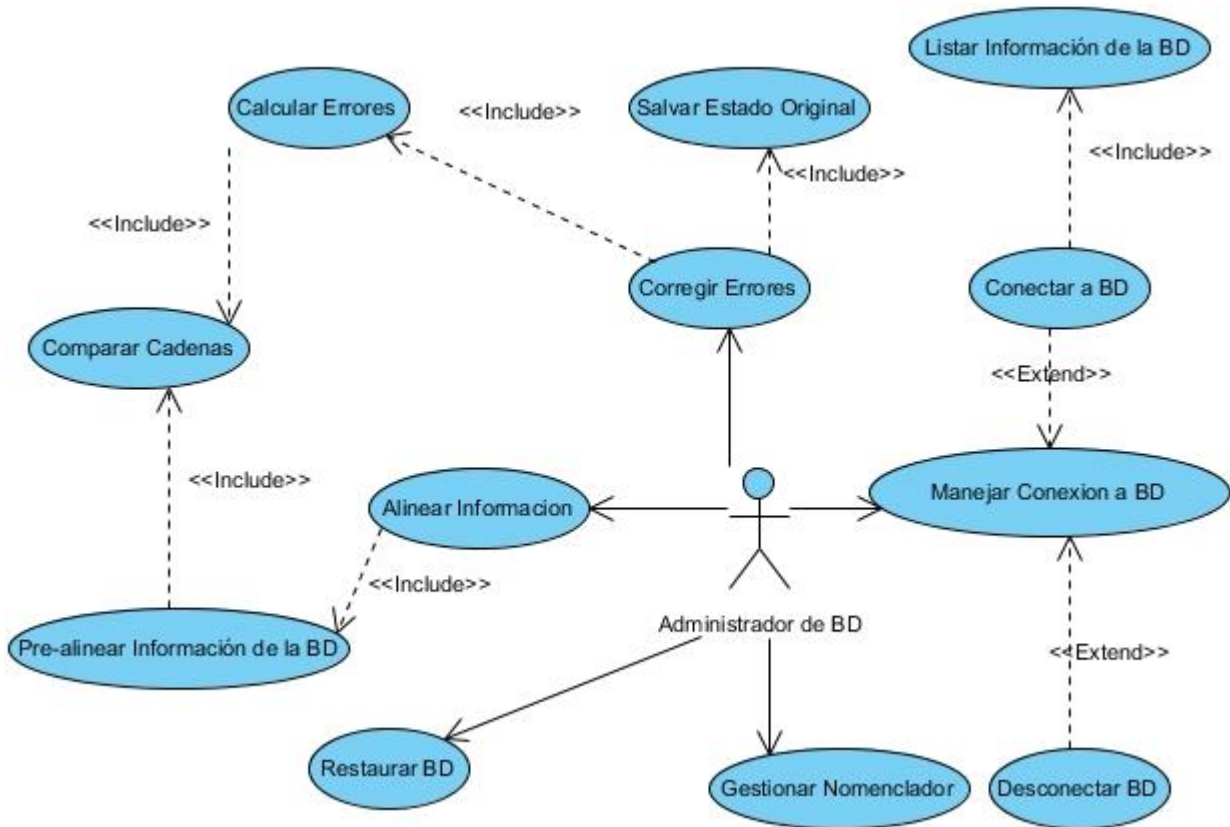


Fig. 2.2 Diagrama de Casos de Uso del Sistema.

2.4.1. Descripción de Casos de Uso arquitectónicamente significativos

“Los casos de uso Arquitectónicamente Significativos son aquellos que representan las partes más críticas de la arquitectura del sistema y demuestran la funcionalidad del sistema” (Ugaz Horna, 2008).

Tabla 2.3 Descripción CU Calcular Errores

Caso de Uso	Calcular Errores
Actor	Administrador de BD
Propósito	Definir los nomencladores representados incorrectamente.
Resumen:	El caso de uso se inicia cuando el administrador de la BD accede al menú del sistema, a la opción de “Corregir Errores”, el sistema identifica todos los nomencladores representados

	incorrectamente y propone una representación correcta en relación con los nomencladores existentes identificados previamente por el usuario.	
Precondiciones	La conexión con la BD debe estar establecida.	
Referencias	RF9	
Prioridad	Crítico	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	<p>1- Selecciona la opción “Corregir Errores” del menú “Trabajo”.</p> <p>3- Selecciona la acción a realizar:</p> <p style="margin-left: 20px;">a) Clic en el botón “Cerrar”, ir al Flujo Alterno¹⁰ “Cerrar”.</p> <p style="margin-left: 20px;">b) Clic en el botón “Buscar Errores” ir al paso 6.</p> <p style="margin-left: 20px;">c) Selecciona la conexión y da clic en el botón “Cargar”.</p> <p>5- Selecciona el campo del que se desea buscar los errores y da clic en el botón “Buscar Errores”.</p>	<p>2- Muestra la interfaz de “Corregir Errores”.</p> <p>4- Carga la configuración de la BD y se muestran los campos al usuario.</p> <p>6- Verifica que haya un campo seleccionado:</p> <p style="margin-left: 20px;">a) En caso de no haberlo, ir al FA “Error de Selección”.</p> <p style="margin-left: 20px;">b) Calcula los errores y los muestra. Habilita el botón “Corregir”.</p> <p>7- Termina el CU.</p>
Flujo Alterno: Cerrar		
	Acción del Actor	Respuesta del Sistema
		<p>1- El sistema cierra la Interfaz.</p> <p>2- Ir al paso 7 del Flujo Normal¹¹.</p>
Flujo Alterno: Error de Selección		
		<p>1- El sistema muestra un mensaje de error, alertando que no hay ningún campo seleccionado.</p> <p>2- Ir al paso 3 del FN.</p>
Prototipo de Interfaz		

¹⁰ Flujo Alterno (FA)

¹¹ Flujo Normal (FN)

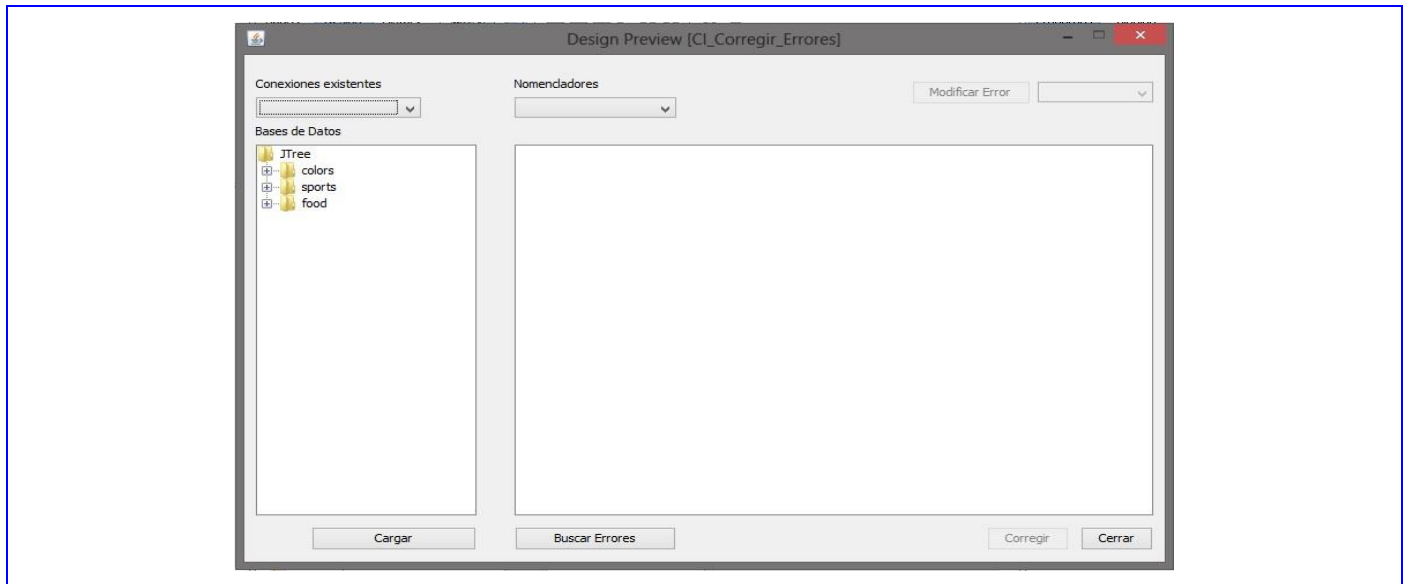


Tabla 2.4 Descripción CU Corregir Errores

Caso de Uso	Corregir Errores	
Actor	Administrador de BD	
Propósito	Corregir los nomencladores identificados como incorrectos.	
Resumen:	El caso de uso se inicia cuando el administrador de la BD acciona el botón “Corregir Errores” del Formulario “Corregir Errores”, una vez identificados los nomencladores incorrectos, el sistema los modifica por la opción seleccionada como correcta.	
Precondiciones	Debe haberse ejecutado el CU: Calcular Errores.	
Referencias	RF10	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	

