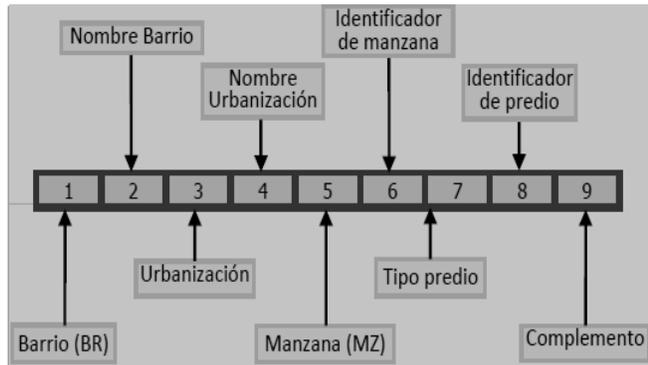
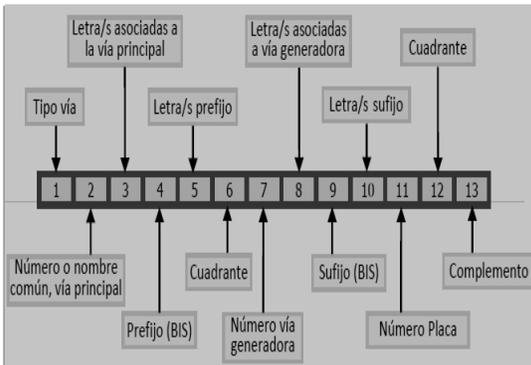
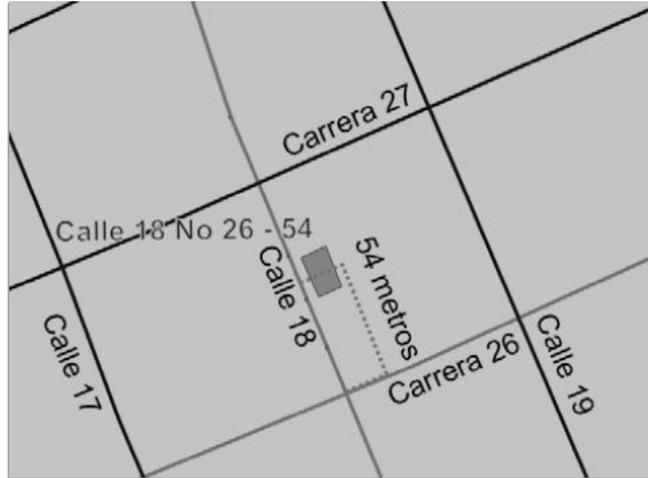
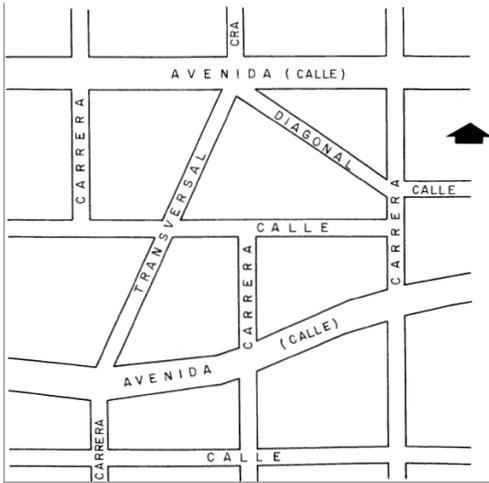


# GEOCODER PASTO:

## Geocodificador de Direcciones Urbanas del Municipio de Pasto



## MANUAL DE REFERENCIA

**Grupo de Investigación GRIAS- KDD**

**Departamento de Sistemas**

**Facultad de Ingeniería**

**Universidad de Nariño**

**San Juan de Pasto**

**2016**



You are free:

-  to **Share** – to copy, distribute and transmit this work
-  to **Remix** – to adapt this work

Under the following conditions:

-  **Attribution** – You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work)
-  **Noncommercial** – You may not use this work for commercial purposes.

Subject to conditions outlined in the license.

This work is licensed under the *Creative Commons Attribution-NonCommercial 3.0 Unported* License. To view a copy of this license, visit

<http://creativecommons.org/licenses/by-nc/3.0/>

or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

# Tabla de Contenido

<b>INTRODUCCIÓN</b>	<b>8</b>
<b>1. CONSTRUCCIÓN DEL GEOCODER PASTO</b>	<b>10</b>
<b>1.1 Construcción del repositorio de información urbana</b>	<b>10</b>
Obtención de datos espaciales	10
Corrección de datos obtenidos y recopilación de direcciones urbanas	12
Definición del modelo de datos	14
Migración de datos OSM a PostgreSQL	15
<b>1.2 Construcción del geocodificador de direcciones urbanas</b>	<b>16</b>
Definición procesos de geocodificación	16
Implementación procesos de geocodificación	24
<b>2. IMPLEMENTACIÓN GEOCODER PASTO</b>	<b>27</b>
<b>2.1 Instalación editor JOSM desde repositorio de paquetes</b>	<b>27</b>
<b>2.2 Atributos de direcciones urbanas en JOSM</b>	<b>31</b>
<b>2.3 Carga de capas base</b>	<b>33</b>
<b>2.4 Digitalización de direcciones urbanas</b>	<b>33</b>
<b>3. INSTALACION GEOCODER PASTO</b>	<b>40</b>
<b>3.1 Software requerido:</b>	<b>40</b>
Gestor de bases de datos PostgreSQL con extensión PostGIS	40
Extensión para PostgreSQL pg_similarity	41
Librerías de Python para la migración de datos OSM	41
Instalación del esquema de datos y funciones de geocodificación de direcciones:	41
<b>3.2 Migración de datos OSM a PostgreSQL</b>	<b>42</b>
Migración de barrios y comunas.	42
Migración de direcciones urbanas.	43
Asignación de identificadores de barrios y comunas a direcciones urbanas	44
<b>3.3 Prueba de funcionamiento del geocodificador</b>	<b>44</b>
<b>4. APORTES DEL GEOCODER PASTO</b>	<b>46</b>
<b>4.1 Carácter Original o innovativo del GEOCODER PASTO</b>	<b>46</b>
<b>4.2 Aporte al fortalecimiento de la capacidad nacional de investigación, innovación y desarrollo tecnológico</b>	<b>46</b>
<b>4.3 Aporte a la consolidación de un área estratégica</b>	<b>47</b>
<b>5. IMPACTOS DEL GEOCODER PASTO</b>	<b>47</b>
<b>5.1 Impacto científico y tecnológico</b>	<b>47</b>
<b>5.2 Impacto en el medio ambiente y sociedad</b>	<b>48</b>
<b>5.3 Aspectos económicos y financieros</b>	<b>48</b>
<b>REFERENCIAS BIBLIOGRÁFICAS</b>	<b>49</b>

## Lista de Figuras

Figura 1. Lado derecho: Mapas de referencia descargados del servidor Open Street Maps. Lado izquierdo: Malla vial del municipio de Pasto filtrada de los datos descargados. ....	11
Figura 2. Capas que corresponden a los barrios (lado izquierdo) y las comunas (lado derecho) del municipio de Pasto .....	12
Figura 3 Segmentación de las comunas para la recopilación de direcciones. Ejemplo: Comuna 10.	13
Figura 4. Address Interpolation: Complemento de la herramienta Java Open Street Maps utilizado para la recopilación de direcciones urbanas. ....	14
Figura 5 Modelo Entidad - Relación de la base de datos con información del área urbana del municipio de Pasto. ....	15
Figura 6. Procesos de geocodificación para direcciones urbanas del municipio de Pasto .....	17
Figura 7. Algunas direcciones urbanas registradas por el observatorio del delito. ....	20
Figura 8. Autómata Finito Determinista que reconoce los componentes de las direcciones asignadas según la malla vial del municipio de Pasto .....	22
Figura 9. Autómata Finito Determinista que reconoce los componentes de direcciones asignadas según la nomenclatura Barrio-Manzana-Identificación de predio .....	23
Figura 10. Ejecución del procedimiento de estandarización de direcciones urbanas asignadas según la malla vial .....	24
Figura 11. Ejecución del procedimiento almacenado de estandarización de direcciones urbanas asignadas según la nomenclatura barrio - manzana - predio.....	25
Figura 12. Ejecución del procedimiento almacenado de búsqueda de direcciones urbanas asignadas según la malla vial.....	26
Figura 13. Ejecución del procedimiento almacenado de búsqueda de direcciones urbanas asignadas según la nomenclatura barrio - manzana - predio .....	26
Figura 14. JOSM en ejecución. ....	28
Figura 15. Instalación complemento AddressInterpolation en JOSM.....	29
Figura 16. Archivo OSM con direcciones urbanas de la Comuna 10 cargados en JOSM .....	30
Figura 17. Edición de los atributos de una geometría (Punto). ....	31
Figura 18. Etiquetas de un punto que corresponde a una dirección asignada según la malla vial ....	32

Figura 19. Etiquetas de un punto que corresponde a una dirección asignada según la nomenclatura Barrio Manzana Predio.....	32
Figura 20. Bing Maps cargado como capa base en JOSM.....	33
Figura 21. Selección de vía principal (vía de mayor longitud) y línea dibujada, ambas de color rojo. ....	34
Figura 22. Complemento Address Interpolation en ejecución .....	35
Figura 23. Configuración de valores para generar puntos iniciando desde la distancia 1 y terminando en la distancia 21 (Impar).....	36
Figura 24. Puntos generados con el complemento AddressInterpolation .....	37
Figura 25. Configuración de valores para generar puntos iniciando desde el predio con identificación 1 hasta el predio con identificación 4.....	38
Figura 26. Puntos generados y sus etiquetas. ....	39

## Lista de Tablas

Tabla 1. Casos de reemplazo de símbolos y nomenclaturas en direcciones urbanas ingresadas. ....	21
Tabla 2. Resultado del proceso ejecutado por el Autómata Finito Determinista para direcciones urbanas asignadas según la malla vial. ....	22
Tabla 3. Resultado del proceso ejecutado por el Autómata Finito Determinista para direcciones urbanas asignadas según la nomenclatura barrio – manzana – predio. ....	23

## Lista de Algoritmos

Algoritmo 1 - RoadNetworkGeocoding. Geocodificación de direcciones asignadas según la malla vial.....	18
Algoritmo 2 - NBlockHouseidGeocoding. Geocodificación de direcciones asignadas según la nomenclatura Barrio Manzana Predio.....	19

## INTRODUCCIÓN

La geocodificación es el proceso de traducción de direcciones, ciudad, estado, código postal, y otra información de ubicación en un mapa de coordenadas (latitud y longitud)[1].

El geocoder Pasto es un geocodificador de código abierto para el municipio de Pasto que permite traducir direcciones urbanas a coordenadas geográficas longitud/latitud. Con este geocodificador, el municipio de Pasto ingresa al conjunto de ciudades del mundo que cuentan con una herramienta de este tipo para la georreferenciación de todo tipo de objetos geográficos. Entre los geocodificadores desarrollados se tiene:

En el ámbito internacional se destaca TIGER Geocoder, un geocodificador de códigos postales y direcciones en los Estados Unidos creado por [2]. Se basa en el Sistema Geográfico de Codificación y Referenciación Topológicamente Integrado (TIGER, por sus siglas en inglés) publicado por la Oficina del Censo de los Estados Unidos [3] para la estandarización de descripciones de objetos geográficos como calles, edificios, ríos o regiones en ese país. Fue desarrollado bajo software libre en el lenguaje procedural PL/SQL del sistema gestor de bases de datos PostgreSQL utilizando funciones de la extensión PostGIS para gestión de datos espaciales.

De igual forma se destaca Cartociudad, adaptando TIGER Geocoder, un sistema de geocodificación y visualización cartográfica digital que recopila información sobre la red vial, la división administrativa censal y la división postal de España. [4] identifica los procesos de Ingeniería inversa realizados para conocer, adaptar e implementar la arquitectura y componentes de TIGER Geocoder en el Sistema Gestor de Bases de Datos con soporte de datos espaciales Oracle Spatial.

En el ámbito nacional se identifica el aplicativo web Mapas.com.co desarrollado por [5], cuyo fin es brindar a usuarios u organizaciones servicios de publicación de información comercial o de negocios en la web sobre mapas digitales mediante contratos ASP (arrendamiento de servicios)[6]. Se destaca los servicios de geocodificación (a nivel de direcciones residenciales y sitios de interés) y búsqueda de rutas óptimas de recorrido con cobertura en gran parte del territorio colombiano, ambos realizados con alto grado de precisión reflejada en sus resultados.

Otro estudio realizado por [7], en el cual comparan los resultados arrojados por los procesos determinístico y probabilístico de geocodificación implementados en la Alcaldía de Medellín para la georreferenciación de información primaria en todos sus procesos de

governabilidad y desarrollo territorial. Las herramientas analizadas fueron: la funcionalidad Address Locator (denominada GEOCODING por la entidad gubernamental) provista por la suite comercial ArcGIS para gestión de información geográfica (geocodificación probabilístico) y el geocodificador desarrollado para la ciudad de Medellín denominado GEOCOD, una aplicación transversal desarrollada en el lenguaje procedural PL/SQL de Sistema Gestor de Bases de datos Oracle y el lenguaje de programación Python (geocodificación determinístico) [7].

En el ámbito regional y del departamento se identifican dos investigaciones importantes. El trabajo realizado por [8] tiene el objetivo de apoyar a la toma de decisiones de la subsecretaría de Planeación Municipal sobre el Plan de Ordenamiento Territorial del municipio de Pasto a través del desarrollo de un sistema de información geográfica web. El sistema recopila información geográfica general (comunas, ríos, perímetros) y con detalle sobre la comuna 3 (delimitación de la comuna, sectores, predios) del municipio de Pasto.

