

**DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO PARA LA ACTUALIZACIÓN  
DE NOMENCLATURA EN LA ZONA DE EXPANSIÓN DE SANTIAGO DE CALI  
PARA EL SIG CORPORATIVO DE UNA EMPRESA DE  
TELECOMUNICACIONES  
-GEOCOORD-**

**ESTUDIANTE:  
MARYBEL ZARAMA CASTAÑEDA - 9930525**

**UNIVERSIDAD DEL VALLE  
FACULTA DE INGENIERÍA  
PROGRAMA ACADÉMICO INGENIERÍA TOPOGRÁFICA  
SANTIAGO DE CALI  
2019**

**DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO PARA LA ACTUALIZACIÓN  
DE NOMENCLATURA EN LA ZONA DE EXPANSIÓN DE SANTIAGO DE CALI  
PARA EL SIG CORPORATIVO DE UNA EMPRESA DE  
TELECOMUNICACIONES  
-GEOCOORD-**

**ESTUDIANTE:  
MARYBEL ZARAMA CASTAÑEDA - 9930525**

**TRABAJO DE GRADO**

**DIRECTOR:  
CESAR EDWIN GARCÍA CORTES**

**UNIVERSIDAD DEL VALLE  
FACULTA DE INGENIERÍA  
PROGRAMA ACADÉMICO INGENIERÍA TOPOGRÁFICA  
SANTIAGO DE CALI  
2019**

Nota de aceptación

---

---

---

---

---

---

---

Presidente jurado

---

Presidente jurado

---

Presidente jurado

Santiago de Cali, 28 de octubre de 2019.

## **AGRADECIMIENTOS**

A Dios por ser mi guía y acompañarme en el transcurso de mi vida, brindándome paciencia y sabiduría para culminar con éxito mis metas propuestas.

A mis padres Jorge y Nidian (QEPD) por ser mi pilar fundamental y haberme apoyado incondicionalmente, pese a las adversidades e inconvenientes que se presentaron.

A mi esposo Alfredo que con su apoyo, consejos, amor, y paciencia me ayudó a concluir esta meta.

A mis hermanos Jorge y Patricia por facilitarme los caminos para seguir, sin pedir nada a cambio y sin dudar de mi capacidad.

A mi Director de proyecto de grado Ing. García, que gracias a sus consejos y correcciones hoy puedo culminar este trabajo.

A los Profesores que gracias a sus conocimientos permitieron desarrollarme profesionalmente y seguir cultivando mis valores.

A los dos ingenieros. desarrolladores, que por su conocimiento apoyaron para que Geocoord fuera una meta cumplida.

Y por supuesto a la Universidad y a todas las autoridades, por permitirme concluir con una etapa de mi vida, gracias por la paciencia, orientación y guiarme en el desarrollo de esta investigación.

## TABLA DE CONTENIDO

|                                                              |    |
|--------------------------------------------------------------|----|
| INTRODUCCIÓN.....                                            | 1  |
| 2. PLANTEAMIENTO DEL PROBLEMA.....                           | 3  |
| 3. JUSTIFICACIÓN .....                                       | 5  |
| 4. OBJETIVOS .....                                           | 6  |
| 4.1. Objetivo general.....                                   | 6  |
| 4.2. Objetivos Específicos .....                             | 6  |
| 5. MARCO CONCEPTUAL .....                                    | 7  |
| 5.1. Base de Datos.....                                      | 7  |
| 5.2. Sistema de gestión de Base de Datos (SGBD) .....        | 7  |
| 5.3. Navegador web .....                                     | 7  |
| 5.4. Servidor Web .....                                      | 8  |
| 5.4.1. Elementos básicos para crear un servidor web .....    | 9  |
| 5.5. Modelo Cliente-Servidor .....                           | 9  |
| 5.6. Modelo-vista-controlador (MVC) .....                    | 10 |
| 5.7. REST .....                                              | 10 |
| 5.8. PostgreSQL .....                                        | 11 |
| 5.9. PHP .....                                               | 11 |
| 5.10. JavaScript .....                                       | 11 |
| 5.11. FrontEnd – Backend.....                                | 11 |
| 5.12. JSON-HTML-CSS.....                                     | 12 |
| 5.12.1. JSON (JavaScript Object Notation) .....              | 12 |
| 5.12.2. HTML (HyperText Markup Language) .....               | 12 |
| 5.12.3. CSS (Cascading Style Sheets) .....                   | 12 |
| 5.13. Servidores de Maps.....                                | 12 |
| 5.13.1. API de Google .....                                  | 12 |
| 5.13.2. Openstreetmaps .....                                 | 12 |
| 5.14. Sistema de información geográfico (SIG) .....          | 13 |
| 5.15. Geocodificación.....                                   | 13 |
| 5.16. Nomenclatura urbana.....                               | 13 |
| 6. MARCO TEÓRICO .....                                       | 14 |
| 6.1. Planos urbanísticos .....                               | 14 |
| 6.2. Direcciones urbanas.....                                | 14 |
| 6.2.1. Conceptos básicos.....                                | 14 |
| 6.2.2. Elementos que componen un registro de dirección ..... | 16 |
| 6.3. Estandarización de direcciones urbanas .....            | 17 |
| 6.4. Geocodificación.....                                    | 17 |
| 6.5. Modelo entidad relación.....                            | 18 |
| 6.5.1. Tipos de relaciones .....                             | 18 |
| 6.6. Expresiones regulares .....                             | 19 |

|        |                                                                   |    |
|--------|-------------------------------------------------------------------|----|
| 6.6.1. | Conceptos de las Regex.....                                       | 19 |
| 6.6.2. | Métodos de las expresiones regulares.....                         | 21 |
| 6.7.   | Tufr.....                                                         | 22 |
| 6.7.1. | Herramientas de análisis que incluye Turf.....                    | 22 |
| 7.     | MATERIALES Y MÉTODOS .....                                        | 24 |
| 7.1.   | Materiales.....                                                   | 24 |
| 7.2.   | Descripción de la zona de estudio .....                           | 24 |
| 7.3.   | Recolección de la información.....                                | 26 |
| 7.3.1. | Recolección de requerimientos de la empresa de telecomunicaciones | 26 |
| 7.3.2. | Recolección de información urbana.....                            | 26 |
| 7.4.   | Diseño Geocoord .....                                             | 27 |
| 7.5.   | Implementación de la prueba piloto de Geocoord .....              | 28 |
| 7.6.   | Validación.....                                                   | 28 |
| 8.     | RESULTADOS .....                                                  | 30 |
| 8.1.   | Recolección de la Información.....                                | 30 |
| 8.1.1. | Recolección de requerimientos de la empresa de telecomunicaciones | 30 |
| 8.1.2. | Recolección de información urbana.....                            | 32 |
| 8.2.   | Diseño.....                                                       | 34 |
| 8.2.1. | Modelo de Datos .....                                             | 34 |
| 8.2.2. | Modelo Conceptual .....                                           | 35 |
| 8.2.3. | Proceso de estandarización.....                                   | 37 |
| 8.2.4. | Proceso digitalización direcciones urbanas.....                   | 38 |
| 8.3.   | Implementación de la prueba piloto de Geocoord .....              | 38 |
| 8.3.1. | Despliegue de Geocoord .....                                      | 38 |
| 8.3.2. | Interfaz Geocoord .....                                           | 39 |
| 8.3.3. | Funcionalidad Geocoord.....                                       | 41 |
| 8.4.   | Validación.....                                                   | 47 |
| 9.     | CONCLUSIONES.....                                                 | 49 |
| 10.    | BIBLIOGRAFIA.....                                                 | 51 |
| 11.    | ANEXOS.....                                                       | 55 |

## ÍNDICE DE FIGURAS

|                                                                  |    |
|------------------------------------------------------------------|----|
| Figura 1. Funcionamiento de un Servidor Web. ....                | 8  |
| Figura 2. Modelo Cliente-Servidor.....                           | 10 |
| Figura 3. Vías principal-generador. ....                         | 15 |
| Figura 4. Nomenclatura predial. ....                             | 15 |
| Figura 5. Relación 1:1. ....                                     | 18 |
| Figura 6. Relación 1 a muchos. ....                              | 19 |
| Figura 7. Relación muchos a muchos. ....                         | 19 |
| Figura 8. Zona de Expansión. ....                                | 25 |
| Figura 9. Modelo conceptual con arquitectura REST. ....          | 28 |
| Figura 10. Proceso en uso Vs Geocoord. ....                      | 30 |
| Figura 11. Interfaz Geocoord. ....                               | 31 |
| Figura 12. Archivo cvs- casas y conjunto cerrados casas. ....    | 32 |
| Figura 13. Archivo .cvs - Edificio- conjunto cerrado torres..... | 32 |
| Figura 14. Carpetas de Insumos. ....                             | 33 |
| Figura 15. Diseño base de datos-Tablas. ....                     | 34 |
| Figura 16. Controladores de Backend de Geocoord. ....            | 36 |
| Figura 17. Modelos de Backend de Geocoord.....                   | 37 |
| Figura 18. Línea de código para el despliegue de Geocoord. ....  | 39 |
| Figura 19. Interfaz Geocoord-Componentes. ....                   | 39 |
| Figura 20. Componente modal. ....                                | 40 |
| Figura 21. Componente modelo exportación. ....                   | 41 |
| Figura 22. Visualización de capa activa/desactiva. ....          | 42 |
| Figura 23. Selección de elementos. ....                          | 43 |
| Figura 24. Representaciones de elementos. ....                   | 44 |
| Figura 25. Formulario elemento manzana. ....                     | 45 |
| Figura 26. Formulario elem. conjunto cerrado. ....               | 45 |
| Figura 27. Formulario elemento Torres. ....                      | 46 |
| Figura 28. Formulario elemento Eje de casas.....                 | 46 |

## ÍNDICE DE TABLAS

|                                                              |    |
|--------------------------------------------------------------|----|
| Tabla 1. Metacaracteres de posicionamiento.....              | 20 |
| Tabla 2. Metacaracteres predefinidos. ....                   | 20 |
| Tabla 3. Metacaracteres cuantificadores. ....                | 21 |
| Tabla 4. Métodos que utilizan expresiones regulares.....     | 22 |
| Tabla 5. Estandarización de abreviaturas de direcciones..... | 33 |
| Tabla 6. Estandarización de abreviaturas de direcciones..... | 37 |
| Tabla 7. Exportado predios malla vial.....                   | 47 |
| Tabla 8. Exportado conjunto cerrado torres. ....             | 47 |

## RESUMEN

En este proyecto se presenta el diseño e implementación de un módulo que permite actualizar la nomenclatura urbana con coordenadas geográficas con el objetivo de obtener un archivo con datos espaciales que hace parte del SIG Corporativo manejado por la empresa de telecomunicaciones.

Para alcanzar el objetivo se realizó una búsqueda bibliográfica de conceptos y herramientas de software libre que tuvieran componente espacial para el diseño del módulo. En la implementación y validación se realizaron pruebas en algunos sectores sin cobertura cartográfica en el municipio de Cali, específicamente en zonas que se consideran de expansión urbana. En el diseño se definió el proceso para realizar la estandarización de las direcciones mediante algoritmos y se buscó poder realizar el módulo mediante un visor geográfico que facilitará la actualización de la información.

Este módulo cumple con requerimientos solicitados por la empresa de telecomunicaciones optimizando parte de los procesos requeridos en la alimentación de datos del SIG Corporativo sin incrementar costos gracias al uso de software de código abierto, a su fácil manejo, obteniendo mediante pruebas de comparación en tiempos, una optimización de un 74% con lo que respecta al proceso que actualmente está en uso y el módulo Geocoord . En general, Geocoord permite al usuario generar direcciones urbanas del municipio de Cali, de manera georreferenciada y la visualización de las direcciones estandarizadas según la malla vial y predios verticales.

Palabras Claves: Sistema de información geográfica, software libre, nomenclatura urbana, geocodificación de direcciones.

## INTRODUCCIÓN

Actualmente la información geográfica cumple un papel protagónico en las diferentes disciplinas debido a la necesidad de ubicar en un mismo sistema espacial un evento, un fenómeno o un problema de una zona extensa o pequeña. Timarán, Hernández y Quemá documentan que los sistemas de Información geográfica (SIG) surgen como herramientas de apoyo en la toma de decisiones que permiten procesar esta información desde su dimensión espacial y suministrar datos para la formulación de soluciones a problemas de orden geográfico (Timarán, Hernández y Quemá, 2016).

“El término SIG, que en la actualidad está ampliamente difundido tanto en la geografía como en otras ciencias, en especial en aquellas vinculadas con la planificación territorial y la resolución de problemas socioeconómicos y ambientales, es de compleja definición habida cuenta de sus capacidades técnicas y analíticas y su carácter multipropósito” (López y Moreno, 1997, p. 789).

Los SIG utilizan como instrumento primordial la georreferenciación, esta permite el análisis de datos espaciales basado en ubicaciones específicas reconocibles en un área geográfica de estudio, para ello se pueden apoyar en un editor espacial que permite asignar coordenadas geográficas (latitud-longitud) a puntos del mapa (direcciones, puntos de interés, etc.). Las coordenadas geográficas producidas pueden luego ser usadas para localizar el punto del mapa en un Sistema de Información Geográfica.

En el V congreso internacional de gestión de tecnológica y de la innovación (2016), se expuso que en Colombia se han trabajado sistemas de georreferenciación basados en límites geográficos como regiones, departamentos, barrios, predios, sectores, veredas, entre otros; para analizar los eventos y posteriormente reflejarlos en mapas temáticos. En muchos casos la información capturada contiene su dirección urbana, una característica que, de ser procesada, permitiría el análisis geográfico de los datos con un grado alto de detalle y exactitud, desafortunadamente estos datos no se toman en cuenta debido a la ausencia de herramientas que permitan este tipo de procesamiento (Timarán, Hernández y Quemá, 2016).

Para las pequeñas y medianas empresas el análisis espacial de mercados debe responder a preguntas como las siguientes: ¿Dónde se localizan los clientes?, ¿Dónde existe una concentración importante de clientes potenciales para el negocio?, ¿Dónde debemos distribuir nuestros productos y servicios?, ¿Dónde se localizan los puntos de oferta de la competencia?, ¿Dónde podemos establecer nuevos puntos de oferta?, o el simple hecho de conocer donde están los activos o la infraestructura de la empresa, por ello de la importancia de la georreferencia, palabra que indica a la localización de un objeto en el espacio en un sistema de coordenadas (Mercado, 2013).

Por lo anterior disponer de cartografía urbana actualizada es de gran beneficio tanto para el sector privado como público. Tal es el caso de la empresa de telecomunicaciones que requiere optimizar la actualización de nomenclaturas de zonas en expansión urbana, esta necesidad generó el objetivo del presente proyecto que contiene el diseño e implementación de una herramienta que permite crear las direcciones con problemas de posicionamiento espacial de la ciudad de Cali, específicamente en las zonas de expansión urbana mediante el uso de un editor espacial que por medio de un servidor web local, estandariza la nomenclatura y agrega el componente de georreferenciación.

## 2. PLANTEAMIENTO DEL PROBLEMA

En Colombia, el relieve y la geomorfología están llenos de accidentes geográficos que ocasionan diferencias en el sentido y orientación de las vías principales (calles y carreteras) de algunas de las principales ciudades del país. Además, la generación de otras vías alternas adicionales a las vías principales, tales como avenidas, transversales, circulares y pasajes, generan algunos inconvenientes en el momento de estandarizar la nomenclatura urbana a nivel nacional (Rosa, 2014), (Rivas, 2016). Además, se le suma el crecimiento poblacional que se está presentando. Se tiene una población urbana cercana al 77 % principalmente en Bogotá, Medellín, Cali y Barranquilla (Gallego y Suarez, 2017) dando como resultado el crecimiento en la construcción de casas, edificios, entre otros tipos de infraestructura.

Sin embargo, se presentan dificultades en los sectores urbanos nuevos, que a pesar de que la cartografía del territorio nacional es elaborada por instituciones públicas, el acelerado crecimiento poblacional también contribuye a tener una cartografía desactualizada (Polo, 2013). Por lo tanto, si se requiere hacer uso de este tipo de información, se debe adquirir a través de elevados pagos a contratistas y/o software privados, razón por la cual, no todas las instituciones, empresas y personas pueden obtenerla. Esto ha conllevado a que los procesos que emplean cartografía utilicen materiales de dudosa calidad y origen (Ordas, J., 2012), con lo cual se pone en riesgo la toma de decisiones de los proyectos que beneficiaran a la población en el acceso a los servicios que requieren.

