



**Universidad de las Ciencias
Informáticas**

Facultad 1

**COMPONENTE PARA GENERAR UN SERVICIO WEB DE
INFORMACIÓN GEORREFERENCIADA**

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor:

Carel Yosvani Rodríguez Mendoza

Tutores:

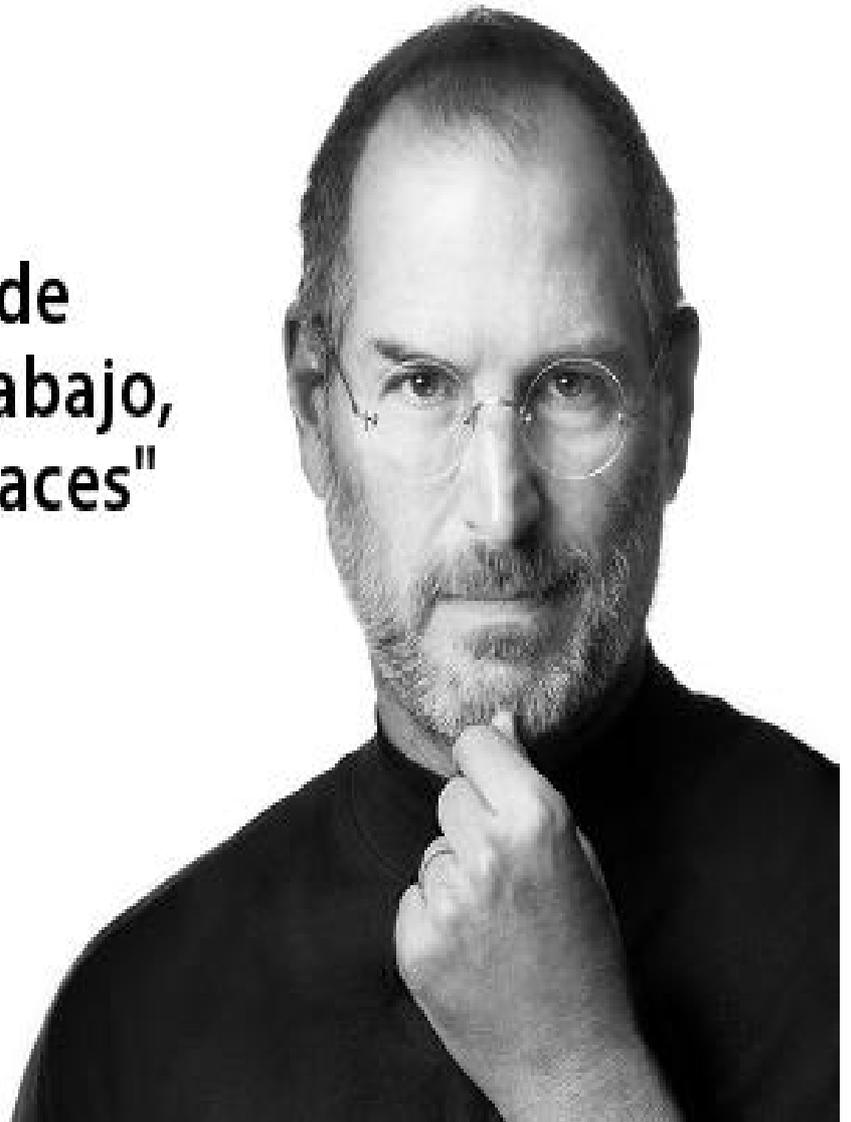
MSc. Leiny Amel Pons Flores

Ing. Alexander García O'reilly

**La Habana, junio de 2019
"Año 60 de la Revolución"**

**"La única forma de
hacer un gran trabajo,
es amar lo que haces"**

—Steve Jobs



DEDICATORIA

A mi mamá, a mi hermana Yailien y a mi sobrinito Jeikson.



AGRADECIMIENTOS

A mi mamá por brindarme su apoyo incondicional en cada momento y estar siempre ahí para mí, y ser un ejemplo a seguir de valentía y perseverancia.

A mi hermana Yailien, por ser más que una hermana. Por ser mi amiga, mi confidente, por siempre estar en cada momento que la he necesitado, por ser mi compañera de toda la vida.

A mi hermano Dixan por apoyarme en el cumplimiento de mis metas.

A mi novia Arlet, por ser mi todo en estos cinco años y estar en cada momento de manera incondicional, por su dedicación, apoyo, paciencia... por existir.

A la familia de mi novia, por acogerme como parte de ella y darme tanto cariño en todo este tiempo.

A toda mi familia, por apoyarme en todo momento.

A mis tutores, en especial a Leiny, por haber sido un amigo y brindarme su confianza, por los regaños, las risas, los consejos, las anécdotas; para ti mi eterno agradecimiento.

A todos mis amigos de la universidad, por ser como una familia para mí y compartir tantos momentos inolvidables.

A la familia del BFB, por tantas risas y tensiones... siempre seremos REALES, siempre seremos 1 Nation.

A mis profesores, de los que aprendí tantas cosas y que nunca voy a olvidar. A Yaima, Israel, Jose Carlos, Yuribel, muchas gracias.



A todas las personas que estuvieron ahí para ayudarme en los momentos que los necesité.

A los miembros del tribunal por su apoyo y confianza, especialmente a Niurvis por ese impulso para la vida.

A Miguel Ángel Chávez, más que un oponente jugaste un papel fundamental en el desarrollo de este trabajo. De ti aprendí mucho en poco tiempo. Gracias.

A todo el 5to año de la facultad 1, la primera FICJ.

A mis compañeros de aula. Especialmente Luisito, Leo, Javier Fuentes.

Al grupo FI-07. A los que están y los que no también. Muchas gracias a todos ustedes.

A Nauria Méndez Trujillo, amiga de muchos momentos inolvidables. Especialmente a ti por todas las risas y los trabajos juntos. Por compartir las alegrías, los nervios, las tristezas. Gracias por haber formado parte de nuestras vidas. Espero que Dios te tenga en la gloria. A ti muchas gracias.

A todas las personas que contribuyeron en este trabajo de una forma u otra.



Declaración de autoría

Declaro por este medio que yo Carel Yosvani Rodríguez Mendoza, con carnet de identidad 95040548029, soy el autor principal del trabajo titulado “**Componente para generar un servicio de geolocalización**” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Declaro que todo lo anteriormente expuesto se ajusta a la verdad, y asumo la responsabilidad moral y jurídica que se derive de este juramento profesional.

Y para que así conste, firmo la presente declaración de autoría en La Habana a los ____ días del mes de junio del año 2019.

Autor:

Carel Yosvani Rodríguez Mendoza

Tutores:

MSc. Leiny Amel Pons Flores

Ing. Alexander García O'reilly



Resumen

En la actualidad las tecnologías evolucionan a gran velocidad, lo que contribuye directamente al aumento del volumen de la información que circula a través de internet. Toda esta información requiere ser almacenada y procesada para posteriormente servir a diversos fines. En este proceso intervienen los sistemas de personalización de información, destinados a brindarles a los usuarios información correspondiente a sus necesidades, entre los que se destacan los sistemas de georreferenciación de información. Cuba no cuenta con servicios propios destinados a la búsqueda de información georreferenciada, por lo que los usuarios acceden directamente a buscadores internacionales para obtener este tipo de información. Como solución a dicha situación se implementó un componente para generar un servicio web de información georreferenciada, disponible para el consumo de aplicaciones de ámbito nacional. La presente investigación detalla además la arquitectura que se utilizó en el desarrollo del sistema y expone los resultados de su validación. Se describen las tecnologías utilizadas que permitieron diseñar e implementar las clases del componente y se siguieron las pautas que propone la metodología AUP-UCI para la construcción de un *software* utilizando las buenas prácticas. En el desarrollo del sistema se utilizó herramientas y tecnologías como NetBeans, PostgreSQL, Spring Boot, Java, Visual Paradigm y UML, las que propiciaron un proceso de implementación y diseño con un alto nivel de calidad.

PALABRAS CLAVES: componente, georreferenciación, información, servicio, web



Índice

Introducción	1
Capítulo 1: Fundamentación teórica	6
1.1. La recuperación de información.....	6
1.1.1. Pre-procesamiento de consulta	6
1.1.1.1. Eliminación de <i>StopWords</i>	7
1.1.1.2. Normalización de consulta (<i>Stemming</i>)	7
1.2. Georreferenciación.....	8
1.2.1. Georreferenciación de la información	9
1.2.2. Modelo geográfico.....	9
1.2.3. Modelo de representación	10
1.2.4. Modelo de almacenamiento.....	11
1.3. Sistema de Información Geográfica (SIG)	11
1.3.1. Definición de SIG	12
1.3.2. Aplicación de los SIG	13
1.4. Soluciones similares.....	14
1.5. Servicios Web	15
1.5.1. Pila de protocolos para servicios web.....	16
1.6. Seguridad.....	18
1.7 Herramientas, tecnologías y metodología	19
1.7.1. Metodología de desarrollo de software	19
1.7.2. Lenguaje de Programación.....	23
1.7.3. Framework de desarrollo	24
1.7.4. Entorno de desarrollo NetBeans.....	24
1.7.5. Herramienta CASE	25
1.7.6. Lenguaje unificado de modelado (UML) 2.1	26
1.7.7. Gestor de base de datos	26



Consideraciones finales del Capítulo 1	27
Capítulo 2: Descripción del componente para generar un servicio web de información georreferenciada.....	28
2.1. Descripción de la propuesta de solución	28
2.2. Modelo conceptual	28
2.3. Especificación de requisitos de software	29
2.3.1. Requisitos funcionales.....	30
2.3.2. Requisitos no funcionales.....	31
2.4. Historias de Usuarios	32
2.5. Diagrama de clases	36
2.6. Arquitectura cliente-servidor.....	37
2.7. Patrones de Diseño.....	38
2.8. Modelo de datos.....	40
2.9. Diagrama de despliegue	41
Consideraciones finales del capítulo 2	42
Capítulo 3: Implementación y validación del Componente para generar un servicio web de información georreferenciada.....	43
3.1. Diagrama de Componentes.....	43
3.2. Estándar de codificación	45
3.3. Estrategia de validación de la propuesta de solución	47
3.3.1. Pruebas de rendimiento (carga y estrés)	47
3.3.2. Pruebas funcionales	50
3.3.3. Prueba de aceptación.....	54
3.3.4. Criterio de expertos	55
Consideraciones finales del capítulo 3	56



Conclusiones	58
Recomendaciones	59
Referencias bibliográficas	60
Anexos.....	66



Introducción

En la actualidad la información existente en todo el mundo se encuentra en constante crecimiento. Con ello se declara el consumo de la misma con una relación directamente proporcional a su volumen. A través de estudios realizados se conoce que luego de alcanzar los 2 billones de usuarios en 2010 y rebasar los 3 billones en 2014, en el 2018 la cifra de personas con acceso a esta información mediante conexión a internet superó los 4 billones (Internet Live Stats, 2018).

Según la información obtenida de *Internet Live Stats* (2018) existen múltiples medios mediante los cuales las personas pueden acceder a la información generada en internet. En la actualidad se evidencia la creciente cifra de casi 2 billones de sitios web implementados, se destacan diversas redes sociales en las que la información se encuentra en flujo constante como Facebook, Twitter e Instagram, y existen plataformas de contenidos multimedia como YouTube. Una de las fuentes de información más relevante es el buscador Google, un sistema de recuperación de información a través del cual en 1998 se realizaban aproximadamente 10 000 búsquedas al día y en el 2018 el total de búsquedas diarias realizadas sobrepasaba la cantidad de 5 billones (Trecebits, 2018).

Las plataformas mencionadas anteriormente, al unísono de otras relevantes en el mundo digital, constituyen fuentes donde los usuarios son los principales generadores de contenido e información. Facebook se describe como una compañía que asume la misión de dar a los usuarios el poder de compartir y hacer el mundo más abierto y conectado. Las personas utilizan esta red social para mantenerse en contacto con amigos y familiares, descubrir lo que está pasando en el planeta y compartir y expresar lo que les importa. El objetivo de Twitter es dar a cada uno el poder de crear y compartir ideas e información al instante y sin barreras. Instagram brinda a todos sus usuarios la posibilidad de inmortalizar el momento que están viviendo y compartirlo con el resto del mundo, solo se necesita tener en posesión un dispositivo inteligente; y por su parte YouTube continúa incrementando la cifra de personas que deciden día a día dedicar su tiempo y crear una inmensa variedad de contenido para el disfrute de todos (Espiritusanto, 2017).

La infografía presentada a continuación, permite evidenciar los datos del crecimiento de internet y la actividad de los usuarios en el mismo. Según especifica Amaya (2017):

- La web móvil recibe 217 nuevos usuarios.

-
- Los usuarios de YouTube suben 48 horas de nuevos videos.
 - Se envían 204.166.667 de correos electrónicos.
 - Google recibe 2.000.000 de búsquedas.
 - Los usuarios de Facebook comparten 684.478 piezas de contenido.
 - Los usuarios de WordPress publican 347 nuevos *posts*.
 - Son creados 571 nuevos sitios.
 - Los usuarios de Foursquare realizan 2.083 *check-ins*.
 - Los usuarios de Instagram comparten 3.600 nuevas fotos.
 - Apple recibe 47.000 descargas de aplicaciones.
 - Los usuarios de Tumblr publican 27.778 nuevos *posts*.
 - Los usuarios de Flickr agregan 3.125 nuevas imágenes.
 - Las organizaciones en Facebook reciben 64.722 nuevos “Me Gusta”.
 - Se envían 100.000 *tweets* en Twitter.
 - Se gastan 272.070 dólares en compras por Internet.

El intercambio de contenido con algunas redes sociales como Instagram resulta necesario realizarlo a través de un *smartphone*. Sin embargo, en el resto de plataformas y redes sociales, no se hace obligatorio el uso de este aparato electrónico, sino que es posible acceder mediante el uso de diversos dispositivos que forman parte del mundo de las TIC (Tecnologías de la Información y las Comunicaciones). Estas tecnologías agrupan elementos y técnicas usadas en el tratamiento y transmisión de informaciones, principalmente de informática, internet y telecomunicaciones. Según plantea Vialart (2016), las mismas procesan, sintetizan, recuperan y presentan información en las más variadas formas; es decir, un conjunto de herramientas, soportes y canales para el tratamiento y acceso a la información con los que se da forma, registran, almacenan y difunden contenidos digitalizados.

Hacer uso cotidiano de las TIC, en la era tecnológica actual, constituye un comportamiento normal en los seres humanos, lo que provoca que varias de sus actividades provean datos. A estos datos almacenados se une la información generada en las redes sociales. Toda la información almacenada es utilizada por sistemas de recuperación de información, donde es procesada para brindar resultados personalizados a los usuarios de acuerdo a sus necesidades (García et al, 2014).

La personalización de información se realiza mediante la utilización de diversos métodos, enfocados en analizar las preferencias de los usuarios como los sitios que visitan, su ubicación y gustos. Entre estos métodos se encuentran la geolocalización y la georreferenciación, los que, aunque disponen de aplicaciones similares, analizan elementos diferentes. En este caso, la investigación y desarrollo del trabajo se realizarán enfocados en la georreferenciación. Expertos de la Universidad Politécnica de Valencia definen la georreferenciación, como un proceso por el cual se dota de un sistema de referencia de coordenadas terreno a una imagen digital que originariamente se encuentra en coordenadas pixel. Dicho de otra manera, la georreferenciación nace a partir de la recolección de mapas creados con información de la geografía del lugar (Gaspar, 2018).