Otra investigación es la realizada por [9]. Esta herramienta fue desarrollada con el objetivo de facilitar a las pequeñas y medianas organizaciones de la región y el país administrar su información de forma georreferenciada. Permite la gestión y visualización de mapas cartográficos creados a partir de múltiples fuentes de información. Se destaca el módulo de geocodificación el cual propone un algoritmo de geocodificación de direcciones denominada Atlas Street Geocoding.

Con el Geocoder Pasto, el municipio dispone de un geocodificador de código abierto que permite traducir nomenclaturas urbanas a coordenadas geográficas longitud/latitud para luego poder visualizar e identificar zonas geográficas específicas de acuerdo al mapa temático que se utilice. Por ser un geocodificador bajo licencia de software libre, permitirá a las diferentes organizaciones gubernamentales o privadas del municipio de Pasto y del país reutilizar el geocodificador construido y acoplarlo a cualquier visualizador de mapas con el fin de registrar e identificar según su contexto funcional, zonas de interés, sitios turísticos, rutas de buses, paraderos, zonas de posibles desastres, entre otras en la cartografía del municipio de Pasto.

Así como los anteriores geocodificadores, el geocoder Pasto fue acoplado en un visor cartográfico denominado PASTO VIEW. Este permite la búsqueda y ubicación de direcciones urbanas en el municipio de Pasto. Por otra parte, fue integrado a SIGEODEP, el Sistema Georreferenciado del Observatorio del Delito del Municipio de Pasto mediante la construcción del aplicativo SIGEODEP-SIG. Este nuevo módulo de SIGEODEP

georreferencia las lesiones fatales y no fatales a nivel de direcciones urbanas permitiendo visualizar zonas en el municipio donde se identifiquen situaciones atípicas o especiales sin depender de barrios o comunas como está actualmente configurado.

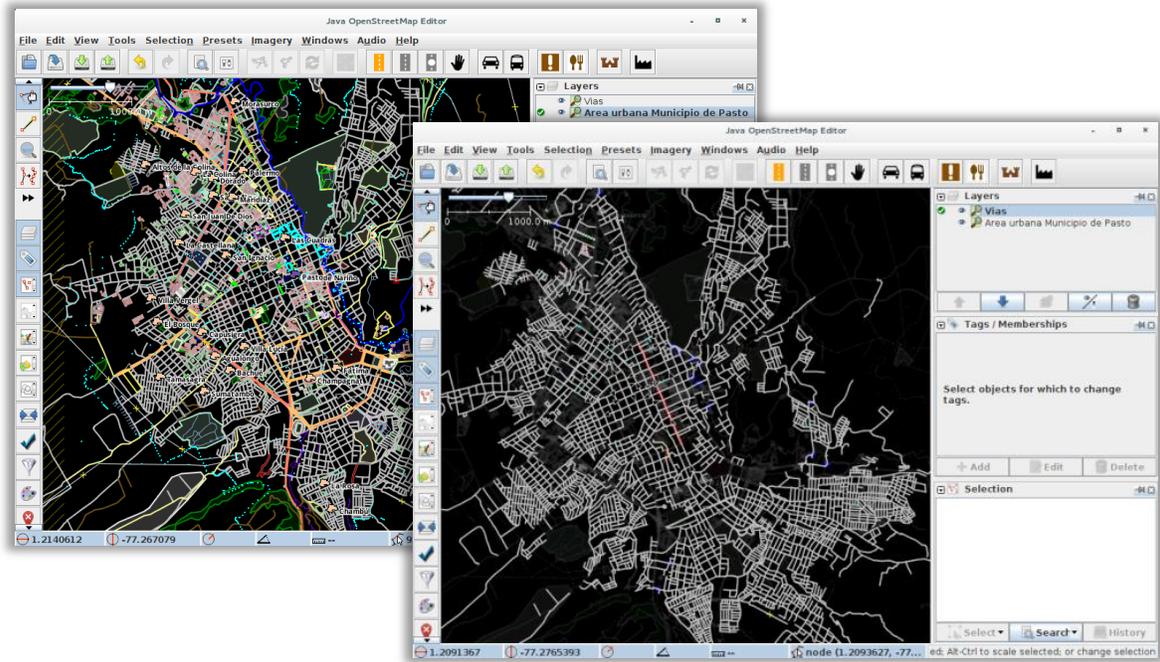
## **1. CONSTRUCCIÓN DEL GEOCODER PASTO**

### **1.1 Construcción del repositorio de información urbana**

Esta fue una fase fundamental en el desarrollo de la investigación, donde se recopiló la información urbana del municipio de Pasto, incluidas las direcciones urbanas que fueron utilizadas posteriormente por el geocodificador de direcciones urbanas. La creación del repositorio de direcciones urbanas digitalizadas en archivos OSM a una base de datos creada en PostgreSQL se explica en las siguientes secciones

#### **Obtención de datos espaciales**

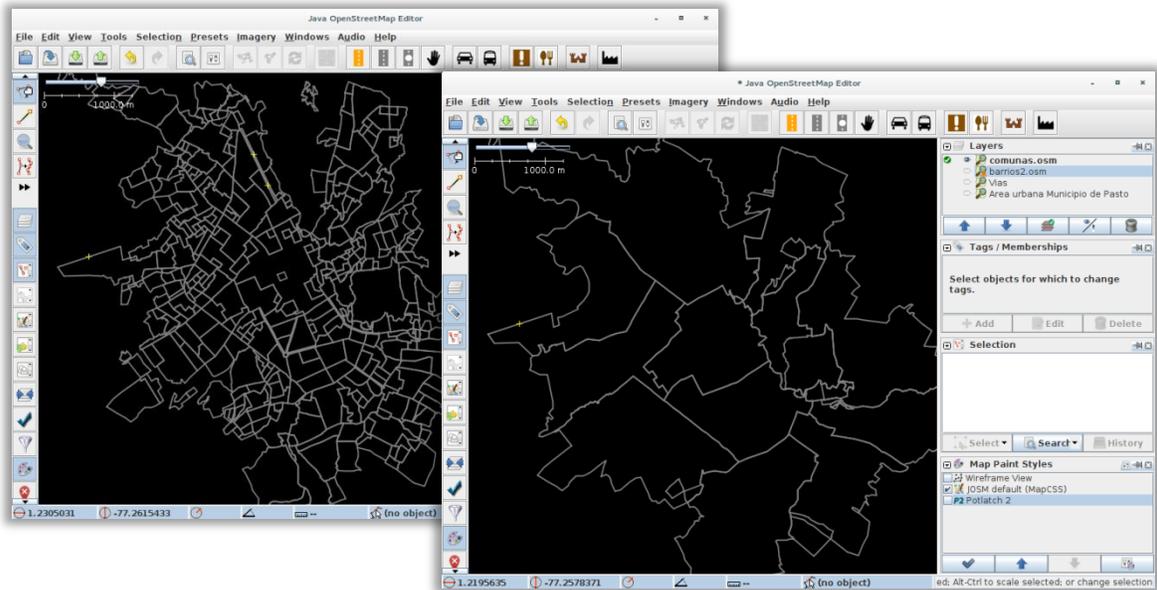
Inicialmente se obtuvo la capa de datos espaciales vectoriales que corresponden a la malla vial del área urbana del Municipio de Pasto, disponibles en el servicio de edición y compartición libre de mapas cartográficos en la web *Open Street Maps* . Para la descarga de estos datos fue utilizada la herramienta de edición de datos cartográficos Java Open Street Maps Editor. Se obtuvo un archivo con extensión OSM que almacena información georreferenciada del área urbana del municipio de Pasto, bajo una estructura de datos XML (*Extended Markup Language*). A partir de estos datos fueron filtrados los datos que corresponden a la malla vial del municipio (véase Figura 1).



*Figura 1. Lado derecho: Mapas de referencia descargados del servidor Open Street Maps. Lado izquierdo: Malla vial del municipio de Pasto filtrada de los datos descargados.*

Las capas de datos espaciales que corresponden a los barrios y comunas se obtuvieron del sistema georreferenciado SIGODEP. Estas capas servirán posteriormente para asignar a cada dirección su correspondiente barrio y comuna. (Véase Figura 2).

Los datos espaciales contenidos en cada archivo se almacenan en 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).



*Figura 2. Capas que corresponden a los barrios (lado izquierdo) y las comunas (lado derecho) del municipio de Pasto*

### **Corrección de datos obtenidos y recopilación de direcciones urbanas**

Dado que gran parte de la información geográfica obtenida se encontraba incompleta, fue realizada una previa corrección de estos datos. Al mismo tiempo, se dio inicio a la recopilación de las direcciones urbanas del municipio de Pasto. Estas tareas se ejecutaron en conjunto a medida que se recorría una a una las cuadras del municipio de Pasto. Para una recopilación ordenada de los datos se definió segmentar la malla vial por cada comuna del municipio de Pasto (véase Figura 3).

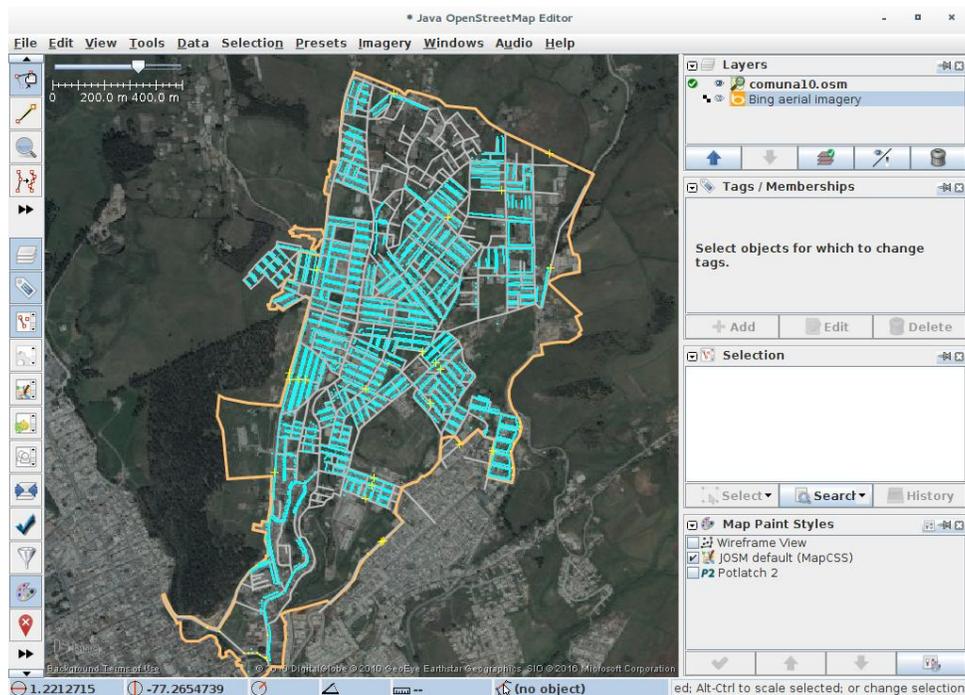


Figura 3 Segmentación de las comunas para la recopilación de direcciones. Ejemplo: Comuna 10.

Como fuente de información de referencia para corregir estos datos se utilizó el visor de mapas Google Street View, el visor de mapas cartográficos Bing Maps, el geoportal de visualización de datos geográficos y catastrales del Instituto Geográfico Agustín Codazzi - IGAC y herramientas libres de edición de mapas cartográficos como Java Open Street Maps Editor y Quantum GIS. Las direcciones digitalizadas corresponden a las cuadras (o zonas) que las fuentes de información seleccionadas permitían observar.

Para la recopilación las direcciones urbanas fue utilizado el complemento Address Interpolation del editor Java Open Street Maps Editor (véase Figura 4), el cual permitió generar puntos por cada cuadra recorrida de forma rápida, configurando los atributos necesarios de acuerdo con las normas de asignación de direcciones urbanas manejados en el municipio de Pasto. Se tuvo como referencia la propuesta de estandarización de direcciones del Instituto Geográfico Agustín Codazzi – IGAC, lo que permitió obtener direcciones con las nomenclaturas estándar que serán procesadas por el geocodificador.

Por cada punto creado, se recopilan de forma implícita atributos de georreferenciación: su correspondiente longitud y latitud. La obtención de los puntos se almacenó dentro de los archivos OSM obtenidos inicialmente junto con los datos correspondientes a vías barrios y comunas corregidos.

Figura 4. Address Interpolation: Complemento de la herramienta Java Open Street Maps utilizado para la recopilación de direcciones urbanas.

## Definición del modelo de datos

Luego de obtener la información urbana del municipio de Pasto corregida, se definió el modelo de datos para el almacenamiento de esta información. Se definió el modelo Entidad – Relación de la base de datos, considerando los atributos disponibles por cada tipo de datos recopilados:

- Comunas (communes)
- Barrios (neighborhoods)
- Direcciones urbanas asignadas según:
  - Malla vial (road\_network\_addresses)
  - Nomenclatura Barrio - Manzana - Identificación del predio (neighborhood\_block\_houseid\_addresses).

Las entidades y relaciones del modelo de datos definido se explican a continuación (véase figura 5):

- Una comuna puede contener varios barrios.
- Un barrio puede contener varios predios, por ende, direcciones urbanas asignadas según la malla vial o según la nomenclatura Barrio Manzana – Identificación del predio.

- Determinar la relación entre las vías con las demás entidades tiene cierto grado de dificultad, puesto que se debe considerar la ubicación espacial de cada una de estas para encontrar los segmentos de vía que se están ubicados principalmente en las comunas y barrios del municipio de Pasto. Por este motivo, esta entidad fue definida sin relación alguna.

