**Departamento Administrativo**

**Nacional De Estadística**



**Dirección de Geoestadística**

**Módulo TANGRAM**

**Generador de mapas en 2D y 3D**

**Por Teselas Vectoriales**

**Guía de Usuario**

**Diciembre 2015**

**Módulo Tangram**

1. Definición

Proyecto para visualizar mapas, basado en Teselas vectoriales. Permite implementar diseños en tiempo real y proporcionan interactividad al usuario final.

1. Componentes

* WebGL🡪 API para procesar mapas tipo 2D y 3D.
* GeoJSON🡪 Lectura de información en formato Geographic JSON
* TopoJSON🡪 Lectura de información en formato Topographic JSON
* Binary Vector Data🡪OJO
* Leaflet🡪 Funcionalidad básica para zoom y desplazamiento sobre la región en el mapa.
* YAML 🡪Estructura de componentes que se cargan en el mapa de la vista. En el API se tiene el archivo principal denominado scene.yaml
* Main Library 🡪 tangram.debug.js

1. Instalación OJO

Para instalar el Core Tangram, debe seguir los siguientes pasos:

Paso 1. Descomprimir paquete

Para descomprimir el paquete, debe usar los siguientes descompresores, de acuerdo al Sistema Operativo que se muestran en la tabla 1.

**Tabla 1. Listado de compresores / descompresores por Sistema Operativo**

|  |  |
| --- | --- |
| **Sistema Operativo** | **Compresor / Descompresor** |
| Windows – XP, 7, 8, 10 | WinZip, WinRar, 7-zip, PowerArchiver |
| Linux – RedHot, Ubuntu |  |

Paso 2. Crear folder en el servidor Web

Paso 3. Verificación de permisos

1. Coordenadas URL

Se procesan por medio de la tupla (<latitud, <longitud>, <nivel\_zoom>) definido en la URL: http://<localhost>/index.html#mapzen, <latitud>,<longitud>,<zoom>.

Donde

Mapzen: Define el servicio de despliegue de los mapas implementados con Tangram.

1. Propiedades

* Lights

Establece como se iluminan los objetos que se cargan en el mapa. Se establecen parámetros como dirección, tipo de iluminación, modalidad de difusión los cuales permiten aplicar aspectos de iluminación sobre los mismos.

* Cameras

Establece como se visualizan los objetos del mapa. Es decir, establece la forma de los objetos que se cargan en el mapa. Existen 3 tipos de vistas: perspectiva (perspective), isométrica (Isometric), y plana (flat). Cada uno de estos, permite personalizar distintos aspectos como tipo (type), ejes (axis), longitud focal (focal length) y punto base (vanishing point), así como si se encuentra activa o no la vista.

* Scene

Determina el fondo del escenario donde se carga el mapa. Para el demo, se define el color del fondo.

* Textures

Define la visualización de sitios como hospitales, árboles, hoteles basado en una serie de iconos que se encuentran encapsulados (sprites) en el archivo images/poi\_icons\_32.png. Determina la manera de filtrar los elementos en el mapa. En ésta directiva, permite definir la animación de los iconos (sprites) con base a un arreglo (array) que representa los siguientes elementos: [Origen coordenada x, Origen coordenada y, longitud, alto].

* Styles

Representa el estilo de visualización de elementos del mapa.

En la tabla 2 se describen cada uno de los estilos de visualización, junto con la imagen que lo representa.

**Tabla 2. Estilos de visualización – Efectos al mapa**

| **ESTILO** | **DESCRIPCIÓN** | **IMAGEN RELACIONADA** | | | |
| --- | --- | --- | --- | --- | --- |
| Water | Estilo donde los se visualizan de manera dinámica los cuerpos de agua en el mapa. |  | | | |
| Elevator | Estilo donde se visualizan los bloques de edificios en movimiento ascendente – descendente |  | | | |
| Popup | Vista de perspectiva de los bloques de edificios al desplazarse a lo largo del mapa (estáticos) |  | | | |
| Halftone | Vista plana de puntos en color negro |  | | | |
| Windows | Vista de forma fotográfica formato negativo del mapa |  | | | |
| Environment | Vista del mapa en formato negativo del mapa. Se habilitan 4 vistas {‘Sunset’, ‘Chrome’, ‘Matte Red’, ‘Color Wheel’}, donde aplica para los objetos del mapa. | Sunset 🡪 Objetos en color azul como tono de salida del sol | |  | |
| Chrome 🡪 Objetos en color blanco | |  | |
| Matte Red 🡪 Objetos en color rojo oscuro | |  | |
| Color Wheel 🡪 Objetos en distintos colores, según la ubicación de éstos sobre el mapa. | |  | |
| Rainbow | Vista dinámica de los elementos del mapa con colores que cambian de manera sucesiva. |  |  | |  |

FUENTE: DANE. Aplicación prueba

* Layers

Permite activar o desactivar las capas que se encuentran en el mapa. Los elementos de las capas se describen por medio de la Tabla 3.

