



UNIVERSIDAD DE CUENCA

Facultad de Ingeniería

Carrera de Ingeniería de Sistemas

Propuesta de una Arquitectura de Software para una plataforma de datos sobre movilidad activa y entorno urbano

Trabajo de Titulación previo a la obtención del
Título de Ingeniero de Sistemas

Autores:

Víctor David Guillermo Flores

CI: 0105867071

Correo electrónico: david.guillermofg@gmail.com

Christian Manuel Quinde Tenemaza

CI: 0105783187

Correo electrónico: christian.quindet@gmail.com

Tutora:

Ing. Paola Gabriela Pesántez Cabrera Ph.D.

CI: 0103232369

Co-tutora:

Ing. Lorena Catalina Sigüenza Guzmán Ph.D.

CI: 0102659687

Cuenca-Ecuador

02-Marzo-2020

Resumen:

Tras el transcurso de los años y con el continuo avance de la tecnología, la información geográfica y geoespacial ha pasado de estar únicamente en papeles en bibliotecas a ser gestionada por computadoras por medio de aplicaciones o sistemas de información geográfica (SIG). Estos sistemas han evolucionado de aplicaciones monolíticas a tener que interactuar con otros sistemas, por lo que ha sido necesario y primordial que éstas cumplan con características como escalabilidad, interoperabilidad, tolerancia a fallos, usabilidad, etc.

LlactaLAB, al ser una entidad dedicada a la movilidad activa y entorno urbano, está relacionada directamente con información geográfica y geoespacial, cuya gestión es compleja por la abundancia y diversidad de medios. Esto resulta en un problema para la entidad en cuanto al manejo de datos, así como la disponibilidad de los mismos.

Para facilitar la gestión de datos y sus procesos, éstos deben ser automatizados de alguna manera y además ser puestos a disposición de los usuarios. Por lo tanto, el objetivo de este estudio es proponer un diseño de Arquitectura de Software que intente solucionar los problemas descritos anteriormente, mediante la utilización de tecnologías y estándares robustos sobre la web. Para este propósito, en el diseño de la propuesta se utilizó la metodología ágil Scrum con una *Iteración 0*, lo que beneficiará el desarrollo en las siguientes *Iteraciones*; así mismo, se utilizó el método de Diseño Dirigido por Atributos ADD para el diseño, cuyo patrón arquitectónico de TI utilizado es la Arquitectura Orientada a Servicios o SOA, facilitando la escalabilidad e interoperabilidad del sistema. Finalmente, se propone el Método de Análisis de Equilibrio de la Arquitectura (ATAM) para la evaluación de la misma y así, dar cumplimiento a las necesidades consideradas inicialmente.

Palabras claves: Arquitectura de Software. Geoprocesamiento. Movilidad. SIG.

Abstract:

Over the years, with the continuous advancement of technology, geographic and geospatial information has gone from being in paper in libraries to being managed by computers through applications or geographic information systems (GIS), which have evolved from monolithic applications to have to interact with other systems, so it has been necessary and essential that these applications comply with features such as: scalability, interoperability, fault tolerance, usability, etc.

LlactaLAB being an entity dedicated to active mobility and urban environment is directly related to geographic and geospatial information, whose management is complex due to the abundance and diversity of media, which is a problem for the entity in terms of data management, as well as the availability of them.

To facilitate the management of data and its processes, they must be automated in some way and made available to users. The objective of this study is to propose a Software Architecture design that attempts to solve the problems described above, through the use of robust technologies and standards on the web; in the proposal design, the agile Scrum methodology is used with a *Sprint 0*, which will benefit the development in the following *Sprints*, as well as the Attribute-driven design or ADD method for the design, whose architectural pattern used is SOA (Service Oriented Architecture), facilitating scalability and interoperability of the system, finally the Architecture Trade-off Analysis Method (ATAM) is proposed for the evaluation of the architecture to comply with the needs initially considered.

Keywords: Software Architecture. Geoprocessing. Mobility. GIS.



Índice

Capítulo I	14
1. Introducción	14
1.1 Motivación y contexto	15
1.2 Planteamiento del problema	16
1.3 Justificación	17
1.4 Solución Propuesta	18
1.5 Objetivos generales y específicos	19
1.5.1 Objetivo General.....	19
1.5.2 Objetivos específicos.....	19
1.6 Metodología de investigación	19
Capítulo II	23
2. Marco Teórico	23
2.1 Arquitectura de Software	23
2.1.1 Arquitectura Orientada a Servicios.....	26
2.1.2 Software Ágil.....	28
2.1.3 Scrum.....	29
2.1.4 Proceso del modelado de una arquitectura de software.....	30
2.1.5 Requerimientos arquitectónicos.....	31
2.1.6 Diseño de arquitecturas de software.....	32
2.1.7 Métodos de diseño arquitectónico.....	33
2.1.8 Diseño dirigido por atributos (ADD).....	34
2.1.9 Evaluación de arquitecturas de software.....	35
2.1.10 Método de análisis de equilibrios de la arquitectura (ATAM).....	36
2.2 Movilidad Urbana	38
2.2.1 LlactaLAB.- Antecedentes.....	39
2.2.2 Proyecto “Patrones de movilidad activa y entorno urbano: Pies y Pedales”.....	40
2.2.3 Geoinformática.....	41
2.2.4 Geoprocesamiento.....	41
2.2.5 Sistemas de Información Geográfica.....	42



2.2.6 Consorcio geoespacial abierto	43
2.3 Estado del arte	44
2.3.1 Criterios de elegibilidad.....	44
2.3.2 Resultados.....	45
2.3.3 Descripción de los estudios incluidos.....	46
Capítulo III.....	49
3. Marco Metodológico	49
3.1 Materiales y Métodos	50
3.1.1 Idea	51
3.1.2 Planteamiento del problema	51
3.1.3 Inmersión inicial en el campo.....	52
3.1.4 Concepción del diseño de estudio	54
3.1.5 Definición de la muestra inicial del estudio y acceso a ésta.....	54
3.1.6 Recolección de los requerimientos	54
3.1.7 Análisis de los requerimientos.....	57
3.2 Concepto de software	60
3.3 Diseño de la arquitectura y núcleo del sistema.....	60
3.4 Documento de la Arquitectura.....	62
3.5 Evaluación de la Arquitectura	62
3.6 Documento refinado de la arquitectura	64
Capítulo IV	65
4. Conclusiones	65
4.1 Aportaciones de la tesis.....	69
4.2 Recomendaciones	69
4.3 Trabajos futuros.....	70
Bibliografía	72

Índice de Tablas Capítulo 2

Tabla 2.1 Vistas Arquitectónicas	24
Tabla 2.2 Enfoques de desarrollo de software	26
Tabla 2.3 Principios Claves de SOA.....	27
Tabla 2.4 Metodologías Ágiles	29
Tabla 2.5 Requerimientos Arquitectónicos.....	32
Tabla 2.6 Métodos de diseño Arquitectónico	33
Tabla 2.7 Métodos de evaluación arquitectónicos	36
Tabla 2.8 Resultados obtenidos de las fuentes bibliográficas.....	45
Tabla 2.9 Artículos obtenidos de las fuentes bibliográficas	46
Tabla 2.10 Plataformas consideradas para el diseño arquitectónico	48

Índice de Tablas Capítulo 3

Tabla 3.1 Descripción de fuentes de datos.....	53
Tabla 3.2 Actores del sistema identificados	58
Tabla 3.3 Identificación de riesgos de la arquitectura	63
Tabla 3.4 Identificación de no riesgos de la arquitectura	63

Índice de Tablas Capítulo 4

Tabla 4.1 Cálculo aproximado de costos de proveedores cloud.....	71
--	----

Índice de Figuras Capítulo 2

Figura 2.1 : Esquema de proceso de modelado de Arquitecturas de Software.....	31
Figura 2.2 Fases del método ATAM.....	37
Figura 2.3 Ejes Principales de Pies y Pedales.....	41

Índice de Figuras Capítulo 3

Figura 3.1 Metodología de la investigación para el diseño y evaluación de la arquitectura ...	50
Figura 3.2 Gestión de la arquitectura de software en Scrum	55
Figura 3.3 Proceso de recolección, almacenamiento, análisis y visualización de datos.....	59
Figura 3.4 Modelo de arquitectura del sistema propuesto	61

Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Víctor David Guillermo Flores en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "Propuesta de una Arquitectura de Software para una plataforma de datos sobre movilidad activa y entorno urbano.", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 02 de marzo de 2020



Víctor David Guillermo Flores

C.I: 0105867071

Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Christian Manuel Quinde Tenemaza en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "Propuesta de una Arquitectura de Software para una plataforma de datos sobre movilidad activa y entorno urbano.", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 02 de marzo de 2020



Christian Manuel Quinde Tenemaza

C.I: 0105783187

Cláusula de Propiedad Intelectual

Víctor David Guillermo Flores, autor del trabajo de titulación "Propuesta de una Arquitectura de Software para una plataforma de datos sobre movilidad activa y entorno urbano.", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 02 de marzo de 2020



Víctor David Guillermo Flores

C.I: 0105867071

Cláusula de Propiedad Intelectual

Christian Manuel Quinde Tenemaza, autor del trabajo de titulación "Propuesta de una Arquitectura de Software para una plataforma de datos sobre movilidad activa y entorno urbano.", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 02 de marzo de 2020



Christian Manuel Quinde Tenemaza

C.I. 0105783187



Agradecimientos

A Dios por guiar mis pasos, a mis queridos padres y hermanos por brindarme su apoyo incondicional.

A todos los docentes, quienes durante toda la carrera universitaria nos han compartido sus experiencias y valiosos conocimientos, de manera especial a la tutora de tesis, Ing. Paola Pesántez y a la Ing. Lorena Sigüenza co-tutora, por ayudarnos, brindarnos su tiempo y guiarnos hacia la culminación este trabajo de titulación.

Al grupo de investigación LlactaLAB y sus colaboradores, por todo el tiempo y apoyo brindado.

David Guillermo



Agradecimientos

A mis padres y hermanos por todo el apoyo que me han entregado, así como también por soportarme en las buenas y en las malas.

A los docentes con quienes tuve el honor de ser su alumno y absorber un poco de su conocimiento durante toda mi carrera universitaria y de manera especial a nuestras tutoras de tesis, Ing. Paola Pesántez y a la Ing. Lorena Sigüenza, gracias infinitamente por su ayuda, brindarnos el tiempo necesario en el transcurso del desarrollo de este trabajo y guiarnos para lograr esta meta tan anhelada.

Finalmente, agradecemos al grupo de investigación LlactaLAB por la colaboración y tiempo ofrecido.

Christian Quinde



Dedicatoria

A mis padres y hermanos, especialmente a Marcelo, por haberme forjado como la persona que soy en la actualidad.

A Carolina, una persona muy especial en mi vida, por estar conmigo y brindarme su apoyo en cada momento.

David Guillermo



Dedicatoria

Mis padres por ser la motivación de mi vida y guiarme por el camino del bien, por enseñarme valores que me han llevado a conseguir esta gran meta que de cierta forma es un logro de ellos también. A mis hermanos por soportarme en las buenas y en las malas, por creer en mí y apoyarme en el transcurso de mi vida.

Christian Quinde

Capítulo I

1. Introducción

Este capítulo ofrece una introducción al presente trabajo de titulación que se encuentra distribuido de la siguiente manera. La Sección 1.1 presenta la motivación y el contexto en el cual se ha desarrollado este trabajo de titulación. La Sección 1.2 expone la problemática actual y existente de las personas a las que va dirigida la propuesta. La Sección 1.3 detalla la justificación por la cual se ve la necesidad de realizar este proyecto y cuáles son los beneficiados. La Sección 1.4 describe la solución propuesta para resolver el problema que se busca solventar. La Sección 1.5 expone los objetivos tanto generales y específicos que se han identificado para este trabajo de titulación. La Sección 1.6 da a conocer brevemente la metodología de investigación adoptada para la elaboración de la propuesta. Finalmente, la Sección 1.7 presenta la estructura del trabajo que se realizará en los capítulos posteriores.

1.1 Motivación y contexto

LlactaLAB es un grupo de investigación que se enfoca principalmente en la construcción de conocimiento científico que aporte a la búsqueda de soluciones a los grandes desafíos de las ciudades (*¿Que es LlactaLAB?*, 2017). LlactaLAB ha creado el proyecto “Pies y Pedales” como solución a la sustentabilidad de las ciudades. Pies y Pedales es un proyecto de investigación cuyo objetivo es construir un marco analítico multidisciplinar para estudiar los patrones de movilidad de ciclistas y peatones, con el fin de producir conocimiento útil para una movilidad sostenible. Dicho proyecto se encuentra financiado por la Dirección de Investigación de la Universidad de Cuenca (DIUC) y con aporte adicional de la Universidad del Azuay. Este proyecto es relevante dado que aporta conocimiento sobre movilidad en Cuenca, y sirve como soporte a la toma de decisiones sobre movilidad sostenible en la región (*Pies y Pedales – LlactaLAB – Universidad de Cuenca*, 2018).

El presente trabajo de titulación está enmarcado dentro del proyecto de investigación “Pies y Pedales 2”, el cual busca consolidar y profundizar el estudio del comportamiento espacial y las percepciones de movilidad activa en Cuenca desde una dimensión multidisciplinar. Uno de los objetivos de Pies y Pedales 2 es el desarrollar un toolbox de análisis de datos de movilidad en donde se apliquen métodos de geoprocésamiento, estadísticos y de minería de datos para analizar factores de movilidad y entorno urbano.

Una de las principales motivaciones para el desarrollo de este proyecto es la del almacenamiento adecuado de los datos de movilidad, dado que la desintegración o ejecución independiente de los procesos de manipulación/gestión de datos que componen el proyecto “Pies y Pedales” para la generación de resultados conlleva a una difícil gestión de la

información, esto tiene como consecuencias un retardo de los resultados y la pérdida y/o desorganización de información. Otro punto motivante e importante es la necesidad de generar conocimiento científico veraz para la toma de decisiones, debido a que los procedimientos de análisis que se realizan actualmente utilizan herramientas de terceros para generar resultados tanto cualitativos como cuantitativos. Además, la necesidad de tener una arquitectura de software integral que soporte procesos de recolección, análisis y visualización de datos de movilidad es un punto primordial, ya que, hasta el momento, LlactaLAB no cuenta con una plataforma que le permita realizar todos estos procesos de manera conjunta.

1.2 Planteamiento del problema

Actualmente, el buen manejo de datos y procesos, a través de un software adecuado, constituye un factor primordial. Sabiendo que la información es el elemento más valioso, éste ha llegado a convertirse en una importante herramienta para la toma de decisiones. De esta manera, la construcción de una arquitectura de software confiable y perdurable en el tiempo es un aspecto crítico para el manejo de la información dentro del contexto de movilidad que permita la toma de decisiones para lograr una ciudad sustentable.

LlactaLAB tiene como misión proveer la mejor evidencia científica posible para que las ciudades puedan tomar decisiones y enfrentar los retos de movilidad y sostenibilidad del siglo XXI. Ahora bien, al ser LlactaLAB un grupo de investigación, tiene la necesidad de recolectar datos para ser almacenados en algún lugar, lo que implica que tiende a acumular grandes cantidades de datos con el paso del tiempo. Estos datos deben convertirse en información y posteriormente interpretados en conocimiento. Hasta el momento, el grupo de investigación LlactaLAB se ha limitado a gestionar los diferentes tipos de datos: archivos geoespaciales, encuestas, registros en hojas de cálculo, etc., por separado y de forma manual.

Éste fue un punto clave para que surja el interés en diseñar un mecanismo apto para el manejo adecuado de estos datos.