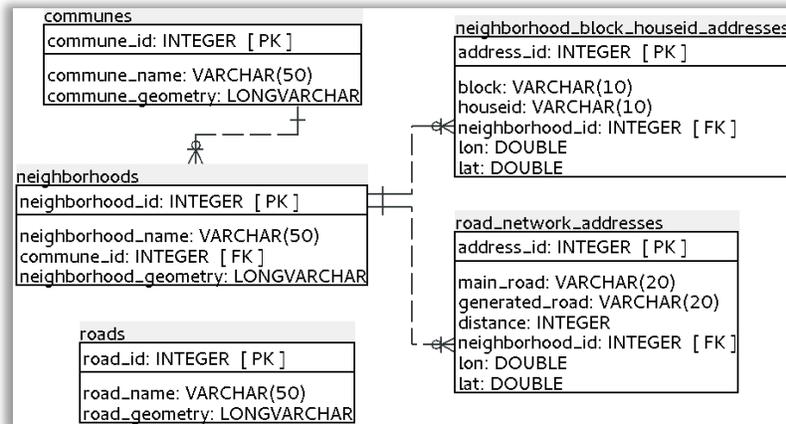


Figura 5 Modelo Entidad - Relación de la base de datos con información del área urbana del municipio de Pasto.

## Migración de datos OSM a PostgreSQL

Dado que no fue posible encontrar una herramienta que permita migrar los datos recopilados, fue necesaria la codificación de procedimientos parametrizados de migración de datos de los archivos OSM obtenidos a una base de datos relacional con el modelo de datos implementado en el gestor de bases de datos PostgreSQL con extensión de datos espaciales PostGIS. Los procedimientos fueron escritos en el lenguaje de programación Python, los cuales se describen a continuación:

- Un procedimiento de migración de geometrías basadas en polígonos y líneas creado con el fin de migrar los datos de los barrios y las comunas corregidas.
- Un procedimiento de migración basado en puntos creado con el fin de migrar los datos de las direcciones urbanas recopiladas. Dado que cada archivo puede contener direcciones urbanas asignadas según la malla vial o según la nomenclatura barrio – manzana- predio, éste reconoce estos dos tipos de direcciones y los inserta en su respectiva entidad (tabla).

Previamente, cada procedimiento transforma el sistema de referencia espacial nativo de los datos al sistema de referencia espacial WGS84 Web Mercator (EPSG: 3857), esto con el fin de facilitar el manejo de las direcciones urbanas con las herramientas utilizadas para construir el visor cartográfico, las cuales trabajan de forma nativa con este sistema.

## **1.2 Construcción del geocodificador de direcciones urbanas**

En esta fase se definieron los procesos de geocodificación de direcciones para el área urbana del municipio de Pasto. Posteriormente, estos fueron implementados en procedimientos almacenados de estandarización y reconocimiento de componentes de las direcciones urbanas para el municipio de Pasto.

### **Definición procesos de geocodificación**

Considerando el proceso de geocodificación definido y los casos de asignación de direcciones del municipio de Pasto seleccionados para la construcción del geocodificador previamente estudiados, se definieron los procesos de geocodificación de direcciones para direcciones asignadas según la malla vial y según la nomenclatura barrio - manzana – predio. Se tuvo como referencia las normas establecidas en la propuesta de estandarización de direcciones del Instituto Geográfico Agustín Codazzi, las cuales fueron adaptadas al contexto del municipio de Pasto.

Los procesos de geocodificación para el área urbana del municipio de Pasto pueden verse en la figura 6.

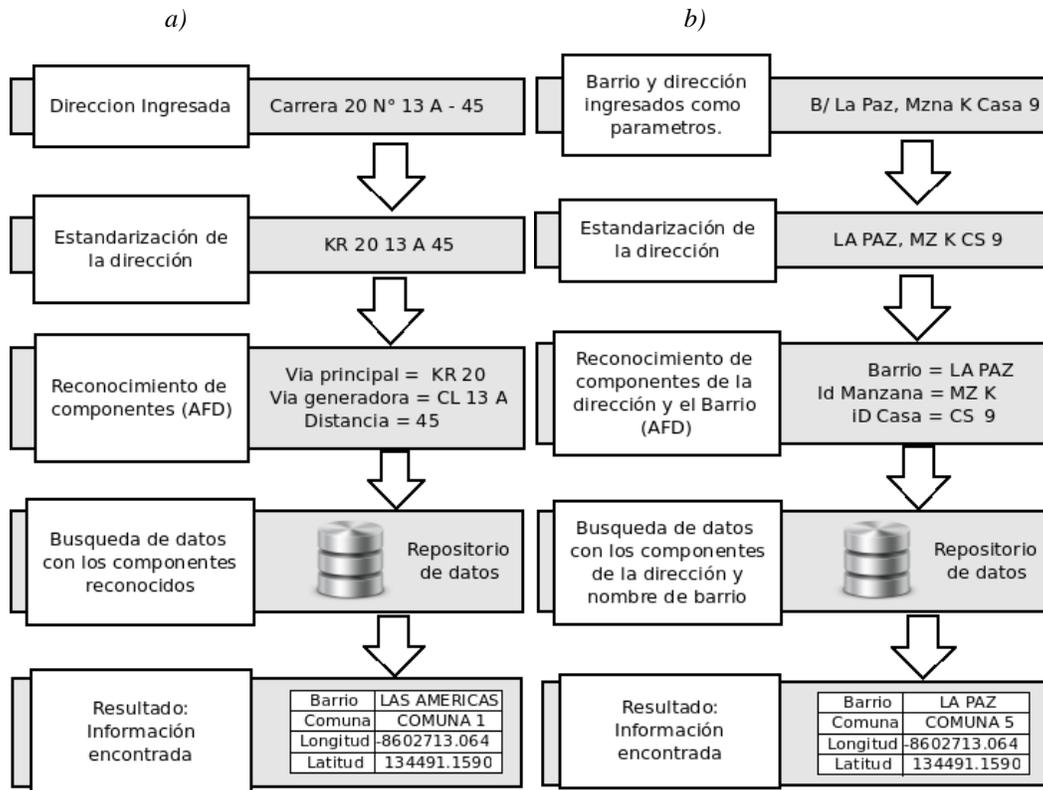


Figura 6. Procesos de geocodificación para direcciones urbanas del municipio de Pasto

a) Según malla vial

b) Según manzanas

Los Algoritmos 1 y 2 muestran el pseudocódigo de los algoritmos de geocodificación definidos por cada caso de direcciones urbanas.

---

**Algorithm** RoadNetworkGeocoding

---

```
1: procedure GEOCODE(searchedAddress)
2:   find and replace searchedAddress components with standardized nomenclatures using regex
3:   addrElements[]  $\leftarrow$  searchedAddress converted to array using space character as delimiter
4:   state  $\leftarrow$  0
5:   for each element in addrElements do
6:     if state = 0 then
7:       if element is similar to [CL] or [KR] or [DG] then
8:         identify the matching nomenclature
9:         main.road  $\leftarrow$  value of standardized nomenclature identified.
10:        state  $\leftarrow$  1
11:       end if
12:     else if state = 1 then
13:       if element is similar to [0 - 9] or [0 - 9][A - Z] then
14:         main.road  $\leftarrow$  main.road concatenated with the current element identified.
15:         state  $\leftarrow$  2
16:       end if
17:     else if state = 2 then
18:       if element is similar to [A - Z] or BIS or NORTE|SUR|ESTE|OESTE then
19:         main.road  $\leftarrow$  main.road concatenated with the current element identified.
20:         state  $\leftarrow$  2
21:       else if element is similar to [0 - 9] or [0 - 9][A - Z] then
22:         generator.road  $\leftarrow$  value of current element identified.
23:         state  $\leftarrow$  3
24:       end if
25:     else if state = 3 then
26:       if element is similar to [A - Z] or BIS or NORTE|SUR|ESTE|OESTE then
27:         generator.road  $\leftarrow$  generator.road concatenated with the current element identified.
28:         state  $\leftarrow$  3
29:       else if element is similar to [0 - 9] or [0 - 9][A - Z] then
30:         distance  $\leftarrow$  value of current element identified discarding letters.
31:         state  $\leftarrow$  4
32:       end if
33:     end if
34:   end for
35:   if state > 3 then
36:     search required address in database using the recognized address components.
37:     if recognized address components are found then
38:       return standardized address, neighborhood, commune, longitude, latitude found
39:     end if
40:   end if
41: end procedure
```

---

Algoritmo 2 - NBlockHouseidGeocoding. Geocodificación de direcciones asignadas según la nomenclatura Barrio Manzana Predio.

---

**Algorithm** NBlockHouseidGeocoding

---

```

1: procedure GEOCODE(searchedNeighborhood, searchedAddress)
2:   neighborhood ← searchedNeighborhoodwithstandardizednomenclatures
3:   address ← searchedAddresswithstandardizednomenclatures
4:   addrElements[] ← address converted to array using space character as delimiter
5:   state ← 0
6:   for each element in addrElements do
7:     if state = 0 then
8:       if element is similar to [MZ] then
9:         identify the matching nomenclature
10:        block ← value of standardized nomenclature identified.
11:        state ← 1
12:      end if
13:    else if state = 1 then
14:      if element is similar to [0 – 9] or [A – Z] or [0 – 9][A – Z] then
15:        identify the matching element
16:        block ← block concatenated with the current element identified.
17:        state ← 3
18:      end if
19:    else if state = 2 then
20:      if element is similar to [A – Z] or [CS] then
21:        identify the matching element
22:        houseid ← value of current element identified.
23:        state ← 3
24:      end if
25:    else if state = 3 then
26:      if element is similar to [CS] then
27:        block ← block concatenated with the current element identified.
28:        state ← 3
29:      else if element is similar to [0 – 9] or [0 – 9][A – Z] then
30:        identify the matching element
31:        houseid ← 'CS' string concatenated with the value of current element identified.
32:        state ← 4
33:      end if
34:    else if state = 4 then
35:      if element is similar to [A – Z] then
36:        houseid ← houseid concatenated with the current element identified.
37:        state ← 4
38:      end if
39:    end if
40:  end for
41:  if state > 3 then
42:    search required address in database using the recognized address components.
43:    if recognized address components and neighborhood are found then
44:      return standardized address, neighborhood, commune, longitude, latitude found
45:    end if
46:  end if
47: end procedure

```

---

A continuación se describen estos algoritmos:

**a. Ingreso de direcciones:** En este primer paso del algoritmo se definen las características de direcciones urbanas que serán ingresadas y procesadas por el geocodificador. Para esto,

se hizo la revisión de las direcciones almacenadas en los registros de delitos del observatorio del delito del municipio de Pasto. Esta revisión sirvió de base para asumir que las direcciones urbanas ingresadas al geocodificador tendrá diversos formatos que no cumple ningún tipo de norma. Algunas de estas direcciones urbanas revisadas pueden observarse en la figura 7.

a)		b)	
direccion character varying	barrio character varying(75)	direccion character varying	barrio character varying(75)
CALLE 16B No 33-61	MARIDIAZ	MZ 21 CASA 13	NUEVO SOL
CRR 23 No 10-37	SANTIAGO	MZ B1 CASA 4	NUEVA ARANDA
CALLE 19A/3E-20	SANTAFÉ I	MZ 19 CASA 1	VILLA FLOR II
CRA 22F 9 26 OBRERO	SAN JOSE OBRERO	MZ E CASA 7	PANORAMICO I
CRA 6B N 18B-31	SENDOYA	MZ 22 CASA 10	SAN ALBANO
CLL 11 B No 4 14 B /	CHAPAL	MZ 8 CASA 27	LA MINGA
CRA 2 N 19-60	BETANIA	MZ A CASA 22	GUALCALOMA
Calle 5 Nro. 22 D-42	SAN JOSE OBRERO	MZ 49 CASA 18	EL CHAMBU I
CRA 29 No 19-81	LAS CUADRAS	MZ 51 CASA 22	TAMASAGRA I
CRA 26 N 21B-24	SANTA BARBARA	MZ B CASA B - 13	NUEVO HORIZONTE
KRA 4 E N 21 B 43	SANTA BARBARA	MZ K CASA 167	CAICEDONIA
CRA 5 E N 21 - 25	SANTA BARBARA	MZ 11 CASA 11	LA MINGA
CARRERA 7 N 20 -28	CHILE	MZ 3 CASA 17	NIZA I
CR 7E Nº 19 A 24	BETANIA	MZ 43 CASA 1	VILLA FLOR II
CRA 7 N 19A-40	SANTA FE	MZ D CASA 5	NIÑO JESUS DE PRAGA
Cra 2 19-18	SANTA FE	MZ 19 CASA 26	VILLA FLOR I

Figura 7. Algunas direcciones urbanas registradas por el observatorio del delito.

a) Según la malla vial

b) Según nomenclatura barrio - manzana - predio

A continuación se describen las principales características de las direcciones urbanas revisadas:

- 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. Carrera 20 13 A – 45.
- Para el caso de direcciones urbanas asignadas según la nomenclatura barrio – manzana – predio, deben ingresarse dos datos sobre la dirección urbana solicitada: **el nombre del barrio y la identificación del predio de acuerdo con este caso**. Este último debe tener mínimo dos componentes: **Identificación de la Manzana e Identificación del predio**. Ej. Barrio La Paz, Manzana K - Casa 9.
- Las direcciones ingresadas pueden contener datos adicionales, estos deben ser posteriormente descartados por el procedimiento de normalización de direcciones.

**b. Estandarización de la dirección urbana (líneas 2-3 Algoritmo 1, líneas 2-4 Algoritmo 2):** 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 1 se puede observar algunos ejemplos de reemplazo.