**Tabla 3. Elementos de las capas del mapa**

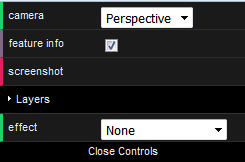
| **CAPA** | **DESCRIPCIÓN** |
| --- | --- |
| Earth | Capa secundaria del mapa, que determina la capa base sobre el mapa. |
| Landuse | Parques y espacios al público. Va en llave con la capa Earth |
| Water | Visualización de los cuerpos de agua sobre el mapa |
| Roads | Visualización de calles principales, y secundarias sobre el mapa |
| Buildings | Visualización de construcciones: Edificios principalmente |
| Road label | Títulos correspondientes a las calles principales, secundarias |
| Poi\_icons | Iconos que describen hospitales, parques, cadenas de hipermercado |
| Places | Visualización de los labels correspondientes a los lugares del mapa (países,departamentos, municipios) |
| Point\_labels | Títulos correspondientes a la capa Poi\_icons (Point Icons) |

FUENTE: DANE. Aplicación prueba

1. Panel de control del mapa

Permite controlar la visualización de elementos descritos en la Tabla 3 sobre el mapa. El panel de control, aparece en la parte superior derecha del mapa, según se visualiza en la figura 1.

**Figura 1. Panel del control del mapa**



FUENTE: DANE. Aplicación prueba

A continuación, se describen cada uno de los elementos del Panel del control, según se muestra en la Tabla 4.

**Tabla 4. Elementos del panel de control**

| **ELEMENTO (ITEM)** | **OPCIONES** | **DESCRIPCIÓN** | **OBSERVACIONES** |
| --- | --- | --- | --- |
| Camera | * Flat * Perspective * Isometric | Tipos de vistas del mapa (Plano, perspectiva e isométrica) |  |
| Feature info | * Check * No Check | Activa (o desactiva) la información del Ítem |  |
| Screenshot | N/A | Permite capturar en un archivo gráfico, la imagen que corresponde al mapa visualizado en pantalla | Genera un archivo exportable tipo imagen |
| Layers | * Check * No Check | Activa (o desactiva) la capa | Ver Tabla 3 |
| Effect | * Water animation * Elevator * Pop-up * Halftone * Windows * Environment Map * Rainbow | Efectos de visualización sobre el mapa base. | Ver Tabla 4 |

1. Navegadores compatibles

* Opera: Todas las versiones.
* Chrome: Todas las versiones.
* Firefox: No funciona en versiones 29 e inferiores. La Versión 42 b tampoco funciona la librería.

Webgrafía

1. Mapzen Tangram. Disponible: <https://mapzen.com/projects/vector-tiles/>
2. GitHub. Real-Time WebGL Maps. Disponible: <https://github.com/tangrams/tangram>
3. Mapzen. Tangram Render 2D and 3D maps with fine control over almost every aspect of the map-making process. Disponible: <https://mapzen.com/documentation/tangram/>

