



Universidad de las Ciencias Informáticas

Facultad 1

**Subsistema de búsqueda por localización geográfica
para la plataforma de servicios c.u.b.a.**

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

Autora:

Beatriz Morales Bermúdez

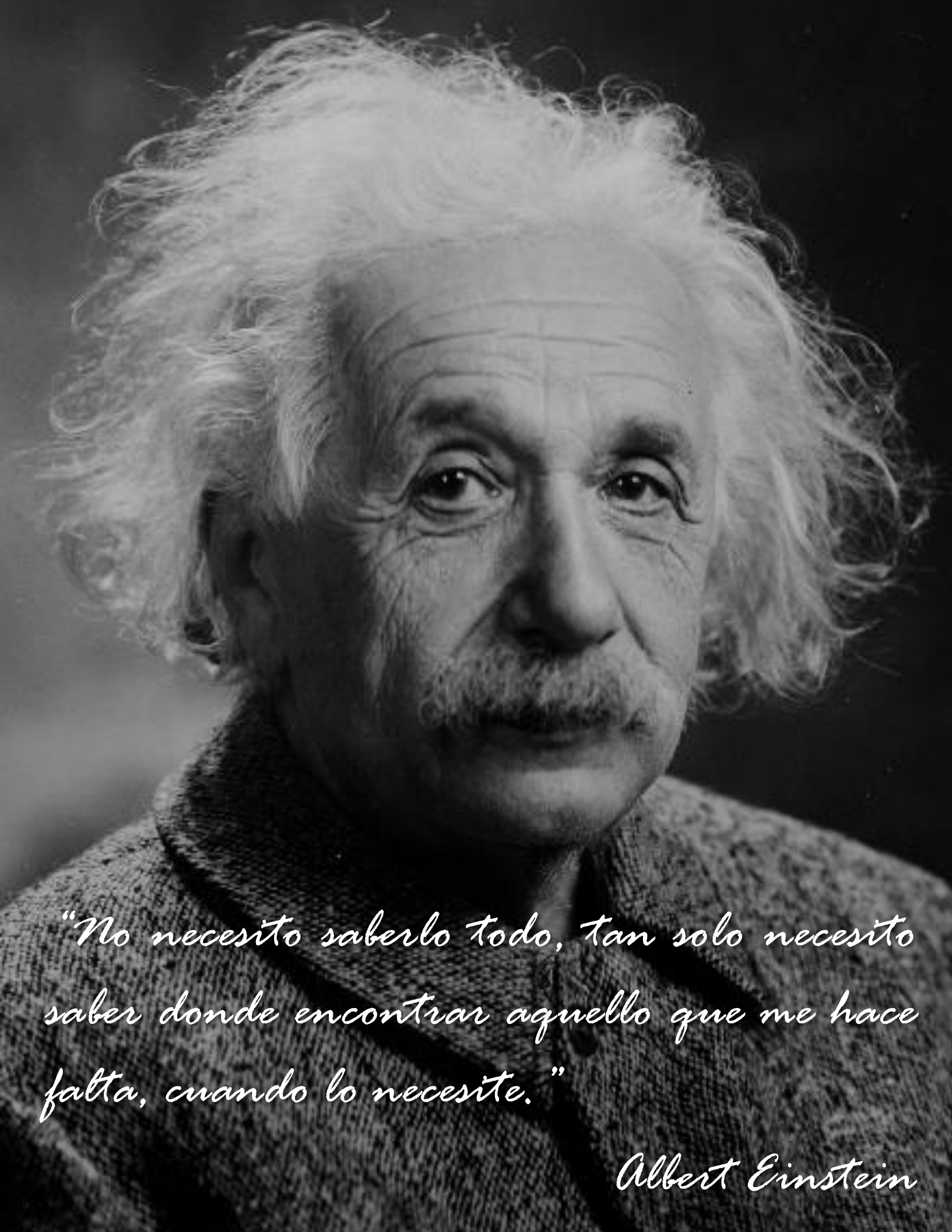
Tutores:

Ing. Martha Luisa Gala Rodríguez

Ing. Odisleysi Martínez Furones

Ing. Roannel Fernández Hernández

La Habana, 2018



"No necesito saberlo todo, tan solo necesito saber donde encontrar aquello que me hace falta, cuando lo necesite."

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo Beatriz Morales Bermúdez, con carnet de identidad 94091441831, soy el autor principal del trabajo titulado “**Subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a.**” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales de la misma con carácter exclusivo.

Y para que así conste, firmo la presente declaración de autoría en La Habana a los ____ días del mes de _____ del año 2018.

Autora:

Beatriz Morales Bermúdez

Tutores:

Ing. Marta Luisa Gala Rodríguez

Ing. Odisleysi Martínez Furones

Ing. Roannel Fernández Hernández

RESUMEN

El presente trabajo consiste en desarrollar un subsistema de búsquedas por localización geográfica para mejorar el proceso de recuperación de información geográfica de la plataforma de servicios c.u.b.a. Se estudiaron los sistemas de información geográfica (SIG) más utilizados en el ámbito nacional e internacional, de ellos se identificaron las principales funcionalidades y tecnologías que contribuyeron a la implementación del subsistema de búsqueda por localización geográfica. Las principales tecnologías seleccionadas para la fase de desarrollo fueron: el marco de trabajo Symfony 2, como bibliotecas fundamentales para el desarrollo del SIG OpenLayers y PostGIS y como servidor de mapa se utilizó el MapServer. Todo el proceso de desarrollo fue guiado bajo la metodología AUP en su versión para la Universidad de las Ciencias Informáticas (UCI). Se aplicaron diferentes pruebas de software de tipo funcional, de seguridad, integración, carga y estrés, donde se comprobó el correcto funcionamiento del subsistema de búsqueda por localización geográfica, así como su seguridad y capacidad de responder a una alta concurrencia de usuarios. Para la validación de la hipótesis se realizó una consulta a expertos mediante el método Delphi donde se validó que el subsistema de búsqueda por localización geográfica tiene un alto valor al mejorar la recuperación geográfica en la plataforma de servicios c.u.b.a.

Palabras claves: buscador web, localización geográfica, sistema de información geográfica.

TABLA DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO 1: “Fundamentos teóricos del subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a.”	5
1.1. Conceptos asociados al dominio de la investigación	5
1.2. Principales sistemas homólogos.....	6
1.3. Metodología de desarrollo de software	9
1.4. Herramientas y tecnologías	10
1.5. Resumen del capítulo	17
CAPÍTULO 2: “Análisis y diseño del subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a.”	19
2.1. Modelo conceptual	19
2.2. Descripción de la propuesta de solución	20
2.3. Arquitectura del sistema	20
2.4. Requisitos de la propuesta de solución	21
2.5. Casos de uso	23
2.6. Patrones utilizados	26
2.7. Diagrama de clases de diseño.....	29
2.8. Diagrama de secuencia	31
2.9. Modelo de datos.....	32
2.10. Diagrama de despliegue.....	33
2.11. Resumen del capítulo	34
CAPÍTULO 3: “Implementación y prueba del subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a.”	35

3.1: Diagrama de componentes.....	35
3.2: Estándares de codificación.....	36
3.3: Validación de la propuesta de solución.....	38
3.4: Resumen del capítulo.....	48
CONCLUSIONES	49
RECOMENDACIONES	50
BIBLIOGRAFÍA.....	51
ANEXOS.....	56
Anexo 1: Encuesta a expertos.....	56
Anexo 2: Casos de prueba	57

ÍNDICE DE TABLAS

Tabla 1. Requisitos funcionales.	22
Tabla 2: Tabla de casos de uso	23
Tabla 3: CU1: Manipular mapa	25
Tabla 4: CU2 Realizar búsqueda	25
Tabla 5: Estándares de codificación.....	36
Tabla 6: Descripción de las variables para el Caso de prueba 1	38
Tabla 7: Caso de prueba 1: SC RF3_Búsqueda de lugares por nombre de los lugares.....	38
Tabla 8: Descripción de las variables para el Caso de prueba 2	39
Tabla 9: Caso de prueba 2: SC RF4_Búsqueda de lugares por coordenadas de los lugares	39
Tabla 10: Pruebas de integración	41
Tabla 11: Pruebas de carga y estrés.....	42
Tabla 12: Resultado de niveles de conocimiento de posibles expertos	44
Tabla 13: Patrón para el cálculo de Coeficiente de argumentación o fundamentación.....	44
Tabla 14: Resultado de niveles de competencia de posibles expertos.....	45
Tabla 15: Expertos seleccionados para la validación de la hipótesis científica.....	46
Tabla 16: Resultado de las encuestas realizadas a los expertos	46
Tabla 17: Frecuencia acumulada de los datos primarios obtenidos	47
Tabla 18: Frecuencia relativa acumulativa de los datos primarios obtenidos	47
Tabla 19: Frecuencia relativa acumulativa de los datos primarios obtenidos	47
Tabla 20: Encuesta a expertos.....	56
Tabla 21: Descripción de las variables para el Caso de prueba 3	57
Tabla 22: Caso de prueba 3: SC RF6_Insertar Lugar	57

ÍNDICE DE FIGURAS

Figura 1: Modelo conceptual.....	19
Figura 2: Arquitectura del subsistema propuesto	21
Figura 3: Diagrama de casos de uso.....	24
Figura 4: DCD del CU Manipular mapa.....	29
Figura 5: DCD del CU Gestionar lugar	30
Figura 6 DCD del CU Realizar búsqueda.....	30
Figura 7: DS del CU Manipular mapa.....	31
Figura 8: DS del CU Realizar búsqueda	31
Figura 9: DS Gestionar lugar.....	32
Figura 10: Modelo de datos	32
Figura 11: Diagrama de componentes	35
Figura 12: Resultados de las pruebas funcionales	40

INTRODUCCIÓN

La revolución de las tecnologías aplicadas a la cartografía y el análisis espacial están relacionados con los significativos avances de las Tecnologías de la Información Geográfica (TIG). Las TIG son un conjunto de herramientas (radares, GPS¹, satélites, dispositivos móviles y ordenadores) que contribuyen a técnicas de ubicación o localización geográfica (Alcántara, et al., 2014).

La ubicación geográfica proporciona al usuario información sobre el lugar exacto en que se encuentra un punto o territorio en la superficie terrestre y se realiza mediante un sistema de coordenadas compuestas por latitud y longitud. Además, las búsquedas por localización geográfica facilitan una breve información sobre el clima, la economía, aspectos sociales e imágenes de los puntos localizados (Orellana, 2015).

Para lograr el manejo de la información obtenida mediante las búsquedas por localización geográfica existen los Sistemas de Información Geográfica (SIG). Los SIG permiten diseñar, almacenar, recuperar y desplegar grandes volúmenes de datos espaciales. Proporcionan los medios para realizar el análisis de la posición geográfica con exactitud, la organización, administración de los datos y facilita a los usuarios la recuperación de información geográfica almacenada en las diversas bases de datos (INEGI, 2017).

Otras herramientas que permiten la recuperación de información son los motores de búsqueda, sistemas recolectores de información, capaces de localizar cualquier contenido existente en la web, su principal objetivo es localizar y procesar la información de forma rápida y automática en Internet. La información se puede encontrar en diversos formatos como textos, imágenes, archivos de sonido, videos y mapas (Fernández et al., 2015). Google, Yahoo! y Bing, son tres de los buscadores más utilizados a nivel mundial. Utilizan el posicionamiento como una estrategia o conjunto de tácticas para mejorar la visibilidad de un sitio web (Berreiros, 2015).

El posicionamiento de la información en los buscadores es un inconveniente en países como Cuba por la denegación de servicios y el limitado acceso a Internet. Como solución a este inconveniente y en aras de asegurar la sostenibilidad y soberanía tecnológica, en nuestro país se realiza un proceso de informatización de la sociedad. El proceso pretende modernizar y potenciar el acceso de los ciudadanos a las nuevas

¹ Sistema de Posicionamiento Global (por sus siglas en inglés GPS).

tecnologías con énfasis en las prioridades del país e integrar la investigación, el desarrollo y la innovación, con la elaboración y comercialización de productos y servicios (Guevara, 2017).

Desde la UCI se apoya el proceso de informatización que realiza el país y entre sus productos se desarrolla la plataforma de servicios Contenido Unificado para la Búsqueda Avanzada (c.u.b.a.). La plataforma de servicios c.u.b.a. facilita el acceso a los contenidos publicados en la red cubana e indexa la gran mayoría de los sitios web nacionales. Representa un paso más hacia la soberanía tecnológica e implica una reducción de costos para el país, al no tener que pagar necesariamente el acceso a Internet para realizar búsquedas de contenidos en la red cubana. Es diversa la información a la que acceden los usuarios, además de enriquecer su documentación en temas de deportes y recreación, entretenimiento, noticias, salud, arte, humanidades y política.

La plataforma de servicios c.u.b.a. permite realizar búsquedas web, de documentos, imágenes, noticias y académica, sin embargo, carece de una herramienta que permita la recuperación de información geográfica. Los usuarios para encontrar direcciones o información sobre temas geográficos dentro de la misma recurren a la búsqueda web. Esta búsqueda en pocas ocasiones contiene o hace referencia a mapas o ubicaciones geográficas por lo que resulta complicado, lento, con baja precisión y exhaustividad en la información que se recupera en la web. Provoca, además, falta de interés del usuario y hace que recurra a otro tipo de herramientas como aplicaciones web de mapas, guías turísticas o directorios.

Luego de analizar lo antes planteado surge el siguiente **problema científico**: ¿Cómo facilitar la recuperación de información geográfica en la plataforma de servicios c.u.b.a.?

Se identifica como **objeto de estudio** el proceso de recuperación de información geográfica y el **campo de acción** está enmarcado en el proceso de recuperación de información geográfica en la plataforma de servicios c.u.b.a.

Para darle solución al problema antes planteado se establece el siguiente **objetivo general**: Desarrollar un subsistema de búsquedas por localización geográfica que facilite el proceso de recuperación de información geográfica en la plataforma de servicios c.u.b.a.

Con el fin de darle cumplimiento al objetivo general se trazan los siguientes **objetivos específicos**:

1. Describir los fundamentos teóricos relacionados el subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a

2. Identificar las tecnologías, herramientas, lenguajes y metodología del subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a.
3. Diseñar el subsistema de búsqueda por localización geográfica para la plataforma de servicio c.u.b.a.
4. Implementar el subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a.
5. Validar la solución propuesta a partir del entorno de prueba.

La **hipótesis** para la presente investigación científica es: El desarrollo de un subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a. mejora el proceso de recuperación de información geográfica.

Para lograr el cumplimiento de los objetivos específicos se definen las siguientes **tareas de investigación**:

1. Estudio de los conceptos asociados al marco teórico de la investigación.
2. Caracterización de los sistemas que permitan la recuperación de información geográfica.
3. Selección de las herramientas, lenguajes y tecnologías que permitan desarrollar la propuesta de solución.
4. Análisis y diseño de la propuesta de solución en correspondencia con la metodología seleccionada.
5. Definición de la arquitectura y patrones a utilizar en el subsistema de búsqueda por localización geográfica.
6. Desarrollo de las funcionalidades que componen el subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a.
7. Validación del subsistema de búsqueda por localización geográfica teniendo en cuenta su funcionamiento, seguridad, rendimiento e integración con la plataforma de servicios c.u.b.a.

Los **métodos de investigación** que se utilizaron para dar solución a la presente investigación son:

Métodos Teóricos:

Histórico-Lógico: Estudia y determina la evolución, comportamiento y tendencias actuales de las tecnologías y herramientas a utilizar en el diseño y desarrollo de la propuesta de solución.

Analítico-Sintético: Es empleado para el análisis de las herramientas, tecnologías y metodologías, para identificar y determinar cuales se utilizarán para el desarrollo de la investigación. Permite llegar a un mayor entendimiento del campo de acción para aplicarlo en el desarrollo de la solución propuesta.

Métodos empíricos:

Modelación: Representación de las características del sistema, así como las relaciones entre los objetos y funcionalidades del mismo.

Observación: Posibilita obtener conocimiento acerca del funcionamiento de los sistemas existentes en la actualidad relacionados con los sistemas de información geográfica.

Encuesta: Se utilizó para la recopilación de información en la validación de la hipótesis, al aplicar el Criterio de expertos mediante el método Delphi.

Entrevista: Se emplea en encuentros con el cliente para definir las funcionalidades de la aplicación web, identificando a la vez particularidades necesarias para su desarrollo.