1.3 Justificación

La movilidad, en sus inicios, fue un tema tratado por la ingeniería de tránsito o transporte, en donde, para resolver los problemas que surgían, se razonaba de la siguiente manera: ¿cuál es el ancho de la calle?, ¿cuántos carros entran?, entre otros. El razonamiento anterior es considerado una visión clásica ya que todos los estudios y análisis de movilidad venían dados por tal razonamiento. En la actualidad, esta forma de dar solución ha ido cambiando a problemas de movilidad, tal es el caso que ahora la movilidad se estudia como un fenómeno de comportamiento humano. Al ser identificado de esta manera, quiere decir que para resolver los distintos problemas que surgen de movilidad no se necesitan únicamente ingenieros, sino también arquitectos, urbanistas, sociólogos, geógrafos, etc. Esto implica que también se tiene una enorme cantidad de datos heterogéneos de movilidad que se deben integrar, analizar y triangular, con el objetivo de generar conocimiento que ayude a la toma de mejores decisiones para una ciudad.

Luego de realizada una investigación exploratoria, no se ha logrado encontrar ninguna arquitectura de software definitiva capaz de gestionar los datos de movilidad de una manera adecuada. El trabajo de titulación propuesto busca resolver este problema de integración y gestión de datos que el grupo de investigación LlactaLAB ha identificado en el transcurso del proyecto Pies y Pedales. Dentro del contexto del grupo de investigación LlactaLAB, el desafío es concretamente la interdisciplinariedad, es decir, existen datos que provienen de diferentes proyectos, investigadores y campos de conocimiento. Por lo tanto, lo que busca este trabajo es proponer una arquitectura de software que permita que toda esta información esté disponible para todos en una sola plataforma, de manera que los distintos involucrados

puedan realizar sus actividades individuales, pero también puedan trabajar en conjunto, discutir y tomar decisiones.

1.4 Solución Propuesta

El propósito de este estudio, de tipo investigación-acción con un enfoque participativo, es diseñar y describir una arquitectura de software orientada a la gestión de datos sobre patrones de movilidad dentro del entorno urbano. Se propone a la ciudad de Cuenca como caso de estudio y que pueda ser extrapolable a otras regiones urbanas, que permita la toma de decisiones para enfrentar los retos del siglo XXI. La arquitectura planteada para el ambiente a tratar debe permitir la gestión e integración de datos sobre movilidad, generados por distintas fuentes de diferente naturaleza con distintos formatos. Estos datos deben ser recolectados, almacenados, analizados y visualizados de una manera adecuada, posibilitando una mejor planificación y toma de decisiones para la ciudad dentro del contexto de movilidad. El diseño de la arquitectura de software a proponer soporta más concretamente las siguientes funcionalidades: a) Pre-procesamiento, que incluye una función para la importación automática o semiautomática de los datos recolectados y su preparación para el almacenamiento e integración en una estructura común; b) Análisis, que soporte métodos basados en minería de datos, análisis geoespacial, y análisis estadístico; c) Post-procesamiento, que soporte funciones de visualización y publicación de los resultados de forma amigable e interactiva.

La solución propuesta en este trabajo de titulación busca aportar una arquitectura de software que permita soportar las funcionalidades detalladas anteriormente.

1.5 Objetivos generales y específicos

1.5.1 Objetivo General

Diseñar una arquitectura de software para una plataforma de análisis de datos sobre movilidad.

1.5.2 Objetivos específicos

- Construir un marco teórico que proporcione un conocimiento fuerte para el desarrollo, así como un estado del arte de arquitecturas realizadas o propuestas para plataformas de movilidad.
- Generar una propuesta de arquitectura de software para la plataforma de gestión y análisis de datos de movilidad.
- Identificar las herramientas o tecnologías de software que mejor se adapten a la arquitectura.
- Validar la arquitectura de software propuesta.

1.6 Metodología de investigación

En esta sección se incluye el enfoque de investigación seleccionado para este proyecto, las técnicas y los instrumentos que serán utilizados para llevar a cabo la indagación.

Para dicho procedimiento, utilizaremos la metodología propuesta por Hernández Sampieri et al. (2006), desde el punto de vista cualitativo. Este enfoque se selecciona cuando el propósito es examinar la forma en que los individuos perciben y experimentan los fenómenos que los rodean, profundizando en sus puntos de vista, interpretaciones y significados. El mismo que es recomendable cuando el tema de estudio ha sido poco explorado o no se ha hecho investigación al respecto en ningún grupo social específico. Este

proceso tiene múltiples características por las cuales se ha optado por el uso del mismo, entre ellas está: el uso de la lógica y proceso inductivo para explorar, describir y generar perspectivas teóricas. Se basa en métodos de recolección de datos no estandarizados ni predeterminados completamente; esto implicaría obtener las perspectivas y puntos de vista de los participantes. Así mismo, permite al investigador utilizar técnicas para recolectar datos como la observación no estructurada, revisión de documentos, discusión en grupo, interacción e introspección con grupos o comunidades. Adicionalmente, el enfoque cualitativo provee un proceso de indagación más flexible y se mueve entre las respuestas y el desarrollo de la teoría, evalúa el desarrollo natural de los sucesos, y permite al investigador introducirse en las experiencias de los participantes y construir el conocimiento.

El enfoque cualitativo se basa en nueve fases: establecimiento de la idea, planteamiento del problema, inmersión inicial en el campo, concepción del diseño de estudio, definición de la muestra inicial y acceso a ésta, recolección de datos, análisis de datos, interpretación de resultados y finalmente la elaboración del reporte de resultados. Es necesario mencionar que esta metodología no es un proceso lineal, sino iterativo o recurrente donde a veces es necesario regresar a etapas previas. Las fases en realidad son acciones para adentrarnos más en el problema de investigación y la tarea de recolectar y analizar datos es permanente.

Este enfoque se basa fundamentalmente en la fase del planteamiento del problema de investigación. En la literatura de Sampieri se mencionan seis elementos fundamentales para el planteamiento del problema: objetivos de investigación, preguntas de investigación, justificación de la investigación, viabilidad de ésta, evaluación de las deficiencias en el conocimiento del problema y definición inicial del ambiente o contexto (Hernández Sampieri et al., 2006).

Para este caso hemos considerado como parte de nuestro ambiente inicial a miembros del grupo de investigación LlactaLAB y profesionales/docentes de la Universidad de Cuenca. Adicionalmente, optamos por consultar opiniones de expertos en temas como: arquitecturas de software para gestión de datos sobre movilidad, métodos de diseño y validación arquitectónica, herramientas y tecnologías adecuadas usadas dentro de este contexto. Además de esto, un acercamiento a los datos y tecnologías que actualmente se gestionan dentro del grupo de investigación ha sido también considerado un elemento importante en el desarrollo de este trabajo. Finalmente, como parte del desarrollo de este proyecto se han realizado anotaciones, conversaciones y grabaciones. Todo esto como ya fue mencionado desde un enfoque cualitativo dentro de un marco o abordaje de investigación-acción.

Referente a la parte técnica, para el desarrollo de la propuesta se hace uso del marco de desarrollo ágil denominado Scrum. Dada la importancia que existe sobre el papel de la arquitectura de software en las metodologías ágiles, los autores Edwin Rafael Mago y Germán Harvey Alférez (El Papel de la Arquitectura de Software en Scrum, 2015), presentan una propuesta para gestionar la arquitectura de software en Scrum. Dicha propuesta consiste básicamente en adicionar una *Iteración* inicial llamada “*Iteración 0*” al inicio del ciclo de desarrollo para analizar y diseñar la generalidad del sistema de manera iterativa, de tal forma que satisfaga los requisitos y sea entendible por los miembros del equipo desde sus diferentes puntos de vista durante el desarrollo. Todo esto con el objetivo de adoptar una estrategia de desarrollo incremental, en lugar de una planificación y ejecución completa, realizando entregas incrementales y permitiendo un mejor control de los avances del proyecto.

1.7 Estructura del trabajo

- El Capítulo 2 se enfoca en el marco teórico de los principales aspectos para el desarrollo del presente proyecto. Entre los temas tratados se mencionan: la movilidad

urbana, arquitecturas de software, software ágil, un proceso para el modelado de una arquitectura de software, sistemas de almacenamiento de datos, entre otros.

- El Capítulo 3 analiza el desarrollo del marco metodológico y su relación con la metodología de desarrollo de la arquitectura, aquí se mencionan los materiales y métodos usados, tanto de investigación como de diseño de la arquitectura de software. Este capítulo describe los puntos principales del presente trabajo que son: levantamiento de requerimientos, diseño arquitectónico, desarrollo de la *Iteración 0* de Scrum para la generación de esta arquitectura, su evaluación y como punto final tiene el cierre del proyecto con la generación final de un diseño arquitectónico de software.
- El Capítulo 4 presenta los resultados, conclusiones, recomendaciones, trabajos futuros y aportes de la tesis, obtenidas a lo largo del desarrollo de este trabajo.

Capítulo II

2. Marco Teórico

En este capítulo se dan a conocer conceptos principales correspondientes a términos fundamentales utilizados en el presente trabajo de titulación. Este capítulo está distribuido de la siguiente manera: La Sección 2.1 presenta el significado de una arquitectura de software, sus características, los tipos de arquitecturas existentes y los más usados en la actualidad. Además, se presentan las metodologías ágiles y los beneficios que conlleva su uso y algunos métodos ágiles más usados con una descripción breve. Por último, se presenta una descripción de la metodología que se va a utilizar, así como el proceso para modelar una arquitectura de software, la determinación de los requerimientos, diseño de la arquitectura y la evaluación del modelo arquitectónico. La Sección 2.2 define conceptos fundamentales sobre movilidad urbana, una descripción sobre el grupo de investigación LlactaLAB y sus antecedentes. Posteriormente, se presenta un enfoque sobre el proyecto “Patrones de movilidad activa y entorno urbano: Pies y Pedales”, asimismo una descripción sobre geoinformática, el geoprocésamiento, sistemas de información geográfica y los estándares establecidos por el consorcio geoespacial abierto. La Sección 2.3 describe el estado del arte del trabajo. Todos estos puntos otorgan una base de conocimiento fuerte para el desarrollo y entendimiento de los siguientes capítulos.

2.1 Arquitectura de Software

La Arquitectura de Software es la subdisciplina más importante de la Ingeniería de Software; en la actualidad, se considera que este término es uno de los más usados y menos

entendidos en las etapas del círculo de software ya que, por ejemplo, hay una lista interminable de diversidad de vistas. Por lo tanto, su aplicación es compleja y hasta considerada como una actividad potencialmente peligrosa para la industria, si es que no se diseña adecuadamente (Gorton, 2014).

Una arquitectura de software describe múltiples aspectos de software; esta descripción se vuelve muchos más clara cuando se realiza mediante vistas o modelos. Existen tres vistas fundamentales (ver Tabla 2.1).

Vista	Característica
Vista estática	Narra los componentes que tiene la arquitectura
Vista funcional	Narra lo que hace o la funcionalidad de cada componente
Vista dinámica	Narra el comportamiento de los componentes en el tiempo y su interacción entre sí

Tabla 2.1 Vistas Arquitectónicas
Elaborado por: (Los Autores)
Fuente: (Sommerville & others, 2011)

Existe un consenso en adoptar al lenguaje de modelamiento unificado (UML, por sus siglas en inglés, *Unified Modeling Language*) como lenguaje estándar para describir todas las vistas o modelos mediante diagramas de actividades, diagramas de componentes, diagramas de flujo, entre otros.

Como se sabe, los sistemas de software están diseñados para satisfacer los objetivos comerciales de las organizaciones, grupos u otras entidades. La arquitectura es un puente entre esos objetivos comerciales y el sistema final resultante. (Bass et al., 2012; Keeling, 2017). Una arquitectura de software define la estructura y relaciones de cada sistema, subsistema o componente de software que se modela para formar un todo cohesivo (Rogers, 2005).

Existen múltiples formas de desarrollar software, las más importantes son:

Tipo	Descripción	Ventajas	Desventajas
Desarrollo de software tradicional	Considera primero los requerimientos del proyecto, luego el análisis y diseño de software, posteriormente, la codificación e integración de los diferentes componentes, seguidamente de las pruebas del software y, finalmente, el mantenimiento.	Forma más simple y tradicional de desarrollar software.	El mantenimiento resulta complicado si no se tienen la respectiva documentación.
Arquitectura dirigida por modelos (ADM o MDA, por sus siglas en inglés, <i>Model-Driven Architecture</i>)	Se enfoca en las etapas de diseño e implementación del desarrollo de software En una MDA, los artefactos se generan automáticamente o semi-automáticamente a partir del Modelo Independiente de la Plataforma (MIP o PMI, por sus siglas en inglés, <i>Platform Independent Model</i>).	Una vez identificados y diseñados los metamodelos, el desarrollo se simplifica y puede lograrse en cortos periodos de tiempo.	Toma más tiempo el preparar, analizar y modelar los metamodelos y no es una tarea trivial.
Arquitecturas orientadas a objetos	Se basa en componentes. Los componentes de un sistema contienen los datos y las operaciones que deben aplicarse para manipularlos. La coordinación y comunicación entre los componentes asociados se realiza a través de la transferencia de mensajes.	Fácil modelamiento de componentes, ya que relaciona o asigna responsabilidades y funcionalidades a clases u objetos.	Puede desencadenar definición de innumerables clases al trabajar con un sistema extenso, lo que dificulta el entendimiento y trazabilidad de funcionalidades.
Arquitectura por capas	Organiza el sistema en capas con funcionalidad relacionada. donde las capas de nivel inferior proveen servicios a las capas de encima. Permite la sustitución de capas completas en tanto se conserve la interfaz.	Fácil reemplazo de los componentes frente a un cambio, agregación o eliminación de funcionalidades o mantenimiento del sistema, sin afectar a las demás capas.	La documentación puede resultar extensa y compleja ya que se debe tener un detalle de cada acción en cada capa, mayor codificación dado que las funcionalidades deben estar relacionadas entre capas.

Arquitectura cliente-servidor	La funcionalidad del sistema se organiza en servicios, y cada servicio lo entrega un servidor independiente. Los clientes son usuarios de dichos servicios y para utilizarlos ingresan a los servidores.	Útil cuando, desde varias ubicaciones, se tiene que ingresar a los datos en una base de datos compartida.	El servidor se transforma en un punto de quiebre, por lo que posiblemente se tenga que implementar un balanceador de carga.
Arquitecturas orientadas a Servicios (SOA, por sus siglas en inglés, <i>Services Oriented Architecture</i>)	Es una forma de desarrollar sistemas distribuidos en la que los componentes del sistema son servicios independientes y se ejecutan de ser posible en computadoras distribuidas geográficamente. Estos servicios son reutilizables, con interfaces públicas bien definidas, donde los proveedores y consumidores interactúan en forma desacoplada.	Débil acoplamiento entre componentes (servicios completamente independientes). Interoperabilidad (servicios funcionando con diferentes tecnologías). Escalabilidad, permite agregar funcionalidades sin mayor esfuerzo. El OGC proporciona estándares basados en SOA para SIGweb	Se puede caer en un deadlock en algún proceso, ya que la orquestación de servicios no es tarea trivial. El orquestador si lo tuviera, puede resultar en un único punto de quiebre y si cae se pierde la disponibilidad de todos los servicios.

Tabla 2.2 Enfoques de desarrollo de software

Elaborado por: (Los Autores)

Fuente: (Vivas et al., 2013; Kleppe et al., 2003; Mellor et al., 2004; Rogers, 2005; Sommerville & others, 2011; Delgado et al., 2006)

Para el desarrollo inicial de una arquitectura se puede usar una técnica de orientación a objetos denominada Diseño Basado en Responsabilidad (DBR), para ayudar a definir los componentes clave de la arquitectura. Según Gorton (2014), esto es extremadamente útil para estructurar componentes a nivel arquitectónico en un proyecto.

