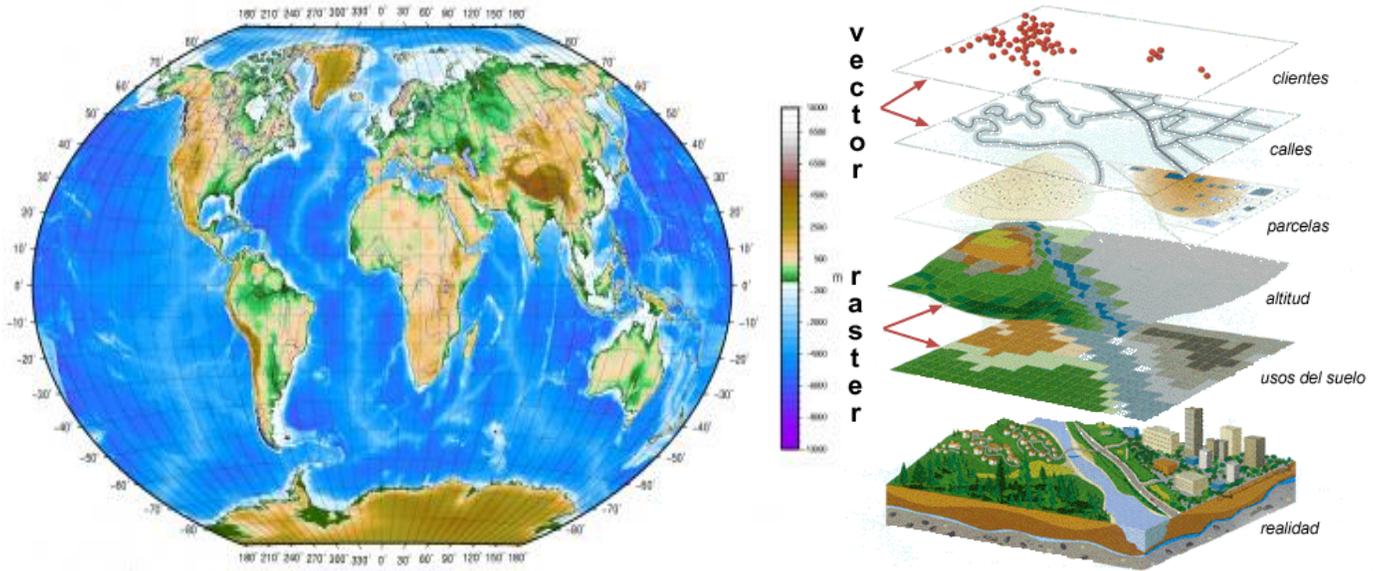


VISOR CARTOGRÁFICO

PASTO VIEW



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

INTRODUCCIÓN	5
1. PROCESO DE ANÁLISIS DEL VISOR CARTOGRÁFICO PASTO VIEW	6
1.1 Actores del Sistema	6
1.2 Funciones del sistema	6
1.3 Casos de uso:	7
2. ARQUITECTURA DEL VISOR CARTOGRÁFICO PASTO VIEW	11
3. INSTALACION VISOR CARTOGRÁFICO PASTO VIEW	17
3.1 Instalación Servidor de Aplicaciones Glassfish	17
4. MANEJO VISOR CARTOGRÁFICO PASTO VIEW	21
4.1 Búsqueda direcciones.	21
4.2 Archivos CSV.	24
REFERENCIAS BIBLIOGRÁFICAS	29
ANEXO	30
INSTALACIÓN DEL GEOCODIFICADOR DE DIRECCIONES URBANAS PARA EL MUNICIPIO DE PASTO.	30
1. Software requerido:	30
1.1. Gestor de bases de datos PostgreSQL con extensión PostGIS	30
1.2. Extensión para PostgreSQL pg_similarity	31
1.3. Librerías de Python para la migración de datos OSM	31
2. Instalación del esquema de datos y funciones de geocodificación de direcciones:	31
3. Migración de datos OSM a PostgreSQL	32
3.1. Migración de barrios y comunas:	32
3.2. Migración de direcciones urbanas:	33
4. Asignación de identificadores de barrios y comunas a direcciones urbanas:	34
5. Prueba de funcionamiento del geocodificador:	34

Lista de Figuras

<i>Figura 1. Diagrama de caso de uso Consulta de direcciones urbanas</i>	<i>9</i>
<i>Figura 2. Diagrama de caso de uso Procesamiento de archivos CSV</i>	<i>10</i>
<i>Figura 3. Logo visor cartográfico PASTO—VIEW</i>	<i>11</i>
<i>Figura 4. Arquitectura del visor cartográfico PASTO-VIEW</i>	<i>12</i>
<i>Figura 5. Búsqueda de direcciones, funcionalidad implementada en PASTO VIEW</i>	<i>13</i>
<i>Figura 6. Procesamiento de archivo CSV, datos agrupados por el atributo Cluster. Funcionalidad implementada en PASTO VIEW</i>	<i>14</i>
<i>Figura 7. Paquete geocoder visor PASTO VIEW</i>	<i>14</i>
<i>Figura 8. Código de conexión del visor PASTO VIEW con el geocodificador de direcciones urbanas.</i>	<i>15</i>
<i>Figura 9. Creación nueva conexión a base de datos</i>	<i>18</i>
<i>Figura 10. Configuración propiedades de conexión.</i>	<i>18</i>
<i>Figura 11. Creación recurso JDBC</i>	<i>19</i>
<i>Figura 12. Carga del aplicativo web en el servidor glassfish</i>	<i>19</i>
<i>Figura 13. Visor PASTOVIEW en ejecución</i>	<i>20</i>
<i>Figura 14. Menú búsqueda de direcciones urbanas</i>	<i>21</i>
<i>Figura 15. Búsqueda de direcciones tipo Manzanas.</i>	<i>22</i>
<i>Figura 16. Búsqueda de dirección según la malla vial en PASTOVIEW</i>	<i>23</i>
<i>Figura 17. Búsqueda de dirección según la nomenclatura barrio - manzana predio en PASTOVIEW</i>	<i>24</i>
<i>Figura 18. Funcionalidad de subida y procesamiento de datos CSV implementada en PASTO VIEW</i>	<i>25</i>
<i>Figura 19. Subida correcta e incorrecta de archivos CSV.</i>	<i>25</i>
<i>Figura 20. Configuración del archivo 'address.csv', el cual contiene direcciones asociadas al caso Manzanas.</i>	<i>26</i>
<i>Figura 21. Configuración del archivo 'cluster-fatales.csv', seleccionando los campos 'longitud' y 'latitud' presentes en este archivo.</i>	<i>27</i>
<i>Figura 22. Procesamiento del archivo 'clusters-fatales.csv' sin agrupar.</i>	<i>27</i>
<i>Figura 23. Procesamiento del archivo 'clusters-fatales.csv'. Los datos se agrupan según los valores del campo 'Clusters'.</i>	<i>28</i>

INTRODUCCIÓN

Aproximadamente un 70% de la información que se maneja en cualquier tipo de disciplina esta georreferenciada, es decir, que hace referencia a algún evento, fenómeno o problema ubicado dentro de una zona geográfica extensa (ej. países, regiones, departamentos, comunas o barrios) o reducida (cuadras, direcciones urbanas, coordenadas geográficas longitud y latitud). Según lo mencionado por [1], 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.