La presente investigación se desglosa en los siguientes capítulos:

Capítulo 1: “Fundamentos teóricos del subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a.”: En el capítulo 1 se describen los conceptos por los cuales se rige el dominio de la investigación y se realiza un análisis de algunos de los motores de búsquedas más conocidos nacional e internacionalmente. Se analizan las herramientas, metodologías y técnicas utilizadas para la solución del problema planteado en la investigación.

Capítulo 2: “Análisis y diseño del subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a.”: En el capítulo se analiza y diseña la propuesta de solución. Se describen los requisitos funcionales y no funcionales que presenta, se define la arquitectura del sistema, patrones de diseño y los artefactos generados según la metodología utilizada.

Capítulo 3: “Implementación y validación del subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a.”: En este capítulo se detalla la propuesta de solución al problema planteado en la investigación. Se especifican los estándares de codificación a utilizar y se realizan las estrategias de pruebas definidas para el sistema de información geográfica. Se aplica el Criterio de expertos mediante el método Delphi para validar la hipótesis científica.

CAPÍTULO 1: “Fundamentos teóricos del subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a.”

Para lograr una mayor comprensión del alcance de la investigación, en el presente capítulo se exponen los fundamentos teóricos asociados al dominio del problema planteado, se realiza un análisis del estudio del estado. Se seleccionan las metodología, herramientas y tecnologías que serán utilizadas para dar cumplimiento a la propuesta de solución.

1.1. Conceptos asociados al dominio de la investigación

En el presente epígrafe se relacionan los principales conceptos asociados al dominio de la investigación con el objetivo de alcanzar un mejor entendimiento de la misma.

Localización geográfica

Se refiere al lugar exacto en que se encuentra un punto o territorio en la superficie terrestre según un sistema de coordenadas geográficas compuesta por latitud y longitud. Las coordenadas geográficas son un conjunto de líneas imaginarias que permiten ubicar con precisión un punto cualquiera en la superficie terrestre (Orellana, 2015).

Latitud: Distancia en grados que existe entre un punto cualquiera de la superficie terrestre y el Ecuador, medida sobre el meridiano que pasa por dicho punto (Orellana, 2015).

Longitud: Distancia en grados que hay desde un punto cualquiera de la superficie terrestre hasta el Meridiano Cero, medida sobre el paralelo que pasa sobre dicho punto (Orellana, 2015).

Sistema de Información Geográfica (SIG)

Las investigaciones de Ávila (2014), definen que en la actualidad un SIG es un conjunto integrado de medios y métodos informáticos, capaz de recoger, verificar, almacenar, gestionar, actualizar, manipular, recuperar, transformar, analizar, mostrar y transferir datos especialmente referidos a la Tierra. Los SIG son herramientas que permiten crear consultas interactivas, analizar la información espacial, editar datos como conjuntos de mapas de un territorio específico o posición geográfica del mismo según sus coordenadas, y presentar los resultados de todas estas operaciones. Además, proporciona la vinculación de diversas bases de datos.

Un SIG no es un sistema informático para hacer mapas, aunque pueda crearlos en diferentes escalas, proyecciones y colores. Es una herramienta de análisis que permite identificar las relaciones espaciales entre características de varios mapas. No almacena un mapa en sentido convencional, ni una imagen concreta o vista de un área geográfica, almacena los datos geográficos de un punto para su localización en la escala deseada. Contienen una base de datos que es la principal diferencia con un sistema de gráficos o sistema informático de cartografía, que solo pueden producir buenos gráficos (Suárez, et al., 2014).

Buscador o motor de búsqueda

Es un sistema informático que busca archivos almacenados en servidores web mediante un programa específico denominado *crawler* o araña que se ocupa de inspeccionar o rastrear las páginas de Internet de manera metódica y automatizada. Construyen una base de datos, que posteriormente se dirigen y consultan, cuando reciben la petición de un usuario a través de una búsqueda (Berreiros, 2015).

Son conocidos como *robots* de búsquedas virtuales que recorren millones de páginas en segundos y las clasifican temáticamente de acuerdo a su contenido. Los buscadores ofrecen ese mismo contenido al público por medio de palabras clave. De esta forma usualmente se localiza en la web la información necesaria a los usuarios (Valdés Pérez, et al., 2016).

1.2. Principales sistemas homólogos

Los SIG tienen un amplio campo de aplicación en diferentes aspectos como sociales-culturales, económicos y ambientales que conducen a la toma de decisiones de una manera más eficaz (Ávila del Campo, 2014). A continuación, se realizará un análisis y estudios en cuanto a sus características, funcionalidades, herramientas, ventajas y desventajas de los principales SIG para la búsqueda por localización geográfica.

1.2.1. Sistemas de Información Geográfica en el ámbito internacional

Google Maps: Es una de las aplicaciones web del motor de búsqueda Google, se encuentra desarrollada con JavaScript. Ofrece mapas de calles, planificación de rutas para viajar a pie, en coche o en transporte público y un localizador de negocios urbanos para numerosos países de todo el mundo (Google Maps, 2017).

Son varias las funcionalidades que permite la aplicación, entre las que se encuentra: ubicarse, explorar una zona, conocer una empresa o negocio, conocer más detalles de un lugar y consultar el tráfico. Proporciona una breve información sobre los lugares localizados.

Algunas de las desventajas de Google Maps son: las resoluciones de las imágenes no son muy buenas en lugares donde el Internet es limitado o no cumplen con las estrategias de posicionamiento y las bases de datos no son óptimas en cuanto a actualización, invaden la privacidad e intimidad de las personas.

Google Maps tiene acceso libre y gratuito. Debido a la implementación de las Google Maps APIs² que generalmente debe ser accesible para los usuarios sin cargo y no debe exigir una suscripción paga ni otro tipo de acceso restringido pago (Google Maps, 2017). Sin embargo, no se tiene acceso a su código fuente, debido a que pertenece a una compañía privada.

Bing Maps: Es un software de mapas interactivos con alta resolución. El software permite a sus usuarios ver ciudades y países, desde la perspectiva de un transeúnte o en tres dimensiones. Asiste paso a paso a los conductores en su ruta, mientras proporciona información respecto al tráfico, condiciones de la carretera y estructuras a los lados de ésta, mapas de calles adyacentes y ubicación de espacios para estacionamiento. Los usuarios pueden disfrutar los beneficios de esta información, encontrando paradas de autobús, estaciones de metro y de trenes, con indicaciones incluidas. Alternativamente, pueden utilizar una calculadora de tarifas de taxi, su proximidad a restaurantes y estadios (Bing Maps, 2017).

Bing Maps es una aplicación de la compañía Microsoft que utiliza tecnologías como MapPoint³, TerraServer y el lenguaje de programación ASP.NET (Bing Maps, 2017). El acceso al sistema es libre y gratuito, además brinda una breve información de los lugares localizados.

1.2.2. Sistemas de Información Geográfica en el ámbito nacional

SIG UCI: Sistema de Información Geográfica de la UCI, desarrollada por el Centro de Geoinformática y Señales Digitales (GEYSED), en el lenguaje de programación del lado servidor ECMAScript⁴. Entre sus

² API: Interfaz de Programación de Aplicaciones (por sus siglas en inglés API).

³ MapPoint: software específico creado por Microsoft que tiene como funcionalidades la capacidad de mostrar, editar e integrar mapas al servicio de lo que pida el usuario.

⁴ ECMAScript define un lenguaje de tipos dinámicos ligeramente inspirado en Java y otros lenguajes del estilo de C.

funcionalidades principales se aprecian: la búsqueda de personas, lugares, cálculo de rutas, cálculo de áreas, vistas panorámicas y una breve información de los lugares y personas localizadas. Todas las funciones enmarcadas solo para el campus universitario. El sistema está implementado bajo los estándares de un software libre (GEYSED, 2017).

Conoce Cuba: Es un directorio que permite conocer el sector empresarial cubano: negocios particulares y estatales. Contiene ofertas, eventos, noticias, artículos interesantes, un mapa de toda la isla que funciona sin necesidad de conexión a Internet, navega con GPS, ubica los puntos de interés y permite trazar rutas. Es una aplicación multiplataforma con fines comerciales (Conoce Cuba, 2018).

Andariego: Es una aplicación para dispositivos móviles, pública y de distribución libre, con un fin social y orientada al gran público, disponible para dispositivos móviles con Sistema Operativo Android. Desarrollada dentro del contexto de la Facilitación Social por la Empresa de Cartografía y Soluciones Geománticas, GeoSÍ, una entidad estatal del Grupo Empresarial GEOCUBA, con el fin de acercar a los usuarios los servicios que ofrecen las entidades estatales y del sector no estatal (Calderín, et al., 2016). Algunas de las utilidades que permite la aplicación es la gestión, búsqueda y localización de lugares, medir distancias y crear rutas. La misma es de libre acceso en el sitio oficial Andariego.

1.2.3. Resultados del estudio de los sistemas homólogos

El estudio de los sistemas que permiten la recuperación de información geográfica, en el ámbito nacional e internacional permitió realizar un resumen donde se analizaron las siguientes características:

1. Las herramientas internacionales estudiadas son sistemas privativos, utilizar las APIs que brindan crean una dependencia tecnológica y no cumple con los objetivos definidos al requerir el uso de Internet.
2. La localización en los sistemas internacionales no es eficiente en países como Cuba, la calidad de las imágenes no es óptima y las bases de datos no están actualizadas.
3. Las herramientas nacionales: Conoce Cuba es un sistema con fines comerciales, SIG UCI está diseñado para realizar búsquedas solo en el campus universitario y Andariego es una aplicación solo para dispositivos móviles con sistema operativo Android.

Se decide desarrollar un subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a. como propuesta de solución, al no poder utilizar ninguno de los sistemas antes estudiados. Sin

embargo, el análisis de los mismos permitió identificar funcionalidades y tecnologías que pueden contribuir a la implementación de la propuesta de solución como la búsqueda, localización, gestión y administración de los lugares, además de los lenguajes de programación y servidores relacionados con los mapas y su manipulación.

1.3. Metodología de desarrollo de software

En los proyectos productivos de la UCI se utiliza la metodología Proceso Unificado Ágil (por sus siglas en inglés AUP) en unión con el modelo CMMI-DEV v1.3⁵. La metodología AUP es una versión simplificada del Proceso Unificado de *Rational* (por sus siglas en inglés RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo (Sánchez, 2015):

- Desarrollo Dirigido por Pruebas (por sus siglas en inglés TDD).
- Modelado ágil.
- Gestión de cambios ágil.
- Refactorización de base de datos para mejorar la productividad.

En AUP se establecen cuatro (4) fases que transcurren de manera consecutiva: inicio, elaboración, construcción y transición. Define siete (7) disciplinas, cuatro (4) ingenieriles y tres (3) de gestión de proyectos: modelo, implementación, prueba, despliegue, gestión de configuración, gestión de proyectos y entornos. Además, cuenta con roles como: administrador de proyecto, ingeniero de procesos, desarrollador, administrador de bases de datos, modelador ágil, administrador de la configuración, administrador de pruebas y probador (Sánchez, 2015).

Con la adaptación de AUP se logra estandarizar el proceso de desarrollo de software en la UCI, dando cumplimiento además a las buenas prácticas que define CMMI-DEV v1.3. Se logra hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos. Se redujo a 1 la cantidad de

⁵ CMMI-DEV: Colección de buenas prácticas de desarrollo procedentes de la industria y del gobierno. Es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software (SEI, 2010).

metodologías que se usaban y de más de veinte (20) roles en total que se definían se redujeron a once (11) (Sánchez, 2015).

1.4. Herramientas y tecnologías

Por ser la propuesta de solución un subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a. se realiza un estudio y análisis de las herramientas, tecnologías existentes y de otras relacionadas con el desarrollo de los SIG. Los elementos planteados facilitan la futura integración y compatibilidad con el sistema.

1.4.1. Lenguajes de desarrollo

Los lenguajes de programación pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina. Son un conjunto de reglas semánticas y sintácticas que se utilizan para la codificación de instrucciones de un programa o algoritmo de programación (Álvarez, 2016). Para el desarrollo de la propuesta de solución los lenguajes de programación se dividen en dos grupos: del lado del cliente y del lado del servidor.

1.4.2. Lenguajes de programación del lado cliente

Los lenguajes de programación del lado del cliente se usan para su integración en páginas web. La base de la programación del lado cliente consiste en escribir un *script* y hacer que este funcione en el navegador sin tener que viajar hasta el servidor para ejecutarlo (Ayoze, 2017). A continuación, se muestran los lenguajes de programación del lado cliente seleccionados.

JavaScript v.1.8.5: Es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. Se utiliza parcialmente para crear páginas web dinámicas y trabaja sobre un enfoque útil y práctico. Los programas escritos en JavaScript pueden ser probados en cualquier navegador web, sin necesidad de procesos intermedios (Librosweb, 2017).

1.4.3. Lenguajes de desarrollo web

HTML5: Es un Lenguaje de Marcado de Hipertexto estándar (por sus siglas en inglés HTML), que utiliza la web para representar la información intercambiada por los usuarios en forma de documentos de hipertexto. Los documentos contienen una serie de etiquetas que indican al navegador como interpretar y dar formato al texto plano. El navegador debe hacer uso de las normas estándar de visualización de documentos web para leer e interpretar adecuadamente el HTML (Tortajada, 2014).

CCS v3.0: Hojas de Estilo en Cascada (por sus siglas en inglés CCS), es un lenguaje diseñado independiente al documento HTML, esto facilita la consistencia y homogeneidad en el diseño del sitio web. Se pueden adecuar a los distintos medios de presentación de documentos como: pantallas del ordenador, impresoras, televisores, navegadores basados en voz y documentos en BRAILLE. Presentan mecanismos para mantener un mayor control sobre el estilo con el que se muestran los elementos de las páginas (Mora, et al., 2015).

1.4.4. Lenguajes de programación del lado del servidor

Los lenguajes de programación del lado del servidor son especialmente útiles en trabajos en que se necesita acceder a información centralizada, situada en una base de datos en el servidor, y cuando por razones de seguridad los cálculos no se pueden realizar en la computadora del usuario (Ayoze, 2017). Se selecciona para el desarrollo de la propuesta de solución Procesador de Hipertexto (por sus siglas en inglés PHP) en su versión 5.2.1 producto a la compatibilidad con el módulo PHP / MapScript⁶.

PHP v5.2.1: Es el lenguaje de lado servidor más extendido en la web de contenido dinámico, es gratuito e independiente de plataforma. Por ser un lenguaje rápido, con una gran librería de funciones y un extenso volumen de documentación, proporciona a los desarrolladores potencia y simplicidad. Facilita el soporte generalizado en la mayoría de los servidores web. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser empleada en aplicaciones gráficas independientes (PHP, 2017).

1.4.5. Lenguaje y herramienta de modelado

Como lenguaje de modelado se utiliza **Lenguaje Unificado de Modelado** (por sus siglas en inglés UML) en su versión 2.1. Es un lenguaje de modelado para visualizar, especificar, construir y documentar partes de un sistema software desde distintos puntos de vista. Puede usarse con cualquier proceso de desarrollo, a lo largo de todo el ciclo de vida y puede aplicarse a todos los dominios de aplicación y plataformas de implementación. También puede usarse en otras áreas, como la ingeniería de negocio y modelado de procesos gracias a los mecanismos de adaptación/extensión mediante perfiles (García, et al., 2018).

⁶ MapScript: Es un módulo PHP cargable dinámicamente que hace que las funciones y clases MapScript de MapServer estén disponibles en un entorno PHP (MapServer, 2017).

Las herramientas de Ingeniería de Software Asistida por Computadora (por sus siglas en inglés CASE) son una de las principales herramientas para el modelado. Estas herramientas, brindan soporte a todas las actividades del proceso propuesto por la Ingeniería del Software y a todas aquellas actividades transversales que se aplican a lo largo de todo el proceso, desde la gestión del proyecto hasta la gestión del producto. Posibilitan al ingeniero automatizar actividades manuales y mejorar la visión general de la ingeniería (Battaglia, et al., 2016).

Para el modelado del subsistema de búsqueda por localización geográfica propuesto se utiliza la herramienta **Visual Paradigm** en su versión 8.0. Es una herramienta CASE multiplataforma, que soporta el ciclo completo de desarrollo de software: análisis, diseño, implementación y pruebas. Ideal para ingenieros de software, analistas y arquitectos de sistemas, que están interesados en la construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos. Permite la generación de bases de datos, conversión de diagramas entidad-relación a tablas de base de datos, mapeos de objetos y relaciones, ingeniería directa e inversa, la gestión de requisitos de software y la modelación de procesos del negocio (Visual Paradigm, 2017).