*Tabla 1. Casos de reemplazo de símbolos y nomenclaturas en direcciones urbanas ingresadas.*

<b>Símbolos/Nomenclaturas reconocidos</b>	<b>Símbolo(s) reemplazo</b>
Texto en minúsculas	Texto en Mayúsculas
CARRERA CRA KRA KR CR CARERA	KR
CALLE CALL CLLE CLL CL CAL CLE CALE	CL
DIAGONAL DIAG DGNAL	DG
MANZANA MZNA MZA MANZ MZN	MZ
CASA CSA CAS  C	CS
N NO NUMERO NUM	Espacio
Signos de puntuación (.,;:-)	Espacio
Caracteres especiales (\$%&/)	Espacio

**c. Reconocimiento de los componentes de la dirección urbana (líneas 5-34 Algoritmo 1, líneas 6-40 Algoritmo 2):** Con el fin de obtener los componentes mínimos de la dirección urbana ingresada y eliminar cualquier tipo de texto adicional, se consideró el geocodificador creado en el grupo GRIAS del departamento de Sistemas de la Universidad de Nariño por C. Arteaga, M. Erazo y R. Timarán dentro del proyecto denominado “Atlas: Herramienta de cartografía web y geocodificación para el desarrollo de sistemas híbridos en áreas urbanas sobre J2EE y PostgreSQL” en el año 2008, para implementar autómatas finitos deterministas (AFD), encargados de reconocer los componentes mínimos de cada dirección, descartando o eliminando texto adicional que haya sido registrado junto con la dirección urbana. Para el diseño de estos autómatas se utilizaron expresiones regulares que permiten el reconocimiento de texto a partir de patrones de búsqueda. El AFD definido para las direcciones según la malla vial se muestra en la figura 8.

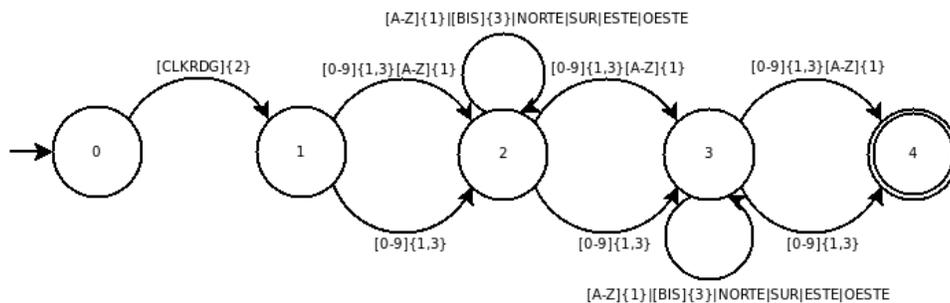


Figura 8. *Autómata Finito Determinista que reconoce los componentes de las direcciones asignadas según la malla vial del municipio de Pasto*

El resultado que arroja el Autómata Finito Determinista se muestra en la tabla 2.

Tabla 2. *Resultado del proceso ejecutado por el Autómata Finito Determinista para direcciones urbanas asignadas según la malla vial.*

Dirección Ingresada	Dirección estandarizada	Componentes reconocidos (Eliminación de texto Adicional)		
		Vía principal	Vía generadora	Distancia
Carrera 20 #13 a 45	KR 20 13 A 45	KR 20	13 A	45
Calle 9 No 34 – 30 Las Acacias	CL 9 34 30 LAS ACACIAS	CL 9	34	30
Cll 18 26 - 54 Centro	CL 18 26 54 CENTRO	CL 18	26	54
Calle 3 15 62 Caicedo Alto	CL 3 15 62 CAICEDO ALTO	CL 3	15	62

El Autómata Finito Determinista definido para direcciones asignadas según la nomenclatura Barrio-Manzana-Identificación de predio aprovecha el ingreso de los datos Barrio y Dirección urbana por separado, procesando únicamente este último elemento. El dato que corresponde al barrio se procesa mediante una búsqueda aproximada de texto utilizando el algoritmo de similitud de texto Jarowinkler. El procesamiento por separado de

estos dos componentes se debe a la identificación de direcciones que no tenían un barrio asociado, dato importante en la geocodificación de este tipo de direcciones. Para este caso, el AFD que procesa el elemento dirección urbana se muestra en la figura 9.

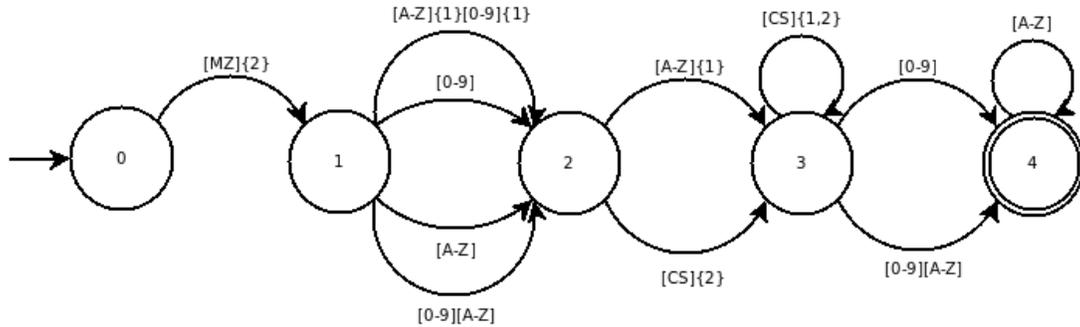


Figura 9. Automata Finito Determinista que reconoce los componentes de direcciones asignadas según la nomenclatura Barrio-Manzana-Identificación de predio

El resultado que arroja el Automata Finito Determinista se muestra en la tabla 3.

Tabla 3. Resultado del proceso ejecutado por el Automata Finito Determinista para direcciones urbanas asignadas según la nomenclatura barrio – manzana – predio.

Dirección Ingresada	Dirección estandarizada	Componentes reconocidos (Eliminación de texto Adicional)	
		Id. Manzana	Id. Predio
Manzana 10 Casa 9	<b>MZ 10 CS 9</b>	MZ 10	CS 9
Mzna 2 Casa 1A	<b>MZ 2 CS 1A</b>	MZ 9	CS 1A
Barrio La Paz Mz K Cs 9	<b>BARRIO LA PAZ MZ K CS 9</b>	MZ K	CS 9
Mzna 10 F Csa 2 B/ Nueva Aranda	<b>MZ 10 CS 2 B NUEVA ARANDA</b>	MZ 10F	CS 2

**d. Búsqueda de la información urbana (líneas 35-39 Algoritmo 1, líneas 41-45 Algoritmo 2):** Una vez obtenidos los componentes de la dirección urbana buscada, finalmente se realiza la búsqueda de la dirección en el repositorio de información urbana. Si la dirección es encontrada se retornará los siguiente datos:

- Dirección urbana estandarizada

- Barrio al que pertenece.
- Comuna a la que pertenece.
- Coordenadas latitud y longitud de ubicación espacial.

## Implementación procesos de geocodificación

Una vez analizado los requerimiento del geocodificador, se optó por implementar los procesos de geocodificación de direcciones en procedimientos almacenados dentro del sistema gestor de bases de datos PostgreSQL con extensión para el soporte de datos espaciales PostGIS. En conjunto, estas herramientas permitieron la manipulación de los datos espaciales mediante recursos de las bases de datos como el lenguaje de consulta estructurado SQL y procedimientos almacenados para el manejo de expresiones regulares y consultas espaciales.

Como resultado de la implementación se tuvo el geocodificador de direcciones urbanas, compuestos por 4 funciones procedurales codificadas en el lenguaje procedural PL/PgSQL de PostgreSQL:

El procedimiento almacenado de estandarización de direcciones asignadas según la malla vial se encarga de estandarizar y reconocer los componentes de las direcciones urbanas del municipio de Pasto. La ejecución del procedimiento almacenado creado para las direcciones se puede ver en la figura 10.

```
WITH normalize AS(
    SELECT normalize_address('Carrera 20 # 13 A - 45 B/ Las Americas') AS result
)
SELECT
    (result).main_road AS via_principal,
    (result).generated_road AS via_generadora,
    (result).distance AS distancia
FROM
    normalize;
```

Output pane

Data Output Explain Messages History

	via_principal character varying	via_generadora character varying	distancia integer
1	KR 20	13 A	45

Figura 10. Ejecución del procedimiento de estandarización de direcciones urbanas asignadas según la malla vial

El procedimiento de estandarización de direcciones asignadas según la nomenclatura barrio– manzana – predio se encarga de estandarizar y reconocer los componentes de las

direcciones urbanas del municipio de Pasto. La ejecución del procedimiento se puede ver en la figura 11.

```
WITH normalize AS(
    SELECT normalize_address('Barrio La Paz', 'Manzana K Casa 9') AS result
)
SELECT
    (result).neighborhood AS barrio,
    (result).block AS id_manzana,
    (result).houseid AS id_predio
FROM
    normalize;
```

Output pane

Data Output Explain Messages History

	barrio character varying(40)	id_manzana character varying(20)	id_predio character varying(20)
1	LA PAZ	MZ K	CS 9

*Figura 11. Ejecución del procedimiento almacenado de estandarización de direcciones urbanas asignadas según la nomenclatura barrio - manzana - predio*

El procedimiento almacenado de búsqueda de direcciones asignadas según la nomenclatura barrio- manzana – predio se encarga de realizar la búsqueda de la dirección asignada según la malla vial estandarizada mediante sus componentes reconocidos. La ejecución del procedimiento almacenado se puede ver en la figura 12.

```

WITH geocode AS (
    SELECT geocode_address('Carrera 20 # 13 A - 45 B\ Las Americas') AS result
)
SELECT
    (result).neighborhood AS barrio,
    (result).address AS direccion,
    (result).commune AS comuna,
    (result).lon AS longitud,
    (result).lat AS latitud
FROM
    geocode;

```

Output pane

Data Output Explain Messages History

	barrio character varying(50)	direccion character varying(25)	comuna character varyi	longitud double precision	latitud double precision
1	LAS AMERICAS	KR 20 13 A 45	COMUNA 1	-8602713.06642356	134491.159088312

Figura 12. Ejecución del procedimiento almacenado de búsqueda de direcciones urbanas asignadas según la malla vial

El procedimiento almacenado de búsqueda de direcciones asignadas según la nomenclatura barrio- manzana – predio se encarga de realizar la búsqueda de la dirección asignada según la nomenclatura barrio - manzana - predio estandarizada, mediante sus componentes reconocidos. La ejecución del procedimiento almacenado se puede ver en la figura 13.

```

WITH geocode AS (
    SELECT geocode_address('Br La Paz', 'Manzana K Casa 9') AS result
)
SELECT
    (result).neighborhood AS barrio,
    (result).address AS direccion,
    (result).commune AS comuna,
    (result).lon AS longitud,
    (result).lat AS latitud
FROM
    geocode;

```

Output pane

Data Output Explain Messages History

	barrio character varying(50)	direccion character varying(25)	comuna character varyi	longitud double precision	latitud double precision
1	LA PAZ	MZ K CS 9	COMUNA 4	-8600523.43231094	133201.206467672

Figura 13. Ejecución del procedimiento almacenado de búsqueda de direcciones urbanas asignadas según la nomenclatura barrio - manzana - predio

## 2. IMPLEMENTACIÓN GEOCODER PASTO

A continuación se describe el procedimiento necesario para modificar los archivos OSM que contienen las direcciones urbanas del municipio de Pasto. Los siguientes pasos fueron ejecutados dentro del sistema operativo Linux Ubuntu en su versión 14.04.

### 2.1 Instalación editor JOSM desde repositorio de paquetes

El procedimiento de la instalación del editor JOSM puede ser consultado con detalle en el siguiente enlace: <https://josm.openstreetmap.de/wiki/Download>

Inicialmente se debe agregar una línea a la lista de recursos de paquetes ubicada en la ruta `/etc/apt/sources.list` con las siguientes instrucciones:

```
$ sudo nano /etc/apt/source.list  
  
#agregar esta linea al final del archivo  
deb https://josm.openstreetmap.de/apt trusty universe
```

Se guardan los cambios realizados, posteriormente se actualiza la lista de paquetes y se instala el editor JOSM con las siguientes instrucciones:

```
$ sudo apt-get update  
$ sudo apt-get install josm-latest
```

Una vez terminado el proceso de instalación podremos ingresar al aplicativo instalado como se muestra en la figura 14.