La georreferenciación como principal herramienta que proveen los SIG, permite el análisis de datos basado en ubicaciones específicas reconocibles de datos dentro de un área geográfica de estudio [2]. En nuestro país se han implementado sistemas de georreferenciación basados en límites geográficos específicos (regiones, departamentos, barrios, sectores, veredas, entre otros) para procesar los eventos y desplegarlos posteriormente en mapas temáticos. En muchos casos la información procesada contiene su dirección urbana de ocurrencia, una característica que de ser procesada, permitiría el análisis geográfico de los datos siniestrados 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.

En este documento se describe la construcción del visor cartográfico Pasto View que integrado a Geocoder Pasto [3], un geocodificador libre de direcciones urbanas del municipio de Pasto, permite al usuario acceder a la búsqueda de direcciones urbanas del municipio de Pasto, de manera georreferenciada y la visualización de estos datos soportando la geocodificación de direcciones urbanas asignadas según la malla vial y según la nomenclatura barrio - manzana - predio.

1. PROCESO DE ANÁLISIS DEL VISOR CARTOGRÁFICO PASTO VIEW

A continuación se describen aspectos del proceso de análisis en el lenguaje de modelamiento UML, realizado para definir las funcionalidades del visor cartográfico Pasto View.

1.1 Actores del Sistema

El visor Pasto View tendrá acceso sin restricciones a la consulta de direcciones urbanas a cualquier usuario que ingrese al sistema. Los usuarios podrán buscar direcciones urbanas, visualizando la ubicación espacial de las mismas.

1.2 Funciones del sistema

La tabla 1 muestra los requerimientos definidos para el visor cartográfico Pasto View.

Tabla 1. Requerimientos visor Pasto View

Búsqueda de direcciones urbanas		
Ref.	Función	Descripción
R1	Conexión con el geocodificador de direcciones urbanas.	El sistema debe conectarse al geocodificador de direcciones urbanas instalado en el gestor de base de datos PostgreSQL para realizar consultas.
R1.1	Búsqueda de direcciones urbanas según malla vial.	El sistema debe permitir al usuario consultar direcciones urbanas asignadas según la malla vial.
R1.2	Búsqueda de direcciones urbanas según nomenclatura barrio manzana predio.	El sistema debe permitir al usuario consultar direcciones urbanas asignadas según la nomenclatura barrio manzana predio.
R2	Visualización de direcciones encontradas.	El sistema debe permitir al usuario visualizar la ubicación de la dirección urbana sobre el mapa del municipio de Pasto sobre mapas temáticos.
R2.1	Visualización de información relacionada con la dirección encontrada	El sistema debe permitir al usuario visualizar la información urbana relacionada con la dirección urbana encontrada.

R3	Carga de cartográfica correspondiente al municipio de Pasto.	El sistema debe cargar el mapa satelital del municipio de Pasto que permita visualizar las calles y vías con aproximación a la realidad.
R4	Navegación sobre los mapas cargados.	El sistema debe permitir al usuario navegar sobre los mapas temáticos cargados (ampliar, reducir la visibilidad del mapa, activación/desactivación de capas),
Archivos CSV		
R5	Procesar archivos CSV	El sistema debe permitir procesar archivos separados por comas CSV con el fin de georreferenciar los registros que se encuentren almacenados.
R6	Seleccionar campos georreferenciados	El sistema debe permitir seleccionar el tipo de campos georreferenciados, los cuales pueden ser direcciones urbanas y/o coordenadas geográficas.
R6.1	Procesamiento de campo dirección urbana y barrio	El sistema debe permitir identificar el tipo de dirección urbana (Malla vial o Manzanas) presente entre los datos CSV y seleccionar los campos relacionados con direcciones urbanas, estos son dirección urbana y barrio.
R6.2	Procesamiento de coordenadas geográficas Longitud/Latitud	El sistema debe permitir seleccionar los campos donde se encuentren las coordenadas geográficas Longitud y Latitud. Estas coordenadas deben estar configuradas según el sistema de referencia espacial 3857.
R7	Agrupar datos	El sistema debe permitir agrupar los datos procesados de acuerdo a la selección de un determinado campo de datos.
R7.1	Visualizar capas de datos agrupados	El sistema debe permitir crear capas o mapas de visualización de datos geocodificados, de acuerdo a la configuración de agrupación de datos que se haya seleccionado.

1.3 Casos de uso:

Un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

La funcionalidad del visor Pasto View se describe en dos casos de uso

- Consulta de direcciones urbanas.

- Procesamiento de archivos separados por comas CSV.

▪ *Caso de uso 1. Consulta de direcciones urbanas.*

Caso de uso:	Consulta de direcciones urbanas
Actores del sistema:	Usuario invitado
Pre-condiciones:	<ul style="list-style-type: none"> ▪ Conexión correcta a base de datos del observatorio del delito. ▪ Geocodificador de direcciones urbanas instalado. ▪ Servidor de publicación de mapas temáticos en ejecución.
Post-condiciones:	Dirección urbana ubicada en el área urbana del municipio de Pasto de acuerdo a las coordenadas geográficas latitud/longitud retornadas por el geocodificador.
Flujo Básico	
<ol style="list-style-type: none"> 1. El sistema despliega los formularios de ingreso de direcciones urbanas según la malla vial y según la nomenclatura barrio – manzana – predio. 2. El usuario ingresa una dirección urbana en el formulario correspondiente y selecciona la opción Buscar. 3. El sistema captura la dirección urbana ingresada generando y ejecutando la consulta SQL que llama al geocodificador de direcciones urbanas. 4. El sistema recupera los datos arrojados por el geocodificador de direcciones. 5. El sistema crea un punto en el mapa temático de acuerdo a las coordenadas latitud longitud recibida. 6. El sistema muestra la información urbana de la dirección consultada sobre el punto ubicado. 7. El sistema muestra un mensaje informando al usuario que la dirección ingresada fue encontrada. 	
Flujo alternativo	
<ol style="list-style-type: none"> 4a. El sistema verifica que el geocodificador no obtuvo resultados asociados a la dirección consultada. 4b. El sistema muestra un mensaje de error informando que la dirección ingresada no fue encontrada. 	

La figura 1 muestra el diagrama de caso de uso de consulta de direcciones urbanas.