**Anexo 1. Uso de la propiedad iluminación**

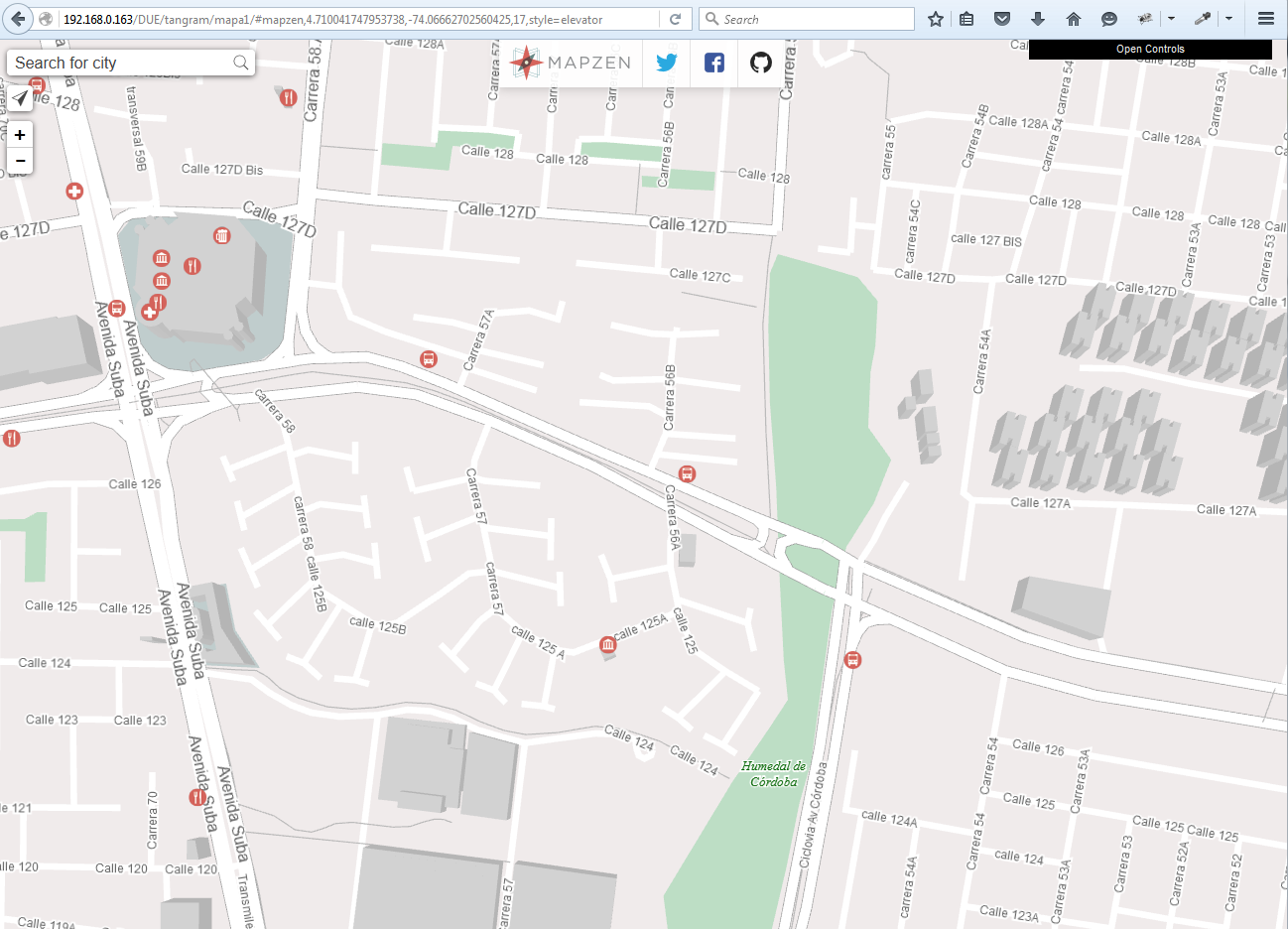
El prototipo maneja un archivo denominado scene.yaml. Éste archivo controla los objetos que se definen para visualizar los objetos en el mapa. Se realiza una prueba de iluminación, definida mediante la Tabla 5.

**Tabla 5. Parámetros de iluminación**

| **PARAMETRO** | **VALOR** |
| --- | --- |