1.4.6. Marcos de trabajo

Marco de trabajo, se define como un conjunto de componentes físicos y lógicos estructurados de tal forma que permiten ser reutilizados en el diseño y desarrollo de nuevos sistemas de información. En una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado, a partir de una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Los marcos de trabajo contienen patrones y buenas prácticas que apoyan el desarrollo de un producto y un proceso con calidad (Guerrero, et al., 2014).

Como marco de trabajo para el desarrollo de aplicaciones web se utiliza **Symfony v2.8**. Ha sido ideado para el desarrollo de aplicaciones web implementadas en su totalidad en PHP. Su arquitectura interna está basada en el Modelo Vista Controlador (MVC), lo que permite reemplazar o eliminar fácilmente aquellas partes que no encajan en el proyecto. Con él se pueden crear aplicaciones y sitios web rápidos y seguros de una forma profesional (Symfony, 2017).

Para el diseño de la interfaz web de la propuesta de solución se utiliza **Bootstrap v3.3.5**. El *framework* CCS Bootstrap, escala de forma fácil y eficiente los sitios web y aplicaciones con una única base de código,

teléfonos inteligentes, tabletas u ordenadores. Contiene plantillas con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS (Bootstrap, 2016).

1.4.7. Bibliotecas

Las librerías son un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que pueden ser utilizados por el programador para realizar determinadas operaciones. Para el desarrollo de la solución propuesta se utilizan las librerías que se muestran a continuación:

Entre las principales librerías JavaScript para mapeo web, podemos destacar OpenLayers, Leaflet y Cartaro de las cuales se realizó el estudio que se mostrará a continuación (Díaz, et al., 2015).

OpenLayers: Permite la creación de mapas dinámicos en cualquier página web. Puede mostrar mosaicos de mapas, datos vectoriales y marcadores cargados desde cualquier fuente. *OpenLayers* ha sido desarrollado para promover el uso de información geográfica de todo tipo. Es completamente gratuito, de código abierto, publicado bajo la licencia *Berkeley Software Distribution* (por sus siglas en inglés BSD) de dos cláusulas también conocida como *FreeBSD*. OpenLayers ofrece un API para acceder a diferentes fuentes de información cartográfica en la red como mapas comerciales de Google Maps, Bing Maps, Yahoo! y OpenStreetMap y distintos formatos vectoriales. Permite el diseño mediante capas de mosaico, es compatible con las últimas versiones de HTML5 (OpenLayers, 2017).

Leaflet: Es una librería JavaScript *open source* distribuida bajo licencia BSD orientada a móviles con mapas interactivos, se centra en un conjunto básico de características. Está basado en HTML5 y CSS3, por lo que permite la simplicidad, rendimiento y usabilidad. Dispone de API y soporte móvil. Fácil de utilizar (Leaflet, 2017).

Cartaro: Es una plataforma de mapeo web con un sistema de código abierto. Configura y ejecuta su propio sitio web compatible con estándares del *Open Geospatial Consortium* (por sus siglas en inglés OGC). Los componentes geoespaciales utilizados en Cartaro son PostGIS, GeoServer, GeoWebCache y OpenLayers. Todos estos se administran desde el poderoso Sistema Gestor de Contenido (por sus siglas en inglés CMS) Drupal. Permite una infraestructura de datos espaciales liviana (Cartaro, 2015).

Se selecciona OpenLayers en su versión 4.5.0 por:

- Permitir la superposición de múltiples capas de mapa en una sola aplicación.
- Prestar todos los servicios de *Web Map Server*⁷ (por sus siglas en inglés WMS).
- Soportar la representación de elementos vectoriales.
- Ser un software libre.

Como librería para el almacenamiento de los datos espaciales se utilizó **PostGIS v2.3.1**. Es un módulo que añade soporte de objetos geográficos a la base de datos objeto-relacional PostgreSQL, convirtiéndola en una base de datos espacial para su utilización en SIG. Se publica bajo la Licencia Pública General de GNU. En relación con otros productos, PostGIS ha demostrado ser muy superior a la extensión geográfica de la nueva versión de MySQL y a juicio de muchos, es muy similar a la versión geográfica de la base de datos Oracle (PostGIS, 2017) .

JQuery v1.11.2: Es una biblioteca de JavaScript rápida, pequeña y rica en funciones, es software libre y de código abierto. Permite el recorrido y manipulación de documentos HTML, manejo de eventos, animación y *Asynchronous JavaScript And XML* (por sus siglas en inglés AJAX) mucho más simple con una API. Con una combinación de versatilidad y extensibilidad, JQuery ha cambiado la forma en que millones de usuarios programan JavaScript (JQuery, 2017).

1.4.8. Entorno Integrado de Desarrollo

NetBeans IDE v8.2: Es un entorno de desarrollo visual de código abierto para aplicaciones programadas mediante Java. NetBeans es posible diseñar aplicaciones con solo arrastrar y soltar objetos sobre la interfaz de un formulario. Posibilita la elaboración de potentes aplicaciones para el escritorio, web y dispositivos portátiles, como teléfonos inteligentes o tabletas, sin que cambie la forma de programar. La programación se realiza a través de componentes de software modulares, también llamados módulos. Incluye un avanzado editor para varios lenguajes, editor de perfiles y un detector de errores, además de herramientas para el

⁷ WMS: Es un estándar que ofrece una sencilla interfaz HTTP, permite realizar una solicitud de imágenes de mapas georreferenciados de una o más bases de datos geográficas distribuidas en más de un servidor. Permiten la visualización, superposición y consulta puntual de mapas generados desde uno o varios servidores en diferentes entidades públicas o privadas.

control de versiones y el desarrollo colaborativo. Proporciona decenas de módulos a través de su página web, que podrás integrar en él para conseguir mejores aplicaciones (NetBeans IDE, 2016).

1.4.9. Servidores de mapas

Para facilitar el proceso de publicación de información geográfica actualizada en tiempo real, y de forma más económica existen los servidores de mapas. Son los encargados de renderizar datos tanto vectoriales como ráster⁸ en diferentes estilos y proyecciones cartográficas, que cumplen con el estándar WMS definido por el OGC⁹. Generan y difunden mapas en la web aprovechando la arquitectura Cliente-Servidor. El cliente puede ser un navegador con un visualizador de mapas o una aplicación que realice una petición al servidor de mapa. La interacción se realiza a través de Internet o la Intranet corporativa, y el servidor de mapas interpretará la petición, recuperará la información de la base de datos espacial o archivo y devolverá una imagen u objeto geográfico de forma interactiva y dinámica (Sayago, et al., 2018).

Entre los servidores de mapas más populares se encuentran MapServer, GeoServer, Deegree (Tamayo, et al., 2015).

Mapserver v6.4.1: Es una plataforma de código abierto para la publicación de datos espaciales y aplicaciones de mapeo interactivo en la web. Permite crear mapas de imágenes geográficas, que pueden dirigir a los usuarios al contenido. Se lanzó bajo una licencia de estilo MIT¹⁰ y se ejecuta en todas las plataformas principales (Windows, Linux, Mac OS X). Proporciona la creación de un SIG con el fin de visualizar, consultar y analizar información geográfica. Presenta la salida cartográfica avanzada que consiste en el dibujo de características dependientes de escala y ejecución de la aplicación. El etiquetado de características que incluye la mediación de colisión de etiquetas y es totalmente personalizable, salida impulsada por plantilla. Tiene soporte para entornos populares de scripting y desarrollo como son: PHP, Python, Perl, Ruby, Java y .NET; es multiplataforma. Es compatible con multitud de formatos de datos de

⁸ Modelos ráster y vectorial. Sistemas de Información Geográfica. La mayoría de los elementos que existen en la naturaleza pueden ser representados mediante formas geométricas (puntos, líneas o polígonos, esto es, vectores) o mediante celdillas con información (ráster).

⁹ El *Open Geospatial Consortium* (por sus siglas en inglés OGC) es una organización internacional sin fines de lucro comprometida con la elaboración de estándares abiertos de calidad para la comunidad geoespacial global.

¹⁰ MIT: Es una de las licencias de software que se origina en el Instituto Tecnológico de Massachusetts (en inglés *Massachusetts Institute of Technology*).

trama y vectoriales, se pueden destacar PostGIS, Oracle Spatial y MySQL. Archivos Shap de ESRI a través de OGR. Proyecta los mapas sobre la marcha con miles de proyecciones a través de la biblioteca PROJ.4. Cumple con los estándares planteados por OGC para los servidores de mapas. (MapServer, 2017).

GeoServer: Es un servidor de software basado en Java que permite a los usuarios ver y editar datos geoespaciales. Mediante el uso de estándares abiertos establecidos por el OGC permite una gran flexibilidad en la creación de mapas y el intercambio de datos. Es un software libre lo que reduce significativamente la barrera financiera a la entrada en comparación con los productos SIG tradicionales, es de código abierto. Puede mostrar datos en cualquiera de las populares aplicaciones de mapas como Google Maps, Google Earth, Yahoo! Maps y Microsoft Virtual Earth. Además, GeoServer puede conectarse con arquitecturas GIS tradicionales como ESRI ArcGIS (GeoServer, 2014).

Deegree: Los servicios web Deegree son implementaciones de las especificaciones del servicio geoespacial web del OGC. Proporciona el acceso a objetos de datos geoespaciales en bruto, mapas representados a partir de estos datos, mosaicos de mapas pretratados y realiza la búsqueda de datos y servicios espaciales. Admite numerosos *backends*, como PostGIS, Oracle Spatial, MS SQL Server, *shapefiles* o documentos de instancia GML. Es distribuido bajo la Licencia Pública General Reducida de GNU. Es un software libre, por lo que permite la interoperabilidad (deegree, 2018).

Luego de haber analizado algunos de los servidores de mapas más utilizados se concluye que estos son software libre, de código abierto y cumplen con estándares abiertos establecidos por el OGC permitiendo la flexibilidad en la creación de los mapas.

Para la selección del servidor de mapa se tuvo en cuenta los servicios disponibles ya sea en mapas (WMS, WFS), teselas¹¹ (WMTS, TMS), datos brutos (llámese vectorial, ráster WCS), metadatos (CSW) y geo procesos (WPS). La sencillez de configuración y administración, si es multiplataforma, la facilidad del acceso y velocidad de los datos, la calidad en la obtención de los resultados (imagen del mapa), costo de empleo y desarrollo. Por tales motivos se selecciona MapServer en su versión 6.4.1, además de cumplir estas características garantiza fiabilidad y sostenibilidad a la propuesta de solución.

¹¹ Tesela: pequeña pieza de piedra, terracota o vidrio coloreado que se utiliza para confeccionar un mosaico.

1.4.10. Servidor de Bases de Datos

Un servidor de bases de datos se utiliza para almacenar, recuperar y administrar los datos de una base de datos. El servidor gestiona las actualizaciones de datos, permite el acceso simultáneo de muchos servidores o usuarios web y garantiza la seguridad y la integridad de los datos (Gabillaud, 2015).

Se utiliza **PostgreSQL** en su versión 10.1 como servidor de base de datos para la propuesta de solución. Es un sistema de base de datos relacional de objetos de código abierto. Tiene más de 15 años de desarrollo activo y una arquitectura comprobada que le ha valido una sólida reputación de fiabilidad, integridad de datos y corrección. Se ejecuta en todos los principales sistemas operativos, incluidos Linux y Windows. Incluye la mayoría de los tipos de datos SQL. También es compatible con el almacenamiento de objetos grandes binarios, incluyendo imágenes, sonidos o video. Cuenta con funciones sofisticadas tales como Control de concurrencia de varias versiones (*Multi-Version Concurrency Control*, MVCC), recuperación puntual, replicación asincrónica, transacciones anidadas (puntos de rescate), copias de seguridad en línea caliente, un sofisticado planificador optimizador de consultas y escritura registro anticipado para la tolerancia a fallas (PostgreSQL, 2017).

1.4.11. Servidor web

Servidor web o también llamado servidor HTTP es un programa informático que espera permanentemente las peticiones del navegador que realizan los clientes web, la principal función del servidor es almacenar archivo de un sitio específico y lanzarlo por internet que puede ser visitado por los usuarios (Sánchez, et al., 2017).

Apache v2.2: Es un servidor web de código abierto, multiplataforma, flexible, rápido y eficiente, altamente configurable por su diseño modular, esta característica permite ampliar considerablemente sus capacidades pues existe un repositorio extenso completamente gratuito de extensiones y módulos (Apache, 2017).

1.5. Resumen del capítulo

El estudio y análisis de los elementos teóricos que sustentan a la propuesta de solución permitió realizar los siguientes resúmenes:

1. El estudio de los principales conceptos asociados al dominio de la investigación y la relación existente entre ellos, posibilitó alcanzar un mayor dominio referente a las técnicas de recuperación de información geográfica.

2. El análisis y estudio de las soluciones existentes para la recuperación de información geográfica permitió identificar funcionalidades que pueden contribuir al desarrollo de la propuesta de solución.
3. El análisis de la documentación de la plataforma de servicios c.u.b.a., sobre la metodología de desarrollo, herramientas, tecnologías y lenguajes de programación utilizados en su implementación, permite especificar el ambiente de desarrollo para la propuesta de solución

CAPÍTULO 2: “Análisis y diseño del subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a.”

Para lograr una mayor comprensión de la propuesta de solución, el presente capítulo tiene como objetivo mostrar los resultados que se obtienen una vez concluida la etapa de análisis y diseño que propone la metodología AUP-UCI. Se describen las características del subsistema, así como la especificación de los requisitos funcionales y no funcionales que debe cumplir. También se presentan los artefactos generados para dar solución al problema planteado en la investigación.

2.1. Modelo conceptual

Un modelo conceptual es un artefacto de la disciplina de análisis, construido con las reglas UML durante la fase de concepción. Los objetos de dominio representan las cosas, entidades, ideas, conceptos u objetos del “mundo real” o dominio de interés. Tiene como objetivo comprender y describir las clases más importantes, así como, identificar y explicar los conceptos significativos en el dominio del problema, identificando los atributos y las asociaciones existentes entre ellos (Sommerville, 2011).

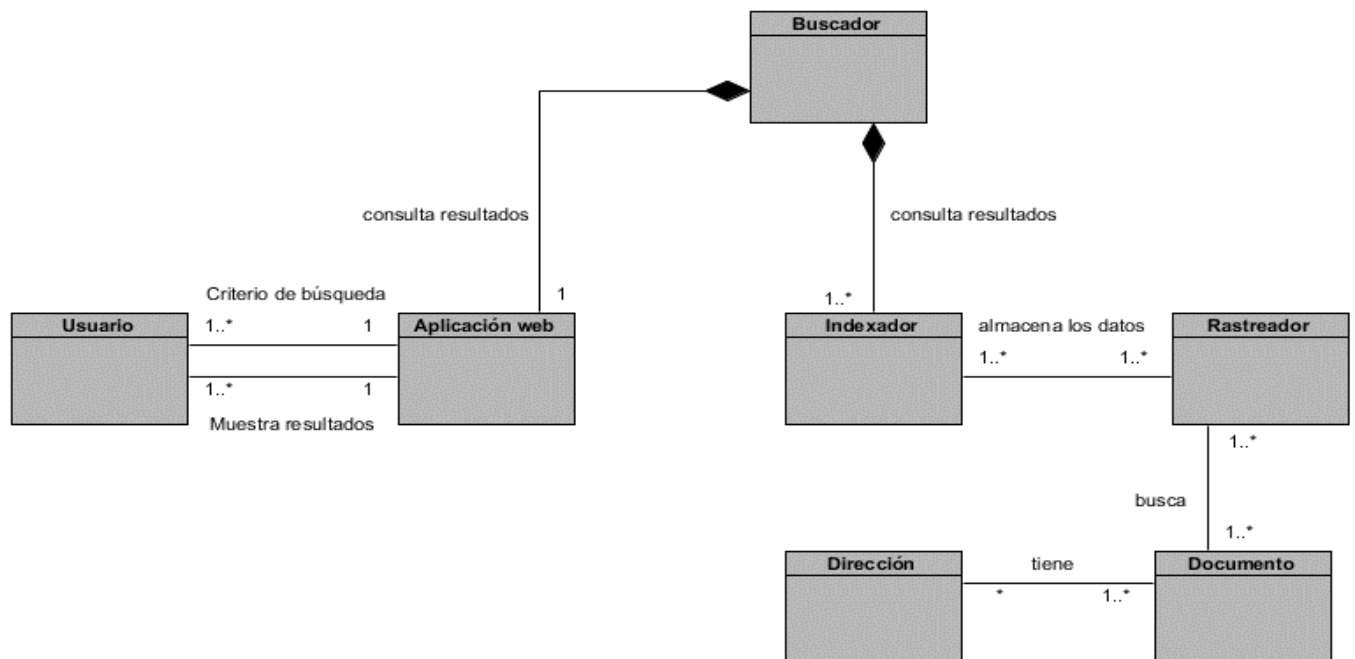


Figura 1: Modelo conceptual

- **Usuario:** Persona que interactúa con la aplicación web.
- **Aplicación web:** Elemento del buscador web que recibe el criterio de búsqueda de los usuarios, consulta el resultado con el buscador y se lo muestra a los usuarios.
- **Buscador:** Sistema recolector de información que recibe las consultas solicitadas por la aplicación web y se las transmite al indexador.
- **Indexador:** Elemento del buscador web que almacena los resultados del rastreo solicitado por el buscador.
- **Rastreador:** Elemento del buscador web que almacena la información pública en la red y procesa los resultados encontrados.
- **Documento:** Tipo de contenido digital.
- **Direcciones:** Tipo de contenido digital del cual se almacenan los datos para ser mostrado a los usuarios.