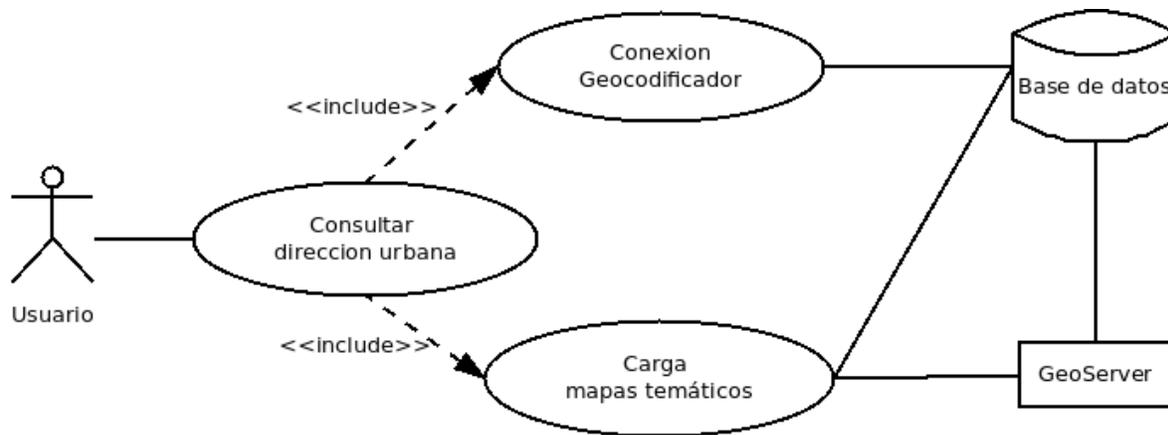


Figura 1. Diagrama de caso de uso Consulta de direcciones urbanas

▪ *Caso de uso Procesamiento de archivos CSV.*

Caso de uso:	Procesamiento de archivos CSV
Actores del sistema:	Usuario invitado
Pre-condiciones:	<ul style="list-style-type: none"> ▪ Conexión correcta a base de datos del observatorio del delito. ▪ Geocodificador de direcciones urbanas instalado. ▪ Servidor de publicación de mapas temáticos en ejecución.
Post-condiciones:	Mapa(s) de puntos asociado(s) al procesamiento de los registros encontrados en el archivo CSV.
Flujo Básico	
<ol style="list-style-type: none"> 1. El sistema despliega el formulario de selección y procesamiento de archivos CSV. 2. El usuario selecciona la opción 'Seleccionar archivo' y buscar el archivo que desea procesar. 3. El sistema procesa el archivo seleccionado cargando los nombres de cada campo encontrado en las listas de selección de campos. 4. El usuario configura las listas de campos de acuerdo a la estructura del archivo CSV y presiona el botón 'Procesar'. 5. El sistema procesa cada registro de acuerdo a los campos georreferenciados seleccionados. 6. El sistema despliega la capa de puntos con cada evento o registro procesado informando a usuario la cantidad de registros procesados exitosamente. 	
Flujo alternativo	
<ol style="list-style-type: none"> 2a. El sistema verifica que el archivo no tiene la extensión .csv. 2b. El sistema informa al usuario que el archivo no es válido. 4a. Si la opción Agrupar datos es habilitada, el usuario debe seleccionar el campo por el cual se agruparan los datos. 4b. El sistema procesa cada registro de acuerdo a los campos georreferenciados seleccionados. 4c. El sistema despliega la capa de puntos con cada evento o registro procesado informando a usuario la cantidad de registros procesados exitosamente. 	

La figura 2 muestra el diagrama de caso de uso de procesamiento de archivos CSV.

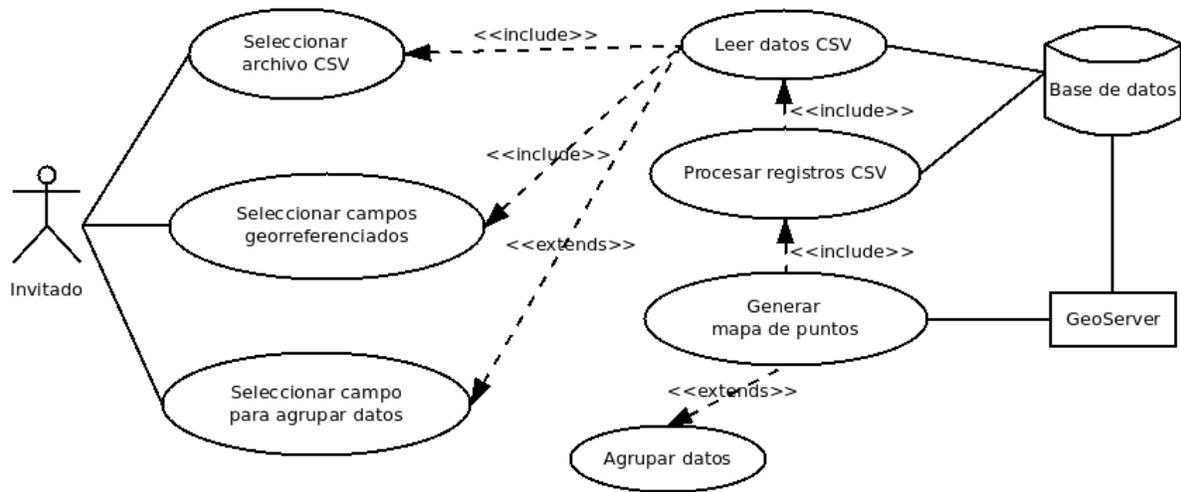


Figura 2. Diagrama de caso de uso Procesamiento de archivos CSV

2. ARQUITECTURA DEL VISOR CARTOGRÁFICO PASTO VIEW

El visor cartográfico PASTO VIEW, cuyo logo inicial se muestra en la figura 3, permite a cualquier usuario acceder al geocodificador de direcciones urbanas a través de una interfaz gráfica de usuario, ingresando direcciones urbanas y visualizando su correspondiente ubicación urbana de ser encontradas en el repositorio de datos.



Figura 3. Logo visor cartográfico PASTO—VIEW

Fuente: Esta investigación.

La construcción del visor cartográfico fue realizada en su totalidad bajo el sistema operativo *Linux Ubuntu* en su versión 14.04. El lenguaje de programación utilizado fue Java, con el entorno de desarrollo integrado (IDE) *Netbeans* versión 8.1. El servidor de aplicaciones utilizado fue *Glassfish* en su versión 3.2.

La arquitectura de PASTO-VIEW se basa en el patrón Modelo Vista Controlador (MVC), implementado en tres módulos: el modulo visor, el modulo núcleo y el modulo conexión a base de datos. La figura 4 muestra la arquitectura del visor cartográfico.

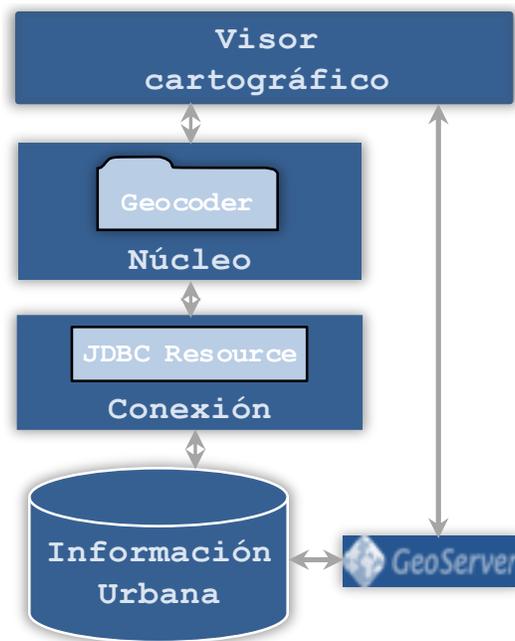


Figura 4. Arquitectura del visor cartográfico PASTO-VIEW

Fuente: Esta investigación.

- **Modulo Visor Cartográfico:** Permite el ingreso de direcciones urbanas para visualizar su ubicación urbana, así como también la carga y procesamiento de archivos separados por comas CSV. Este módulo se conecta con el modulo núcleo de forma directa.

Específicamente este módulo se compone de una página con extensión *xhtml* del *framework Java Server Faces* (JSF), encargada de interactuar con el usuario e intercambiar información con el controlador que hace parte del kernel de PASTO VIEW. Para la creación de la interfaz gráfica fue utilizado el *framework Primefaces* en su versión 3.4. Para la visualización de mapas fue utilizada la interfaz programable de aplicaciones (API) *OpenLayers* en su versión 3.14, la cual está escrita en el lenguaje de programación *JavaScript*. Esta interfaz se conecta con servicios como *Bing Maps* y *Open Street Maps* para la carga de mapas de referencia. A continuación, se describe la página *xhtml* de acceso principal dentro de este módulo:

- *index.xhtml*: Mediante esta página el usuario accede a dos funcionalidades: La consulta y visualización de la ubicación de direcciones urbanas en el área urbana del municipio de Pasto y la geocodificación de datos contenidos en archivos separados por comas CSV.