| Type | Direccional |
| Direction | [.1, .5, -1] |
| Diffuse | .2 |
| Ambient | 1.1 |

El resultado de los parámetros definidos, me visualiza el mapa mostrado en la figura 2.

**Figura 2. Iluminación predeterminada**



FUENTE: DANE. Aplicación prueba

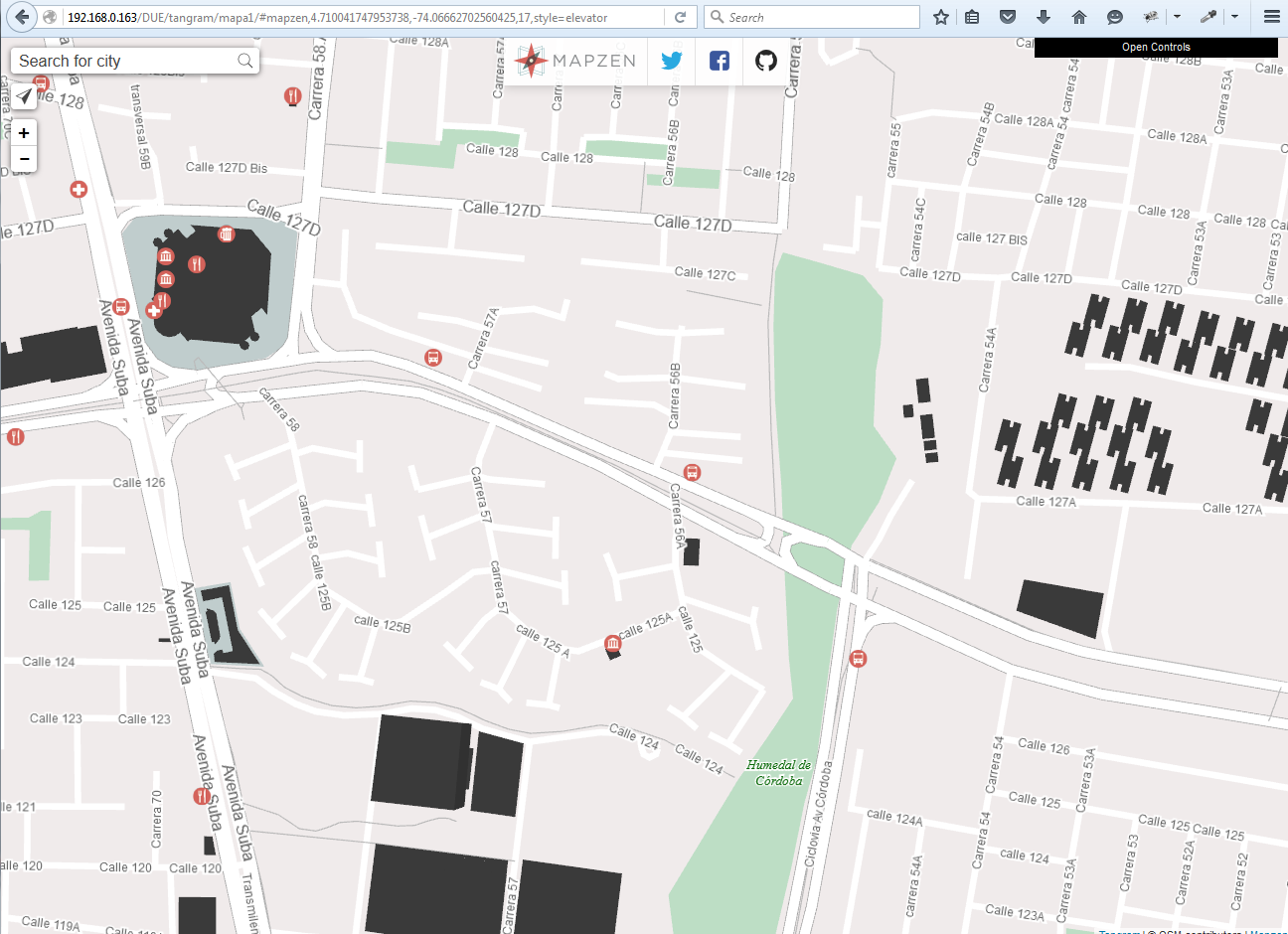
Para el ejemplo, se actualizó la iluminación según se muestra en tabla 5a.