2.1.1 Arquitectura Orientada a Servicios

Una Arquitectura Orientada a Servicios (SOA) es un patrón arquitectónico en el diseño de software en el que los componentes de la aplicación proveen servicios a otros

componentes mediante un protocolo de comunicaciones a través de una red. En este patrón, un usuario puede ser un usuario humano o una aplicación cliente.

El término servicio es esencialmente el servicio de negocios de la entidad. Una aplicación diseñada utilizando SOA está compuesta típicamente por varios servicios. Estos servicios tienen una naturaleza ligeramente acoplada que permite que se agreguen nuevos servicios y que los existentes se modifiquen (o incluso se retiren) rápidamente, según la naturaleza dinámica de la entidad (Güner, 2015). SOA se basa en algunos principios clave que se mencionan a continuación:

Principios	Descripción
Contrato de servicio estandarizado	Debe tener algún tipo de descripción que detalle de qué se trata el servicio, nombre, función, etc.
Bajo Acoplamiento	Debe existir la menor dependencia posible entre los servicios web y el cliente que los invoca.
Abstracción del servicio	El servicio no debe exponer cómo ejecuta su funcionalidad; solo debe decirle a la aplicación cliente lo que hace y no cómo lo hace.
Reutilización del servicio	Todos los servicios deben tener en mente y como base el concepto de reutilización.
Interoperabilidad del servicio	Los servicios deben usar estándares que permitan a diversos suscriptores utilizarlos. En los servicios web, los estándares como XML y la comunicación a través de HTTP se utilizan para garantizar que estos se ajustan a este principio.

Tabla 2.3 Principios Claves de SOA

Elaborado por: (Los Autores)

Fuente: (Alvarez Ballestas & Rodelo Calvo, 2008; Sommerville & others, 2011)

La Arquitectura Orientada a Servicios tiene varios beneficios sobre las arquitecturas cliente-servidor, ya que proporciona un entorno abierto e interoperable. Promueve el intercambio de datos geoespaciales (C. Zhang et al., 2007). Según Friis-Christensen,

Ostkander y Bernard (2007), SOA es más eficiente que las aplicaciones SIG estándar y proporciona información en tiempo real con mayor facilidad.

2.1.2 Software Ágil

Concepto usado en la ingeniería de software para describir las metodologías de desarrollo incrementales o ágiles (Leffingwell, 2007). Sus beneficios son: promover la entrega rápida de valor a los clientes, proporcionar visibilidad oportuna, regular la solución a todas las partes interesadas y permitir mayor calidad, productividad en un grupo de desarrollo.

La Tabla 2.4 describe tres de los métodos ágiles más populares y utilizados en la actualidad; estos podrían posteriormente ser considerados en la implementación o desarrollo de la plataforma de movilidad.

Método	Descripción	Ventajas	Desventajas
Programación Extrema (XP, por sus siglas en inglés, <i>Extreme Programming</i>)	Se centra en el desarrollo, en lugar de aspectos de gestión de proyectos. XP fue diseñada para que las organizaciones tuvieran libertad de adoptar la totalidad o parte de la metodología.	Útil para proyectos pequeños y poco complejos. Resultados en poco tiempo ya que se evita el proceso de elaboración de diagramas y modelos.	Código extenso, no reutilización del mismo. Si la documentación no es adecuada, gestionar un mantenimiento o requisito resulta complejo.
Proceso Unificado de Rational (PUR o RUP, por sus siglas en inglés, Rational Unified Process)	Es un marco de proceso, más no un proceso específico, usado por entidades en donde se requiere un mayor control con los procesos desarrollados.	Se adapta al contexto particular del proyecto. Es una forma disciplinada de asignar tareas y responsabilidades en un equipo de desarrollo (quién hace qué, cuándo y cómo).	Puede resultar en procesos muy largos para un desarrollo corto o simple de realizar, ya que debe pasar por cada área o proceso determinado.
Scrum	Propuesta por Ken Schwaber y Jeff Sutherland en 1995. Es la metodología ágil más usada y extendida en el mundo. A diferencia de XP, Scrum incluye	Pensado para proyectos en entornos complejos, donde se necesita obtener resultados pronto,	Es necesaria una exhaustiva definición de tareas y plazos. Requiere de una alta formación para los

	tanto los procesos de gestión como los de desarrollo. En Scrum se realizan entregas parciales y regulares del producto final.	donde los requisitos son cambiantes o poco definidos.	que la utilicen.
--	---	---	------------------

Tabla 2.4 Metodologías Ágiles

Elaborado por: (Los Autores)

Fuente: (Leffingwell, 2007; Sommerville & others, 2011)

2.1.3 Scrum

La metodología define los siguientes roles: el dueño del producto (*Product Owner*), quien determina las prioridades de los entregables; el maestro de Scrum (*Scrum Master*), quien administra y facilita la ejecución del proceso; el equipo de desarrollo (*Development Team*), quienes trabajan en conjunto para entregar resultados en cada iteración.

La metodología está compuesta por eventos como:

- La iteración: es considerado el núcleo del Scrum, en el cual se crea un incremento de producto “Terminado” utilizable y potencialmente desplegable.
- La Planeación de Iteración: en donde se establece el trabajo, se crea mediante la colaboración del Equipo Scrum completo.
- Un Objetivo de Iteración: es una meta que puede lograrse mediante la implementación de la Lista de Producto.
- Iteración diaria: consiste en reunirse al menos una vez al día para planificar entregas.
- Revisiones diarias: consisten en verificar o validar diariamente si los requerimientos se están cumpliendo.

Para este trabajo de titulación se ha optado por hacer uso de la metodología Scrum con una pequeña variación en la cual se adiciona una *Iteración* inicial llamada “*Iteración 0*” al inicio del ciclo de desarrollo. Esta variación se la realiza con el objetivo principal de

analizar y diseñar la generalidad del sistema (arquitectura) que satisfaga los requisitos y sea entendible por los miembros del equipo desde sus diferentes puntos de vista durante el desarrollo (Bass et al., 2012). En la *Iteración 0* se construye la arquitectura de forma iterativa mediante un análisis preliminar de los conductores arquitectónicos (requisitos funcionales, de calidad y del negocio). No se necesitan tener todos los requisitos claros para comenzar la fase de diseño arquitectónico (Muhammad Ali Babar & Mistrik (Eds.), 2013).

Para determinar los conductores o también llamados drivers arquitectónicos, se deben identificar los objetivos del negocio de más alta prioridad. Estos objetivos del negocio se convierten en los escenarios de calidad o casos de uso. De esta lista, se deben escoger los que tendrán un mayor impacto sobre la arquitectura. El resultado de la *Iteración 0* es un documento inicial que explica la arquitectura. El documento puede basarse en los pasos definidos por el método de diseño dirigido por atributos (ADD, por sus siglas en inglés, *Attribute Driven Design*) o cualquier otro que se crea conveniente. Para este trabajo de titulación se hace uso de ADD dado que este método ya ha sido probado exitosamente en proyectos anteriores (Bass et al., 2012; Wood, 2007). El documento inicial de la arquitectura se debe evaluar con el fin de saber si la arquitectura cumple con los requisitos de calidad. Para realizar esta evaluación, se propone el método de análisis de la compensación de arquitectura (ATAM, por las siglas en inglés, *Architecture Tradeoff Analysis Method*). El ATAM revela cuán bien una arquitectura satisface los objetivos particulares de calidad y provee una aproximación de cómo los objetivos de calidad interactúan (Bass et al., 2012).

2.1.4 Proceso del modelado de una arquitectura de software

El proceso de modelado de una arquitectura de software se basa en tres etapas (ver Fig. 2.1): 1) definir los requerimientos para la arquitectura, comprende en crear un modelo a partir de un análisis de los requerimientos que serán una guía para el diseño de la arquitectura

basado en los atributos de calidad esperados; 2) diseño de la arquitectura, implica delimitar la estructura y las responsabilidades de los componentes que comprenderán la arquitectura de software; 3) validación de la arquitectura, esto significa “probar” la arquitectura, típicamente pasando a través del diseño contra los requerimientos del sistema (Sommerville & others, 2011). Este proceso puede ser modificado o alterado dependiendo de las necesidades del proyecto o la organización.

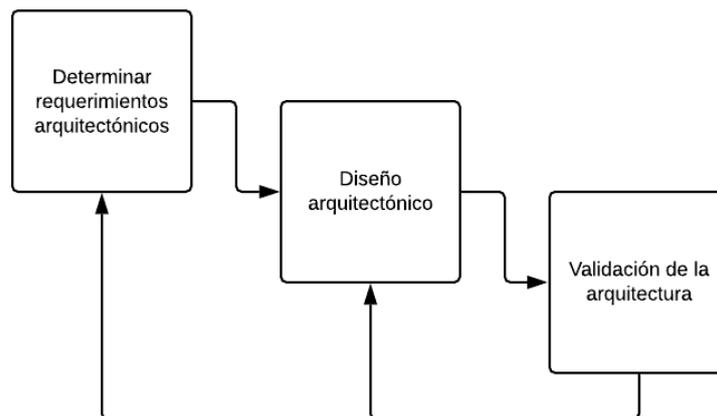


Figura 2.1 : Esquema de proceso de modelado de Arquitecturas de Software

Elaborado por: (Los Autores)

2.1.5 Requerimientos arquitectónicos

Los requerimientos arquitectónicos para un sistema son descripciones de lo que el sistema debe hacer, el servicio que ofrece y las restricciones en su operación (Sommerville & others, 2011). La Tabla 2.5 describe los diversos tipos de requerimientos arquitectónicos.

Requerimientos arquitectónicos	Características
Requerimientos de Usuario	Representan los requerimientos abstractos de alto nivel de un sistema. Son enunciados descritos en lenguaje natural, es decir, lo que el usuario ha solicitado.

Requerimientos de Sistema	Son usados para describir detalladamente lo que el sistema debe hacer. Son descripciones muy bien detalladas de las funciones a cumplir del sistema.
Requerimientos Funcionales	Son aquellos relacionados con la funcionalidad de la aplicación. Son declaraciones de los servicios que el sistema debe proveer.
Requerimientos no Funcionales (requerimientos de calidad)	Pueden relacionarse con propiedades emergentes del sistema, como la fiabilidad, tiempo de respuesta, y uso de almacenamiento. No hablan de “lo que” hace el sistema, sino de “cómo” lo hace.

Tabla 2.5 Requerimientos Arquitectónicos

Elaborado por. (Los Autores)

Fuente: (Sommerville & others, 2011; Cervantes Maceda et al., 2016; Mora, 2015)

2.1.6 Diseño de arquitecturas de software

El diseño se presenta como la especificación de un objeto, establecido por cierto agente, que busca conseguir diversos objetivos, en un ambiente particular, utilizando un conjunto de componentes básicos, cumpliendo una diversidad de requerimientos y acogiéndose a determinadas restricciones (Ralph & Wand, 2009). En un contexto de arquitectura de software: el objeto hace referencia a las distintas estructuras (físicas, lógicas, de ejecución) que componen la arquitectura de software; el agente representa al(los) arquitecto(s) de software u otros encargados del diseño, los objetivos son la satisfacción de los requerimientos que influyen a la arquitectura (los *drivers*).

En un diseño de una arquitectura se presentan diversos niveles. El diseño de la arquitectura es el enfocado en la toma de decisiones para establecer los *drivers* arquitectónicos y la generación de una estructura para satisfacerlos. El diseño de interfaces ocurre con la identificación de todos los módulos, elementos faltantes y el diseño de las interfaces o contratos que deben satisfacer los módulos identificados. El diseño detallado de los módulos ocurre durante la construcción del sistema, es decir, una vez que se han

establecido los módulos y sus interfaces, se pueden diseñar los detalles de implementación de estos módulos previo a su codificación y prueba (Cervantes Maceda et al., 2016).

2.1.7 Métodos de diseño arquitectónico

Existen diversos métodos que nos permiten un diseño mucho más sistemático de una arquitectura, los más usados en la actualidad están detallados en la Tabla 2.6.

Método	Descripción	Ventaja	Desventaja
Diseño dirigido por atributos (ADD, por sus siglas en inglés, <i>Attribute Driven Design</i>)	Se centra en un enfoque de “divide y vencerás”, la arquitectura se diseña de manera iterativa y en cada <i>Iteración</i> se toma una parte o elemento y se descompone en subelementos y así continuamente hasta cubrir todos los elementos.	Proporciona la guía más detallada para el diseño de la arquitectura	No se especifica claramente cuánto abarca una iteración. No proporciona criterios precisos de cuándo termina la actividad de diseñar
Método de diseño centrado en la arquitectura (ACDM, por sus siglas en inglés, <i>Architecture-Centric Design Method</i>)	Tiene como propósito principal que el equipo correspondiente establezca el diseño arquitectónico inicial, o refine este como resultado de la evaluación.	Diseñar y documentar no son etapas distintas, sino que aquí se crea o se refina la arquitectura y se crea o se actualiza la documentación.	Difícil de implantar en organizaciones muy orientadas al desarrollo de sistemas bajo un enfoque secuencial (en cascada), en el cual se diseña toda la arquitectura sin realizar <i>Iteraciones</i>
Método de definición de arquitecturas de Rozanski y Woods	Cubre actividades que van más allá de la toma de decisiones de diseño y abarca en cierta forma todo el ciclo de desarrollo arquitectónico.	Permite considerar cambios en los requerimientos originales que deban hacerse posterior a la evaluación.	No proporciona pasos tan detallados para realizar las actividades de toma de decisiones de diseño en sí mismas.

Tabla 2.6 Métodos de diseño Arquitectónico

Elaborado por: (Los Autores)

Fuente: (Cervantes Maceda et al., 2016; Lattanze, 2009; Rozanski & Woods, 2005; Bass et al., 2000)

El método de diseño arquitectónico considerado para la elaboración de este proyecto es ADD, el mismo que se describe con un poco más de detalle a continuación.

2.1.8 Diseño dirigido por atributos (ADD)

El método de diseño dirigido o guiado por atributos (ADD) es una propuesta del Instituto de Ingeniería de Software (SEI, por sus siglas en inglés, *Software Engineering Institute*) para realizar el diseño de las arquitecturas (Wojcik et al., 2006). Los pasos del método son:

1. Confirmar que se tiene información suficiente sobre los drivers arquitectónicos
2. Elegir un elemento del sistema a descomponer
3. Identificar drivers arquitectónicos asociados al elemento
4. Elegir un concepto de diseño que satisfaga los drivers
5. Instanciar los elementos y asignar responsabilidades
6. Definir interfaces para los elementos instanciados
7. Verificar y refinar requerimientos y transformarlos en restricciones para los elementos instanciados
8. Repetir los pasos anteriores para elementos que requieran un refinamiento mayor hasta cubrir la mayoría de los drivers

Las iteraciones en este método se generan hasta que se vea necesario tomar decisiones de diseño adicionales para satisfacer los drivers arquitectónicos o hasta que termine el tiempo estipulado para diseñar.

2.1.9 Evaluación de arquitecturas de software

La evaluación de las arquitecturas permite la detección de riesgos e incumplimientos que pueden ser causa de retrasos o problemas muy serios en los proyectos. En la evaluación se examina el sistema o subsistema para ver si cumple con los criterios de calidad o que tanto se desvía de los requerimientos arquitectónicos. Se considera como una de las mejores prácticas de ingeniería de software, ya que se recorre los artefactos para detectar errores o inconsistencias. (Cervantes Maceda et al., 2016).

Los métodos más conocidos de revisiones arquitectónicas se encuentran detallados en la Tabla 2.7.