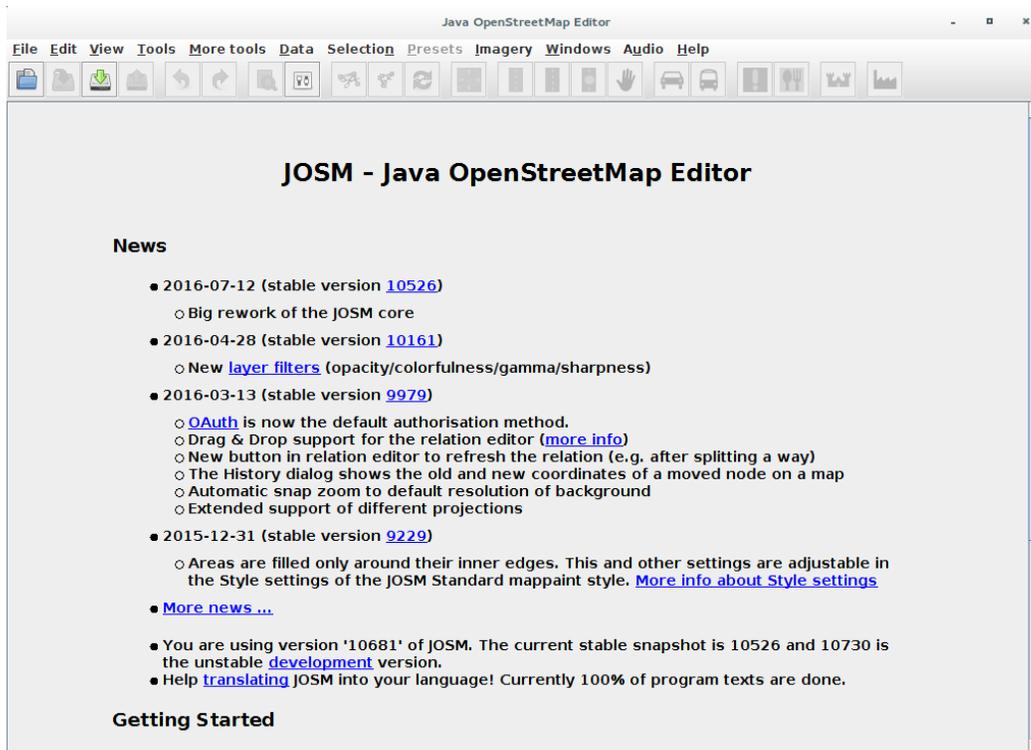
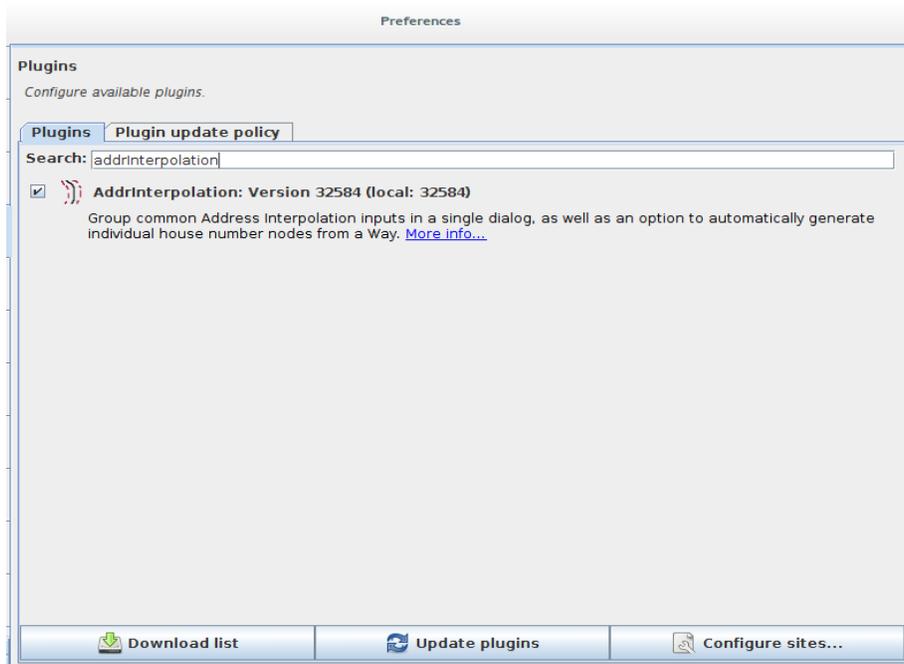


Figura 14. JOSM en ejecución.

Posteriormente instalamos el complemento AddressInterpolation, necesario para la digitalización de direcciones urbanas. En JOSM nos dirigimos al menú Editar/Preferencias/Complementos y buscamos el complemento que deseamos instalar, a continuación seleccionamos la opción Descargar Lista, como se muestra en la figura 15.



*Figura 15. Instalación complemento AddressInterpolation en JOSM*

Una vez instaladas las herramientas necesarias para digitalizar direcciones abrimos el archivo OSM que deseamos editar o corregir. Como ejemplo en este tutorial se utilizará el archivo OSM con las direcciones urbanas de la comuna 10. En el menú Archivo/Abrir buscamos y seleccionamos el archivo deseado, el cual se cargará en el aplicativo como se muestra en la figura 16.

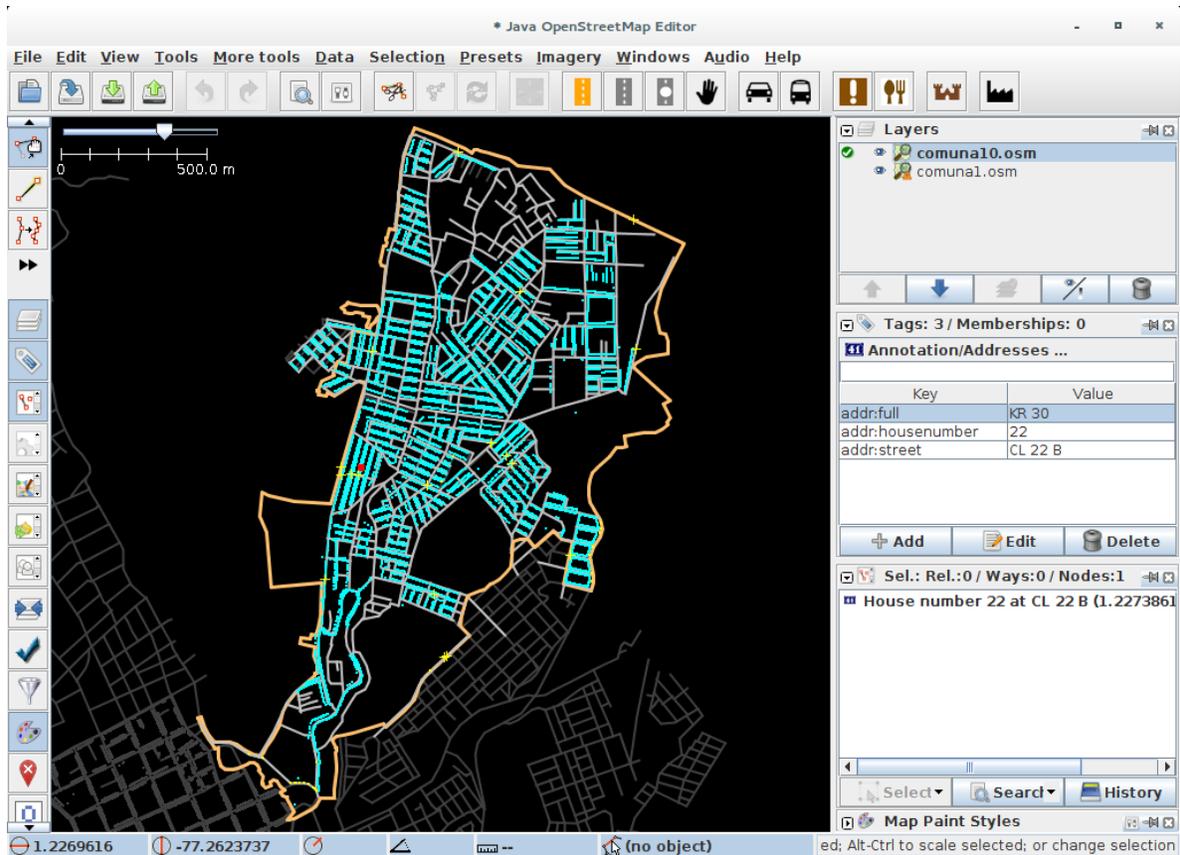


Figura 16. Archivo OSM con direcciones urbanas de la Comuna 10 cargados en JOSM

El archivo OSM seleccionado se cargará como una capa de datos espaciales, en este caso de geometrías basadas en puntos. Al seleccionar cualquiera de estos puntos se puede acceder a funcionalidades como moverlos en el espacio y visualizar sus características. Las características de cada punto cargado pueden verse en el panel derecho de la aplicación habilitando el subpanel Etiquetas (Tags). Estos valores pueden editarse dando doble clic sobre la Etiqueta que desea modificarse, como se muestra en la figura 17.

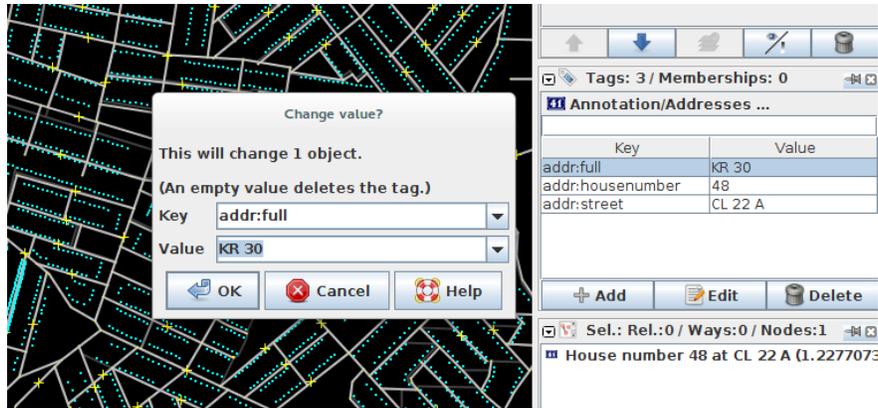


Figura 17. Edición de los atributos de una geometría (Punto).

## 2.2 Atributos de direcciones urbanas en JOSM

Antes de digitalizar o corregir los archivos OSM con direcciones urbanas del municipio de Pasto se debe considerar los atributos o componentes que deben ingresarse. Dado que el complemento AddressInterpolation esta creado para la digitalización de direcciones de Estados Unidos, para el contexto del municipio de Pasto se hizo una adaptación como se describe a continuación:

Para el caso de direcciones asignadas según la malla vial cada punto debe tener las siguientes etiquetas (véase figura 18):

- *addr:street* es la etiqueta que almacena la vía principal de la dirección urbana, esta debe iniciar con la nomenclatura estandarizada correspondiente: CL (Calle), KR (Carrera) ò DG (Diagonal). Las letras y/o sufijos que acompañen al identificador de calle deben escribirse con espacio y de acuerdo al orden establecido por la propuesta de estandarización de direcciones urbanas. Ejemplos: KR 3, KR 3 A, KR 3 A BIS, KR 3 A SUR.
- *addr:full* es la etiqueta que almacena la vía generadora o secundaria de la dirección urbana, esta debe iniciar con la nomenclatura estandarizada correspondiente: CL (Calle), KR (Carrera) ò DG (Diagonal). Las letras y/o sufijos que acompañen al identificador de calle deben escribirse con espacio y de acuerdo al orden establecido por la propuesta de estandarización de direcciones urbanas. Ejemplos: CL 20, CL 20 A, CL 20 A BIS, CL 20 A SUR.
- *addr:housenumber* es la etiqueta que almacena la distancia de la dirección urbana. Este campo solo debe contener valores numéricos.

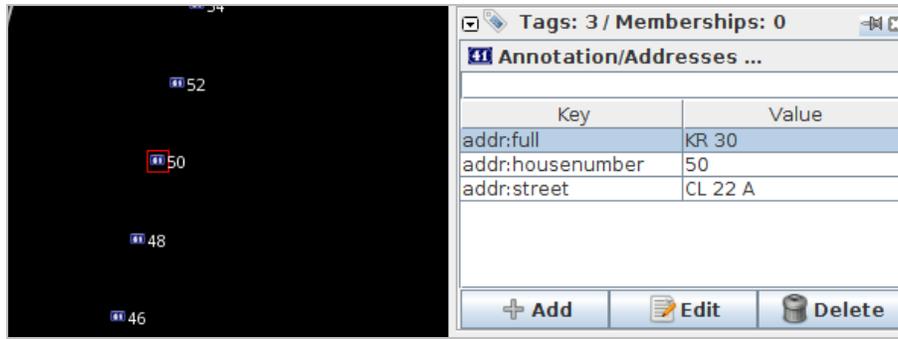


Figura 18. Etiquetas de un punto que corresponde a una dirección asignada según la malla vial

Para el caso de direcciones asignadas según la nomenclatura barrio – manzana – predio cada punto debe tener las siguientes etiquetas (véase figura 19):

- *addr:full* es la etiqueta que almacena el identificador de la manzana (block), este debe siempre iniciar con la nomenclatura MZ En caso de que este identificador contenga números y letras, este último elemento debe escribirse en mayúscula y sin espacio. Ejemplos: MZ 10, MZ 10A, MZ F, MZ F10,
- *addr:housenumber* es la etiqueta que almacena el identificador de predio (houseid), este debe solo contener este valor sin ninguna nomenclatura adicional. En caso de que este identificador contenga números y letras, este último elemento debe escribirse en mayúscula y sin espacio. Ejemplos: 11, 11A.

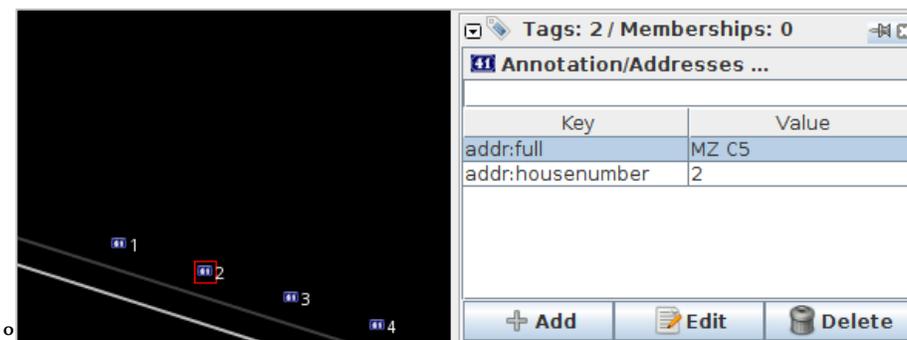


Figura 19. Etiquetas de un punto que corresponde a una dirección asignada según la nomenclatura Barrio  
Manzana Predio

## 2.3 Carga de capas base

Tanto para la corrección (modificación) como para la digitalización de nuevas direcciones es necesario tener como referencia una capa base que permita visualizar el mapa del área urbana del municipio de Pasto. En el menú Imágenes (véase figura 20) seleccionamos Bing Maps, mapa que fue utilizado para digitalizar la mayor parte de direcciones urbanas.