Además de la desactualización cartográfica se tiene que las direcciones que tienen problemas de calidad de datos dificulta la búsqueda acertada de direcciones válidas en una ciudad esto afecta en la toma de decisiones en los procesos operativos de una empresa. Moore (2007) expresa que en el caso de direcciones erradas las empresas tendrán problemas en el momento de contactar clientes, la distribución de los productos no se realizará de una manera eficiente y se generará costos y re-procesos adicionales que afecten el sostenimiento y rentabilidad a largo plazo (Rivas, 2016).

El crecimiento urbano que presenta la ciudad de Cali y sus alrededores (Lopez, 2018), ha ocasionado que no existan cartografía actualizada de manera rápida y por ende la empresa de telecomunicaciones, por la necesidad de la información trabajan con datos que no tienen una geolocalización. El ingeniero cartográfico de la empresa de telecomunicaciones nos relata que la información cartográfica es suministrada por la empresa contratista Servinformación, sin embargo la dinámica de actualización de la información cartográfica en esta entidad privada no avanza con la misma velocidad que los proyectos de urbanización, razón por la cual existen zonas de la ciudad que no tienen información actualizada y son precisamente estas zonas donde se requiere completar la información, con una aproximación que permita ubicar el predio a su posición más exacta posible; para ello la empresa utiliza recursos como planos de las constructoras y levantamientos topográficos realizados por la compañía ya sea en

formato físico o digital (dwg), esta información aunque valiosa en general por el contenido, tiene algunos limitantes debido a que solo es información gráfica, sin atributos específicos ni georreferenciada. No obstante, con el uso del paquete de Excel de Microsoft Office y el servidor web Google Map permite solventar de forma parcial esta situación y generar información requerida para la alimentación de la base de datos, pese a que es poco sistematizado se obtienen los datos requeridos como son, una dirección normalizada y las coordenadas geográficas de cada punto que representa casas, edificios y/o conjuntos residenciales.

La participación laboral en la empresa de telecomunicaciones permite identificar que la cartografía desactualizada por las nuevas urbanizaciones, conjuntos cerrados y edificios genera inconsistencias en la gestión de actualización de nomenclatura urbana georreferenciada, ya que el proceso usado actualmente toma mucho tiempo y genera errores humanos debido a su proceso manual. La problemática se presenta a nivel nacional, por ello se hace necesario una herramienta que permita obtener un producto que sea compatible con el gestor de base de datos privado que tiene la empresa y que pueda ser aplicado en las diferentes regiones del país.

Para proponer una solución a esta problemática se plantea una interrogante, ¿De qué manera se puede facilitar a la empresa de telecomunicaciones de la ciudad de Cali ingresar nuevos puntos de direcciones georreferenciados en zonas de expansión urbana permitiendo de manera rápida tener una actualización en sus datos y en corto tiempo?

### **3. JUSTIFICACIÓN**

La gestión de la empresa de telecomunicaciones en la que se realiza el proyecto, en busca de tener actualizada la base de datos de nomenclatura predial en zonas de expansión urbana aprovecha los recursos existentes en servidores web de mapas y documentación levantada en terreno que permite tener direcciones georreferenciadas, sin embargo la metodología utilizada es obsoleta debido a que maneja la generación de las direcciones prediales y su componente espacial de forma separada y poco automatizada, tomando más tiempo y dando cabida a mayores errores en su digitación.

Siendo para la empresa el sector urbano su campo de acción, por tal motivo darle solución a la optimización del proceso de actualización de nomenclatura de predios urbanos en zonas de expansión urbana de la ciudad de Cali es de gran importancia debido a que tener cartografía actualizada en tiempo real es casi imposible, además de la necesidad de que la información se encuentre estructurada según los requerimientos de la empresa y con la facilidad de que pueda ser trabajado en diferentes sedes regionales.

Se hace evidente la necesidad optimizar un módulo que permita actualizar la nomenclatura y la ubicación geográfica de los nuevos predios producto de la actual zona de expansión urbana y que se pueda acceder desde cualquier punto con acceso a internet, es decir, se requiere una información de mejor calidad mediante una interfaz de usuario tipo Web que permita tomar la mejor decisión en cuanto aproximación de coordenadas y precisión en la ubicación sin dejar de lado la estandarización que incluye la normalización de las direcciones según lo indicado por la empresa.

## **4. OBJETIVOS**

### **4.1. Objetivo general**

Diseñar e implementar una aplicación que permita la actualización de nomenclatura urbana con geolocalización de una manera interactiva y colaborativa para la empresa de telecomunicaciones.

### **4.2. Objetivos Específicos**

- Diseñar un modelo de datos que permita ingresar nuevos registros prediales en el SIG de la empresa implementando la estandarización manejada.
- Desarrollar la arquitectura, metodología, diseño y mecanismos para la visualización, consulta y creación de información espacial, que permita facilitar la generación de registros espaciales.
- Establecer los parámetros de control en las condiciones propicias para validar el funcionamiento del editor mediante los resultados de direcciones consultadas o generadas.

## **5. MARCO CONCEPTUAL**

Debido a la existencia de varias herramientas que se utilizan para el manejo de información espacial se considera necesario conocer conceptos generales claves para el desarrollo del proyecto.

### **5.1. Base de Datos**

Conjuntos de datos interrelacionados entre sí y su fin es servir a uno o varios usuarios, sin redundancias perjudiciales e innecesarias, es independiente de la aplicación que la utilice y tiene operaciones específicas. El manejo de las bases de datos se lleva mediante software de sistemas de gestión (Hernández, 2010).

### **5.2. Sistema de gestión de Base de Datos (SGBD)**

Definimos un Sistema Gestor de Bases de Datos o SGBD como una colección de datos relacionados entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos (Irlés y Outerio, 2013).

La colección de esos datos se denomina Base de Datos o BD, (DB Data Base). Como objetivos principales de los SGBD constan los siguientes:

- Independencia de datos: o Los programas de aplicación deben verse afectados lo menos posible por cambios efectuados en datos que no usan.
- Integridad de los datos: o La información almacenada en la BD debe cumplir ciertos requisitos de calidad, para ello hace falta, en el momento de introducirse los valores de los datos, que éstos se almacenen debidamente, y que posteriormente no se deterioren.
- Seguridad de los datos: o A la información almacenada en la BD sólo pueden acceder las personas autorizadas y de la forma autorizada.

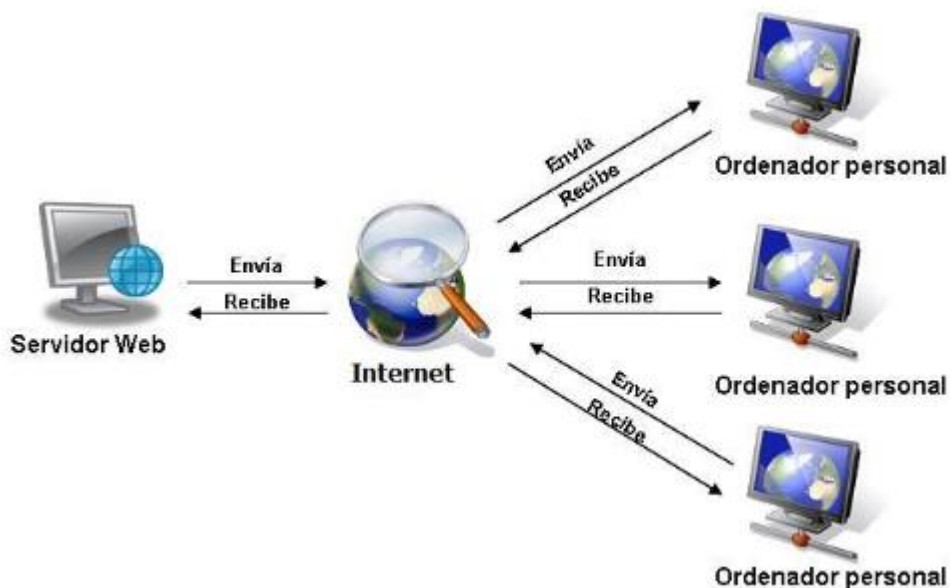
### **5.3. Navegador web**

Un navegador web (en inglés, web browser) es un software, aplicación o programa que permite el acceso a la Web, interpretando la información de distintos tipos de archivos y sitios web para que estos puedan ser vistos.

La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Además, permite visitar páginas web y hacer actividades en ella, es decir, enlazar un sitio con otro, imprimir, enviar y recibir correo, entre otras funcionalidades más.(Wikipedia, 1991).

#### 5.4. Servidor Web

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente. El código recibido por el cliente es recibido por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se usa el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI (Modelo de interconexión de sistemas abiertos) (Web.Código Integral, 2015). Este funcionamiento se puede observar en la Figura 1.



*Figura 1. Funcionamiento de un Servidor Web.*

En el caso del proyecto se trabajó con un servidor web local, que está localizado en un determinado ordenador. De este modo, en vez de acceder a través de un dominio como `www.servidorlocal.com` predeterminado, el servidor local tiene la IP y una sección en los archivos que determina el contenido también conocida como localhost. (Axarnet, 2018).

#### **5.4.1. Elementos básicos para crear un servidor web**

Es primordial contar con una serie de elementos simples para la creación de un servidor web. Estos son:

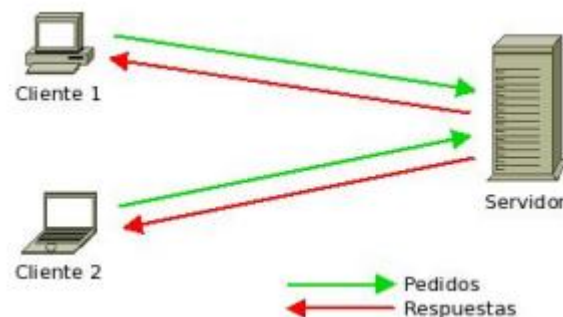
- Un PC que estemos dispuestos a utilizar como servidor. Es recomendable que dicho equipo se utilice sólo para esta labor, aunque podemos utilizar un PC que se use regularmente.
- Una conexión a Internet, que debe ser de preferencia ADSL.
- Un paquete de software que permite instalar todos los programas necesarios para convertir nuestro PC en un servidor, es decir, que cuente con el software de servicio web, un lenguaje de desarrollo web (PHP) y un gestor de base de datos y demás elementos secundarios.

#### **5.5. Modelo Cliente-Servidor**

El modelo Cliente/Servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Las aplicaciones Clientes realizan peticiones a una o varias aplicaciones Servidores, que deben encontrarse en ejecución para atender dichas demandas (Marini, 2012).

El modelo Cliente/Servidor permite diversificar el trabajo que realiza cada aplicación, de forma que los Clientes no se sobrecarguen, cosa que ocurriría si ellos mismos desempeñan las funciones que le son proporcionadas de forma directa y transparente.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. Tanto el Cliente como el Servidor son entidades abstractas que pueden residir en la misma máquina o en máquinas diferentes. Figura 2.



*Figura 2. Modelo Cliente-Servidor.*

### 5.6. Modelo-vista-controlador (MVC)

MVC es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento (Wikipedia,2019).

### 5.7. REST

REST es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web. El término se originó en el año 2000, en una tesis doctoral sobre la web escrita por Roy Fielding, uno de los principales autores de la especificación del protocolo HTTP y ha pasado a ser ampliamente utilizado por la comunidad de desarrollo.

La REST es una arquitectura de desarrollo web que puede ser utilizada en cualquier cliente HTTP. Además, es mucho más simple que otras arquitecturas ya existentes, como pueden ser XML-RPC o SOAP. Esta simplicidad se consigue porque emplea una interfaz web que usa hipermedios para la representación y transición de la información (GaussWebApp, 2014).

## **5.8. PostgreSQL**

PostgreSQL es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto, publicado bajo la licencia PostgreSQL. Contiene un módulo llamado PostGIS que es una extensión del sistema de base de datos relacional que permite almacenar objetos SIG (Sistemas de Información Geográfica) en la base de datos. PostGIS incluye soporte de índices de tipos basados en GiST R-Tree, y funciones de análisis y procesamiento de objetos SIG (PostGis, 2014).

## **5.9. PHP**

PHP Es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el preprocesador de texto plano en UTF-8. Posteriormente se aplicó al desarrollo web de contenido dinámico, dando un paso evolutivo en el concepto de aplicación en línea, por su carácter de servicio (Wikipedia,2019).

## **5.10. JavaScript**

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo, en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo. Algunas de sus librerías son React (Biblioteca de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página), Turf (Turf incluye operaciones espaciales tradicionales -buffer, TINs, isolíneas- , funciones de ayuda para crear datos en formato GeoJSON y herramientas para clasificar datos y para crear estadísticas) (Morales, 2012).

## **5.11. FrontEnd – Backend**

El FrontEnd es la parte de un programa o dispositivo a la que un usuario puede acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios. El Backend es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos. El Backend también accede al servidor, que es una

aplicación especializada que entiende la forma como el navegador solicita cosas (Chapaval, 2017).

## **5.12. JSON-HTML-CSS**

### **5.12.1. JSON (JavaScript Object Notation)**

Es un formato de datos basado en texto que sigue la sintaxis de objeto de JavaScript, popularizado por Douglas Crockford. Aunque es muy parecido a la sintaxis de objeto literal de JavaScript, puede ser utilizado independientemente de JavaScript, y muchos ambientes de programación poseen la capacidad de leer (analizar; parse) y generar JSON (MDN Web Docs, 2019).

### **5.12.2. HTML (HyperText Markup Language)**

Hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros (Wikipedia, 2019).

### **5.12.3. CSS (Cascading Style Sheets)**

Es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML (Wikipedia, 2019).

## **5.13. Servidores de Maps**

### **5.13.1. API de Google**

API de Google es un conjunto de API desarrollada por Google, las cuales permiten la comunicación e integración de los Servicios de Google con otros servicios. Ejemplos incluyen las API de Búsqueda, Gmail, Traductor o Maps. Las aplicaciones de terceros pueden usar esas API para extender la funcionalidad de sus servicios (Wikipedia, 2019).

### **5.13.2. Openstreetmaps**

También conocido como OSM, es un proyecto colaborativo para crear mapas editables y libres. En lugar del mapa en sí, los datos generados por el proyecto se consideran su salida principal

Los mapas se crean utilizando información geográfica capturada con dispositivos GPS móviles, ortofotografías y otras fuentes libres. Esta cartografía, tanto las imágenes creadas como los datos vectoriales almacenados en su base de datos, se distribuye bajo licencia abierta Licencia Abierta de Bases de Datos (Wikipedia,2019).

#### **5.14. Sistema de información geográfico (SIG)**

SIG es una integración organizada de hardware, software y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y de gestión (Laboratorio Unidad Pacifico Sur, 2019).

#### **5.15. Geocodificación**

La Geocodificación es una manera de determinar coordenadas geográficas (e.g. latitud-longitud) a partir de información como direcciones urbanas, puntos de interés, etc. Las coordenadas geográficas producidas pueden ser luego usadas para localizar elementos en un Sistema de Información Geográfico. (Mineducación, 2009).

#### **5.16. Nomenclatura urbana**

Es la identificación tanto de vías como de predios que conforman el área de una ciudad o población, con signos numéricos y alfanuméricos. La nomenclatura consiste en referenciar la ubicación de edificaciones y lotes con respecto a las vías próximas y adyacentes, señalando sus accesos. La nomenclatura se divide en dos partes, nomenclatura vial y nomenclatura predial (Departamento Administrativo de Planeación Municipal, 2019).

## 6. MARCO TEÓRICO

### 6.1. Planos urbanísticos

Se define planos urbanísticos a la representación gráfica de la urbanización, identificando todos los elementos que la componen para facilitar su comprensión, tales como: afectaciones, cesiones públicas para parques, equipamientos y vías locales, áreas útiles y el cuadro de áreas en el que se cuantifique las dimensiones de cada uno de los anteriores elementos y se haga su amojonamiento. Es la representación cartográfica que refleja la división de un terreno en manzanas y lotes. Se debe tener en cuenta que el proceso de urbanización y por ende los planos urbanísticos se presentan tanto en el ámbito público como en el privado (ANDI, 2013).