Método	Descripción	Ventaja	Desventaja
Recorridos informales al diseño	Ampliamente usado. El equipo de arquitectura hace una presentación a otras audiencias que pueden o no ser personal técnico, la cual no conlleva precisamente un proceso definido y puede usarse para mostrar el tipo de diseño que sea, incluido el arquitectónico.	Muy fácil de realizar y no requiere de mucho esfuerzo	Limitado en su capacidad de encontrar elementos de riesgo de la arquitectura
Método de análisis de equilibrios de la arquitectura (ATAM, por sus siglas en inglés, <i>Architecture Tradeoff Analysis Method</i>)	Toma como entrada cualquier artefacto que haya sido producido por el diseño, evalúa las consecuencias de las decisiones arquitectónicas respecto de los requerimientos de drivers del sistema y finalmente el resultado de la evaluación es una lista de riesgos en ella	Genera “recomendaciones” para mejorar el diseño de la arquitectura	Se requiere personal experimentado
Método de diseño centrado en la arquitectura (ACDM)	Plantea que la evaluación puede ser realizada por el equipo de desarrollo o por uno externo y pone énfasis en que la arquitectura sea capaz de satisfacer los requerimientos y escenarios funcionales, sin dejar a un lado los atributos de calidad.	Se centra en los requerimientos funcionales principalmente	Complicado y extenso.
Revisiones	Nombra a un diseñador de la arquitectura;	Si se realiza	Conlleva tiempo

<p>activas para diseños intermedios (ARID, por sus siglas en inglés, <i>Active Reviews for Intermediate Designs</i>)</p>	<p>sin embargo, puede considerarse idéntico al rol de arquitecto. Plantea dos fases y nueve pasos; de estos, cuatro son actividades necesarias para tener la junta de revisión, y cinco se realizan durante la junta</p>	<p>correctamente se puede llegar a un ahorro de tiempo. Informa a los diseñadores sobre si su diseño es adecuado</p>	<p>y puede no darse satisfactoriamente si el proyecto se encuentra presionado. Revisores requieren un nivel técnico considerable</p>
<p>Prototipos o experimentos</p>	<p>Una creación de prototipos adecuados es un elemento crítico de una arquitectura completa, así como también una estrategia de evaluación, y las posibilidades de éxito de un programa mejoran drásticamente cuando se proporciona y se utiliza correctamente.</p>	<p>Muestra si las ideas de los diseños son o no realizables, permitiendo la adquisición de conocimiento valioso sin tener que desarrollar el sistema por completo</p>	<p>Determinar la efectividad en cuanto a sus atributos de calidad es complejo. Consume tiempo y recursos.</p>

Tabla 2.7 Métodos de evaluación arquitectónicos

Elaborado por: (Los Autores)

Fuente: (Cervantes Maceda et al., 2016; Kazman, Klein y Clements, 2000; Bass et al., 2013; Borky & Bradley, 2019)

El método de evaluación de la arquitectura considerado para la elaboración de este proyecto es ATAM y como un complemento adicional el de prototipos, los mismos que se describen a continuación.

2.1.10 Método de análisis de equilibrios de la arquitectura (ATAM)

Este método tiene como objetivo “evaluar las consecuencias de las decisiones arquitectónicas respecto de los requerimientos de drivers del sistema” (Kazman, Klein y Clements, 2000). ATAM toma como entradas la documentación de la arquitectura y cualquier otro artefacto que haya sido producido por el diseño de ésta. El resultado de hacer una evaluación usando el método es una lista de riesgos en ella. Se dan cinco fases para realizar evaluaciones de esta clase, como se muestra en la Figura 2.2.

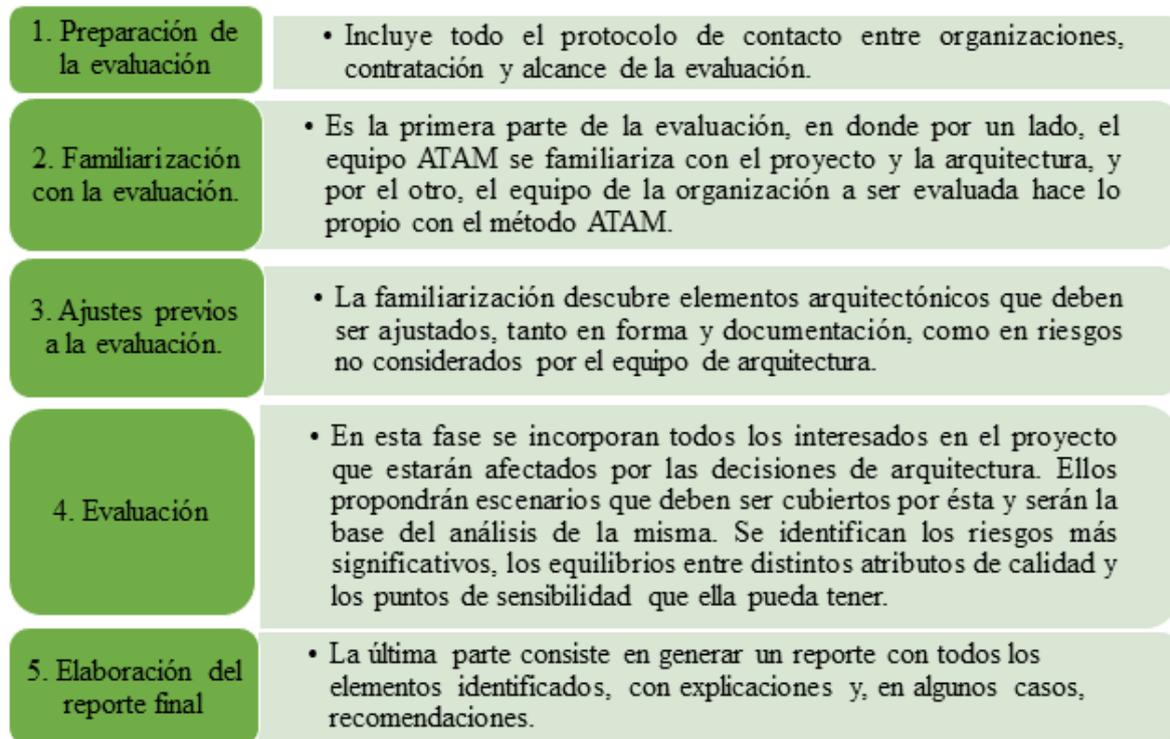


Figura 2.2 Fases del método ATAM

Elaborado por: (Los Autores)

Fuente:(Kazman, Klein y Clements, 2000).

Después de una evaluación con el método, a los planes de proyectos deben incorporarse acciones que implementen las estrategias de mitigación de riesgos (Bass et al., 2012).

2.1.11 Método por Prototipos

Este método como ya fue mencionado anteriormente se basa en la creación de prototipos con el único objetivo de tener un sistema final “estático” pero no funcional de lo que se pretende realizar. Los prototipos o también llamados experimentos buscan exponer que las ideas de los diseños son o no viables, permitiendo la adquisición de conocimiento que puede ser muy valioso sin tener que desarrollar el sistema por completo y si se los aplica de la manera correcta reducir el tiempo de desarrollo.

Este método también permite obtener conocimiento sobre tecnologías nuevas o desconocidas para el arquitecto, y observar así que su uso en la arquitectura será o no riesgoso.

Un prototipo es una solución de desarrollo rápido que permite afirmar o negar las suposiciones en las cuales se basa la arquitectura. Un experimento es el planteamiento de una situación práctica con el que se confirma o niega una hipótesis en la cual se fundamenta la arquitectura.

2.2 Movilidad Urbana

El Instituto Mexicano para la Competitividad menciona que la movilidad es un elemento fundamental para la competitividad de las ciudades, ya que determina la forma en que las personas se mueven para realizar sus actividades diarias (IMCO, 2018). La movilidad urbana es el conjunto de desplazamientos, de personas y mercancías, que se producen en una ciudad (*Movilidad Urbana • ESMARTCITY*, 2019). Estos desplazamientos son realizados caminando o en diferentes medios o sistemas de transporte: bicicleta, coche, autobús, metro, etc. Los problemas que genera la movilidad referente a pérdida en tiempo y contaminación medioambiental junto como el deseo de generar una movilidad urbana sostenible para mantener o mejorar la calidad de vida de las ciudades medias y grandes, son algunos de los principales retos para las ciudades (Lizárraga Mollinedo, 2006). Las Ciudades Inteligentes trabajan para resolver estos problemas con plataformas de gestión de tráfico y aplicaciones informáticas para facilitar la movilidad (*ESMARTCITY • Noticias Diarias sobre Ciudades Inteligentes*, 2017).

Existen estudios en los que se resalta la importancia del paisaje visible para los peatones, a la vez que se ha encontrado que la selección de la ruta de peatones y ciclistas está

influenciada por la continuidad visual del entorno (Manum & Nordström, 2013). Otros estudios realizados también han encontrado una influencia positiva de los elementos de diseño de la calle, como el ancho y la calidad de las aceras, la presencia de bancas y pasos peatonales señalizados con el número de peatones o el tiempo destinado a caminar (Alfonzo et al., 2008; Rodríguez et al., 2009).

En Cuenca (Ecuador), diversos estudios presentados en seminarios de la EMOV han sido encontrados que se enfocan en aspectos como: el porcentaje de la huella de carbono producido por diferentes ámbitos como la industria, transporte, etc. y de igual forma en parámetros de movilidad (Piedra, 2016). También han sido cubiertos otros temas adicionales como: la visualización, el análisis y modelación de información entorno a la movilidad teniendo en cuenta la huella de carbono producida (Velez, 2017); la planeación, implementación y operación de un sistema de bicicletas públicas que incentiva el uso de la bicicleta como medio alternativo de transporte (Alvarez, 2017).

Dicho todo esto, se puede inferir que la movilidad urbana es entonces un factor determinante que se debe tener en mente, tanto para la productividad económica de la ciudad como para la calidad de vida de sus ciudadanos y el acceso a servicios básicos de salud y educación.

2.2.1 LlactaLAB.- Antecedentes

LlactaLAB – Ciudades Sustentables es un grupo de investigación parte del Departamento Interdisciplinario de Espacio y Población de la Universidad de Cuenca que tiene el propósito de construir o generar conocimiento científico que aporte a la búsqueda de soluciones a los grandes desafíos de las ciudades en el siglo XXI (LlactaLAB, 2018). Entre

los principales desafíos se encuentran: el manejo de información, la salud y el bienestar de la población.

LlactaLAB se centra en una investigación interdisciplinaria con el propósito de aprender entre diversos profesionales, entender la ciudad como un sistema complejo y transformar nuestra realidad para mejorarla en el futuro. Sus líneas de investigación son: el espacio construido como en edificios, espacios públicos y/o barrios.

Además, LlactaLAB trabaja y se involucra en diversos proyectos de investigación relacionados con movilidad, entre ellos se encuentran: “Pies y Pedales”. Este proyecto se enfoca en realizar estudios de los Patrones de movilidad de ciclistas y peatones en Cuenca para una movilidad sustentable. El proyecto cuenta con una segunda parte denominada “Pies y Pedales 2”, en el cual tiene participación este proyecto de titulación.

2.2.2 Proyecto “Patrones de movilidad activa y entorno urbano: Pies y Pedales”

La movilidad es un componente clave de la sustentabilidad de las ciudades, y se ha demostrado que el desplazamiento en bicicleta y a pie resuelve innumerables problemas derivados del tráfico vehicular. (*Pies y Pedales – LlactaLAB – Universidad de Cuenca, 2018*).

El proyecto de investigación “Pies y Pedales 2” propone la construcción de un marco analítico multidisciplinario para el estudio de los patrones de movimiento a pie y en bicicleta y su relación con la movilidad sustentable. El proyecto se estructura en tres ejes principales que se visualizan en la Figura 2.3.

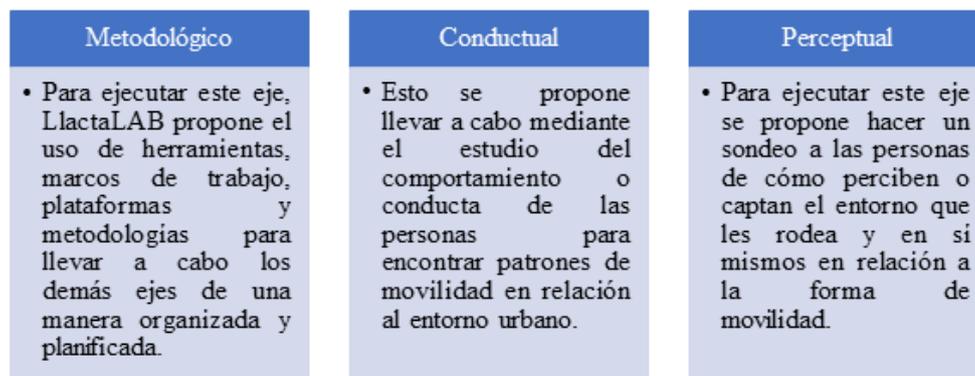


Figura 2.3 Ejes Principales de Pies y Pedales

Elaborado por: (Los Autores)

Fuente: (LlactaLAB, 2019)

Este proyecto técnico se enfoca en el primer eje, es decir, en el metodológico.

2.2.3 Geoinformática

La geoinformática es una de las ramas de la informática orientada a la recolección, análisis y procesamiento de información relacionada con la ubicación, relación e interacción de uno o más objetos sobre la tierra para resolver una problemática en particular (Buehler & McKee, 1996).

2.2.4 Geoprocesamiento

El geoprocesamiento consiste en una operación de tratamiento o manipulación de datos espaciales gracias al uso de sistemas de información geográfica (SIG o GIS, por sus siglas en inglés, *Geographic Information System*). El objetivo fundamental del geoprocesamiento es facilitar herramientas y un marco de trabajo para realizar análisis y administrar los datos geográficos. Por ejemplo, calcular las rutas óptimas a través de la red de transporte, analizar y buscar patrones en ubicaciones de delitos, predecir qué áreas son

propensas a derrumbes o predecir los efectos de una inundación (Argis, 2016; Buehler & McKee, 1996).

2.2.5 Sistemas de Información Geográfica

Un Sistema de Información Geográfica o SIG es considerado como una integración organizada de hardware conjuntamente con software. Los SIG a la vez incorporan datos geográficos con el fin de capturar, almacenar, manipular, analizar y desplegar la información referenciada geográficamente (Santovenia Díaz et al., 2009).

Existen dos formas de almacenar los datos en un SIG, tipo ráster y tipo vectorial. El de tipo ráster es cualquier tipo de imagen digital representada en mallas (Pixels), dividiendo el espacio en celdas regulares, donde cada una de ellas representa un único valor, El de tipo vectorial, en el cual, los datos están basados en la representación vectorial del componente espacial de los datos geográficos. Su representación es mediante puntos, líneas y polígonos.

El objetivo de tener los datos sobre un SIG es para el análisis, manipulación y para la toma de decisiones, ya que la integración de SIG con la web ha hecho que esté disponible de una manera fácil y eficiente para el público. Luego del surgimiento de la web en los 90's, se han acoplado o migrado los SIG sobre éste, naciendo así la SIGweb, que con el transcurso del tiempo lo han llevado a ser interactivo, de integración y transmisión en tiempo real, así como al acceso a herramientas de análisis SIG independientes de la plataforma. Después, la web 2.0 trajo el nacimiento de Google Maps, Google Earth y Microsoft Bing Map. Mientras que, en el año 2000, Refrations Research lanzó PostGIS, un soporte de código abierto para objetos espaciales en PostgreSQL. En la actualidad, con la creación de dispositivos como GPS, teléfonos inteligentes y redes sociales, han surgido nuevas fuentes de datos. Las mismas son manejadas por aficionados que se convierten en proveedores de datos espaciales, lo que se

conoce como Información Geográfica Voluntaria (IGV), entre los más populares se encuentran OpenStreetMap, Flickr y Wikimapia.