2.2. Descripción de la propuesta de solución

El subsistema de búsqueda por localización geográfica que se propone desarrollar tiene como objetivo optimizar la búsqueda de direcciones dentro de la plataforma de servicios c.u.b.a. y les brinda a los usuarios una fuente de conocimiento sobre las localizaciones geográficas del país. La principal característica de la propuesta de solución es señalar la región y localizar en el mapa de Cuba los lugares deseados por los usuarios con una alta calidad en las imágenes. Permite a los usuarios autenticados en la plataforma de servicios c.u.b.a. la gestión de lugares con la autorización previa de los administradores del sistema. Facilita la navegación sobre el mapa (acercarse, alejarse y desplazarse) y realiza búsquedas por categorías, nombre de sitios y por coordenadas.

2.3. Arquitectura del sistema

La arquitectura de un SIG debe permitir la interacción y manejo de grandes volúmenes de datos espaciales. Está compuesta por un dispositivo cliente (ordenador, *Smartphone* o *Tablet*) que se conecta con un servidor remoto de mapas, que genera a través de librerías de cálculos espaciales el modelo geográfico con la ayuda de una base de datos geográfica (INEGI, 2017).

Dada la arquitectura de un SIG explicada anteriormente se sugiere la arquitectura que se muestra en la Figura 2.

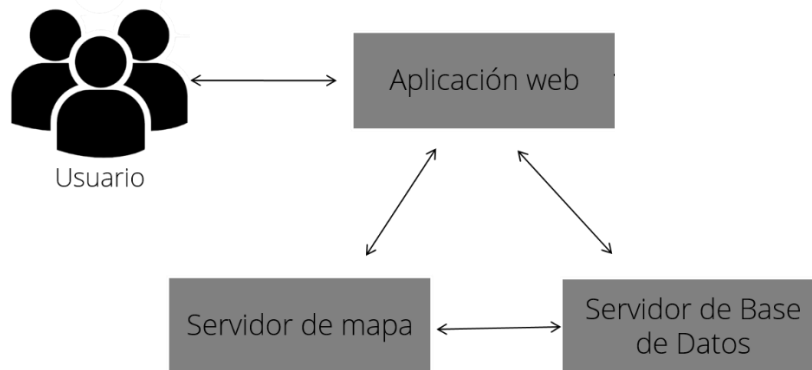


Figura 2: Arquitectura del subsistema propuesto

El sistema propuesto debe tener un componente **Aplicación web** que contiene la interfaz del sistema e interactúa con los componentes **Servidor de Mapa** y **Servidor de Base de Datos**. El Servidor de Mapa permite renderizar los datos vectoriales para obtener como resultado la imagen del mapa, el mismo intercambia con el Servidor de Base de Datos que facilita el almacenamiento de los datos espaciales y posibilita la localización exacta de los lugares.

2.4. Requisitos de la propuesta de solución

Los requisitos se pueden clasificar en funcionales y no funcionales. Pueden documentar las interfaces externas, describir la funcionalidad y el rendimiento del sistema. En esta sección describen los servicios que se proponen al usuario (Sommerville, 2011).

2.4.1. Requisitos funcionales

Los requisitos funcionales (RF) describen lo que el sistema debe hacer. Dependen del tipo de software que se desarrolle, los posibles usuarios y el enfoque tomado por el proyecto. Describen con detalle la función del sistema, sus entradas, excepciones y salidas (Sommerville, 2011).

En la Tabla 1 se muestran los requisitos funcionales de la propuesta de solución, a los cuales se les asignó una prioridad teniendo en cuenta la importancia fijada por el cliente a partir de sus necesidades.

Tabla 1. Requisitos funcionales

N °	Nombre	Prioridad	Complejidad
RF1	Realizar zoom del mapa	Media	Baja
RF2	Mover mapa	Media	Baja
RF3	Búsqueda de lugar por nombres de los lugares	Alta	Alta
RF4	Búsqueda de lugar por coordenadas de los lugares	Alta	Alta
RF5	Localizar lugar	Alta	Alta
RF6	Insertar lugar	Alta	Media
RF7	Eliminar lugar	Alta	Media
RF8	Editar lugar	Alta	Media

2.4.2. Requisitos no funcionales

Los requisitos no funcionales (RnF) hacen referencia a las propiedades emergentes del sistema: fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento, son limitaciones sobre servicios o funciones que ofrece el mismo. Los requisitos no funcionales se aplican al sistema como un todo, más que a características o a servicios individuales (Sommerville, 2011).

Requisitos de usabilidad

RnF1. El subsistema de búsqueda por localización geográfica debe presentar una interfaz agradable e intuitiva para el usuario.

Requisitos de hardware

RnF2. Los servidores destinados para el traductor de consultas y la aplicación web deben tener una memoria *RAM* de 2GB o superior y un procesador a una velocidad 2.10 GHz o superior.

RnF3. Los servidores para el mapeo deben tener un disco duro de más de 500 GB de almacenamiento.

Requisitos de software

RnF4. Los servidores donde se instalarán cada uno de los componentes del sistema deben tener una distribución GNU/Linux Ubuntu v14.04.

RnF5. Los dispositivos de los usuarios deben contar con navegadores web que soporten HTML5 y CCS3.

RnF6. Se requiere la instalación de un servidor web (recomendándose Apache en su versión 2.4) y PHP v5.5 para el correcto funcionamiento de la aplicación web.

RnF7. Se requiere la instalación de un servidor de mapa (recomendándose Mapserver en su versión 6.4.1) para el correcto funcionamiento de los mapas.

Requisitos de seguridad

RnF8. La información manejada en el sistema debe estar protegida del acceso no autorizado mediante los mecanismos de control de acceso que establece la plataforma de servicios c.u.b.a.

Requisitos de eficiencia

RnF9. El sistema debe permitir que más de 1000 usuarios interactúen con él de manera concurrente.

Requisitos de diseño

RnF10. Como lenguajes de programación para el subsistema de búsqueda por localización geográfica se deberá utilizar PHP v5.5, JavaScript v1.8.5 con la librería OpenLayers v4.

RnF11. Para el desarrollo del subsistema de búsqueda por localización geográfica se deberá utilizar Symfony v2.8 como marco de trabajo.

Requisitos para la documentación de usuarios en línea de ayuda del sistema

RnF12. La interfaz gráfica del subsistema de búsqueda por localización geográfica debe cumplir con las pautas de diseño definidas por el cliente.

2.5. Casos de uso

Un caso de uso (CU) es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Describe los pasos o actividades que se deben realizar los proceso. Los personajes o entidades que participan en un caso de uso se denominan actores. Sigue una secuencia de interacciones que se desarrollan entre un sistema y sus actores en respuesta a un evento que inicia un actor principal para conseguir un objetivo específico (Pressman, 2010).

Tabla 2: Tabla de casos de uso

Referencia a requisitos	Nombre del caso de uso
RF1, RF2	CU1: Manipular mapa
RF3, RF4, RF5	CU2: Realizar búsqueda
RF6, RF7, RF8	CU3: Gestionar lugar

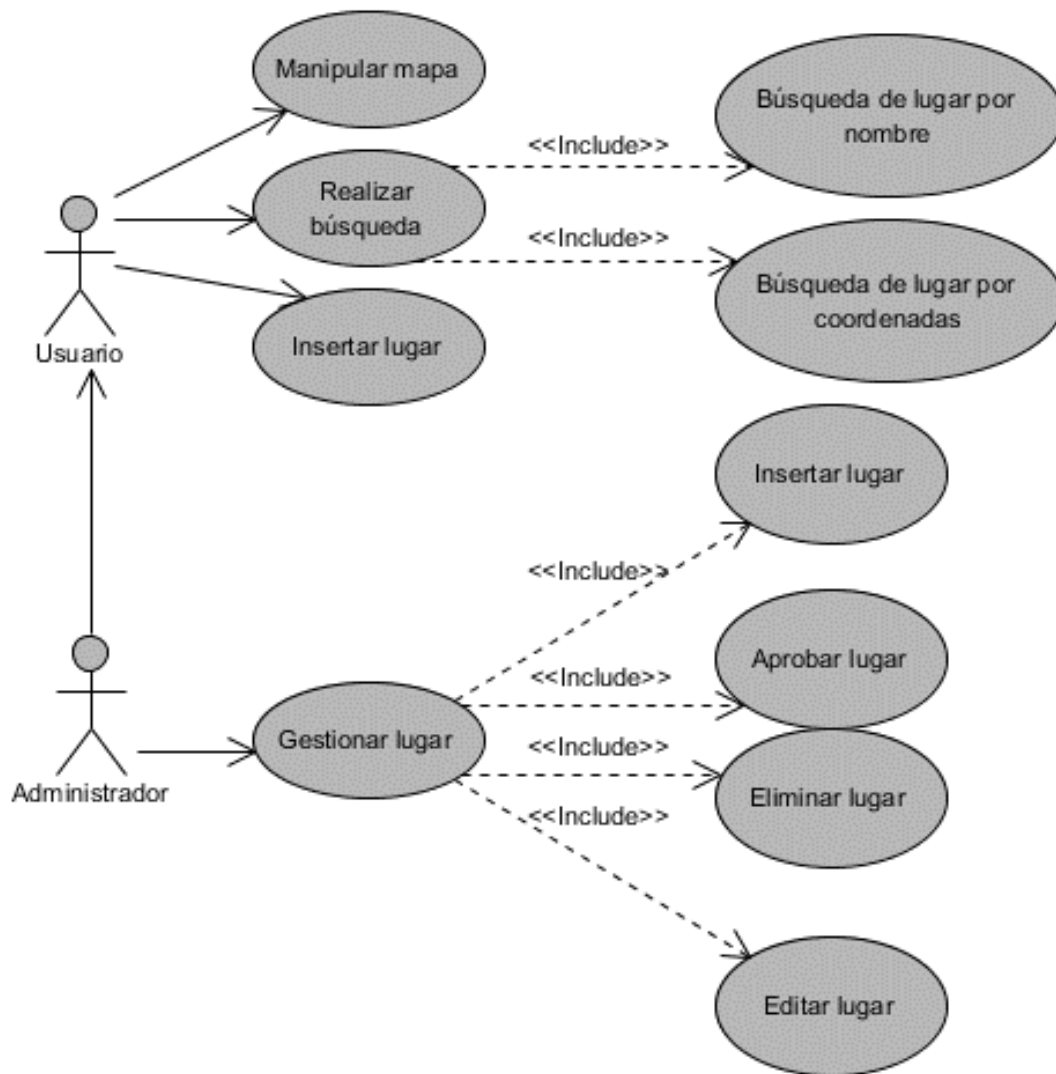


Figura 3: Diagrama de casos de uso

2.5.1. Especificación de casos de uso

Tabla 3: CU1: Manipular mapa

Objetivo	Alejar, acercar y mover el mapa	
Actores	Usuario	
Resumen	El usuario se desplaza por el mapa, además de acercarlo y alejarlo	
Complejidad	Baja	
Prioridad	Media	
Precondiciones	El usuario debe haber entrado a la plataforma de servicios c.u.b.a.	
Postcondiciones	Se debe visualizar todo el mapa	
Flujo de eventos		
Flujo básico Manipular mapa		
	Actor	Sistema
1.	Entra a la plataforma de servicios c.u.b.a. y selecciona “Mapas”	
2.		Se abre la web de “Mapa”
3.	Manipula el mapa	
Relaciones	CU incluidos	
	CU extendidos	

Tabla 4: CU2 Realizar búsqueda

Objetivo	Realizar las búsquedas solicitadas por los usuarios	
Actores	Usuario	
Resumen	El usuario debe realizar las búsquedas por coordenadas o por nombres de lugares	
Complejidad	Alta	
Prioridad	Alta	
Precondiciones	El usuario debe haber entrado a la plataforma de servicios c.u.b.a	
Postcondiciones	Se deben mostrar en el mapa las localizaciones de los lugares de las búsquedas solicitadas	
Flujo de eventos		
Flujo básico Realizar búsqueda		
	Actor	Sistema
1.	Realiza la búsqueda por nombre o por coordenadas	
2.		Muestra la localización de los lugares que estén almacenados en la base de datos

Flujos alternos		
N ° No se encontró el lugar		
	Actor	Sistema
3.		Muestra una alerta "No disponible"
Relaciones	CU incluidos	Realizar búsqueda del lugar por coordenadas Realizar búsqueda del lugar por nombre
	CU extendidos	

Tabla 5: CU3 Gestionar lugar

Objetivo	Gestionar los lugares	
Actores	Administrador, usuario	
Resumen	El usuario debe insertar un lugar y el administrador aprueba si se inserta o no, además el usuario puede gestionar sus lugares	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	El usuario debe haber insertado un lugar	
Postcondiciones	El administrador aprueba y gestiona los lugares	
Flujo de eventos		
Flujo básico gestionar lugar		
	Actor	Sistema
1.	El usuario inserta un lugar	
2.	El administrador aprueba el lugar	
3.		Muestra el lugar
4.	El usuario elimina o edita el lugar si es necesario	
5.		Muestra la modificación
Relaciones	CU incluidos	Insertar lugar Editar lugar Eliminar lugar
	CU extendidos	

2.6. Patrones utilizados

Los patrones de diseño son una técnica para resolver problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Establece una relación entre un determinado

contexto, un problema y una solución. Para el diseño de software, el contexto permite al lector entender el ambiente en el que reside el problema y qué solución sería apropiada en dicho ambiente. Un conjunto de requerimientos, incluidas limitaciones y restricciones, actúan como sistema de fuerzas que influyen en la manera en la que puede interpretarse el problema en este contexto y en cómo podría aplicarse con eficacia la solución (Pressman, 2010).

2.6.1. Patrones de casos de uso

Para el modelado de los casos de uso del subsistema de búsqueda por localización geográfica, su correcta estructura y organización, se utilizan los patrones de casos de uso, específicamente "Inclusión concreta". Este patrón indica que cuando se tiene algún comportamiento parcial común a varios casos de uso, es conveniente separarlo en uno propio e indicar su inclusión para evitar duplicaciones (Pressman, 2010).

Esta inclusión se indica en la Figura 2, cuando se crea el CU Realizar búsqueda ya que incluye a los CU Realizar búsqueda por nombre y Realizar búsqueda por coordenadas. También se ve reflejado en el CU Gestionar lugar, en el mismo se incluye los CU Eliminar, Editar y Aprobar lugar. Estos últimos tienen en común la gestión del lugar y en el primer ejemplo realizar la búsqueda, es por ello por lo que se decide separarlos en CU independientes.

2.6.2. Patrones Generales de Software para la Asignación de Responsabilidades (GRASP)

Los patrones GRASP son patrones generales de software para asignación de responsabilidades, son una serie de "buenas prácticas" de aplicación recomendable en el diseño de software (Cárdenas, 2014). Para el desarrollo de la propuesta de solución se hace uso de los siguientes patrones (Hamon, 2014):

Experto: Es uno de los patrones que más se utiliza cuando se trabaja con Symfony, con la inclusión de la librería Doctrine para mapear la base de datos. Symfony utiliza esta librería para realizar su capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases con todas las funcionalidades comunes de las entidades, las clases de abstracción de datos poseen un grupo de funcionalidades que están relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla que representan. En el subsistema de búsqueda por localización geográfica se evidencia en la clase entidad *Lugar.php*, encargada de obtener la información sobre los datos espaciales de los lugares almacenados en la base de datos.