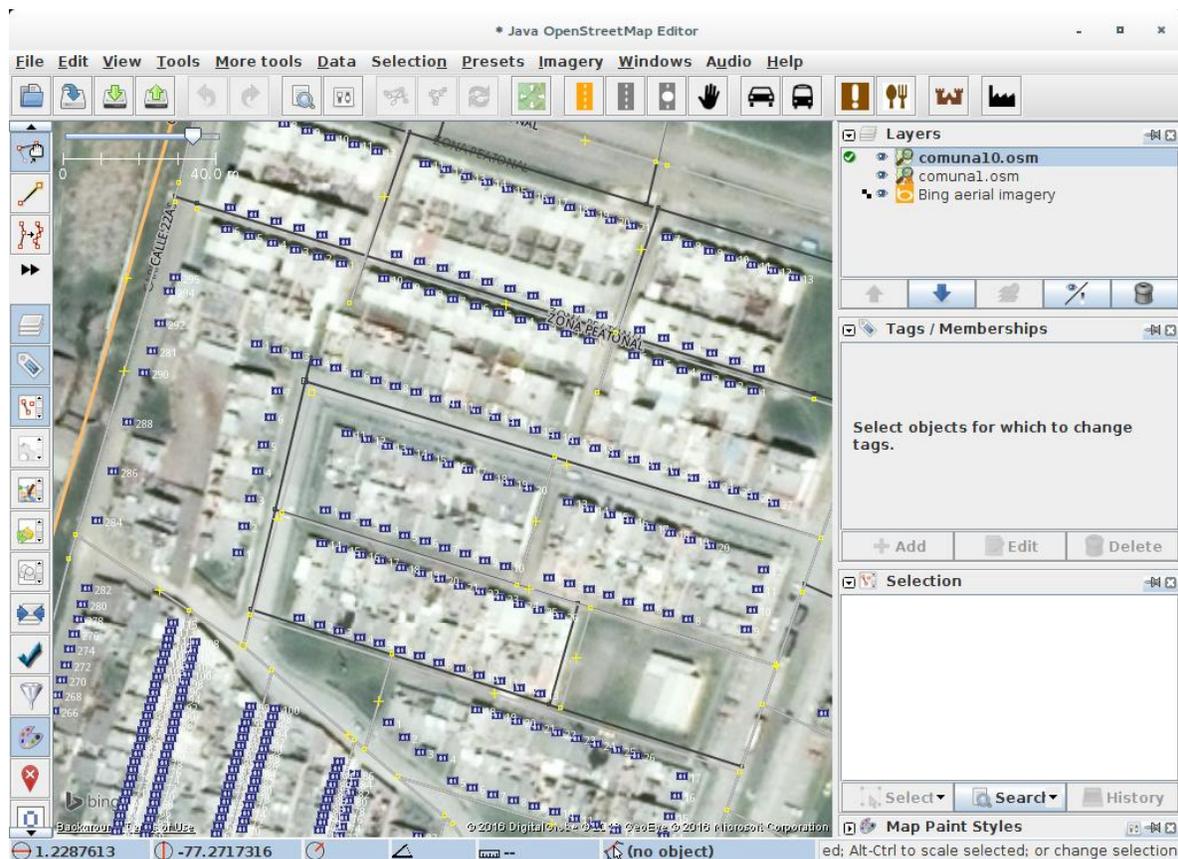


Figura 20. Bing Maps cargado como capa base en JOSM

## 2.4 Digitalización de direcciones urbanas

Para la digitalización de direcciones urbanas utilizaremos el complemento AddressInterpolation previamente instalado. Este permite acceder a una funcionalidad muy útil para la recolección de direcciones urbanas, como lo es la interpolación de puntos a partir de una geometría o línea base. Como resultado se tiene un conjunto de puntos generados a partir de una línea dibujada, cada punto tendrá los atributos de las direcciones urbanas según se especifique en la previa configuración de la interpolación.

Inicialmente identificamos una cuadra donde deseamos crear un conjunto de puntos que representarán direcciones urbanas ubicadas en la zona seleccionada. Con ayuda de la malla vial dibujamos una línea paralela a la vía principal que atraviesa la cuadra seleccionada. Manteniendo seleccionada la tecla Ctrl, seleccionamos tanto la vía principal como la línea paralela que será interpolada (véase figura 21).

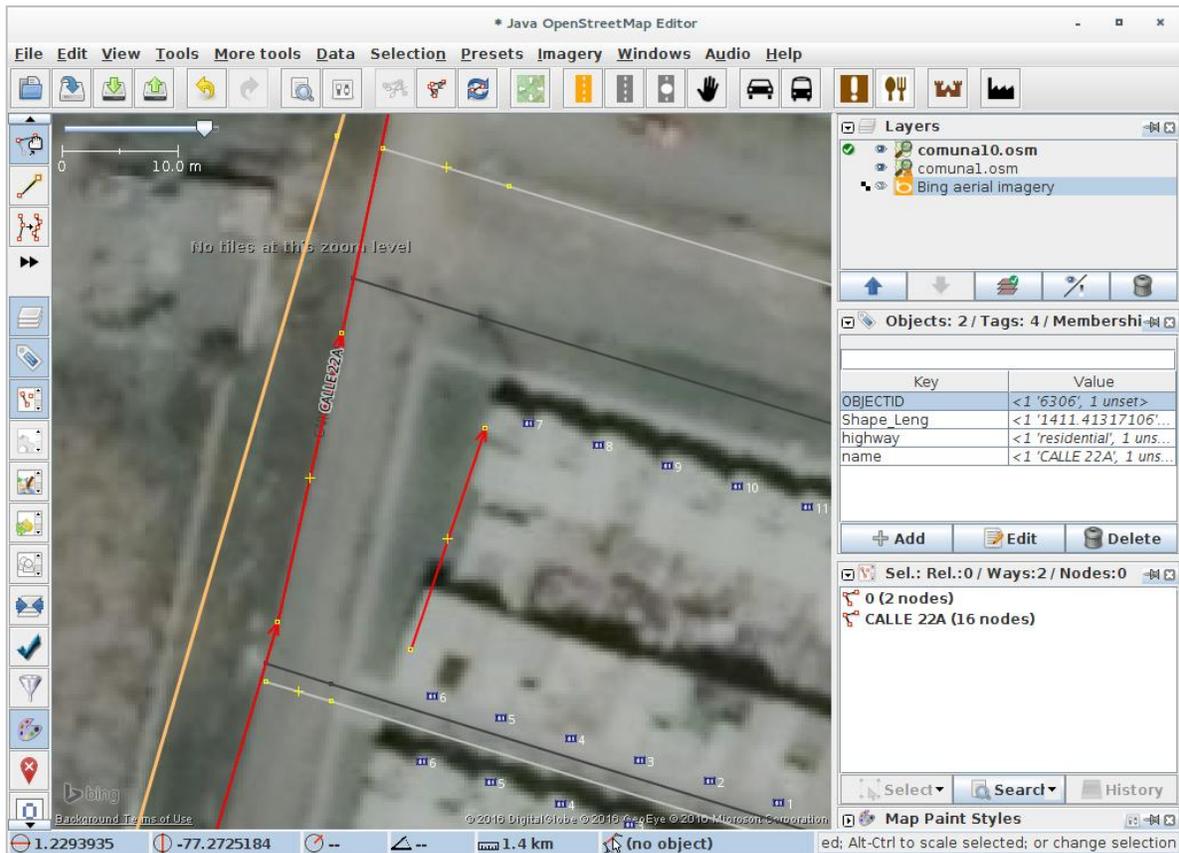


Figura 21. Selección de vía principal (vía de mayor longitud) y línea dibujada, ambas de color rojo.

Una vez seleccionadas estas geometrías, en el menú Datos seleccionamos la opción AddressInterpolation, la cual nos dirige al cuadro de configuración de complemento (véase figura 22). Para mayor comodidad se recomienda acceder desde la combinación de teclas Ctrl + Alt + Z.

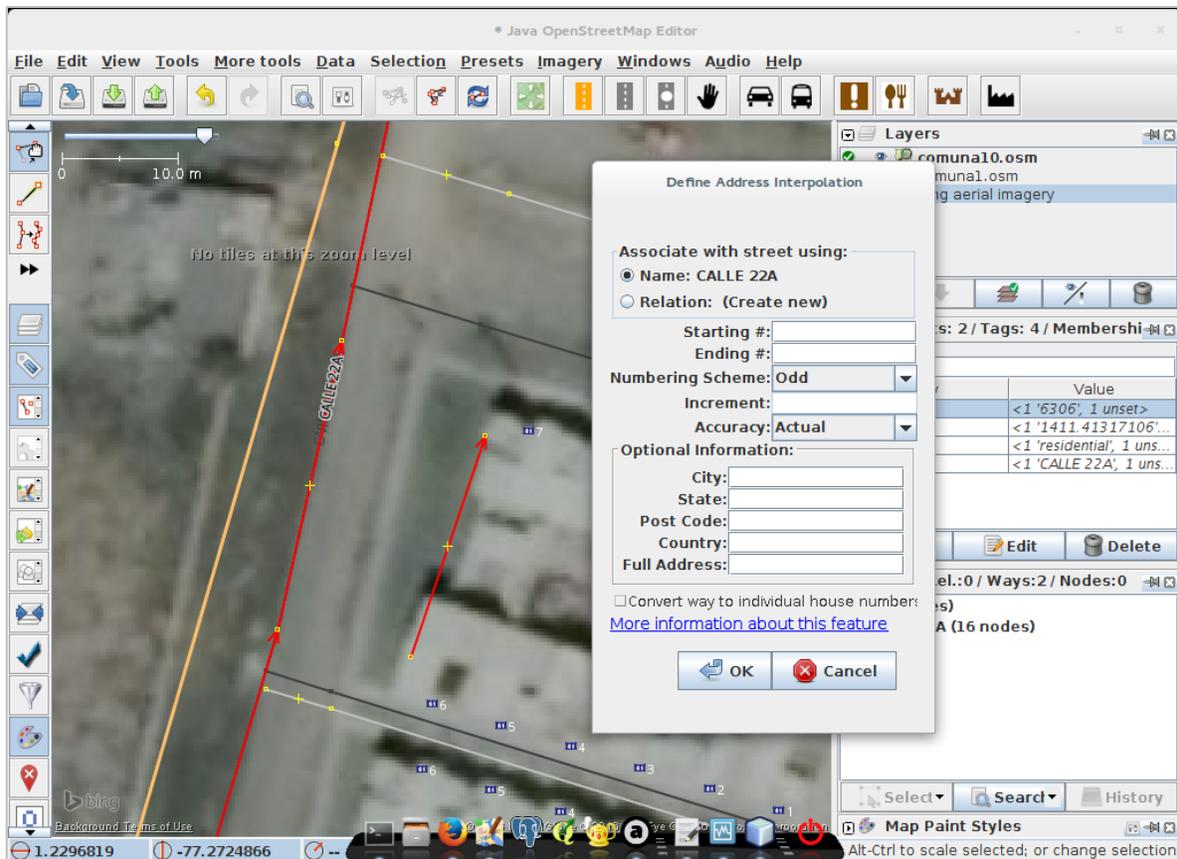


Figura 22. Complemento Address Interpolation en ejecución

A continuación se describen las principales opciones de este complemento que deben configurarse para obtener correctamente un conjunto de puntos que representan las direcciones urbanas del municipio de Pasto.

Para el caso de direcciones urbanas asignadas según la malla vial se tiene la siguiente configuración:

- La opción *Name* corresponde a la etiqueta `addr:street`
- Las opciones *Inicio (Starting)* y *Fin (Ending)* corresponden al componente distancia de las direcciones. Cabe aclarar que estos valores deben configurarse según la definición de direcciones de la cuadra a digitalizar. De acuerdo a la norma de asignación de direcciones del Instituto Geográfico Agustín Codazzi, un lado de la cuadra tendrá valores pares y el otro tendrá valores impares.
- El esquema de numeración define si la numeración de los puntos generados con la interpolación serán pares o impares.

- La opción *Full Address* corresponde al campo *addr:full*
- Debe habilitarse la opción *Convertir vía en direcciones individuales*.
- Los demás campos deben dejarse vacíos.

La figura 23 muestra un ejemplo de configuración de estos valores en el cuadro de configuración de AddressInterpolation. La figura 24 muestra el resultado de la interpolación de puntos generando direcciones urbanas.

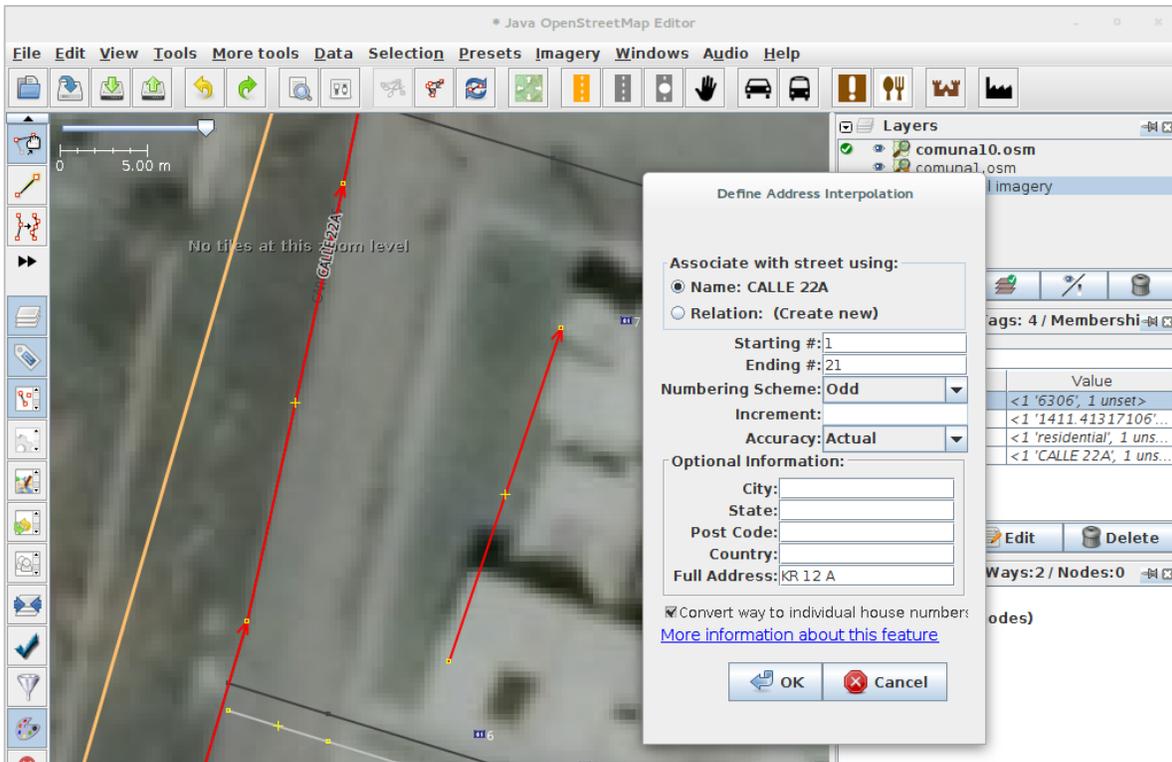


Figura 23. Configuración de valores para generar puntos iniciando desde la distancia 1 y terminando en la distancia 21 (Impar).