### 6.2. Direcciones urbanas

En base a estos componentes se debe especificar un modelo de estándar para direcciones urbanas; el estándar de direcciones en el presente proyecto, está en base a la estandarización de abreviaturas de la Circular 300/01 (IGAC) y la manejada por la empresa de telecomunicación. En el momento de realizar la captura y el almacenamiento de direcciones urbanas, se debe tener presente algunos conceptos básicos (Centro de investigación y desarrollo, 2009) :

#### 6.2.1. Conceptos básicos

- **Vía:** Es la zona de uso público o privado destinada al tránsito de vehículos y/o peatones.
- **Vía principal:** Es aquella vía sobre la cual está ubicado el acceso principal del predio. Se compone de tres partes: tipo de vía, identificación alfanumérica y el cuadrante al que pertenece la vía. Ver Figura 4.
- **Vía generadora:** Es la vía de menor denominación numérica que tiene intersección con la vía principal, y se emplea para generar nomenclatura predial (depende de la ubicación de cada predio). Ver Figura 4.

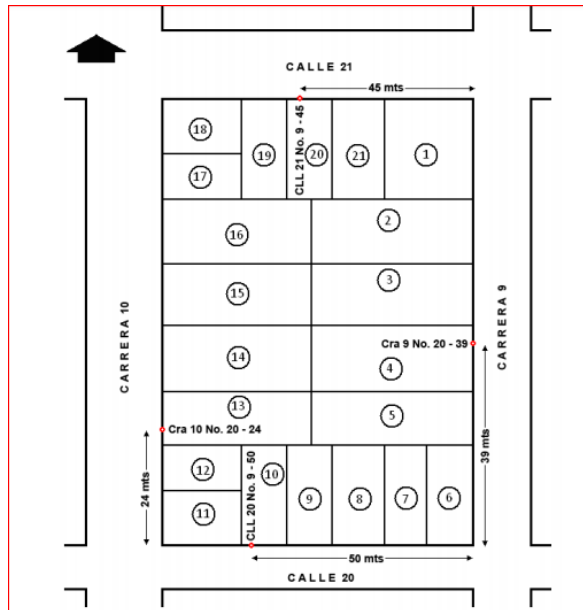


Figura 3. Vías principal-generador.

- **Nomenclatura Predial:** Identificador alfanumérico único asignado a un predio. Consta de dos partes: la nomenclatura vial principal y la placa domiciliaria, esta última a su vez conformada por la nomenclatura vial generadora y el número de placa, el cual representa la distancia aproximada en metros de la vía generadora al acceso del predio ajustándose al número par o impar correspondiente. Si el predio está ubicado al costado derecho de las calles y carreras este valor será par, e impar en el costado izquierdo de las mismas, siguiendo el sentido de crecimiento de la numeración de los ejes viales. Ver Figura 5.

|                             |                               |
|-----------------------------|-------------------------------|
| Nomenclatura predial        | • Avenida Calle 3 No. 53 - 25 |
| Nomenclatura Vía principal  | • Avenida Calle 3             |
| Placa Domiciliaria          | • 53 - 25                     |
| Nomenclatura vía generadora | • 53                          |
| Número de placa             | • 25                          |

Figura 4. Nomenclatura predial.

### 6.2.2. Elementos que componen un registro de dirección

Una vía puede ser clasificada de acuerdo a su orientación y diseño en:

- **Calle:** Se codifica como CL. Vía pública con orientación predominante y sentido de crecimiento numérico de acuerdo al modelo que cada ciudad le haya asignado.
- **Carrera:** Se codifica como KR. Vía pública generalmente perpendicular a la calle con orientación predominante y sentido de crecimiento numérico de acuerdo al modelo que cada ciudad le haya asignado.
- **Diagonal:** Se codifica como DG. Vía pública que generalmente tiene el mismo sentido de la calle sin ser paralela a ésta, puede o no generar nomenclatura predial.
- **Transversal:** Se codifica como TV. Vía pública que generalmente tiene el mismo sentido de la carrera sin ser paralela a ésta, puede o no generar nomenclatura predial.
- **Avenida:** Vía pública cuyas especificaciones y características son notoriamente superiores a las vías predominantes. Genera nomenclatura como la vía generadora que esté representando. En algunos casos locales las avenidas tienen denominaciones como: bulevar, vía, paseo, circunvalar, autopista, entre otros. Se codifica como AV.
- **Número Nombre común de la vía principal o vía generadora:** Valor numérico o nombre común que identifica la vía, en este caso la vía principal, por lo general las avenidas o vías principales tienen asociado un nombre común como por ejemplo: "SIMÓN BOLÍVAR" , "CIUDAD DE CALI", "1 DE MAYO", etc.
- **Letra, letra-letra o letra-número-letra de la vía principal o vía generadora:** Campo alfanumérico, sirve para diferenciar las vías internas, generalmente siguen un orden lógico, ya sea alfabético o numérico o sea combinación de ambas.
- **Sufijo (BIS):** El prefijo BIS es asignado para diferenciar vías de igual denominación. Se emplea generalmente en tramos de vía no continuos, tanto para vías principales como generadora vías generadoras.
- **Cuadrante (NORTE, SUR, ESTE, OESTE):** Cuadrante geográfico en donde está ubicado el eje vial, indica el cuadrante al que pertenece la vía principal, o la generadora.
- **Número de la placa:** Valor numérico, generalmente indica la distancia en metros desde la intersección entre la vía principal y la vía generadora hasta

el acceso al predio. Corresponde con el segundo valor de la placa domiciliaria o predial (número que está después del guion).

- **Nombre de barrio:** el barrio es el espacio geográfico en que se divide el área urbana, y donde se ubican un conjunto de manzanas, el nombre identifica el Barrio ciudadela o Supermanzana. Se recomienda el nombre asignado por la autoridad Competente.
- **Nombre de urbanización o conjunto cerrado:** Núcleo residencial en un terreno delimitado artificialmente.
- **Nombre de manzana:** la manzana es el espacio geográfico donde se ubica un conjunto de predios urbanos edificados o sin edificar delimitado generalmente por vías públicas, el nombre es un campo alfanumérico que contiene el nombre o identificador de la Manzana.
- **Nombre predio:** Campo alfanumérico que contiene el nombre o identificador del predio.

### 6.3. Estandarización de direcciones urbanas

Una dirección urbana puede ser escrita de muchas maneras, el problema más común es la forma como se escribe cada elemento que compone una dirección como por ejemplo el tipo de vía (Carrera, Cra, Kra, Cr, etc.). Otro problema que se presenta tiene que ver con la organización de estos elementos (Departamento Administrativo de Planeación Municipal, 2019).

Se deben estandarizar las direcciones para optimizar procesos en el momento de emplear una geocodificación, por lo tanto, es necesario establecer algunos parámetros para la captura y almacenamiento de direcciones, y especificar una forma normalizada de escribir esta información.

Un estándar se puede definir como el conjunto de reglas que proporcionan una forma de estructurar datos para facilitar su desempeño automático e interoperable.

### 6.4. Geocodificación

La geocodificación es el proceso de transformar una descripción de una ubicación (por ejemplo, un par de coordenadas, una dirección o un nombre de un lugar) en una ubicación de la superficie de la Tierra. Se puede geocodificar introduciendo una descripción de una ubicación a la vez o proporcionando muchas de ellas al mismo tiempo en una tabla. Las ubicaciones que se obtienen se transforman en entidades geográficas con atributos, que se pueden utilizar para la representación cartográfica o el para análisis espacial. (Arcgis, 2016)

Con la geocodificación, puede buscar varios tipos de ubicaciones de manera rápida. Los tipos de ubicaciones que puede buscar incluyen: puntos de interés o nombres de un diccionario geográfico, como montañas, puentes y negocios; coordenadas basadas en latitud y longitud o en otros sistemas de referencia.

## 6.5. Modelo entidad relación

Silberschatz, Korth y Sudarshan, citan que el modelo entidad relación (E-R) permiten observar gráficamente las relaciones entre entidades de una base de datos. Las entidades son definidas como objetos que son distinguibles de otros objetos y que además poseen atributos. Por otro lado, el modelo (E-R) tiene restricciones de correspondencia de cardinalidad, esto es, el grado de asociación que tienen las entidades (Niño, 2019).

Estas restricciones son definidas de la siguiente forma: Uno a uno 1:1, uno a muchos 1: \* y Muchos a Muchos \*: \*.

A continuación, se muestra un breve resumen de los principales conceptos del modelo entidad relación:

- Entidad: Es la representación de un objeto acerca del cual se desea guardar información.
- Atributo: característica de la entidad dada por un conjunto de atributos o propiedades.
- Relación: Una relación es una conexión entre dos entidades.

### 6.5.1. Tipos de relaciones

Relación 1:1 (uno a uno): Se representa mediante una línea que une las dos entidades relacionadas. A cada ocurrencia de la entidad A le corresponde una ocurrencia de la entidad B, y viceversa.



*Figura 5. Relación 1:1.*

Relación 1:N (uno a muchos): Se representa mediante una flecha que une las dos entidades relacionadas. A cada ocurrencia de la entidad A le corresponden varias ocurrencias de la entidad B, pero a cada ocurrencia de la entidad B sólo le corresponde una ocurrencia de la entidad A.



*Figura 6. Relación 1 a muchos.*

Relación N:M (muchos a muchos): se representa mediante una línea con flechas en sus dos extremos que unen las dos entidades relacionadas. A cada ocurrencia de la entidad A le corresponden varias ocurrencias de la entidad B, y a cada ocurrencia de la entidad B le corresponden varias ocurrencias de la entidad A.



*Figura 7. Relación muchos a muchos.*

## 6.6. Expresiones regulares

Lukacs y Bhadra definen las expresiones regulares (Regex) como una secuencia de caracteres que forman un patrón representativo de un conjunto de datos mayor, lo que permite comparar dicho patrón con otro conjunto de datos. Esto a su vez permite encontrar, extraer, validar y reemplazar cadenas de caracteres con determinadas características. Estos patrones pueden estar compuestos por un conjunto de letras, signos y números (ABC123), como también por metacaracteres. Los metacaracteres son símbolos que representan otros caracteres (`*.[]$+{}'/...`) y tienen como particularidad principal que no se representan a ellos mismos ya que no tienen significado literal, pues son interpretados de una forma diferente por el lenguaje de programación usado. Las expresiones regulares son usadas generalmente para procesamiento de texto, detectando números y letras que cumplen con un patrón definido, estas expresiones constituyen en si un lenguaje conocido como Regex. Las Regex pueden incluir patrones de coincidencia literal, patrones de repetición, de ramificación entre otras reglas de reconocimiento de texto (Niño, 2019).

### 6.6.1. Conceptos de las Regex

Para trabajar con expresiones regulares se requiere tener claridad en los siguientes conceptos:

- **Metacaracter:** Se conoce como metacaracteres aquellos símbolos que dependiendo del contexto tienen un significado especial dentro de la expresión regular. Algunos de los metacaracteres más usados son: `*? . / + [ ] ( ) { } ^ $ | \`
- **Meta caracteres de posicionamiento o anclas:** Este tipo de meta caracteres permiten delimitar la búsqueda del patrón dentro de la cadena de caracteres. Algunos se encuentran en la Tabla 1. :

*Tabla 1. Metacaracteres de posicionamiento.*

| Meta caracteres de posición |                                     |
|-----------------------------|-------------------------------------|
| Meta caracteres             | Descripción                         |
| \$                          | Fin de línea                        |
| \A                          | Inicio de texto                     |
| ^                           | Inicio de línea                     |
| \B                          | Buscar destino a límite de palabra  |
| \b                          | Busca límite de palabra             |
| .                           | Comodín para cualquier carácter     |
| :                           | Coincidencia con cualquier carácter |

- Metacaracteres predefinidos: Los metacaracteres predefinidos son funciones creadas previamente para facilitar el uso de las expresiones regulares, permitiendo abreviar el código de las regex. En la Tabla 2., podemos observar algunos:

*Tabla 2. Metacaracteres predefinidos.*

| Meta caracteres predefinidos |                                      |
|------------------------------|--------------------------------------|
| Meta caracteres              | Descripción                          |
| \w                           | Detecta letras, números y guion bajo |
| \W                           | Detecta caracteres no alfanuméricos  |
| \d                           | Detecta un carácter numérico         |
| \D                           | Detecta un carácter NO numérico      |
| \s                           | Detecta tabulaciones                 |
| \S                           | Detecta No espacios                  |

- Secuencias de escape: Se utilizan para especificar acciones de tabulación o retroceso dentro de una línea de caracteres, algunos comandos de secuencia de escape son \n, \t, los cuales sirven correspondientemente para pasar a la primera posición de la línea siguiente y para pasar a la siguiente posición de tabulación.
- Clases de caracteres: Sirven para buscar caracteres dentro de un rango de posibles opciones. Estas clases son delimitadas por corchetes [ ] y dentro de ellas se encuentra el conjunto de caracteres que queremos convertir en un patrón. Por ejemplo, la clase [^Avenida] encontrará cualquier carácter que no se encuentre en lo descrito en la clase [^Avenida]. Si tenemos una cadena de caracteres llamada

AvC 56, esta clase traerá como resultado C 56. La clase [^0-9] no mostrará ningún número.

- Rangos: Un rango es una clase de caracteres abreviada que se escribe agregando el primer carácter del rango un guion y el ultimo carácter del rango. Por ejemplo [3-7a-c] equivale a escribir [34567abc]. También existen rangos negados los cuales consisten en listar los caracteres que no deben aparecer dentro de la cadena alfanumérica [^3-7a-c].
- Cuantificadores: Los cuantificadores son símbolos que permiten definir cuantas veces puede o no aparecer un patrón o cadena de carácter. Un ejemplo de ello es la expresión [Car?era]. esta expresión permite detectar las cadenas de caracteres Carrera o Carera y desecha las otras posibles opciones. En la Tabla 3., se muestran algunos de los cuantificadores más usados.

*Tabla 3. Metacaracteres cuantificadores.*

| Meta caracteres Cuantificadores |                                                   |
|---------------------------------|---------------------------------------------------|
| Meta caracteres                 | Descripción                                       |
| {n,m}                           | Coincide con n,m ocurrencias donde m es el máximo |
| {n}                             | Coincide con n ocurrencias del patrón             |
| {n,}                            | Coincide por lo menos n veces con el patrón       |
| ?                               | Coincide con 0 o una ocurrencia del patrón        |
| +                               | Coincide con 1 o más ocurrencias del patrón       |
| *                               | Coincide con 0 o más ocurrencias                  |

### 6.6.2. Métodos de las expresiones regulares

En JavaScript, las expresiones regulares también son objetos. Estos patrones se utilizan en los métodos exec y test de RegExp, así como los métodos match, replace, search y split de String. ( MDN Web Docs, 2019). En la Tabla 4. se observa una descripción de ellos:

*Tabla 4. Métodos que utilizan expresiones regulares.*

| MÉTODO  | DESCRIPCIÓN                                                                                                                                         |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| exec    | Un método RegExp que ejecuta una búsqueda por coincidencia en una cadena. Devuelve un array de información.                                         |
| test    | Un método RegExp que verifica una coincidencia en una cadena. Devuelve true o false.                                                                |
| match   | Un método String que ejecuta una búsqueda por coincidencia en una cadena. Devuelve un array de información o null si no existe coincidencia alguna. |
| search  | Un método String que verifica una coincidencia en una cadena. Devuelve el índice de la coincidencia, o -1 si la búsqueda falla.                     |
| replace | Un método String que ejecuta una búsqueda por una coincidencia en una cadena, y reemplaza la subcadena encontrada con una subcadena de reemplazo.   |
| split   | Un método String que utiliza una expresión regular o una cadena fija para cortar una cadena y colocarlo en un array de subcadenas.                  |

## 6.7. Turf

Turf es una librería JavaScript para realizar análisis espacial en la web. Sería algo así como el GIS de los mapas web, incluye operaciones espaciales tradicionales (buffer, TINs, isolíneas...), funciones de ayuda para crear datos en formato GeoJSON y herramientas para clasificar datos y para crear estadísticas.

Turf se puede añadir en una web completamente en el cliente para realizar todas las operaciones o ejecutarse en el servidor con Node.js. Turf utiliza los datos geográficos en formato GeoJSON y en coordenadas geográficas WGS84 (longitud, latitud), es decir el EPSG:4326. (Morales, 2012)

### 6.7.1. Herramientas de análisis que incluye Turf

Con turf, podemos hacer casi cualquier tipo de análisis espacial. Las funciones se organizan en grupos para que sean más fáciles de encontrar. Las herramientas más importantes que tiene Turf son las siguientes:

- Herramientas de agregación. Nos permiten ejecutar operaciones estadísticas en un conjunto de puntos dentro de un conjunto de polígonos. ¿Quieres comparar el tamaño promedio de las viviendas de la provincia de Barcelona? Podemos utilizar la función de agregación de turf. Las herramientas de agregación son: aggregate, average, count, deviation, max, median, min, sum, variance.
- Herramientas de medición. Permiten medir distancias, crear objetos geográficos, y calcular tamaños. Con los métodos de medición, no sólo podemos encontrar el centro exacto de la ciudad de Madrid (utilizando turf-centroid), sino que también

podemos medir la distancia entre ese punto y el del ayuntamiento (turf-distance). Las herramientas de medición son: Along, area, bbox-polygon, bearing, center, centroid, destination, distance, midpoint, point-on-surface, size, square.