Cuba trabaja arduamente en el desarrollo de las Tecnologías de la Información y las Comunicaciones para el beneficio de la nación. Como consecuencia se han desarrollado en el país herramientas destinadas a la recuperación de información personalizada. A raíz de investigaciones realizadas se arribó a la conclusión de que Cuba no cuenta con servicios propios destinados a la búsqueda de información georreferenciada, por lo que los usuarios acceden directamente a buscadores internacionales para obtener este tipo de información. Entre los buscadores a los que recurren los usuarios se encuentra Google, un buscador perteneciente a una compañía propiedad de Estados Unidos, nación que históricamente ha presentado una fuerte política agresiva hacia Cuba. Esto trae consigo que, en función de sus propios intereses, la información obtenida de dicho buscador (Google), podría ser manipulada en vista de mostrar resultados que afecten negativamente el estado cubano (El Independiente, 2018). Esto impactaría negativamente en la calidad y veracidad de la información obtenida por los usuarios. Además, la nación cubana no controla el contenido ni la calidad de la información mostrada, por lo que se hace necesario e indispensable el desarrollo de un componente nacional para la generación de un servicio que recupere información georreferenciada, que cuente con una base de datos confeccionada a partir de la información proporcionada y verificada por fuentes que defiendan los intereses de la nación.

A partir de la situación problemática existente, se arriba al planteamiento del siguiente **problema científico**:
¿Cómo mejorar la recuperación de información georreferenciada en Cuba?

La recuperación de información georreferenciada se establece como **objeto de estudio**, centrado en el **campo de acción** desarrollo de componentes para recuperación de información georreferenciada en sistemas de recuperación de información.

Como **objetivo general** se propone: Desarrollar un componente de generación de servicio web utilizando la georreferenciación para mejorar el proceso de recuperación de información.

Para dar cumplimiento al objetivo general planteado se definen los siguientes **objetivos específicos**:

- Caracterizar los fundamentos teóricos relacionados con la recuperación de información georreferenciada.
- Definir las tecnologías, herramientas y metodología de desarrollo a utilizar en la implementación de la propuesta de solución.
- Diseñar las funcionalidades del componente de generación de servicio web que se propone.
- Implementar el componente de generación de servicios web utilizando la georreferenciación para mejorar el proceso de recuperación de información georreferenciada.
- Validar el componente de generación de servicios web desarrollado.

La **hipótesis** queda definida de la siguiente forma: Si se desarrolla un componente de generación de servicio web utilizando la georreferenciación, mejorará el proceso de recuperación de la información.

Para darle cumplimiento al objetivo de la investigación se utilizaron métodos de trabajo científico. A continuación, se describen los métodos utilizados:

Métodos teóricos:

- **Histórico - Lógico:** Se empleó con el propósito de constatar teóricamente cómo ha evolucionado en el tiempo la recuperación de la información georreferenciada, así como las herramientas y tecnologías utilizadas en el desarrollo de soluciones de este tipo.
- **Analítico-Sintético:** Para analizar las teorías, documentos, y materiales además de permitir la extracción de los elementos más importantes que se relacionan con los servicios de información georreferenciada.

-
- **Hipotético-Deductivo:** Se utilizó en la realización de la hipótesis planteada y arribar a conclusiones de la misma en el transcurso de la investigación.

Métodos empíricos:

- **Observación:** Posibilita obtener conocimiento acerca del funcionamiento de los sistemas existentes en la actualidad para el desarrollo del componente de generación de servicio de información georreferenciada.
- **Experimentación:** mediante experimentos se evaluará la capacidad de la solución propuesta para mejorar los resultados brindados a los usuarios.

Capítulo 1: Fundamentación teórica

El presente capítulo se enfoca en abordar las principales ideas y conceptos relacionados con la recuperación de información y los sistemas que accionan en este proceso. Del mismo modo se abordan las técnicas y herramientas utilizadas en la recuperación de información. Se realizará un análisis de la metodología, tecnologías y herramientas a utilizar, además de abordar aspectos referentes a los Sistemas de Información Geográfica y Servicios Web.

1.1. La recuperación de información

La Recuperación de Información (RI) es la representación, almacenamiento, organización y acceso a ítems de información. El objetivo principal de la RI es satisfacer la necesidad de información planteada por un usuario en una consulta en lenguaje natural, especificada a través de un conjunto de palabras claves (Reyes et al., 2012).

De acuerdo con la investigación desarrollada por Martínez (2015), existen varios grupos de autores que expresan de maneras diferentes su concepción de recuperación de información. Algunos de los conjuntos de definiciones están muy influenciadas por la tecnología informática, llegándose a olvidar que se puede recuperar información sin procedimientos informáticos. De esta manera el Diccionario Mac Millan de Tecnología de la Información considera a la recuperación de información como las “técnicas empleadas para almacenar y buscar grandes cantidades de datos y ponerlos a disposición de los usuarios” (Martínez, 2015).

El problema de la RI, según Pons (2018), se plantea como “dada una necesidad de información y un conjunto de documentos, ordenar los documentos de más a menos relevantes para esa necesidad y presentar un subconjunto de aquellos de mayor relevancia”. El mismo autor señala que a pesar de la trayectoria de la RI desde 1950, es en la actualidad donde adquiere un rol más importante debido al creciente valor que adquiere la información para los procesos de la sociedad.

1.1.1. Pre-procesamiento de consulta

La función del pre-procesamiento es convertir el texto de entrada (secuencia de caracteres) a un formato adecuado para su tratamiento por el resto del sistema. En el desarrollo de la solución de este trabajo, se

escogieron tres algoritmos de pre-procesamiento para aumentar la efectividad y rapidez del sistema. Los algoritmos se clasifican en dos tipos:

- Eliminación de *Stop-Words* (Palabras Vacías).
- Normalización.

1.1.1.1. Eliminación de *StopWords*

Según Ramírez (2012a) las palabras que aparecen con frecuencia entre las consultas no son buenas para la recuperación de información, por lo que no son consideradas y se les llama *stopwords*:

- Los artículos, los pronombres, las preposiciones, y las conjunciones son candidatos naturales.
- Algunos verbos, adverbios, y adjetivos se podían tratar como *stopwords*.
- Los términos específicos de un dominio se podían tratar como *stopwords*.

Se suele tener una lista de palabras que no son buenos términos de indexación llamada *STOPLIST*, Lista de Palabras Vacías o Diccionario Negativo. La consulta de entrada es comprobada con la *STOPLIST* y se eliminan los términos que aparecen en ella.

Ventajas:

- Las palabras vacías aparecen mucho y su lista de referencias es muy grande:
 - Si las quitamos la consulta será más pequeña.
- Mejora la eficiencia, porque hay una mejor selección de palabras claves.

1.1.1.2. Normalización de consulta (*Stemming*)

Stemming significa hallar el *stem* (lema) de las palabras. En la implementación de la solución se lematiza durante la indexación, debido a que se tiene la ventaja de que mejora la eficiencia y se comprimen los índices de búsqueda. El usuario especifica una palabra en una consulta pero solamente una variante de esta palabra está presente en la base de datos (Ramírez, 2012b).

Un *stem* (lema) es la porción que queda de una palabra después de retirar los afijos (prefijos y sufijos). Un ejemplo típico es la palabra **conecta** que es el *stem* para las variantes: conectadas, conectando, conexión y conexiones. Consiste en convertir todas las palabras parecidas a una forma común.

Se pretende agrupar términos en un solo término de indexación como:

- Plurales y género.
- Formas del gerundio.
- Sufijos de tiempo para los verbos.
- Prefijos, como de negación (in), ya no pertenece (ex), interno (intra), externo (inter), etc.

Técnicas:

- Búsqueda en una tabla que tiene todas las derivaciones de un término común.
- Obtención de la variedad de sucesores.
- N-gramas.
- Algoritmos de eliminación de afijos.

Ventajas:

- Reduce el tamaño del índice, ya que el número de palabras también es reducido.
- Mejora la importancia de la recuperación porque las variantes de la misma palabra se reducen a un concepto común.
- Aumenta la eficiencia.
- La indexación se realiza más rápidamente.

1.2. Georreferenciación

La georreferenciación es un aspecto fundamental en el análisis de datos geoespaciales, es la base para la correcta localización de la información de mapa y, por ende, de la adecuada fusión y comparación de datos procedentes de diferentes sensores en diferentes localizaciones espaciales y temporales. Es un concepto relativamente nuevo que aún no es reconocido por la Real Academia de Lengua Española (RAE), sin embargo, es comúnmente empleado en múltiples textos (Beltrán, 2012). La definición más aceptada hace

referencia al posicionamiento con el que se define la localización de un objeto espacial (representado mediante punto, vector, área, volumen) en un sistema de coordenadas determinado. Es un proceso frecuentemente realizado en los Sistemas de Información Geográficos (Fuentes, 2015).

1.2.1. Georreferenciación de la información

Los datos geográficos son el elemento fundamental dentro de un SIG (Sistema de Información Geográfico). Estos presentan una estrecha relación con el resto de los elementos que son necesarios para la visualización y el análisis. El factor organizativo y las personas tienen como principal objetivo buscar y extraer de ellos la mayor cantidad de datos posible. Al mismo tiempo la confiabilidad de los datos es un elemento muy importante para un SIG, lo que representa la exactitud de cualquier análisis que se realice (Fuentes, 2015).

La información geográfica se divide para su estudio en una componente espacial y una temática (Olaya, 2014). La primera proporciona una ubicación dentro de un sistema de referencia, mientras la segunda establece las características particulares del fenómeno o proceso que tiene lugar en la localización establecida por la componente espacial.

El surgimiento de los SIG ha derivado en diversos problemas, uno de los fundamentales ha sido convertir la información que se puede extraer de un área geográfica en datos que puedan ser utilizados por estos sistemas. El principal elemento ha sido el infinito número de detalles existentes en la naturaleza, mientras que la representación y el almacenamiento de dicha realidad, son finitos. Este proceso de traducción de la realidad en un conjunto de valores numéricos pasa por tres etapas de acuerdo con Fuentes (2015):

1. Establecimiento de un modelo geográfico.
2. Establecimiento de un modelo de representación o modelo de datos.
3. Establecimiento de un modelo de almacenamiento.

1.2.2. Modelo geográfico

El punto de partida en la creación del dato geográfico es establecer un modelo conceptual que permita interpretar el espacio y la variable estudiada, así como la variación de la misma a lo largo del espacio. Fuentes (2015) plantea que los dos modelos geográficos más importantes son:

-
- **Campo:** Es un modelo de variación dentro de un espacio n-dimensional, donde para cada punto dentro de dicho espacio existe un valor para la variable estudiada. Habitualmente consiste en la asociación de un único valor escalar a cada vector dentro del campo. No obstante, en casos como en la medición del movimiento de los vientos se asocia a cada vector dentro del espacio, otro vector compuesto por dos valores: la dirección y la velocidad en la que sopla dicho viento. En los SIG estas situaciones se tratan frecuentemente utilizando varios campos escalares, cada uno de ellos en distintas capas. Este modelo resulta adecuado para el estudio de variables continuas tales como temperatura del aire, presión atmosférica o elevación.
 - **Entidades discretas:** Concibe un espacio geográfico vacío, en el que se ubican elementos (entidades) que lo van llenando. Una de sus principales características es que un punto del espacio puede pertenecer a varios elementos o no pertenecer a ninguno. Las entidades pueden ser cualquier elemento geométrico existente en el espacio definido, tales como puntos, líneas o polígonos. En el caso de espacios de al menos tres dimensiones pueden ser figuras volumétricas. La modelización de variables como la presencia de vías de comunicación y puntos de interés se realizará mejor con este modelo.

1.2.3. Modelo de representación

Una vez establecido el modelo geográfico a utilizar, el siguiente paso es reducir las propiedades de dicho modelo a un conjunto finito de características que al ser almacenadas describan la realidad estudiada. Para ello se emplean los modelos de representación o de datos, estos se clasifican en dos grupos: ráster y vectorial (Olaya, 2014).

El modelo ráster subdivide todo el espacio y lo maneja como un conjunto ordenado de entidades elementales. Estas se pueden entender como celdas regulares de una malla, en cada una de las cuales se almacena un valor que describe la zona ocupada por dicha celda. La exactitud de este modelo es inversamente proporcional al área que representa cada celda. Frecuentemente estos valores son transformados en un color mediante una escala, dando lugar a una imagen.

El modelo vectorial define el espacio mediante un conjunto de figuras geométricas (puntos, líneas y polígonos) con valores asociados. La disposición de estos está en correspondencia con los accidentes geográficos presentes en la zona que se estudia. Se define así la componente espacial en la propia primitiva

y la componente temática con sus valores asociados. Es importante destacar que una única entidad puede necesitar para su representación de varias primitivas geométricas (Fries, 2017).

1.2.4. Modelo de almacenamiento

Los modelos de almacenamiento son el paso final en el proceso de convertir el medio circundante en datos manejables dentro de un SIG. Estos definen cómo almacenar los valores en un soporte digital de manera más eficiente. Los aspectos fundamentales tenidos en cuenta por estos modelos son: minimizar el espacio ocupado por los datos y maximizar la eficiencia de los cálculos realizados sobre ellos (Olaya, 2014). El tipo de modelo de almacenamiento a utilizar está restringido por el modelo de representación.

En el caso del modelo ráster, lo más sencillo es utilizar una matriz de datos. El empleo de esta estructura presenta un formato muy intuitivo. Además, es cómoda de implementar y resulta sencilla de recorrer, lo que facilita las operaciones que se realicen sobre ella (Egenhofer et al., 1991). En este caso el hecho de que valores consecutivos dentro de la matriz tiendan a tener valores similares, conlleva al desperdicio de gran cantidad de espacio de almacenamiento. Por tal razón se utilizan procesos de codificación sobre la matriz o estructuras de datos más complejas como los árboles cuaternarios (Sanet, 1994). Estos últimos dividen el espacio en cuadrantes de forma recursiva, alcanzando mayor profundidad en las zonas que contengan mayor cantidad de elementos distintos.

Para el caso vectorial los modelos no priorizan reducir el volumen de almacenamiento sino optimizar las operaciones sobre los datos. En este sentido el factor fundamental es la facilitación de acceso a los datos y para ello se emplean estructuras de indexación espacial. Las más comunes son el R-Tree y sus variantes R+-Tree y R*-Tree (Fuentes, 2015). Estas pueden ser consideradas extensiones multidimensionales de los árboles binarios. En ellos, los puntos cercanos entre sí son almacenados en el mismo nodo hoja representado como un rectángulo encapsulador mínimo. Los nodos son progresivamente agrupados siguiendo el mismo principio hasta el nivel superior, la raíz.

1.3. Sistema de Información Geográfica (SIG)

El desarrollo de la solución del problema de investigación de este trabajo no necesita hacer uso de Sistemas de Información Geográfica, sin embargo, debido al estrecho vínculo existente entre ambos, se aborda en este epígrafe puntos fundamentales para el entendimiento de los mismos.

El uso de la información geográfica en la toma de decisiones se encuentra presente en muchas de nuestras actividades cotidianas. Al seleccionar inconscientemente la ruta para ir al trabajo, la ruta de vacaciones, ir a una reunión de amigos, dirigirse a una tienda usando un teléfono inteligente, se están tomando decisiones que implican el análisis de la información geográfica sin ser conscientes de ello. Cuando este tipo de análisis o de toma de decisiones se lleva a cabo empleando computadoras se suele hacer por medio de lo que se conoce como Sistemas de Información Geográfica.

1.3.1. Definición de SIG

Los Sistemas de Información Geográfica, también conocidos como Sistemas de Información Georreferenciada, surgen en la década de los sesenta, orientados en un principio a las áreas del conocimiento asociadas a la geografía, la geodesia, la cartografía y las ciencias de la tierra en general. Con el devenir de los años y el desarrollo de las nuevas TIC, estas herramientas han incidido de forma decisiva en otras esferas de la actividad humana.

Un SIG es un conjunto de herramientas compuestas por *hardware*, *software*, datos y usuarios, que permite capturar, almacenar, administrar y analizar información digital, así como realizar gráficos y mapas, y representar datos alfanuméricos. También puede verse como un modelo informatizado de la realidad geográfica para satisfacer unas necesidades de información concretas, esto es, crear, compartir y aplicar información útil basada en datos y en mapas (Fries et al., 2017).