Bajo Acoplamiento: Symfony permite la organización de la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Es una medida de la fuerza en las conexiones de una clase con

otras, la recurrencia y las conexiones a ella. Una clase con bajo acoplamiento no depende de muchas clases. Su utilización se evidencia en que las clases controladoras del sistema no se relacionan entre sí, lo que disminuye las dependencias entre las mismas.

Controlador: Permite que a través de un archivo se procesen todas las peticiones de manipulación por analizar a través de un objeto de controlador único. Estas peticiones web son tratadas por un solo controlador frontal que posee el marco de trabajo llamado “app.php” siendo este el único punto de entrada a la aplicación. En el subsistema de búsqueda por localización geográfica los eventos generados por el usuario son redirigidos a la clase controladora *MapaController.php* encargada de realizar las operaciones existentes.

2.6.3. Patrón arquitectónico

Los patrones arquitectónicos se abocan a un problema de aplicación específica dentro de un contexto dado y sujeto a limitaciones y restricciones. El patrón propone una solución arquitectónica que sirve como base para el diseño de la arquitectura (Pressman, 2010). Para la implementación de la propuesta de solución se utiliza el patrón arquitectónico Modelo Vista Controlador.

Modelo Vista Controlador (MVC): Es un patrón de arquitectura de software que se basa en separar la codificación de los programas de acuerdo con sus concernientes responsabilidades como lo son los datos, interfaz de usuario y la lógica del negocio en sus respectivas capas: modelo, vista y controlador. El patrón MVC es muy utilizado para el desarrollo de aplicaciones web (Astudillo, et al., 2015).

Modelo: En esta capa es donde se ubican los datos por lo que permite acceder y modificar dicha información. Estos datos están almacenados en una base de datos (Herrera, 2016). El modelo está representado en los archivos del directorio: *Entity*.

Vista: Es la interfaz que es visible al usuario, con el objetivo de mostrar datos del objeto (Herrera, 2016). Las vistas están representadas en los archivos del directorio: *app/Resources/view*.

Controlador: Es el que se encuentra en el medio por lo que actúa como mediador entre las capas del modelo y vista. Es la que responde las acciones que se solicitan en la aplicación por ejemplo la visualización de un elemento (Herrera, 2016). El controlador está representado en los archivos del directorio: *Controller*.

En el controlador se encuentran las acciones que permiten la gestión y localización de los lugares del sistema de información geográfica. Estas acciones obtienen los datos del modelo, capa que permite el

manejo de la información espacial relacionada con los lugares almacenados en la base de dato del sistema. La información es enviada por la capa controlador a la vista, encargada de renderizar y mostrar los resultados al usuario. Atendiendo a lo anteriormente planteado y a que el marco de trabajo seleccionado para el desarrollo de la propuesta de solución Symfony2 establece el uso de este patrón, se decide utilizarlo en el desarrollo del subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a.

2.7. Diagrama de clases de diseño

Un diagrama de clases de diseño (DCD) puede usarse cuando se desarrolla un modelo de sistema orientado a objetos para mostrar la especificación de las clases de una aplicación. Incluye las clases con sus asociaciones y atributos, las interfaces, con sus operaciones y constantes, los métodos, navegabilidad y las dependencias (Sommerville, 2011). Para el desarrollo del subsistema propuesto se generaron tres (3) DCD en la Figura 4 se ilustra el DCD del CU Manipular mapa, en la Figura 5 el DCD del CU Gestionar lugar y en la Figura 6 el DCD del CU Realizar búsqueda.

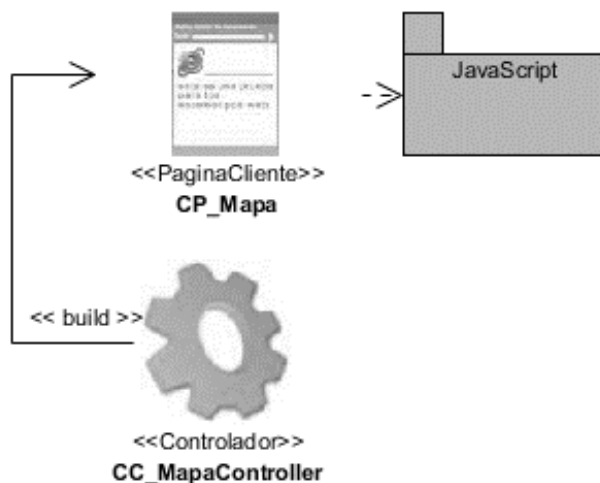


Figura 4: DCD del CU Manipular mapa

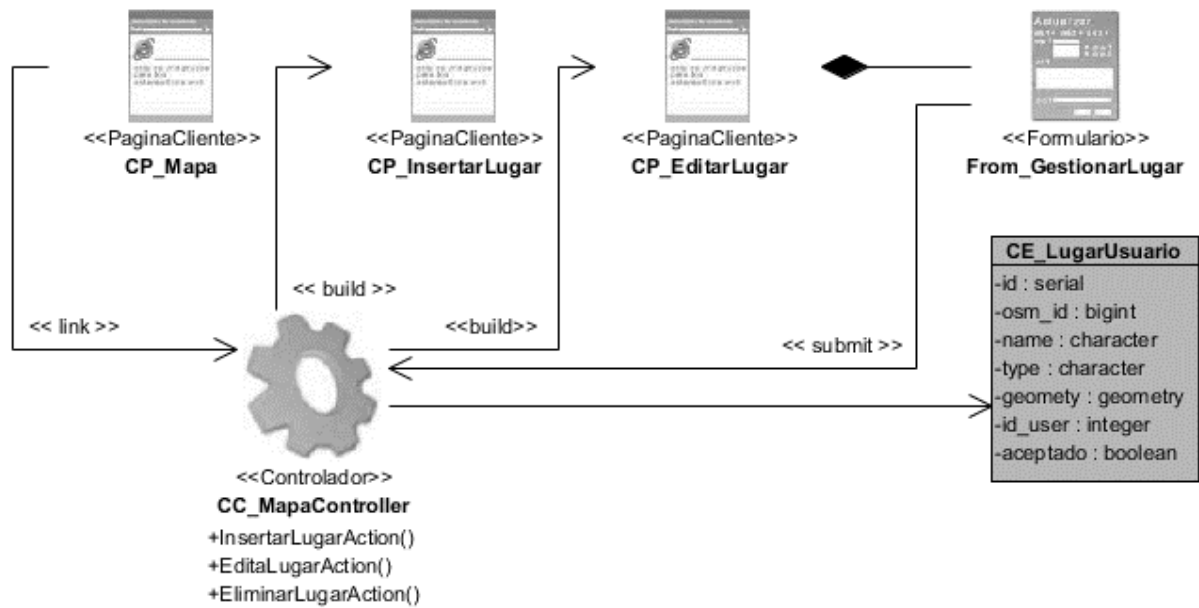


Figura 5: DCD del CU Gestionar lugar

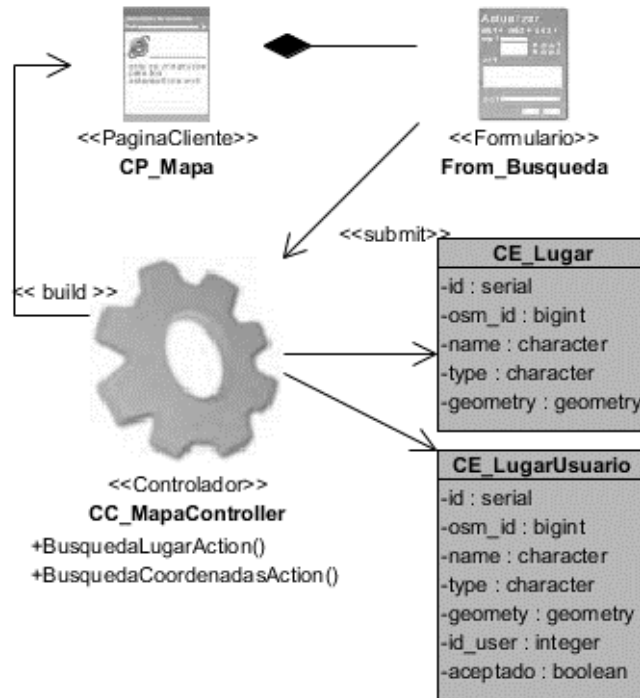


Figura 6: DCD del CU Realizar búsqueda

2.8. Diagrama de secuencia

Un diagrama de secuencias (DS) muestra la interacción de un conjunto de objetos de una aplicación a través del tiempo. Se indican los módulos o clases que formaran parte del sistema. Muestra una interacción, que representa la secuencia de mensajes entre instancias de clases, componentes, subsistemas o actores. Permite observar la perspectiva cronológica de las interacciones. Los diagramas de secuencias se realizan a partir de la descripción de un caso de uso (Cevallos, 2015). Se generaron tres (3) DS en la Figura 7 se muestra el DS del CU Manipular mapa, en la Figura 8 el DS del CU Realizar búsqueda y en la Figura 9 el DS del CU Gestionar lugar.

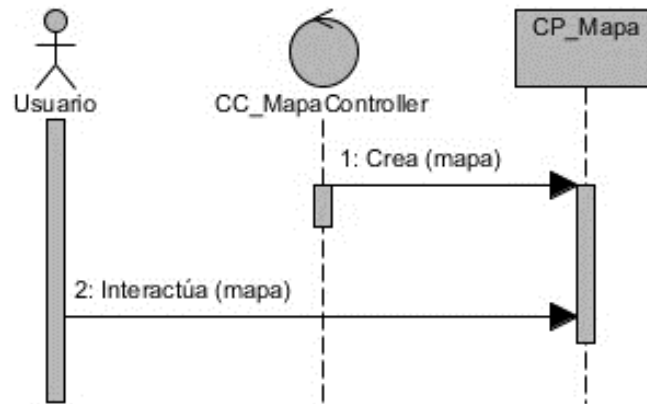


Figura 7: DS del CU Manipular mapa

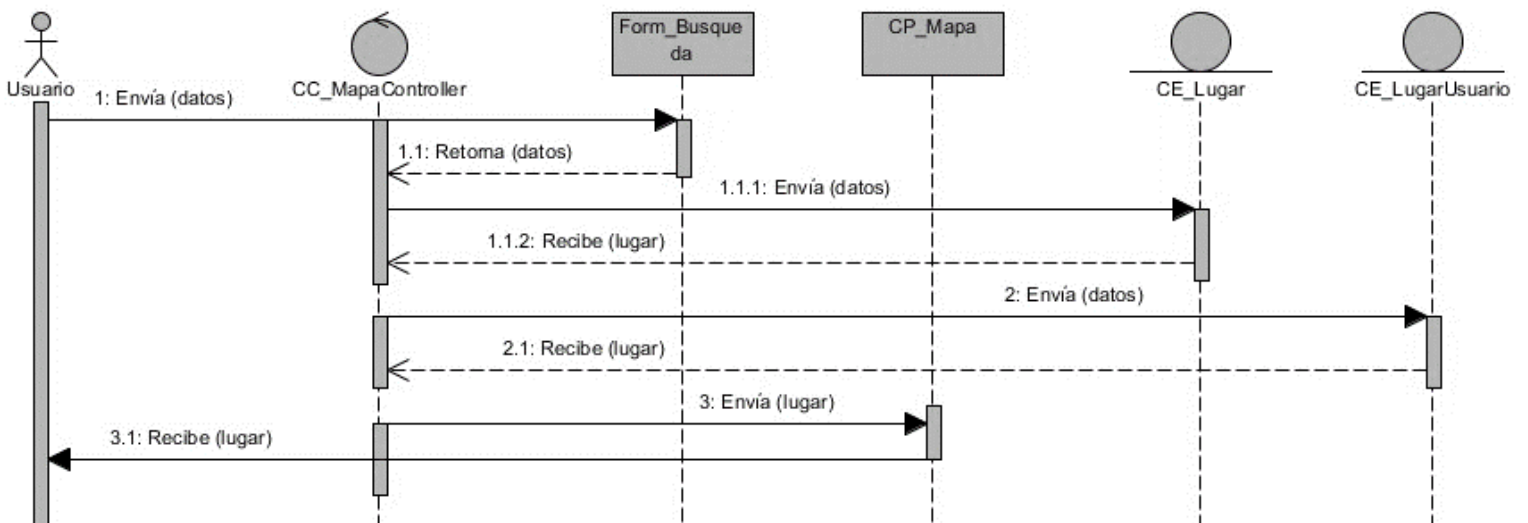


Figura 8: DS del CU Realizar búsqueda

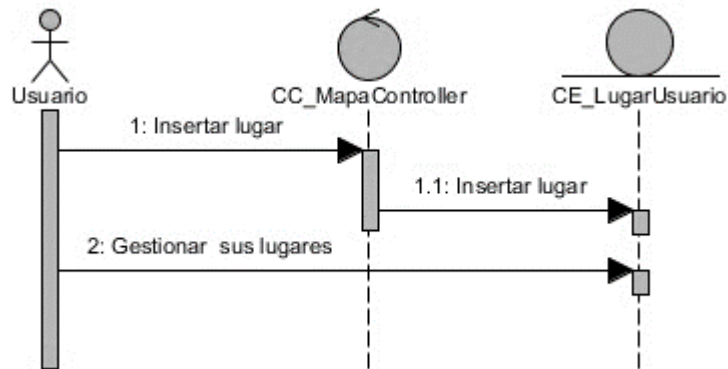


Figura 9: DS del CU Gestionar lugar

2.9. Modelo de datos

Un modelo de base de datos muestra la estructura lógica de la base, incluidas las relaciones y limitaciones que determinan cómo se almacenan los datos, la relación que existe entre sí, los procesos que los transforman y cómo se accede a ellos. Se basa en la identificación de los objetos primarios que va a procesar el sistema, la composición y atributos de los mismos. En algunos casos, esta base de datos es independiente del sistema software (Sommerville, 2011). En la Figura 10 se observa el modelo de datos generado para el subsistema propuesto, cuenta con tres tablas Usuario, Lugar y Usuario_Lugar.

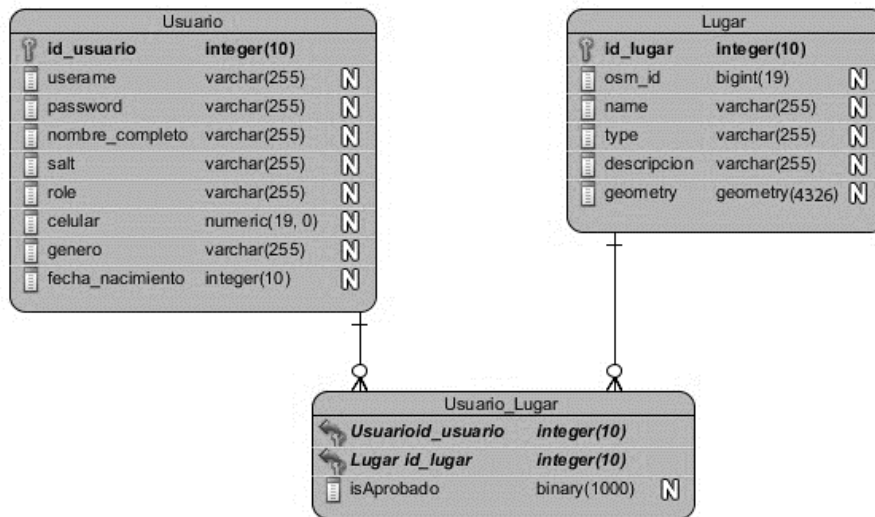


Figura 10: Modelo de datos

Usuario: Registra la información referente a los usuarios del sistema, de los mismos almacena los atributos: identificador, nombre, contraseña, semilla para encriptación de la contraseña, el rol que desempeña, el número del celular, género, correo electrónico y fecha de nacimiento.

Lugar: Registra la información de los lugares, de los mismos almacena los atributos: identificador, nombre, tipo y descripción, además, almacena el identificador de la base de datos OpenStreetMap y los datos espaciales para obtener las coordenadas.

Usuario_Lugar: Esta tabla surge de la relación de mucho a mucho entre las tablas Usuario y Lugar, por lo que almacena los atributos de ambas y uno en particular que decide si el lugar insertado por el usuario fue aprobado o no.

