Trabajando en la nube con CartoDB

Silvia González López[1], Anna Muñoz Bollas[2]

[1]sgonzalezlopez012@uoc.edu
[2]amunozbo@uoc.edu

Herramientas como CartoDB permiten realizar análisis SIG de manera sencilla y desde cualquier ordenador, no se necesita disponer de un software especial, tan solo una conexión a internet y un navegador web. Funciona además como una base de datos espacial (postGIS) cuyos datos se almacenan en la nube, lo que permite realizar consultas a esas tablas desde cualquier ordenador. En este proyecto se ha creado una aplicación web con un mapa interactivo que muestra la evolución de la renta familiar en los barrios de Barcelona desde 2008 a 2011. La distribución espacial de estos datos demuestra que hay una zonificación muy clara entre los barrios más ricos y los barrios más pobres, y la evolución de las rentas en el periodo indicado muestran que las diferencias económicas entre unos y otros cada vez son mayores.

***Keywords*— PostGIS, CartoDB, Visualización de datos, Open Data, Índice RFD, Renta Familiar, Barcelona**

I. INTRODUCCIÓN

En los últimos años el mundo de los sistemas de información geográfica se está enfocando principalmente a crear desarrollos en entornos web, y sobre todo a tener los servicios en la nube, el llamado *cloud GIS*. CartoDB (http://cartodb.com/) puede usarse tanto como un servicio de software (SaaS - *Software As A Service*) creando *web maps* que son utilizados por otro usuario final, o como una plataforma (PaaS - *Platform As A Service*), en el caso de que se utilice como una base de datos espacial en la nube, o como fuente de datos para crear teselas o *tiles* (los datos no se representan mediante vectores sino como una imagen) que añadir a un mapa determinado.

CartoDB se puede definir como una base de datos geoespacial desarrollada con PostGIS que funciona en la nube. Es Cloud GIS, ya que permite realizar la manipulación de los datos y la visualización de estos desde la web, tanto mediante una interfaz de usuario o GUI (en el *dashboard* de su web, ver figura 1), o mediante programación (básicamente SQL, CSS, *JavaScript* y *JQuery*), lo que facilita su uso tanto a usuarios inexpertos como avanzados.

La gran ventaja del Cloud GIS es que no se necesita instalar ningún software en el ordenador, todas las operaciones y análisis de los datos se llevan a cabo en los servidores que CartoDB tiene en Amazon y que cuentan con los mejores ordenadores disponibles actualmente. Su plataforma nos permite utilizar sus servidores para almacenar nuestras tablas y donde, además, ya tienen instalado todo el software y librerías necesarias para que el usuario realice sus análisis o cree sus mapas en la web.

CartoDB es además *Open Source*, todo su código fuente se encuentra disponible desde su cuenta de GitHub (http://github.com/cartodb/cartodb20/).

4 julio 2013

CartoDB ofrece varios planes de contratación, pero disponen de un plan gratuito (límite de 5MB y 5 tablas) que para el desarrollo de este proyecto ha sido más que suficiente.

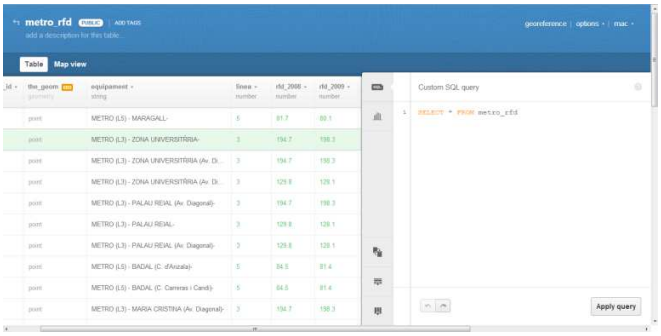

Fig. 1. Desde el *dashboard* de CartoDB se pueden visualizar los datos de la tabla sobre la que estemos trabajando y realizar consultas SQL u otro tipo de operaciones como *joins* entre tablas.

El objetivo de este trabajo es utilizar CartoDB para realizar un análisis de la evolución de la distribución de la renta en los barrios de Barcelona durante el periodo comprendido entre 2008 y 2011. Se ha realizado además un análisis de la relación espacial entre la distribución de la renta y la distribución de las líneas de metro de la ciudad. Para la representación de los datos se ha tomado como referencia el Mapa de la pobreza en Chicago como un mapa de transporte (*Poverty in Chicago as a Transit Map* [1]).