La búsqueda de direcciones puede realizarse según los dos casos de asignación de direcciones seleccionados para esta investigación: según malla vial ò según la nomenclatura barrio - manzana predio. La figura 5 muestra la implementación de esta búsqueda.

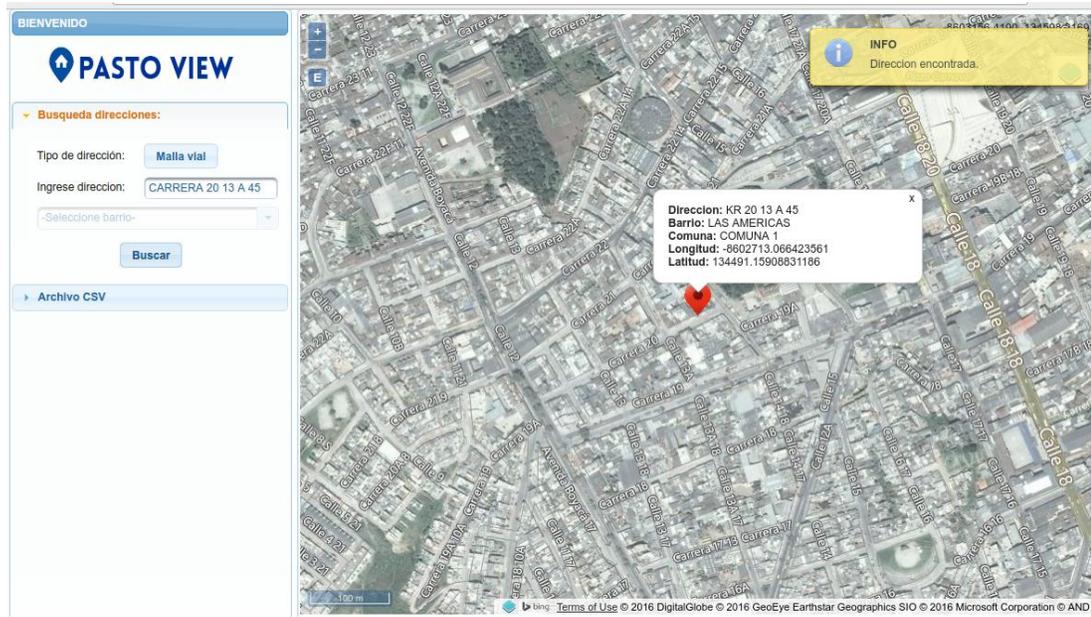


Figura 5. Búsqueda de direcciones, funcionalidad implementada en PASTO VIEW

Fuente: Esta investigación.

El procesamiento de archivos CSV se realiza mediante la selección de campos georreferenciados como pueden ser direcciones urbanas o coordenadas latitud/longitud las cuales deben estar procesados con el sistema de referencia espacial EPSG:3857. De igual forma se dispone de una opción que permite agrupar datos según el campo que se haya seleccionado. Finalmente, el usuario podrá consultar mediante un dialogo de información las características o campos restantes asociados a los puntos desplegados haciendo clic sobre cada uno de estos. La figura 6 muestra la implementación de este procesamiento.

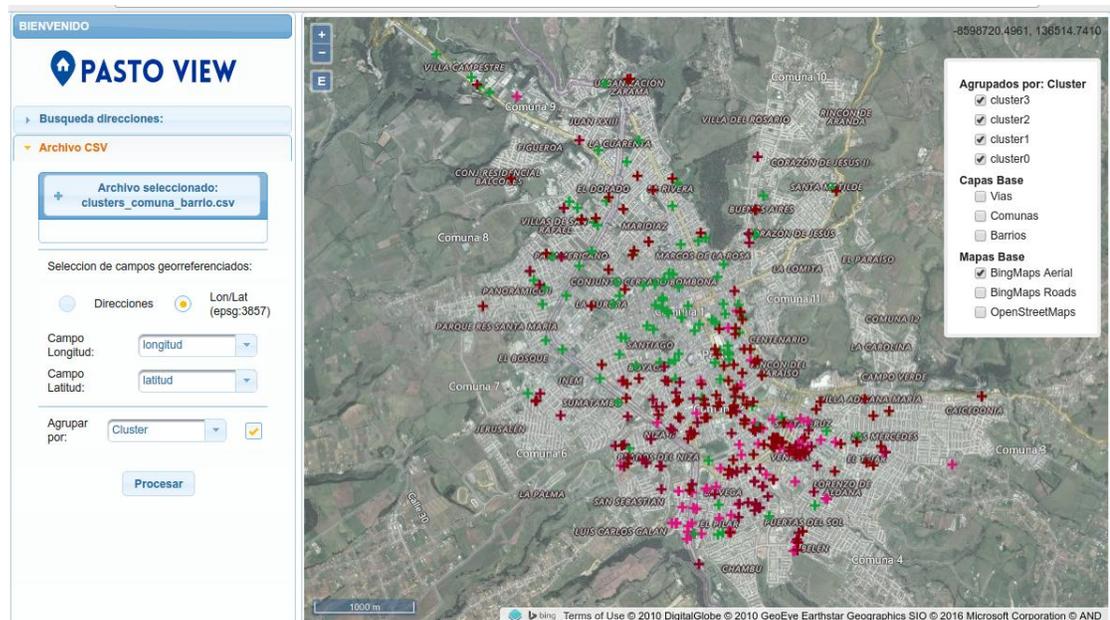


Figura 6. Procesamiento de archivo CSV, datos agrupados por el atributo Cluster. Funcionalidad implementada en PASTO VIEW

Fuente: Esta investigación

- **Módulo Núcleo:** En este módulo se encuentra un único paquete encargado de procesar la información solicitada por el usuario, el cual se describe a continuación:
 - Paquete Geocoder: En este paquete se almacenan una única clase encargada de procesar los datos asociados con la búsqueda de direcciones urbanas que son solicitados por el usuario. La estructura de este paquete puede verse en la figura 7.

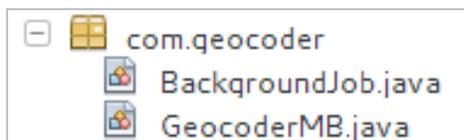


Figura 7. Paquete geocoder visor PASTO VIEW

Fuente: Esta investigación.

BackgroundJob.java es la clase encargada de eliminar los datos temporales almacenados en la base de datos generados por los usuarios que acceden al sistema en el procesamiento de archivos CSV. El único método de esta clase denominado *hoursJob()* se ejecuta cada hora eliminando aquellos datos cuyo *timestamp* (fecha y

hora) de inserción sea superior a una hora en comparación con el *timestamp* que corresponde al momento de la ejecución del procedimiento.

GeocoderMB.java es la clase encargada de conectarse con el geocodificador de direcciones urbanas para la búsqueda de direcciones urbanas y el procesamiento de los archivos CSV. Esta clase tiene dos métodos principales: Uno de ellos se denomina *processAddress()* el cual genera y ejecuta una sentencia SQL sobre el geocodificador de direcciones solicitando los datos asociados a la dirección buscada para ser recuperados y visualizados posteriormente. El procedimiento de ejecución se puede observar en la figura 8.

```
sql = ""
+ "WITH geocoded AS(\n"
+ "    SELECT\n"
+ "        geocode_address(' + searchedRAddress + ') AS result\n"
+ ")\n"
+ "SELECT\n"
+ "    (result).address,\n"
+ "    (result).neighborhood,\n"
+ "    (result).commune,\n"
+ "    (result).lon,\n"
+ "    (result).lat\n"
+ "FROM\n"
+ "    geocoded\n"
+ "WHERE\n"
+ "    (result).lon IS NOT NULL AND (result).lat IS NOT NULL;";

try {
    ResultSet rs = connectionJdbcMB.consult(sql);

    if (rs.next()) {
        resultAddress = rs.getString("address");
        resultNeighborhood = rs.getString("neighborhood");
        resultCommune = rs.getString("commune");
        resultLongitude = rs.getDouble("lon");
        resultLatitude = rs.getDouble("lat");
        JsFUtil.addSuccessMessage("Dirección encontrada");
    } else {
        resultLongitude = 0;
        resultLatitude = 0;
        JsFUtil.addErrorMessage("Dirección no encontrada.");
    }
} catch (SQLException e) {
    System.out.println("Error 1 en " + this.getClass().getName() + ":" + e.toString());
}
```

Figura 8. Código de conexión del visor PASTO VIEW con el geocodificador de direcciones urbanas.