**Tabla 5a. Nuevos valores de iluminación**

|  |  |
| --- | --- |
| **PARAMETRO** | **VALOR** |
| visible | true |
| type | point |
| position | [-74.0170, 40.7031, 100] |
| origin | world |
| ambient | 0.3 |
| diffuse | 1 |
| specular | .2 |

Al aplicar éstos valores de la configuración, se obtiene el mapa visualizado en figura 3.

**Figura 3. Iluminación del nuevo mapa**



FUENTE: DANE. Aplicación prueba

Como se visualiza en la figura 3, la iluminación se aplica sobre los bloques sólidos del mapa.**Anexo 2. Adición de capas**

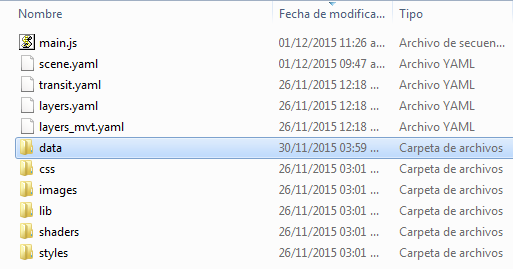
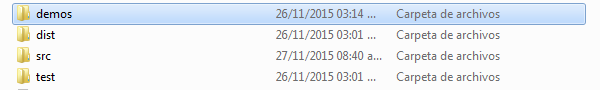
En el archivo scene.yaml, ubicado sobre el folder demos se adicionan las capas personalizadas que se visualizarán en el mapa.

Para lograr visualizar la información sobre el mapa, se deben seguir los siguientes pasos:

Paso 1. Colocar archivo fuente (source)

Descargar el archivo xxxxxx.json o yyyyy.geojson y colocar en folder según figura 4.

**Figura 4. Ubicación de los archivos GeoJSON / TopoJSON**



NOTA: El archivo debe tener el formato GeoJSON (.json) o TopoJSON (.tjson)

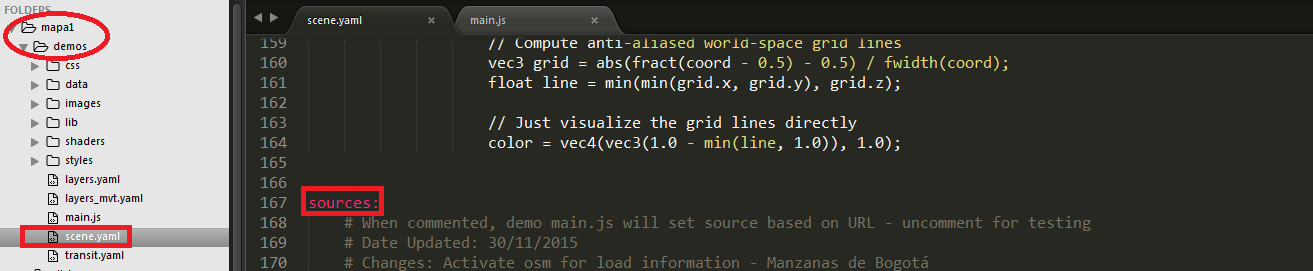
Paso 2. Incluir la data en el mapa

Buscar el fuente scene.yaml, sobre el folder /demos y realizar el siguiente procedimiento.