Existen herramientas de software que permiten manejar estos datos geoespaciales tales como QGIS (software libre) y ArcGIS (software privado). Los mencionados anteriormente son los más populares dentro del geoprocesamiento (Buehler & McKee, 1996). QGIS es un SIG de código libre para todas las plataformas. Esta herramienta maneja los dos formatos más populares: el ráster y el vectorial. Su característica sobresaliente es que tiene soporte para la extensión espacial de PostgreSQL, PostGIS. (QGIS, 2018). ArcGIS es un GIS de software propietario, el cual es un completo sistema que permite administrar, analizar, compartir y distribuir información geográfica. Esta aplicación se engloba en familias temáticas como: ArcGIS Server, para la publicación y gestión web; ArcGIS Desktop, que es la familia de aplicaciones SIG de escritorio, una de las más utilizadas (Esri, 2018).

2.2.6 Consorcio geoespacial abierto

El consorcio geoespacial abierto (OGC, por sus siglas en inglés, *Open Geospatial Consortium*) es una organización internacional sin fines de lucro comprometida con la creación de estándares abiertos de calidad para la comunidad geoespacial global. Los estándares de OGC se utilizan en una amplia variedad de dominios que incluyen: geociencias y medio ambiente, ciudades inteligentes, incluidas las webs de IoT y sensores, la tecnología móvil y el entorno 3D, servicios públicos y muchos más.

El consorcio de grupo de objetos OGC define los siguientes estándares para la gestión de datos geoespaciales: Web Mapping Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS), Catalog Service for the Web (CSW), entre otros (Welcome to The Open Geospatial Consortium | OGC, s. f.). Los estándares que actualmente se soportan y se mantienen por los procesos que se realizan en LlactaLAB son: WMS y WFS.

2.3 Estado del arte

En la presente sección se procede a indicar el estado del arte sobre las arquitecturas de software y las plataformas web existentes relacionadas con gestión de datos geográficos y movilidad siguiendo el protocolo de revisión definido por el “PRISMA extension for scoping reviews (PRISMA-ScR)”, según Tricco et al (2018).

Para identificar documentos potencialmente relevantes, se realizaron búsquedas bibliográficas contra las bases de datos de literatura científica Scopus y ACM Digital Library (Fink, 2014). La búsqueda se realiza contra el título, abstract y palabras clave, abarcando los últimos 13 años: desde 2007 hasta (diciembre) 2019. Aunque el término “Arquitectura de Software” se acuñó a inicios de los años 60 (Sommerville & others, 2011) y los sistemas SIGWeb a inicios del año 2000 (Ramsey, 2007), únicamente se consideran estos últimos años ya que lo que se busca son implementaciones o diseños que se han aplicado o realizado actualmente y en el año 2007 se encuentra un estudio referente al estado del arte sobre arquitecturas de software para movilidad, por lo que se decidió muy importante considerarlo por el aporte que este brinda al trabajo.

La búsqueda definitiva fue la siguiente:

```
TITLE-ABS-KEY ( ( "software architecture" ) AND ( "SIGWeb" OR "SIG" OR "mobility platform" OR "geographic" ) ) AND DOCTYPE ( ar OR cp ) AND PUBYEAR > 2006
```

2.3.1 Criterios de elegibilidad

Los criterios de elegibilidad (de inclusión) son los siguientes:

- Idioma: inglés.
- Tipo de documentos: peer-review journal papers and conference papers.

- Ventana temporal: los trece últimos años, desde 2007 hasta 2019.
- Documentos no duplicados
- Los documentos deben centrarse en el diseño de arquitecturas de software para movilidad, es decir con el uso de información geográfica y otros tipos de datos.
- Dado el anterior criterio, los documentos pueden tratar sobre aplicaciones específicas, diseños conceptuales, casos de estudio, estilos arquitectónicos, etc.

2.3.2 Resultados

La Tabla 2.8 muestra los resultados obtenidos en dos etapas: Etapa I (screening) documentos arrojados por la base de datos, Etapa II (elegibilidad) documentos seleccionados luego de la revisión del abstract.

Fuente bibliográfica	Etapa I (Screening)	Etapa II (Elegibilidad)
Scopus	29	4
ACM Digital Library	142	2
Total	171	6

Tabla 2.8 Resultados obtenidos de las fuentes bibliográficas

Elaborado por: (Los Autores)

De la misma manera en la Tabla 2.9 se exponen los artículos encontrados y considerados para este capítulo.

Nº	Artículo
1	Krämer, M., & Senner, I. (2015). A modular software architecture for processing of big geospatial data in the cloud. <i>Computers and Graphics (Pergamon)</i> , 49, 69-81. Scopus. https://doi.org/10.1016/j.cag.2015.02.005
2	Castanhari, R. E. S., Dos Santos Rocha, R., De Andrade, S. C., & De Albuquerque, J. P. (2016). A software architecture to integrate sensor data and volunteered: Geographic information for flood risk management. <i>Proceedings of the International ISCRAM</i>

	Conference. Scopus.
3	Olivera, H. V., & Holanda, M. (2012). A GIS Web with integration of sheet and soil databases of the Brazilian Cerrado. Iberian Conference on Information Systems and Technologies, CISTI. Scopus.
4	Melis, A., Mirri, S., Prandi, C., Prandini, M., & Salomoni, P. (2016). A microservice-based architecture for the development of accessible, crowdsensing-based mobility platforms. 498-505. Scopus. https://doi.org/10.1109/CTS.2016.91
5	Malek, S., Edwards, G., Brun, Y., Tajalli, H., Garcia, J., Krka, I., ... Sukhatme, G. S. (2010). An architecture-driven software mobility framework. <i>Journal of Systems and Software</i> , 83(6), 972–989. doi: 10.1016/j.jss.2009.11.003
6	Ramsey, P. (2007). The state of open source GIS. Refrations Research Inc, 1–49.

Tabla 2.9 Artículos obtenidos de las fuentes bibliográficas

Elaborado por: (Los Autores)

2.3.3 Descripción de los estudios incluidos

Krämer y Senner (2015) proponen una arquitectura de software que permite el procesamiento de grandes conjuntos de datos geoespaciales en la nube. Este proceso se realiza mediante técnicas o algoritmos de big data como Map-Reduce permitiendo la ejecución distribuida entre varios servidores sobre la nube, ofreciendo una interfaz de usuario sobre la web. Se propone la ejecución de trabajos mediante un administrador de trabajo, esto es considerado como el orquestador de los procesos. La manera de comunicación entre servidores es mediante un archivo distribuido, procesando y almacenando grandes cantidades de datos sobre la nube (Krämer & Senner, 2015). Al considerar que en la movilidad se podrían integrar datos recolectados por sensores y la información geográfica voluntaria IGV, Castanhari et al. (2016) proponen una solución al desafío de la integración de múltiples fuentes de datos heterogéneos con diferencias en sus estructuras y contenidos mediante una arquitectura de datos orientada a servicios. Esta arquitectura tiene como núcleo un modelo de componentes arquitectónicos y sus relaciones, en la que cada componente arquitectónico está

destinado a integrar una fuente de datos distinta y comunicándolos entre sí de una manera automatizada mediante algoritmos. Esto con el objetivo de brindar información útil que sirva para la toma de decisiones referente a inundaciones (Castanhari et al., 2016). De la misma manera, Olivera y Holanda (2012) proponen una arquitectura orientada a servicios para gestión de datos ambientales en los que es necesario la interoperabilidad, integración y el intercambio de datos relacionados sobre el uso del suelo usando el concepto de ontología para apoyar el problema de la heterogeneidad semántica en distintas bases de datos. Así mismo, Melis et al. (2016) proponen una arquitectura basada en micro-servicios para descubrir barreras urbanas que puedan impedir la fácil transitabilidad de personas con capacidades especiales, exponiendo cada parte del proceso como un microservicio y gestionando su ejecución mediante una orquestación de componentes. De manera más general, Malek (2010) considera que muchos sistemas enfocados al manejo de datos geoespaciales son complejos y distribuidos, pero la mayoría de los avances en este dominio no tienen un enfoque arquitectónico explícito del software. Según Ramsey (2007), no existe un enfoque arquitectónico más apropiado para el diseño o construcción de plataformas geográficas o sistemas de información geográficas web. En este trabajo se evalúan las fortalezas y ventajas de tres enfoques arquitectónicos propuestos que son: cliente-servidor, SOA y cloud computing, de los cuales, dependiendo el contexto de aplicación, recursos y necesidades de la entidad que lo requiere se aplicará uno u otro estilo arquitectónico, inclusive se propone el diseño híbrido entre estilos arquitectónicos, con el objetivo de satisfacer las necesidades de la entidad si así lo requiere. De esta manera se puede concluir que el uso de servicios en una arquitectura de software con un enfoque de movilidad ofrece ventajas como escalabilidad, interoperabilidad e integración de fuentes de datos heterogéneos y que la aplicación de varios estilos arquitectónicos es posible si fuera necesario.

Al contar con muy pocos trabajos relacionados al tema, se ha considerado importante también tomar en cuenta plataformas funcionales que están desplegadas sobre el internet y que servirán de base para la generación de la arquitectura a nivel de diseño de interfaz gráfica de usuario o GUI. Dicho esto, se han identificado plataformas que tienen funcionalidades y características en común a la plataforma esperada, las mismas se detallan en la Tabla 2.10

Plataforma	País	Descripción
Hexaba ¹	Argentina	Es un módulo experimental de <i>data science</i> y urbanismo, que está escrito en R, y que cuenta con diversas funcionalidades como son las de visualización por capas, <i>scoring</i> urbano para la generación de índices mediante variables, análisis de regresión mediante selección de variables a explorar, clusterización con posibilidad de determinar el número de <i>clusters</i> a generar y finalmente un laboratorio de variables en donde se permite combinar variables previamente almacenadas para su posterior uso. Es la que más se apega a los requisitos del actual proyecto.
Autotrafic ²	México	Este sistema provee una interfaz simple de acceso rápido a las diferentes capas o servicios ofrecidos por parte de la comunidad que gestiona la información de movilidad de la ciudad expuesta sobre la web.
Cidade Plena ³	Brasil	Está compuesta por seis ámbitos que proporcionan datos sobre la compacidad, uso del suelo, cohesión social, habitabilidad, soporte ambiental e infraestructura del municipio brasileño de Lajeado en el estado de Rio Grande del Sur en Brasil

Tabla 2.10 Plataformas consideradas para el diseño arquitectónico

Elaborado por: (Los Autores)

¹ <https://nahuelpatino.shinyapps.io/hexaBA/>

² <https://autotraffic.com.mx>

³ <http://cidadeplena.com.br/lajeado/#COS01>

Capítulo III

3. Marco Metodológico

En este capítulo se da a conocer la aplicación del enfoque de investigación seleccionado para este proyecto y una descripción de cómo se relaciona la metodología de investigación con la metodología de desarrollo de la arquitectura a proponer. Así como también el desarrollo, las técnicas y los instrumentos fueron utilizados para llevar a cabo la indagación. La Sección 3.1 define, principalmente, las fases de la metodología seleccionada, una descripción de cada fase y cuál es su relación con la metodología de diseño de la arquitectura. La Sección 3.2 expone una descripción de cómo fue concebido el software. La Sección 3.3 presenta el diseño de la Arquitectura y el núcleo del sistema, de tal forma que se vean reflejados los requerimientos establecidos. La Sección 3.4 presenta el documento resultante de la arquitectura con su respectiva explicación. La Sección 3.5 presenta los resultados de la evaluación de la arquitectura generada mediante el método de análisis de la compensación de arquitectura (ATAM). Finalmente, la Sección 3.6 menciona el refinamiento de la arquitectura como mejora del documento final.

3.1 Materiales y Métodos

Esta sección describe la aplicación del enfoque de investigación expuesto en la Sección 1.6 dentro del grupo de investigación LactaLAB.

La Figura 3.1 muestra como las fases de la metodología de investigación fueron aplicadas en el desarrollo de este proyecto de titulación.

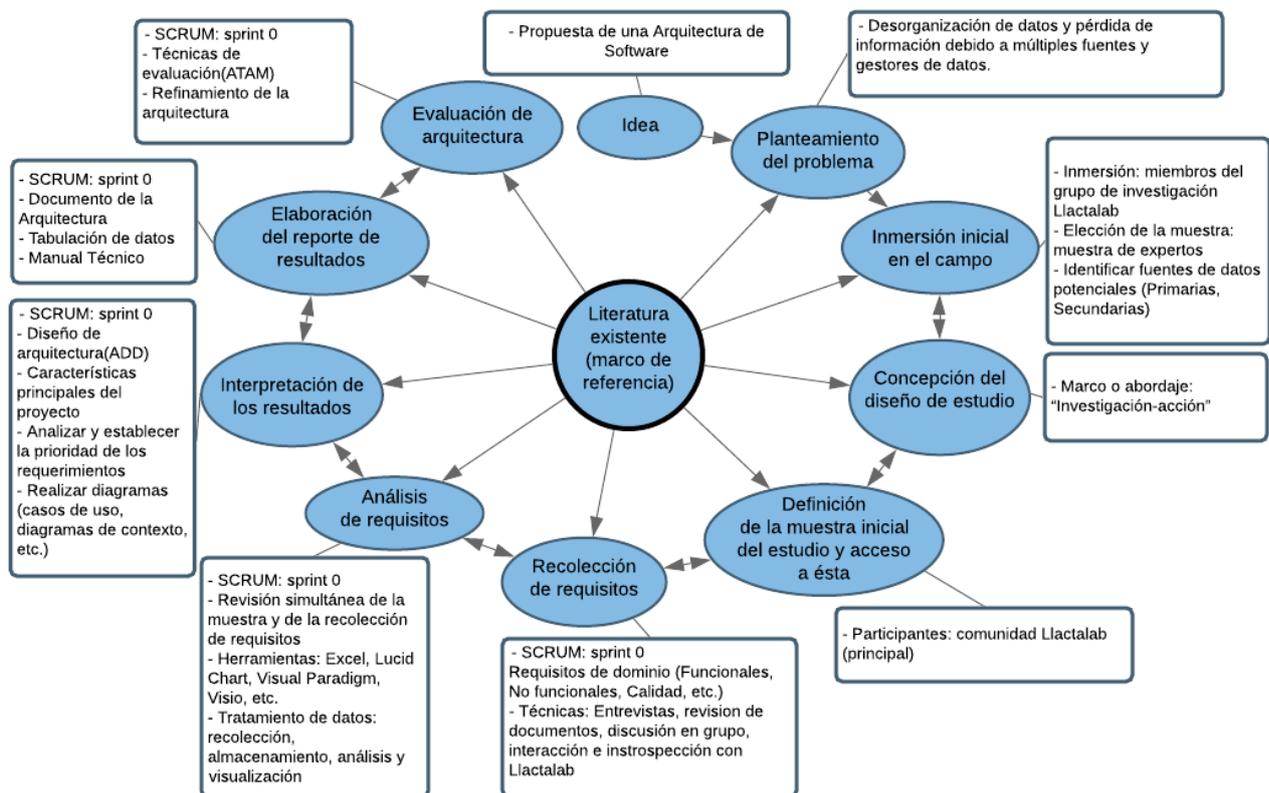


Figura 3.1 Metodología de la investigación para el diseño y evaluación de la arquitectura

Fuente: (Hernández Sampieri et al., 2006)

La idea para este trabajo es la de plantear un diseño de arquitectura que permita solventar las necesidades del grupo de movilidad LactaLAB. Esto, basándonos en la problemática actual que tiene la entidad, que hace referencia a la gestión adecuada de sus

recursos, los mismos que sirven para el análisis y toma de decisiones de manera multidisciplinar para dar solución a diferentes problemáticas relacionadas con los vehículos motorizados. Se considera que esta entidad podría tener diversas fuentes de datos heterogéneas como el Municipio, EMOV, experimentos, etc., así como también que los principales participantes del proyecto son los miembros del grupo de movilidad, se plantea la recolección, análisis de requisitos, diseño y evaluación de la arquitectura basándose en las etapas de la metodología de desarrollo ágil Scrum con una *Iteración 0*. La misma sirve como antecesora de las demás *Iteraciones* tradicionales del marco de trabajo, dando como resultado la generación de un sistema esqueleto o diseño de arquitectura. Todo esto con el uso de metodologías que permiten estandarizar el proceso a seguir, tales como ADD para el diseño de la arquitectura y ATAM para la evaluación de la misma.