- Herramientas de transformación. Con estas herramientas podemos resolver problemas espaciales. Si queremos comprar una casa que esté a menos de 500 metros de un parque y a menos de 1000 metros de una boca de metro, podemos averiguar dónde se superponen esas áreas con turf-intersect, un método de transformación. Las herramientas de transformación son: bezier, buffer, concave, convex, difference, intersect, merge, simplify, union.
- Herramientas de datos. Nos permiten filtrar los datos a partir de propiedades o crear datos para testear. Las herramientas de datos son: filter, random, remove, sample.
- Herramientas de interpolación. Con estas herramientas es posible estimar o medir datos y visualizar el resultado. Un ejemplo de esto es la creación de líneas de contorno (o isobandas) de un conjunto de puntos que contienen información de elevación mediante el uso de turf-isobands. Las herramientas de interpolación son: hex, isobands, isolines, planepoint, point-grid, tin.
- Herramientas unión pueden determinar relaciones espaciales entre objetos espaciales. Las herramientas de unión son: inside, tag, within.
- Herramientas de clasificación pueden clasificar los datos en grupos discretos. Si desea cambiar la propiedad «color» de una serie de features individuales, simplemente, «azul» o «no azul», podemos utilizar el método turf-reclass. Las herramientas de clasificación son: jenks, nearest, quantile, reclass

## **7. MATERIALES Y MÉTODOS**

Teniendo presente la problemática que se identificó en la empresa de telecomunicaciones que presta sus servicios en varias ciudades a nivel nacional, se trabajó con unos materiales y métodos que permitieron diseñar e implementar un editor que se ajustará a las necesidades de la empresa, generando direcciones normalizadas, para sectores del municipio de Cali en la zona de expansión urbana que no contaban con cobertura cartográfica, permitiendo obtener datos espaciales de puntos que representan lugares como casas, edificios, conjuntos residenciales necesarios para el gestor de base de datos utilizado en la empresa de telecomunicaciones. Las fases realizadas para el desarrollo de la herramienta Geocoord fueron:

### **7.1. Materiales**

Para el entorno de prueba de la aplicación se trabajó en un servidor local utilizado por servidor, con sistema operativo Windows 10. Para la implementación de Geocoord se utilizaron las siguientes tecnologías PHP versión 7.3.9., HTML, CSS, JSON y JavaScript.

La base de datos utilizada fue PostgreSQL en la versión 11 con el uso de su módulo Postgis versión 2.5 utilizando la herramienta pgAdmin 4 en la versión 4, que es la parte administrativa para la base de datos PostgreSQL.

Se programó el FrontEnd usando las siguientes librerías:

- ReactJS: Componentes reactivos.
- Leaflet: Librería de gestión del mapa y las interacciones respectivas.
- TurfJS: Librería que permite realizar geoprocursos en el FrontEND (con esta se dibujó, editó y verificaron las capas creadas).
- Redux: La edición del mapa se manejó por medio de estados, permitiendo así tener control y seguimiento de lo que se editaba.
- Material-UI: Estilos de los componentes.
- Axios: Permite conectar el cliente con el backend.

### **7.2. Descripción de la zona de estudio**

El área de estudio se encuentra ubicada en la ciudad de Cali en un área de 1629 hectáreas definidas para el proceso de expansión urbana y consolidación de la ciudad hacia el sur, constituida como la puerta de entrada y salida al sur del país que articula el área urbana de Cali con el municipio de Jamundí, dentro de este sector se encuentran Ciudad Meléndez, Bochalema, Las Vegas de Comfandi, Zonamérica, Las Vegas del Lili entre otros.

Los límites de la zona de expansión son, al margen izquierdo por la Calle 25 (conexión Cali- Jamundí), al oriente por la Calle 61 (borde oriental del área de expansión), al norte por las Carreras 50 (Canal Interceptor Sur) y al sur por la Carrera 143. Figura 6.

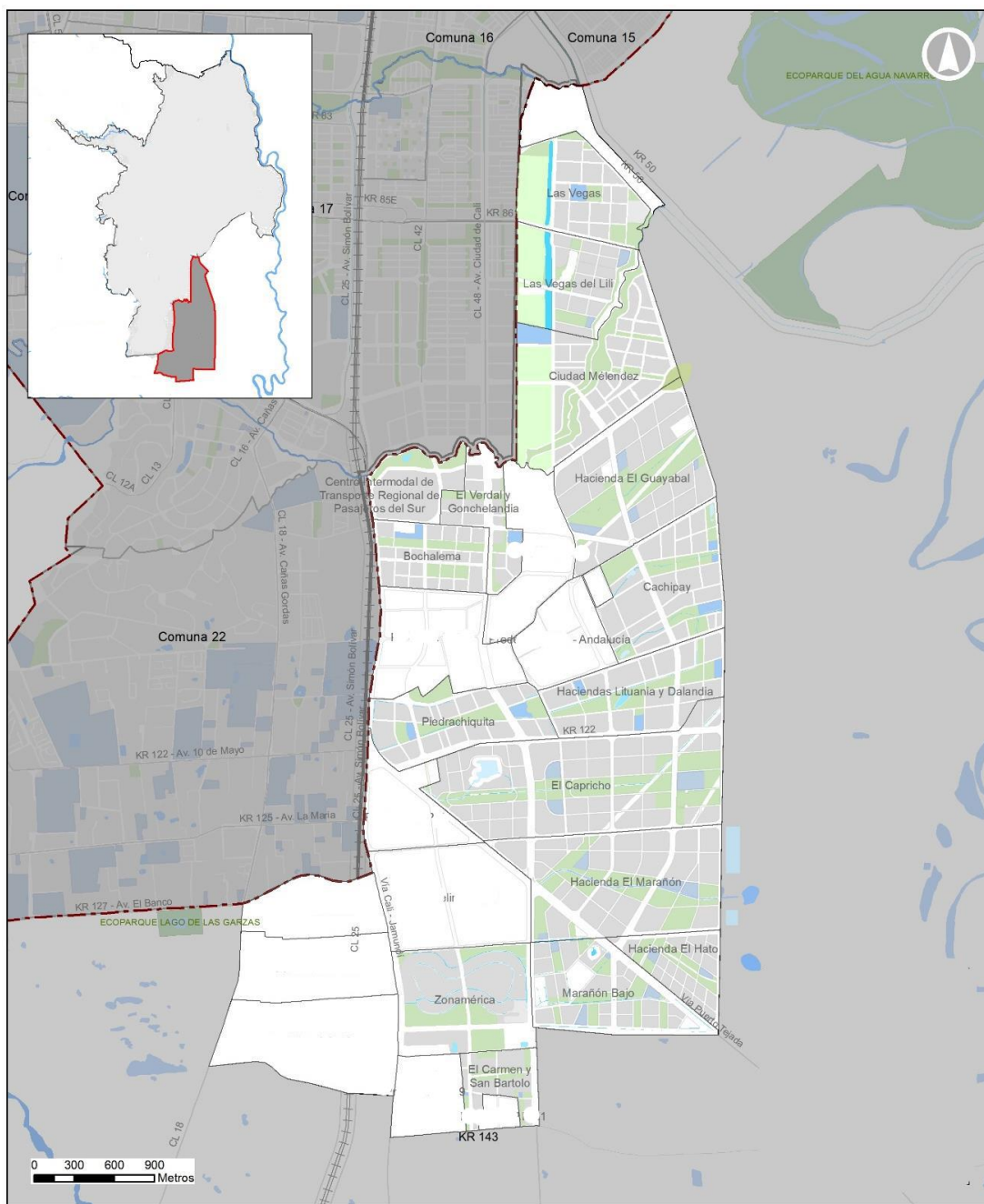


Figura 8. Zona de Expansión.

### **7.3. Recolección de la información**

#### **7.3.1. Recolección de requerimientos de la empresa de telecomunicaciones**

En el mes de septiembre del 2019, se realizó reunión con funcionario de la empresa del área de cartografía, con el fin de conocer la dinámica realizada en los casos de cartografía faltante en la ciudad, donde se obtuvo como resultado los aspectos más importantes en cuanto a sus necesidades con respecto a la optimización de la información para alimentar la base de datos de las zonas con faltante cartográfico. Se definieron los requerimientos de la empresa para automatizar el proceso:

- Reducir tiempos en la ejecución de proceso.
- Reducir la probabilidad de errores en digitalización.
- Desempeñar las actividades del proceso en forma paralela. (Estandarización-Georreferenciación)
- Permitir la consulta de predios generados, con su dirección normalizada y coordenadas teniendo en cuenta los atributos que se requieren y los casos de predios (casas) y conjuntos cerrados(torres).(archivo exportado)
- Posibilidad de ejecutar el módulo en las diferentes sedes de la empresa.

#### **7.3.2. Recolección de información urbana**

Esta fue una fase fundamental en el desarrollo del proyecto, donde se recopiló la información urbana del municipio de Cali que se encuentra en el Geoportal de Infraestructura de Datos Espaciales de Santiago de Cali -IDESC-, específicamente las capas de datos espaciales vectoriales que corresponden a las manzanas y barrios del área urbana del Municipio. Estos datos en el proyecto se utilizaron en formato .shp y son objetos geométricos basados en líneas. Cada objeto geométrico posee sus correspondientes atributos de identificación, como también las coordenadas geográficas que lo conforman. Las coordenadas asociadas a cada objeto geométrico están configuradas según el sistema de referencia espacial WGS84 (EPSG: 4326).

Por parte de la empresa se obtuvo permisos para utilizar el visor de cartografía que permitió visualizar qué zonas se encuentran sin cartografía, el material físico o digital que permitió alimentar el editor en las zonas que no tienen información predial (formato digital en extensión .dwg o .pdf). Estos datos son objetos geométricos basados en líneas, pero a diferencia de los .shp, estos no tienen atributos y manejan sistemas de referencia espacial asumidas; y documento en formato pdf con la normalización de las direcciones urbanas, Anexo 2, que fueron utilizadas posteriormente para la estandarización de direcciones urbanas, para realizar la estandarización se desarrolló con el mínimo de elementos básicos que conforman un registro de dirección.

#### **7.4. Diseño Geocoord**

En la fase de diseño se contó con el apoyo de personas que manejan procesos de desarrollo y poder generar los modelos y procesos que permitieron la construcción de la aplicación como un servidor web, teniendo en cuenta los requerimientos de la empresa. Se considera necesario indicar que el nombre asignado para el aplicativo web service desarrollado en el presente proyecto es Geocoord, ya que así se encontrará en el proyecto para identificar al editor cartográfico. Se explicará de manera general las actividades realizadas en el diseño.

El diseño de Geocoord, se desarrolló en el patrón MVC (Modelo-Vista-Controlador), Se diseñó la base de datos y se construyó en PostgreSQL. El Backend se trabajó con el lenguaje de programación PHP y el FrontEnd fue desarrollado con el lenguaje JavaScript utilizando los lenguajes json, css y html.

En el modelo de datos se utilizó el modelo Entidad – Relación. La estructura básica de datos del modelo relacional es solamente la relación (tabla), donde la información de una determinada entidad (p. ej. "predio") se almacena en filas, cada una con un conjunto de atributos (columnas). Las columnas de cada tabla enumeran los distintos atributos de la entidad (el nombre del "municipio", dirección, número de la placa, entre otros), de modo que cada fila de la relación "predio" representa una dirección específica guardando los datos de ese predio en específico.

El modelo conceptual del sistema de información geográfico permite visualizar a grandes rasgos la estructura funcional. La Figura 7 muestra los componentes del modelo utilizando un servidor REST.

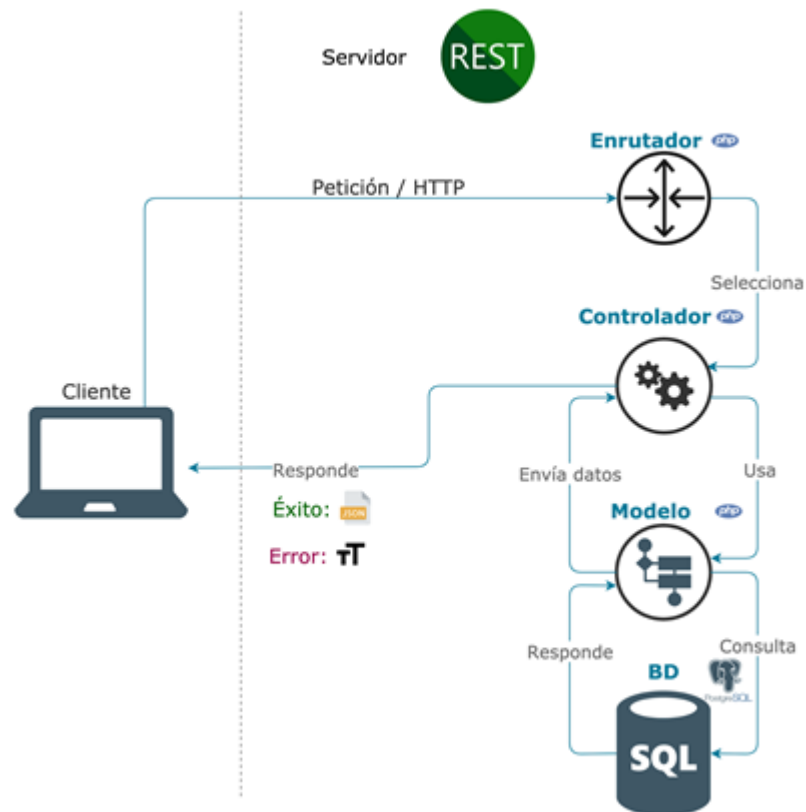


Figura 9. Modelo conceptual con arquitectura REST.

## 7.5. Implementación de la prueba piloto de Geocoord

Esta fase se dividió en dos componentes, teniendo en cuenta la arquitectura: el cliente y el servidor; donde el cliente hace referencia a todos los procesos que se ejecutan en la vista de Geocoord, mientras que el servidor permite interactuar de manera directa con la base de datos y los archivos que se almacenan en la red. Todo el sistema se desarrolló siguiendo el patrón MVC.

Detallando un poco los procesos ejecutados en la implementación se encuentran:

- Sistema de captura, el cual recibe la información digitalizada por el cliente.
- Sistema de análisis y procesamiento, el cual realiza internamente el cálculo de aproximación de coordenadas ingresando los datos necesarios por el cliente.
- Sistema de salida, el cual arroja al cliente direcciones normalizadas con sus coordenadas geográficas cuando se ha solicitado la consulta.

## 7.6. Validación

Para esta última fase se revisa constantemente la retroalimentación del Servicio Web Geocoord. Se ejecuta una evaluación del sistema implementando los ajustes

requeridos como: nombres de calles (ingreso de la vía principal y generadora), transformación de coordenadas geográficas como producto final.

Para la validación del proyecto se tuvo en cuenta dos procesos que concentran los requerimientos de la empresa de telecomunicaciones, estos son la creación de los elementos predios y la estandarización de las direcciones ingresadas en los formularios de Geocoord que se ven reflejadas en el archivo exportado.

Para la prueba de validación se tuvo en cuenta los casos de predios y edificios según malla vial y los predios en conjunto cerrado en torres o casas, tomando como muestra ejemplos de cada caso.

## 8. RESULTADOS

### 8.1. Recolección de la Información

#### 8.1.1. Recolección de requerimientos de la empresa de telecomunicaciones

A continuación, se presenta el resultado de cada uno de los requerimientos que fueron cumplidos por la herramienta Geocoord:

- Reducir tiempos en la ejecución de proceso  
Por medio de la toma de registros de tiempos consignados en un archivo de Excel, se comparó el tiempo tomado con el proceso actualmente en uso y el tiempo tomado con Geocoord. En el anexo 1, se visualizan las cantidades que se tomaron de muestra. En la Figura 8, se visualiza la totalidad de predios y la cantidad de minutos que tomó realizar la actividad con los dos procesos mencionados, resultando que con Geocoord se tiene un 74% de ahorro en el tiempo.