2.10. Diagrama de despliegue

El diagrama de despliegue muestra la configuración de nodos de procesamiento en tiempo de ejecución y las instancias de componentes y objetos que se encuentran dentro de esos nodos. Los componentes representan manifestaciones de ejecución de unidades de código. Es un diagrama estructurado que muestra la arquitectura del sistema desde el punto de vista de distribución de los artefactos del software en los destinos de despliegue; definiendo los artefactos como representaciones de elementos concretos en el mundo físico que son el resultado de un proceso de desarrollo (Sarmiento, 2016). A continuación, se muestra en la Figura 11 el diagrama de despliegue propuesto para el subsistema de búsqueda por localización geográfica. En el diagrama muestra la distribución física de los componentes del sistema propuesto, donde se define el nodo Dispositivo cliente que origina las peticiones mediante el protocolo HTTPS al subsistema que se encuentra en el servidor web Apache representado por c.u.b.a. Mapa. Los servidores de mapa MapServer y de base de datos PostgreSQL se comunican a través del protocolo TCP que a su vez los mismos mantienen una comunicación con la plataforma de servicios c.u.b.a. por los protocolos HTTPS y TCP respectivamente.

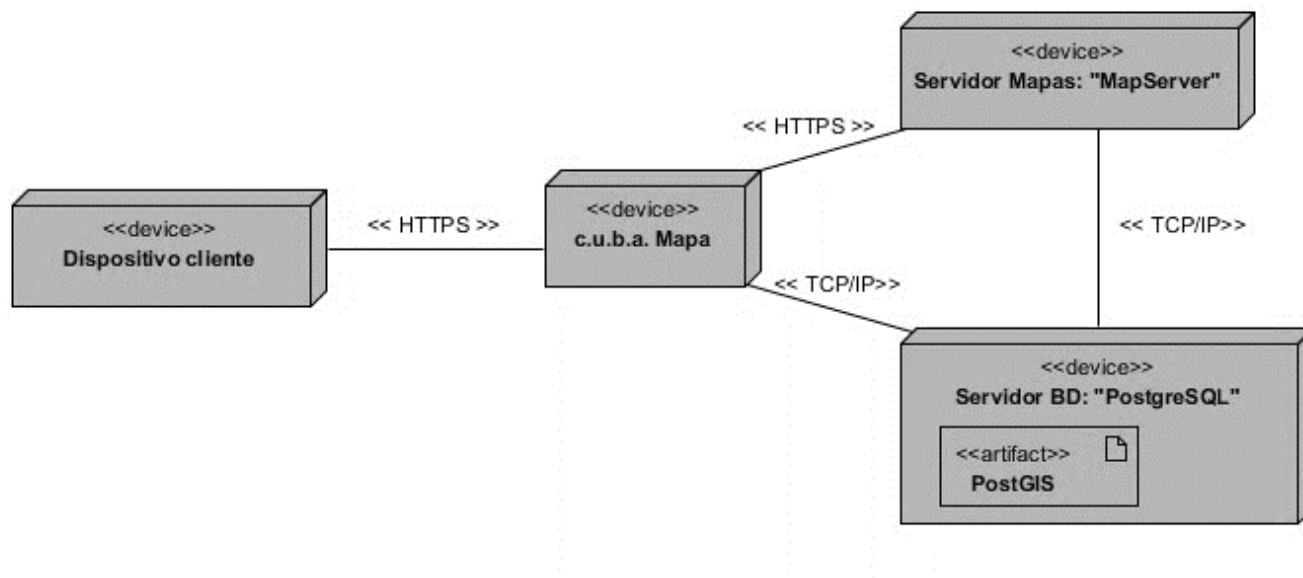


Figura 11: Diagrama de despliegue

2.11. Resumen del capítulo

Realizado el análisis y diseño de la propuesta de solución y haber generado los diferentes artefactos la autora resume que:

1. El análisis de las características del subsistema de búsqueda por localización geográfica y la modelación del dominio, permiten identificar los principales requisitos funcionales y no funcionales del mismo, los cuales fueron agrupados y categorizados por casos de uso.
2. La identificación de los patrones de diseño y la arquitectura propuesta para el subsistema de búsqueda por localización geográfica, evidencia que presenta un alto grado de resistencia ante posibles modificaciones.
3. El diseño de los diagramas de clases y de secuencia, facilita la visión de los componentes de software y su interacción.
4. La elaboración del diagrama de despliegue permitió identificar la disposición física de los componentes del subsistema de búsqueda por localización geográfica que se propone.

CAPÍTULO 3: “Implementación y prueba del subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a.”

En el presente capítulo se desarrolla la etapa de implementación y codificación del sistema, la cual genera los artefactos: diagrama de componentes, los estándares de codificación y casos de prueba. Con la finalidad de asegurar un correcto funcionamiento del sistema se realiza la validación de la solución propuesta mediante las estrategias de pruebas, lo cual garantiza la calidad y el cumplimiento de las especificaciones definidas en la etapa de diseño y análisis del software.

3.1. Diagrama de componentes

Un diagrama de componentes muestra los elementos del diseño de un sistema, permite visualizar la estructura de alto nivel, el comportamiento del servicio y representa la división lógica de las responsabilidades (Microsoft, 2015). En la Figura 11 se muestra el diagrama de componentes generado para la propuesta de solución.

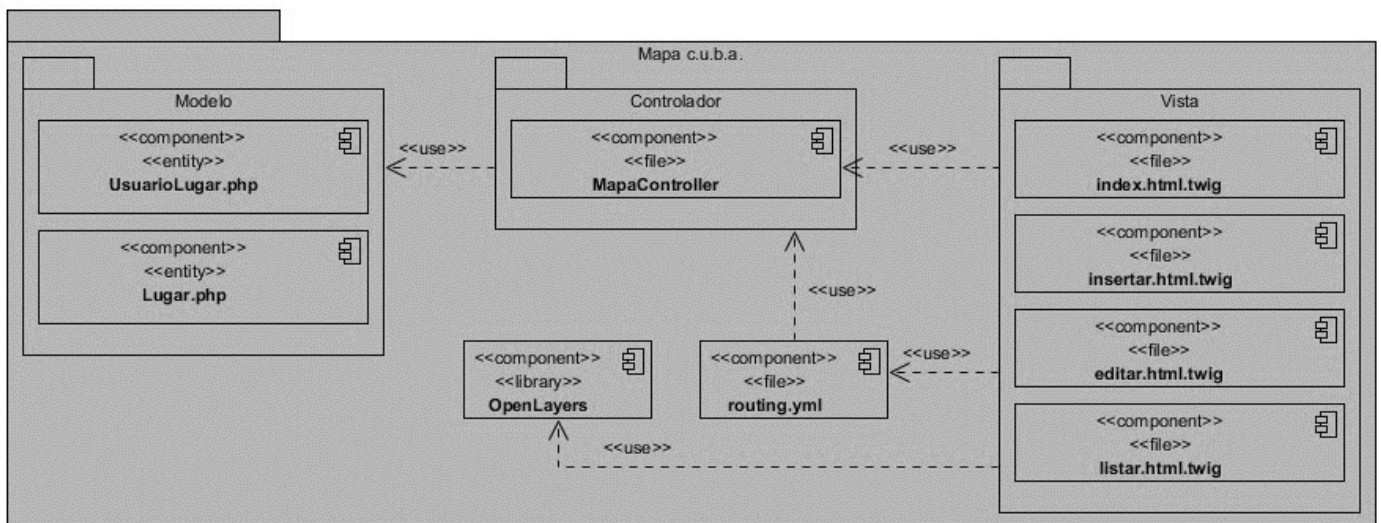


Figura 11: Diagrama de componentes

UsuarioLugar.php: Clase entidad que almacena la información espacial de los lugares insertados por los usuarios.

Lugar.php: Clase entidad que almacena la información espacial de los lugares existentes en el sistema.

MapaController.php: Clase controladora que contiene las acciones relacionadas con el sistema de información geográfica.

index.html.twig: Es la interfaz donde se muestra el mapa.

insert.html.twig: Muestra un formulario para insertar un lugar en el sistema.

edit.html.twig: Muestra un formulario para editar los lugares insertados en el sistema.

list.html.twig: Muestra un listado de los lugares insertados por el usuario y esperan ser aprobados por el administrador.

OpenLayers: Librería JavaScript para el manejo de los mapas.

Routing.yml: Contiene el mapa de rutas *URL* que se enlazan a las acciones de los controladores del módulo.

3.2. Estándares de codificación

Los estándares de codificación permiten la lectura rápida y simple del código de fuente. Garantizan su calidad, comprensión, mantenibilidad y gestión. Para el subsistema de búsqueda por localización geográfica desarrollado se utilizan los estándares de codificación para el lenguaje PHP con el marco de trabajo Symfony, los mismos se evidencian en la Tabla 5 (PHP-FIG, 2018).

Tabla 5: Estándares de codificación

Tipo de Estándar	Descripción
Estructura	<ol style="list-style-type: none">1. Se debe utilizar un espacio en blanco después de la palabra clave en las estructuras de control.2. No deben existir espacios en la apertura y cierre de paréntesis.3. Debe usarse un espacio en blanco entre los operadores lógicos, aritméticos, de comparación y asignación. <pre data-bbox="448 1560 1500 1728">if (\$request -> isXmlHttpRequest()) { \$criterio = \$request->request->get('criterio'); \$em = \$this->getDoctrine()->getEntityManager(); \$db = \$em->getConnection(); }</pre>

<p>Convenciones de nombres</p>	<p>4. Se debe usar el estilo de escritura <i>camelCase</i> para la declaración de variables y no guiones bajos para separar palabras.</p> <p>5. Se debe usar el estilo de escritura <i>StudlyCaps</i> para los nombres de clases.</p> <pre data-bbox="448 415 1511 890"> public function insertarlugarAction(Request \$request) { \$idUserio = \$this->getUser()->getId(); \$form = \$this->createFormBuilder() ->add('nombre', 'text') ->add('longitud', 'number') ->add('latitud', 'number') ->add('tipo', 'text') ->add('descripcion', 'textarea') ->add('Enviar', 'submit') ->getForm(); } </pre>
<p>Organización del código</p>	<p>6. La apertura de llaves para clases y funciones deben estar en la próxima línea después de su declaración.</p> <p>7. La apertura de llaves en las estructuras de control debe estar en la misma línea.</p> <p>8. El cierre de llaves para clases, funciones y estructuras de control deben estar en la próxima línea después del bloque de código.</p> <pre data-bbox="542 1142 1252 1539"> foreach (\$po as \$p) { \$longitud = \$p['longitud']; \$latitud = \$p['latitud']; \$aprobado = \$p['aprobado']; } if (!\$aprobado) { \$longitud = ""; \$latitud = ""; } } </pre>
<p>Líneas</p>	<p>9. Las líneas no deberían ser más largas de 80 caracteres; las líneas más largas de estos 80 caracteres deberían dividirse en múltiples líneas de no más de 80 caracteres cada una.</p> <p>10. Pueden añadirse líneas en blanco para mejorar la lectura del código y para indicar bloques de código que estén relacionados.</p> <p>11. No debe haber más de una sentencia por línea.</p>

	<pre>\$query = "SELECT name, ST_AsText(ST_CENTROID(geometry)) as pos, ST_AsText(geometry) as poligono FROM lugar WHERE name ILIKE '%\$criterio%' LIMIT 1"; \$stmt = \$db->prepare(\$query);</pre>
Codificación	<p>12. La etiqueta de cierre ?> debe ser omitida en los archivos que sólo contengan código PHP.</p> <p>13. Todos los archivos PHP deben terminar con una línea en blanco.</p>

3.3. Validación de la propuesta de solución

Para validar el sistema de información geográfica desarrollado se utilizan un conjunto de actividades en las que se incluyen las pruebas funcionales, de seguridad, integración, carga y estrés y se valida la hipótesis científica. Se comprueba con las mismas que el sistema cumpla correctamente con los requisitos funcionales definidos y con las necesidades del cliente.

3.3.1. Pruebas funcionales

Las pruebas funcionales son basadas en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. Se realizan mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones que presenta el sistema, específicas, concretas y exhaustivas. Para realizar las pruebas se utiliza la técnica de Caja Negra que permiten derivar conjuntos de condiciones de entradas que revisaran por completo todos los funcionamientos del sistema. Determinan las funciones incorrectas o ausentes, errores de interfaces, de estructuras de datos, rendimiento, inicialización y terminación (Pressman, 2010).

Tabla 6: Descripción de las variables para el Caso de prueba 1

N.º	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Buscar	Campo de texto	No	No permite caracteres extraños.

Tabla 7: Caso de prueba 1: SC RF3_Búsqueda de lugares por nombre de los lugares

Caso de prueba 1: SC RF3_Búsqueda de lugares por nombre de los lugares.				
Condiciones de ejecución: No requiere condiciones de ejecución para realizar la búsqueda.				
Escenario	Descripción	1	Respuesta del sistema	Flujo central
		V		

EC 1.1 Realizar búsqueda de lugares por nombre de forma correcta.	El sistema realiza la búsqueda de forma correcta.	Capitolio Nacional	El sistema muestra la ubicación exacta del Capitolio Nacional, lo señala con un puntero y enmarca la región con líneas rellenas de color azul.	1. El usuario introduce los datos en el formulario de búsqueda que se encuentra en la parte superior izquierda debajo de la barra de navegación. 2. Se presiona la tecla "enter" del teclado o se da clic al botón "Buscar" para obtener el resultado.
EC 1.2 Realizar búsqueda de lugares por nombre con valores no registrados.	El sistema no realiza la búsqueda con valores no registrados.	V Coliseo Romano	El sistema notifica al usuario "No se encontraron resultados para su búsqueda".	

Tabla 8: Descripción de las variables para el Caso de prueba 2

N.º	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Latitud	Campo numérico	No	Permite solo caracteres numéricos
2	Longitud	Campo numérico	No	Permite solo caracteres numéricos

Tabla 9: Caso de prueba 2: SC RF4_Búsqueda de lugares por coordenadas de los lugares

Caso de prueba 2: SC RF4_Búsqueda de lugares por coordenadas de los lugares.					
Condiciones de ejecución: No requiere condiciones de ejecución para realizar la búsqueda.					
Escenario	Descripción	1	2	Respuesta del sistema	Flujo central
EC 2.1 Realizar búsqueda de lugares por coordenadas de forma correcta.	El sistema realiza la búsqueda de coordenadas de forma correcta.	V 22.98908	V -82.46733	El sistema muestra la ubicación aproximada de las coordenadas introducidas.	1. El usuario presiona la opción "BUSCAR POR COORDENADAS", que se encuentra en la parte superior derecha de la barra de navegación. 2. El sistema muestra un formulario con los campos de latitud y longitud. 3. Se presiona la tecla "enter" del teclado o se da clic al botón "Buscar" para obtener el resultado.
EC 2.2 Realizar la búsqueda por coordenadas de forma incorrecta.	El sistema no realiza la búsqueda por coordenadas de forma incorrecta.	I aaaaeaaa	I *efwe##*	El sistema notifica al usuario "Se produjo un error, revise los datos de los campos".	

Resultados de las pruebas funcionales

En la primera iteración se generaron cuatro (4) no conformidades. Dos (2) de ellas son funcionales, relacionadas con la ausencia de la notificación al usuario cuando los lugares no están registrados en la base de datos o cuando los datos de los campos son incorrectos, para resolverlo se incluyen los mensajes antes mencionados al código de fuente. Una no conformidad de excepción que se produce en la búsqueda por coordenadas; al introducir incorrectamente los datos en los campos con caracteres extraños o letras, se muestra un error fatal que interrumpe el flujo normal del sistema, fue resuelta al validar que en los campos Longitud y Latitud el tipo de dato sea numérico. La última no conformidad identificada fue un error ortográfico, la misma fue corregida al instante. Se realizó una segunda iteración donde no se identificaron no conformidades. Luego de realizadas las pruebas funcionales se llega a la conclusión de que el sistema cumple satisfactoriamente con las necesidades del cliente y con los requisitos funcionales definidos, esto se evidencia en el gráfico de la Figura 12.