Este artículo se divide en tres partes diferenciadas: obtención de los datos, visualización y análisis. En primer lugar se expone como se han obtenido los datos usados para el análisis (todos son open data), y las modificaciones realizadas, tanto antes de subir los datos a CartoDB, como las realizadas mediante las herramientas de que dispone CartoDB desde su *dashboard*. A continuación se detalla cómo se ha realizado la visualización en la aplicación web, y las funcionalidades incluidas, así como las tecnologías usadas para su desarrollo. Finalmente se realiza el análisis de los datos obtenidos en el

web map, y se exponen las conclusiones a las que se ha llegado con este proyecto.

II. OBTENCIÓN DE LOS DATOS

Todos los datos usados en este proyecto son abiertos (open data) y están disponibles para su descarga en las direcciones indicadas a continuación:

1) Líneas de metro:

Fuente: http://download.geofabrik.de/europe/spain.html

Desde esta web se puede descargar toda la cartografía *OpenStreetMap* de España en formato *shape*. En concreto, en este proyecto se utilizó un archivo shape llamado '*railways*' y mediante ArcGIS se realizó la selección de las líneas de metro de Barcelona que se subieron a CartoDB.

Para obtener un campo que contenga sólo el número de línea se realizó la siguiente consulta SQL que elimina la L de los registros de la columna 'name':

UPDATE líneas_metro SET línea= substring (name from 2 for 2)

2) Estaciones de metro de Barcelona:

Fuente: web open data del ayuntamiento de Barcelona:
http://opendata.bcn.cat/opendata/cataleg/TRANSPORT/transports/

En esta web se descarga un archivo .csv que dispone de dos columnas llamadas coord_x y coord_y, que están en el sistema de referencia ED50 UTM31N. Una vez subida la tabla hay que rellenar la columna 'the_geom' que es donde se guarda la información geométrica, y para ello hay que modificar los datos y pasar del sistema UTM31ED50 (srid: 25831) a WGS84 (srid: 4326) que es el que utiliza por defecto CartoDB. La sentencia SQL utilizada es:

UPDATE transports SET the_geom = ST_Transform(ST_SetSRID(ST_Point(coord_x, coord_y),25831),4326)

Para seleccionar sólo los elementos que corresponden a las estaciones de metro se realiza:

SELECT the_geom, equipament
FROM transports
WHERE equipament LIKE 'MET%'

La tabla con las estaciones incluye todas las salidas que tiene una misma estación de metro, se podían haber dejado en solo una por estación pero finalmente se ha decidido dejarlas todas ya que se considera que aportan información, sobre todo las situadas en la frontera entre dos barrios con índices RFD muy diferentes.

3) Renta Familiar por Barrios (index RFD):

Fuente Renta familiar (2008, 2009,2010):
http://opendata.bcn.cat/opendata/cataleg/SOCIETAT_I_BENESTAR/rendafam-a/
Renta familiar (2011):
http://www.bcn.cat/estadistica/catala/dades/economia/renda/rdfamiliar/index.htm

Una vez obtenida una tabla excel con los datos correspondientes al periodo 2008-2011 se sube a CartoDB y se crea una columna llamada 'cbarri' que contendrá el código identificativo de cada barrio.

4) Porcentaje de población con estudios superiores por barrios, para el año 2011:

Fuente: Departamento de estadística del ayuntamiento
http://www.bcn.cat/estadistica/castella/dades/index.htm

Este es un dato que inicialmente no se contemplaba incluir en este estudio, pero que se ha incluido ya que está claramente relacionado con el nivel de renta.

5) Límites administrativos de los barrios de Barcelona:

Fuente: GeoPortal del ayuntamiento
http://w24.bcn.cat/geoportal/descargas/DIVADM_SHP.ZIP

El archivo descargado está en formato shape y en el sistema ED50, de nuevo se ha utilizado ArcGIS para transformar la proyección a WGS84 antes de subir la tabla a CartoDB. Esta tabla con los barrios ya tiene una columna llamada 'cbarri' con el código de cada barrio y es la que utilizaremos posteriormente para unirla con la tabla de la renta.