De acuerdo con el análisis realizado por Luna (2010), un Sistema de Información Geográfica es una tecnología basada en computadoras de propósitos generales para almacenar, manejar y explotar datos geográficos en forma digital. Es un sistema que tiene un conjunto de subsistemas que sirven para: la captura, el almacenamiento, el análisis, la visualización y graficación de diversos conjuntos de datos espaciales georreferenciados.

Un Sistema de Información Geográfica según Ornay (2013) supone la posibilidad de controlar de una forma dinámica y flexible variables como índices de audiencia televisivos, sondeos preelectorales y problemática social. La fusión de estas tres variantes referida a la definición de un SIG, resulta en la elaboración de un concepto más amplio y abarcador. Un SIG se define en un conjunto de técnicas y herramientas informáticas para la gestión y manejo de datos espaciales, que permiten georreferenciar información aprovechando su

componente geográfica. Un SIG brinda nuevas perspectivas de análisis, lo que lo convierte en un poderoso sistema de apoyo a la planificación y la toma de decisiones.

1.3.2. Aplicación de los SIG

Más del 70% de la información existente se puede georreferenciar, a raíz de esto el planteamiento de los SIG cuentan con amplio campo de aplicaciones (Muñiz, 2010). El empleo como sistemas de apoyo a la planificación y la toma de decisiones así lo demuestra:

- **Sector bancario:** Localización de red de sucursales en función de las características de la población. Estudio de modelos de mercado potenciales. Estudio de riesgos en la gestión de seguros. Seguimiento de inversiones y de los resultados del mercado bancario en su dimensión territorial.
- **Sector de estudios de mercado:** Segmentaciones de mercado, distribución territorial de la población y de sus características socioeconómicas.
- **Sector sanitario:** Seguimiento de estudios epidemiológicos. Planificación de la red de asistencia sanitaria en relación con la población que habita una zona. Análisis de la distribución geográfica de los perfiles sanitarios de la población.
- **Sector logístico:** Gestión de flota. Planificación y optimización de rutas. Determinación de centros de distribución. Posicionamiento de puntos de venta y análisis de itinerarios de recogida y suministros. Control de los envíos.
- **Sector de las telecomunicaciones:** Planificación de las redes de telefonía móvil, análisis de cobertura del medio, etc.
- **Sector de la comunicación:** Información sectorial para su transmisión gráfica hacia los sectores deseados de la opinión pública. Análisis de los efectos de las campañas de publicidad y promoción. Teletrabajo, educación a distancia, tiempo libre, información sobre ocio, etc.
- **Sector de franquicias:** Localización de nuevos puntos de venta, captación de clientes potenciales, etc.
- **Sector medioambiental:** Para realizar inventarios de suelos o controlar el tipo del uso del mismo (González, 2010).

Los SIG son sistemas que utilizan datos georreferenciados para su funcionamiento. Esto constituye un factor muy importante a tener en cuenta en el desarrollo del presente trabajo, debido a que los SIG representan una gran parte de las aplicaciones que harán uso del componente resultante de esta investigación.

1.4. Soluciones similares

Google Maps: Aplicación creada por Google Inc., su funcionalidad principal es mostrar mapas del mundo. Entre sus características está la sugerencia de rutas para llegar a un punto indicado por el usuario en dependencia de los medios de transporte de que disponga, es decir, si caminará utilizará el transporte público, bicicleta o automóvil. El servicio ofrece la posibilidad de realizar búsquedas sobre negocios locales y obtener sugerencias sobre los mismos (Google Inc., 2012). A pesar de las ventajas que ofrece el uso de esta aplicación, la misma depende de la conexión constante a internet para su correcto funcionamiento. Además, la información mostrada sobre Cuba pierde precisión y en muchos casos no se encuentra actualizada.

OsmAnd: Es un visor de mapas de código abierto. Desarrollado para ser empleado tanto conectado directamente a internet (*on-line*) como de manera aislada (*off-line*). Posee entre sus principales funcionalidades guiar giro por giro al usuario hasta llegar a la posición indicada por este. Trabaja por defecto con la cartografía del proyecto OpenStreetMap (OSM) aunque es posible configurarlo para que utilice Google Maps. Permite la búsqueda y edición de puntos de interés, pero para ello debe conectarse a internet. Sus funcionalidades *off-line* se basan en la descarga de mapas vectoriales de las regiones deseadas (OsmAnd Team, 2012).

MapDroyd: Es una aplicación concebida para la visualización de mapas *off-line*. La obtención de los mismos se realiza mediante la descarga de los ficheros necesarios desde un servidor. Utiliza el formato *MicroMap* para el almacenamiento de la información geográfica en lugar de los tradicionales *tiles*, reduciendo el espacio de almacenamiento requerido. Sus funcionalidades son limitadas, no considerando por ejemplo la búsqueda de puntos de interés. Utiliza los datos del proyecto OSM para compilar sus archivos de mapas (MapDroyd, 2010).

El análisis de las soluciones similares anteriores arribó a resultados que resaltan sus características negativas:

-
- MapDroyd: Necesita descargar ficheros del servidor para su funcionamiento y presenta largos períodos de tiempo sin actualizar las bases de datos.
 - OsmAnd: Largos tiempos de carga.
 - Google Maps: Tiempos variables de actualización de datos, baja calidad en las imágenes y altos requerimientos de *hardware* para su óptimo funcionamiento.

Tomando como referencia estos factores se llevará a cabo el desarrollo de la solución al problema de esta investigación. Por tanto, se implementará un componente para generar un servicio web que contribuya a que las aplicaciones que lo utilicen presenten tiempos de actualización estables de la base de datos, veloces tiempos de carga y navegación fluida. Además, en la implementación se desarrollarán aspectos como la seguridad y la precisión de la información brindada al cliente.

1.5. Servicios Web

Cruz (2017) define servicio *web* como una tecnología que utiliza un conjunto de protocolos y estándares para llevar a cabo el intercambio de información entre un cliente y un servidor, mientras que EcuRed (2018) plantea que un servicio *web* consiste en un programa informático que permite la comunicación y el intercambio de datos entre aplicaciones y sistemas heterogéneos en entornos distribuidos. Los servicios *web* son por ende un conjunto de funcionalidades expuestas en una intranet o a través de internet, por y para aplicaciones y computadoras sin la intervención humana.

De acuerdo con Cruz (2017), la utilización de servicios *web* posibilita diversas ventajas como las mencionadas a continuación:

- Aporta interoperabilidad entre aplicaciones de *software* independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Fomenta los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Permite que servicios y *software* de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.

La implementación de la solución presente se basa directamente en el desarrollo de un servicio web que provea la georreferenciación de sitios especificados por el cliente. Consecuentemente se aborda el tema en función de explicar los aspectos fundamentales que constituyen los servicios *web*.

1.5.1. Pila de protocolos para servicios web

La pila de protocolos es una colección de protocolos y estándares para redes de computadores que son utilizados para definir, localizar, implementar y concebir la interacción entre servicios web. La pila de protocolos está compuesta por cuatro áreas principalmente:

Servicio de Transporte: Tiene la responsabilidad de transportar mensajes entre las aplicaciones de red y los protocolos en los cuales se incluyen HTTP, SMTP, FTP y el más reciente Protocolo de Intercambio de Bloques Extensible (BEEP).

Mensajería XML: Codifica mensajes en un formato común en Lenguaje de Marcado Extensible (XML), logrando que sean entendidos en cualquier extremo de una conexión de red. Actualmente, esta área incluye protocolos tales como XML-RPC, SOAP y REST.

Descripción del servicio: Describe la interfaz pública de un servicio web específico. El formato de interfaz web WSDL es típicamente usado para este propósito.

Lenguaje de Marcado Extensible

XML es un formato simple de texto muy flexible derivado del Estándar de Lenguaje de Marcado Generalizado (SGML) (ISO 8879). Está diseñado para afrontar los retos de la gran edición electrónica, además de desempeñar un papel cada vez más importante en el intercambio de una amplia variedad de datos en la web y en otros lugares (Cárdenas, 2013).

WSDL

Esta es una especificación que detalla la localización de los servicios y los métodos que el servicio ofrece. Es abierta, basada en XML y describe las interfaces y las instancias de servicios web en la red. La misma es ampliable, de modo que permite detallar los puntos finales independientemente de los formatos de

mensaje o de los protocolos de red que se utilicen para comunicarse. Las empresas pueden poner a disposición de sus servicios web los documentos WSDL mediante UDDI, WSIL o divulgando los URL a su WSDL mediante correo electrónico o sitios Web (IBM Knowledge Center, 2018).

SOAP

SOAP es un estándar basado en XML para la transferencia de mensajes en HTTP y otros protocolos de Internet. Asume la ventaja de ser ligero para el intercambio de información en un entorno descentralizado y distribuido (IBM Knowledge Center, 2018). Se basa en XML y consta de tres partes:

- Un sobre que define una infraestructura para describir el contenido del mensaje y cómo procesarlo.
- Un conjunto de normas de codificación para expresar instancias de tipos de datos definidos por la aplicación.
- Una convención para representar llamadas y respuestas a procedimiento remoto.

SOAP permite el enlace y la utilización de servicios web encontrados definiendo una ruta de mensaje para el direccionamiento de mensajes. Se puede utilizar SOAP para consultar UDDI para servicios web.

REST

BBVA (2018) define el estándar REST como cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON. Plantea que es una alternativa en auge a otros protocolos estándar de intercambio de datos como SOAP, que disponen de una gran capacidad, pero también mayor complejidad. REST es el estándar más lógico, eficiente y utilizado en la creación de APIs para servicios de internet.

Una vez desarrollada una breve descripción de algunos de los estándares más recurridos en la implementación de servicios web, se utilizará en la solución del problema tratado en este trabajo el estándar REST. Esta decisión se toma sobre la base de la eficiencia y bajo nivel de complejidad brindado por dicho estándar. Además, el protocolo REST separa totalmente la interfaz de usuario del servidor y el almacenamiento de datos, mejora la portabilidad de la interfaz a otro tipo de plataformas, aumenta la escalabilidad de los proyectos y permite que los distintos componentes de los desarrollos se puedan evolucionar de forma independiente.

REST permite migrar a otros servidores o realizar todo tipo de cambios en la base de datos, siempre y cuando los datos de cada una de las peticiones se envíen de forma correcta. Siempre se adapta al tipo de sintaxis o plataformas con las que se estén trabajando, lo que ofrece una gran libertad a la hora de cambiar o probar nuevos entornos dentro del desarrollo. Es indispensable que las respuestas a las peticiones se hagan siempre en el lenguaje de intercambio de información usado, en el caso de la propuesta de solución se utilizará JSON.

1.6. Seguridad

La publicación de servicios accesibles a todo el mundo supone una clara ventaja comercial, pero también puede convertirse en un serio problema de seguridad si no se toman medidas a tiempo. La seguridad de la información persigue proteger la información de posibles accesos y modificaciones no autorizadas. Para conocer cuáles amenazas pueden comprometer los servicios web con los que interactúan los clientes, es requerido, primeramente, tener el conocimiento de qué aspectos resultan considerables a la hora de diseñar un sistema de seguridad en dichos servicios (Pérez, 2013).

Pérez (2013) define como amenaza a todo elemento o acción capaz de atentar contra la seguridad de la información. Las amenazas surgen a partir de la existencia de vulnerabilidades y sólo puede existir accionando sobre una vulnerabilidad que pueda ser aprovechada y comprometa la seguridad de un sistema de información.

Las principales amenazas que pueden materializarse en los servicios web son:

- Acceso inapropiado en favor de entidades no identificadas correctamente.
- Interceptación de mensajes.
- Corrupción del canal de comunicación y de los recursos.
- Acceso inapropiado a los recursos.
- Denegación de servicio.
- No repudio.
- Uso de la información.
- Confidencialidad.

Con el conocimiento de las amenazas existentes es posible reducir o cancelar su impacto y estar preparado para los ataques que puedan ocasionarse. Una vez realizado el análisis de las principales amenazas que se pueden explotar, si algunas de estas se materializaran podrían ocurrir en los servicios web diversos ataques; los más comunes serían de acuerdo con Pérez (2013):

- Alteración de los mensajes.
- Ataques a la confidencialidad.
- *Spoofing*.
- Ataques de denegación de servicio.
- Inyecciones XML.

La solución a desarrollar no necesita la implementación de métodos de seguridad. Aunque presente libre acceso de conexión, no existe la posibilidad de realizar inyecciones a la base de datos, debido a que la implementación solo permitirá a los clientes realizar operaciones de tipo GET (obtener). Además, las consultas realizadas a la base de datos se efectuarán utilizando el *framework* Java Persistence API (JPA), el que presenta protocolos integrados de seguridad para hacer las consultas.

1.7 Herramientas, tecnologías y metodología

El desarrollo de sistemas informáticos está determinado por las metodologías, herramientas y tecnologías que se utilicen. El principal objetivo de estas, es guiar y facilitar el proceso de creación, garantizando a su vez la calidad del producto final. A continuación se definen las utilizadas por el autor en la implementación de la solución propuesta.

1.7.1. Metodología de desarrollo de software

Una metodología de desarrollo de *software* tiene entre sus objetivos el aumento de la calidad del *software* que se produce. En la actualidad no existe una metodología de carácter universal, debido a que cada una se debe adaptar al as características propias del proyecto. Cada proyecto debe valorar y realizar la selección de una metodología que se adapte a sus necesidades.

Programación extrema (XP)

Es una metodología ágil basada en una serie de valores y de prácticas de buenas maneras que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y facilidades para enfrentar los cambios. Además, se define especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre (Penadés, 2013).

Una de las principales ventajas que ofrece la programación extrema es que se adapta al desarrollo de sistemas pequeños y grandes; optimiza el tiempo de desarrollo; el código es sencillo y entendible, además de la poca documentación a elaborar para el desarrollo del sistema. Por otra parte, no se tiene la definición del costo y el tiempo de desarrollo; el sistema va creciendo después de cada entrega al cliente y nadie puede decir que el cliente no querrá una función más; se necesita de la presencia constante del cliente, lo cual en la realidad es muy difícil de lograr.

Herramientas de XP

Historias de usuarios: Son tarjetas físicas en las cuales se anota una descripción de una funcionalidad del sistema, en una oración, se le da un número y un título para ser identificada.

Casos de prueba de aceptación: Son tarjetas que se elaboran para realizar las pruebas de cada historia de usuario.

Tarea de ingeniería: Son tarjetas que se elaboran para ayudar y simplificar la programación de una historia de usuario.

Gutierrez (2012) plantea que el ciclo de vida ideal consta de 4 fases:

Exploración: En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las

herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo.

Planificación: En esta fase el cliente establece la prioridad de cada historia de usuario y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El plan de entrega está compuesto por iteraciones de no más de tres semanas.

Construcción: La fase de construcción requiere de las tarjetas CRC, estas permiten desprenderse del método de trabajo basado en procedimientos y trabajar con una metodología basada en objetos. En esta fase se realiza el desarrollo de las iteraciones, donde las historias de usuario se descomponen en tareas de programación o en tareas de ingeniería.

Prueba: Esta fase garantiza el correcto funcionamiento de las funcionalidades que el sistema posee. Las pruebas se dividen en dos: en pruebas unitarias y pruebas de aceptación. Las de aceptación verifican que todas las funcionalidades devuelvan el valor correspondiente a los requisitos funcionales del sistema a implementar.