* Buscar la etiqueta “sources”

Activar la etiqueta especificada según se muestra en figura 5.

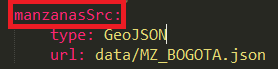
**Figura 5. Activación de la etiqueta “sources”**



* Crear la etiqueta de origen del archivo

Dentro de la sección “sources”, se crea la etiqueta que representa el origen del archivo (manzanasSrc por ejemplo), y se especifica el archivo que contiene la data en formato: <folder>/<archivo>.json. Por ejemplo, con el nombre “manzanasSrc”. Se debe especificar el tipo (type): GeoJSON / TopoJSON, lo mismo que el origen del archivo en la cláusula url: según se visualiza en figura 6.

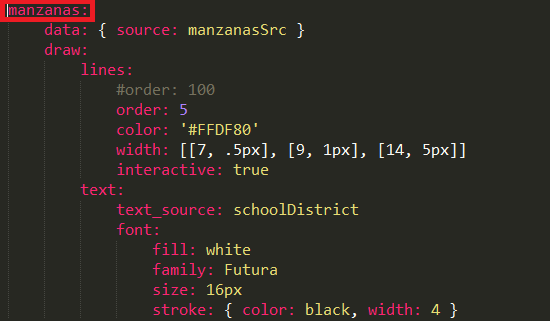
**Figura 6. Creación del origen del archivo para la capa**



* Crear la capa con el nombre

Dentro de la sección “layers”, se crea la etiqueta que representa el nombre de la capa que se monta sobre el mapa. Por ejemplo, con el nombre manzanas, según se visualiza en figura 7.

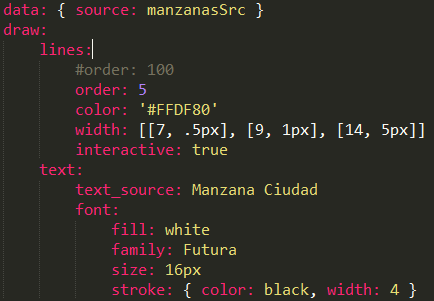
**Figura 7. Creación de la capa**



* Configuración de la capa

Según figura 7, se debe configurar el origen que toma la información la capa, los parámetros que se dibujan sobre el mapa (etiqueta draw), el tipo de gráfico que se va a dibujar (dentro de la etiqueta draw especificar lines, polygons), el orden en que se carga la capa, su color, su tamaño sobre el mapa. Éstos parámetros se visualizan sobre la figura 7a.

**Figura 7a. Parámetros de configuración de la capa**



Los parámetros requeridos son:

* data: Especifica el origen del archivo de información que toma la capa. Allí se debe usar la palabra clave “source”, y colocar la etiqueta de origen de archivo, definido según figura 7b.
* draw: Especifica Orden de visualización, color, tamaño del gráfico que visualiza según el archivo fuente, definido según figura 7b.
  + lines | polygons: Tipo de gráfico según el archivo fuente.