<p>1- Revisa la información y selecciona la acción a realizar:</p> <p>a) Modifica uno de los errores de la lista, ir al flujo alterno “Modificar Error”.</p> <p>b) Da clic en el botón “Corregir”.</p>	<p>2- Corrige los errores y muestra un mensaje al Administrador de la BD.</p> <p>3- Termina el CU.</p>
--	--

Flujo Alternativo: Modificar Error

Acción del Actor	Respuesta del Sistema
<p>1- Selecciona el error a modificar.</p> <p>2- Selecciona el nuevo valor del nomenclador.</p> <p>3- Da clic en el botón “Modificar Error”.</p>	<p>4- El sistema modifica el error seleccionado y actualiza la lista de errores.</p> <p>5- Ir al paso 1 del FN.</p>

Prototipo de Interfaz

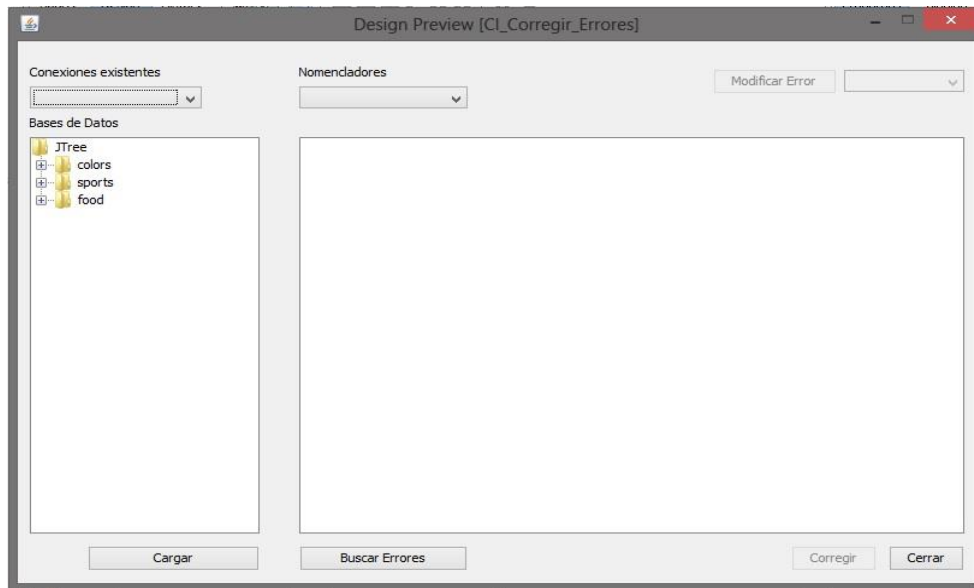
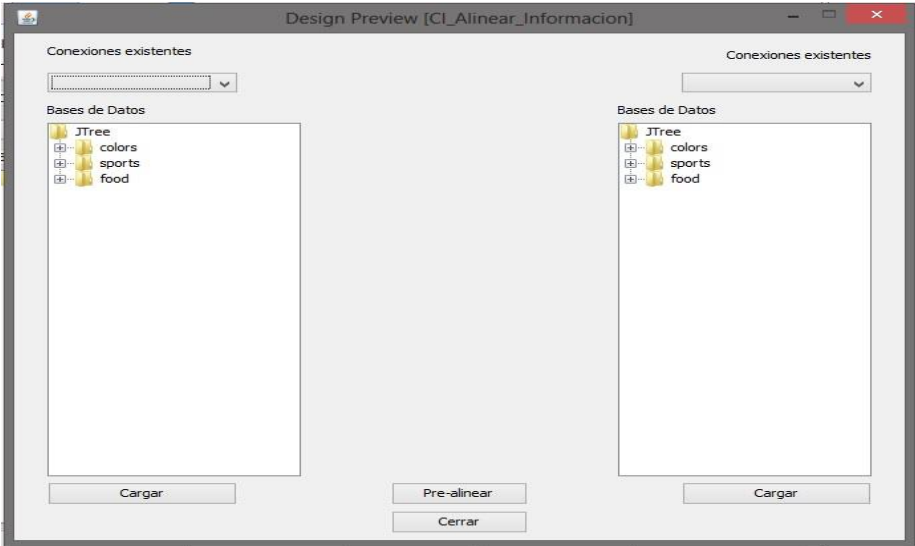


Tabla 2.5 Descripción CU Alinear Información

Caso de Uso	Alinear Información
Actor	Administrador de BD
Propósito	Unir dos tablas por determinado parámetro.

Resumen:	El caso de uso se inicia cuando el administrador de la BD accede a la opción “Alinear Información”. El sistema permite seleccionar al usuario las tablas a alinear de las BD conectadas, así como el campo por el que desea unir estas tablas, posteriormente muestra una lista de cómo quedaría y si el usuario acepta el cambio, se realiza la unión por el parámetro seleccionado.	
Precondiciones	La conexión con la BD debe estar establecida.	
Referencias	RF14, RF15	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
<p>1- Selecciona la opción “Alinear Información” del menú “Trabajo”.</p> <p>3- Selecciona la acción a seguir:</p> <p style="margin-left: 20px;">a) Seleccionar la conexión y dar clic en los botones “Cargar”.</p> <p style="margin-left: 20px;">b) Dar clic en el botón “Pre Alinear”, ir al paso 6 del FN.</p> <p style="margin-left: 20px;">c) Dar clic en el botón “Cerrar”, ir al FA “Cerrar”.</p> <p>5- Selecciona los campos a alinear y da clic en “Pre Alinear”.</p> <p>7- Selecciona la acción a realizar:</p> <p style="margin-left: 20px;">a) Da clic en el botón “Cerrar”, ir al FA “Cancelar Alineación”.</p> <p style="margin-left: 20px;">b) Dar clic en el botón “Alinear”.</p> <p style="margin-left: 20px;">c) Modifica un valor de alineación, ir al FA “Modificar Valor de Alineación”.</p>	<p>2- Muestra la Interfaz Alinear Información.</p> <p>4- Carga las configuraciones de las BD y muestra los datos.</p> <p>6- Verifica que los valores estén seleccionados:</p> <p style="margin-left: 20px;">a) En caso de no estarlo, ir al FA “Error de Selección”.</p> <p style="margin-left: 20px;">b) En caso de estarlo, muestra la interfaz Pre Alinear con los datos a alinear.</p> <p>8- Alinea la información e informa al usuario.</p> <p>9-Ir al paso 3.</p>	
Flujo Alterno: Cerrar		
Acción del Actor	Respuesta del Sistema	
	<p>1- Cierra la interfaz.</p> <p>2- Termina el CU.</p>	
Flujo Alterno: Error de Selección		
	1- Muestra un mensaje de error.	

	2- Ir al paso 3 del FN.
Flujo Alterno: Cancelar Alineación	
	1- Cierra la ventana de "Pre Alineación". 2- Ir al paso 3 del FN.
Flujo Alterno: Modificar Valor de Alineación	
1- Selecciona un valor de la lista y el nuevo valor que se va a cambiar. 2- Dar clic en el botón "Modificar Valor".	3- Actualiza el valor. 4- Ir al paso 7 del FN.
Prototipo de Interfaz	
	

Se le recomienda al lector para un mejor entendimiento del negocio, remitirse al expediente de proyecto, donde se encuentran el resto de las descripciones de los casos de usos del sistema.

Para el desarrollo de la aplicación, es necesario definir una arquitectura de desarrollo de software, así como las herramientas a utilizar con el fin de organizar y controlar la evolución del sistema. El seguimiento de una buena arquitectura, no solo garantizará la estética de la presente investigación, sino que también permite que futuros analistas o desarrolladores que pretendan hacer cambios o realizar acciones de mantenimiento sobre la aplicación, puedan entender el código y funcionamiento de la misma.

2.5. Arquitectura

Según Barry Boehm, 1995: *“Si un proyecto no ha logrado una arquitectura del sistema, incluyendo su justificación, el proyecto no debe empezar el desarrollo en gran escala. Si se especifica la arquitectura como un elemento a entregar, se la puede usar a lo largo de los procesos de desarrollo y mantenimiento”* (Boehm, 1995).

Se decidió utilizar un patrón arquitectónico de tipo N-capas. Los sistemas basados en capas, están organizados jerárquicamente en una o varias capas, donde cada una provee servicios a la superior y es servida por la inferior. Esto permite una mejor descomposición del problema, lo cual, en caso de realizar cambios, estos afectarían solamente a la capa en la que se aplicaron los mismos, por lo que el futuro mantenimiento y/o actualización de la aplicación sería más sencillo. Para el desarrollo de la presente investigación se determinó realizar una capa de presentación, capa de lógica de negocio y una capa de acceso a datos.

De acuerdo con el “Departamento de Lenguajes y Sistemas Informáticos” de la Universidad de Alicante: *“La Capa de Presentación contiene la interfaz de usuario, a través del cual interacciona con la aplicación. La Capa de Lógica de Negocio, contiene el núcleo de la aplicación, y la implementación de las funcionalidades de la aplicación. Finalmente en la Capa de Datos, es donde se gestiona la persistencia de los datos”* (DLSI, 2004-2005). Teniendo en cuenta lo planteado anteriormente se agruparon las diferentes clases (Ver epígrafe 3.1) que componen el sistema en las capas relacionadas a continuación.