AUP UCI

La metodología AUP propone 7 disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno), se decide para el ciclo de vida de los proyectos de la UCI tener 7 disciplinas también, pero a un nivel más atómico que el definido en AUP. Los flujos de trabajos: Modelado de negocio, Requisitos y Análisis y diseño en AUP están unidos en la disciplina Modelo, en la variación para la UCI se consideran a cada uno de ellos disciplinas. Se mantiene la disciplina Implementación, en el caso de Prueba se desagrega en 3 disciplinas: Pruebas Internas, de Liberación y Aceptación. Las restantes 3 disciplinas de AUP asociadas a la parte de gestión para la variación UCI se cubren con las áreas de procesos que define CMMIDEV v1.3 para el nivel 2, serían CM (Gestión de la configuración), PP (Planeación de proyecto) y PMC (Monitoreo y control de proyecto) (ibíd.) (Sánchez, 2015).

Fases

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, a la que llamaremos Ejecución y se agrega la fase de Cierre (Sánchez, 2015).

Escenarios

Esta versión de AUP define cuatro escenarios en los que se puede ubicar el desarrollo de una aplicación de acuerdo a sus características.

- Escenario No 1: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que puedan modelar una serie de interacciones entre los trabajadores del negocio/actores del sistema (usuario), similar a una llamada y respuesta.
- Escenario No 2: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio.
- Escenario No 3: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad.
- Escenario No 4: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una Historias de usuario (HU) no debe poseer demasiada información.

Consideraciones de la metodología

Se escoge **AUP-UCI** en su escenario 4 para el desarrollo del presente trabajo, ya que se adapta al ciclo de vida definido para la actividad productiva de la UCI. Esta metodología se ajusta a los requerimientos del proyecto, además de que se cuenta con la disposición del cliente en todo momento y una clara definición del negocio en cuestión.

1.7.2. Lenguaje de Programación

Un lenguaje de programación, es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. En la actualidad existen diversos lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. PHP y Java se destacan como los dos lenguajes de programación de servicios web más utilizados.

PHP, versión 7.0.1

El lenguaje PHP es un lenguaje que se acopla al HTML (páginas web) para definir procedimientos que ha de realizar el servidor web. PHP soporta el trabajo con interfaz de entrada común (CGI, en inglés *Common Gateway Interface*), permite procesar la información de formularios, generar páginas con contenidos dinámicos, y el trabajo con *cookies*. Es un lenguaje de programación de estilo clásico, con variables, sentencias condicionales, bucles y funciones. El resultado es normalmente una página HTML. Permite la posibilidad de ejecutarlo en diferentes tipos de servidores, entre ellos Apache. Además, incorpora un gran número de extensiones, aumentando las potencialidades del lenguaje. Con una sintaxis similar a los lenguajes de programación C, Java y Perl, este lenguaje hace posible el desarrollo de páginas web que se generen dinámicamente y de forma rápida, y que pueden ejecutarse en distintos tipos de servidores web (PHP, 2012).

Una de las ventajas de este lenguaje es que tiene la capacidad de conexión con la mayoría de los manejadores de base de datos como: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras, además de que es un lenguaje multiplataforma. PHP se caracteriza por ser un lenguaje muy rápido y fácil de aprender, que ofrece una solución simple y universal para las paginaciones dinámicas de la web de fácil programación.

Java 8

Java es un lenguaje de programación desarrollado por la compañía Sun Microsystems. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito del internet como en la informática en general.

Una de las ventajas de Java es que es un lenguaje de programación de distribución libre, no es necesario pagar una licencia para poder comenzar a desarrollar en él. Así mismo es un lenguaje muy completo y poderoso, pues posee librerías y utilidades muy completas que facilitan la programación.

Consideraciones de los lenguajes de programación

Después de un estudio realizado sobre los lenguajes de programación anteriormente mencionados y analizar las características y ventajas de cada uno, se decidió utilizar Java. La elección se basa en que dicho lenguaje posee un conjunto de funciones y librerías que facilitan el desarrollo de servicios web.

1.7.3. Framework de desarrollo

Un *framework* puede considerarse como una estructura de *software*, compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Básicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Spring Framework 5

Spring reactivo representa una iniciativa de toda la plataforma para ofrecer soporte reactivo en todos los niveles de la pila de desarrollo: web, seguridad, datos, mensajes, etc. Spring Framework 5 ofrece esta visión al proporcionar una nueva pila web reactiva llamada Spring WebFlux. Este *framework* ofrece el tiempo de ejecución, el modelo de programación y la arquitectura eficaces para los requisitos de desarrollo de servicios web, características que lo destacan como ideal para la implementación del presente trabajo.

1.7.4. Entorno de desarrollo NetBeans

NetBeans es una herramienta de desarrollo para aplicaciones, escrita puramente sobre la base de la tecnología Java, de modo que puede ejecutarse en cualquier ambiente que ejecute Java. Es un producto de código abierto, con todos los beneficios del *software* disponible en forma gratuita, el cual ha sido examinado por una comunidad de desarrolladores. Aparte de la filosofía de distribución y desarrollo que respalda a NetBeans, el IDE ofrece a los desarrolladores numerosas ventajas, en la creación de nuevas aplicaciones multiplataforma.

Es un IDE de desarrollo maduro y con soporte a gran cantidad de librerías, lo cual lo convierte en un entorno de desarrollo ideal, por lo que permite a los programadores el uso de las mismas de una forma muy fácil e intuitiva. Es muy sencillo de utilizar y exige muy poco conocimiento por parte del usuario para su uso.

La decisión de utilizar NetBeans se basa en sus diversas ventajas como el manejo automático de la memoria, gratuidad y desarrollo de aplicaciones web dinámicas. Es completamente *software* libre y tiene versiones para los principales sistemas operativos como: Windows, MacOS y LINUX que es el sistema operativo donde se va a implementar (Hernández, 2012).

1.7.5. Herramienta CASE

Son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de un *software*. Estas herramientas son de utilidad en cada parte del ciclo de vida del mismo pues en la planificación de tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

Visual Paradigm 8.0

Es una poderosa herramienta CASE que utiliza UML para el modelado, es la herramienta por excelencia para ser utilizada en un ambiente de *software* libre. Permite crear tipos diferentes de diagramas en un ambiente totalmente visual. Es muy sencillo de usar, fácil de instalar y actualizar. Soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Visual Paradigm ofrece:

Un entorno de creación de diagramas para UML 2.1.

- Diseño centrado en casos de uso y enfocado al negocio que genera un *software* de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.

-
- Disponibilidad de múltiples versiones, para cada necesidad.
 - Disponibilidad de integrarse en los principales IDEs.
 - Disponibilidad en múltiples plataformas.

1.7.6. Lenguaje unificado de modelado (UML) 2.1

El Lenguaje Unificado de Modelado (UML) es un lenguaje de visualización, especificación, construcción y documentación de los artefactos de un sistema. Ofrece un estándar para confeccionar modelos o descripciones del mismo, los cuales incluyen aspectos conceptuales, tales como procesos de negocios y funciones del sistema, y aspectos concretos, como son expresiones de lenguajes de programación, esquemas de bases de datos y componentes de *software* reutilizables.

Se utiliza para definir un sistema de *software*, detallar los artefactos en el sistema, documentar y construir. Está compuesto por elementos gráficos que se relacionan entre sí dando lugar a diagramas.

Se toma la decisión de utilizar UML 2.1. El mismo admite crear un lenguaje de modelado utilizado tanto por humanos como por máquinas y mejora el soporte a la planeación y al control de proyectos. Ostenta una alta reutilización y minimización de costos y mejora los tiempos totales de desarrollo hasta en un 50 por ciento.

1.7.7. Gestor de base de datos

Los Gestores de Bases de Datos (GBD) permiten crear y mantener una base de datos, además actúan como interfaz entre los programas de aplicación y el sistema operativo. El objetivo principal es proporcionar un entorno eficiente a la hora de almacenar y recuperar la información de la base de datos. Estos *software* facilitan el proceso de definir, construir y manipular bases de datos para diversas aplicaciones (Cobo, 2016).

PostgreSQL

PostgreSQL es un sistema de GBD objeto-relacional, de propósito general, multiusuario y de código abierto, que soporta gran parte del estándar SQL y ofrece modernas características como consultas complejas, disparadores, vistas, integridad transaccional, control de concurrencia multiversión. Puede ser extendido por el usuario añadiendo tipos de datos, operadores, funciones agregadas, funciones ventanas y funciones recursivas, métodos de indexado y lenguajes procedurales (PostgreSQL, 2018).

PostgreSQL soporta los tipos de datos cláusulas, funciones y comandos de tipo estándar SQL92/SQL99 y extendidos propios de PostgreSQL, puede operar sobre distintas plataformas incluyendo Linux, Windows, Unix, Solaris y MacOS X, presenta buen sistema de seguridad mediante la gestión de usuarios, grupos de usuarios y contraseñas. Sostiene buena escalabilidad, ya que es capaz de ajustarse al número de CPU y a la cantidad de memoria disponible de forma óptima, soportando una mayor cantidad de peticiones simultáneas a la base de datos de forma correcta. Las ventajas expuestas sustentan la base de la elección de PostgreSQL en el desarrollo de la solución.

PgAdmin

Como herramienta gráfica para gestionar el GBD PostgreSQL se utilizó la herramienta PgAdmin III en su versión 1.20.0. PgAdmin está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. Soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código para la parte del servidor y un agente para lanzar *scripts* programados (Robinson, 2017).

Consideraciones finales del Capítulo 1

Con el objetivo de sustentar la investigación, en el presente capítulo se ha llevado a cabo un estudio de algunos elementos teóricos, a partir del cual se obtienen las siguientes consideraciones finales:

- El marco teórico antes expuesto, ha contribuido a un mayor entendimiento del problema planteado.
- El uso de tecnologías y herramientas como los sistemas de georreferenciación y los servicios web son tendencias actuales y necesarias que deben utilizarse en la solución a desarrollar.
- Se escogió AUP-UCI como metodología para guiar el desarrollo del componente para generar un servicio web de información georreferenciada.
- Se definieron los algoritmos de pre-procesamiento de consultas que intervienen en el proceso de recuperación de información georreferenciada.

Capítulo 2: Descripción del componente para generar un servicio web de información georreferenciada

En el presente capítulo se presenta y describe la propuesta de solución y se identifican los requisitos funcionales y no funcionales. Se detalla la arquitectura de *software*, las historias de usuario, el diagrama de despliegue y los artefactos generados para darle solución al problema planteado en la investigación.

2.1. Descripción de la propuesta de solución

El componente para generar un servicio *web* de información georreferenciada tiene como objetivo mejorar el proceso de recuperación de información. El funcionamiento del componente inicia con el envío de una consulta desde el cliente a través del protocolo HTTP. La consulta se captura y se le realiza el procedimiento de **pre-procesamiento** correspondiente. Una vez pre-procesada la consulta, se obtiene un arreglo de palabras que se utilizan en la realización de la búsqueda de la información georreferenciada en la base de datos. En este proceso se utiliza una base de datos que contiene información relacionada a los lugares de interés nacional. La búsqueda se realiza comparando los elementos resultantes del pre-procesamiento de la consulta con palabras predefinidas asociadas a los campos que componen la base de datos. Obtenidos los elementos de la base de datos se procede al envío de la respuesta con dicha información utilizando el formato JSON, procedimiento que concluye el proceso.



Figura 1. Propuesta de solución. Fuente: Elaboración propia.

2.2. Modelo conceptual

El siguiente modelo conceptual describe los procesos importantes del componente y las relaciones entre ellos. El modelo conceptual describe la interacción del cliente con el servicio, el que gestiona la información de la base de datos. A continuación, se describen las clases que componen el modelo conceptual:

- **Ciente:** Actor que realiza la petición del servicio y recibe información como respuesta.
- **Servicio:** Representa el servicio con el que interactúa el Cliente.
- **Generador de información georreferenciada:** Tipo de servicio que genera la información enviada al Cliente.
- **Base de datos:** Representa la base de datos que contiene la información utilizada.
- **Fichero de configuración:** Archivo que representa las configuraciones pertinentes del servicio.

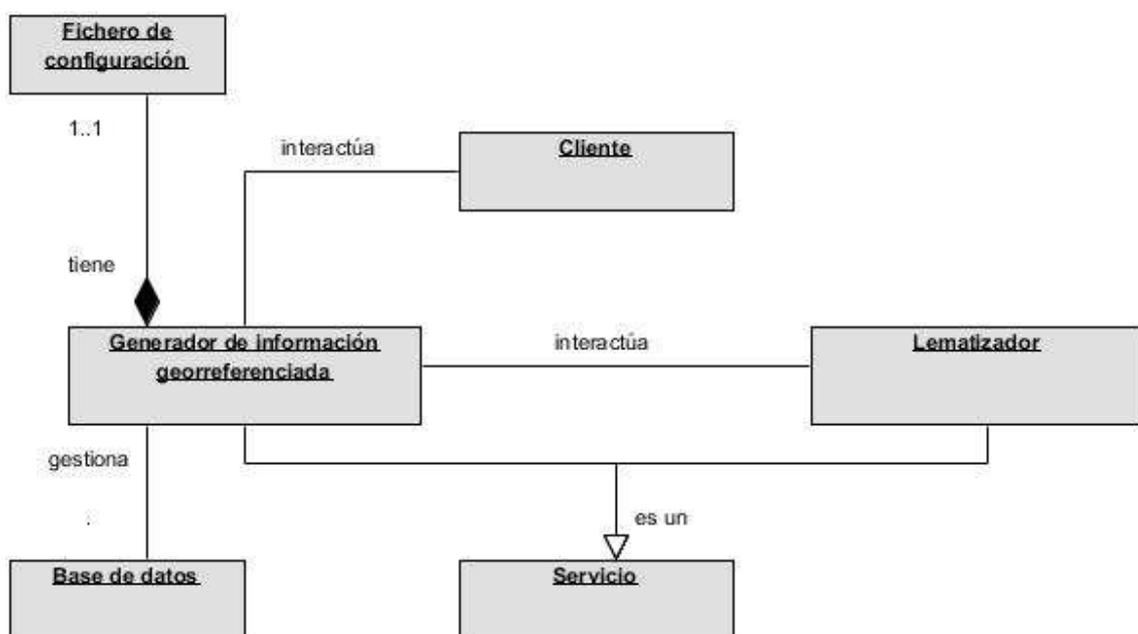


Figura 2. Modelo conceptual. Fuente: Elaboración propia.

2.3. Especificación de requisitos de software

El análisis de requisitos es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refina en detalle el ámbito del *software*, y se crean modelos de los requisitos de datos, flujo de información y control, y del comportamiento operativo. Se analizan soluciones alternativas y se asignan a diferentes elementos del *software*. Permite al desarrollador o desarrolladores especificar la función y el rendimiento del *software*, indica la interfaz del *software* con otros elementos del sistema y establece las restricciones que debe cumplir el *software* (Pressman, 2002).

2.3.1. Requisitos funcionales

Los requisitos funcionales (RF) son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse en situaciones específicas (Sommerville, 2011).

Se realizaron encuestas y entrevistas que resultaron en la definición de 3 requisitos funcionales. A los requisitos se les asignó una prioridad teniendo en dependencia a la importancia fijada por el cliente a partir de sus necesidades y la dificultad con la que se realiza su implementación respectivamente.

Tabla 1: Requisitos Funcionales. Fuente: Elaboración propia.