Una vez en el *dashboard* de CartoDB se crea una nueva tabla a partir de 'barris' y 'rendafam_2008_11 con un merge (column joint) por el campo común entre ambas: el código del barrio ('cbarri'). Y se seleccionan los campos que queremos que aparezcan en la tabla. (Figura 2)

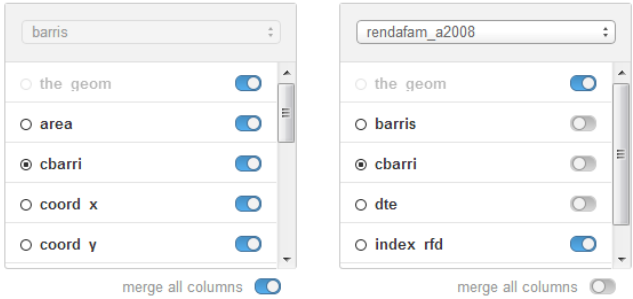

Fig. 2. Una de las herramientas más útiles del *dashboard* de CartoDB es "*merge*" que permite hacer un *join* entre dos tablas, ya sea por un campo común, como es este caso (donde la unión se hace a partir de la columna cbarri) o un *join* espacial.

Para poder unir la tabla con las estaciones de metro y la tabla con la renta se utiliza la siguiente consulta SQL, que selecciona los campos que nos interesan sólo donde intersectan ambas geometrías:

```
SELECT metro.*,
        renta_barris_08_11.rfd_2008,
        renta_barris_08_11.rfd_2009,
        renta_barris_08_11.rfd_2010,
        renta_barris_08_11.rfd_2011
FROM metro, renta_barris_08_11
WHERE
        ST_Intersects(metro.the_geom,
        renta_barris_08_11.the_geom)
```

Una vez hemos terminado de preparar los datos nos quedan tres tablas:

- **renta_barris_08_11_e,** que contiene la geometría de los barrios, el índice RFD y el porcentaje de estudios superiores para 2011.
- **Metro_rfd,** que contiene las estaciones de metro en formato punto, con su índice RFD asociado
- **Líneas_metro,** que contiene las geometría de las líneas de metro de Barcelona

III. VISUALIZACIÓN DE LOS DATOS

A partir de este momento ya se puede decidir de qué manera se van a representar visualmente los datos de las tablas creadas, esta personalización se puede realizar a través de la web de CartoDB, y a través de una aplicación web usando las diferentes APIs de que dispone. El proyecto finalizado se puede visitar en la dirección: http://level423.hol.es/

A. Visualización en la web de cartoDB

Una forma sencilla de crear una visualización es hacerlo mediante las herramientas de que dispone el *dashboard* en la vista del mapa (*Map View*):

SQL: se pueden realizar consultas a la base de datos para visualizar unos datos u otros en función del objetivo del mapa.

Visualización: Manera en que se muestran los datos el mapa, en función de si la geometría es punto, línea o polígono aparecen unos estilos u otros (simple, coropletas, burbujas, etc.). (Figura 3)

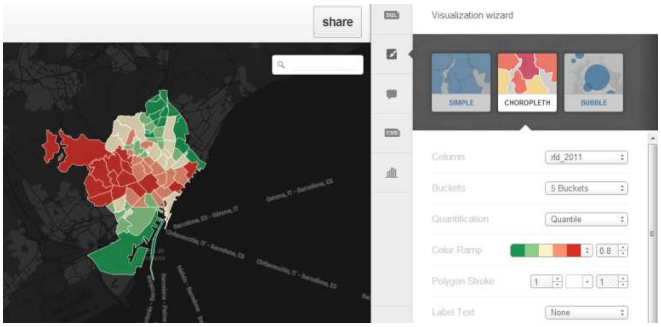

Fig. 3. Con el *Visualization wizard* se puede elegir la forma en que aparecen representados los datos en el mapa.