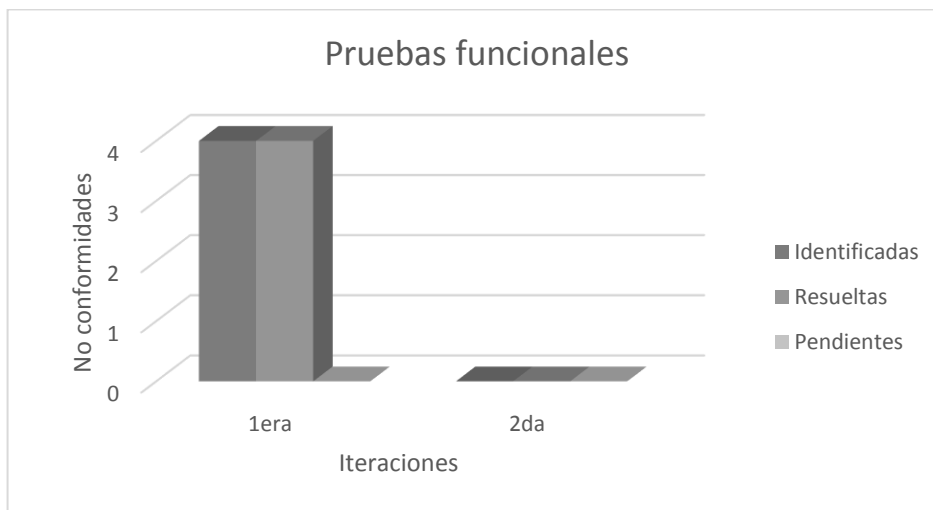


Figura 12: Resultados de las pruebas funcionales

3.3.2. Pruebas de seguridad

Las pruebas de seguridad se diseñan para identificar las vulnerabilidades que presenta el sistema. Cada uno de estos dominios pueden atacarse, ocasionando pérdidas para la entidad y es tarea del examinador de seguridad descubrir las debilidades que puedan explotar quienes tengan intención de hacerlo (Pressman, 2010).

Para realizar las pruebas de seguridad se utilizó la herramienta *Acunetix Web Vulnerability Scanner*. Mediante estas pruebas se detectaron cincuenta y ocho (58) alertas de ellas cuarenta y nueve (49) son de nivel medio y nueve (9) de carácter informativo. De las vulnerabilidades de nivel medio cuarenta y ocho (48) están relacionadas con los mensajes de error producidos en el modo de desarrollo del marco de trabajo Symfony y se detectó un formulario que no está protegido contra la falsificación de solicitudes entre sitios (CSFR por sus siglas en inglés). Las alertas de carácter informativo se dividen en ocho (8) enlaces rotos y una relacionada con el fichero *parameters.yml* que contiene los atributos para la conexión con el sistema gestor de base de datos PostgreSQL. La alerta del formulario se soluciona manualmente como recomienda la herramienta *Acunetix Web Vulnerability*, denominándose un falso positivo y el resto se resuelve cuando se pasa el sistema al modo producción. Luego de ser solucionadas las alertas de vulnerabilidad encontradas se obtiene un sistema seguro que cumple con los requisitos de seguridad avalados por la Dirección de Calidad de la UCI.

3.3.3. Pruebas de integración

Las pruebas de integración se realizan con el objetivo de asegurar que todos los componentes del sistema se integren entre sí de forma correcta. Estas pruebas implican construir la aplicación desde sus componentes independientes y probar el sistema resultante en busca de errores que puedan surgir con el proceso de integración (Sommerville, 2011). El desarrollo de la prueba demostró la correcta integración del sistema con componentes evaluados, demostrándose en la Tabla 10

Tabla 10: Pruebas de integración

Sistema	Funcionalidades	Funcionalidad integrada	Resultado de prueba
Servidor de mapa MapServer	La integración con el servidor de mapa MapServer permite la creación de los mapas a partir de los datos obtenidos con el PostGIS.	La funcionalidad de integración se evidencia en el sistema al mostrar el mapa.	El sistema se integra satisfactoriamente con el MapServer al mostrar un mapa en la interfaz para la localización geográfica.
PostGIS	La integración del PostGIS con el subsistema de	La funcionalidad de integración se evidencia	El sistema se integra satisfactoriamente con

	búsqueda por localización geográfica permite gestionar los datos espaciales de los lugares.	al localizar un lugar en el mapa.	el PostGIS al mostrar la ubicación exacta de un lugar en el mapa dado un criterio de búsqueda.
--	---	-----------------------------------	--

3.3.4. Pruebas de carga y estrés

Para determinar y validar la respuesta del sistema cuando es sometida a una concurrencia de usuarios o transacciones que se espera en el ambiente de producción se realizan las pruebas de carga y estrés o también conocidas como pruebas de rendimiento. Con ellas se determina el volumen de datos o de tiempo en que el sistema comienza a fallar o es incapaz de responder a las peticiones (Sommerville, 2011).

Para la realización de las pruebas de carga y estrés se utilizó la herramienta Apache Jmeter. Las pruebas se realizaron desde un ordenador con 8GB de RAM, microprocesador Intel Core i7 con 2.50 GHz y sistema operativo Ubuntu 14.04. A continuación, se describen las variables que miden el resultado de las pruebas de carga y estrés realizadas al sistema.

Muestra: Cantidad de peticiones realizadas para cada URL.

Media: Tiempo promedio en milisegundos en el que se obtienen los resultados.

Mediana: Tiempo en milisegundos en el que se obtuvo el resultado que ocupa la posición central.

Min.: Tiempo mínimo que demora un hilo en acceder a una página.

Max.: Tiempo máximo que demora un hilo en acceder a una página.

Línea 90 %: Máximo tiempo utilizado por el 90% de la muestra, al resto de la misma le llevo más tiempo.

% Error: Por ciento de error de las páginas que no se llegaron a cargar de manera satisfactoria.

Rendimiento (Rend): El rendimiento se mide en cantidad de solicitudes por segundo.

KB/s: El rendimiento se mide en cantidad de kilobytes por segundo.

Los resultados de las pruebas de rendimiento se muestran en la Tabla 11, se simularon las peticiones al subsistema de búsqueda por localización geográfica por un total de 1000, 1500 y 2000 usuarios.

Tabla 11: Pruebas de carga y estrés

Usuarios	Media	Mediana	Línea 90%	Min	Max	% Error	Rend	KB/s
1000	20938	14606	55572	71	61645	0.0	12,7	77.9

1500	23989	21696	41872	37	84965	0.0	13,04	79.8
2000	55387	42810	119890	109	124345	0.27	15,9	80.5

Las pruebas realizadas muestran que el subsistema es capaz de responder a 20938 peticiones de 1000 usuarios conectados simultáneamente en un tiempo promedio de 14606 milisegundos con 0% de error. Esto evidencia que el sistema puede procesar la carga esperada, cumpliéndose de este modo el requisito no funcional 10.

Luego se realizaron 20938 peticiones iniciadas por 1500 usuarios y el subsistema respondió en un tiempo promedio de 21696 milisegundos con 0% de error. Con el objetivo de analizar el comportamiento del subsistema en altas condiciones, se realizó una prueba de estrés para un conjunto de 2000 usuarios conectados simultáneamente. En este caso, el sistema responde satisfactoriamente, siendo capaz de responder a 55387 peticiones con un porcentaje de error del 0,27%. Se demuestra que el subsistema es capaz de responder satisfactoriamente a las peticiones realizadas, con un tiempo y porcentaje de error adecuados.

3.3.5. Validación de la hipótesis científica

Para evaluar la confiabilidad de la hipótesis científica de la investigación se aplica el Criterio de expertos mediante el método Delphi. Este método consiste en una técnica de obtención de información, basada en la consulta de expertos en un área de estudio, con el fin de obtener la opinión de consenso más fiable proveniente del conocimiento y la experiencia de los participantes del grupo (Reguant, et al., 2016)

Variable independiente: El subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a., consiste en un servicio que permita encontrar direcciones e información geográfica con precisión y exhaustividad.

Variable dependiente: Mejorar el proceso de recuperación de información geográfica, refiriéndose a la optimización del tiempo, esfuerzo, precisión y exhaustividad en los resultados obtenidos.

Para aplicar el método Delphi se emplearán los pasos siguientes:

- Identificación y selección de posibles expertos sobre el tema.
- Aplicación de encuestas a los expertos identificados.
- Valoración de la información obtenida.

Identificación y selección de los posibles expertos sobre el tema.

Se identificaron siete (7) posibles expertos sobre el tema de los sistemas de información geográfica. Para evaluar el nivel de experiencia y conocimiento del tema se aplicó una encuesta. En la Tabla 12 se presentan los resultados obtenidos.

Tabla 12: Resultado de niveles de conocimiento de posibles expertos

Expertos	Nivel de conocimiento										Kc
	1	2	3	4	5	6	7	8	9	10	
Alejandro Orogelio										x	1
Ernesto C González									x		0,9
Yosiel Mederos									x		0,9
Leiny Ruiz Cervera									x		0,9
Fidelkys Alcantara									x		0,9
Laritz Azan Caballero									x		0,9
Ariel Labrada Delgado											0,9

Para calcular el coeficiente de conocimiento o información (Kc) se utiliza la siguiente ecuación:

$$Kc = n(0,1)$$

Donde n es el conocimiento o información determinado por el experto.

Para calcular el coeficiente de argumentación o fundamentación de cada experto es necesario utilizar como factores, los que aparecen a continuación en la Tabla 13.

Tabla 13: Patrón para el cálculo de Coeficiente de argumentación o fundamentación

Fuentes de argumentación	Alto	Medio	Bajo
Análisis teórico realizado por usted.	0.3	0.2	0.1
Su experiencia obtenida.	0.5	0.4	0.2
Trabajos de autores nacionales.	0.05	0.05	0.05
Trabajos de autores extranjeros.	0.05	0.05	0.05
Su conocimiento del estado del problema en el extranjero.	0.05	0.05	0.05
Su intuición.	0.05	0.05	0.05

Luego del análisis antes realizado se calcula el coeficiente de argumentación (Ka) con la ecuación:

$$Ka = \sum_{i=1}^n n_i$$

Donde n_i es el valor correspondiente a la fuente de argumentación i (1 hasta 6). Luego de aplicada la ecuación se obtuvieron los siguientes resultados:

Alejandro Orogelio: $Ka = 0.9$

Ernesto C González: $Ka = 0.7$

Yosiel Mederos: $Ka = 0.7$

Leiny Ruiz Cervera: $Ka = 0.8$

Fidelkys Alcantara: $Ka = 0.9$

Laritz Azan Caballero: $Ka = 0.8$

Ariel Labrada Delgado: $Ka = 0.9$

Una vez obtenido los valores del Coeficiente de Conocimiento (Kc) y el Coeficiente de Argumentación (Ka) se procede a obtener el valor del Coeficiente de Competencia (K), coeficiente que determina en realidad que experto se toma en consideración para trabajar en la hipótesis científica de la presente investigación. Este coeficiente (K) se calcula de la siguiente forma:

$$K = 0,5(Kc + Ka)$$

Obtenido los resultados se valoran de la siguiente manera:

- $0,8 < K < 1,0$ K es Alto.
- $0,5 < K < 0,8$ K es Medio.
- $K < 0,5$ K es Bajo.

Tabla 14: Resultado de niveles de competencia de posibles expertos

Expertos	K	Valoración
Alejandro Orogelio	0,95	Alto
Ernesto C González	0,8	Alto
Yosiel Mederos	0,8	Alto
Leiny Ruiz Cervera	0,85	Alto
Fidelkys Alcantara	0,9	Alto

Laritzza Azan Caballero	0,85	Alto
Ariel Labrada Delgado	0,9	Alto

Se seleccionaron los expertos con un nivel de competencia Alto, como se evidencia en la Tabla 15.

Tabla 15: Expertos seleccionados para la validación de la hipótesis científica

Nombre y Apellidos	Entidad	Años de experiencia
Alejandro Orogelio	Centro de Geoinformática y Señales Digitales.	8
Ernesto C González	Centro de Geoinformática y Señales Digitales.	5
Yosiel Mederos	Centro de Geoinformática y Señales Digitales.	3
Leiny Ruiz Cervera	Centro de Geoinformática y Señales Digitales.	5
Fidelkys Alcantara	Centro de Geoinformática y Señales Digitales.	4
Laritzza Azan Caballero	Centro de Geoinformática y Señales Digitales	4
Ariel Labrada Delgado	Centro de Geoinformática y Señales Digitales	4

Aplicación de encuestas a expertos:

Con el objetivo de validar el subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a. se exhorta a los expertos seleccionados que trabajarán en la validación de la hipótesis científica a realizar una encuesta. Los resultados se evidencian en la Tabla 16. según las sentencias que se muestran a continuación:

1. Facilidad de la navegación con el mapa.
2. Optimización del tiempo en las búsquedas por nombre de lugares y coordenadas.
3. Precisión de las localizaciones de los lugares deseados.
4. Facilidad y seguridad en la gestión de los lugares.

Para determinar la categoría o grado de adecuación de cada pregunta según la opinión de los expertos se clasifican en: muy adecuado (MA), adecuado (A), poco adecuado (PA) e inadecuado (I), los resultados de las categorías se evidencian en la Tabla 16.

Tabla 16: Resultado de las encuestas realizadas a los expertos

Sentencias	Categorías evaluativas				Total
	MA	A	PA	I	
1	6	0	0	0	6
2	0	6	0	0	6

3	5	1	0	0	6
4	4	1	1	0	6

Una vez plasmados los criterios de los expertos, se realizan los siguientes pasos establecidos hasta llegar a concluir que valoración cumple con cada una de las preguntas.

- Obtención de la tabla de frecuencia acumulada.
- Obtención de la tabla de frecuencia relativa acumulativa.
- Asignación del valor de la imagen que corresponde a cada frecuencia relativa acumulativa, por la inversa de la curva normal (Tabla Z de la distribución normal) y obtención de puntos mediante el cálculo N-P para el promedio relativo.

Tabla 17: Frecuencia acumulada de los datos primarios obtenidos

Sentencias	Categorías evaluativas			
	MA	A	PA	I
1	7	7	7	7
2	7	7	7	7
3	6	7	7	7
4	5	6	7	7

Tabla 18: Frecuencia relativa acumulativa de los datos primarios obtenidos

Sentencias	Categorías evaluativas		
	MA	A	PA
1	1	1	1
2	1	1	1
3	0,85	1	1
4	0,71	0,85	1

Tabla 19: Imagen de la frecuencia relativa acumulativa de los datos primarios obtenidos

Sentencias	Categorías evaluativas			Suma	Promedio	Promedio relativo
	MA	A	PA			
1	3,49	3,49	3,49	10,47	3,49	-1,36
2	3,49	3,49	3,49	10,47	3,49	-1,36
3	0,80	3,49	3,49	7,98	2,66	-0,53
4	0,76	0,80	3,49	5,05	1,68	0,44
Puntos de Corte	2,13	2,81	3,49	33,97		

Analizado los promedios relativos (N-P) de las sentencias, se evidencia que ninguno sobrepasa el límite superior o punto de corte de la categoría MA. Los expertos con un 100 % de concordancia evalúan con el más alto nivel de adecuación MA las sentencias apoyando la hipótesis de la presente investigación. Se demostró la calidad y el alto valor que presenta el subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a.

3.4. Resumen del capítulo

El análisis de los elementos de la implementación del subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a., las pruebas de software realizadas y los resultados obtenidos de las encuestas al panel de expertos, permitieron realizar los siguientes resúmenes:

1. La elaboración del diagrama de componentes, permite una mejor comprensión de la estructura de los componentes del subsistema de búsqueda por localización geográfica implementado.
2. Utilizar estándares de codificación permitió lograr una mayor legibilidad y mantenibilidad del código creado.
3. La validación de la propuesta de solución mediante la estrategia de pruebas definida, dio como resultado un subsistema que responde correctamente a los requisitos definidos por el cliente, integrable con los componentes que utiliza y cumple con los estándares de seguridad de la Dirección de Calidad de la UCI.
4. La validación de la hipótesis de la presente investigación demostró con un alto nivel de adecuación en las sentencias evaluadas por los expertos, que el subsistema de búsqueda por localización geográfica mejora la recopilación de información geográfica en la plataforma de servicios c.u.b.a.

CONCLUSIONES

1. El análisis y estudio de los diferentes sistemas de recuperación de información geográfica permitió identificar las funcionalidades necesarias para la correcta localización geográfica de los lugares.
2. La selección de las bibliotecas OpenLayers y PostGIS facilitaron el trabajo con los mapas y la exacta localización geográfica de los lugares.
3. El correcto uso de las herramientas y tecnologías seleccionadas, permitió la implementación del subsistema de búsqueda por localización geográfica para la plataforma de servicios c.u.b.a.
4. La estrategia de pruebas de software definida, demostró un sistema funcional, integrable con los componentes que utiliza y cumple con los requisitos de seguridad establecidos por la Dirección de Calidad de la UCI.
5. La aplicación del método Delphi para validar la hipótesis científica demostró con un alto nivel de adecuación que el subsistema de búsqueda por localización geográfica mejora la recopilación de información geográfica en la plataforma de servicios c.u.b.a.