3.1.1 Idea

El motor fundamental para generar nuestra idea de investigación es la necesidad de resolver la problemática de construir una arquitectura de software para la gestión, manipulación, análisis y visualización de los datos relacionados con patrones de movilidad del equipo de LlactaLAB. Para esto se realizó una investigación sobre estilos arquitectónicos usados y plataformas web para presentación de información geográfica, mejores técnicas para la construcción de arquitecturas cuando no se es experto en el ambiente de negocio.

3.1.2 Planteamiento del problema

El propósito de este estudio de investigación-acción con un enfoque participativo es diseñar y describir una arquitectura de software para una plataforma de movilidad dentro del entorno urbano de la ciudad de Cuenca. Como instrumento de recolección de los datos se utilizaron principalmente entrevistas a expertos haciendo uso de anotaciones o grabaciones,

motores de búsqueda como ACM Digital Library o Google Académico, entre otros, que tienen relación con la comunidad informática. La arquitectura planteada para el ambiente que se trató puede definirse en un término genérico como una arquitectura para un sistema de información geográfica, que debe acoplarse a una plataforma de movilidad propuesta por la comunidad LlactaLAB de la Universidad de Cuenca dentro del proyecto Pies y Pedales. Este proyecto se enfoca en el estudio de los patrones de movimiento a pie y en bicicleta y su relación con el espacio público para la movilidad sustentable.

3.1.3 Inmersión inicial en el campo

Como inmersión inicial se ha decidido tener en cuenta a los miembros del grupo de investigación LlactaLAB, ya que para ellos se enfoca y se realiza este trabajo y, posteriormente, por la Universidad de Cuenca específicamente en estudiantes o docentes que obviamente hagan uso de algún medio no motorizado para desplazarse de un punto a otro.

Según Hernández Sampieri et al. (2006), en una investigación cualitativa existen cuatro principales tipos de muestra, estos son: de voluntarios, de expertos, de casos-tipo, por cuotas. Para este trabajo se eligió una muestra de expertos y de casos-tipo para consultar sus opiniones en temas como arquitecturas para gestión de datos y sistemas de información geográfica, herramientas y tecnologías adecuadas usadas dentro de este contexto, así como un acercamiento a los datos que actualmente se gestionan dentro del grupo de investigación. Las fuentes de datos que se han logrado identificar, actuales y potenciales, tanto primarias (P) como secundarias (S) se detallan en la Tabla 3.1.

N°	Set de datos	Tipo Fuente	Tipo archivo	Geometría	Fuente
1	Aforos peatonales en	S	Shape, Tabla	Punto	Plan de movilidad y

	tramos				espacios públicos
2	Mapeo de características de tramos	P, S	Shape	Línea	Plan de movilidad y espacios públicos, levantamiento
3	Red vial peatonal, Ciclistica	S	Shape	Línea	Open Street Map
4	Usos de suelo y servicios	P, S	Shape	Polígono	Levantamiento, Municipio
5	Rastreo GPS (Mov. Natural)	P	Shape	GPX, Puntos	Experimento
6	Experimento controlado	P	Shape	GPX	Experimento
7	Datos STRAVA	S	Shape, Tabla	Línea, Punto	STRAVA
8	Camaras central semafórica	S	Video, Tablas, Shape	Punto	Semaforización movilidad
9	Encuesta Origen-Destino Municipio	S	Tabla, Shape	Punto	Municipio de Cuenca - LlactaLAB
10	Propuesta ciclovías	S	Shape	Líneas	Plan de movilidad y espacios públicos
11	Encuesta Movilidad	P, S	Tabla	Punto	LlactaLAB
12	Base de datos Estudiante Universidad de Cuenca	S	Tabla	N/A	Universidad de Cuenca
13	Datos Twitter	S	Shape, Tabla	Punto	Twitter
14	Grupos Focales	P	Archivos	N/A	Levantamiento
15	Observación en pasos cebra	P	Videos, Tablas, Shape	Punto	Levantamiento
16	Características pasos cebra	P	Tablas, Shape	Punto	Levantamiento

Tabla 3.1 Descripción de fuentes de datos

Elaborado por: (Los Autores)

Fuente: (LlactaLAB, 2019)

El grupo de investigación LlactaLAB se encuentra localizado en las instalaciones de la Universidad de Cuenca específicamente en el campus central y es totalmente accesible para cualquier consulta o gestión de información para la investigación.

3.1.4 Concepción del diseño de estudio

Para este diseño, el marco a utilizar en esta investigación, se eligió un diseño de “Investigación-Acción”. Hernández Sampieri et al. (2006) recomiendan elegir el abordaje de “Investigación-acción” cuando: la disciplina en donde se la va usar es la Ingeniería; el objeto de estudio es una problemática de un grupo como en este caso el de LlacataLAB; la estrategia de análisis de datos es involucrar a la comunidad en las decisiones sobre cómo analizar los datos y el análisis mismo; o, el reporte o producto será un programa o proyecto para resolver un problema. Dicho esto, se puede decir que se encaja perfectamente dichas características en el ambiente tratado.

3.1.5 Definición de la muestra inicial del estudio y acceso a ésta

Para este proyecto, se considera a LlacataLAB como la principal unidad inmersa en la investigación, dado que sus investigadores y colaboradores están directamente relacionados con el proyecto y es para quienes se propondrá el diseño de la arquitectura de software.

3.1.6 Recolección de los requerimientos

La Figura 3.2 presenta la metodología de diseño de la arquitectura que se utilizó. La idea principal como ya se mencionó anteriormente es la de adicionar una *Iteración* inicial llamada “*Iteración 0*” al inicio del ciclo de desarrollo Scrum. El objetivo principal fue que el arquitecto en la *Iteración 0* analice y diseñe la generalidad del sistema, llegando así a satisfacer los requisitos y de esta forma sea entendible por los miembros del equipo desde sus diferentes puntos de vista durante el desarrollo. Un punto clave, es reutilizar artefactos de software creados a partir de la arquitectura para ser más ágiles en el desarrollo de productos específicos. En esta figura se puede observar que la *Iteración 0* comprende tres fases que fueron tomadas del ciclo de vida de entregas evolutivas (Bass et al., 2012). En la *Iteración 0*

se construye la arquitectura de forma iterativa mediante un análisis preliminar de los drivers arquitectónicos, es decir, los requisitos funcionales principales, de calidad y del negocio; y de un estudio de factibilidad del proyecto. No se necesita tener todos los requisitos claros para comenzar la fase de diseño arquitectónico dado que todos estos se aclaran a medida que nos introducimos en el ambiente a tratar.

Para determinar los conductores arquitectónicos, se deben identificar los objetivos del negocio de más alta prioridad, que son pocos. Estos objetivos del negocio se convierten en los escenarios de calidad o casos de uso. De esta lista, se deben escoger los que tendrán un mayor impacto sobre la arquitectura. El diseño arquitectónico puede comenzar una vez que se encuentran definidos los conductores arquitectónicos. El proceso de análisis de requisitos será entonces influenciado por las preguntas generadas durante el diseño arquitectónico («El Papel de la Arquitectura de Software en Scrum», 2015).

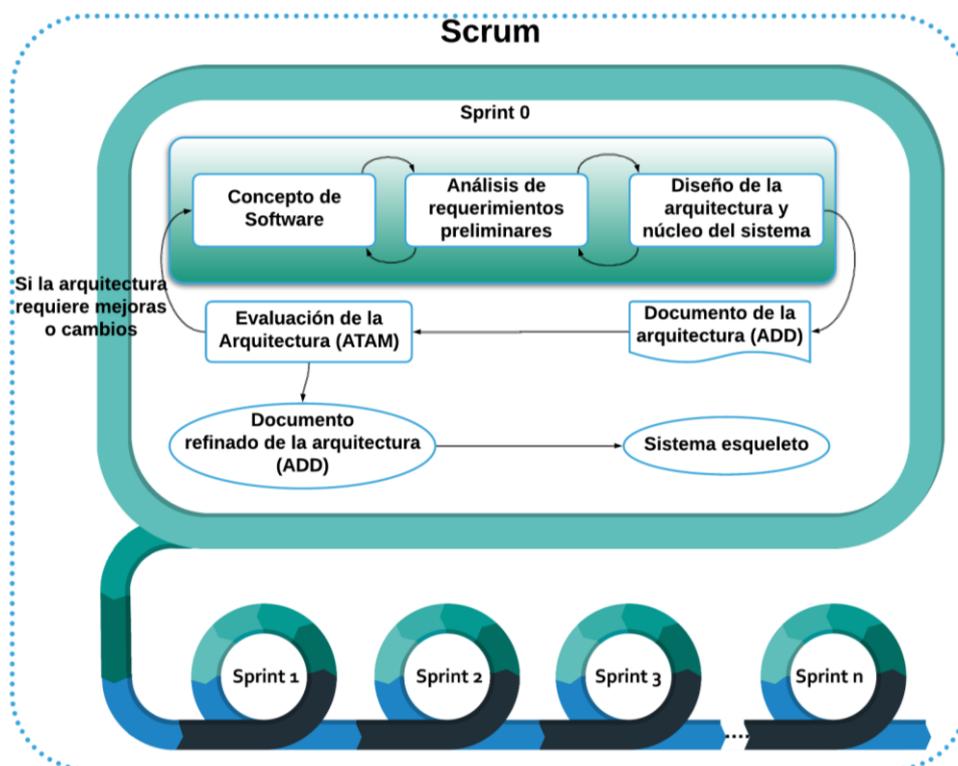


Figura 3.2 Gestión de la arquitectura de software en Scrum

Fuente: (El Papel de la Arquitectura de Software en Scrum, 2015)

Para esta fase se recolectó información de requerimientos tanto funcionales, como no funcionales, de calidad, tipo de archivos o datos a utilizar, restricciones, entre otros, que se ven necesarios y están reflejados a la hora del desarrollo de la arquitectura.

Se utilizaron además técnicas para recolectar requerimientos como: revisión de documentos, discusión en grupo, evaluación de experiencias personales e interacción e introspección con LlactaLAB (Hernández Sampieri et al., 2006). Todo este proceso se cumplió mediante reuniones en su mayoría quincenales con la presencia de: directores del proyecto LlactaLAB o de representantes del proyecto encargados de solventar inquietudes, dirigir y/o guiar el desarrollo; tutores dirigentes de proyecto que supervisaron y verificaron los avances; y estudiantes encargados del diseño de la arquitectura.

Como recomienda Hernández Sampieri et al. (2006), todo esto se realizó con ayuda de anotaciones, bitácoras, mapas, fotografías, diferentes medios audiovisuales, consultas directas en reuniones con los expertos, en este caso expertos de movilidad (preguntas e inquietudes), así como también de la parte técnica.

Finalmente, se considera importante mencionar que se utilizó información generada por el grupo de investigación LlactaLAB en la primera etapa del proyecto Pies y Pedales, que consta de documentos esenciales que pueden ayudar a comprender mejor lo que se requiere realizar. Entre ellos está la propuesta del proyecto de investigación ganadora del XVII concurso universitario de proyectos de investigación titulada “Patrones de movilidad activa y entorno urbano: Pies y Pedales 2” que fue considerado primordial para el proceso de extracción de requerimientos en donde se detalla la mayoría de las necesidades solicitadas y

planteadas en esta propuesta. Los detalles resultantes de todo este proceso se detallan en el Manual Técnico-Sección 1.

3.1.7 Análisis de los requerimientos

Para esta fase se van a analizar los requisitos de usuario definidos por LactaLAB, que serán transformados a requisitos funcionales, los mismos que se especifican basándose en la plantilla propuesta por IEEE para especificación de requisitos de software, estos detalles se encuentran descritos en el Manual Técnico-Sección 1. El objetivo es aclarar hacia dónde se va a llegar y además definir lo que se quiere lograr para la arquitectura de software, todo esto con la ayuda de herramientas como hojas de cálculo, procesadores de texto y Lucidchart. Esta fase permitió definir algunas características principales del proyecto, como por ejemplo, el tener archivos que representan información geográfica en diferentes formatos como: .shp, .kml, .geojson, fueron analizados y sirvieron para establecer qué tipos de formatos debe soportar la arquitectura para la gestión de los datos. En esta etapa también se analizó y estableció la prioridad de la información y de los requerimientos que fueron recolectados junto con los interesados, se establecieron los drivers arquitectónicos, se realizaron las tablas o gráficas acorde a las necesidades establecidas (casos de uso, diagramas de contexto, diagramas de componentes, etc).

Las necesidades de usuario se descomponen en: requisitos funcionales, no funcionales y restricciones del sistema. En esta sección se detallan también los múltiples casos de uso que se pueden dar en el sistema, sus dependencias y relaciones, así como los actores relacionados con los casos de uso, en la Tabla 3.2 se describe brevemente cada actor, dando una idea de las funcionalidades asociadas a este.

Actor	Descripción
-------	-------------

Administrador	Representa al administrador total; es el usuario que, aparte de acceder al sistema con un nombre de usuario y contraseña, tiene la potestad de gestionar miembros, proyectos, roles, privilegios, es decir, tiene permisos y privilegios para realizar absolutamente todo.
Miembro	Representa al usuario que accede al sistema mediante un usuario y contraseña provista por el administrador total del sistema. Este usuario tendrá ciertos roles o privilegios dependiente como el Administrador se los asigne.
Usuario	Representa al usuario que puede acceder al sistema mediante la url de la página y sin necesidad de estar registrado o de haber ingresado mediante un usuario y contraseña al sistema (considerado solo un visitante), este usuario en su totalidad solo tendrá acceso al visor de capas y algunas funcionalidades básicas del sistema.

Tabla 3.2 Actores del sistema identificados

Elaborado por: (Los Autores)

Hay que tener en cuenta la jerarquía de cada actor, ya que los casos de uso o funcionalidades del sistema se heredarán de un actor a otro dependiendo la herencia que tenga.

La Figura 3.3 presenta la metodología que solventará el tratamiento (recolección, almacenamiento, análisis y visualización) de datos sobre patrones de movilidad.