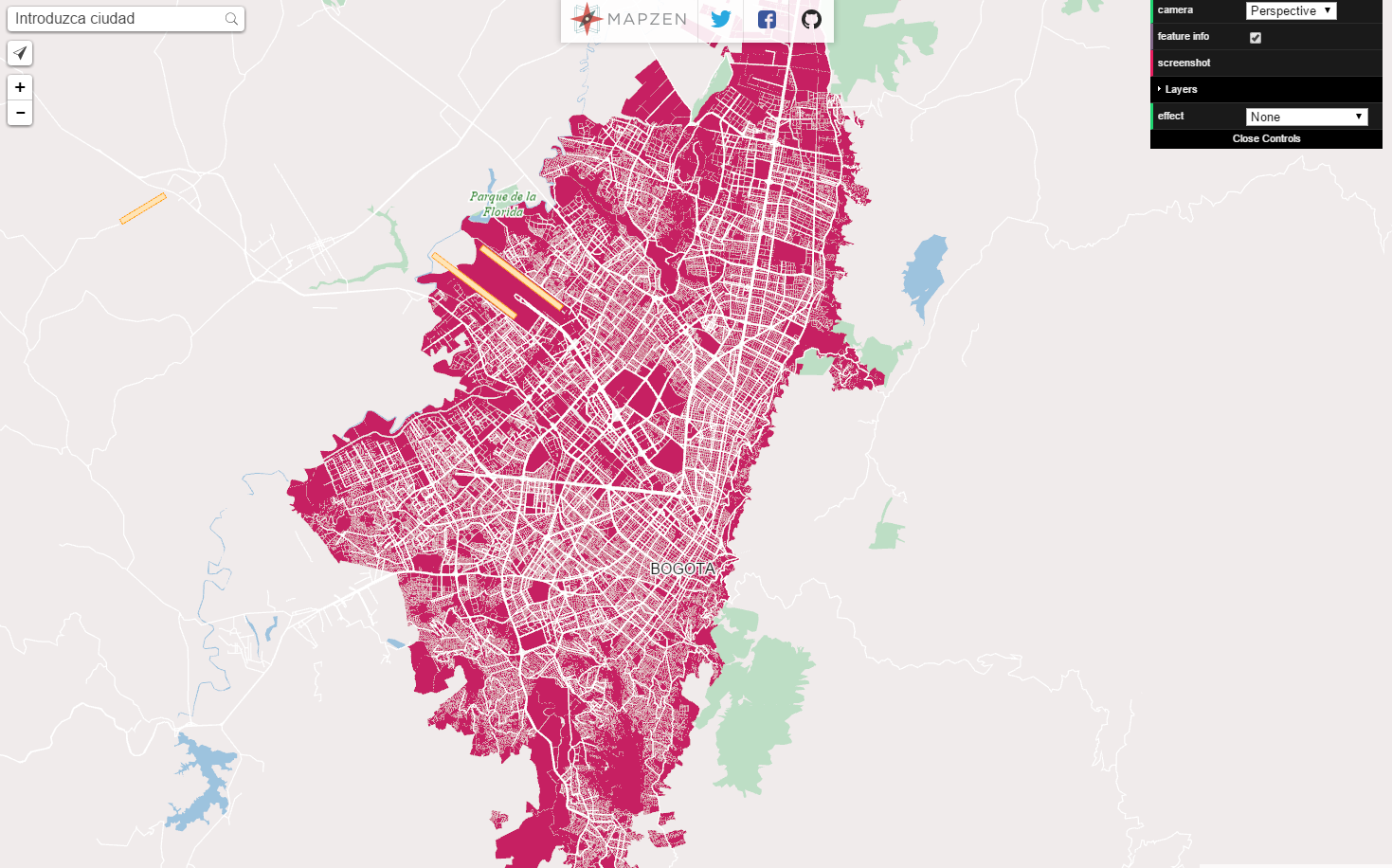
Dentro del tipo de gráfico, se definen las propiedades del mismo, estás son:

* Order 🡪 orden de visualización, por ejemplo, 5
* Color 🡪 Define color del gráfico, por ejemplo color Durazno ‘FFDF80’ (Código Hexadecimal).
* Width 🡪 Define largo del gráfico en notación px.
  + Text: Parámetros para visualizar label asociado al área según el archivo fuente.

Anexo 3. Visualización del prototipo

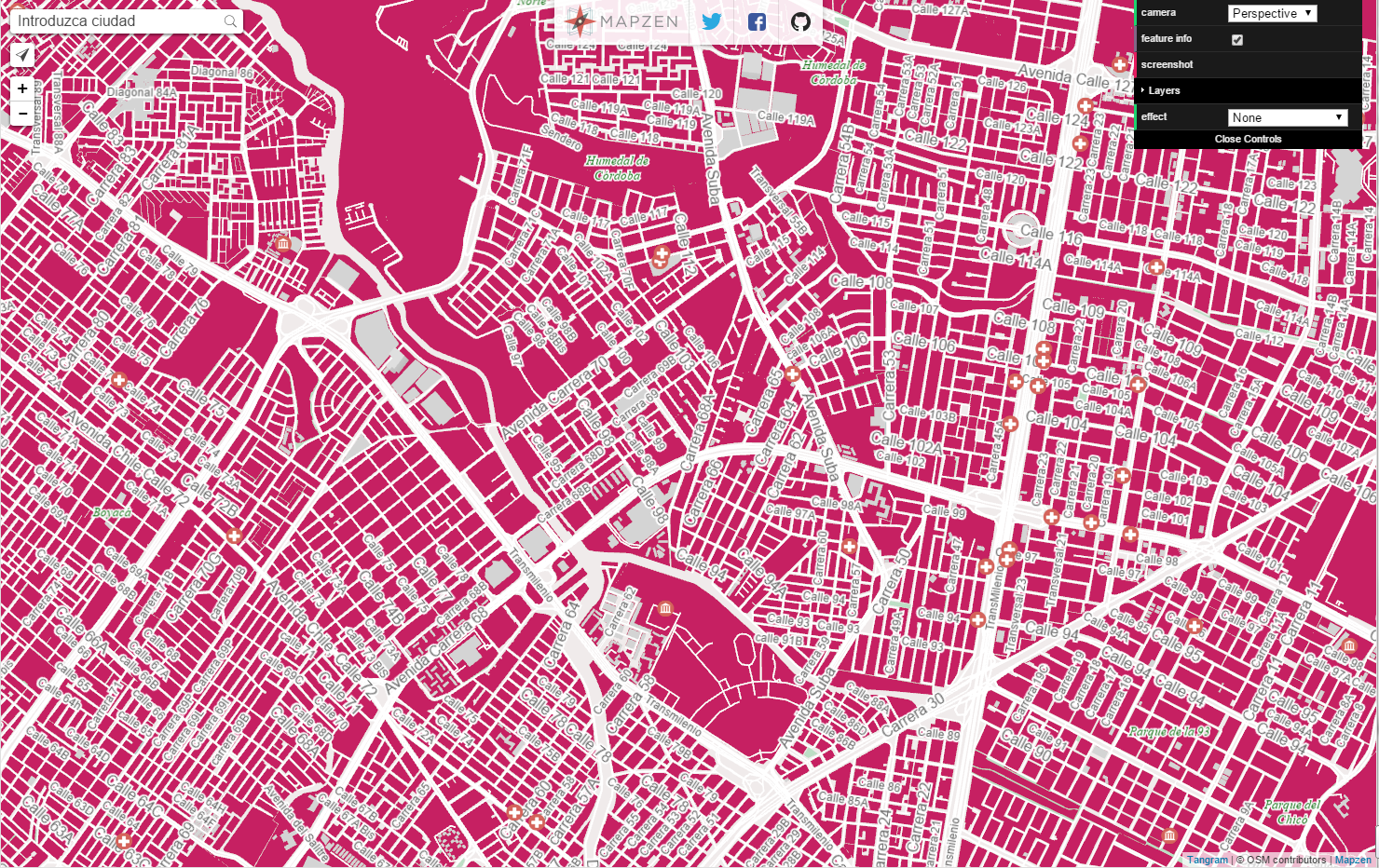
En figura 8, se visualiza el mapa con un archivo tipo GeoJSON que representa las manzanas de la ciudad de Bogotá. En éste, se muestran las 3 capas: plana, Tangram y manzanas.