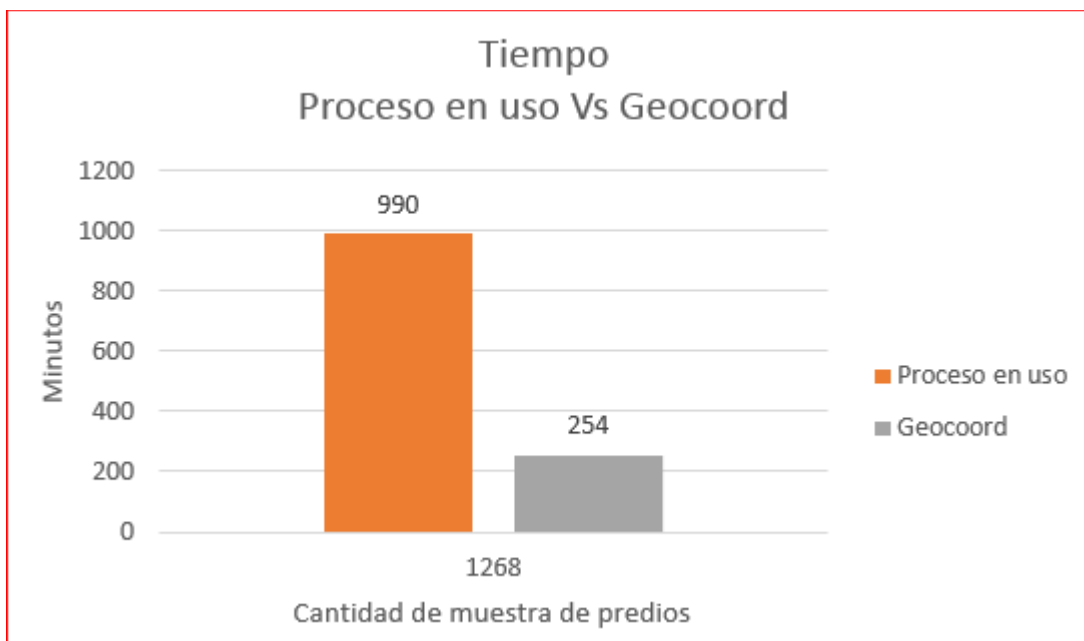


Figura 10. Proceso en uso Vs Geocoord.

- Reducir la probabilidad de errores en digitalización.  
El requerimiento fue analizado teniendo en cuenta que digitar información alfanumérica por medio del proceso en uso, genera pasividad que puede ocasionar mayores errores debido a la rutina y poca actividad del proceso. En el caso de Geocoord, se redujo en lo posible, repetir la información que se puede estandarizar y hacer interactiva la herramienta por el uso de figuras

geométrica, botones con respuestas, ventanas, que generan mayor dinámica en la interfaz de vista del usuario, ver Figura 9.

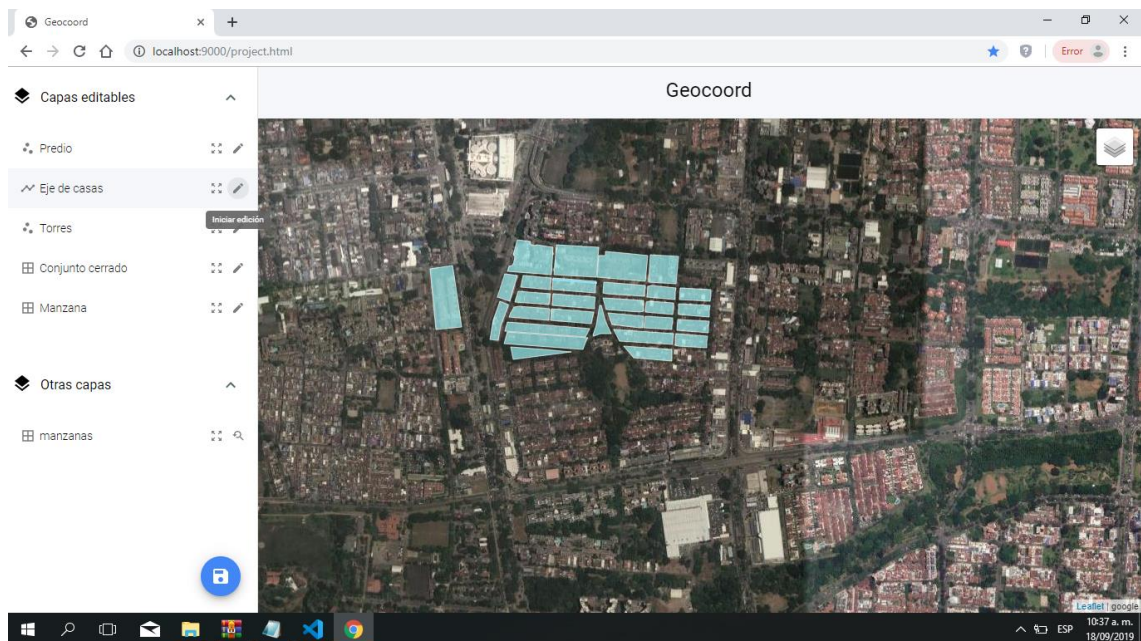


Figura 11. Interfaz Geocoord.

- Desempeñar las actividades del proceso en forma paralela. (Estandarización-Georreferenciación)

El proceso en uso, no permite realizar de manera simultánea los procesos de estandarización y georreferenciación de las direcciones de predios. Como se ha indicado en los requerimientos de la empresa, el proceso en uso primero ingresa de manera manual las direcciones estandarizadas en el archivo de Excel y luego pasa a obtener las coordenadas del predio en Google Map con el marcador de ubicación y con la función copiar, se pegan la longitud y latitud en el archivo de Excel, predio a predio.

Geocoord por medio de los servidores de mapas (API de Google Maps y Open Street Map) y las librerías de Javascript (Leaflet-Tufr) , permite generar la geometría para obtener los puntos que representan la localización de los predios, que mediante un clic sobre esta geometría, se despliega la solicitud de datos que al ingresar la información y ser guardados, se genera de manera automática la asignación de coordenadas de los puntos requeridos, cumpliendo con el requerimiento de desempeñar las actividades de forma paralela a los ojos del usuario. Esta información se puede observar en el archivo exportado de la base de datos.

- Permitir la consulta de predios generados, con su dirección normalizada y coordenadas teniendo en cuenta los atributos que se requieren y los casos de predios (casas) y conjuntos cerrados(torres). (archivo exportado).

Geocoord por medio de la librería Leaflet, exporta el archivo en formato CVS desde la base de datos del servidor, teniendo en cuenta el formato de atributos solicitado por la empresa. Dando como resultado la información para los campos que se encuentran la Figura 10, para casas y conjunto cerrados de casas y para edificios y conjuntos cerrados torres como la Figura 11.

| A                    | B                      | C                   | D      | E       | F    | G       | H        |
|----------------------|------------------------|---------------------|--------|---------|------|---------|----------|
| Dirección del predio | Dirección del conjunto | Nombre del conjunto | Barrio | Manzana | Casa | Latitud | Longitud |

*Figura 12. Archivo cvs- casas y conjunto cerrados casas.*

| A                   | B                      | C            | D                   | E                      | F      | G       | H       | I        |
|---------------------|------------------------|--------------|---------------------|------------------------|--------|---------|---------|----------|
| Nombre del edificio | Dirección del edificio | Apartamentos | Nombre del conjunto | Dirección del conjunto | Barrio | Manzana | Latitud | Longitud |

*Figura 13. Archivo .cvs - Edificio- conjunto cerrado torres.*

- Posibilidad de ejecutar el módulo en las diferentes sedes de la empresa. Geocoord con una estructura de servidor web local permitirá ser utilizada en una red local de la empresa con el objetivo de que los usuarios que tienen la actividad de generar el archivo para actualizar la información de nomenclaturas urbanas georreferenciadas tengan acceso a la herramienta. Actualmente Geocoord solo se encuentra funcionando en un equipo personal debido a que está a modo de pruebas y no se tiene permisos oficiales por parte de la empresa para realizar su uso en la red local de la compañía.

### 8.1.2. Recolección de información urbana

- **Capas shp manzanas-barrios:** Se descargaron las capas manzanas-barrio del municipio de Cali para alimentar la base de datos de Geocoord. Cabe aclarar que en los sectores donde no se visualiza cartografía en el visor de la empresa, se podrán generar las manzanas en Geocoord con los planos digitales o físicos, pero la capa de los barrios no debido a que no se tienen datos para realizar los límites de esta última a causa de que los planos suministrados por la empresa en formato físico o digital no contienen el límite de barrios.
- **Planos DWG-PDF:** Para la recolección de la información se procedió a revisar y organizar los archivos de planos digitales que se han permitido a partir del mes de julio, según instrucción del funcionario con quien se realizó la reunión, para realizar la gestión con el proceso manual utilizado hasta el momento. Debido al deber de proteger la información de la empresa, los archivos solo pueden ser guardados en el equipo que se tiene asignado dentro de la empresa.

En el disco D:/ del equipo de la empresa, se generaron carpetas para guardar los archivos con la siguiente categorización, ver Figura 12:

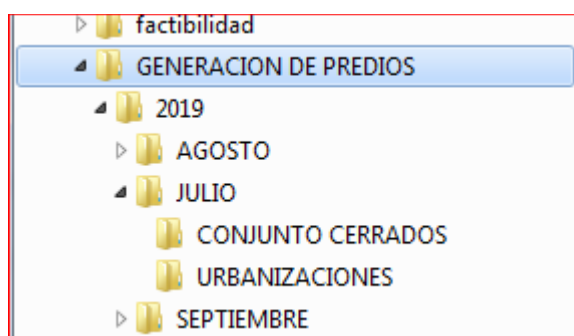


Figura 14. Carpetas de Insumos.

- **Modelo de estandarización de direcciones urbanas:** Para las direcciones según la malla vial, se trabajó con Tipo de vía, nombre o número de vía, prefijo o cuadrante, número de vía generadora, prefijo o cuadrante de la vía generadora y un número de placa. Por ejemplo: Calle 11 Sur Número 23 A BIS – 50.

En el caso de conjuntos cerrados de torres o edificios se trabajó con el nombre del conjunto, torres, pisos y apartamento, también se tiene en cuenta la dirección según la malla vial como atributo opcional. Por ejemplo: Conjunto Residencial Las Palmas Torre A Apartamento 201

Teniendo en cuenta lo anterior se presenta en la Tabla 5. las abreviaturas utilizadas para realizar la estandarización:

Tabla 5. Estandarización de abreviaturas de direcciones.

| Elemento             | Abreviatura |
|----------------------|-------------|
| CARRERA              | KR          |
| CALLE                | CL          |
| AVENIDA              | AV          |
| TRANSVERSAL          | TV          |
| DIAGONAL             | DG          |
| AUTOPISTA            | AU          |
| EDIFICIO             | ED          |
| CONJUNTO RESIDENCIAL | CR          |
| TORRE                | TO          |
| APARTAMENTO          | AP          |
| NORTE                | N           |
| SUR                  | S           |
| ESTE                 | E           |
| OESTE                | O           |

## 8.2. Diseño

### 8.2.1. Modelo de Datos

Luego de obtener los requerimientos de la empresa y la recolección de información, se define el modelo de datos para el almacenamiento de esta información. Se realiza el modelo de la base de datos con 6 tablas como se aprecian en la Figura 13, con su relaciones y atributos:

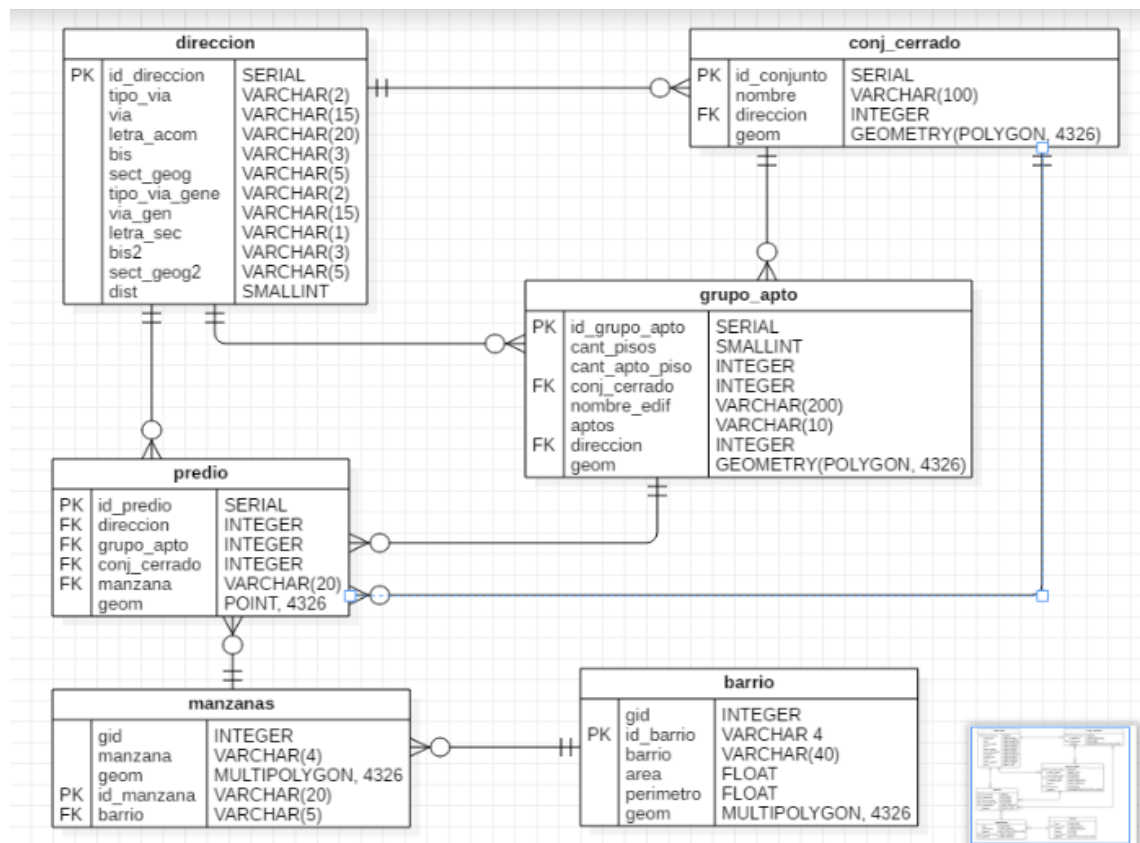


Figura 15. Diseño base de datos-Tablas.

En la figura anterior se puede apreciar que las tablas de las entidades conj\_cerrado, grupo\_apto, predio tienen una relación de 0 a varios con solo uno, con respecto a la entidad dirección, lo que quiere decir que cada una de las primeras entidades nombradas solo puede tener una dirección. Igual sucede con la entidad predio, varios predios pueden hacer parte de solo un grupo de apartamentos y varios grupos de apartamentos pueden hacer parte de un solo conjunto cerrado. También se tiene la relación de que varios predios pueden hacer parte de solo una manzana y varias manzanas pueden hacer parte de un solo barrio.

### 8.2.2. Modelo Conceptual

La estructura utilizada se montó sobre un servidor con sistema operativo Windows, en el cual se instaló el motor de base de datos PostgreSQL con su módulo espacial para administrar los datos geográficos(postgis). A continuación, se documenta en un lenguaje menos técnico cómo se logra orquestar todos los componentes del Geocoord.

#### **a. Cliente(FrontEnd)**

En el patrón de diseño utilizado en el desarrollo de la herramienta GeoCoord MVC (Modelo vista Control), las vistas son la unión de varias tecnologías del FrontEnd tales como los lenguajes HTML, CSS, JAVASCRIPT, MATERIAL-UI que juntas en sincronía son las encargadas de recibir y transmitir al controlador las interacciones que el usuario puede llegar a realizar en el aplicativo mediante el navegador, Este conjunto de tecnologías son conocidas en el ambiente de desarrollo como frontEnd, ya que están de frente con la interacción del usuario.

Es importante resaltar que en el cliente es donde se encuentra alojado o se ve reflejado la estandarización de las direcciones usando expresiones regulares. Esto se implementó en javascript y lo que hace es reconocer con un patrón definido todas las posibles combinaciones que alguien pudiese estar escribiendo y cambiar automáticamente a una sentencia que se necesite. También en el cliente se ve reflejado el proceso de digitalización, donde por medio de figuras geométricas se realiza la interpolación de los puntos de predio, por medio de librerías de Javascript.

#### **b. Comunicación (Cliente - Controlador)**

La comunicación entre el frontend y el controlador se realizó mediante services REST lo cual desacopla el frontend con el backend, esto quiere decir que se puede realizar cambios en el backend sin necesariamente modificar el frontend, Esta comunicación se soporta bajo http y se usaron los métodos POST y GET.