Capa de Presentación: Se agrupan las clases que constituyen las interfaces del sistema con las que va a interactuar el usuario. Son las clases donde se recogen los datos y se invocan a las clases de la capa de lógica de negocio.

Capa de Lógica de Negocio: Se agrupan las clases donde se implementan las funcionalidades con las que debe contar el sistema. Estas clases son las encargadas de invocar a las clases pertenecientes a la capa de acceso a datos.

Capa de Acceso a Datos: Se agrupan las clases encargadas de salvar los datos, recogidos o modificados por el sistema.

En la fig. 2.3 se muestra la comunicación entre las tres capas.

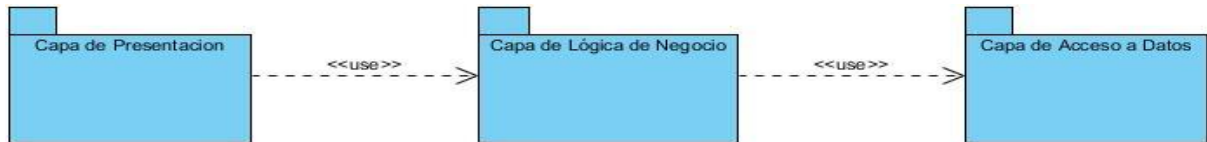


Fig. 2.3 Capas del patrón arquitectónico del sistema

Un estilo arquitectónico: *“Expresa componentes y las relaciones entre estos, con las restricciones de su aplicación y la composición asociada, así como también las reglas para su construcción. Se consideran como un tipo particular de estructura fundamental para un sistema de software, conjuntamente con un método asociado que especifica cómo construirlo incluyendo información acerca de cuándo usar la arquitectura que describe, sus invariantes y especializaciones, así como las consecuencias de su aplicación”* (Camacho, 2004).

Para la construcción de componentes, así como las relaciones entre ellos, se utiliza el estilo arquitectónico Llamada y Retorno. Este estilo *“Permite construir una estructura de programa relativamente fácil de modificar y ajustar. Esta familia de estilos enfatiza la modificabilidad y la escalabilidad. Son los estilos más generalizados en sistemas en gran escala. Miembros de la familia son las arquitecturas de programa principal y subrutina, los sistemas basados en llamadas a procedimientos remotos, los sistemas orientados a objeto y los sistemas jerárquicos en capas”* (Reynoso, 2004).

2.6. Algoritmo de similitud de cadenas de Levenshtein

Después de haber realizado el análisis de los diferentes algoritmos de similitud en el capítulo 1, se decidió seleccionar el algoritmo de Levenshtein, pues permite identificar cuán semejantes son dos cadenas de caracteres, pertenecientes a dos listas diferentes. Esta característica permitiría seleccionar los valores asociados a los nomencladores que son similares dentro de tablas diferentes, lo cual facilitaría identificar y solucionar los problemas de irregularidad en la información de las BD de la LPS Aplicativos SIG.

Este algoritmo, a diferencia del de Boyer-Moore, permite solucionar el problema que se presenta, ya que permite comparar dos cadenas de caracteres dadas e identificar cuan semejantes son. Su complejidad

temporal, es la menor posible para un algoritmo con esta característica. Asimismo, su documentación es más extensa en cuanto a implementación y funcionamiento refiere, que el algoritmo de Ratcliff-Obershelp. Estas razones fueron las fundamentales en la selección de este algoritmo como vía de solución al problema tratado en la presente investigación.

A continuación se presenta el pseudo-código de dicho algoritmo con el fin de lograr un mejor entendimiento de su funcionamiento.

```
(1)  int DistanciaLevenshtein (char str1[1..lenStr1], char str2[1..lenStr2])
(2)  declare int d[0..lenStr1, 0..lenStr2] // d tabla con longitudStr1+1
                                           filas y longitudStr2+1 columnas
(3)  declare int i, j, costo
(4)  for i from 0 to lenStr1
(5)  d[i, 0] := i
(6)  for j from 0 to lenStr2
(7)  d[0, j] := j
(8)  for i from 1 to lenStr1
(9)  for j from 1 to lenStr2
(10) if str1[i] = str2[j] then costo: = 0
(11) else costo: = 1
(12) d[i, j] := minimum(d[i-1, j] + 1, d[i, j-1] + 1, d[i-1, j-1] + costo)
(13) return d[lenStr1, lenStr2]
```

Línea 2. Creación de un arreglo bidimensional de enteros, donde se almacenará en valores de 0 y 1, la igualdad entre los caracteres en una misma posición, de los arreglos recibidos por parámetros, siendo 0 en caso de no ser iguales y 1 en caso de que sean iguales.

Línea 3. Declaración de tres variables enteras, **i** y **j** para los ciclos **for**, y **costo** para almacenar el valor de similitud entre los caracteres de una misma posición.

Líneas 4 - 7. Inserción dentro del arreglo, de numeración de cantidad de filas y cantidad de columnas.

Líneas 8 - 12. Comparación entre los elementos de ambos arreglos de caracteres. Asignación al arreglo de enteros, del valor de similitud entre los caracteres comparados, a través del método **mínimum**, el cual retorna el menor de los valores pasados por parámetro.

Línea 13. Retorno del valor final de similitud entre las dos cadenas de caracteres a comparar.

2.7. Conclusiones del capítulo

Al finalizar este capítulo, después de abordar la justificación de la fundamentación de la propuesta descrita por el autor, así como, haber definido todos los parámetros necesarios para lograr un correcto análisis y diseño de la solución, podemos concluir que:

La propuesta de solución permitirá realizar un correcto alineamiento de las información de las BD de la LPS de forma automática, haciéndola superior al compararla con las soluciones similares analizadas. A partir del estudio realizado en el presente capítulo, se lograron definir todos los parámetros necesarios para lograr el correcto desarrollo de la solución. La definición de una arquitectura permitió estructurar el sistema acorde a las necesidades y características del mismo. Con la especificación de los requisitos, se lograron definir todas las funcionalidades con las que debe cumplir el sistema, las cuales fueron agrupadas posteriormente en una serie de casos de uso para su mejor comprensión. Se seleccionó el algoritmo de comparación de cadena a utilizar, lo cual permite identificar los valores asociados a los nomencladores, que se encuentran identificados de manera incorrecta en las diferentes BD. De esta forma queda trazado el camino para la elaboración de la solución.

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA DE ALINEAMIENTO DE NOMENCLADORES EN BASES DE DATOS GEOESPACIALES.

En el presente capítulo se realiza una descripción de las clases con las que contará el sistema, así como la relación entre estas. Asimismo se describen todos los procesos asociados al despliegue del sistema y se le da un seguimiento a través de pruebas dando una valoración crítica a partir de los resultados obtenidos. Finalmente se realiza la validación del mismo.

3.1. Patrones de diseño

Los patrones de diseño son una serie de estrategias a seguir en la construcción de un software, que facilitan la descripción del problema y su solución. *“Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software”* (Tedeschi, 2014).

Estos patrones *“Describen el esquema básico para estructurar subsistemas y componentes”* (Pavón Mestras, 2004).

En la presente investigación se emplean Patrones de Principios Generales para Asignar Responsabilidad (GRASP), así como patrones Pandilla de Cuatro (GoF por sus siglas en inglés). Dentro de los GRASP, se utilizaron:

Experto en Información: Este patrón permite asignar la responsabilidad a la clase que contiene la información necesaria para llevar a cabo dicha responsabilidad. Este patrón puede verse evidenciado en la clase CE_Conexion, la cual es la encargada de realizar la conexión con la BD.

Creador: Permite crear una instancia de la clase que tiene la información necesaria para la creación del objeto. Este patrón permite almacenar o manejar varias instancias de dicha clase. Este patrón puede verse evidenciado en la clase CC_Principal, donde se crean instancias de las clases CC_Conexion y CC_Nomencladores.

Controlador: Permite asignar la responsabilidad de gestionar un mensaje de un evento del sistema a una clase que represente al sistema global, o un escenario de caso de uso donde tiene lugar dicho evento. El uso de este patrón puede verse evidenciado en la capa de Lógica de Negocio, donde la clase CC_Principal, le asigna la responsabilidad de realizar las tareas de gestionar nomencladores, a la clase CC_Nomencladores, y esta le retribuye dicha tarea a la clase CE_Nomenclador.

Alta Cohesión: Permite asignar responsabilidades a una clase de manera que la información que maneje dicha clase, sea solamente la necesaria. Su utilización se evidencia ya que las clases de la aplicación tienen una función bien definida dentro del sistema, por lo que cada una de ellas es responsable de las tareas relacionadas con esta. Por ejemplo, la clase CC_Nomenclador, solamente se encarga de las tareas relacionadas a los nomencladores.

Bajo Acoplamiento: Consiste en lograr un diseño de forma tal que se tenga la menor dependencia posible entre las clases, potenciando qué, en caso de ocurrir algún cambio, afecte en la menor medida posible al resto de las clases. Este patrón se evidencia en las clases interfaz de la aplicación. Cada clase recibe todos los objetos que necesitan ya creados y configurados, logrando así reducir las dependencias entre las clases de la aplicación en la creación de las clases.