Fuente: Esta investigación.

El método *processCSV()* se encarga de leer archivos CVS, identificando el nombre de los campos y procesando cada registro almacenando en dicho archivo de acuerdo con la configuración seleccionada por el usuario.

- **Conexión a base de datos:** Para conectar PASTOVIEW con el repositorio de datos, se utiliza la anotación *@Resouce* disponible con la edición empresarial de *Java J2EE*, esta permite referenciar y utilizar una conexión JDBC creada desde el servidor de aplicaciones *Glassfish*.

3. INSTALACION VISOR CARTOGRÁFICO PASTO VIEW

Como prerequisite de la instalación del visor cartográfico Pasto View se debe tener instalado el geocodificador de direcciones urbanas *Geocoder Pasto*, instalación descrita en el Anexo A.

La instalación de visor cartográfico Pasto View se realizó bajo el sistema operativo Linux/Ubuntu en su versión 14.04.

3.1 Instalación Servidor de Aplicaciones Glassfish

Primero se descarga la versión 3.1.2 para GNU/Linux desde el sitio web de Oracle y se ejecuta lo siguiente:

```
$ sh ogs-3.1.2.2-unix-ml.sh
```

Se debe descargar el driver JDBC de postgresql desde el sitio web de PostgreSQL y copiarlo en el directorio *glassfish3/glassfish/domains/domain1/lib*. Para iniciar el servidor se ejecuta lo siguiente:

```
$ ./glassfish3/glassfish/bin/startserv
```

En el navegador se ingresa el URL *http://localhost:4848*. A continuación aparece la consola de administración, en la cual se tecléa el usuario y contraseña que se suministró en el momento de la instalación de *glassfish*.

Ir a “*Resources/JDBC/Connection Pools*” y crear una nueva conexión con los datos que muestra la Figura 9 y luego clic en <siguiente>.

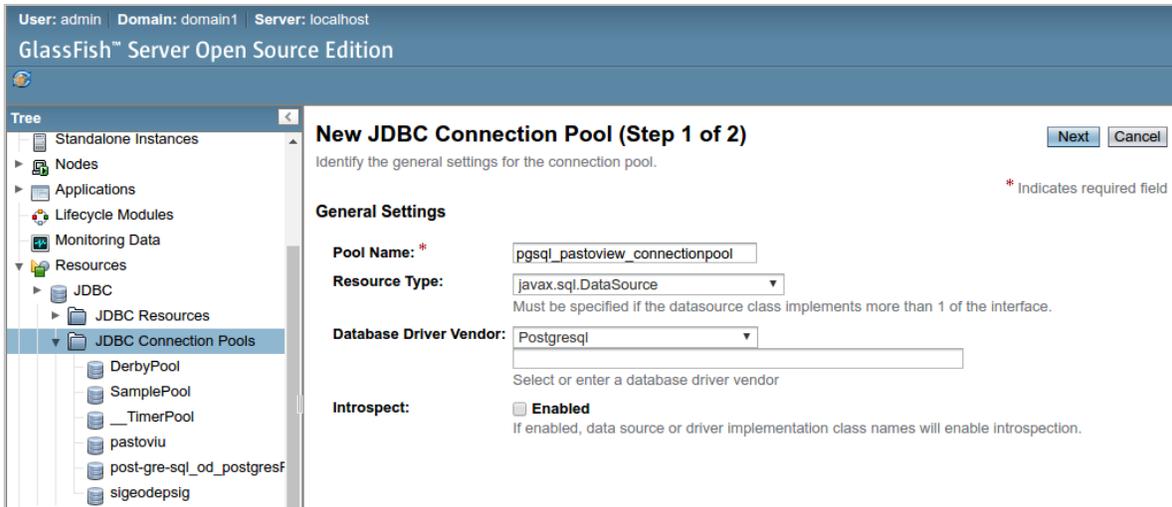


Figura 9. Creación nueva conexión a base de datos

Seleccionar el origen de datos de nombre de clase *org.postgresql.ds.PGSimpleDataSource* y escribir las siguientes propiedades adicionales como se muestra en la figura 10.

Additional Properties (5)			
<input type="checkbox"/> <input type="checkbox"/> <input type="button" value="Add Property"/> <input type="button" value="Delete Properties"/>			
	Name	Value	Description:
<input type="checkbox"/>	password	<contraseña_usuario_postgresql>	
<input type="checkbox"/>	user	<nombre_usuario_postgresql>	
<input type="checkbox"/>	port	5432	
<input type="checkbox"/>	host	localhost	
<input type="checkbox"/>	databaseName	geocoder	

Figura 10. Configuración propiedades de conexión.

Con esto se guarda las conexiones y se da clic en <finalizar > para guardar la conexión. Luego en “Resources/JDBC/JDBC Resources” se escribe en nombre JNDI y escoge el *Pool Name* creado anteriormente como lo muestra la figura 11.



Figura 11. Creación recurso JDBC

Por último, en el menú Aplicaciones se selecciona la opción <desplegar> cargando el archivo “.war” donde se tiene almacenada la aplicación web como muestra la figura 12. Si todo sale bien se direccionará a la página donde se muestra la URL de la aplicación instalada. Al ingresar se desplegará la página principal de la aplicación como lo muestra la figura 13.