- GET: Se usa para solicitar consultar a los recursos
- POST: Se usa para insertar nuevos recursos

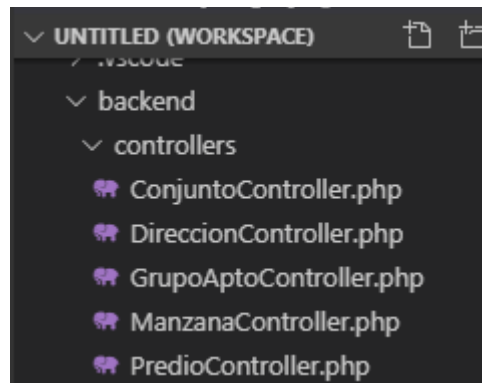
#### **c. Controlador (BackEnd)**

El controlador, como su nombre lo dice es el que recibe las peticiones del frontend y controla los llamados a los distintos modelos que se usaron, allí se plasma todo el requerimiento inicial y es quien supe y los hace realidad.

Ahí también están alojados los Web Services que los usa el frontend para tener comunicación en el backend. El backend fue desarrollado en PHP un lenguaje de programación muy popular en el desarrollo web.

#### **d. Comunicación (Controlador - Modelo)**

La comunicación entre el controlador y el modelo se realiza mediante .dll (biblioteca de conexión) ya que el desarrollo se realizó sobre el sistema operativo Windows y en una aplicación standalone o ambiente de pruebas. PHP lenguaje en el cual está realizado el backend se conecta a la base de datos Postgres y mediante esta biblioteca realiza las acciones en la base de datos según lo que el controlador le indique, Como por ejemplo guardar o eliminar algún dato. En la Figura 14 se pueden observar los controladores creados para el backend:



*Figura 16. Controladores de Backend de Geocoord.*

#### **e. Modelo (Como backend)**

El modelo se constituye de todas las tablas que fueron necesarias crear para poder contener toda la información. En este modelo fue donde se modelaron todos los métodos que cada tabla está en facultad de realizar, esto a su vez está a disposición del controlador para que el haga uso de esto y haga las veces de orquestador según la lógica de negocio que se va a aplicar. El modelo de Geocoord se compone de los siguientes elementos que se observan en la Figura 15:

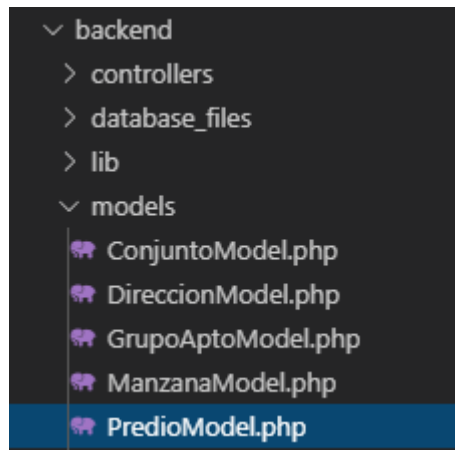


Figura 17. Modelos de Backend de Geocoord.

### 8.2.3. Proceso de estandarización

Ampliando los procesos que se realizan en el frontend y que hacen parte esencial del proyecto, se debe hablar del proceso de estandarización, aquí se tuvo en cuenta varias etapas que realiza Geocoord para el ingreso de las direcciones de los predios antes de guardarse en la base de datos del servidor y se hace mediante expresiones regulares -regex-, el algoritmo se puede observar en el Anexo 3 :

#### a. Ingreso de direcciones

Para el caso de las direcciones urbanas asignadas según malla vial se ingresa la dirección urbana solicitada. Esta debe tener tres componentes: vial principal, vía generadora y distancia. Ej. Calle 15 # 45-03

#### b. Estandarización de la dirección

En la etapa de estandarización de direcciones se reemplazan las nomenclaturas y símbolos reconocidos en la dirección urbana ingresada con las sugeridas por la propuesta de estandarización. En la Tabla 6. se pueden observar algunos ejemplos de reemplazo:

Tabla 6. Estandarización de abreviaturas de direcciones.

| Símbolos/Nomenclaturas reconocidos                                                 | Símbolo reemplazo   |
|------------------------------------------------------------------------------------|---------------------|
| Texto en minúsculas                                                                | Texto en Mayúsculas |
| KRA K KR CARRERA C CA CAR CARR CARRE CARRER CRA CRR CRR CR                         | KR                  |
| CALLE CAL CALL CLL CL                                                              | CL                  |
| AVENIDA AV AVE AVEN AVENI AVENID AVN AVNIDA                                        | AV                  |
| TRANSVERSAL T TR TRA TRAN TRANS TRANSV TRANSVE TRANSVER TRANSVERS TRANSVERSA TR TV | TV                  |
| DIAGONAL D DI DIA DIAG DIAGO DIAGON DIAGONA DG DGNAL DGN DNA                       | DG                  |
| AUTOPISTA A AU AUT AUTO AUTOP AUTOPI AUTOPIIS AUTOPIST AT                          | AU                  |

### **c. Reconocimiento de los componentes adicionales de la dirección urbana.**

Con el fin de obtener los componentes mínimos de la dirección urbana ingresada y eliminar cualquier tipo de texto adicional, se incluyeron otros elementos a considerar en la dirección, como son la letras, el sufijo y el cuadrante, el algoritmo reconocerá como correctos estos elementos de nomenclatura, de lo contrario generará un error.

Se adjunta una línea del algoritmo donde se puede observar las etapas que se mencionan mediante expresiones regulares:

“...let template =

```
/(?<typeStreet1>\D\S*)?(.)?s?(?<street1>'.*\w+)?s?(?<bis1>bis|bi|b)?s?  
(?<cuadrante1>norte|n|no|nor|nort|nt|nte|sur|s|su|sr|este|e|es|est|et|oeste|o  
|oe|oes|pest|o|ot|ost|os)?,s?(?<typeStreet2>\D\S*)?(.)?s?(?<street2>'.*\w+)  
)?s?(?<bis2>bis|bi|b)?s?(?<cuadrante2>norte|n|no|nor|nort|nt|nte|sur|  
s|su|sr|este|e|es|est|et|oeste|o|oe|oes|pest|o|ot|ost|os)?s?(-  
)?s?(?<placa>d*)?/i ...”
```

### **8.2.4. Proceso digitalización direcciones urbanas**

El proceso de digitalización de direcciones por medio de vectores, en Geocoord, se realizó mediante la librería Turf junto con la librería leaflet, la primera permitió realizar operaciones espaciales en la web mediante elementos geográficos sobre el map web interactivo creado con leaflet.

Mediante la interpolación de puntos a partir de una geometría línea base se obtiene como resultado un conjunto de puntos, cada punto tendrá los atributos de las direcciones urbanas según se especifique en la previa configuración de la interpolación. En casos de unidades compuestas por torres, se dibuja mediante un polígono los límites de conjunto cerrado y se generarán las torres con geometría punto. Cabe resaltar que las torres deben tener sus atributos que permitan generar como resultado la numeración de sus apartamentos como se requiere para el archivo exportado.

## **8.3. Implementación de la prueba piloto de Geocoord**

### **8.3.1. Despliegue de Geocoord**

La aplicación se despliega a partir de un servidor WEB, que se debe levantar utilizando una línea de código, mediante una herramienta de editor de código fuente, para Geocoord se trabajó con Visual Studio Code. La línea de código es “npm run build:dev”, que se debe correr en la ruta de la carpeta htdocs>/geocoder-master/geocoder. Ver Figura 16:

```

Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. Todos los derechos reservados.

PS C:\xampp\htdocs\geocoder-master> cd geocoder
PS C:\xampp\htdocs\geocoder-master\geocoder> npm run build:dev

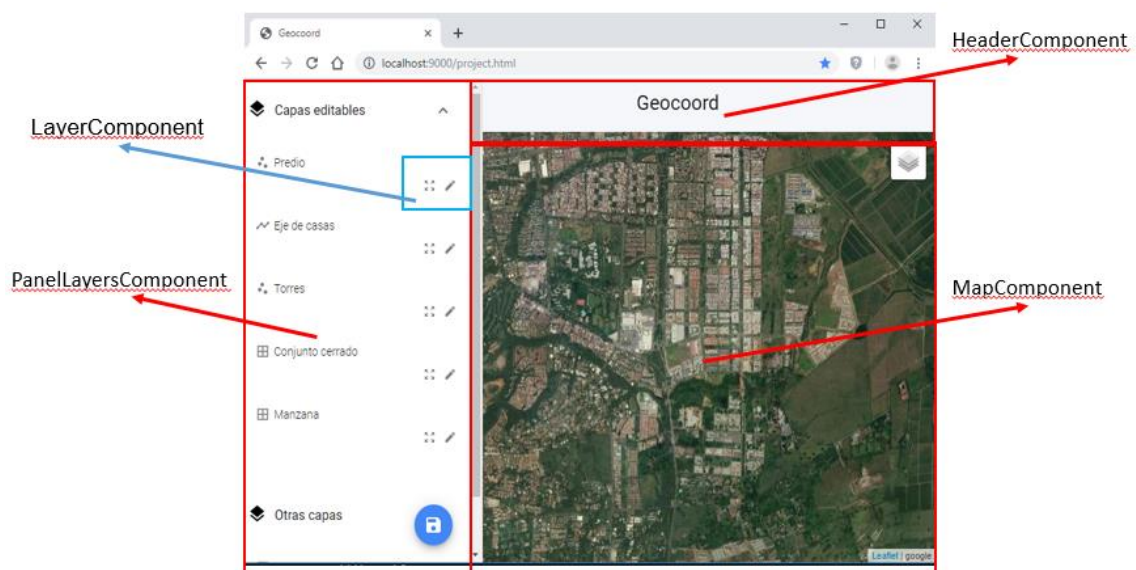
```

*Figura 18. Línea de código para el despliegue de Geocoord.*

Este comando se encarga de empaquetar el código fuente en un único archivo para su disposición en un servidor web y poder ser accesible desde el protocolo http. Este proceso es indispensable para el funcionamiento del aplicativo y es lo que permite que el aplicativo pueda ser accedido mediante la web.

### 8.3.2. Interfaz Geocoord

La interfaz de Geocoord está compuesta por el conjunto de elementos que se ven en la pantalla y que permiten al usuario realizar acciones. El aspecto de la interfaz señalando sus componentes se observa en la Figura 17 :



*Figura 19. Interfaz Geocoord-Componentes.*

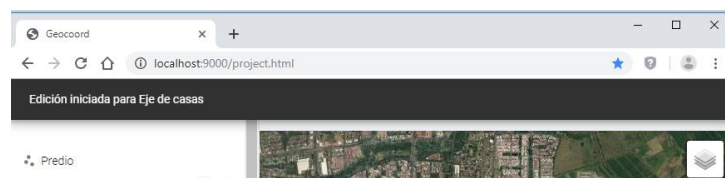
Geocoord contó con dos componentes principales para su interfaz:

- **ProjectContainer**: Se encarga de procesar todo lo relacionado con la interfaz del usuario, como ventanas modales, barras de notificación, encendido y apagado de botones, despliegue y ocultación de menús, etc

- **MapContainer:** Se encarga de procesar toda la interacción con el mapa, como la edición, encendido y apagado de capas, Zoom a las capas y acciones desplegadas por eventos (click, doble click y sobreposición del mouse) en los objetos.

A parte de estos, se programaron otros que solo contienen la parte visual de la aplicación, ya que toda su interacción es administrada por ProjectContainer. Estos componentes son:

- ProjectComponent: contiene todos los subcomponentes de la aplicación (todos los que se mencionan a continuación incluyendo a MapContainer).
- HeaderComponent: Solo contiene el título de la aplicación.
- MapComponent: Contiene el mapa.
- PanelLayersComponent: Contiene el panel con las capas y el botón flotante para enviar la información al servidor.
- LayerComponent: Se replica según la cantidad de capas existentes y contiene los botones de inicio de edición y zoom de la capa.
- ModalComponent: Es un componente reusable, por lo que su contenido es dinámico y varía de acuerdo a la capa asociada. Solo se muestra cuando se dispara un evento click en un elemento de alguna capa de edición (si esta no se encuentra con la edición habilitada desaparece). Ver Figura 18.



*Figura 20. Componente modal.*

- ModalExportComponent: Contiene la opción para descargar los datos consultados en formato CSV, teniendo en cuenta los datos solicitados por la empresa y otra para eliminar la capa de la base de datos. Ver Figura 19.

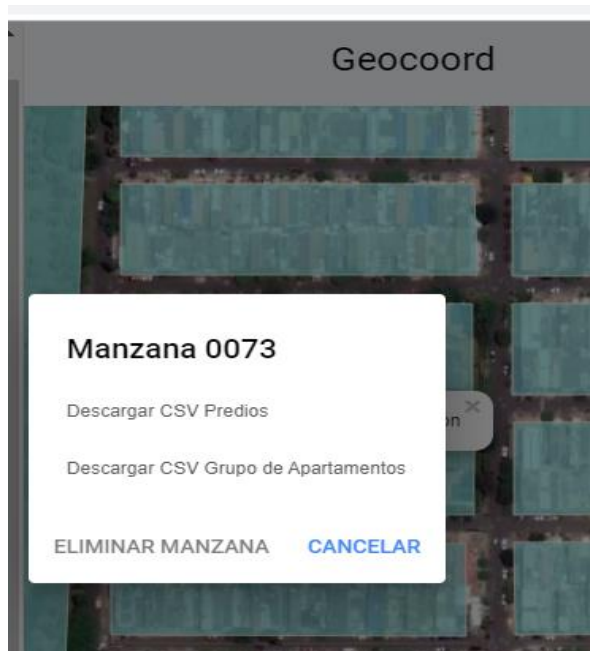


Figura 21. Componente modelo exportación.

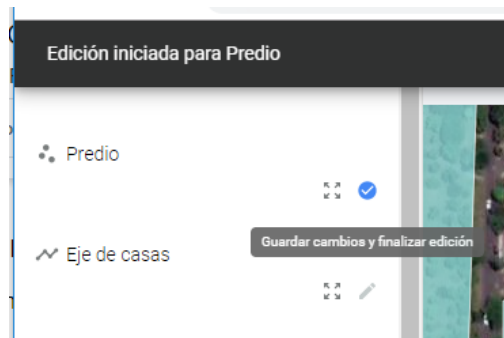
La relación entre todos los componentes se muestra en la siguiente estructura JSON:

```
Geocoord = {
  ProjectContainer : {
    ProjectComponent : {
      HeaderComponent,
      modalComponent,
      ModalExportComponent,
      MapContainer : {
        MapComponent,
        PanelLayersComponent : {
          LayerComponent : []
        }
      }
    }
  }
}
```

### 8.3.3. Funcionalidad Geocoord

A continuación, vamos a enumerar la funcionalidad del servidor web Geocoord:

#### a. Gestión de la visualización (activar/desactivar capas, cambios estéticos)



*Figura 22. Visualización de capa activa/desactiva.*

Mediante un click se puede determinar el encendido o apagado de las capas sobre ella poder añadir nuevos elementos. Ver Figura 20. También existen cambios que modifican la estética de una gráfica determinada (color y grosor de línea, color y nivel de transparencia del relleno). Además, se pueden visualizar intermitencia en gráficas que brindan información, ya sea en los botones de editar y zoom o en datos de elementos geométricos.

### ***b. Navegación***

Geocoord cuenta con las funciones más habituales de navegación por la cartografía, como son:

- Zoom + a un punto.
- Zoom – a un punto.
- Zoom ventana.
- Desplazamiento de la cartografía.
- Vista inicial. Zoom +/- con la rueda del ratón.
- Al editar elementos lineales o puntuales podemos utilizar las funciones de navegación sin por ello interrumpir la edición.

### ***c. Selección de elementos***

Gráficamente mediante un click o mediante una ventana podemos seleccionar elementos pertenecientes a cualquiera de las capas actualmente visibles. Ver Figura 21.

Geocoord

**Formulario**

Antes de continuar, por favor asegurese de que la información es veráz

Nombre de la torre. Ingresar abreviatura TO + nombre de la torre \*

Cantidad de pisos \*

Direccion \*

Cantidad de apartamentos por piso \*

0

CANCELAR GUARDAR

*Figura 23. Selección de elementos.*

La función de selección permite ejecutar otra serie de funciones, por ejemplo: desplazar los elementos seleccionados, ingresar datos, obtener información de ellos. También cuenta con el layer “otras capas”, que hasta el momento solo maneja la capa manzanas, y nos permite visualizar en la vista que se tenga del map, las manzanas que se encuentran en la base de datos, esta herramienta es útil para realizar el exportado de la información.