De los patrones GoF, se emplearon:

Singleton: Consiste en la creación de una instancia de una clase como único punto de acceso a dicha clase. El patrón es empleado en las Capas de Presentación y Lógica de Negocio. Solo se crea una instancia de cada clases, y esta es la utilizada para acceder a las funcionalidades de cada clase. En la clase CI_Principal, solamente se crea una instancia de la clase CC_Principal, como único punto de acceso a dicha clase, de manera que, se utilice el mismo objeto durante la ejecución del sistema. Dicho objeto será enviado a las demás clases de interfaz del sistema.

Observer: Consiste en la definición de una dependencia de uno a muchos entre diferentes objetos, de manera que, si cambia el estado del objeto observado, sean notificados los elementos asociados a este objeto. Este patrón puede verse evidenciado en los eventos relacionados a las interfaces del sistema. En la clase CI_Gestionar_Nomenclador, al dar clic sobre uno de los nomencladores existentes en el sistema, se muestra automáticamente, una lista con los valores asociados a dicho nomenclador.

3.2. Diagrama de clases de diseño

“El Diagrama de Clases de Diseño describe gráficamente las especificaciones de las Clases de Software y las Interfaces de una aplicación. Contiene información relacionada con clases, asociaciones y atributos, interfaces, con sus operaciones y constantes, información sobre los tipos de atributos, navegabilidad y dependencia.” (Bustos Thames, 2011).

En el presente diagrama se puede observar el uso del estilo arquitectónico llamada-retorno, específicamente el patrón arquitectónico en capas. Se puede evidenciar la existencia de tres capas (Presentación, Lógica de Negocio y Acceso a Datos). En la Capa de Lógica de Negocio se emplean los patrones GRASP (Experto en Información, Creador, Controlador, Alta Cohesión y Bajo Acoplamiento), así como los GoF (Singleton, Observer), definidos en el epígrafe 3.1.

A continuación se presenta en la fig. 3.1 el diagrama de clases de diseño, en el cual se describe la estructura del sistema, definiendo sus clases e interfaces, así como la relación entre ellas.

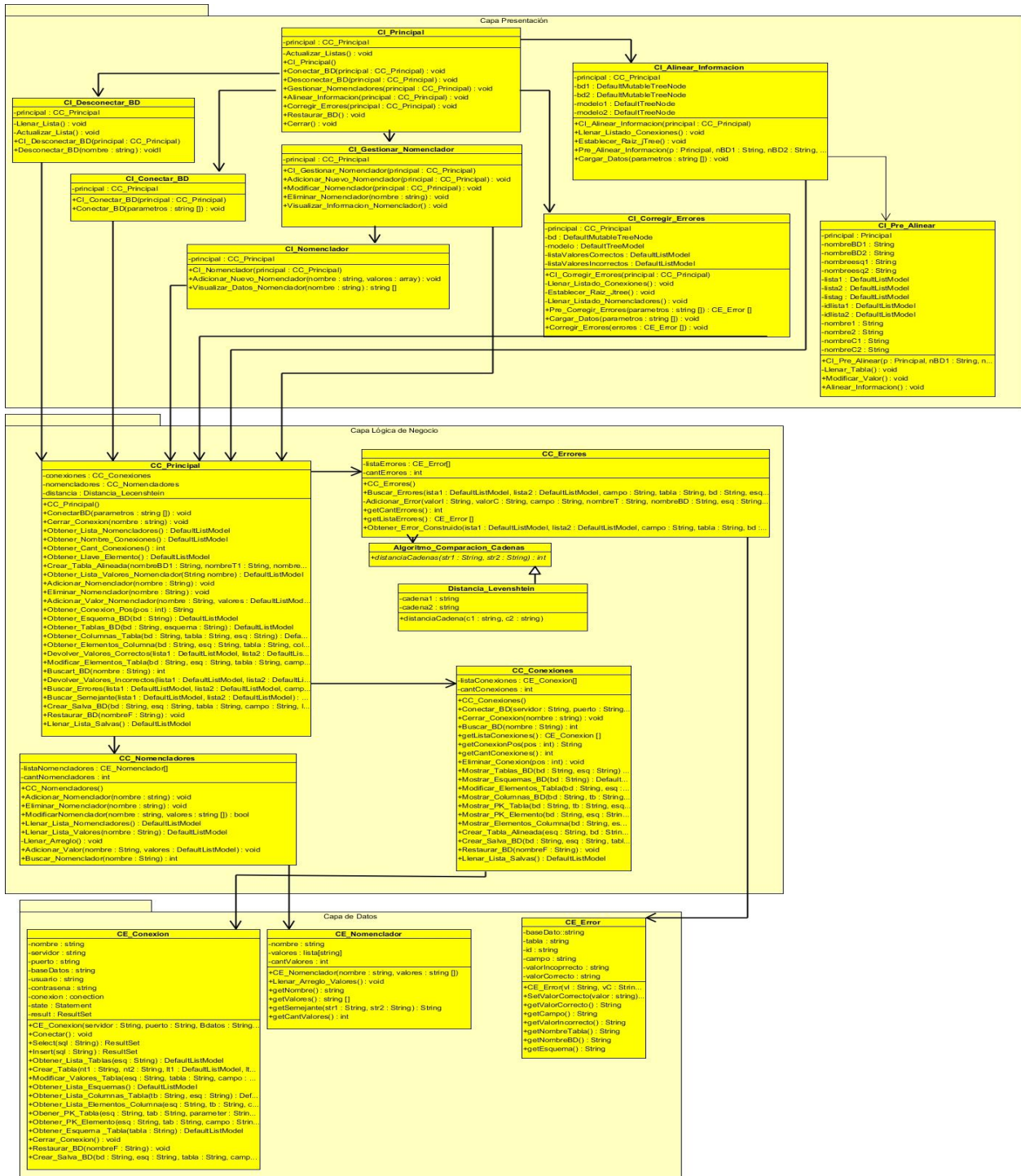


Fig. 3.1 Diagrama de clases del diseño.

3.3. Diagrama de componentes

“Un diagrama de componentes muestra las dependencias lógicas entre componentes software, sean éstos componentes fuentes, binarios o ejecutables, ilustran las piezas del software, controladores embebidos, etc. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema, es decir para describir la vista de implementación estática de un sistema” (Arizaca, 2009).

A continuación en la fig. 3.2, se presenta el diagrama de componentes asociado al sistema propuesto como solución.

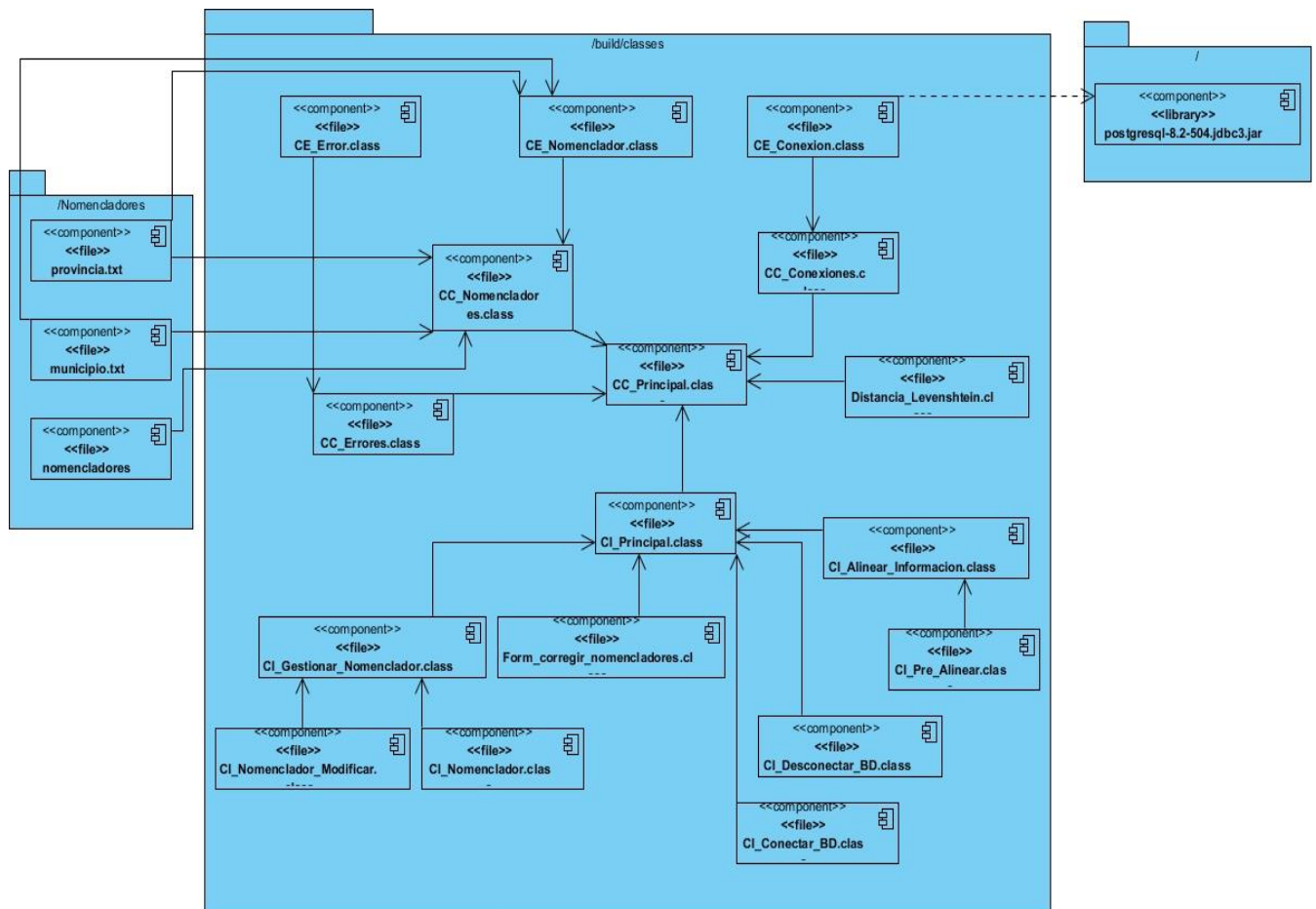


Fig. 3.2 Diagrama de componentes.