Requisitos	Descripción	Prioridad
RF1: Pre-procesar consulta.	El sistema captura la consulta y le realiza los procedimientos de pre-procesamiento correspondientes.	Alta
RF2: Obtener coordenadas.	El sistema realiza consultas a la base de datos y devuelve la información correspondiente a las coordenadas.	Alta
RF3: Obtener elevación.	El sistema realiza consultas a la base de datos y devuelve la información correspondiente a la elevación.	Alta
RF4: Obtener código postal.	El sistema realiza consultas a la base de datos y devuelve la información correspondiente al código postal.	Alta
RF5: Obtener código de área.	El sistema realiza consultas a la base de datos y devuelve la información correspondiente al código del área.	Alta
RF6: Obtener extensión de área.	El sistema realiza consultas a la base de datos y devuelve la información correspondiente a la extensión del área.	Alta

RF7: Obtener cantidad de habitantes.	El sistema realiza consultas a la base de datos y devuelve la información correspondiente a cantidad de habitantes.	Alta
RF8: Obtener información general.	El sistema realiza consultas a la base de datos y devuelve la información correspondiente a toda la información correspondiente al lugar requerido.	Alta

2.3.2. Requisitos no funcionales

Los requisitos no funcionales (RNF) son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares. Los requerimientos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del mismo (Sommerville, 2011). A partir de un análisis se identificaron 13 RnF, distribuidos en especificaciones de hardware, software y seguridad, los cuales se relacionan a continuación:

Software

RnF 1: Servidor de bases de datos relacional PostgreSQL 9.4

RnF 2: Sistema operativo Ubuntu GNU/Linux 17.10 o superior para servidor web.

RnF 3: sistema operativo Ubuntu GNU/Linux 17.10 o superior para servidor de base de datos.

Hardware

RnF 4: El servidor de base de datos debe poseer una capacidad mínima de 100 GB de almacenamiento.

RnF 5: El servidor de aplicaciones web debe poseer una capacidad mínima de 80 GB de almacenamiento.

RnF 6: Los servidores web y de base de datos deben poseer como mínimo 2 GB de memoria RAM.

Seguridad

RnF 7: Se empleará un sistema de autenticación mediante el uso de *tokens*.

Eficiencia

RnF 8: El sistema debe permitir que los usuarios interactúen con él de manera concurrente.

RnF 9: El tiempo de demora de una petición al servidor debe ser menor de cinco (5) segundos aproximadamente.

Restricciones de implementación y diseño

RnF 10: El sistema deberá ser implementado en el lenguaje de programación JAVA versión 8.0 o superior.

RnF 11: Se empleará la herramienta de desarrollo NetBeans 8.0.

RnF 12: El sistema empleará el sistema gestor de base de datos PostgreSQL 9.4.

2.4. Historias de Usuarios

Las historias de usuario (HU) son la técnica utilizada para especificar los requisitos del *software*. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (Penadés, 2013).

Relación de las Historias de Usuarios

Como parte del proceso de trabajo dentro de la fase de exploración se identificaron un total de 8 Historias de Usuarios, las cuales se muestra a continuación:

Tabla 2. HU: Pre-procesar consulta. Fuente: Elaboración propia.

Historia de Usuario			Número: 1
Nombre: Pre-procesar consulta			
Prioridad: Alta	Complejidad: Alta	Estimación: 288 h	Iteración:1
Programador responsable: Carel Yosvani Rodríguez Mendoza			

Descripción: Realizar pre-procesamiento de la consulta realizada por el cliente.

Información adicional: Se captura el mensaje HTTP enviado por el cliente. Si el cliente introduce parámetros incorrectos se lanzará una excepción informando el error de los mismos. La consulta pasa por el proceso de normalización y posteriormente se le eliminan las *StopWords*.

Tabla 3. HU: Obtener coordenadas. Fuente: Elaboración propia.

Historia de Usuario			Número: 2
Nombre: Obtener coordenadas			
Prioridad: Alta	Complejidad: Media	Estimación: 84 h	Iteración: 2
Programador responsable: Carel Yosvani Rodríguez Mendoza			
Descripción: Se consulta la base de datos y se obtiene la información de las coordenadas correspondiente al lugar especificado en la consulta.			
Información adicional: Se comparan los <i>tokens</i> resultantes del pre-procesamiento de la consulta con las palabras predefinidas y asociadas al campo de coordenadas de la base de datos. Se obtiene la información de las coordenadas y se devuelven en formato JSON.			

Tabla 4. HU: Obtener elevación. Fuente: Elaboración propia.

Historia de Usuario			Número: 3
Nombre: Obtener elevación			
Prioridad: Alta	Complejidad: Media	Estimación: 84 h	Iteración: 2
Programador responsable: Carel Yosvani Rodríguez Mendoza			
Descripción: Se consulta la base de datos y se obtiene la información de la elevación correspondiente al lugar especificado en la consulta.			
Información adicional: Se comparan los <i>tokens</i> resultantes del pre-procesamiento de la consulta con las palabras predefinidas y asociadas al campo de elevación de la base de datos.			

Se obtiene la información de la elevación y se devuelven en formato JSON.

Tabla 5. HU: Obtener código postal. Fuente: Elaboración propia.

Historia de Usuario			Número: 4
Nombre: Obtener código postal			
Prioridad: Alta	Complejidad: Media	Estimación: 84 h	Iteración: 2
Programador responsable: Carel Yosvani Rodríguez Mendoza			
Descripción: Se consulta la base de datos y se obtiene la información del código postal correspondiente al lugar especificado en la consulta.			
Información adicional: Se comparan los <i>tokens</i> resultantes del pre-procesamiento de la consulta con las palabras predefinidas y asociadas al campo del código postal de la base de datos. Se obtiene la información de la elevación y se devuelven en formato JSON.			

Tabla 6. HU: Obtener código de área. Fuente: Elaboración propia.

Historia de Usuario			Número: 5
Nombre: Obtener código de área			
Prioridad: Alta	Complejidad: Media	Estimación: 84 h	Iteración: 3
Programador responsable: Carel Yosvani Rodríguez Mendoza			
Descripción: Se consulta la base de datos y se obtiene la información del código del área correspondiente al lugar especificado en la consulta.			
Información adicional: Se comparan los <i>tokens</i> resultantes del pre-procesamiento de la consulta con las palabras predefinidas y asociadas al campo del código de área de la base de datos. Se obtiene la información del código del área y se devuelven en formato JSON.			

Tabla 7. HU: Obtener extensión de área. Fuente: Elaboración propia.

Historia de Usuario			Número: 6
Nombre: Obtener extensión de área			
Prioridad: Alta	Complejidad: Media	Estimación: 84 h	Iteración: 3
Programador responsable: Carel Yosvani Rodríguez Mendoza			
Descripción: Se consulta la base de datos y se obtiene la información de la extensión correspondiente al lugar especificado en la consulta.			
Información adicional: Se comparan los <i>tokens</i> resultantes del pre-procesamiento de la consulta con las palabras predefinidas y asociadas al campo extensión de la base de datos. Se obtiene la información de la extensión y se devuelven en formato JSON.			

Tabla 8. HU: Obtener cantidad de habitantes. Fuente: Elaboración propia.

Historia de Usuario			Número: 7
Nombre: Obtener cantidad de habitantes			
Prioridad: Alta	Complejidad: Media	Estimación: 84 h	Iteración: 3
Programador responsable: Carel Yosvani Rodríguez Mendoza			
Descripción: Se consulta la base de datos y se obtiene la información de la cantidad de habitantes correspondiente al lugar especificado en la consulta.			
Información adicional: Se comparan los <i>tokens</i> resultantes del pre-procesamiento de la consulta con las palabras predefinidas y asociadas al campo cantidad de habitantes de la base de datos. Se obtiene la información de la cantidad de habitantes y se devuelven en formato JSON.			

Tabla 9. HU: Obtener información general. Fuente: Elaboración propia.

Historia de Usuario			Número: 8
Nombre: Obtener información general			

Prioridad: Alta	Complejidad: Media	Estimación: 84 h	Iteración: 3
Programador responsable: Carel Yosvani Rodríguez Mendoza			
Descripción: Se consulta la base de datos y se obtiene la información general correspondiente al lugar especificado en la consulta.			
Información adicional: Se comparan los <i>tokens</i> resultantes del pre-procesamiento de la consulta con las palabras predefinidas y asociadas al campo información de la base de datos. Se obtiene la información general y se devuelven en formato JSON.			

2.5. Diagrama de clases

En la fase de diseño del software se hace un refinamiento del análisis y se centra en cómo van a ser implementados los requerimientos, logrando adaptar el diseño para que se ajuste al entorno de implementación, persiguiendo así el desarrollo de una arquitectura estable y sólida (UNAD, 2016). El diagrama de clases muestra los atributos y métodos de cada clase, representando de forma sencilla la colaboración y las responsabilidades de cada una de ellas, entorno al sistema que conforman. El siguiente diagrama de clases contiene las principales clases con sus respectivos atributos.

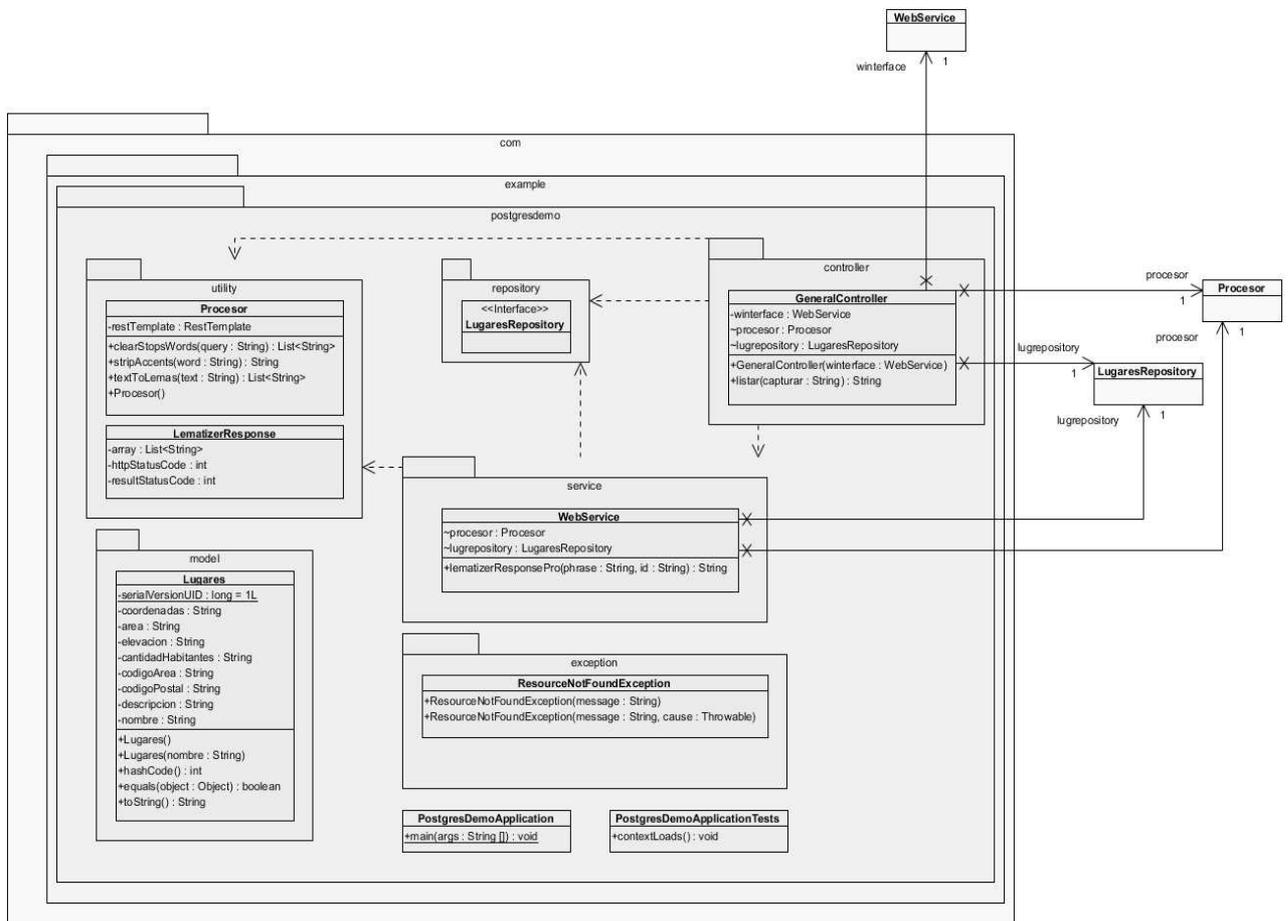


Figura 3. Diagrama de clases. Fuente: Elaboración propia.

2.6. Arquitectura cliente-servidor

En el servicio web a implementar se utilizará el tipo o modelo de arquitectura de entorno web: Cliente-Servidor. Este modelo arquitectónico se caracteriza por el envío de mensajes a través del cliente para realizar una determinada petición al servidor, a partir de lo cual, este último envía uno o varios mensajes con la respuesta (servicio). Se basa en la existencia de dos tipos de aplicaciones que se ejecutan de manera independiente (Llorente, 2010).

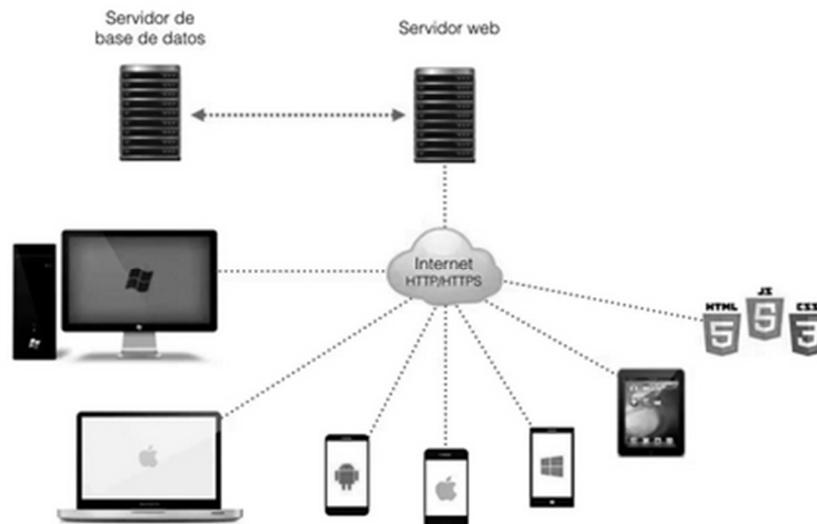


Figura 4. Arquitectura Cliente-Servidor. Fuente: Elaboración propia.

Es una arquitectura que se emplea para sistemas distribuidos pues de esta forma el servicio web es publicado por el servidor y consumido por distintas aplicaciones clientes mediante protocolos de conexión de red, los clientes pueden hacer una o más peticiones y a su vez el servidor le responde con el resultado de la misma. Con esto se garantiza el acceso centralizado a los datos que éste gestiona y que son almacenados en la base de datos.

2.7. Patrones de Diseño

Los patrones de diseño son una descripción de un problema/solución, con un nombre y que puede emplearse en otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas. Por lo que se puede concluir que los patrones de diseño no son más que pautas definidas que aplican ciertos estilos con el objetivo de encaminar la construcción de un *software* (Larman et al., 2003).

Patrones GRASP

Los patrones GRASP describen la asignación de responsabilidades a objetos, expresados en forma de patrones. Para la construcción de la solución propuesta se seleccionaron los patrones siguientes (Cárdenas, 2016):

- **Experto:** Encargado de asignar una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir la responsabilidad. Es la asignación de responsabilidades a cada clase, facilitando que los sistemas sean más fáciles de entender, mantener y ampliar. Presentando la oportunidad de reutilizar los componentes en futuras aplicaciones. Está presente en las clases *Procesor* y *WebService* de manera que cada responsabilidad está asignada a la clase que cuenta con la información necesaria para realizarla.
- **Controlador:** Se encarga de asignar la responsabilidad del manejo de los eventos de un sistema a una clase. Es un objeto de interfaz no destinado al usuario que se encarga de manejar un evento del sistema. Permitiendo, un mayor potencial de los componentes reutilizables. Presente en la clase *GeneralController*, que es la encargada de atender los eventos del sistema que son generados por un actor externo y se asocian a operaciones del sistema.
- **Bajo acoplamiento:** Es asignar una responsabilidad para mantener el bajo acoplamiento. Mantener las clases lo menos ligadas posibles, es la medida de la fuerza con que una clase está conectada a otras. Soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, acrecentando la oportunidad de una mayor productividad. Está presente en todas las clases controladoras del proyecto.
- **Alta cohesión:** Se refiere a asignar una responsabilidad de modo que la cohesión siga siendo alta. Es la medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una clase con alta cohesión es fácil darle mantenimiento, entenderla y reutilizarla. Está presente en todas las clases del proyecto.