#### **d. Funciones de edición**

Geocoord cuenta con las siguientes funciones de edición:

- Creación de nuevos elementos, correspondientes a la capa activa. El comportamiento varía en función del tipo de elemento contenido en dicha capa: Para el caso de los predios y torres, estos pueden ser desplazados, guardados o eliminados, Para elementos lineales, como el eje de casas, cada segundo clic da como finalizado el extremo de la línea, esta se puede desplazar moviendo cada vértice, se puede guardar o eliminar. Para polígonos, como las manzanas y los conjuntos cerrados, se comportan como los lineales salvo que al clicar cerca del primer punto se cierra y crea el elemento. Ver Figura 22.



*Figura 24. Representaciones de elementos.*

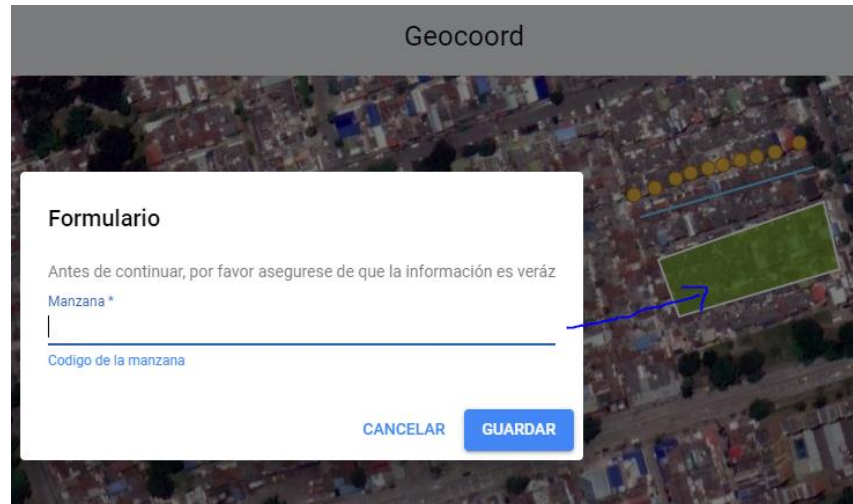
- Edición de vértices de elementos lineales y superficiales, a cualquiera de ellos se les puede desplazar por medio de sus vértices, los cuales según su movimiento van creando nuevos vértices intermedios.

Se debe tener en cuenta que los cambios de desplazamiento y eliminación de las geometrías se hacen estando activa la edición y solo se pueden realizar antes de guardar en el servidor.

#### ***e. Ingreso de datos formulario***

Cada elemento gráfico creado requiere de ingreso de datos para poder ser guardado en el servidor. A continuación, se visualizarán cada uno de los formularios que aparecen como ventanas flotantes al clickear sobre la geometría:

- Elemento Manzana: El atributo solicitado, es el código de la manzana. que para nuestro caso es un consecutivo que puede ser extraído de la base de datos de geocoord. Ver Figura 23.



*Figura 25. Formulario elemento manzana.*

- Elemento conjunto cerrado: contiene los atributos nombre y dirección. Ver Figura 24.



*Figura 26. Formulario elem. conjunto cerrado.*

- Elemento torres o edificios: contiene los atributos de nombre de la torre, cantidad de pisos, dirección y cantidad de apartamentos por piso. Ver Figura 25.

**Geocoord**

**Formulario**

Antes de continuar, por favor asegurese de que la información es veráz

Nombre de la torre. Ingresar abreviatura TO + nombre de la torre \*

---

Cantidad de pisos \*

---

Direccion \*

---

Cantidad de apartamentos por piso \*

---

CANCELAR GUARDAR

*Figura 27. Formulario elemento Torres.*

- Elemento eje de casas: permite crear los predios por medio de interpolación sobre un eje de línea. Para generar las direcciones de los predios, los hace mediante los atributos ingresados, cantidad de predios en el segmento, vía principal, vía generadora, distancia desde la esquina (1 para generar placas impares- 0 para generar placas pares) y valor inicial para enumerar los predios. Ver Figura 26.

**Geocoord**

**Formulario**

Antes de continuar, por favor asegurese de que la información es veráz

Cantidad de predios en el segmento \*

---

Vía principal \*

KR 50B (bis) o 'Av Simón Bolívar'

nombre de la vía sobre la que se encuentra el eje

Vía generadora \*

KR 50B (bis) o 'Av Simón Bolívar'

nombre de la primer vía que intersecta la vía principal en la cuadra

Distancia desde la esquina \*

---

distancia del inicio del predio con respecto a la esquina. \* Para generar placas impares digite un valor impar (1 si es desde la esquina). Para generar placas pares digite un valor par (0 si es desde la esquina).

CANCELAR GUARDAR

*Figura 28. Formulario elemento Eje de casas.*

- Elemento predio: permite convertir un predio en un edificio, solicitando los mismos valores del elemento torres.

## 8.4. Validación

En esta fase se realizó la exportación de los archivos en formato CVS, validando que la dirección ingresada por el usuario se encuentre estandarizada y con sus coordenadas, teniendo en cuenta el requerimiento de datos solicitados por la empresa.

*Tabla 7. Exportado predios malla vial.*

| Dirección del predio | Dirección del conjunto | Nombre del conjunto | Barrio | Manzana | Casa | Latitud     | Longitud     |
|----------------------|------------------------|---------------------|--------|---------|------|-------------|--------------|
| KR 85 CL 48A - 4     |                        |                     | Caney  | 94      |      | 3.381899937 | -76.51531383 |
| KR 85 CL 48A - 12    |                        |                     | Caney  | 94      |      | 3.38189705  | -76.51525622 |
| KR 85 CL 48A - 18    |                        |                     | Caney  | 94      |      | 3.381898029 | -76.5151832  |
| KR 85 CL 48A - 26    |                        |                     | Caney  | 94      |      | 3.381888906 | -76.51511955 |
| KR 85 CL 48A - 34    |                        |                     | Caney  | 94      |      | 3.381884533 | -76.51506022 |
| KR 85 CL 48A - 40    |                        |                     | Caney  | 94      |      | 3.381886548 | -76.514993   |
| KR 85 CL 48A - 48    |                        |                     | Caney  | 94      |      | 3.381884683 | -76.5149319  |
| KR 85 CL 48A - 56    |                        |                     | Caney  | 94      |      | 3.3818872   | -76.51485851 |
| KR 85 CL 48A - 62    |                        |                     | Caney  | 94      |      | 3.381884393 | -76.51480655 |
| KR 85 CL 48A - 70    |                        |                     | Caney  | 94      |      | 3.381886259 | -76.51473819 |
| KR 85 CL 48A - 76    |                        |                     | Caney  | 94      |      | 3.381880411 | -76.51469086 |
| KR 85 CL 48A - 84    |                        |                     | Caney  | 94      |      | 3.381882734 | -76.51463742 |
| KR 85 CL 48A - 92    |                        |                     | Caney  | 94      |      | 3.381879806 | -76.5145687  |
| KR 85 CL 48A - 98    |                        |                     | Caney  | 94      |      | 3.381883551 | -76.51452291 |
| KR 85 CL 48A - 106   |                        |                     | Caney  | 94      |      | 3.381891622 | -76.51446313 |
| KR 85 CL 48A - 114   |                        |                     | Caney  | 94      |      | 3.38188947  | -76.51440696 |
| KR 85 CL 48A - 120   |                        |                     | Caney  | 94      |      | 3.381888477 | -76.51432435 |

En la Tabla 7, se observa la información solicitada por la empresa para el caso de predios en los campos: Dirección predio-Manzana-Latitud-Longitud, el campo barrio no es obligatorio, pero si la base de datos de Geocoord tiene la información es aceptada para el proceso de actualización en la base de datos corporativa. Los campos dirección de conjunto y nombre de conjunto se encuentran con celdas vacías ya que estos campos se exportan con información cuando se trata de un conjunto residencial.

*Tabla 8. Exportado conjunto cerrado torres.*

| Nombre del edificio | Dirección del edificio | Apartamentos                                                                          | Nombre del conjunto | Dirección del conjunto | Barrio | Manzana | Latitud  | Longitud   |
|---------------------|------------------------|---------------------------------------------------------------------------------------|---------------------|------------------------|--------|---------|----------|------------|
| TO A                |                        | 101,102,103,104,201,202,203,204,301,302,303,304<br>,,401,402,403,404,,501,502,503,504 | CR VERDE CANEY      | CI 45 CR 83D - 40      | Caney  | 101     | 3.382576 | -76.518705 |
| TO B                |                        | 101,102,103,104,201,202,203,204,301,302,303,304<br>,,401,402,403,404,,501,502,503,504 | CR VERDE CANEY      | CI 45 CR 83D - 40      | Caney  | 101     | 3.382558 | -76.519052 |
| TO C                |                        | 101,102,103,104,201,202,203,204,301,302,303,304<br>,,401,402,403,404,501,502,503,504  | CR VERDE CANEY      | CI 45 CR 83D - 40      | Caney  | 101     | 3.383    | -76.518689 |
| TO D                |                        | 101,102,103,104,201,202,203,204,301,302,303,304<br>,,401,402,403,404,501,502,503,504  | CR VERDE CANEY      | CI 45 CR 83D - 40      | Caney  | 101     | 3.383014 | -76.518959 |

En la Tabla 8, se observa la información solicitada por la empresa para el caso de conjunto cerrado de torres en los campos: Nombre del edificio-Apartamentos-Nombre del conjunto-Dirección del conjunto-Manzana-Latitud-Longitud, se estandarizó la dirección y se extrajeron las coordenadas de la ubicación de cada torre con sus respectivos apartamentos cumpliendo con la solicitud de la empresa. El campo

dirección edificio se encuentra vacío ya que es para el caso de una edificación vertical que se encuentre fuera de un conjunto cerrado.

## 9. CONCLUSIONES

El resultado de este proyecto cumplió con los requerimientos solicitados por la empresa de telecomunicaciones que se podían resumir en obtener una herramienta que optimizará el proceso de obtener actualización de nomenclatura urbana en sectores de faltante cartográfico para luego ser usado en sus herramientas, sin embargo Geocoord no solo es de utilidad para la empresa, podría ser utilizado en cualquier ámbito que requiera generar direcciones georreferenciadas, ya que su desarrollo mediante software libres permite que se pueda implementar sin ningún tipo de restricción.

En el caso de la nomenclatura urbana se considera de alta importancia que este actualizada para estudios de cualquier índole, pero se presenta un problema en la limpieza de las direcciones por ello la necesidad de estar acompañada de procesos de estandarización, no obstante, con el desarrollo del proyecto se identificó que las correcciones de errores en textos influyen en el resultado de una validación. En el proyecto se dio por cumplida la estandarización debido a que la petición de la empresa para la prueba de Geocoord eran direcciones de poca complejidad, a pesar de ello se recomienda continuar revisando reglas de validación que garanticen optimizar la estandarización. En trabajos a futuro no solo se debe pensar en la calidad de las direcciones urbanas sino también pensar en el manejo de direcciones rurales, en busca de formatos que estandarice la asignación de direcciones para estas zonas.

Al revisar el marco conceptual y teórico se conoce la relación estrecha que tienen los desarrolladores o ing. de sistemas con los ing. topográficos debido que permiten generar sistemas completos debido a los productos que se encuentran en el mercado y que se han convertido en uso común. El uso de los mapas web, las bases de datos con módulos espaciales, los lenguajes de programación con librerías para el manejo de información espacial, crean la necesidad de interactuar. Geocoord es el resultado de esta combinación, contando con apoyo de personas con conocimiento en desarrollos web, fue posible obtener un producto, que se puede mejorar, considerando esta entrega una versión 1.

Debido a que la empresa de telecomunicación exige manejo confidencial, se tuvo limitaciones en el manejo de la información, el acceso a ella es solo dentro de la instalaciones de la empresa y en los equipos que son asignados en los puestos de trabajo, esto generó algunos inconvenientes como por ejemplo el acceder a la información de la cartografía existente en la base de datos de la empresa, al manejo de la recolección de datos que solo puede estar en equipos de la empresa, el no poder instalar la herramienta en el equipo asignado, generaba limitaciones de tiempo y de alcance del proyecto.

Como ingeniera topográfica daría como conclusión final que el estudio espacial en la necesidad de conocer dónde se encuentra ubicado cada objeto o persona nos

invita a indagar en la gran utilidad de los SIGs y a la interacción que tiene con la tecnología en busca de que pueda ser compartido trabajando de la mano con servicios Web, van dando como resultado el perfeccionamiento no de una ciencia sino de la unión de la diferentes ciencias, por lo que es un buen momento para prepararse en el conocimiento de estos temas y en el manejo de sus utilidades.

## 10. BIBLIOGRAFIA

- ANDI (2013). Normatividad urbana y uso del suelo. Asociación Nacional de empresarios de Colombia. Recuperado de:  
<http://webcache.googleusercontent.com/search?q=cache:http://proyectos.andi.co.m.co/es/GAI/GuiInv/CPR/Paginas/NUUS.aspx&strip=0&vwsr=0>
- ArcGis (2016). ¿Qué es la geocodificación?. Recuperado de:  
<http://desktop.arcgis.com/es/arcmap/10.3/guide-books/geocoding/what-is-geocoding.htm#>
- Axarnet. (2018). Cómo crear un servidor en local sin ser un experto. Recuperado de:  
<https://www.axarnet.es/blog/crear-servidor-local/>
- Chapaval, N. (2017). ¿Qué es FrontEnd y BackEnd?. Recuperado de:  
<https://platzi.com/blog/que-es-frontend-y-backend/>
- CIAF(2009). Propuesta de Estándar de las Direcciones Urbanas para los Equipamientos del Ministerio de Educación. Centro de investigación y desarrollo. Recuperado de: [https://www.mineducacion.gov.co/1621/articles-193290\\_estandar\\_direcciones\\_urbanas.pdf](https://www.mineducacion.gov.co/1621/articles-193290_estandar_direcciones_urbanas.pdf)
- Código Integral (2015). Funcionamiento de un servidor Web. Recuperado de:  
<https://codigointegral.wordpress.com/2015/09/25/funcionamiento-de-un-servidor-web/>
- Consejo de Santiago de Cali (2014). Acuerdo No. 0673 Por medio del cual se adopta la revisión ordinaria de contenido de largo plazo del plan de ordenamiento territorial del municipio de Santiago de Cali. Recuperado de:  
<https://sidap.cvc.gov.co/sites/default/files/2019-03/7.%20Acuerdo%200373%20Cali%20de%202014%20POT-SIMAP.pdf>
- Departamento Administrativo de Planeación Municipal (2019). Guía para la estandarización de la nomenclatura urbana. Alcaldía de Santiago de Cali. Recuperado de:  
[http://idesc.cali.gov.co/download/nomenclatura/guia\\_nomenclatura\\_cali\\_2019.pdf](http://idesc.cali.gov.co/download/nomenclatura/guia_nomenclatura_cali_2019.pdf)
- Departamento Administrativo de Planeación Municipal (2019). Guía para la estandarización de nomenclatura urbana. Recuperado de:  
[http://idesc.cali.gov.co/download/nomenclatura/guia\\_nomenclatura\\_cali\\_2019.pdf](http://idesc.cali.gov.co/download/nomenclatura/guia_nomenclatura_cali_2019.pdf)
- Gallego, P. y Suárez, J. (2017). Análisis del proceso de concentración urbana de Colombia: origen, evolución e impacto en el desarrollo económico. Recuperado de:  
[http://repository.lasalle.edu.co/bitstream/handle/10185/20689/10121087\\_2017.pdf?sequence=1](http://repository.lasalle.edu.co/bitstream/handle/10185/20689/10121087_2017.pdf?sequence=1)

GaussWebApp(2014). La arquitectura Rest: La Arquitectura del momento. Recuperado de: <https://gausswebapp.com/arquitectura-rest.html>

Hernández,P. (2010) Bases de datos. Recuperado de: [https://www.ecured.cu/Bases\\_de\\_datos](https://www.ecured.cu/Bases_de_datos)

IDESC (2019). Geoportal. Infraestructura de datos espaciales de Santiago de Cali. Recuperado de: <http://idesc.cali.gov.co/geovisor.php>

Irles, A. y Outerio A. (2013) Sistemas de gestores de bases de datos. Recuperado de: <http://gpsl.dlsi.ua.es/bbdd/bd1/lib/exe/fetch.php?media=bd1:0910:trabajos:aimsgbd.pdf>

Laboratorio Unidad Pacifico Sur (2019). ¿Qué es un SIG?. Recuperado de: <https://langleruben.wordpress.com/%c2%bfque-es-un-sig/>