3.4. Diagrama de Despliegue

El Modelo de Despliegue se define como “*un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos del cómputo*” (Rumbaugh, 1999). El modelo de despliegue se utiliza como entrada fundamental en las actividades de diseño e implementación debido a la gran influencia que tiene para el diseño de un sistema la distribución física del mismo. A continuación se muestra en la fig. 3.3 el modelo de despliegue del sistema propuesto.

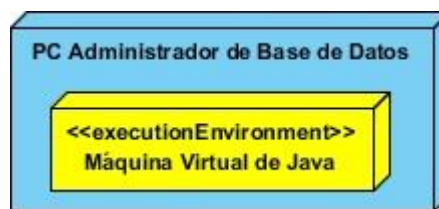


Fig. 3.3 Diagrama de despliegue

En el diagrama anterior, se representa la PC del Administrador de Base de Datos, en la cual se instalará el sistema. Como único requerimiento, para la instalación, se hace necesaria la Máquina Virtual de Java.

3.5. Pruebas de software

“La prueba es el proceso de ejecución de una aplicación con el propósito de comprobar que el producto satisfaga los requerimientos enunciados y que el mismo tenga el comportamiento esperado. Para que las pruebas sean exitosas se hace necesario confeccionar casos de pruebas que tengan probabilidades de descubrir los errores del sistema utilizando técnicas que rijan el proceso de la prueba” (Pérez, 2011).

3.5.1. Pruebas de caja negra

“Las pruebas de caja negra son un método de ensayo en el que los datos de prueba se derivan de los requisitos funcionales especificados sin tener en cuenta la estructura final del programa. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos” (Pressman, 2004).

Se utilizará el método de Caja Negra, el cual se realiza sobre la interfaz del software mediante la confección de los casos de prueba basados en los caso de uso definidos para el sistema, utilizando la técnica de Partición de Equivalencia. Esta técnica divide el campo de entrada en clases de datos que ejercitan determinadas funciones del software, es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software. El objetivo del diseño de casos de prueba es que sean efectivos descubriendo defectos en los programas y muestren que el sistema satisface sus requisitos. Para diseñar un caso de prueba, se selecciona una característica del sistema o componente que se está probando.

El uso de las pruebas de caja negra permitirá detectar no conformidades, así como mal funcionamiento en el sistema, no detectados durante la implementación. Se realizarán tantas iteraciones como sean necesarias, con el fin de obtener un sistema estable y acorde a las necesidades del interesado.

A continuación se muestran las tablas 3.1 a la 3.5, correspondientes al caso de prueba para el caso de uso Gestionar Nomenclador.

Tabla 3.1 Secciones a probar en el caso de uso Gestionar Nomenclador.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC 1: "Añadir nomenclador"	EC 1.1: Añadir nomenclador exitosamente.	<p>El sistema muestra una interfaz donde el usuario adiciona los datos de un nomenclador:</p> <ul style="list-style-type: none"> - Nombre. - Valor del Nomenclador. - Listado de Valores <p>El usuario entra los datos asociados al nuevo nomenclador y hace clic izquierdo en el botón Insertar. El sistema guarda los cambios y se muestra la lista de nomencladores actualizada</p>	<p>Alineamiento de información en base datos espaciales:</p> <ol style="list-style-type: none"> 1. Página principal. 2. Clic en el menú Trabajo 3. Clic en la opción Gestionar Nomenclador. 4. Clic en el botón Adicionar Nomenclador 5. Clic en el botón Adicionar valores

		con el nuevo nomenclador y termina el caso de uso.	6. Clic en el botón Insertar.
	EC 1.2: Añadir nomenclador sin éxito.	El sistema muestra un mensaje de información "Inserte nombre". El usuario hace clic izquierdo en el botón Aceptar para cerrar la ventana que contiene el mensaje de error.	<p>Alineamiento de información en base datos espaciales:</p> <ol style="list-style-type: none"> 1. Página principal. 2. Clic en el menú Trabajo 3. Clic en la opción Gestionar Nomenclador. 4. Clic en el botón Adicionar Nomenclador 5. Clic en el botón Insertar. 6. Clic en el botón Aceptar para cerrar la ventana de error.
	EC 1.3: Cancelar petición.	El usuario hace clic izquierdo en el botón Cancelar. El sistema no guarda los cambios y termina el caso de uso.	<p>Alineamiento de información en base datos espaciales:</p> <ol style="list-style-type: none"> 1. Página principal. 2. Clic en el menú Trabajo 3. Clic en la opción Gestionar Nomenclador. 4. Clic en el botón Adicionar Nomenclador.

			5. Clic en el botón Cancelar.
	<p>EC 1.4: Añadir Nomenclador sin éxito por la existencia del mismo.</p>	<p>El sistema muestra un mensaje de error "El nomenclador especificado ya existe". El usuario hace clic izquierdo en el botón Aceptar para cerrar la ventana que contiene el mensaje de información.</p>	<p>Alineamiento de información en base datos espaciales:</p> <ol style="list-style-type: none"> 1. Página principal. 2. Clic en el menú Trabajo 3. Clic en la opción Gestionar Nomenclador. 4. Clic en el botón Adicionar Nomenclador 5. Clic en el botón Insertar. 6. Clic en el botón Aceptar para cerrar la ventana de error.
<p>SC 2: "Modificar nomenclador"</p>	<p>EC 2.1: Modificar nomenclador exitosamente.</p>	<p>El sistema muestra una interfaz donde el usuario modifica los datos de un nomenclador seleccionado:</p> <ul style="list-style-type: none"> - Nombre. - Valor del Nomenclador. - Listado de Valores <p>El usuario modifica los datos asociados al nomenclador y hace</p>	<p>Alineamiento de información en base datos espaciales:</p> <ol style="list-style-type: none"> 1. Página principal. 2. Clic en el menú Trabajo 3. Clic en la opción Gestionar Nomenclador. 4. Clic en el botón

		<p>clic izquierdo en el botón Insertar. El sistema guarda los cambios y se muestra la lista de nomencladores actualizada con el nomenclador modificado y termina el caso de uso.</p>	<p>Modificar Nomenclador</p> <ol style="list-style-type: none"> 5. Clic en el botón Adicionar valores 6. Clic en el botón Insertar.
	<p>EC 2.2: Modificar nomenclador sin éxito.</p>	<p>El sistema muestra un mensaje de información “Los nomencladores provincia y municipio no pueden ser modificados”. El usuario hace clic izquierdo en el botón Aceptar para cerrar la ventana que contiene el mensaje de error.</p>	<p>Alineamiento de información en base datos espaciales:</p> <ol style="list-style-type: none"> 1. Página principal. 2. Clic en el menú Trabajo 3. Clic en la opción Gestionar Nomenclador. 4. Clic en el botón Modificar Nomenclador 5. Clic en el botón Aceptar para cerrar la ventana de error.
	<p>EC 2.3: Cancelar petición.</p>	<p>El usuario hace clic izquierdo en el botón Cancelar. El sistema no guarda los cambios y termina el caso de uso.</p>	<p>Alineamiento de información en base datos espaciales:</p> <ol style="list-style-type: none"> 1. Página principal. 2. Clic en el menú Trabajo 3. Clic en la opción Gestionar Nomenclador. 4. Clic en el botón Modificar Nomenclador.