RECOMENDACIONES

Para el desarrollo de futuras investigaciones relacionadas con la presente, se propone:

1. Adicionar una funcionalidad que le proporcione al usuario la información del lugar localizado.
2. Adicionar una funcionalidad que le posibilite al usuario trazar la ruta de cómo llegar a los lugares deseados.

BIBLIOGRAFÍA

ALCÁNTARA, Jorge y MARTÍNEZ Ramón. El uso de las TIG en la Educación. Realización de un inventario de flora. Tesis (Ciencias Sociales y Experimentales). Universidad de Córdoba, 2014. 77pp

ÁLVAREZ, Sara. Conceptos básicos y definiciones sobre programación [En línea]. 2016. [Fecha de consulta: 14 noviembre 2017]. Disponible en: <https://desarrolloweb.com/articulos/2357.php>.

APACHE. What is the Apache HTTP Server Project? [En línea] Apache Software Foundation 2017. [Fecha de consulta: febrero 2018] Disponible en: https://httpd.apache.org/ABOUT_APACHE.html.

ASTUDILLO, Niama y FERNANDO, Pablo. Sistema de gestión de historias clínicas para la clínica panamericana – CEDITEM utilizando el framework symfony2. Tesis (Tesis Ingeniero en Sistemas Informáticos). Riobamba: Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería en Sistemas, 2015. 119 pp.

AVILA DEL CAMPO, Arturo. Diseño de un SIG para la ubicación óptima de una instalación minera. Tesis (Máster en teledetección y sistemas de información geográficas) Oviedo: Universidad de Oviedo, Escuela Politécnica de Mieres, 2014. 52 pp

AYALA, Evelyn Elizabeth y GONZÁLEZ, Santiago Raúl. Tecnologías de la Información y la Comunicación, Lima-Perú, Depósito Legal en la Biblioteca Nacional del Perú N° 2015-11001

AYOZE, Alberto. Curso de Programación Web: JavaScript, Ajax y jQuery. 2ª Edición, Plaza América, Vigo, España, IT Campus Academy: Libro, 2017, ISBN-13: 978-1542787406.

BATTAGLIA, Nicolás NEIL, Carlos, DE VINCENZI, Marcelo y MARTÍNEZ, Roxana. UAICase: Integración de un Entorno Académico con una Herramienta CASE en una Plataforma Virtual Colaborativa, San Luis-Argentina, Facultad de Tecnología Informática, Universidad Abierta Interamericana, 2016, ISBN: 978-987-3977-30-5.

BERREIROS, Osmar Rafael. Método de posicionamiento web basado en responsive web design y herramientas SEO. Tesis (Licenciatura en Informática). La Paz-Bolivia, Universidad de Mayor San Andrés, 2015, 98 pp.

BING MAPS. Bing Maps [En línea] Downloadastro 2017. [Fecha de consulta: 2 de noviembre de 2017.] Disponible en: http://bing_maps.es.downloadastro.com/.

BOOTSTRAP. Bootstrap [En línea] GitHub, Inc. 2016. [Fecha de consulta: febrero de 2018] Disponible en: <http://getbootstrap.com/docs/3.3/>.

CALDERÍN, Jorge Luis, MIRANDA, Giosvany, ÁLVAREZ, Yasiel, QUINTANA, José Sebastian, ALONSO, Renato y CAPOTE, Alejandro. Andariego, sistema de información geográfica para dispositivos móviles como alternativa para acercar la información de los servicios al ciudadano, Artemisa, Empresa Cartografía y Soluciones Geomáticas, 2016.

CÁRDENAS, Lain. Aplicación de Patrones de Diseño para Garantizar Alta Flexibilidad en el Software. Universidad César Vallejo, 7(1), 2016. ISSN: 1819-4575.

CARTARO. Cartaro [En línea] geOps 2015. [Fecha de consulta: 12 de abril de 2018] Disponible en <http://cartaro.org/>.

CEVALLOS, Karla. UML: Diagrama de Secuencia. Calceta [En línea] Calceta. 2015 [Fecha de consulta: marzo de 2018]. Disponible en: <https://es.scribd.com/document/352646876/Diagramas-de-Secuencias>.

CONOCE CUBA. Conoce Cuba. [En línea] CYRBES 2018. [Fecha de consulta: 27 de marzo de 2018.] Disponible en: <https://conocecuba.com/>.

DEEGREE. deegree. [En línea] deegree.org 2018. [Fecha de consulta: 12 de abril de 2018] Disponible en: <https://www.deegree.org/about-deegree#project-organisation>.

DÍAZ, Cando y RAMÓN, Diego. Análisis, diseño, desarrollo e implementación de un sistema de información geográfica para la visualización de servicios para excelencia corporativa. Tesis (Tesis de Ingeniero en Sistemas e Informática). Sangolqui, Universidad de las Fuerzas Armadas, 2015. 181 pp.

FERNÁNDEZ, Roannel y FERREIRO Vázquez, Luis Reinier. Subsistema de búsqueda de videos para el buscador cubano Orión. Tesis (Tesis de postgrado). La Habana, Universidad de las Ciencias Informáticas, 2015. 80pp.

GABILLAUD, Jérôme. Administracion de una base de datos transaccional con SQL Server Managment Studio. Barcelona, ENI, 2015. ISBN: 08940.

GARCIA, Francisco José, MORENO, María N. y GARCÍA, Alicia. Ingeniería de software I, Tema 8 – UML. Unified Modeling Language. Tesis (Tesis de Ingeniero en Sistemas). Plaza América, Vigo, España, Departamento de Informática y Automática, Universidad de Salamanca, 2018.

GEYSED. ¿Qué es ULTRON?, La Habana, Cuba, Universidad de las Ciencias Informáticas, 2017.

GOOGLE MAPS. Google Maps. [En línea] Google Developers 2017 [Fecha de consulta: 2 de noviembre de 2017] Disponible en: <https://developers.google.com/maps/support/availability?hl=es-419>.

GUERRERO, Carlos; LONDOÑO, Jorge; SUÁREZ y Johanna; GUTIÉRREZ, Luz E. Estudio comparativo de marcos de trabajo para el desarrollo software orientado a aspectos [En línea] Inf. Tecnol 2014 [Fecha de consulta: febrero de 2018] Disponible en: https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-07642014000200008

GUEVARA, Yurisander. Informatización en Cuba, ¿por dónde andamos? [En línea] Juventud Rebelde, Cuba. 26 de julio de 2017. [Fecha de consulta: 22 de septiembre de 2017] Disponible en: <http://www.juventudrebelde.cu/suplementos/informatica/2017-07-26/informatizacion-en-cuba-por-donde-andamos>.

HAMON, Hugo. Applying Design Patterns to Symfony. Estambul, Turquía: SensioLabs, 2014.

HERRERA, Nathaly Soledad. Construcción de una aplicación web para la Gestión de Citas, Tratamiento y Expedientes Médico de los pacientes del Centro Implantológico y de Especialidades Dental Access. Tesis (Tesis de Ingeniero en Sistemas). Quito, Pontificia Universidad Católica del Ecuador, Facultad De Ingeniería, Escuela De Sistemas, 2016. 206 pp.

NEGI. Sistemas de Información Geográfica. [En línea] Instituto Nacional de Estadísticas y Geografía, 2017. [Fecha de consulta: septiembre de 2017] Disponible en: <http://gaia.inegi.org.mx/NLB/mdm5.wms>.

JQUERY. What is jQuery? [En línea] The jQuery Foundation, 2017. [Fecha de consulta: abril de 2018] Disponible en: <https://jquery.com/>.

LEAFLET. Leaflet. [En línea] OpenStreetMap, 2017. [Fecha de consulta: abril de 2018] Disponible en: <http://leafletjs.com/index.html>.

LIBROSWEB. Librosweb [En línea] LibrosWeb.es 2017. [Fecha de consulta: marzo de 2017] Diposnible en: <http://librosweb.es/libro/javascript/>.

LLENERA, Zerpa y GIOVANNA, Karen. Uso De Las Tecnologías de Información y Comunicación y los Contenidos del Aprendizaje. Tesis (Tesis de Maestría en Investigación y Docencia Universitaria) Bellavista, Perú, Universidad Inca Garcilaso de la Vega Nuevos Tiempos, Nuevas Ideas, 2016. 109 pp.

MAPSERVER. MapServer. [En línea] 2017 [Fecha de consulta: 15 de noviembre de 2017] Disponible en: <http://mapserver.org/>.

MICHELINI, Juan J y GARCÍA, Juan Carlos. Tecnologías de la información geográficas de promoción de territorios innovadores: aplicaciones en la ciudad de Madrid. [En línea] Madrid Huellas, 2008 [Fecha de consulta: 15 de mayo de 2018] ISSN 0329-0573. Disponible en: <http://www.biblioteca.unlpam.edu.ar/pubpdf/huellas/n12a09michelini.pdf>

MICROSOFT. Diagramas de componentes de UML. [En línea] Microsoft 2015. [Fecha de consulta: 28 de marzo de 2018.] Disponible en: <https://msdn.microsoft.com/es-es/library/dd409390.aspx>.

NetBeans IDE. NetBeans IDE. [En línea] NetBeans IDE, 2016. [Fecha de consulta: diciembre de 2017] Disponible en: <https://netbeans-ide.uptodown.com/windows>.

OPENLAYERS. OpenLayers. [En línea] 2017. [Fecha de consulta: diciembre de 2017] Disponible en: <https://openlayers.org/>.

ORELLANA, Miguel. Localización geográfica 2 relativa y absoluta. [En línea] Chile, Colegio Florencia Nightingale, La Serena, 2015. [Fecha de consulta: 25 de septiembre de 2017] Disponible en: <https://es.slideshare.net/MiguelOrellanaGuerra/localizacin-geografica-2-relativa-y-absoluta>

PHP. ¿Qué es PHP? [En línea] 2017 [Fecha de consulta: 7 de noviembre de 2017"] Disponible en: <http://php.net/manual/es/intro-what-is.php>.

PHP-FIG. PHP Framework Interop Group. [En línea] PHP Framework Interop Group.2018. [Fecha de consulta: 7 de noviembre de 2017]. Disponible en: <https://www.php-fig.org/psr/>

PostGIS. PostGIS. [En línea] 2017 [Fecha de consulta: 7 de noviembre de 2017.]. Disponible en: <http://postgis.net/>.

PostgreSQL. PostgreSQL. [En línea] The PostgreSQL Global Development Group, 2017 [Fecha de consulta: 7 de noviembre de 2017.] Disponible en: <https://www.postgresql.org/about/>.

PRESSMAN, Roger. Ingeniería del Software. Un enfoque práctico. 7ª ed. México, D.F: The MacGraw-Hill Companies Inc., 2010. ISBN: 978-607-15-0314-5.

REGUANT Álvarez, Mercedes y TORRADO Fonseca, Mercedes. El método Delphi. Revista d'Innovació i Recerca en Educació, 9(1), 2016. ISSN:2013-2255.

SÁNCHEZ Rodríguez, Tamara. Metodología de desarrollo para la Actividad productiva de la UCI v1.2. La Habana: Universidad de las Ciencias Informáticas, 6 de marzo de 2015.

SÁNCHEZ, Iván y TOMALÁ, Raúl David. Desarrollo de un sistema de ventas de artesanías de los grupos de interés prioritario afines al FENEDIF, utilizando referencias en sistemas de reconocimiento de voz. Tesis (Tesis de Ingeniería en Sistemas) LA Libertad, Ecuador: Universidad Estatal Península De Santa Elena, 2017.

SARMIENTO, Julio. UML: Diagrama de despliegue. Obtenido de Visión general del diagrama de despliegue [Artículo de blog]. [Fecha de consulta: febrero de 2016] Disponible en: <http://umldiagramadespliegue.blogspot.com/>.

SEI. Mejora de los procesos para el desarrollo de mejores productos y servicios. Software Engineering Institute. Technical Report, 2010.

SOMMERVILLE, Ian. Ingeniería del software. 7ª ed. Madrid: Pearson Educación S.A, 2011. ISBN: 84-7829-074-5.

SYMFONY. What is Symfony. Symfony para programadores [En línea]. Follow Symfony 2017. [Fecha de consulta: 8 de noviembre de 2017]. Disponible en: <http://symfony.com/what-is-symfony>.

TECHOPEDIA. Modeling Language [En línea]. Techopedia Inc 2017. Disponible en: <https://www.techopedia.com/definition/20810/modeling-language>

TORTAJADA, Juan Jesús. La Guía Definitiva del Diseño Web: Html, Xhtml, Css y herramientas de Diseño. Santander y alrededores, España, Universidad Europea del Atlántico, 2014

VALDÉS, Hansel Lázaro, ALEMAN Jiménez, Yulio y GARCÍA Rivas, Dairelys. Evaluación de la prensa digital cubana respecto a la optimización para motores de búsqueda. La Habana, Cuba, Serie Científica de la Universidad de las Ciencias Informáticas, 2016. Vol. Vol. 9, 5. 2306-2495.

VISUAL PARADIGM. Visual Paradigm for UML. [En línea] Visual Paradigm, 2017. Disponible en: <http://www.visual-paradigm.com/product/vpuml/>.

ANEXOS

Anexo 1: Encuesta a expertos

Usted ha sido seleccionado como miembro del panel de expertos, por ello, se necesita de su cooperación para validar la hipótesis: El desarrollo de un sistema de información geográfica para el buscador web de la plataforma de servicios c.u.b.a facilitará la ubicación de las diferentes entidades del país. A continuación, se muestran las sentencias a responder según su criterio.

Respecto al SIG para el buscador web de la plataforma de servicios c.u.b.a, diga su valoración de acuerdo a los siguientes parámetros: **MA**, si valora la sentencia como Muy Adecuado; **A**, si la valora como Adecuado; **PA**, si considera que es Poco Adecuado; o **I** si considera que es Inadecuado. Marcar con una (X):

1. Facilidad de la navegación con el mapa.
2. Optimización del tiempo en las búsquedas por nombre de lugares y coordenadas.
3. Precisión de las localizaciones de los lugares deseados.
4. Facilidad y seguridad en la gestión de los lugares.

Tabla 20: Encuesta a expertos

N.º	Sentencias	Categorías Evaluativas			
		MA	A	PA	I
1	Facilidad de la navegación con el mapa.				
2	Optimización del tiempo en las búsquedas por nombre de lugares y coordenadas.				
3	Precisión de las localizaciones de los lugares deseados.				
4	Facilidad y seguridad en la gestión de los lugares.				
5	Información de los lugares.				

Anexo 2: Casos de prueba

Tabla 21: Descripción de las variables para el Caso de prueba 3

N.º	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Nombre del lugar	Campo de texto	No	No permite caracteres extraños.
2	Tipo de lugar	Campo de texto	No	No Permite caracteres extraños
3	Descripción del lugar	Campo de texto	No	Permite caracteres extraños
4	Longitud	Campo numérico	No	No permite letras y caracteres extraños
5	Latitud	Campo numérico	No	No permite letras y caracteres extraños

Tabla 22: Caso de prueba 3: SC RF6_Insertar Lugar

Caso de prueba 3: SC RF6_Insertar Lugar								
Condiciones de ejecución: No requiere condiciones de ejecución para realizar la búsqueda.								
Escenario	Descripción	1	2	3	4	5	Respuesta del sistema	Flujo central
EC 3.1	El usuario inserta un lugar de forma correcta.	V	V	V	V	V	El sistema muestra la lista de los lugares del usuario, con el estado "sin Aprobar". El administrador debe aprobar el lugar.	1. El usuario introduce los datos en el formulario de insertar lugar que se encuentra en la parte superior derecha debajo de la barra de navegación con un icono de "usuario".
Insertar un lugar de forma correcta.	El usuario inserta un lugar de forma correcta.	Mi Casa	Privado	Esta es mi casa	-82.0000	23.0379		
EC 3.2	El usuario inserta un lugar de forma incorrecta.	V	V	V	V	V	El sistema muestra la lista de los lugares del usuario, con el estado "sin Aprobar". El administrador no debe aprobar el lugar.	2. Se presiona la tecla "enter" del teclado o se da clic al botón "Aceptar" para obtener el resultado.
Insertar un lugar de forma incorrecta.	El usuario inserta un lugar de forma incorrecta.	#\$%&/()	#\$%&	Casa	sadasfe	argga		