Lopez, E., Posada, C. y Moreno, J. (1997). Los sistemas de información geográfica, I Congreso de Ciencia Regional de Andalucía: Andalucía en el umbral del siglo XXI. Recuperado de: <http://imsturex.unex.es/MUIETSIG/TEMA1.pdf>

López, R.(2018). Atlas sistema de ciudades-DNP. Recuperado de: <http://crcvalle.org.co/wp-content/uploads/Presentaci%C3%B3n-Departamento-Nacional-de-Planeaci%C3%B3n.pdf>

Marini, E. (2012). Modelo Cliente/Servidor. Recuperado de: <https://www.linuxito.com/docs/el-modelo-cliente-servidor.pdf>

MDN Web Docs (2019). Expresiones Regulares. Recuperado de: [https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular\\_Expressions](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions)

MDN Web Docs (2019). Trabajando con JSON. Recuperado de: <https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>

Mercado, E.(2013) Implementación de un sistema de información geográfica con software libre para el apoyo a la toma de decisiones en la Pymes. Recuperado de: <http://148.204.210.201/tesis/1386623427681TESISEDGARCAR.pdf>

Morales, A. (2012). Turf: Análisis espacial en la web. MappingGis. Recuperado de: <https://mappinggis.com/2015/11/turf-analisis-espacial-web/>

Morales, A.(2012). Primeros pasos con la API JavaScript de Google Maps. MappingGis. Recuperado de: <https://mappinggis.com/?s=javascript>

Niño, L. (2019). Evaluación de la calidad de la información geográfica voluntaria mediante un enfoque de análisis multivariado - caso de estudio malla vial Bogotá-Colombia. Recuperado de: <http://bdigital.unal.edu.co/72164/4/LuisNino.2019.pdf>

- Ordás, J. (2012). Aplicación de SIG en proyecto de gestión de redes hidráulicas en la República de Sudáfrica. Recuperado de: [http://scielo.sld.cu/scielo.php?pid=S168003382012000200007&script=sci\\_arttext&lng=en](http://scielo.sld.cu/scielo.php?pid=S168003382012000200007&script=sci_arttext&lng=en)
- Polo, I. (2013). Proyección del crecimiento urbano del área metropolitana de Barranquilla a 20 años, mediante el uso de los SIG. Recuperado de: <http://manglar.uninorte.edu.co/bitstream/handle/10584/8149/114850.pdf?sequence=1&isAllowed=y>
- PostGis (2014). Manual PostGis. Recuperado de: <https://postgis.net/docs/manual-dev/postgis-es.html>
- Rivas, D. (2016). Propuesta de guía metodológica para el manejo de problemas en estandarización y calidad de datos de direcciones urbanas en Colombia. Recuperado de: <https://repository.upb.edu.co/bitstream/handle/20.500.11912/2839/Trabajo%20de%20Grado%20Daniel%20Rivas.pdf?sequence=1&isAllowed=y>
- Rosa, M. (2014). Se eliminarán transversales y diagonales en Cali para evitar enredos. Periódico El País, p. 2. Cali - Valle. Recuperado de: <http://www.elpais.com.co/elpais/cali/noticias/calenos-le-dicen-adios-direcciones-enredadas>
- Timarán, R., Hernández, G. y Quema, N. (2016). Geocodificador de eventos delictivos georreferenciados a nivel de direcciones urbanas en el municipio de Pasto. V congreso internacional de gestión tecnológica y de la innovación. Recuperado de: [http://cogestec.ingenio.com.co/db/separated/2016%20\(44\).pdf](http://cogestec.ingenio.com.co/db/separated/2016%20(44).pdf)
- Vargas, J. y Horfan, D. (2013). Proceso de Geocodificación de direcciones en la ciudad de Medellín, una técnica determinística de Georreferenciación de direcciones. Recuperado de: <http://revistas.usbbog.edu.co/index.php/IngUSBmed/article/view/278/193>
- Wikipedia (1991). Navegador Web. Recuperado de: [https://es.wikipedia.org/wiki/Navegador\\_web](https://es.wikipedia.org/wiki/Navegador_web)
- Wikipedia (2019) Modelo-Vista-Controlador. Recuperado de: <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>
- Wikipedia (2019) PHP. Recuperado de: <https://es.wikipedia.org/wiki/PHP>
- Wikipedia (2019). Api de Google. Recuperado de: [https://es.wikipedia.org/wiki/API\\_de\\_Google](https://es.wikipedia.org/wiki/API_de_Google)
- Wikipedia (2019). HTML. Recuperado de: <https://es.wikipedia.org/wiki/HTML>

Wikipedia (2019). OpenStreetMap. Recuperado de:  
<https://es.wikipedia.org/wiki/OpenStreetMap>

Wikipedia(2019). CSS. Recuperado de:  
[https://es.wikipedia.org/wiki/Hoja de estilos en cascada](https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada)

## 11. ANEXOS

### Anexo 1. Tabla de Tiempos - Proceso en uso Vs Geocoord

| Tipo de Predio           | Cant. de predios | Tiempo de Gestión en MIN |          |
|--------------------------|------------------|--------------------------|----------|
|                          |                  | Proceso Antiguo          | Geocoord |
| URBANIZACION             | 20               | 30                       | 13       |
| CONJUNTO CERRADO-BLOQUES | 224              | 60                       | 15       |
| CONJUNTO CERRADO-BLOQUES | 56               | 30                       | 8        |
| CONJUNTO CERRADO-BLOQUES | 83               | 25                       | 8        |
| CONJUNTO CERRADO-CASAS   | 170              | 260                      | 60       |
| URBANIZACION             | 80               | 120                      | 35       |
| URBANIZACION             | 80               | 130                      | 35       |
| CONJUNTO CERRADO-CASAS   | 150              | 230                      | 45       |
| CONJUNTO CERRADO-BLOQUES | 196              | 45                       | 15       |
| CONJUNTO CERRADO-BLOQUES | 209              | 60                       | 20       |

**Anexo 2. Documento de la empresa con estandarización de direcciones**  
[Anexo Estandarización\Estandarizacion de direcciones emp tel.pdf](#)

### Anexo 3. Algoritmo estandarización de direcciones mediante expresiones regulares.

```
import VerEx from 'verbal-expressions'

export default {
  arrayDifference : function (arr1, arr2){
    //menor [2,4,6]
    //mayor [1,2,3,4]
    arr1.forEach( (iarr1, index1) => {
      arr2.forEach((iarr2, index2) => {
        if(iarr1 == iarr2){
          arr1[index1] = null
        }
      })
    })
    return arr1.filter(Boolean)
  },
  streetValidator(street){
    let template = /(?!<typeStreet>\D\S*)?(.)?ls?(?!<street>.*\w*)?ls?(?!<bis>bis|bijb)?ls?(?!<cuadrante>\D\S*)?/i
    let streetMatch = street.match(template)
    let json = {
      typeStreet: "",
      number: "",
      letter: "",
      cuadrante: "",
      bis: "",
    }
  },
  let rexStreet = [
    {
      exp: /^kra$|^k$|^kr$|^carrera$|^c$|^ca$|^car$|^carr$|^carre$|^carrer$|^cra$|^craa$|^crr$|^cr$/i,
      string: 'KR'
    },
    {
      exp: /^calle$|^ca$|^call$|^cl$|^cl$/i,
      string: 'CL'
    },
    {
      exp: /^avenida$|^av$|^ave$|^aven$|^aveni$|^avenid$|^avn$|^avnida$/i,
      string: 'AV'
    },
    {
      exp: /^transversal$|^t$|^tr$|^tra$|^tran$|^trans$|^transv$|^transve$|^transver$|^transvers$|^transversa$|^tr$|^tv$/i,
      string: 'TV'
    },
    {
      exp: /^diagonal$|^d$|^di$|^dia$|^diag$|^diago$|^diagon$|^diagona$|^dg$|^dgnal$|^dgn$|^dnal$/i,
      string: 'DG'
    },
    {
      exp: /^autopista$|^a$|^au$|^aut$|^auto$|^autop$|^autopi$|^autopis$|^autopist$|^at$/i,
      string: 'AU'
    },
  ]
}
```

```

let rexCuadrante = [
  {
    rexp: /^norte$|^n$|^no$|^nor$|^nort$|^nt$|^nte$/i,
    string : 'N'
  },

  {
    rexp: /^sur$|^s$|^su$|^sr$/i,
    string : 'S'
  },

  {
    rexp: /^este$|^e$|^es$|^est$|^et$/i,
    string : 'E'
  },

  {
    rexp: /^oeste$|^o$|^oe$|^oes$|^oest$|^o$|^ot$|^ost$|^os$/i,
    string : 'O'
  }
]

if(streetMatch != null ){
  for(let item in streetMatch.groups){
    switch (item){
      case 'typeStreet':
        if(streetMatch.groups[item] != undefined){
          rexStreet.forEach(exp => {
            streetMatch.groups[item] = streetMatch.groups[item].replace(exp.rexp, exp.string)
            json.typeStreet = streetMatch.groups[item]
          })
        }
        break

      case 'street':
        if(streetMatch.groups[item] && streetMatch.groups[item] != null){
          if (/^\d{1,3}$/.test(streetMatch.groups[item])){
            let match = streetMatch.groups[item].match(/^(?<number>\d{1,3})(?<letter>w*)$/i)
            json.number = match.groups.number
            match.groups.letter != undefined ? json.letter = match.groups.letter.toUpperCase() : json.letter = ""
          }else{
            json.number = streetMatch.groups[item]
          }
        }
        break

      case 'bis':
        if (streetMatch.groups[item] != undefined) json.bis = 'BIS'
        break

      case 'cuadrante':
        if(streetMatch.groups[item] != undefined){
          rexCuadrante.forEach(exp => {
            streetMatch.groups[item] = streetMatch.groups[item].replace(exp.rexp, exp.string)

```

```

        json.cuadrante = streetMatch.groups[item]
    })

    }
    break
default:
    break
}
}

let normalizedStreet = `${json.typeStreet} ${json.number}${json.letter} ${json.bis} ${json.cuadrante}`
return (normalizedStreet)
}
return ('la dirección ingresada no coincide con ningun formato')
},
addressToJson(address){
    let
                                template
                                =
/(?<typeStreet1>\D\S*)?(.)?s?(?<street1>'.*\|w+)?s?(?<bis1>bis|bilb)?s?(?<cuadrante1>norte|n|no|nor|nort|nt|nte|sur|s|su
|sr|este|e|es|est|et|oeste|o|oe|oes|oest|o|ot|ost|os)?,s?(?<typeStreet2>\D\S*)?(.)?s?(?<street2>'.*\|w+)?s?(?<bis2>bis|bil
b)?s?(?<cuadrante2>norte|n|no|nor|nort|nt|nte|sur|s|su|sr|este|e|es|est|et|oeste|o|oe|oes|oest|o|ot|ost|os)?s?(-
)?s?(?<placa>\d*)?/i
    let addressMatch = address.match(template)

    let streetJson = {
        typeStreet1: "",
        number1: "",
        letter1: "",
        bis1: "",
        cuadrante1: "",
        typeStreet2: "",
        number2: "",
        letter2: "",
        bis2: "",
        cuadrante2: "",
        placa: ""
    }

    let rexStreet = [
        {
            rexp: /^kra$|^k$|^kr$|^carrera$|^c$|^ca$|^car$|^carr$|^carre$|^carrer$|^cra$|^crra$|^crr$|^cr$/i,
            string: 'KR'
        },
        {
            rexp: /^calle$|^cal$|^call$|^cl$|^cl$/i,
            string: 'CL'
        },
        {
            rexp: /^avenida$|^av$|^ave$|^aven$|^aveni$|^avenid$|^avn$|^avnida$/i,
            string: 'AV'
        },
        {
            rexp: /^transversal$|^t$|^tr$|^tra$|^trans$|^trans$|^transv$|^transve$|^transver$|^transvers$|^transversa$|^tr$|^tv$/i,
            string: 'TV'
        },
    ]

```

```

    rexp : /^diagonal$|^d$|^di$|^dia$|^diag$|^diago$|^diagon$|^diagona$|^dg$|^dgnal$|^dgn$|^dnal$/i,
    string : 'DG'
  },
  {
    rexp : /^autopista$|^a$|^au$|^aut$|^auto$|^autop$|^autopi$|^autopis$|^autopist$|^at$/i,
    string : 'AU'
  },
]

let rexCuadrante = [
  {
    rexp : /^norte$|^n$|^no$|^nor$|^nort$|^nt$|^nte$/i,
    string : 'N'
  },
  {
    rexp : /^sur$|^s$|^su$|^sr$/i,
    string : 'S'
  },
  {
    rexp : /^este$|^e$|^es$|^est$|^et$/i,
    string : 'E'
  },
  {
    rexp : /^oeste$|^o$|^oe$|^oes$|^oest$|^o$|^ot$|^ost$|^os$/i,
    string : 'O'
  }
]

if (addressMatch != null) {
  try {
    for (let item in addressMatch.groups) {
      switch (item) {
        case 'typeStreet1':
          if (addressMatch.groups[item] != undefined) {
            rexStreet.forEach(exp => {
              addressMatch.groups[item] = addressMatch.groups[item].replace(exp.rexp, exp.string)
              streetJson.typeStreet1 = addressMatch.groups[item]
            })
          }
          break

        case 'street1':
          if (addressMatch.groups[item] != undefined && addressMatch.groups[item] != null) {
            if (!/^\d{1,3}$/.test(addressMatch.groups[item])) {
              let street1tMatch = addressMatch.groups[item].match(/^(?<number1>\d{1,3})(?<letter1>w*)$/i)
              streetJson.number1 = street1tMatch.groups.number1
              street1tMatch.groups.letter1 != undefined ? streetJson.letter1 = street1tMatch.groups.letter1.toUpperCase() :
streetJson.letter1 = "
            } else {
              streetJson.number1 = addressMatch.groups[item]
            }
          }
        }
      }
    }
  }
}

```

```

        break

    case 'bis1':
        if (addressMatch.groups[item] != undefined) streetJson.bis1 = 'BIS'
        break

    case 'cuadrante1':
        if (addressMatch.groups[item] != undefined){
            rexCuadrante.forEach(exp => {
                addressMatch.groups[item] = addressMatch.groups[item].replace(exp.rexp, exp.string)
                streetJson.cuadrante1 = addressMatch.groups[item]
            })
        }
        break

    case 'typeStreet2':
        if (addressMatch.groups[item] != undefined){
            rexStreet.forEach(exp => {
                addressMatch.groups[item] = addressMatch.groups[item].replace(exp.rexp, exp.string)
                streetJson.typeStreet2 = addressMatch.groups[item]
            })
        }
        break

    case 'street2':
        if (addressMatch.groups[item] != undefined && addressMatch.groups[item] != null){
            if (!/^\d{1,3}$/.test(addressMatch.groups[item])){
                let stree2tMatch = addressMatch.groups[item].match(/^(?<number2>\d{1,3})(?<letter2>w*)$/i)
                streetJson.number2 = stree2tMatch.groups.number2
                stree2tMatch.groups.letter2 != undefined ? streetJson.letter2 = stree2tMatch.groups.letter2.toUpperCase() :
streetJson.letter2 = "
            }else{
                streetJson.number2 = addressMatch.groups[item]
            }
        }
        break

    case 'bis2':
        if (addressMatch.groups[item] != undefined) streetJson.bis2 = 'BIS'
        break

    case 'cuadrante2':
        if (addressMatch.groups[item] != undefined){
            rexCuadrante.forEach(exp => {
                addressMatch.groups[item] = addressMatch.groups[item].replace(exp.rexp, exp.string)
                streetJson.cuadrante2 = addressMatch.groups[item]
            })
        }
        break

```

```

    case 'placa':
      if (addressMatch.groups[item] != undefined) streetJson.placa = addressMatch.groups[item]
      break

    default:
      break
  }
}
return (streetJson)
} catch (error){
  console.log(error)
  throw (error)
}
}
return({})
},
addressValidator(address){
  let streetJson = this.addressToJson(address)
  if (streetJson.typeStreet1){
    let normalizedDir = `${streetJson.typeStreet1} ${streetJson.number1}${streetJson.letter1} ${streetJson.bis1}
    ${streetJson.cuadrante1} , ${streetJson.typeStreet2} ${streetJson.number2}${streetJson.letter2} ${streetJson.bis2}
    ${streetJson.cuadrante2} - ${streetJson.placa}`
    return (normalizedDir.replace(/s+/g, ' ').replace(/s/,g, ','))
  } else{
    return ('la dirección ingresada no coincide con ningun formato')
  }
}
}
}

```