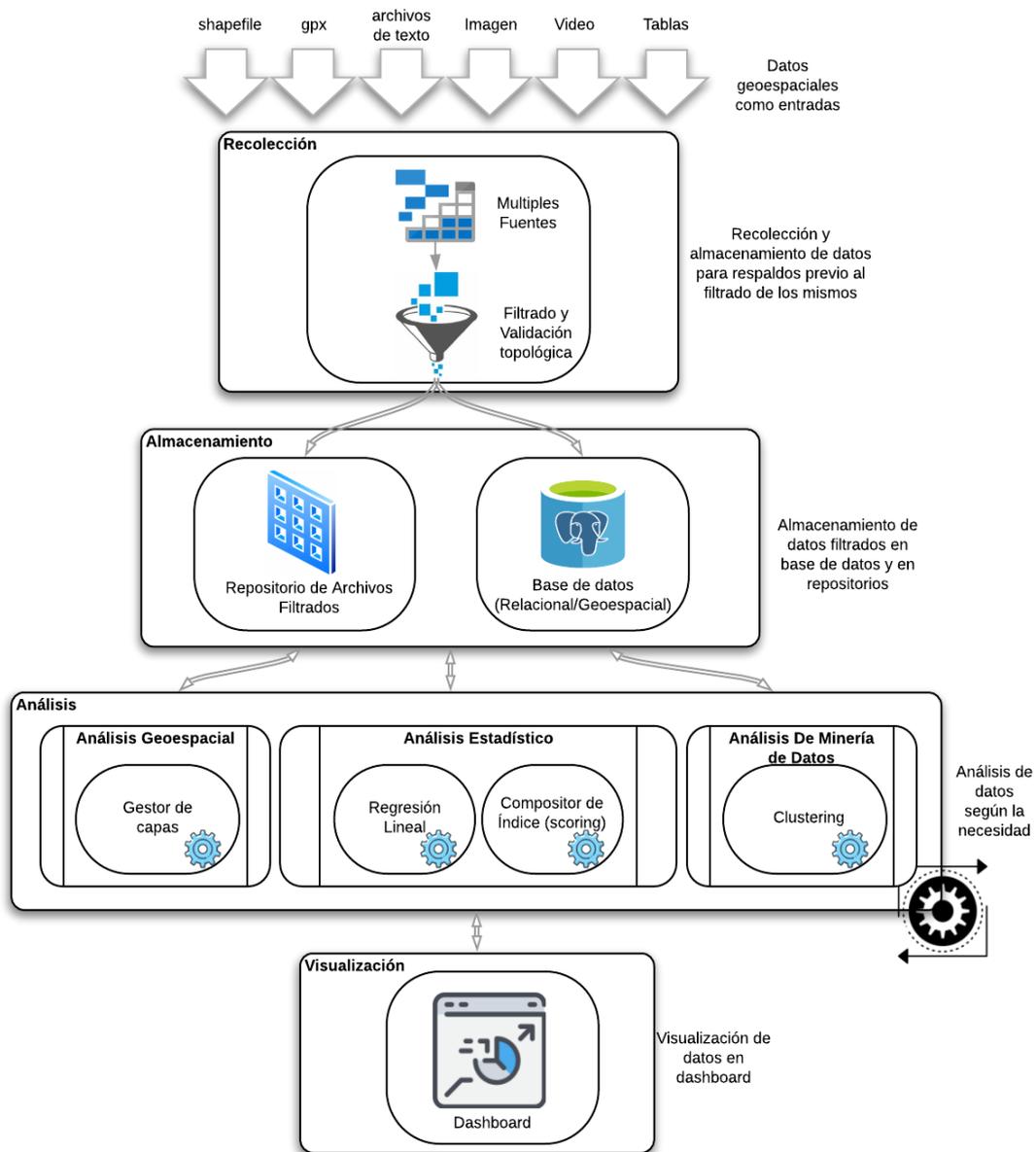


Figura 3.3 Proceso de recolección, almacenamiento, análisis y visualización de datos

Elaborado por: (Los Autores)

El proceso de la Figura 3.3 se compone de cuatro etapas: 1) la etapa de recolección que se enfoca en la recopilación de datos en múltiples formatos (shapefile, gpx, tablas, videos y archivos) provenientes de diferentes fuentes y, finalmente, estos son recolectados en una estructura de directorios para su futuro filtrado; 2) la etapa de almacenamiento, en donde se realiza el depósito de datos filtrados y validados en un repositorio específico y además de ello

se registran los datos geospaciales en una base de datos geoespacial, esto con el fin de reutilizarlos más eficientemente; 3) la etapa de análisis que se centra en realizar mediante el uso de funciones específicas el análisis geoespacial, estadístico o la minería de datos; y 4) la etapa de visualización que representa el paso posterior a la realización de algún análisis, es decir, la visualización en un dashboard presentando los resultados para la toma de decisiones. Los detalles de esta fase se encuentran descritos en el Manual Técnico-Sección 1.10.

3.2 Concepto de software

La concepción del software que se diseñó es de tipo geográfico, es decir un diseño para un Sistema de Información Geográfica sobre la web o SIGweb, que permite la toma de decisiones de manera multidisciplinaria.

3.3 Diseño de la arquitectura y núcleo del sistema

En el Manual Técnico-Sección 2: “Diseño para la Arquitectura de Software” se detallan los componentes involucrados en el sistema para satisfacer los requisitos identificados en la etapa anterior, así como su interacción y responsabilidades asignadas, todo esto mediante el uso de patrones para brindar atributos como robustez, disponibilidad, portabilidad y usabilidad al sistema. En la Figura 3.4 se observan los componentes considerados para la arquitectura propuesta, así como los enlaces que detallan la comunicación con los demás.

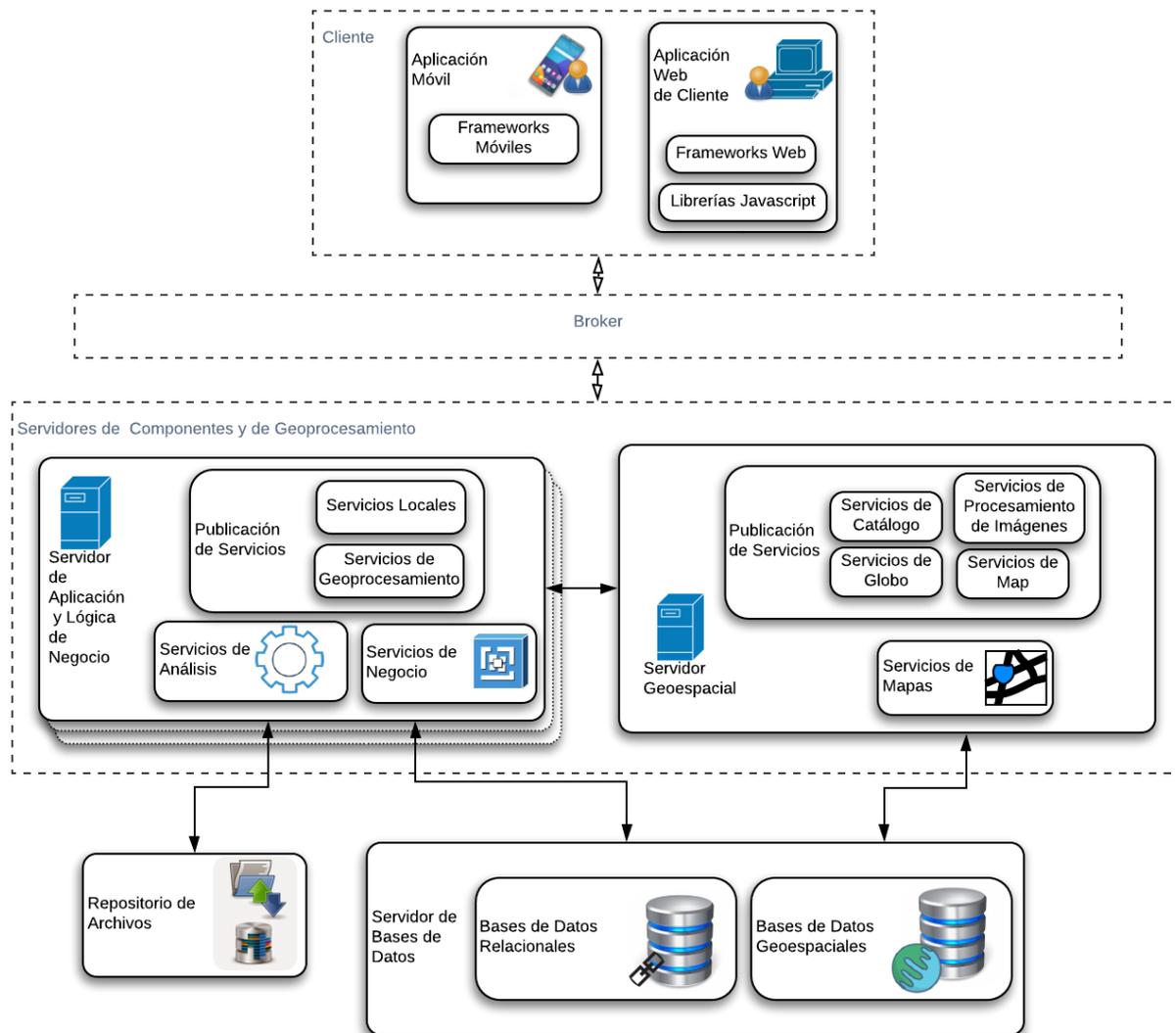


Figura 3.4 Modelo de arquitectura del sistema propuesto

Elaborado por: (Los Autores)

En el modelo propuesto se pueden observar las partes que lo componen como: servidores de aplicaciones, servidor de mapas, servidor de base de datos, etc., estos deben cumplir su responsabilidad de manera adecuada para que el sistema funcione correctamente como un todo y pueda cumplir las necesidades de LlactaLAB.

3.4 Documento de la Arquitectura

El Documento de Arquitectura para este trabajo es representado por el Manual Técnico presentado, el mismo que le sirve al desarrollador o equipo de desarrollo como guía y para saber cómo se debe implementar el sistema.

3.5 Evaluación de la Arquitectura

Tras obtener el Documento de Arquitectura con todas las especificaciones técnicas de diseño se procedió a evaluar la misma, esto mediante el método ATAM. En el Manual Técnico-Sección 3: “Validación de la Arquitectura de Software” de manera superficial se comprueba que la arquitectura efectivamente cumple con algunos atributos como interoperabilidad, usabilidad, escalabilidad, etc. Sin embargo esto no garantiza el cumplimiento de los atributos descritos. Además de esto, presenta ciertos riesgos (R) que deben considerarse al momento de implementar esta arquitectura para poder dar un seguimiento y tratar de mitigar los mismos, así como los no riesgos (NR) o fortalezas por los que la plataforma se identifica.

Riesgos	Descripción
R1	No se puede saber la disponibilidad exacta de servicios externos dado que no están al alcance del personal interno.
R2	El número de consultas y peticiones crezca incontrolablemente sobrecargando al servidor.
R3	Servidor intermediario (Broker) podría tender a convertirse en un cuello de botella.
R4	Posibilidad de aumentar el tiempo de respuesta.
R5	Caída temporal de servidor de capas.

*Tabla 3.3 Identificación de riesgos de la arquitectura**Elaborado por: (Los Autores)*

Para la mitigación de los posibles riesgos es necesario que se consideren ciertos protocolos o acciones periódicas, así como un plan de contingencia para tratar de controlar al máximo estos incidentes, como el considerar necesario hacer respaldos de la información cada cierto tiempo mediante mecanismos manuales o automáticos en horarios que no afecten el uso del servicio, tener un centro de procesamiento alternativo para dar continuidad del servicio en caso de caída de los servidores principales, considerar el uso de un balanceador de carga para no saturar el servidor y distribuir uniformemente la carga a los demás evitando así la sobrecarga del servidor, etc. La planificación, análisis y ejecución de estas acciones dependerán de las políticas, normas o estándares de la entidad en caso de tenerlas.

No-Riesgos	Descripción
NR1	Escalabilidad dada la separación de nodos.
NR2	Mantenibilidad dada la separación por capas en el servidor principal.
NR3	Interoperabilidad y usabilidad de cierta manera dado que la interfaz del cliente está separada del servidor principal.
NR4	Seguridad ya que se controla la ejecución de procesos, mediante la incorporación del patrón proxy para el control de usuarios.
NR5	Rendimiento ya que los servidores se dedican exclusivamente a ciertas tareas específicas.
NR6	Mantenibilidad dado que se provee una documentación detallada de la arquitectura generada y, además, se genera una instancia global que se facilita a las clases mediante el uso del patrón singleton, permitiendo reusabilidad de instancias.

*Tabla 3.4 Identificación de no riesgos de la arquitectura**Elaborado por: (Los Autores)*



La arquitectura se caracteriza por tener los atributos mencionados en la Tabla 3.4, sin embargo, están propensas a sufrir afectaciones en el transcurso del tiempo ya que como se sabe los avances tecnológicos son cada vez mayores y dejan en la obsolescencia tanto software como hardware, lo que podría intervenir o afectar colateralmente a las fortalezas identificadas.

3.6 Documento refinado de la arquitectura

Una vez obtenida y evaluada la arquitectura propuesta, el Documento de Arquitectura podrá ser refinado considerando los resultados de la evaluación, obteniendo así una versión mejorada. Por el alcance del trabajo de titulación no se ha considerado realizar muchas *Iteraciones* para refinamiento de la arquitectura, pero lo recomendable es que sea refinado hasta considerar que los resultados de la evaluación son los apropiados y óptimos. Por lo tanto, el documento refinado para este trabajo se considera el Documento de Arquitectura en su versión inicial.

Capítulo IV

En este último capítulo se darán a conocer las conclusiones finales que se han obtenido en base a los objetivos que fueron planteados al inicio del proyecto, las aportaciones que se han logrado con el trabajo realizado, las recomendaciones propuestas y finalmente los trabajos que se podrían efectuar a futuro.

4. Conclusiones

La movilidad es una importante disciplina que permite que las ciudades puedan convertirse en sustentables frente a los diferentes desafíos del siglo XXI, tales como la problemática del tráfico derivado de los vehículos motorizados, la correcta utilización de las calles y aceras, el respeto a peatones y ciclistas, entre otros. De forma particular, el grupo de movilidad LlactaLAB considera el desplazamiento a pie o bicicleta como una solución sana y amigable con el medio ambiente y para ello se deben realizar diferentes análisis y estudios multidisciplinarios que permitan incorporar estas propuestas en la ciudad. El proyecto técnico realizado, se ha centrado en proponer una arquitectura de software para una plataforma de análisis de datos sobre movilidad. La arquitectura propuesta es factible, viable y escalable de acuerdo a los recursos actuales con los que cuenta LlactaLAB debido a que el proyecto requiere la implementación de la misma de forma inmediata. También se establecieron recomendaciones para la implementación como son lenguajes de programación, capacidad en cuanto a hardware para un desempeño adecuado, patrones arquitectónicos que otorgan beneficios a la arquitectura, patrones de diseño que en cierto grado proveen un desarrollo elegante y organizado, o frameworks de desarrollo que facilitan la vida al desarrollador. Además, se hace énfasis en posibles riesgos que pudiesen emerger en el transcurso del tiempo

y que deberán ser considerados, en el cual, no solo el desarrollador deberá tener en cuenta sino también los interesados en el sistema.

En el desarrollo de este trabajo se han alcanzado los objetivos inicialmente planteados en cuanto a:

- **Diseñar una arquitectura de software para una plataforma de análisis de datos sobre movilidad.**

En cuanto a metodología de desarrollo del diseño se identificó a Scrum con una *Iteración 0* (Antecesor de las demás *Iteraciones*, considerado para diseño de Arquitecturas de Software) como la metodología idónea para la entidad, ya que requieren generar resultados de manera ágil y simple. En cuanto a la ERS, ésta se ha estructurado basándose en el estándar IEEE 830 (Manual Técnico-Sección 1). Asimismo, para el diseño de la arquitectura se identificó a ADD como la metodología de diseño idónea (Manual Técnico-Sección 2) y, finalmente, ATAM como la metodología de evaluación (Manual Técnico-Sección 3), resaltando los riesgos y no riesgos resultantes de la evaluación. Se han identificado patrones que soportan los requisitos y restricciones necesarias que LactaLAB requiere para un correcto funcionamiento del sistema, así como también considerando el uso del patrón n-niveles para la estructuración de los componentes del sistema, ya que este permitirá un fácil reemplazo y/o mantenimiento de algún componente localizado en el n-nivel, a nivel de comunicación se decidió utilizar el estilo arquitectónico orientado a servicios, ya que por medio de interfaces provistas se logran generar componentes débilmente acoplados, y permitiendo una alta escalabilidad. Esta propuesta arquitectónica se puede observar en el Figura 1.2 del Manual Técnico-Sección 1.