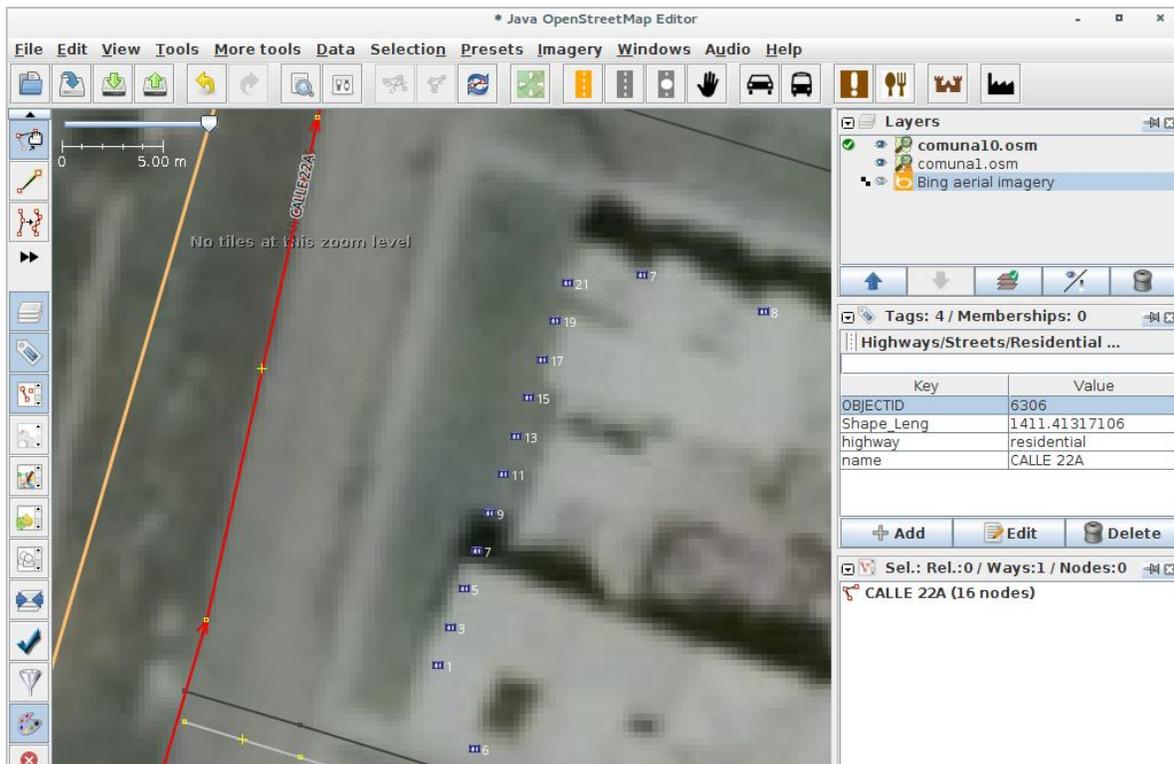


Figura 24. Puntos generados con el complemento AddressInterpolation

Para el caso de direcciones urbanas asignadas según la nomenclatura Barrio Manzana Predio se tiene la siguiente configuración:

- La opción *Name* corresponde a la etiqueta *addr:street*, para este caso de direcciones urbanas debe eliminarse de los puntos generados
- Las opciones *Inicio (Starting)* y *Fin (Ending)* corresponden al componente distancia de las direcciones. Cabe aclarar que estos valores deben configurarse según la definición de direcciones de la cuadra a digitalizar.
- El esquema de numeración debe definirse en la opción Todos, con lo cual se definirán valores continuos con incremento en 1
- La opción *Full Address* corresponde al campo *addr:full*
- Debe habilitarse la opción Convertir vía en direcciones individuales.
- Los demás campos deben dejarse vacíos.

La figura 25 muestra un ejemplo de configuración de estos valores en el cuadro de configuración de AddressInterpolation. La figura 24 muestra el resultado de la interpolación de puntos generando para este caso direcciones urbanas.

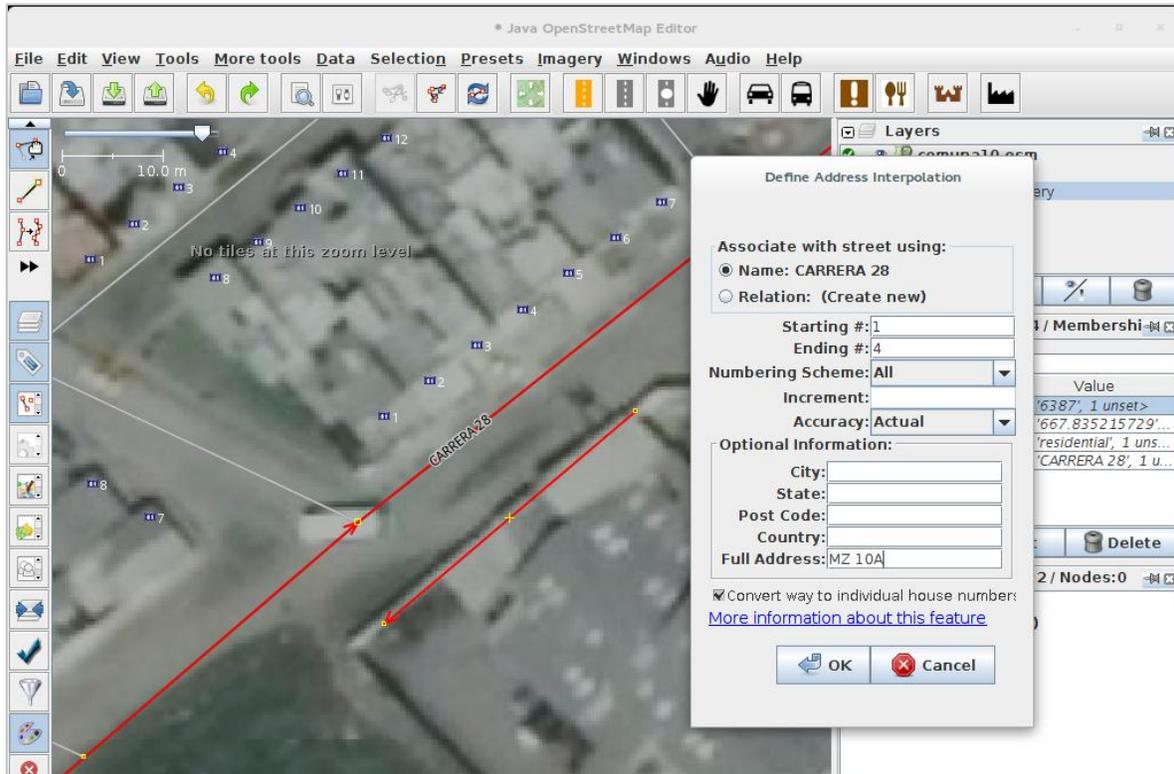


Figura 25. Configuración de valores para generar puntos iniciando desde el predio con identificación 1 hasta el predio con identificación 4.

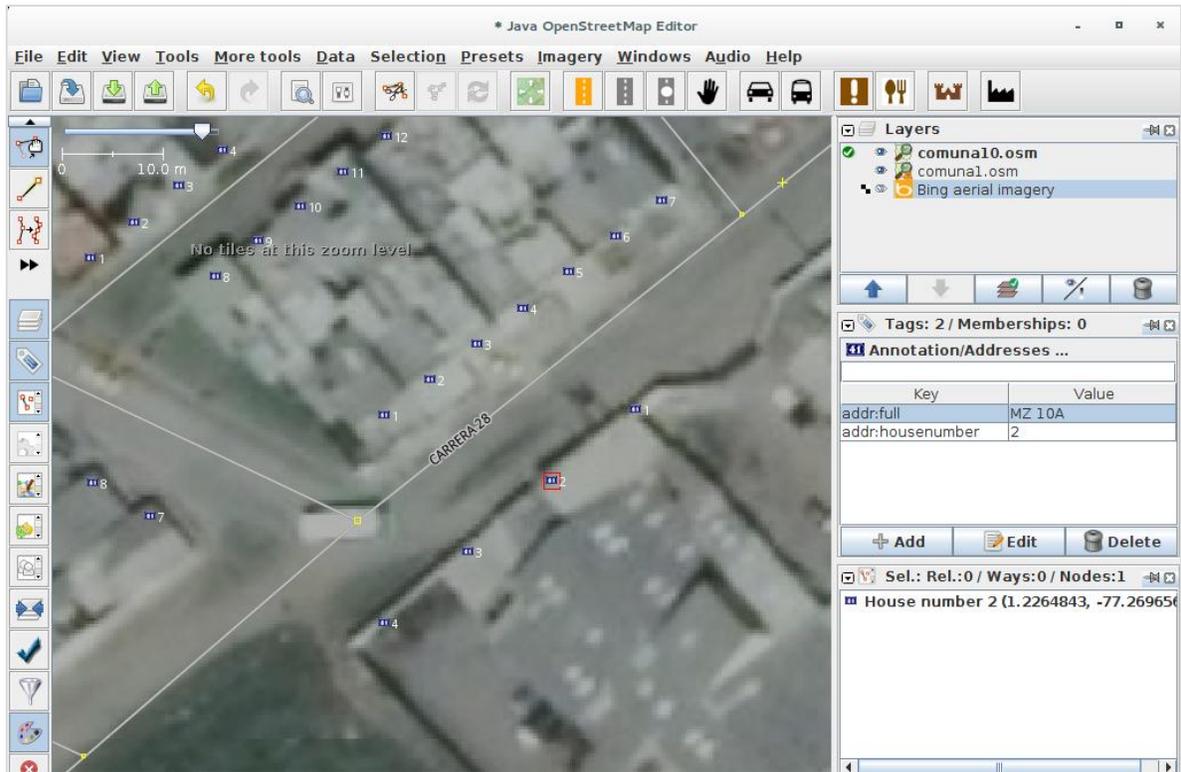


Figura 26. Puntos generados y sus etiquetas.

### 3. INSTALACION GEOCODER PASTO

A continuación se describen los requerimientos mínimos, instalación y uso del geocodificador de direcciones urbanas. Los pasos descritos fueron ejecutados sobre la plataforma de sistema operativo Linux/Ubuntu en su versión 14.04. Los archivos requeridos para la instalación pueden descargarse del repositorio <http://grias.udenar.edu.co>

#### 3.1 Software requerido:

##### Gestor de bases de datos PostgreSQL con extensión PostGIS

El geocodificador fue implementado con el gestor PostgreSQL en su versión 9.4. Para el manejo de datos espaciales fue utilizado PostGIS en su versión 2.1. El acceso y manipulación de la base de datos fue realizada con el entorno Pgadmin3.

Instalación librerías requeridas:

```
$ sudo apt-get install libxml2 libxml2-dev libproj-dev libgeos-3.4.2  
libgdal-dev libjson0 libjson0-dev
```

Instalación del gestor PostgreSQL 9.4 - PostGIS 2.1 - Pgadmin3:

```
$ sudo apt-get install postgresql-9.4 postgresql-9.4-postgis-2.1  
postgresql-client-9.4 postgresql-server-dev-9.4 postgresql-contrib-9.4  
pgadmin3
```

Configuración de la contraseña del usuario postgres:

```
$ su postgres  
$ psql postgres
```

```
=# ALTER USER 'postgres' WITH PASSWORD 'contraseña'  
=# \q
```

## **Extensión para PostgreSQL pg\_similarity**

Fue utilizado el algoritmo jarowinkler de la extensión pg\_similarity, permitiendo calcular la similitud entre dos cadenas de texto. Esta funcionalidad es utilizada en la estandarización del barrio ingresado junto con la dirección asignada según la nomenclatura barrio – manzana - predio.

Instalación:

```
$ cd pg_similarity
$ USE_PGXS=1 make
$ USE_PGXS=1 make install
```

## **Librerías de Python para la migración de datos OSM**

Para la migración de datos OSM a PostgreSQL se utilizó la plataforma del lenguaje de programación Python en su versión 2.7, el adaptador de bases de datos Psycopg2 para la conexión con PostgreSQL y la librería GDAL (Geospatial Data Abstraction Library) para el manejo de las geometrías.