Patrones Gang of Four (GOF)

Describen las formas comunes en que diferentes tipos de objetos, pueden ser organizados para trabajar unos con otros. Gestionan la relación entre clases y la formación de estructuras de mayor complejidad.

Además, permiten crear grupos de objetos para ayudar a realizar tareas complejas (Craig, 2003). Los patrones GOF utilizados en el módulo se relacionan a continuación:

Registry: Este patrón es útil para los desarrolladores en la Programación Orientada a Objetos. Es un medio sencillo y eficiente de compartir datos y objetos en la aplicación, sin la necesidad de conservar numerosos parámetros o utilizar variables globales. Se aplica en la clase de configuración (*application.properties*) que es la encargada de guardar las variables globales del módulo, como drivers de conexión a la base de datos y rutas bases del proyecto.

2.8. Modelo de datos

Un modelo de datos consiste en un conjunto de herramientas conceptuales para describir la representación de la información en términos de datos. Los modelos de datos comprenden aspectos relacionados con: estructuras y tipos de datos, operaciones y restricciones (Zorrilla, 2011). La propuesta de solución utiliza una base de datos de prueba, la cual genera el modelo de datos representado en la Figura 5, el que está estructurado en 1 tabla que se explica a continuación:

lugar: Atributo que represente los lugares almacenados en la tabla. Es la llave primaria de la tabla.

coordenada: Atributo que represente las coordenadas correspondientes a los lugares almacenados en la tabla.

codigo_postal: Atributo que represente los códigos postales correspondientes a los lugares almacenados en la tabla.

codigo_area: Atributo que represente los códigos de área correspondientes a los lugares almacenados en la tabla.

elevacion: Atributo que represente las elevaciones correspondientes a los lugares almacenados en la tabla.

extension_area: Atributo que represente las extensiones de área correspondientes a los lugares almacenados en la tabla.

cantidad_habitantes: Atributo que represente la cantidad de habitantes correspondiente a los lugares almacenados en la tabla.

informacion: Atributo que represente la información adicional correspondientes a los lugares almacenados en la tabla.

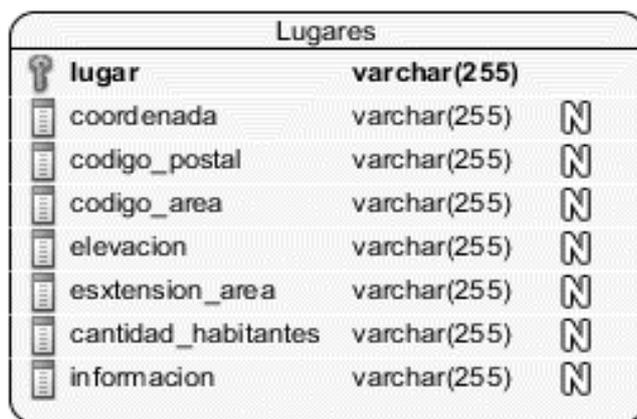


Figura 5. Modelo de datos. Fuente: Elaboración propia.

2.9. Diagrama de despliegue

Un diagrama de despliegue es la forma de mostrar la configuración de nodos de procesamientos en tiempo de ejecución y los componentes que en ellos residen. Estos nodos forman la topología de *hardware* sobre la que se ejecuta el sistema. Este diagrama se preocupa principalmente de la distribución, entrega e instalación de las partes que constituye el sistema físico (Taboada et al., 2009).

El diagrama mostrado en la Figura 2, permite capturar los elementos de configuración del procesamiento de la información y las conexiones que existen entre estos elementos. Está formado por nodos, dispositivos y conectores. Los nodos son elementos de procesamiento con al menos un procesador, memoria y posiblemente otros dispositivos; los dispositivos a su vez son nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela, y los conectores expresan el protocolo o conector utilizado entre los elementos del diagrama.

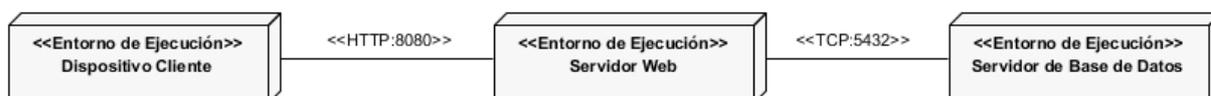


Figura 6. Diagrama de despliegue. Fuente: Elaboración propia.

En este diagrama se define el nodo Dispositivo Cliente desde donde se originan las peticiones mediante el protocolo HTTP al servicio web alojado en el Servidor Web. Este servidor debe utilizar el sistema operativo Ubuntu GNU/Linux 17.10 o superior y sus propiedades mínimas deben ser 2GB de memoria RAM y 80 GB de almacenamiento. El sistema mantendrá una comunicación con el servidor de base de datos por el protocolo TCP y el puerto 5432, este correrá en una PC 2 GB de RAM y utilizará SGBD PostgreSQL 9.4 con una capacidad mínima de 100 GB de disco duro.

Consideraciones finales del capítulo 2

Con el desarrollo del presente capítulo se arribó a las siguientes conclusiones:

- El proceso de levantamiento de requisitos de software permitió la definición de 8 RF y 13 RnF, lo que propició el desarrollo de 8 historias de usuario, establecidas para implementarse en 3 iteraciones.
- Los artefactos generados en correspondencia con la metodología AUP-UCI permitieron una comprensión clara y precisa para el desarrollo de la propuesta de solución.
- La descripción de la propuesta de solución, permitió definir la arquitectura utilizada, sirviendo como antesala a su implementación.

Capítulo 3: Implementación y validación del Componente para generar un servicio web de información georreferenciada

En el presente capítulo se realiza una descripción de las funcionalidades implementadas en la capa de servicios web y su organización una vez desplegado el mismo. Se realiza un análisis de los casos de prueba, validando la aceptación de la solución propuesta para el componente de generación de servicios web de información georreferenciada. Durante el proceso de desarrollo del servicio web se utilizarán pautas de codificación con vistas a generar un código de alta calidad, siendo de gran importancia para la implementación de los mismos.

3.1. Diagrama de Componentes

El diagrama de componentes proporciona una visión física de la construcción del sistema. Muestra la organización de los componentes software, sus interfaces y las dependencias entre ellos (Ramos, 2016). Basándose en la arquitectura de software que propone el marco de trabajo, seleccionada en la fase de análisis y diseño, la Figura 10 muestra el diagrama de componentes del componente para generar un servicio web de información georreferenciada, a continuación se describen los elementos que componen el diagrama:

Tabla 10. Descripción de los componentes del sistema. Fuente: Elaboración propia.

Componentes	Descripción
applications.properties	Contiene las configuraciones del servicio y conexión a la BD.
Lematizador	Servicio externo que se consume por la aplicación en el proceso de normalización de consulta.
GeoRef.sql	Base de datos contenedora de la información utilizada por la aplicación.
PostgresDemoApplications.java	Clase principal encargada de ejecutar el proyecto.

com	example	postgresdemo	controller	GeneralController.java	Clase controladora que gestiona todos los procesos del proyecto.
			utility	Procesor.java	Clase contenedora de los métodos de pre-procesamiento.
				LematizerResponse.java	Clase que contiene métodos auxiliares de la clase Procesor.
			repository	LugaresRepository.java	Interface que gestiona los métodos de interacción con la base de datos.
			model	Lugares.java	Clase entidad generada a partir de la base de datos.
			service	WebService.java	Clase que gestiona el servicio y contiene los métodos de búsqueda en la base de datos.
			exception	ResourceNotFoundException.java	Clase encargada de gestionar las excepciones ocurridas.

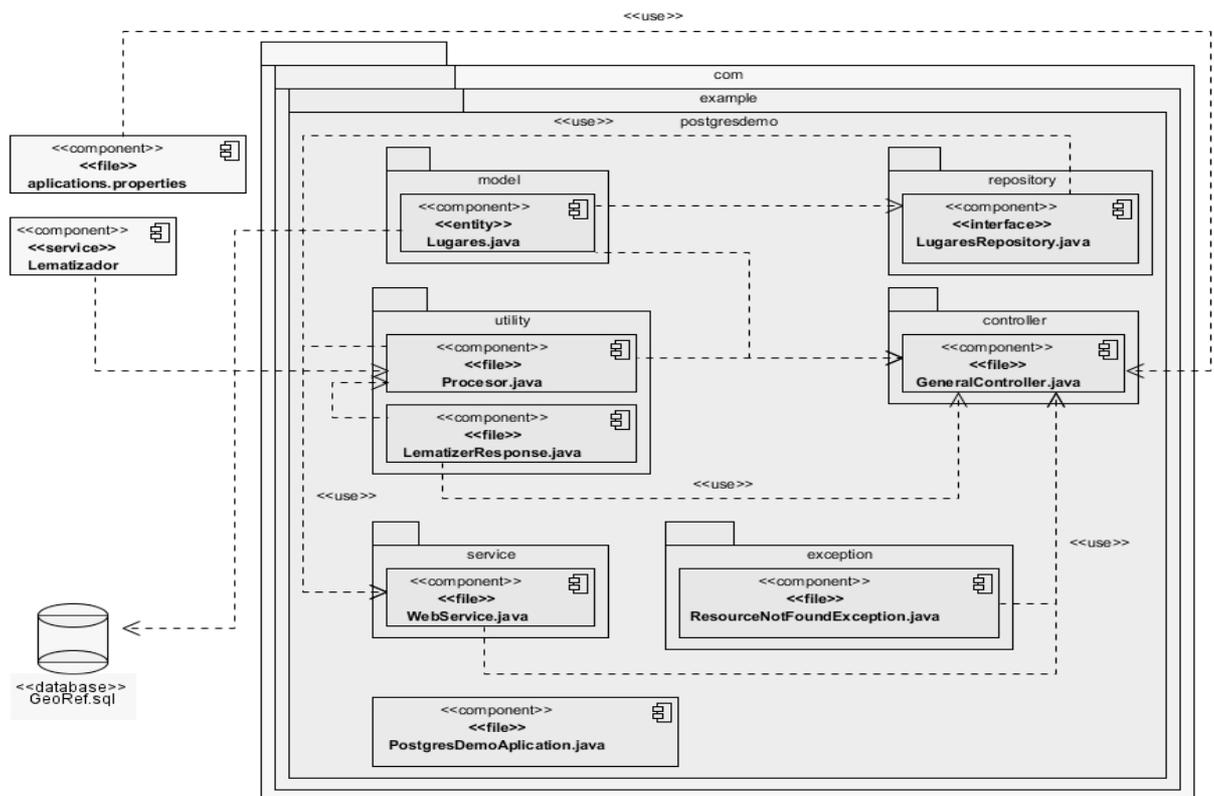


Figura 7. Diagrama de componentes. Fuente: Elaboración propia.

3.2. Estándar de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código, incluyendo pautas sobre la nomenclatura de las variables, clases y paquetes; la correcta indentación del código, entre otros aspectos. La elaboración y utilización de los estándares de codificación permiten a todos los desarrolladores realizar la implementación siguiendo las mismas pautas y así el resto de los desarrolladores puedan entender el código, facilitando así que un proyecto de *software* se convierta en un sistema legible, uniforme y fácil de mantener.

Conversión de nomenclatura

El nombre de las variables, clases y métodos se encuentra escrito en inglés. Las variables siempre comienzan con minúscula y en caso de nombres compuestos, la primera letra de la segunda palabra comienza con mayúscula. Las clases siempre comienzan con mayúscula, en caso de nombres compuestos,

las palabras que componen el nombre de la clase comienzan con mayúsculas. En las funciones se utilizó el estilo *lowerCamelCase*, donde siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula, exceptuando la primera palabra. Los parámetros son separados por espacio, luego de la coma que los separa.

Indentación

La indentación se realiza a un TAB, con un equivalente a dos espacios para los bloques de instrucciones. El uso de las llaves de apertura y de cierre se realizó en una nueva línea. Se utilizó un máximo de 100 caracteres por cada línea de código. Cuando una expresión no cabe en una línea, se separa de acuerdo con los siguientes principios:

- Romper después de una coma.
- Romper antes de un operador.
- Alinear la nueva línea con el comienzo de la expresión al mismo nivel de la línea anterior.

```
WordController(@NonNull WordRepository words) {  
    if (words == null) {  
        throw new NullPointerException("words");  
    } else {  
        this.words = words;  
    }  
}
```

Figura 8. Ejemplo de estándar de codificación (conversión de nomenclatura e indentación).

Comentarios

Todo todos los comentarios del código se realizaron en el idioma inglés y no deben nunca incluir caracteres especiales.

Formatos de los comentarios en la implementación

Los comentarios de implementación se utilizan para dar descripciones de código y facilitar información adicional que no es legible en el código mismo. Los comentarios de implementación se delimitan entre `/*...*/` o se utiliza la variante `//`. Se utilizan tres formatos de comentarios: Bloque de comentario, comentario de una línea y comentario de fin de línea.

Comentarios de una línea: Utilizado para hacer comentarios cortos de una única línea al nivel del código al que se hace referencia.

Comentario de fin de línea: El delimitador de comentario // puede convertir en comentario, una línea completa o una parte de una línea, como se muestra en la Figura 9. No es utilizado para hacer comentarios de varias líneas consecutivas, aunque podrá utilizarse en líneas consecutivas para comentar secciones de código.

```
public class CountryNameJpaController implements Serializable {  
  
    //EMF was modified to making lighter the execution of the project  
    public CountryNameJpaController() {  
        emf = jpaUtility.getEntityManagerFactory();  
    }  
}
```

Figura 9. Ejemplo de comentarios de fin de línea.

3.3. Estrategia de validación de la propuesta de solución

Las pruebas de validación son un proceso de ejecución de un programa, con la intención de descubrir un error. Consiste en la realización de pruebas funcionales, de carga y estrés, aceptación y validación de los expertos. La realización de las pruebas reduce el número de errores no detectados y disminuye el tiempo transcurrido entre la aparición de un error y su detección. A continuación, se documentan los resultados de las pruebas realizadas al Componente para generar un servicio web de información georreferenciada, con el objetivo de evaluar su calidad y correcto funcionamiento.

3.3.1. Pruebas de rendimiento (carga y estrés)

La prueba de rendimiento se diseña para poner a prueba el rendimiento del software en tiempo de ejecución, dentro del contexto de un sistema integrado. Específicamente las pruebas de carga según Pressman (2002) consisten en simular una carga de trabajo similar y superior a la que tendrá cuando el sistema esté funcionando, con el fin de detectar si el software instalado cumple con los requerimientos de muchos usuarios simultáneos. Además, se comprueba si el *hardware* es capaz de soportar la cantidad de peticiones esperadas.

Por otra parte las pruebas de estrés evalúan la robustez y la confiabilidad del *software* sometiéndolo a condiciones de uso extremas. Estas condiciones incluyen el envío en exceso de peticiones y la ejecución en condiciones de hardware limitadas. El objetivo de estas pruebas es la saturación del programa hasta un punto de quiebre donde aparezcan defectos potencialmente peligrosos (Pressman, 2002).

En la realización de las pruebas de carga y estrés se utilizó como herramienta el Apache JMeter en su versión 2.10, diseñado para pruebas de carga de comportamientos funcionales y la medición del rendimiento. El ambiente de prueba se integró como se muestra a continuación:

Sistema de prueba (PC servidor):

- Sistema Operativo: Linux Mint 19.
- Microprocesador: AMD A10-8700P @1.8GHz-2.5GHz.
- Memoria RAM: 12.00 GB.
- Disco Duro: 1024 GB.

Software instalado en la PC:

- Tipo de servidor web: Apache Tomcat.
- Servidor de BD: PostgreSQL 9.4.