- **Construir un marco teórico que proporcione un conocimiento fuerte para el desarrollo, así como un estado del arte de arquitecturas realizadas o propuestas para plataformas de movilidad.**

El marco teórico se organizó en dos grandes bloques: Arquitectura de Software y Movilidad, en los que se analizaron los conceptos más importantes dentro de cada uno, de tal manera de llegar a obtener una base de conocimiento robusta para la elaboración del proyecto. De igual manera, se realizó una búsqueda de estudios que aborden el tema expuesto, sin embargo, no se ha encontrado uno que responda directamente a los objetivos planteados en este proyecto técnico. Cabe recalcar que, muchos de los trabajos encontrados en las fuentes estaban relacionados a la movilidad de una arquitectura de software como característica o atributo deseable, más no se refiere al hecho de un diseño o enfoque arquitectónico para la movilidad desde el ámbito geoespacial del que se aborda en este trabajo, por lo que se considera casi nulo el abordaje de este tema en los últimos años.

- **Generar una propuesta de arquitectura de software para la plataforma de gestión y análisis de datos de movilidad.**

De la bibliografía y necesidades planteadas por LlactaLAB se concluye que, el mejor estilo de arquitectura enfocado a la movilidad es el orientado a servicios por los principios inherentes que implica el uso del mismo y que, a nivel de diseño de plataformas web, de los tres sitios web que gestionan información geográfica encontrados, el mejor modelo o plataforma usada como guía a nivel de interfaz web es Hexaba considerando como el más idóneo e importante según los requerimientos de LlactaLAB, el mismo que sirvió de guía o base para el diseño de las interfaces gráficas de usuario.

- **Identificar las herramientas o tecnologías de software que mejor se adapten a la arquitectura.**

Estas tecnologías fueron identificadas en la Tabla 2.1 del Manual Técnico-Sección 2, las mismas fueron propuestas dependiendo de la funcionalidad y contexto de cada componente. En cuanto a nivel de modelo de datos se ha propuesto realizarla con tecnología de base de datos relacional para mantener la consistencia de los datos de los usuarios, ya que se tendrá que gestionar proyectos y recursos asociados a los mismos. El modelo proyecto, así como su respectivo diccionario de datos se pueden observar en el Manual Técnico-Sección 1.19: “Modelo del Sistema”. Además, se debe tener en cuenta que la plataforma se va a desplegar sobre el centro de datos de CEDIA, que cuenta con la infraestructura necesaria para soportar el correcto funcionamiento de la futura implementación. Además, las tecnologías, patrones y metodologías propuestas en este trabajo son soportadas por CEDIA, haciendo de este trabajo un proyecto viable y proponiendo un modelo arquitectónico que se pueda desarrollar o implementar.

- **Validar la arquitectura de software propuesta.**

De la bibliografía considerada para este trabajo, se encontraron diversas metodologías de evaluación de arquitecturas de software, de lo cual se concluye que la que mejor se acopla a las necesidades y contexto del diseño es la ATAM, adicionalmente se ha incorporado prototipos como complemento para la validación. Finalmente, de la validación de la arquitectura se concluye que la propuesta de diseño cumple con las necesidades planteadas por LactaLAB, logrando también identificar sus riesgos y no riesgos como resultados de la evaluación, cabe recalcar que esta evaluación es a modo superficial por el alcance del trabajo, por lo tanto no se garantiza los atributos de calidad descritos. Esto se puede observar con detalle en el Manual Técnico-Sección 3.

4.1 Aportaciones de la tesis

La propuesta arquitectónica generada en el presente trabajo es una contribución de gran importancia dentro de la movilidad urbana, ya que servirá como modelo para futuros trabajos, investigaciones o implementaciones, no solo del grupo de movilidad LlactaLAB sino para otros grupos. Esta propuesta puede ser mejorada y acoplada a las necesidades de la entidad que lo requiera, independientemente de las tecnologías que se usen, esto mediante la interoperabilidad y escalabilidad que ofrece la misma.

La utilización de la presente propuesta permitirá la reducción de costos, tiempo y esfuerzo en la implementación final de la plataforma de movilidad activa y entorno urbano de LlactaLAB.

Una aportación muy importante es el diseño de base de datos y el proceso de recolección, análisis y visualización de datos propuesto, ya que son la base de la arquitectura de software y por ende soportará los requisitos de LlactaLAB. Así como la interoperabilidad y escalabilidad ofrecida mediante el estilo arquitectónico orientado a servicios de manera que exista la posibilidad de migrar componentes a Cloud Computing cuando se cuente con los recursos necesarios.

Otra de las aportaciones que se lograron fue la de realizar los prototipos guía y el Manual Técnico, que contiene las especificaciones de la Arquitectura del sistema, tales como: vistas arquitectónicas, requerimientos funcionales, no funcionales, restricciones del sistema, etc.

4.2 Recomendaciones

Una vez concluido el trabajo de titulación, se considera interesante investigar sobre otros aspectos relacionados con el geoprocesamiento y distribución equitativa de trabajo. Por lo tanto, entre otras cosas se propone: extender los estudios expuestos en este trabajo al estudio

del diseño de un balanceador de carga para distribuir de manera equitativa el trabajo entre los recursos disponibles, así como para evitar un único punto de paso del tráfico y por tanto eliminar el cuello de botella que podría en el futuro causar la no disponibilidad de la información a través de la plataforma.

También, se sugiere trabajar en mejorar el modelo de arquitectura utilizado en este trabajo de titulación para determinar las tareas de geoprocesamiento, esto mediante el refinamiento del Documento de Arquitectura como lo indica la *Iteración 0* de Scrum.

4.3 Trabajos futuros

Como trabajos futuros de esta tesis, se considera:

- Extender los estudios expuestos en este trabajo de titulación al estudio de otros estilos de arquitecturas de TI, en especial a la computación en la nube o Cloud Computing. Este tema en particular no fue considerado en este trabajo por los siguientes motivos: el proyecto PyP2 de LactaLAB, al ser un proyecto financiado por la DIUC XVII convocatoria, tiene limitaciones de tiempo y asignación de un presupuesto específico. Por lo que, si se requiere del despliegue de algún componente sobre la nube se deberá realizar una planificación y análisis profundo de los componentes a migrar. Para esto se debe considerar el aprovisionamiento de uno o más servidores, que en el caso de un escenario de big data se recomienda al menos tres servidores en la nube para sustentar el funcionamiento correcto del componente desplegado (*Hadoop – Apache Hadoop 3.2.1*, s. f.). Esto está asociado a un costo por la cantidad, flujos de datos y tipos de recursos utilizados (ver Tabla 4.1), lo que por el momento no es viable su consideración.

Proveedor	Detalle VM	Costo/Mes	Costo/Año	Cantidad	Total/Año
Azure	Windows, 2vCPU, 8GB RAM, 1024 GB HDD	\$198.53	\$2382.36	3 VMs	\$7147.08
AWS	Windows, 2vCPU, 8GB RAM, 1024 GB HDD	\$189.44	\$2273.28	3 VMs	\$6819.84
Google Cloud	Windows, 2vCPU, 7.5GB RAM, 1025 GB HDD	\$199.87	\$2398.44	3 VMs	\$7195.32

Tabla 4.1 Cálculo aproximado de costos de proveedores cloud

Elaborado por: (Los Autores)

Fuente: (Calculadora de precios / Microsoft Azure, s. f.; Amazon Web Services Simple Monthly Calculator, s. f.; Calculadora de precios de Google Cloud Platform, s. f.)

Los detalles de costos aproximados mostrados en la Tabla 4.1 corresponden a un escenario planteado para LactaLAB sobre la nube, en el que se realizan algoritmos Map-Reduce en un mínimo de tres máquinas virtuales (Hadoop – Apache Hadoop 3.2.1, s. f.). Cabe recalcar que en las VMs no están agregadas instantáneas para opciones de redundancia, ni tampoco ninguna opción de soporte por parte del proveedor, ya que esto elevaría aún más el precio considerado.

- Analizar con mayor detenimiento el uso de un mockup como instrumento de validación de requisitos a nivel de casos de uso.

Bibliografía

- ACM Digital Library. (s. f.). Recuperado 1 de septiembre de 2019, de <https://dl.acm.org/>
- Alfonzo, M. A., Boarnet, M. G., Day, K., Mcmillan, T., & Anderson, C. L. (2008). The relationship of neighbourhood built environment features and adult parents' walking. *Journal of Urban Design*, 13(1), 29–51. <https://doi.org/10.1080/13574800701803456>
- Alvarez, A. (2017). *Sistema de bicicletas públicas encicla*.
- Amazon Web Services Simple Monthly Calculator. (s. f.). Recuperado 1 de septiembre de 2019, de <https://calculator.s3.amazonaws.com/index.html>
- Argis. (2016). *What is geoprocessing?*
<http://desktop.arcgis.com/en/arcmap/10.3/main/analyze/what-is-geoprocessing.htm>
- Bass, L., Clements, P., & Kazman, R. (2012). *Software architecture in practice* (3a ed.). Addison-Wesley Professional.
<http://gen.lib.rus.ec/book/index.php?md5=80a2b31e3b36d1ce9124cf69d0b7c9ab>
- Buehler, K., & McKee, L. (1996). *The OpenGIS Guide: Introduction to Interoperable Geoprocessing: Part I of the Open Geodata Interoperability Specification (OGIS)*. Open GIS Consortium, Incorporated.
- Calculadora de precios | Microsoft Azure. (s. f.). Recuperado 1 de septiembre de 2019, de <https://azure.microsoft.com/es-es/pricing/calculator/>
- Calculadora de precios de Google Cloud Platform. (s. f.). Google Cloud. Recuperado 1 de septiembre de 2019, de <https://cloud.google.com/prodjhucts/calculator?hl=es>
- Castanhari, R. E. S., Dos Santos Rocha, R., De Andrade, S. C., & De Albuquerque, J. P. (2016). A software architecture to integrate sensor data and volunteered: Geographic information for flood risk management. Proceedings of the International ISCRAM Conference. Scopus.
- Fink, Arlene. (2014). Conducting research literature reviews: from the internet to paper.

Thousand Oaks, California :SAGE.

El Papel de la Arquitectura de Software en Scrum. (2015). SG Buzz.

<https://sg.com.mx/revista/30/el-papel-la-arquitectura-software-scrum>

ESMARTCITY • Noticias Diarias sobre Ciudades Inteligentes. (2017). ESMARTCITY.

<https://www.esmartcity.es/>

Esri. (2018, noviembre). *ArcGIS*. <https://www.esri.com>

Hadoop – Apache Hadoop 3.2.1. (s. f.). Recuperado 1 de septiembre de 2019, de

<https://hadoop.apache.org/docs/stable/>

Hernández Sampieri, R., Fernández Collado, C., Baptista Lucio, P., & others. (2006).

Metodología de la investigación (Vol. 3). México: McGraw-Hill.

Hoffman, C. (2016). *How GPS Actually Works*. How-To Geek.

<https://www.howtogeek.com/137862/htg-explains-how-gps-actually-works/>

IMCO. (2018). *Índice de movilidad urbana*.

Inteligencia logística y los beneficios de Big Data. (s/f). énfasis. Recuperado el 1 de

septiembre de 2019, de [http://www.logisticasud.enfasis.com/articulos/73759-](http://www.logisticasud.enfasis.com/articulos/73759-inteligencia-logistica-y-los-beneficios-big-data)

[inteligencia-logistica-y-los-beneficios-big-data](http://www.logisticasud.enfasis.com/articulos/73759-inteligencia-logistica-y-los-beneficios-big-data)

Krämer, M., & Senner, I. (2015). A modular software architecture for processing of big

geospatial data in the cloud. *Computers and Graphics* (Pergamon), 49, 69-81.

Scopus. <https://doi.org/10.1016/j.cag.2015.02.005>

Lattanze, A. J. (2009). *Architecting software intensive systems: A practitioner's guide*. CRC

Press.

Lizárraga Mollinedo, C. (2006). Movilidad urbana sostenible: Un reto para las ciudades del

siglo XXI. *Economía Sociedad y Territorio*. <https://doi.org/10.22136/est002006260>

LlactaLAB. (2018, enero). *Universidad de Cuenca—LlactaLAB*.

<https://llactalab.ucuenca.edu.ec/>

Malek, S., Edwards, G., Brun, Y., Tajalli, H., Garcia, J., Krka, I., ... Sukhatme, G. S.

(2010). An architecture-driven software mobility framework. *Journal of Systems and*

Software, 83(6), 972–989. doi:10.1016/j.jss.2009.11.003

- Manum, B., & Nordström, T. (2013). Integrating Bicycle Network Analysis in Urban Design: Improving bikeability in Trondheim by combining space syntax and GIS-methods using the place syntax tool. *International Space Syntax Symposium*, 14.
- Melis, A., Mirri, S., Prandi, C., Prandini, M., & Salomoni, P. (2016). A microservice-based architecture for the development of accessible, crowdsensing-based mobility platforms. 498-505. Scopus. <https://doi.org/10.1109/CTS.2016.91>
- Movilidad Urbana • ESMARTCITY*. (2019). ESMARTCITY.
<https://www.esmartcity.es/movilidad-urbana>
- Olivera, H. V., & Holanda, M. (2012). A GIS Web with integration of sheet and soil databases of the Brazilian Cerrado. Iberian Conference on Information Systems and Technologies, CISTI. Scopus.
- Piedra, D. A. C. (2016). *Hacia la construcción de una Ordenanza de Movilidad Sostenible*. 27.
- Pies y Pedales – LlactaLAB – Universidad de Cuenca*. (2018).
<https://llactalab.ucuenca.edu.ec/investigacion/pies-y-pedales/>
- QGIS. (2018, noviembre). *QGIS Un Sistema de Información Geográfica libre y de Código Abierto*. <https://www.qgis.org/>
- ¿Que es LlactaLAB? – LlactaLAB – Universidad de Cuenca*. (2017).
<https://llactalab.ucuenca.edu.ec/que-es-llactalab/>
- Ramsey, P. (2007). The state of open source GIS. *Refractions Research Inc*, 1–49.
- Rodríguez, D. A., Brisson, E. M., & Estupiñán, N. (2009). The relationship between segment-level built environment attributes and pedestrian activity around Bogota's BRT stations. *Transportation Research Part D: Transport and Environment*, 14(7), 470–478. <https://doi.org/10.1016/j.trd.2009.06.001>
- Rozanski, N., & Woods, E. (2005). *Software systems architecture: Working with stakeholders using viewpoints and perspectives*. Addison-Wesley.



Santovenia Díaz, J., Tarragó Montalvo, C., & Cañedo Andalia, R. (2009). Sistemas de información geográfica para la gestión de la información. *ACIMED*, 20(5), 72–75.

Scopus. (s. f.). Recuperado 1 de septiembre de 2019, de <https://www.scopus.com/>

Torre Llera, F. (2018). *Aplicaciones del Big Data a la logística* [Master, UNIVERSIDAD DE VALLADOLID].

<https://repositorio.utb.edu.co/bitstream/handle/20.500.12585/1939/0043898.pdf?sequence=1>

Tricco, AC, Lillie, E, Zarin, W, O'Brien, KK, Colquhoun, H, Levac, D, Moher, D, Peters, MD, Horsley, T, Weeks, L, Hempel, S et al. PRISMA extension for scoping reviews (PRISMA-ScR): checklist and explanation. *Ann Intern Med*. 2018,169(7):467-473. doi:10.7326/M18-0850.

Velez, A. (2017). *SMARTMOBILITY PLATFORM*.

Welcome to The Open Geospatial Consortium | OGC. (s. f.). Recuperado 1 de septiembre de 2019, de <https://www.opengeospatial.org/>