**Figura 8. Visualización del prototipo Tangram Map – Zoom: 12**



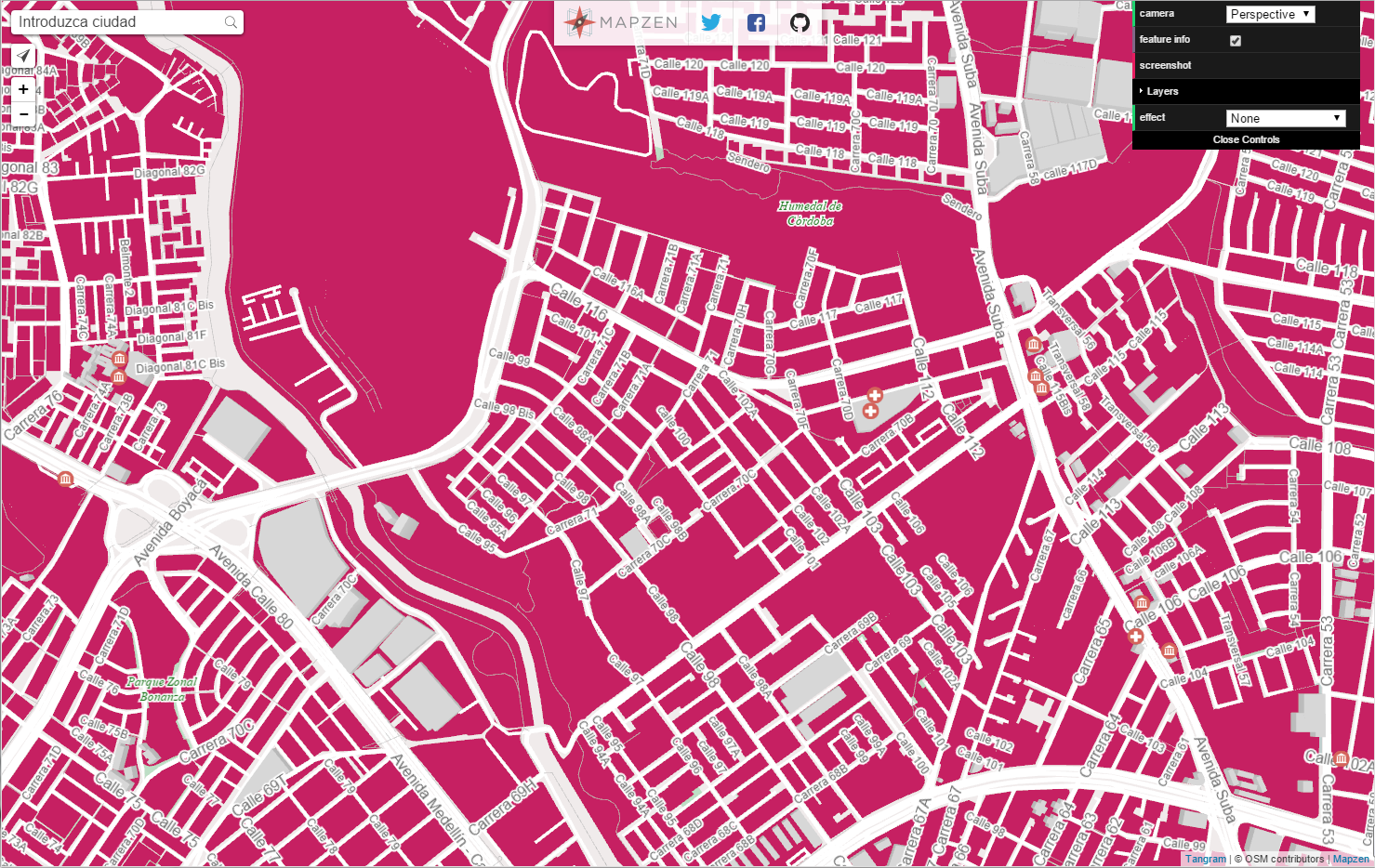
FUENTE: DANE. Aplicación prueba

**Figura 8a. Escala de manzanas ampliadas con zoom 15**



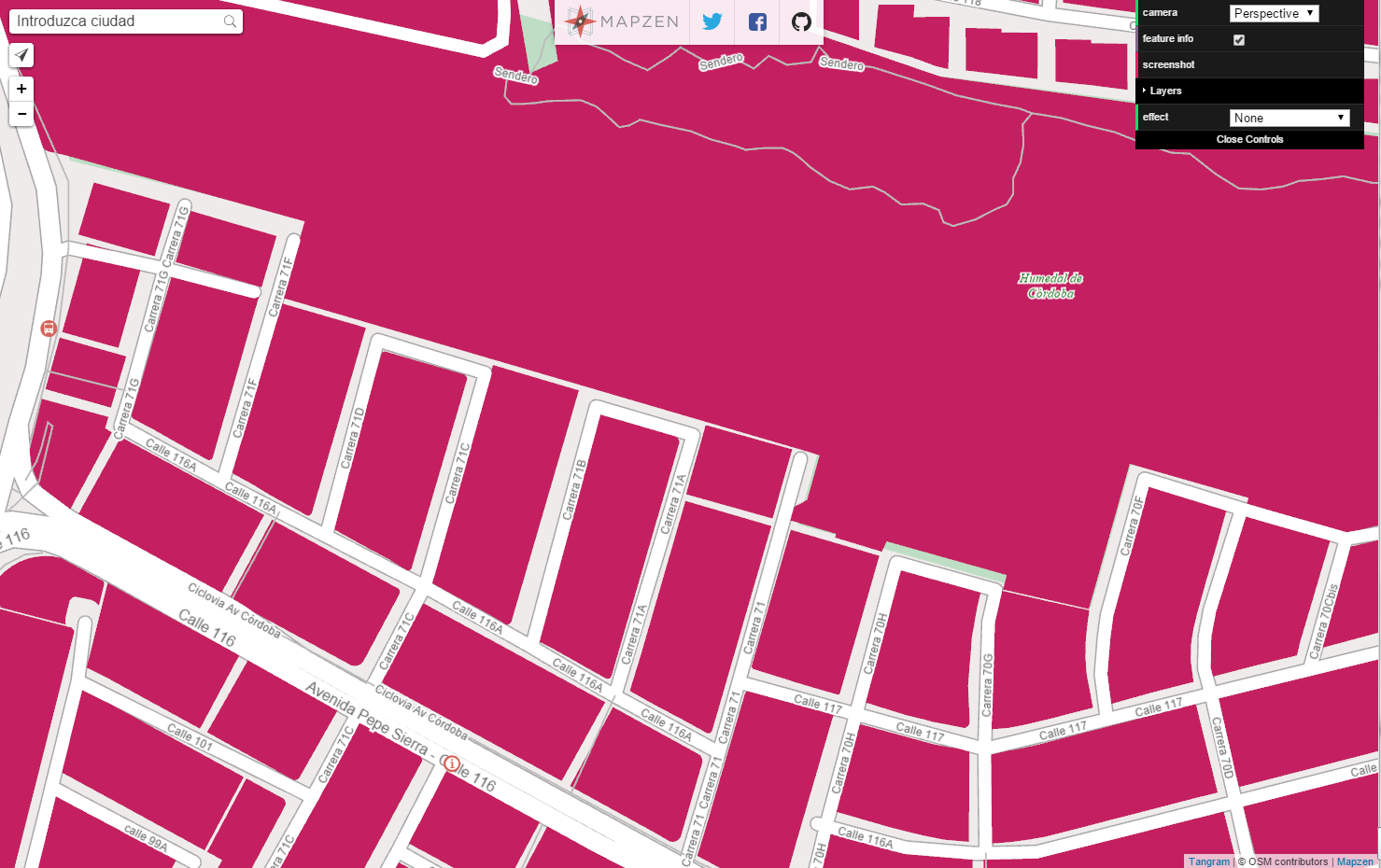
FUENTE: DANE. Aplicación prueba

**Figura 8b. Escala de manzanas ampliadas con zoom 16**



FUENTE: DANE. Aplicación prueba

**Figura 8c. Escala de manzanas ampliadas con zoom 18**



FUENTE: DANE. Aplicación prueba