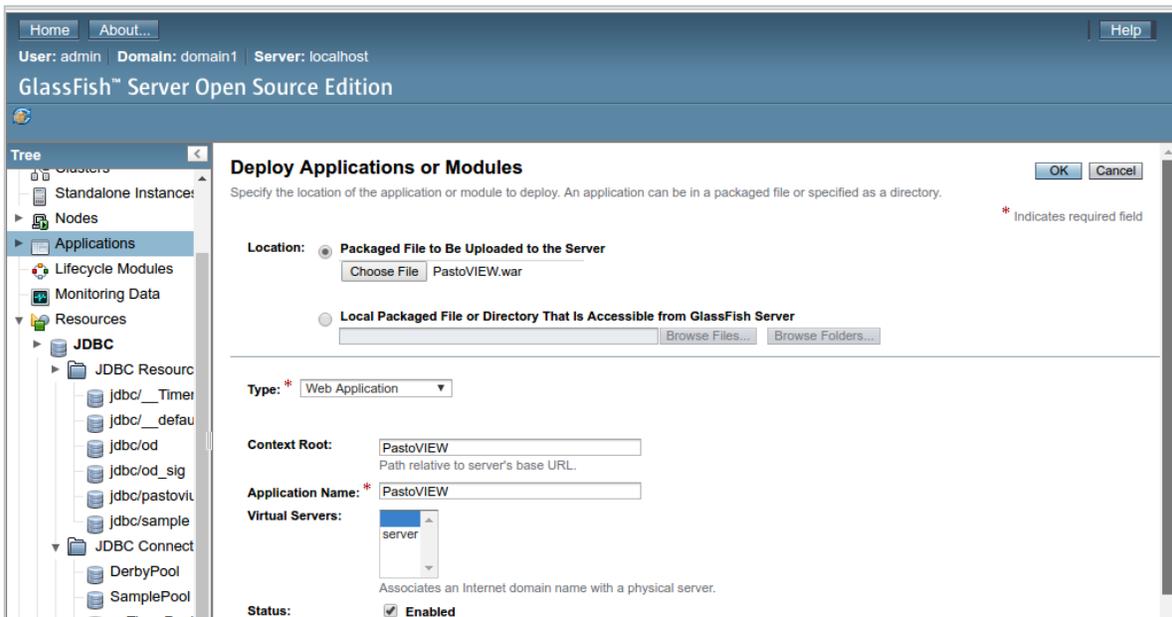


Figura 12. Carga del aplicativo web en el servidor glassfish

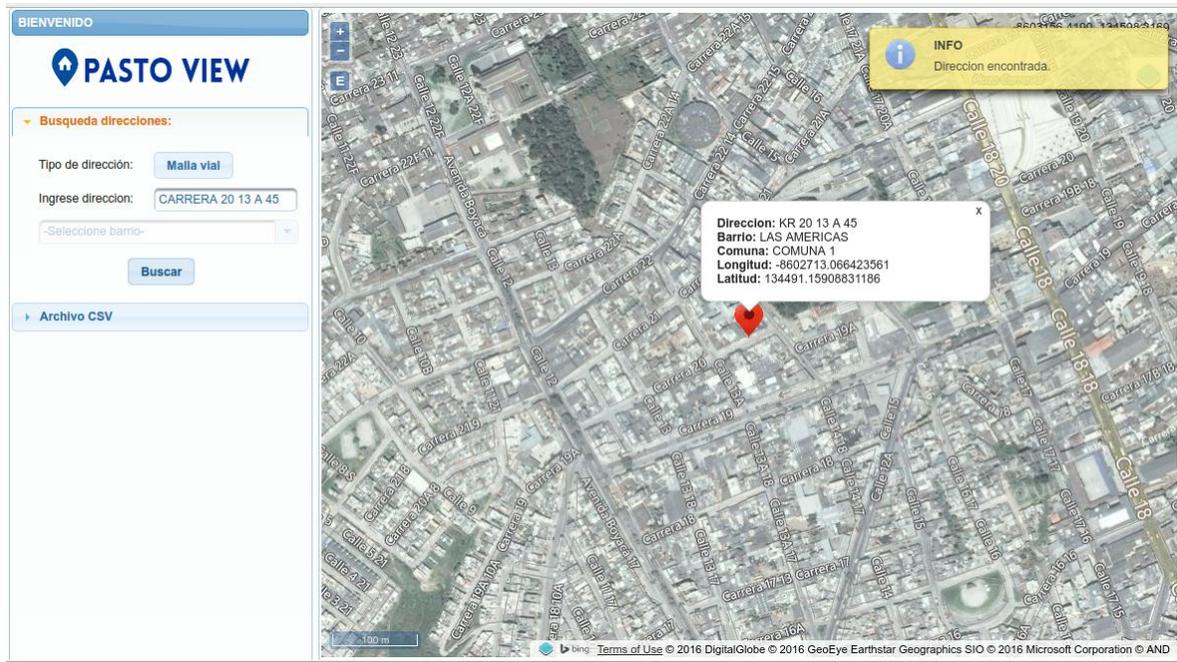


Figura 13. Visor PASTOVIEW en ejecución

4. MANEJO VISOR CARTOGRÁFICO PASTO VIEW

El visor cartográfico PASTO VIEW es un aplicativo web que permite al usuario acceder a la búsqueda de direcciones urbanas, procesamiento de archivos separados por comas CSV y la visualización de estos datos procesados, soportando la geocodificación de direcciones urbanas asignadas según la malla vial y según la nomenclatura barrio - manzana - predio.

El acceso a las funcionalidades de búsqueda de direcciones urbanas y procesamiento de archivos CSV se encuentran en el menú principal del aplicativo web.

4.1 Búsqueda direcciones.

A continuación se explicará la búsqueda de direcciones urbanas, opción que se muestra en la figura 14.



The screenshot displays the main menu of the PASTO VIEW application. At the top, there is a blue header with the text "BIENVENIDO". Below this is the application logo, which consists of a blue house icon followed by the text "PASTO VIEW". The main content area is divided into two sections. The first section is titled "Busqueda direcciones" and contains a form with the following elements: a "Tipo de dirección:" label with a "Malla vial" button; an "Ingrese direccion:" label with a text input field; a dropdown menu with the placeholder text "-Seleccione barrio-"; and a "Buscar" button. The second section is titled "Archivo CSV" and is currently collapsed, indicated by a right-pointing arrow.

Figura 14. Menú búsqueda de direcciones urbanas

En la primera opción denominada <Tipo de dirección> se selecciona el tipo de dirección que se desea consultar. Al hacer clic sobre este botón, este intercambiará su valor según las dos opciones disponibles: Malla vial y Manzanas.

Si la opción Manzanas es seleccionada, se habilitará la opción <Seleccionar barrio>, campo requerido por el geocodificador para realizar la búsqueda de este tipo de direcciones urbanas. Esta configuración puede verse en la figura 15.



The screenshot shows a web interface for 'PASTO VIEW'. At the top, there is a blue header with the text 'BIENVENIDO' and a logo consisting of a house icon and the text 'PASTO VIEW'. Below the header, there is a section titled 'Busqueda direcciones' with a dropdown arrow. Under this section, there are three input fields: 'Tipo de dirección:' with a button labeled 'Manzanas', 'Ingrese dirección:' with a text input containing 'Manzana K Casa 9', and a dropdown menu currently showing 'LA PAZ'. At the bottom of this section is a blue button labeled 'Buscar'.

Figura 15. Búsqueda de direcciones tipo Manzanas.

El campo <Ingrese dirección> se ingresa textualmente la dirección a buscar. Al ingresar la dirección que se desea consultar, se debe considerar los componentes mínimos de cada tipo de dirección.

Para el caso de malla vial debe ingresarse la vía principal, la vía generadora o secundaria y la distancia de la intersección de estas dos vías con la puerta de ingreso al predio buscado (ej. Carrera 20 13 A 45). Un ejemplo de esta búsqueda puede verse en la figura 16.