			5. Clic en el botón Cancelar.
	EC 2.4: Modificar nomenclador sin éxito por la existencia de un nomenclador con el mismo nombre.	El sistema muestra un mensaje de error "El nomenclador especificado ya existe". El usuario hace clic izquierdo en el botón Aceptar para cerrar la ventana que contiene el mensaje de información.	<p>Alineamiento de información en base datos espaciales:</p> <ol style="list-style-type: none"> 1. Página principal. 2. Clic en el menú Trabajo 3. Clic en la opción Gestionar Nomenclador. 4. Clic en el botón Modificar Nomenclador 5. Clic en el botón Insertar. 6. Clic en el botón Aceptar para cerrar la ventana de error.
SC 3: "Eliminar área"	EC 3.1: Eliminar nomenclador exitosamente.	<p>El sistema muestra un mensaje de confirmación donde el usuario elimina un nomenclador:</p> <ul style="list-style-type: none"> - Si. - No. - Cancelar <p>El usuario confirma que desea eliminar el nomenclador. El sistema guarda los cambios y se muestra lista de nomencladores actualizada y termina el caso de uso.</p>	<p>Alineamiento de información en base datos espaciales:</p> <ul style="list-style-type: none"> • Página principal. • Clic en el menú Trabajo • Clic en la opción Gestionar Nomenclador. • Clic en el botón Eliminar Nomenclador. • Clic en el botón Sí.
	EC 3.2: Eliminar	El sistema muestra un mensaje de	Alineamiento de

nomenclador sin éxito.	información “Los nomencladores provincia y municipio no pueden ser eliminados”. El usuario hace clic izquierdo en el botón Aceptar para cerrar la ventana que contiene el mensaje de error.	información en base datos espaciales: <ol style="list-style-type: none"> 1. Página principal. 2. Clic en el menú Trabajo 3. Clic en la opción Gestionar Nomenclador. 4. Clic en el botón Eliminar Nomenclador 5. Clic en el botón Aceptar para cerrar la ventana de error.
EC 3.3: Cancelar petición.	El usuario hace clic izquierdo en el botón Cancelar. El sistema no guarda los cambios y termina el caso de uso.	Alineamiento de información en base datos espaciales: <ol style="list-style-type: none"> 1. Página principal. 2. Clic en el menú Trabajo 3. Clic en la opción Gestionar Nomenclador. 4. Clic en el botón Eliminar Nomenclador. 5. Clic en el botón Cancelar.

Tabla 3.2 Tabla de variables.

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Nombre	Campo de texto	No	El usuario escribe el nombre del nuevo

CAPÍTULO 3

				Nomenclador.
2	Valor Nomenclador	Campo de texto	Si	El usuario escribe el valor del Nomenclador
3	Valores Nomenclador	Lista de texto	Si	El usuario visualiza los valores del Nomenclador
4	Adicionar valor	Botón	No	El usuario añade los valores del Nomenclador.
5	Eliminar Valor	Botón	No	El usuario elimina valores del Nomenclador.

Tabla 3.3 Matriz de datos SC Adicionar Nomenclador.

Escenario	Nombre del nomenclador	Valor de Nomenclador	Valores	Respuesta del Sistema	Resultado de la Prueba
EC 1.1 Añadir Nomenclador satisfactoriamente.	V "Calles"	V "Luz Caballero"	V "Luz Caballero"	El sistema añade satisfactoriamente un nomenclador.	Satisfactorio
EC1.2 Añadir nomenclador sin éxito.	V " "	V "La Habana"	V "La Habana" V "Pinar del Río"	El sistema muestra un mensaje "Debe escribir el nombre del nomenclador".	Satisfactorio
EC1.3 Cancelar petición.	NA	NA	NA	El sistema no guarda los cambios y cierra la ventana.	Satisfactorio
Añadir nomenclador sin éxito por la existencia del mismo.	V " provincia"	V "Sancti Spíritus"	V "Sancti Spíritus" V "Matanzas"	El sistema muestra un mensaje "Ya existe un nomenclador con ese nombre".	Satisfactorio

CAPÍTULO 3

Tabla 3.4 Matriz de datos SC Modificar Nomenclador.

Escenario	Nombre del nomenclador	Valor de Nomenclador	Valores	Respuesta del Sistema	Resultado de la Prueba
EC 1.1 Modificar Nomenclador satisfactoriamente.	V " Calles" se modifica con V " Calles"		V "Luz Caballero"	El sistema modifica satisfactoriamente el Nomenclador.	Satisfactorio
EC1.2 Modificar nomenclador sin éxito.	V " provincia"			El sistema muestra un mensaje "Los nomencladores provincia y municipio no pueden ser modificados".	Satisfactorio
EC1.3 Cancelar petición.	NA	NA	NA	El sistema no guarda los cambios y cierra la ventana.	Satisfactorio
Modificar nomenclador sin éxito por la existencia del mismo.	V " Calles" se modifica con V "provincia"	V " Sancti Spíritus"	V "Sancti Spíritus" V "Matanzas"	El sistema muestra un mensaje "Ya existe un nomenclador con ese nombre".	Satisfactorio

Tabla 3.5 Matriz de datos SC Eliminar Nomenclador.

Escenario	Nombre del nomenclador	Respuesta del Sistema	Resultado de la Prueba

Escenario	Nombre del nomenclador	Respuesta del Sistema	Resultado de la Prueba
EC 1.1 Eliminar Nomenclador satisfactoriamente.	V "Calles"	El sistema elimina satisfactoriamente el Nomenclador.	Satisfactorio
EC1.2 Eliminar nomenclador sin éxito.	V "Provincia"	El sistema muestra un mensaje "Los nomencladores provincia y municipio no pueden ser eliminados".	Satisfactorio
EC1.3 Cancelar petición.	NA	El sistema no guarda los cambios y cierra la ventana.	Satisfactorio

Después de aplicar los casos de pruebas al sistema, se detectaron en una primera iteración 12 no conformidades asociadas a errores ortográficos, de interfaz y de validación, las cuales fueron resueltas en período de tiempo de 3 días. Se realizó una segunda iteración y se detectaron 5 no conformidades, asociadas a errores ortográficos y de validación, siendo solucionadas en 2 días. En una tercera iteración no se detectaron no conformidades siendo eliminadas en su totalidad en las dos iteraciones anteriores. En la fig. 3.4 se muestra una gráfica con la cantidad de no conformidades que se detectaron por cada iteración.

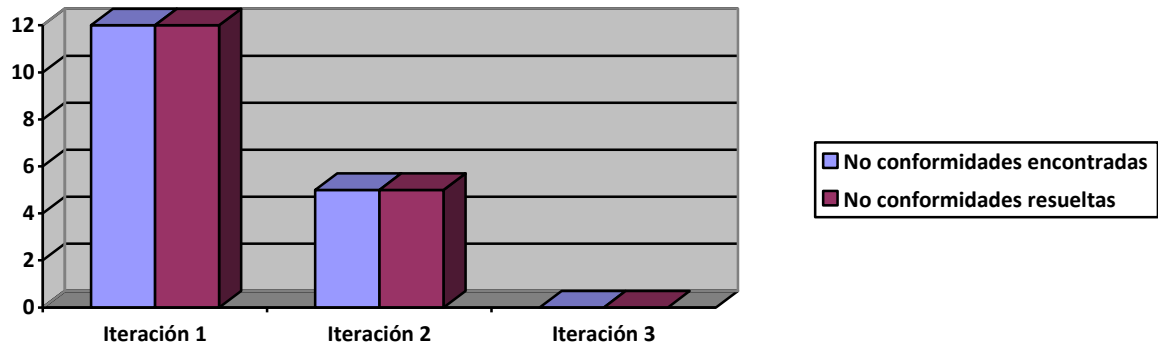


Fig. 3.4 Gráfico de no conformidades encontradas. Caso de uso Gestionar Nomenclador

3.5.2. Encuesta de aceptación al interesado

Una vez presentado el sistema desarrollado al interesado, se realizó una encuesta al grupo de desarrolladores de la factoría de Base de Datos Espaciales y Cartografías, de la LPS Aplicativos SIG, con el objetivo de determinar si el producto cumple con las necesidades de la factoría. En dicha encuesta, se evaluó en una escala de 1 – 5, siendo 1 malo y 5 excelente, aspectos como: eficiencia, usabilidad, rendimiento y cumplimiento de los objetivos propuestos. Esta encuesta puede ser encontrada en el anexo 2 de la presente investigación.

Tras la aplicación de la encuesta, se pudo determinar que, el sistema desarrollado:

- Cuenta con una gran aceptación por parte del equipo de desarrollo de la factoría de Base de Datos Espaciales y Cartografías.
- Cumple con los objetivos propuestos en la investigación, así como, con las necesidades existentes hoy en la factoría, relacionadas al tema del alineamiento de información.
- Es eficiente y cuenta con un buen rendimiento en cuanto al tiempo de respuesta, en el cumplimiento de las funcionalidades.
- Cuenta con una interfaz sencilla y asequible a los usuarios.

3.6. Conclusiones del Capítulo

El uso de los patrones de diseño, permitió obtener un sistema bien estructurado, donde cada clase cumple con las responsabilidades que les corresponde, evitando de esta manera, la existencia de clases sobredimensionadas. La definición del diagrama de clases de diseño, permitió un mejor entendimiento de la estructuración de las clases del sistema, así como la comunicación entre ellas. Dentro de las pruebas de aceptación aplicadas al software, se utilizó el método de caja negra con la técnica partición equivalente, la cual permitió identificar las no conformidades con las que contaba el sistema. Se logró identificar que en la primera iteración de pruebas, existían 12 errores dentro de las funcionalidades del sistema, lo cual permitió que dichos errores fueran erradicados totalmente. La descripción de las pruebas utilizadas para asegurar la calidad del software, permitió obtener resultados satisfactorios, asegurando que el sistema implementado no contiene errores y tiene la aceptación requerida. Asimismo, la aplicación de una encuesta a los interesados, permitió identificar que el sistema cumple con las necesidades y cuenta de una gran aceptación por parte del equipo de desarrollo de la factoría de Base de Datos Espaciales y Cartografías.