Los parámetros configurados en la herramienta Apache JMeter se definieron con 100 usuarios conectados de manera concurrente, logrando un ambiente de simulación donde se realizaron peticiones al servicio. En la figura se observan los resultados finales obtenidos por el sistema.

Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes	Latency	Connect Time...
977	06:51:04.284	Usuarios conectados 1-18	HTTP Request	14370		1793	14370	0
978	06:51:05.077	Usuarios conectados 1-10	HTTP Request	13901		1793	13901	0
979	06:51:07.887	Usuarios conectados 1-24	HTTP Request	12213		1793	12213	0
980	06:51:08.132	Usuarios conectados 1-44	HTTP Request	12537		1793	12537	0
981	06:51:09.750	Usuarios conectados 1-41	HTTP Request	11406		1793	11406	0
982	06:51:09.373	Usuarios conectados 1-80	HTTP Request	12028		1793	12028	0
983	06:51:09.848	Usuarios conectados 1-54	HTTP Request	11659		1793	11659	0
984	06:51:10.102	Usuarios conectados 1-46	HTTP Request	11440		1793	11440	1
985	06:51:09.425	Usuarios conectados 1-13	HTTP Request	12171		1793	12171	4
986	06:51:13.393	Usuarios conectados 1-6	HTTP Request	9248		1793	9248	0
987	06:51:13.773	Usuarios conectados 1-86	HTTP Request	9282		1793	9282	0
988	06:51:15.142	Usuarios conectados 1-15	HTTP Request	8083		1793	8083	0
989	06:51:15.350	Usuarios conectados 1-56	HTTP Request	7962		1793	7962	0
990	06:51:14.902	Usuarios conectados 1-31	HTTP Request	8589		1793	8589	0
991	06:51:17.536	Usuarios conectados 1-67	HTTP Request	7025		1793	7025	0
992	06:51:18.979	Usuarios conectados 1-10	HTTP Request	5956		1793	5956	0
993	06:51:20.670	Usuarios conectados 1-44	HTTP Request	4667		1793	4667	0
994	06:51:21.402	Usuarios conectados 1-80	HTTP Request	4283		1793	4283	0
995	06:51:21.508	Usuarios conectados 1-54	HTTP Request	4270		1793	4270	0
996	06:51:21.157	Usuarios conectados 1-41	HTTP Request	4716		1793	4716	0
997	06:51:23.314	Usuarios conectados 1-56	HTTP Request	3259		1793	3259	0
998	06:51:25.340	Usuarios conectados 1-44	HTTP Request	1901		1793	1901	0
999	06:51:26.576	Usuarios conectados 1-56	HTTP Request	1190		1793	1190	0
1000	06:51:27.767	Usuarios conectados 1-56	HTTP Request	2014		1793	2014	0

Figura 10. Apache JMeter: Prueba de carga y estrés. Fuente: Elaboración propia.

Las variables que miden el resultado de las pruebas se describe a continuación:

- Usuarios: total de usuarios.
- # Muestras: El número de peticiones.
- Media: El tiempo medio transcurrido en milisegundos para un conjunto de resultados.
- Mín: El mínimo tiempo transcurrido en milisegundos para las muestras de la URL definida.
- Máx: El máximo tiempo transcurrido en un milisegundo para las muestras de la URL definida.
- % Error: Porcentaje de las peticiones con errores.
- Rendimiento: Rendimiento medido en base a peticiones por segundo/minuto/hora.
- Kb/s Recibidos: Rendimiento medido en Kbytes por segundos.

Tabla 11. Resultado de la prueba de carga y estrés. Fuente: Elaboración propia.

Usuarios	# Muestras	Media	Mín	Máx	% Error	Rendimiento (peticiones/segundos)	Kb/s Recibidos
100	1000	30224	2957	33162	0 %	1.8	1.4

Para un total de 100 usuarios conectados de forma concurrente, el sistema respondió 1000 peticiones al servidor en un promedio de 30.224 segundos, lo que equivale a 1.8 peticiones por segundo. En base a las prestaciones del *hardware* donde se realizaron las pruebas y a la cantidad de peticiones por segundo efectuadas al servidor, se considera el resultado el resultado de las pruebas como satisfactorio.

3.3.2. Pruebas funcionales

Las pruebas funcionales están centradas en comprobar que las funcionalidades descritas en el documento de requisitos del sistema se cumplen con la implementación realizada. A este tipo de pruebas también se les denomina pruebas de comportamiento o de caja negra, debido a que los analistas enfocan su atención a las respuestas del sistema de acuerdo a los datos de entrada y sus resultados en los datos de salida (Pressman, 2002).

Tabla 12. Caso de prueba Procesar solicitud. Fuente: Elaboración propia.

Caso de prueba Pre-procesar consulta.	
Código de caso de prueba: 1	Nombre de Historia de Usuario: Pre-procesar consulta.
Nombre de la persona que realiza la prueba: Carel Rodríguez Mendoza.	
Descripción de la prueba: Prueba a la funcionalidad pre-procesar consulta.	
Entrada / Pasos de la ejecución: El cliente realiza la petición mediante el protocolo HTTP: http://localhost:8080/buscar/consulta/parametro?= las coordenadas de varadero	
Resultado esperado: [coordenada, varadero]	
Evaluación de la prueba: Satisfactoria	

Tabla 13. Caso de prueba Obtener coordenadas. Fuente: Elaboración propia.

Caso de prueba Obtener coordenadas.	
Código de caso de prueba: 2	Nombre de Historia de Usuario: Obtener coordenadas.
Nombre de la persona que realiza la prueba: Carel Rodríguez Mendoza.	
Descripción de la prueba: Prueba a la funcionalidad obtener coordenadas.	

Entrada / Pasos de la ejecución: El cliente realiza la petición mediante el protocolo HTTP: http://localhost:8080/buscar/consulta/parametro?= coordenadas de varadero
Resultado esperado: Valor de las coordenadas.
Evaluación de la prueba: Satisfactoria

Tabla 14. Caso de prueba Obtener elevación. Fuente: Elaboración propia.

Caso de prueba Obtener elevación.	
Código de caso de prueba: 3	Nombre de Historia de Usuario: Obtener elevación.
Nombre de la persona que realiza la prueba: Carel Rodríguez Mendoza.	
Descripción de la prueba: Prueba a la funcionalidad obtener elevación.	
Entrada / Pasos de la ejecución: El cliente realiza la petición mediante el protocolo HTTP: http://localhost:8080/buscar/consulta/parametro?= elevación de varadero	
Resultado esperado: Valor de la elevación.	
Evaluación de la prueba: Satisfactoria	

Tabla 15. Caso de prueba Obtener código postal.

Caso de prueba Obtener código postal.	
Código de caso de prueba: 4	Nombre de Historia de Usuario: Obtener código postal.
Nombre de la persona que realiza la prueba: Carel Rodríguez Mendoza.	
Descripción de la prueba: Prueba a la funcionalidad obtener código postal.	
Entrada / Pasos de la ejecución: El cliente realiza la petición mediante el protocolo HTTP: http://localhost:8080/buscar/consulta/parametro?= código postal de varadero	
Resultado esperado: Valor del código postal.	
Evaluación de la prueba: Satisfactoria	

Tabla 16. Caso de prueba Obtener código de área.

Caso de prueba Obtener código de área.	
Código de caso de prueba: 5	Nombre de Historia de Usuario: Obtener código de área.
Nombre de la persona que realiza la prueba: Carel Rodríguez Mendoza.	
Descripción de la prueba: Prueba a la funcionalidad obtener código de área.	
Entrada / Pasos de la ejecución: El cliente realiza la petición mediante el protocolo HTTP: http://localhost:8080/buscar/consulta/parametro?= código telefónico del área de varadero	
Resultado esperado: Valor de código del área.	
Evaluación de la prueba: Satisfactoria	

Tabla 17. Caso de prueba Obtener extensión de área.

Caso de prueba Obtener extensión de área.	
Código de caso de prueba: 6	Nombre de Historia de Usuario: Obtener extensión de área.
Nombre de la persona que realiza la prueba: Carel Rodríguez Mendoza.	
Descripción de la prueba: Prueba a la funcionalidad obtener extensión de área.	
Entrada / Pasos de la ejecución: El cliente realiza la petición mediante el protocolo HTTP: http://localhost:8080/buscar/consulta/parametro?= extensión del área correspondiente a varadero	
Resultado esperado: Valor de la extensión del área.	
Evaluación de la prueba: Satisfactoria	

Tabla 18. Caso de prueba Obtener cantidad de habitantes.

Caso de prueba Obtener cantidad de habitantes.	
Código de caso de prueba: 7	Nombre de Historia de Usuario: Obtener cantidad de habitantes.
Nombre de la persona que realiza la prueba: Carel Rodríguez Mendoza.	

Descripción de la prueba: Prueba a la funcionalidad obtener cantidad de habitantes.
Entrada / Pasos de la ejecución: El cliente realiza la petición mediante el protocolo HTTP: http://localhost:8080/buscar/consulta/parametro?= cantidad de personas que viven en varadero
Resultado esperado: Valor de la cantidad de habitantes.
Evaluación de la prueba: Satisfactoria

Tabla 19. Caso de prueba Obtener información general.

Caso de prueba Obtener información general.	
Código de caso de prueba: 8	Nombre de Historia de Usuario: Obtener información general.
Nombre de la persona que realiza la prueba: Carel Rodríguez Mendoza.	
Descripción de la prueba: Prueba a la funcionalidad obtener información general.	
Entrada / Pasos de la ejecución: El cliente realiza la petición mediante el protocolo HTTP: http://localhost:8080/buscar/consulta/parametro?= información de varadero	
Resultado esperado: Información general.	
Evaluación de la prueba: Satisfactoria	

Resultados de las pruebas funcionales

Las pruebas se realizaron en un total de 4 iteraciones. En la figura se muestra el total de las no conformidades detectadas y resueltas en cada iteración. Para un total de 8 requisitos funcionales se detectaron 21 no conformidades en la primera iteración de las que se resolvieron 18. En la segunda se detectaron 13 no conformidades a las que se les sumaron las 3 de la iteración pasada, conformando un total de 16 no conformidades, de las que se resolvieron 15. En la iteración 3 se detectaron 3 no conformidades incluyendo la que restó de la iteración 2, de las cuales se resolvieron todas satisfactoriamente. Finalmente en una última iteración no se detectó ninguna no conformidad, validando el correcto funcionamiento del sistema.

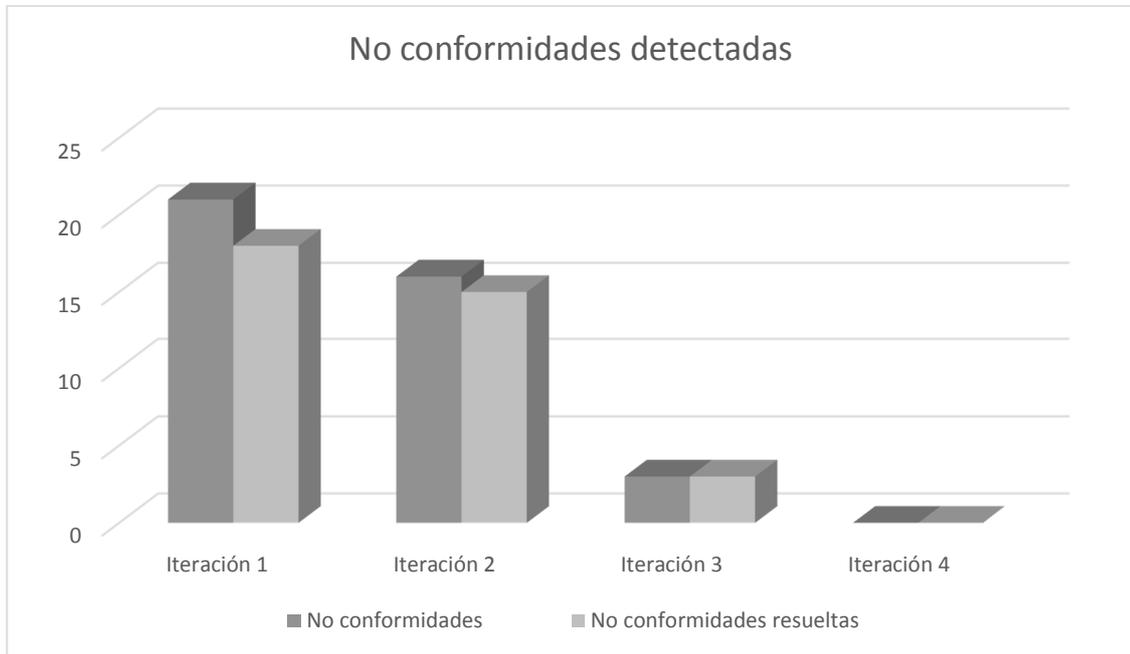


Figura 11. No conformidades.

Ejemplos de no conformidades detectadas en la realización de las pruebas funcionales:

- Pre-procesamiento incorrecto de consultas.
- No devolución de la información requerida.
- Información mal estructurada.
- Incorrecto lanzamiento de excepciones.

3.3.3. Prueba de aceptación

La Junta Internacional de Cualificaciones de Pruebas de Software (*ISTQB* por sus siglas en inglés) define la “Aceptación” como pruebas formales con respecto a las necesidades del usuario, requerimientos y procesos de negocio, las que se realizan para determinar si un sistema satisface los criterios de aceptación de un usuario, cliente u otra entidad autorizada pueda determinar si acepta o no el sistema (Báez & Fernanda, 2015).

Al finalizar el período de desarrollo se entregó la aplicación al cliente para realizar la aceptación del producto, el que expresó su criterio mediante una carta de aceptación, a partir de sus consideraciones respecto a las ventajas ofrecidas por el componente en función de la recuperación de información georreferenciada y las

ventajas que ofrece. Se realizó la solicitud de colaboración de un grupo de expertos en el trabajo con servicios web y recuperación de información para respaldar dicho criterio.

Se realizó una encuesta (ver Anexo 1) para validar el nivel de aceptación del componente en los expertos. La encuesta fue realizada a 5 expertos con experiencia profesional en el desarrollo de servicios web y sistemas de recuperación de información. A continuación se detalla la técnica del criterio de expertos aplicada en la investigación.

3.3.4. Criterio de expertos

Para la validación del portal web se utiliza el método criterio de expertos empleando los siguientes pasos:

- Identificación de los posibles expertos.
- Selección de los expertos.
- Realización de la consulta a los expertos, procesamiento y valoración de la información obtenida.

En la siguiente tabla se muestran los expertos seleccionados.

Tabla 20. Expertos seleccionados.

No.	Experto	Entidad	Años de experiencia
1	Leiny Amel Pons Flores	CIDI	5
2	Alexander García O'Relly	CIDI	2
3	Miguel Ángel Chávez Alfonso	CIDI	8
4	Yadiel Pérez Villazón	CIDI	5
5	José Gabriel Espinosa Ramírez	CIDI	8

La tabla que a continuación se presenta resume el resultado de los juicios emitidos por los expertos, de acuerdo a las siguientes interrogantes:

1. ¿Cómo valora los métodos de procesamiento de consultas utilizados en el desarrollo de la solución implementada?
2. ¿Cómo considera el proceso de recuperación de información georreferenciada con la utilización del componente desarrollado?
3. ¿Cómo valora los componentes utilizados en el desarrollo de la solución?

Para el procesamiento y análisis de la información obtenida se analizaron las respuestas de cada uno de los parámetros que aparecen en la encuesta. Los niveles empleados para la valoración fueron: **MA**: Muy adecuado, **A**: Adecuado, **PA**: Poco adecuado y **NA**: No adecuado. En la tabla 21, **N** se refiere a la cantidad de encuestados que emitieron su valoración y **%** al porcentaje que representa con respecto al total de encuestados.

Tabla 21. Valoración de los expertos. Fuente: Elaboración propia.