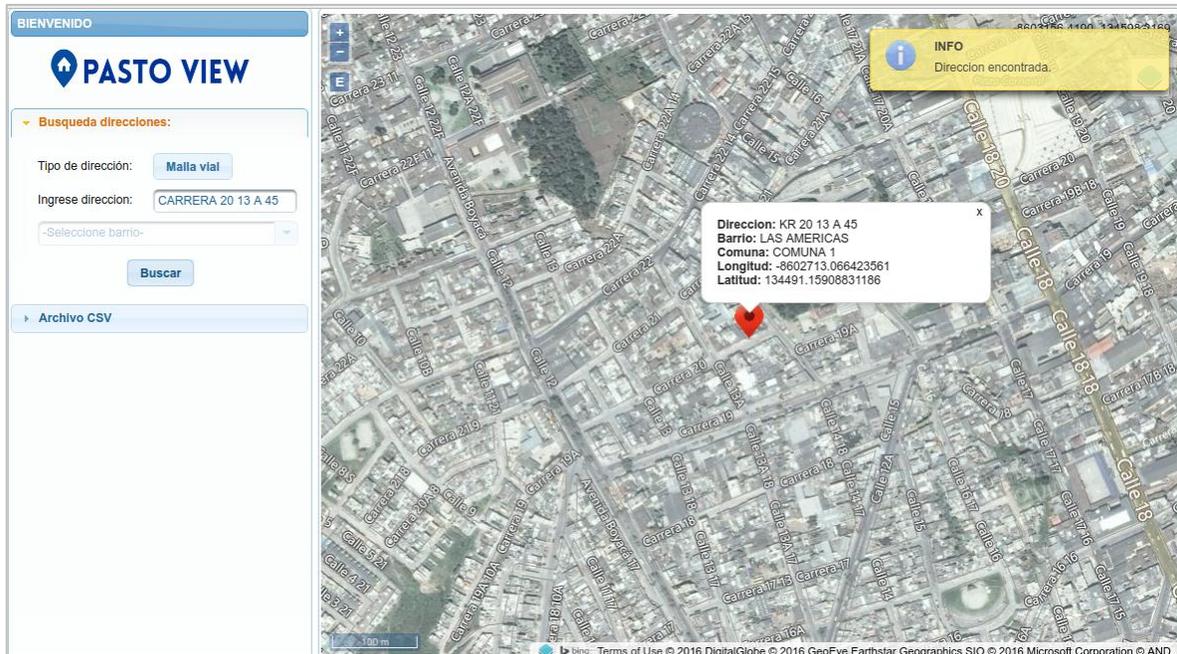


Figura 16. Búsqueda de dirección según la malla vial en PASTOVIEW

Si la opción seleccionada es manzanas, en el campo de texto solo debe ingresarse el identificador de manzana y el identificador de la casa o predio (ej. manzana k casa 9). Para este caso se habilita el campo Selecciona barrio, con el cual se especifica el barrio al que corresponde la nomenclatura de manzana ingresada.

Un ejemplo de esta búsqueda puede verse en la figura 17.

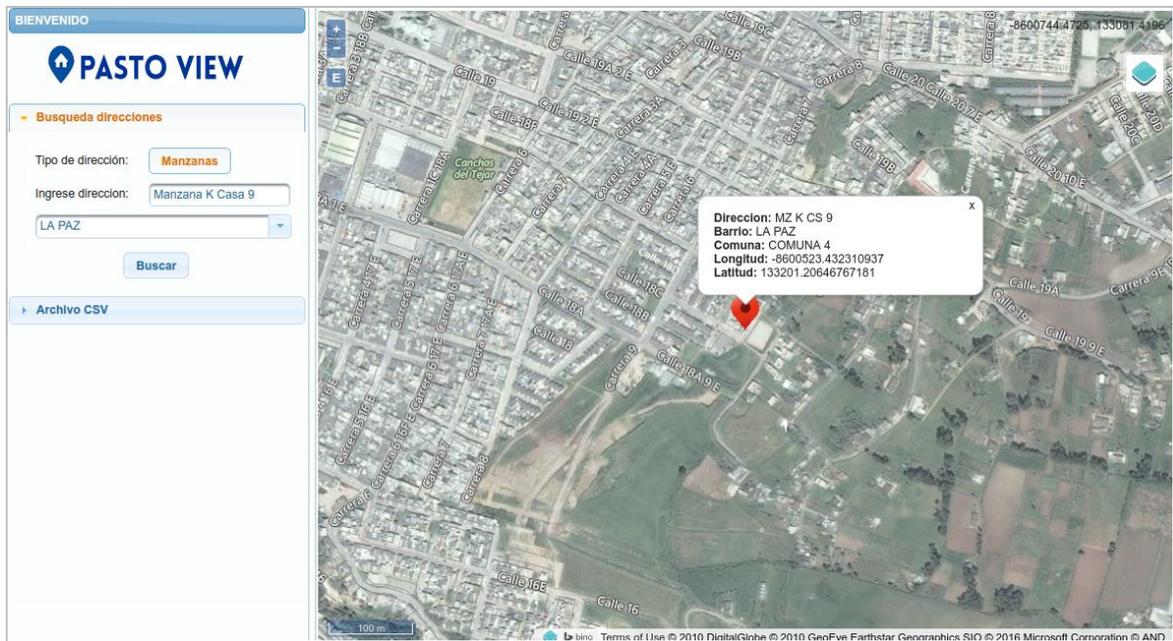


Figura 17. Búsqueda de dirección según la nomenclatura barrio - manzana predio en PASTOVIEW

Cualquiera sea el tipo de dirección seleccionado, si la dirección es encontrada en el repositorio de datos, se visualizará la ubicación urbana de la dirección mostrando información como la dirección estandarizada, el barrio y la comuna a la que pertenece y las coordenadas geográficas latitud/longitud encontradas.

4.2 Archivos CSV.

A continuación se explica la funcionalidad subida y procesamiento de archivos CSV. Su implementación puede observarse en la figura 18.



Figura 18. Funcionalidad de subida y procesamiento de datos CSV implementada en PASTO VIEW

El panel Archivos CSV se compone de 3 opciones principales:

- **Subir archivo CSV:** En esta opción se selecciona el archivo que será procesado por el visor cartográfico. Las condiciones que debe cumplir este archivo son:
 - ✓ Los datos almacenados en este archivo deben utilizar el carácter coma (,) como delimitador.
 - ✓ Debe tener un registro cabecera (header) que identifique los nombres de cada campo.
 - ✓ Este debe tener la extensión .CSV, de lo contrario se mostrara un mensaje de archivo inválido.

El archivo será subido al servidor de la aplicación correctamente si en la etiqueta Archivo subido se muestra el nombre del archivo seleccionado.. La figura 19 muestra un ejemplo de esta subida de archivos. De igual forma se cargará las listas de selección de los campos restantes.

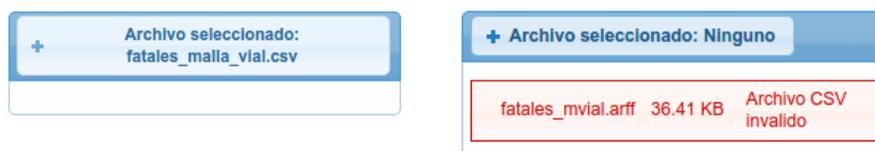
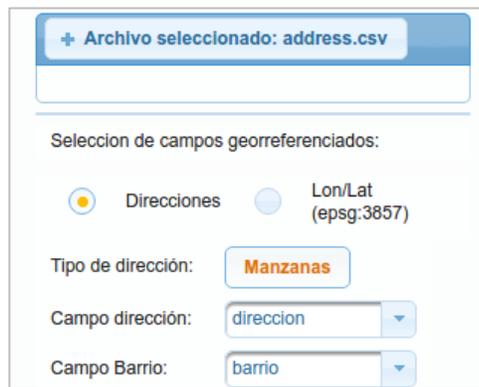


Figura 19. Subida correcta e incorrecta de archivos CSV.

- **Selección de campos georreferenciados:** Esta opción permite seleccionar los campos del archivo CSV donde se encuentran los datos georreferenciados.

Un primer tipo de campo es Direcciones, aquí se especifica los campos tipo texto donde estén almacenados direcciones urbanas, según sea el caso, Malla vial y nomenclatura barrio - manzana – predio (Manzanas), que estén configuradas dentro de los datos CSV. Un ejemplo de esta configuración puede observarse en la figura 20.



The image shows a configuration window for a CSV file named 'address.csv'. Under the heading 'Selección de campos georreferenciados:', there are two radio button options: 'Direcciones' (which is selected) and 'Lon/Lat (epsg:3857)'. Below this, there are three configuration fields: 'Tipo de dirección:' with a dropdown menu set to 'Manzanas', 'Campo dirección:' with a dropdown menu set to 'direccion', and 'Campo Barrio:' with a dropdown menu set to 'barrio'.