CONCLUSIONES

Una vez finalizada la fundamentación teórica que sustentó la presente investigación, definidas las características del sistema y efectuado su desarrollo y validación, se obtuvieron resultados que permiten arribar a las siguientes conclusiones:

- El empleo del algoritmo de Levenshtein facilitó el proceso referente al alineamiento de información en las bases de datos del proyecto Aplicativos SIG.
- La selección de herramientas libres y de elevada calidad permitió la construcción de un sistema libre del pago de licencias a terceros, eficiente y multiplataforma.
- El sistema constituye un paso más en la soberanía tecnológica del País, puesto que permite la solución de un problema existente además de generar nuevo conocimiento a la Universidad y la sociedad.

RECOMENDACIONES

Una vez concluido el proceso de desarrollo, el autor del presente trabajo recomienda:

- Extender el sistema desarrollado permitiéndole la conexión a otras bases de datos además de PostgreSQL.
- Extender el sistema desarrollado a través de la inclusión de otros algoritmos de comparación de cadenas, permitiéndole al usuario elegir con cuál de ellos se realizará el análisis de la información.

REFERENCIAS

1. **Abarca A., Héctor. 2008.** *Corrección de Errores*. Chile : s.n., 2008.
2. **ACIMED. 2007.** Scielo. [En línea] ACIMED, 10 de 2007. [Citado el: 28 de 02 de 2014.] http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352007001000005.
3. **Ager, Ingenieros. 2003.** *ageringenieros*. [En línea] AGER INGENIEROS, 2003. [Citado el: 12 de 01 de 2014.] <http://www.ager.es/productos/gis/datos.php>.
4. **Algoritmos, Google. 2006.** Comparación de cadenas. Algoritmo levenshtein. [En línea] 15 de 11 de 2006. [Citado el: 03 de 03 de 2014.] <http://webj2ee.blogspot.com/2006/11/algoritmo-levenstein.html>.
5. **Arizaca, Lic. Elisa Ramírez. 2009.** *Diagrama de Componentes*. Bolivia : Universidad Saleciana de Bolivia, 2009.
6. *Arquitectura Cliente/Servidor. DLSI, Departamento de Lenguajes y Sistemas Informáticos. 2004-2005.* Alicante : s.n., 2004-2005.
7. **Benavides, Paula Andrea Cañón y Rodríguez, Sandra Correa. 2007.** *Procesamiento Del Lenguaje Natural En Colombia* : Universidad De La Salle, 2007.
8. **Boehm, Barry. 1995.** *Anchoring the Software Process*. Estados Unidos : s.n., 1995.
9. **Bordón L, D'Avanzo E. 2004.** *Perspectivas para la integración de la minería de textos y la gestión del conocimiento*. Luxemburgo : The IPTS Report, 2004. No. 85.
10. **Bustos Thames, Juan Pablo. 2011.** slideshare. [En línea] Universidad Tecnológica Nacional, 08 de 2011. [Citado el: 01 de 05 de 2014.] <http://www.slideshare.net/jpbthames/diagramas-de-clases>.
11. **Camacho, Erika and Nuñez, Fabio Cardeso and Gabriel. 2004.** *Arquitecturas de Software, Guías de Estudio*. . 2004.
12. **CENATAV. 2014.** *Sistema de Información Geográfica como Herramienta para la Integración e Interpretación Semántica de la Información Espacial*. La Habana , Cuba : s.n., 2014.
13. **Consultores, WebMining. 2011.** WebMining. [En línea] 23 de 06 de 2011. [Citado el: 28 de 02 de 2014.] <http://www.webmining.cl/2011/06/distancia-de-levenshtein/>.

14. **creativecommons. 2014.** Conoce3000. [En línea] [Citado el: 01 de 03 de 2014.] <http://www.conoce3000.com/html/espaniol/Libros/PascalConFreePascal/Cap08-07-Busqueda%20de%20cadenas%20de%20caracteres%20Boyer%20&%20Moore.php>.
15. **Date, C. J. 2001.** *Introducción a los Sitemas de Bases de Datos*. Mexico : Person Education, 2001.
16. **Diccionario Enciclopédico, Vox 1. 2009.** TheFreeDictionary. [En línea] 2009. [Citado el: 12 de 01 de 2014.] <http://es.thefreedictionary.com/nomenclador>.
17. *Discovering technologies using techmining: the case of waste recycling. The 6th International Scientific Conference "Business and Management 2010"*. **Rio, R.M. y Cilleruelo, E. 2010.** Lituania : Technika, 2010.
18. **DLSI, Departamento de Lenguajes y Sistemas Informáticos. 2005.** slideshare. [En línea] 2005. http://www.slideshare.net/angel_ito/03b-arquitectura-clienteservidor-n-capas.
19. **DLSI, Departamento de Lenguajes y Sitemas Informáticos. 2004-2005.** Alicante. [En línea] Asociación Cartográfica Internacional, 2004-2005. <http://icaci.org>.
20. **Ehrig, M. 2007.** *Ontology Alignment: Bridging the Semantic Gap*. s.l. : Springer, 2007.
21. **Engineer, Who - Principal Software. 2013.** CODE PROJECT. [En línea] 4 de Mayo de 2013. <http://www.codeproject.com/Articles/430014/N-Tier-Architecture-and-Tips>.
22. **Entorno Virtual de Aprendizaje. 2007-2008.** EVA. [En línea] 2007-2008. [Citado el: 12 de 4 de 2014.] <http://eva.uci.cu/course/view.php?id=161..>
23. **Espinosa, Daniel Bravo. 2014.** wiki-ApliDaniel. [En línea] 2014. <http://wiki-aplidaniel.wikispaces.com/Modelo+n-capas>.
24. **Garea Llano, Dr. C. Eduardo, Vera Varonisky, Lic. Francisco. 2009.** *Alineamiento de Ontologías en el dominio geoespacial*. Ciudad de La Habana, Cuba : s.n., 2009.
25. **Garea Llano, Dr. Eduardo. 2007.** *Estado Actual de la Interpretación Semántica de Datos Espaciales*. La Habana, Cuba : s.n., 2007.
26. **Gilfillan, Ian. 2000.** *La Biblia de MySQL*. s.l. : ANAYA Multimedia, 2000.
27. **Gruber, T.R. 1995.** *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*. s.l. : International Journal of Human - Computer Studies, 1995.

28. **Güting, R. H. 1994.** *An Introduction to Spatial Database Systems*. Alemania : s.n., 1994.
29. **Hernández, Enrique Orallo. 2002.** *El Lenguaje Unificado de Modelado (UML)*. Valencia : s.n., 2002.
30. **J. Borrego-Días, A.M. Chávez-González. 2005.** *Extension of Ontologies Assisted by Automated Reasoning Systems*. 2005.
31. **J.A. Alonso-Jimenez, J. Borrego-Díaz, A. M. Chávez-Gonzalez, F.J. Martín-Mateos. 2006.** *Foundational Challenges in Automated Data and Ontology Cleaning in the Semantic Web*. IEEE Intelligent System 21 : s.n., 2006.
32. **JACOBSON, Ivar, BOOCH, Grady y RUMBAUGH, James. 2000.** *El Proceso Unificado de Desarrollo de Software*. [ed.] Andrés Otero. [trad.] Salvador Sánchez, y otros. Madrid : Addison Wesley, 2000. ISBN: 84-7829-036-2.
33. **López, Adriana Cumplido. 2011.** *Desambiguación de Topónimos para el Tratamiento de Ambigüedad en Consultas de Recuperación de Información Geográfica usando una Ontología Mixta para el idioma Español*. Universidad Autónoma de Puebla. Mexico : Universidad Autónoma de Puebla, 2011.
34. **Manual de la Lengua Española, Diccionario Vox. 2007.** TheFreeDictionary. [En línea] 2007. [Citado el: 13 de 01 de 2014.] <http://es.thefreedictionary.com/alineamiento>.
35. **Microsoft. 2014.** TechNet. [En línea] Microsoft, 2014. [Citado el: 12 de 01 de 2014.] <http://technet.microsoft.com/es-es/library/bb933790.aspx>.
36. **Münster, Institute For Geoinformatics University of. 2014.** MUNSTER SEMANTIC INTEROPERABILITY LAB. [En línea] Münster University, 2014. [Citado el: 04 de 02 de 2014.] <http://www.uni-muenster.de/Geoinformatics/en/musil/index.html>.
37. **National Center for Geographic Information and Analysis. 1990.** [En línea] 1990. [Citado el: 22 de 03 de 2014.] www.ncgia.ucsb.edu.
38. **Office, Microsoft. 2014.** Microsoft Office. [En línea] Microsoft, 2014. <http://office.microsoft.com/es-mx/access-help/conceptos-basicos-sobre-bases-de-datos-HA010064450.aspx>.