Instalación Python 2.7 – Psycopg2 – GDAL:

```
$ sudo apt-get install python2.7 python-psycopg2 python-gdal
```

## **Instalación del esquema de datos y funciones de geocodificación de direcciones:**

Cabe mencionar que esta instalación puede realizarse en una base de datos nueva o existente, ya que el geocodificador se instalará en un esquema de datos independiente llamado geocoder. Para instalar este esquema de datos y las funciones del geocodificador debe ejecutarse las siguientes líneas de comandos:

```
$ su postgres
$ psql -c 'CREATE DATABASE geocoder;'
$ psql -d geocoder -c '\i
/<ruta_archivos>/geocoder/install_geocoder.sql'
```

## 3.2 Migración de datos OSM a PostgreSQL

Para la migración de los datos almacenados en los archivos OSM se requieren de los scripts codificados en los archivos **polygon\_linestring\_migration.py** y **points\_migration.py**

### Migración de barrios y comunas.

El script `polygon_linestring_migration.py` se encarga de leer los archivos OSM que corresponden a los barrios y comunas del municipio de Pasto, recorrer la estructura de datos XML almacenada, identificar los datos de cada barrio o comuna e insertarlos como nuevos registro en la relación barrios o comunas según se haya especificado. A continuación se especifica la línea de ejecución por terminal con sus posibles parámetros:

```
$ ./<ruta_archivo>/polygon_linestring_migration.py <parámetros>
```

Donde los parámetros ingresados pueden ser:

- `-f <ruta del archivo>`: Ruta donde se encuentra el archivo OSM, valor obligatorio.
- `-h <dirección IP>`: Dirección IP del servidor donde está instalado el gestor PostgreSQL. Por defecto está configurado localhost.
- `-p <puerto>`: Puerto de conexión con el gestor PostgreSQL. Por defecto está configurado el puerto 5432
- `-d <nombre base de datos>`: Nombre de la base de datos donde está instalado el esquema de datos. Este valor es obligatorio.
- `-u <nombre usuario>`: Nombre del usuario que administra la base de datos. Por defecto está configurado postgres.
- `-w <contraseña>`: Contraseña del usuario ingresado, valor obligatorio.
- `-o <id_tipo_geometría>`: Selección de un tipo de geometría. Los valores permitidos son l (geometría tipo Linestring) o p (geometría tipo Polygon).
- `-t <id_tabla_destino>`: Selección de la tabla donde se almacenará cada geometría procesada con sus atributos. Los valores permitidos son: c (tabla comunas), n (tabla barrios), r (tabla vías).

La instrucción de ejecución para migrar los datos que corresponden a las comunas es la siguiente:

```
$ ./<ruta_archivo>/polygon_linestring_migration.py -f data/communes.osm  
-d geocoder -u postgres -w 1234 -o p -t c
```

La instrucción de ejecución para migrar los datos que corresponden a los barrios es la siguiente:

```
$ ./<ruta_archivo>/polygon_linestring_migration.py -f  
data/neighborhoods.osm -d geocoder -u postgres -w 1234 -o p -t c
```

La instrucción de ejecución para migrar los datos que corresponden a las vías es la siguiente:

```
$ ./<ruta_archivo>/polygon_linestring_migration.py -f data/roads.osm -d  
geocoder -u postgres -w 1234 -o l -t r
```

### **Migración de direcciones urbanas.**

El script `points_migration.py` se encarga de leer los archivos OSM que corresponden a las direcciones urbanas, recorrer la estructura de datos XML almacenada, identificar las direcciones recopiladas según la malla vial y según la nomenclatura barrio - manzana – identificación de predio e insertarlas en su respectiva tabla.

La línea de ejecución por terminal con sus posibles parámetros es la siguiente:

```
$ ./<ruta_archivo>/points_migration.py <parámetros>
```

Donde los parámetros ingresados pueden ser:

- `-f <ruta del archivo>`: Ruta donde se encuentra el archivo OSM, valor obligatorio.
- `-h <dirección IP>`: Dirección IP del servidor donde está instalado el gestor PostgreSQL. Por defecto está configurado localhost.
- `-p <puerto>`: Puerto de conexión con el gestor PostgreSQL. Por defecto está configurado el puerto 5432
- `-d <nombre base de datos>`: Nombre de la base de datos donde está instalado el esquema de datos. Este valor es obligatorio.

- `-u <nombre_usuario>`: Nombre del usuario que administra la base de datos. Por defecto está configurado postgres.
- `-w <contraseña>`: Contraseña del usuario ingresado, valor obligatorio.

La instrucción de ejecución para migrar los datos que corresponden a las comunas es la siguiente:

```
$ ./<ruta_archivo>/points_migration.py -f data/addresses.osm -d
geocoder -u geocoder -u postgres -w 1234
```

### Asignación de identificadores de barrios y comunas a direcciones urbanas

Dado que la información sobre el barrio y comuna al que pertenece cada dirección urbana no fue recopilada, esta información se encuentra en sentencias creadas con el SQL utilizando algunas de las funciones disponibles en la extensión PostGIS para el análisis de relaciones espaciales entre geometrías también llamadas Topologías.

Para actualizar estos datos debe ejecutarse la siguiente instrucción:

```
$ su postgres
$ psql -d geocoder -c "\i /<ruta_archivos>/geocoder/update_fields.sql"
```

### 3.3 Prueba de funcionamiento del geocodificador

Para verificar el funcionamiento del geocodificador inicialmente debe ingresarse a la base de datos donde está instalado el geocodificador:

```
$ su postgres
$ psql geocoder;
```

Posteriormente se ejecuta las siguientes consultas SQL, como resultado se tendrá la información urbana correspondiente a la dirección ingresada:

```
=# SELECT geocode_address(`kra 20 13 A 45`);

                                geocode_address
-----
("KR 20 13 A 45", "LAS AMERICAS", "COMUNA 1", -602713.06642356, 134491.159088312)
```

Si la dirección es correctamente normalizada y encontrada dentro del repositorio de información urbana, la función retornará la información geográfica asociada a esa dirección.

Para acceder a los elementos de forma independiente, se debe ejecutar la siguiente instrucción SQL:

```
=# WITH geocode AS (  
    SELECT geocode_address('Kra 20 13 A 45') AS result  
)  
SELECT  
    (result).address AS direccion_normalizada,  
    (result).neighborhood AS barrio,  
    (result).commune AS comuna,  
    (result).lon AS longitud,  
    (result).lat AS latitud  
FROM  
    geocode;
```

El resultado será el siguiente

direccion_norm	barrio	comuna	longitud	latitud
KR 20 13 A 45	LAS AMERICAS	COMUNA 1	-8602713.06642356	134491.159088312

Un ejemplo similar puede ejecutarse para la geocodificación de direcciones según la nomenclatura barrio manzana predio:

```
=# WITH geocode AS (  
    SELECT geocode_address('Br La Paz', 'Manzana K Casa 9') AS result  
)  
SELECT  
    (result).address AS direccion_normalizada,  
    (result).neighborhood AS barrio,  
    (result).commune AS comuna,  
    (result).lon AS longitud,  
    (result).lat AS latitud  
FROM  
    geocode;
```

El resultado será el siguiente

direccion_norm	barrio	comuna	longitud	latitud
MZ K CS 9	LA PAZ	COMUNA 4	-8600523.43231094	133201.206467672

## **4. APORTES DEL GEOCODER PASTO**

### **4.1 Carácter Original o innovativo del GEOCODER PASTO**

Hasta el momento, el municipio de Pasto no contaba con una herramienta libre que realice el proceso de traducción de direcciones urbanas en coordenadas geográficas latitud y longitud, con el fin de georreferenciarlas y visualizarlas en una cartografía del municipio, de acuerdo al uso que se le vaya a dar a estas direcciones

Con el Geocoder Pasto, el municipio dispone de un geocodificador de código abierto que permite traducir nomenclaturas urbanas a coordenadas geográficas longitud/latitud para luego poder visualizar e identificar zonas geográficas específicas de acuerdo al mapa temático que se utilice.

Por ser un geocodificador bajo licencia de software libre, permitirá a las diferentes organizaciones gubernamentales o privadas del municipio de Pasto y del país reutilizar el geocodificador construido y acoplarlo a cualquier visualizador de mapas con el fin de registrar e identificar según su contexto funcional, zonas de interés, sitios turísticos, rutas de buses, paraderos, zonas de posibles desastres, entre otras en la cartografía del municipio de Pasto.

### **4.2 Aporte al fortalecimiento de la capacidad nacional de investigación, innovación y desarrollo tecnológico**

Al utilizar tecnologías y plataformas libres para el desarrollo de la herramienta geocoder Pasto, su licencia es libre y por lo tanto el resultado de esta investigación, tanto su código, como su documentación están disponibles para ser descargados en el sitio del grupo de investigación GRIAS del departamento de Sistemas de la facultad de Ingeniería de la Universidad de Nariño en el enlace <http://grias.udenar.edu.co>, como un aporte para fortalecer la capacidad nacional de investigación, innovación y desarrollo tecnológico en el área de la geocodificación. De esta manera, otros investigadores de diferentes ciudades de

Colombia, o de otros países, que no cuenten con un geocodificador de direcciones urbanas, pueden tomar a geocoder Pasto, como modelo, para construir su propio geocodificador.

Este proyecto aporta las bases de conocimiento para futuros proyectos de investigación y trabajos de grado que se realicen en la Universidad de Nariño, en Colombia o en el exterior.

### **4.3 Aporte a la consolidación de un área estratégica**

El geocoder Pasto es un gran aporte al área de la georreferenciación ya que aporta las bases de conocimiento y un modelo para que se emprendan nuevos proyectos que tengan que ver con el proceso de traducción de direcciones urbanas en coordenadas geográficas latitud y longitud, en los municipios que no cuenten con este tipo de geocodificadores o para nuevos proyectos en la ciudad de Pasto que haga uso de este geocoder y que la encamine hacia una ciudad inteligente, con el uso progresivo de las Tecnologías de la Información y Comunicación (TIC).

## **5. IMPACTOS DEL GEOCODER PASTO**

### **5.1 Impacto científico y tecnológico**

A nivel de la Universidad de Nariño, la construcción del geocodificador de direcciones urbanas del municipio de Pasto Geocoder Pasto es uno de los desarrollos más importantes en la línea de investigación de Sistemas de Georreferenciación del grupo de investigación GRIAS, de departamento de Sistemas de la Facultad de Ingeniería, aportando bases de conocimiento para futuros proyectos y trabajos de grado.

A nivel del municipio de Pasto, el Geocoder Pasto facilitará los desarrollos de futuros proyectos que involucren la traducción de direcciones urbanas a coordenadas geográficas para ser visualizados en la cartografía del municipio.

A nivel nacional, el Geocoder Pasto se convierte en un referente para desarrollar nuevos geocodificadores en municipios donde no los hay o su acceso no es libre y gratuito.

## **5.2 Impacto en el medio ambiente y sociedad**

El impacto de esta herramienta ya es medible en los trabajos realizados recientemente para el municipio de Pasto como lo es el visor cartográfico Pasto View que gracias al geocoder Pasto, permite buscar y visualizar en un mapa del municipio de Pasto, la localización geográfica de una dirección urbana. Por otra parte, SIGEODEP SIG utiliza el geocoder para la localización y visualización en la cartografía del municipio de Pasto, todos los eventos delictivos fatales y no fatales que sucedieron en este municipio y que son registrados en el observatorio del delito del municipio de Pasto. El aplicativo Rikhuna acopla este geocodificador para la localización y visualización inteligente de sitios de interés del municipio de Pasto y finalmente el App SitApp utiliza este geocodificador para la localización y visualización inteligente de paraderos de buses del sistema estratégico de transporte público del municipio de Pasto.

## **5.3 Aspectos económicos y financieros**

Por ser el geocodificador de direcciones urbanas del municipio de Pasto “Geocoder Pasto” desarrollado bajo software libre, su distribución es libre y además gratuita. Hecho que permite que cualquier organización gubernamental o privada del municipio de Pasto o Colombia lo pueda utilizar sin ningún tipo de restricción, siempre y cuando los futuros desarrollos sean también libres y gratuitos.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] O. Huisman y R. A. By, *Principles of Geographic Information Systems: An Introductory Textbook*. International Institute for Geo-Information Science and Earth Observation (ITC), 2009.
- [2] Refractions Research, «PostGIS Extras, Tiger Geocoder», *Chapter 12. PostGIS Extras*, 2004. [En línea]. Disponible en: <http://postgis.net/docs/Extras.html>.
- [3] US Census Bureau, «TIGER Products». [En línea]. Disponible en: <http://www.census.gov/geo/maps-data/data/tiger.html>.
- [4] I. Moya Honduvilla y M. Á. Manso Callejo, «Servicio web de Geocodificación para Cartociudad», en *Actas de I Jornadas Ibéricas de Infra-estructuras de Datos Espaciales, IIIDE 2010*, Lisboa, Portugal, 2010.
- [5] Procálculo Prosis S.A., «Mapas.com.co», *Mapas.co, Cartografía e Imágenes Satelitales*, 2013. [En línea]. Disponible en: <http://www.mapas.com.co>.
- [6] F. Benítez, «Mapas.com.co», *ArcGIS - www.mapas.com.co*, 2012. [En línea]. Disponible en: <http://www.arcgis.com/home/item.html?id=b197b9067c534d9b874778cca16c71a0>.
- [7] J. A. V. Benjumea y D. H. Álvarez, «Proceso de Geocodificación de direcciones en la Ciudad de Medellín: una técnica determinística de georreferenciación de direcciones», *Rev. Ing. USBMed*, vol. 4, n.º 1, pp. 6-21, 2013.
- [8] J. Román y I. Solarte, «Geopasto: Un sistema de información geográfica web orientado al apoyo para la toma de decisiones basados en el plan de ordenamiento territorial del municipio de Pasto», *Línea de Software Y Manejo de Información*, Universidad de Nariño, Pasto, Colombia, 2008.
- [9] C. Arteaga y M. Erazo, «Atlas: Herramienta de cartografía web y geocodificación para el desarrollo de sistemas híbridos en áreas urbanas sobre J2EE y PostgreSQL», *Línea de Software Y Manejo de Información*, Universidad de Nariño, Pasto, Colombia, 2008.