Figura 20. Configuración del archivo 'address.csv', el cual contiene direcciones asociadas al caso Manzanas.

Un segundo tipo de campo es Longitud/Latitud, aquí se especifica dos campos numéricos con decimales que corresponden a las coordenadas geográficas longitud/latitud asociada a ese registro. Estas coordenadas deben estar calculadas según el sistema de referencia espacial EPSG:3857. Un ejemplo de esta configuración puede observarse en la figura 21.

+ Archivo seleccionado: clusters-fatales.csv

Selección de campos georreferenciados:

Direcciones Lon/Lat (epsg:3857)

Campo Longitud:

Campo Latitud:

Figura 21. Configuración del archivo 'cluster-fatales.csv', seleccionando los campos 'longitud' y 'latitud' presentes en este archivo.

- Agrupar por:** Esta opción por defecto aparecerá desactivada por lo cual los datos procesados se visualizarán georreferenciados a nivel de direcciones urbanas. La figura 23 muestra el procesamiento del archivo 'clusters-fatales.csv', ubicando cada registro procesado en su correspondiente ubicación.

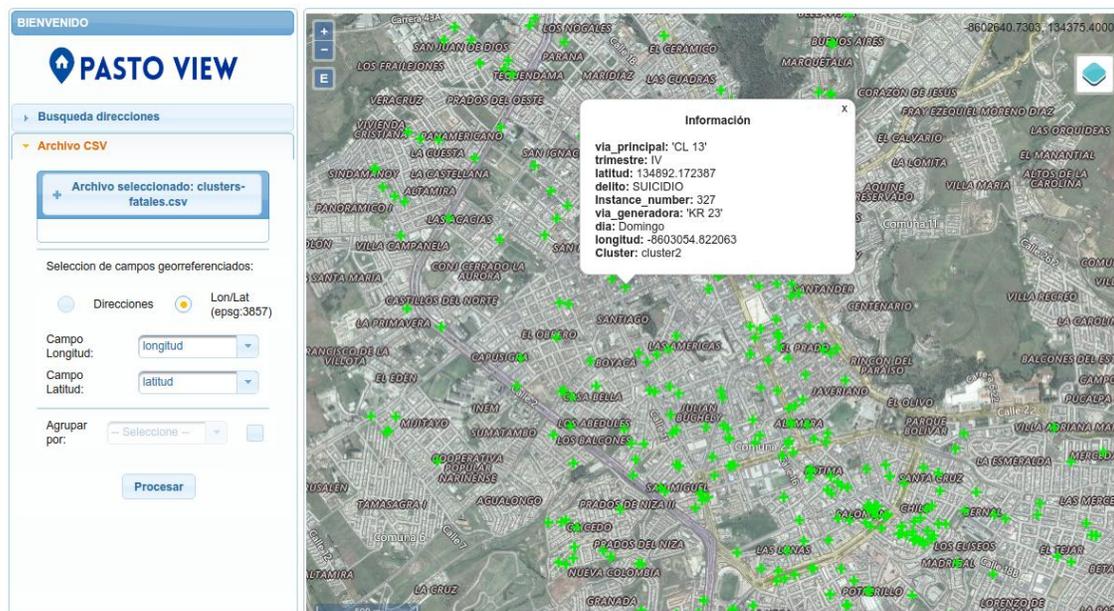


Figura 22. Procesamiento del archivo 'clusters-fatales.csv' sin agrupar.

Si esta opción se habilita, el visor cartográfico creará capas o mapas de visualización de acuerdo al campo seleccionado. Estas capas agruparán los datos según la característica

o campo seleccionado, permitiendo deshabilitar una o varias de estas capas. La figura 24 muestra un ejemplo de esta configuración, en donde se agrupan los datos por el campo 'Cluster' deshabilitando las capas que corresponden a los valores 'cluster0' y 'cluster1'.

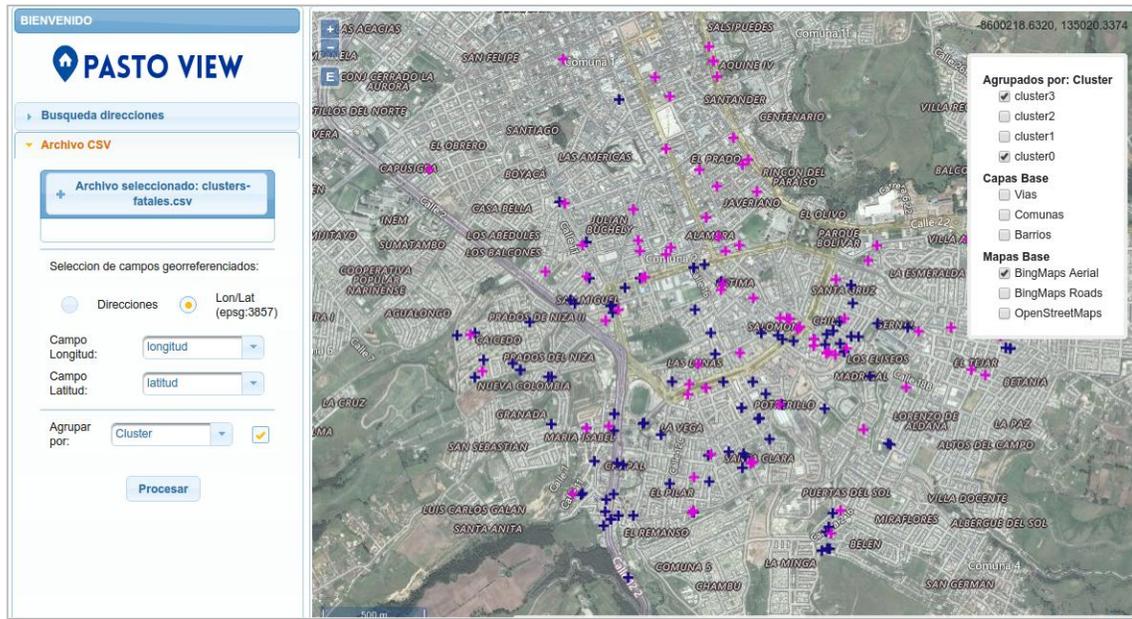


Figura 23. Procesamiento del archivo 'clusters-fatales.csv'. Los datos se agrupan según los valores del campo 'Clusters'.

El usuario podrá visualizar a los datos asociados a cada evento o registro procesado haciendo clic en cada punto. Al hacerlo se desplegará un dialogo con cada característica asociada a dicho punto o registro.

REFERENCIAS BIBLIOGRÁFICAS

- [1] O. Huisman y R. A. *Principles of Geographic Information Systems: An Introductory Textbook*. International Institute for Geo-Information Science and Earth Observation (ITC), 2009.
- [2] V. Olaya. *Sistemas de Información Geográfica Versión 1.0*. 2011.
- [3] Timarán, R. y Quemá, N. *GEOCODER PASTO: Geocodificador de Direcciones Urbanas del Municipio de Pasto*. GRIAS, Departamento de Sistemas, Universidad de Nariño. Pasto (Colombia). 2016.

ANEXO

INSTALACIÓN DEL GEOCODIFICADOR DE DIRECCIONES URBANAS PARA EL MUNICIPIO DE 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/>

1. Software requerido:

1.1. 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
```

1.2. 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
```

1.3. 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
```

2. 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. 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**

3.1. 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 registros 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
```

3.2. 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
```

4. 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 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"
```

5. 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