REFERENCIAS

39. **Oracle.** 2014. TecnoProgramas. [En línea] 2014. <http://www.tecnoprogramas.com/descargar-java-version-6-26-espanol-32-64-bits-por-tu-seguridad.htm>.
40. **Padrón, Liudmila Torres.** 2006. monografias.com. [En línea] 04 de 03 de 2006. [Citado el: 18 de 03 de 2014.] <http://www.monografias.com/trabajos34/similitud/similitud.shtml>.
41. **Pavón Mestras, Juan.** 2004. *Patrones de diseño orientado a objetos*. Madrid, España : Universidad Complutense Madrid, 2004.
42. **Pérez, Adrián Fuentenegra.** 2011. *Generación de consultas para la manipulación de Geontologías desde la Plataforma GeneSIG*. 2011.
43. **Pressman, Roger.** 2004. *Ingeniería de Software. Un enfoque práctico*. 2004. Quinta Edición..
44. **Pressman, Roger S.** 2006. *Ingeniería de Software: Un enfoque práctico*. México : McGraw Hill, 2006.
45. **RAE, Diccionario de la lengua española.** 2005. WordReference.com. [En línea] Online Language Dictionaries, 2005. [Citado el: 03 de 04 de 2014.]
46. **Ramírez, M.Sc. Kryscia Daviana Benavides.** 2012. *Algoritmos de Similitud y Distancia*. Costa Rica : s.n., 2012.
47. **Real Academia, Española.** 2014. Real Academia Española. [En línea] 2014. [Citado el: 13 de 01 de 2014.] <http://lema.rae.es/drae/?val=nomencl%C3%A1tor>.
48. **Reynoso, Carlos and Kiccillof, Nicolás.** 2004. *Estilos y Patrones en la Estrategia de Arquitectura*. Buenos Aires : Universidad de Buenos Aires, 2004. V 1.0..
49. **Rica, Universidad de Costa.** 2006. Introducción al Agile UP. [En línea] Universidad de Costa Rica, 13 de mayo de 2006. <http://www.cc.una.ac.cr/AUP/html/overview.html>.
50. **Rica, Universidad Nacional de Costa.** 2006. DTIC- Centro de Gestión Infomática. [En línea] 13 de Mayo de 2006. <http://www.cc.una.ac.cr/AUP/html/overview.html>.
51. **Rumbaugh, James.** 1999. *“El Lenguaje Unificado de Modelado. Guía del usuario”*. Madrid : PEARSON EDUCATION S.A., 1999.

52. **Samet, Hanan. 1994.** *Spatial Data Models and Query Processing. In Modern Database Systems: The Object Model, Interoperability, and Beyond.* Reading, MA : Addison Wesley/ACM Press, 1994.
53. **Sinclair, QL. 2010.** QBlog. [En línea] 01 de 12 de 2010. [Citado el: 02 de 03 de 2014.] <http://sinclairqls.wordpress.com/2010/12/01/el-algoritmo-de-ratcliffobershelf-en-superbasic/>.
54. **Synergix. 2011.** Tecnología y Synergix. [En línea] Synergix, enero de 2011. <http://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>.
55. **Tedeschi, Nicolás. 2014.** Microsoft Developer Network. [En línea] Microsoft, 2014. [Citado el: 01 de 05 de 2014.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.
56. **Ugaz Horna, Patricia. 2008.** *Sistema De Información De Control De Almacén (SICAL).* 2008.
57. *Untangling text data mining.* **HEARST, M. 1999.** Estados Unidos : Proceedings of ACL'99: the 37th Annual Meeting of the Association For Computational Linguistic ACL, 1999. págs. 3-10.
58. **Vicomtech-IK4. 2014.** vicomtech. [En línea] Vicomtech-IK4, 2014. [Citado el: 15 de 3 de 2014.] <http://www.vicomtech.org/t4/e11/procesamiento-del-lenguaje-natural>.
59. **Visual Paradigm, International. 2014.** visual-paradigm. [En línea] Visual Paradigm International, 2014. [Citado el: 13 de 4 de 2014.] <http://www.visual-paradigm.com/>.
60. **Vzla., Diseño y Mantenimiento Grafikart de. 2011.** TecnoCiencia. [En línea] 2011. <http://www.tecnoprogramas.com/>.
61. **Weigand, H. 1997.** *Multilingual Ontology-Based Lexicon for News Filtering - The TREVI Project.* Holanda : Tilburg University, 1997. págs. 138-159.

ANEXOS

Anexo 1.

Anexo 1.1. Entrevista realizada al Ing. Lister Cruz, jefe del equipo de desarrollo de Bases de Datos Espaciales y Cartografías, perteneciente a la LPS Aplicativos SIG. Día 2 de febrero de 2014. El objetivo de esta entrevista fue definir los principales problemas existentes en la LPS Aplicativos SIG y el flujo de trabajo del equipo de desarrollo de Bases de Datos Espaciales y Componentes.

¿Es postgres + postgis el único gestor de bases de datos empleados en el proyecto?

Sí.

¿Todos los sistemas desarrollados en el proyecto necesitan de una base de datos?

La mayoría.

¿Todas las bases de datos con las que se trabaja en el proyecto son geoespaciales?

Sí.

¿Cada cuánto, aproximadamente, es necesaria la creación de una nueva base de datos?

En dependencia de los nuevos negocios que surjan, esto es relativo ya que no se puede predecir cuándo se presentará un nuevo producto a desarrollar por el equipo de la LPS.

¿Cada cuánto, aproximadamente, es necesaria la realización de una alineación entre los datos de una base de datos del proyecto y otra base de datos, ya sea del cliente o del propio proyecto?

Por lo general cada vez que surge un nuevo proyecto al menos hay que tenerse en cuenta posibles alineaciones y en caso de existir entonces hay que implementarlas.

¿Cada cuánto, aproximadamente, es necesaria la corrección de errores en los datos de una base de datos?

Las correcciones de errores por lo general se realizan durante la etapa del diseño y elaboración de las BD de los sistemas a desarrollar, luego de esto durante la etapa de implementación del proyecto también se hacen algunos cambios en conjunto con los desarrolladores, para adecuar mejor la BD a la implementación de cada sistema.

¿Se posee en el proyecto algún mecanismo, forma o facilidad que permita al administrador de la base de datos realizar las tareas de alineación de información y corrección de errores de manera más simple que manualmente?

No existe.

¿Cuáles son los principales campos que tienen en común las bases de datos existentes en el proyecto actualmente, qué tipo de información almacenan estos campos?

Por lo general son los nomencladores y las cartografías bases las que se utilizan para la visualización de los mapas. Los datos almacenados son datos geo-espaciales y socioeconómicos. Los nomencladores que más comúnmente podemos encontrar en las BD son provincia y municipio.

Anexo 1.2. Entrevista realizada al Ing. Alain Companioni, Jefe de la LPS Aplicativos SIG. Día 2 de febrero de 2014. El objetivo de esta entrevista fue definir cómo afectan los problemas existentes en el equipo de desarrollo de Bases de Datos Espaciales y Cartografías, a la LPS Aplicativos SIG.

¿Cómo afectan los problemas que puedan presentarse en la factoría de BD espaciales y cartografías al resto de la LPS?

La LPS Aplicativos SIG trabaja como un todo, donde cada equipo de desarrollo, diseña y elabora una parte específica del software, para finalmente, componer el sistema completo. Si el trabajo de uno de los equipos falla, puede incurrirse en retrasos en el cronograma, así como en mal funcionamiento de la aplicación. De ocurrir lo mencionado anteriormente, surgirían no conformidades en los clientes, lo cual significaría la pérdida de prestigio de la LPS y de la Universidad en general. Todo esto se traduciría en pérdida de clientes, lo cual significa, pérdida de ingresos, afectando de esta forma, no solo a la Universidad, sino a todo el país. Por eso es de vital importancia que todos los equipos de desarrollo de la LPS, funcionen de acuerdo a la planificación y cumplan con las tareas que se le asignan.

Anexo 2.

Anexo 2.1. Entrevista realizada al Ing. Liester Cruz, jefe del equipo de desarrollo de Bases de Datos Espaciales y Cartografías, perteneciente a la LPS Aplicativos SIG. Día 23 de mayo de 2014. El objetivo de esta entrevista fue definir si el sistema desarrollado, cumple con las necesidades de la factoría, relacionadas al tema de alineamiento de información, y si cuenta con aceptación por parte del equipo de desarrollo de dicha factoría.

Usuario: lcruz.

Rol: Especialista en Base de Datos. LPS Aplicativos SIG.

Para la evaluación del sistema propuesto valore las siguientes características de acuerdo a sus criterios, ofreciendo una puntuación de 1 a 5, siendo 1 malo y 5 excelente.

1. Eficiencia de la respuesta de las funcionalidades del sistema. _5__
2. Rendimiento de la aplicación en tiempo de ejecución. _5__
3. Cumplimiento de los objetivos propuestos por la investigación. _4__
4. Usabilidad de la aplicación, existencia y comprensión del manual de usuario. _5__
5. Diseño asequible, de calidad visual con colores apropiados y correcto posicionamiento y configuración de las ventanas de las funcionalidades. _4__
6. Requerimientos de la aplicación para su correcto funcionamiento (necesidad de la plataforma JAVA). _4__
7. Espacio necesario para el sistema teniendo en cuenta el objetivo para el que fue propuesto. _5__

Consideraciones:

Considero que el sistema obtenido cumple con las expectativas trazadas al inicio de la tesis y da respuesta a las necesidades identificadas en la LPS Aplicativos SIG sobre la alineación de información.