Parámetros evaluados	Niveles de valoración									
	MA		A		PA		NA		Total	
	N	%	N	%	N	%	N	%	N	%
1	5	100	0	0	0	0	0	0	5	100
2	5	100	0	0	0	0	0	0	5	100
3	4	80	1	20	0	0	0	0	5	100

De acuerdo con los criterios establecidos, así como otras consideraciones expresadas por los expertos, se puede afirmar que la solución implementada tiene un nivel satisfactorio de aceptación. Los aspectos evaluados, determinados sobre la base del objetivo general de la investigación, fueron valorados todos entre los niveles de adecuado y muy adecuado, lo que demuestra el cumplimiento de este objetivo desde el punto de vista de los expertos. Adicionalmente, se obtuvo un conjunto de recomendaciones y valoraciones que perfeccionan el resultado final de la investigación, así como futuras profundizaciones sobre el mismo.

Consideraciones finales del capítulo 3

- La construcción de la capa de servicios web por iteraciones, permitió obtener versiones del producto al culminar cada iteración.
- Mediante las pruebas de aceptación aplicadas se pudo demostrar que el sistema se encuentra listo y cumple con los requisitos propuestos en las Historias de Usuarios.

-
- Al finalizar la última iteración se obtuvo un componente completamente funcional que cumple con las especificaciones establecidas por el cliente.
 - La aplicación de los estándares definidos permitió la implementación de un código bien estructurado, facilitando los futuros procesos de actualización del sistema.

Conclusiones

Luego de realizar el análisis, el diseño, la implementación y las pruebas al Componente para generar un servicio web de información georreferenciada, se arribaron a las siguientes conclusiones:

1. El uso de los métodos teóricos seleccionados contribuyó a una mejor comprensión del objeto de estudio y el contexto del problema.
2. Los referentes teóricos y metodológicos establecidos demostraron la importancia de contar con mecanismos para la recuperación de información georreferenciada.
3. Se establecieron todos los artefactos que justifican el proceso desarrollo y la documentación técnica asociada a la propuesta de solución.
4. Las técnicas de validación aplicadas a la propuesta de solución permitieron la detección y corrección de las no conformidades reveladas, evidenciaron que el componente establece una solución funcional.
5. La validación de la hipótesis mediante el criterio de expertos demostró que el componente constituye una mejora para el proceso de recuperación de información georreferenciada.

Recomendaciones

- Realizar actualizaciones sistemáticas a la base de datos con el objetivo de brindar información un servicio que no contenga información obsoleta.

Referencias bibliográficas

1. Aguilar Cuesta, A. I., & Garcia Juan, L. (2017). Big Data, una tecnología de hoy para el análisis del pasado. Recuperado 30 octubre, 2018, de <https://www.researchgate.net/publication/320685993+>
2. Amaya Ramírez, M. A. (2017). Evaluación de herramientas de recuperación de información electrónica. Recuperado 6 noviembre, 2018, de <https://dialnet.unirioja.es/servlet/articulo?codigo=6068216>
3. API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos. (2018). Recuperado 12 noviembre, 2018, de <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>
4. Agustí, T. E. (2017). La georeferenciación. Diario La Ley, (8943), 3.
5. Alarcón Altamirano, K. A., Pérez Rodríguez, G. P., & Moya Gómez, B. (2017). Análisis Geográfico de la Red Social Twitter en Ecuador. Recuperado 26 octubre, 2018, de https://www.rmlconsultores.com/revista/index.php/crv/article/download/580/pdf_400
6. Córdova, B; Fernanda, D. (2015). Análisis comparativo de los modelos y estándares de calidad de software y aplicación de las mejores prácticas para el levantamiento del proceso de gestión de calidad de productos de software. <http://www.dspace.uce.edu.ec/handle/25000/4184>
7. Beltrán López, G. (2015, 26 diciembre). La geolocalización social |Polígonos. Revista de Geografía. Recuperado 30 octubre, 2018, de <http://revpubli.unileon.es/ojs/index.php/poligonos/article/view/3290>
8. Cossio Taboada, Miguel Humberto y Floren, Huanca A. Diagrama de Despliegue. Universidad Salesiana de Bolivia. Bolivia. 2009.
9. Caribbean News Digital. (2018). Cuba expone avances tecnológicos en Informática 2018. Recuperado 1 noviembre, 2018, de <https://www.excelenciascuba.com/noticia/cuba-expone-avances-tecnologicos-en-informatica-2018>

-
10. Casanova, J.L.; Sanz, J.; Salvador, P. (2014). Curso teórico experimental de teledetección espacial. Laboratorio de Teledetección. Departamento de Física Aplicada: Universidad de Valladolid.
 11. Cruz, J. S. (2017). CONSUMO DE SERVICIOS WEB POR MEDIO DE CLIENTES JAVA AUTOGENERADOS PARA LA EMPRESA ST&T L.T.D.A. Recuperado 7 noviembre, 2018, de <http://repository.udistrital.edu.co/bitstream/11349/6781/1/Sebasti%C3%A1n%20Cruz%20Mora%202017.pdf>
 12. Cobo, Á. Diseño y programación de bases de datos. Disponible en: <http://books.google.es/books?hl=es&lr=&id=anCDr9NkGsC&oi=fnd&pg=PA7&dq=Dise%C3%B1o+y+programaci%C3%B3n+de+bases+de+datos&ots=UXEBp8mpzV&sig=jPWxCyBUit3XHIQIr4NpzhIbUwQ>
 13. Domínguez López, A. (2013). Geolocalización de obras de arte en la ciudad de Valencia Desarrollo de un portal web y aplicación Android. Recuperado 14 noviembre, 2018, de <https://riunet.upv.es/bitstream/handle/10251/44465/Memoria.pdf?sequence=1>
 14. Egenhofer, M.; Herring, J. 1991. High-Level Spatial Data Structures. Geographical Information Systems, Vol. 1: Principles. s.l.: Longman, 1991.
 15. Espiritusanto Nicolás, O. (2017). Generación Z: Móviles, redes y contenido generado por el usuario. Recuperado 29 octubre, 2018, de http://www.injuve.es/sites/default/files/2017/28/publicaciones/documentos_8._generacion._.moviles_redes_y_contenido_generado_por_el_usuario.pdf
 16. Fries, A., Pucha Cofrep, F., & Oñate Valdivies, F. (2017). Fundamentos de SIG. Recuperado 10 noviembre, 2018, de https://www.researchgate.net/publication/318447525_Fundamentos_de_SIG
 17. García Fernández, J. A. (2017). Técnicas Avanzadas de Geolocalización en Redes UMTS. Recuperado 8 noviembre, 2018, de https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=8&ved=2ahUKEwjg5bLdrsTeAhWq7oMKHXQ3B3wQFjAHegQIAxAC&url=https%3A%2F%2Friuma.uma.es%2Fxmlui%2Fbitstream%2Fhandle%2F10630%2F14133%2FTD_GARCIA_FERNANDEZ_Juan_Antonio.pdf%3Fsequence

nce%3D1&usg=AOvVaw0RJscXb8u_Sa_1uYX07Inj

18. García Palomares, J. C., Mínguez García, M. A., & Gutiérrez Puebla, J. (2014). Nuevas fuentes de información geográfica en turismo: las oportunidades de sightsmat.com. Recuperado 26 octubre, 2018, de <http://hdl.handle.net/10045/46830>
19. García-Cuesta, J.L.; Molina, I.; García-Gómez, F.M.; Arroyo, P. (2013). Curso de introducción a los sistemas de información geográfica. Departamento de Geografía: Universidad de Valladolid.
20. Gaspar Muñoz, G. G., & Abanto Flores, H. E. (2018, 20 junio). Repositorio UTP: Desarrollo e implementación de un aplicativo móvil para la geolocalización de unidades terrestres en una empresa de seguimiento satelital. Recuperado 18 octubre, 2018, de <http://repositorio.utp.edu.pe/handle/UTP/1147>
21. Gómez-Lahoz, J. (2014). Georreferenciación de imágenes de satélite. Departamento de Ingeniería Cartográfica y del Terreno: Universidad de Salamanca.
22. Google. 2012. Acerca de Map Maker. [En línea] Google, 14 de noviembre de 2012. [Citado el: 14 de enero de 2019.] <http://support.google.com/mapmaker/answer/157176?hl=es>
23. Google Inc. 2013. Google Map - Google Play. Google Map - Google Play. [En línea] Google, 17 de diciembre de 2013. [Citado el: 14 de enero de 2019.]
24. Google Maps. 2014. Add Google Maps to your website - Advanced. [En línea] 2014. [Citado el: 14 de enero de 2019.] maps.google.es/help/maps/getmaps/advanced.html.
25. Hernández Carvajal, Juan José. 2012. Desarrollo del módulo de Administrador de Reportes del Generador Dinámico de Reportes. La Habana: s.n., 2012
26. IBM Knowledge Center. (2018). Recuperado 11 noviembre, 2018, de https://www.ibm.com/support/knowledgecenter/es/SS4JE2_7.5.5/org.eclipse.jst.ws.doc.user/concepts/cwsstandards.html
27. Internet Live Stats - Internet Usage & Social Media Statistics. (2018). Recuperado 15 octubre, 2018,

de <http://www.internetlvestats.com/>

28. J.J. Gutiérrez, M.J. Escalona, Mejías, Torres. 2012. Sistema de Programacion Extrema. [En línea] 2012. [Citado el: 4 de mayo de 2018.] http://www.lsi.us.es/~javierj/investigacion_ficheros/PSISEXTREMA.pdf.
29. Larman, Craig; Hall, Prentice. (2003). UML y Patrones. 2da Edición. 2003.
30. Llorente, C.; Zorrilla, U.; Barroso, M. (2010). Guía de Arquitectura N-Capas orientada al Dominio.
31. Luna, P. (2010). Sistema de Información Geográfica para la ayuda de toma de decisiones en políticas sociales. Departamento de Computación, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional. Mexico DF : s.n., 2010. Tesis de Maestría en Ciencias de la Computación.
32. Martínez Méndez, F. J. (2015). Recuperación de Información: Modelos, Sistemas y Evaluación. Recuperado 2 noviembre, 2018, de <https://www.researchgate.net/publication/27339898>
33. MapDroyd. 2010. MapDroyd Maps on Android - Unplugged! Sitio Oficial de MapDroyd. [En línea] 24 de julio de 2010. [Citado el: 14 de enero de 2019.] <http://www.mapdroyd.com/mapandroid-home>
34. Montejo Sánchez, S., Sierra Dávila, R., & Gorrita Michel, E. (2012). Aplicaciones y servicios basados en localización. REVISTA CUBANA DE INGENIERÍA, III (1). Recuperado de <https://docplayer.es/1049455-Revista-cubana-de-ingenieria.html>
35. Olaya, V. (2014). Sistemas de Información Geográfica. Recuperado 11 noviembre, 2018, de https://www.icog.es/TyT/files/Libro_SIG.pdf
36. Ornay, E. (2013). Infovoto. Aplicación SIG en diferentes campañas electorales. [En línea] 22 de octubre de 2013. [Citado el: 5 de noviembre de 2018.] http://infovoto.blogspot.com/2010_07_01_archive.html
37. OsmAnd Team. 2012. OsmAnd Automated Navigation Directions. [En línea] OsmAnd Team, 21 de noviembre de 2012. [Citado el: 14 de enero de 2019.] <http://osmand.net/>.

-
38. Pérez Martínez, Y. (s.f.). Serie Científica de la Universidad de las Ciencias Informáticas. Recuperado 14 noviembre, 2018, de <https://publicaciones.uci.cu/?journal=SC>
39. Penadés, Patricio Letelier y M^a Carmen. 2013. Metodologías ágiles para el desarrollo de software: Extreme Programming (XP). Valencia: Universidad Politécnica de Valencia, 2013.
40. PHP. PHP: Hypertext Processor. [En línea] [Citado el: 14 de Enero de 2019] www.php.net.
41. PostgreSQL: Documentation: 9.0: Release 9.0.1. Disponible en: <https://www.postgresql.org/docs/9.0/static/release-9-0-1.html>.
42. Ramírez, K. (2012a). Análisis Léxico. Recuperado 28 mayo, 2019, de <http://www.kramirez.net/RI/Exposiciones/AnalisisLexico/>
43. Ramírez, K. (2012b). Steamming. Recuperado 28 mayo, 2019, de <http://www.kramirez.net/RI/Exposiciones/AnalisisLexico/>
44. Reguant, M; Torrado, M. (2016). El método Delphi. Revista d'Innovació i Recerca en Educació. 2016. vol. 9, no. 1. pp. 87-102. ISSN:2013-2255
45. Reyes, N., Deco, C., & Bender, C. (2012, 23 julio). Recuperación de información en bases de datos no estructuradas. Recuperado 6 noviembre, 2018, de <http://sedici.unlp.edu.ar/handle/10915/18824>
46. Sanet, Hanan. 1994. The Design and Analysis of Spatial Data Structures. University of Maryland: Addison - Wesley Publications Company Inc., 1994.
47. Robinson, C. Basic introduction into pgAdmin III and SQL queries. Disponible en: <https://library.thehumanjourney.net/658/>.
48. Timbiano, R. (2010). Servicio de localización a través de una red umts [Conjunto de datos]. Recuperado 1 noviembre, 2018, de https://www.researchgate.net/publication/44788628_Servicio_de_localizacion_a_traves_de_una_red_umts

-
49. UNAD. Lenguaje Unificado de Modelado UML. Diagrama de Clases de Diseño [En línea]. Universidad Nacional Abierta y a Distancia, 2016 [Fecha de consulta: 1 de marzo de 2018]. Disponible en: http://stadium.unad.edu.co/ovas/10596_9836/diagrama_de_clases_de_diseo.html.
50. Vialart Vidal, M. N. (2016). Las Tecnologías de la Información y las Comunicaciones un desafío para la Gestión del Cuidado. Recuperado 27 octubre, 2018, de http://scielo.sld.cu/scielo.php?pid=S086403192016000100014&script=sci_arttext&lng=en
51. Vilca Espinoza, R. A. (2018, 17 julio). Influencia de un sistema de geolocalización en el control y monitoreo de vehículos con dispositivos GPS en una empresa logística, 2015. Recuperado 9 noviembre, 2018, de <http://repositorio.ucv.edu.pe/handle/UCV/17105>
52. ZORRILLA, Marta. Modelo de datos [En línea]. Cantabria: Universidad de Cantabria, 2011 [Fecha de consulta: 24 marzo 2018]. Disponible en: <https://es.scribd.com/document/73596837/02-Modelos-de-DatosER-UML-Relacional>.

Anexos

Anexo 1. Entrevista a los expertos para la validación de la propuesta de solución

Estimados expertos atendiendo a su experiencia profesional en el desarrollo de servicios web y sistemas de recuperación de información, queremos conocer cuál es su grado de satisfacción con respecto al componente para generar un servicio web de información georreferenciada. La información que nos facilite, ayudará para la validación del componente. Le pedimos que conteste lo más sinceramente posible las siguientes preguntas. Gracias por su colaboración.

1. ¿Cómo valora los métodos de procesamiento de consultas utilizados en el desarrollo de la solución implementada?
 - a) ___Muy adecuado
 - b) ___Adecuado
 - c) ___Poco adecuado
 - d) ___No adecuado

2. ¿Cómo considera el proceso de recuperación de información georreferenciada con la utilización del componente desarrollado?
 - a) ___Muy adecuado
 - b) ___Adecuado
 - c) ___Poco adecuado
 - d) ___No adecuado

3. ¿Cómo valora los componentes utilizados en el desarrollo de la solución?
 - a) ___Muy adecuado
 - b) ___Adecuado
 - c) ___Poco adecuado
 - d) ___No adecuado

4. ¿Qué elementos se pudieran incorporar para mejorar el proceso de recuperación de información georreferenciada?

5. ¿Cómo evalúa la aplicabilidad del componente desarrollado?