También permite elegir el mapa de fondo (figura 4)

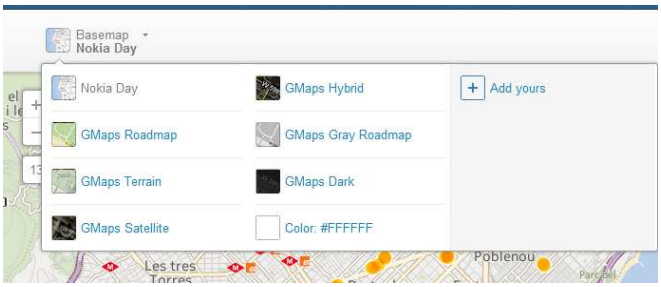

Fig. 4. Selección del mapa base de la visualización

Infowindows: con esta herramienta se seleccionan los campos que aparecerán al seleccionar un punto del mapa.

CartoCSS: permite personalizar el estilo del mapa (Figura 5).

Fig. 5. *CartoCSS editor*. Al seleccionar un estilo con el *visualization wizard* cambia automáticamente el CSS pero con esta herramienta se puede personalizar un poco más.

Desde el *Map View* también se pueden añadir, borrar o modificar registros de la tabla. (Figura 6)

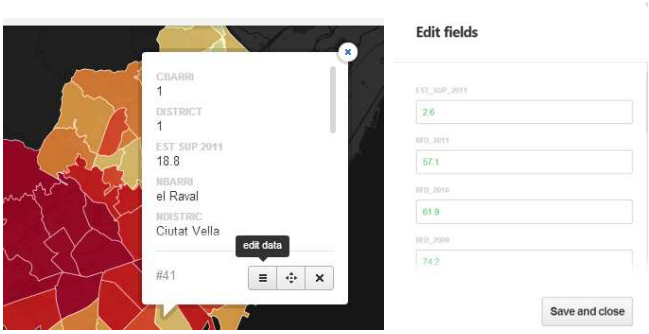

Fig. 6. CartoDB permite modificar los campos de un determinado registro desde la vista del mapa, no solo desde la vista de la tabla.

A partir de estos sencillos pasos CartoDB ya permite crear un mapa y compartirlo. Hay tres opciones (Figura 7):

- URL. Enlace a una página web que contendrá el mapa y la visualización que se ha decidido en el *dashboard*.
- EMBED MAP. Código para incluir el mapa dentro de otra página web, un blog etc.
- API. Enlace que se introduce dentro una aplicación web y que permite mostrar los datos de la tabla en forma de teselas (tiles).

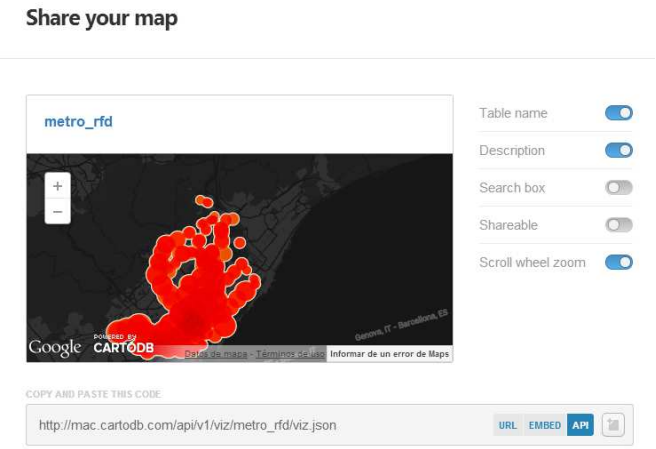

Fig. 7. Enlaces para compartir el mapa creado en el *dashboard*

B. Visualización en una aplicación web

Para este proyecto se ha creado una aplicación web en la que se muestran en tres capas distintas los datos de las tablas guardadas en los servidores de CartoDB.

Cada vez que una persona accede a la aplicación el navegador (el cliente) realiza una petición a la base de datos que está en CartoDB (el servidor) y sobre esas tablas devuelve los datos solicitados, no en formato vectorial, que sería demasiado pesado y lento para el tráfico web, sino en pequeños mosaicos de imágenes llamados teselas o tiles.

CartoDB ofrece varias *templates* que facilitan mucho el diseño de la aplicación. Para este proyecto se ha usado una llamada "*sidepanel*", que se puede descargar desde este enlace: http://cartodb.github.com/cartodb-publishing-templates/dist/sidepanel.zip

Se ha utilizado como mapa de fondo (*tile layer*) un mapa en blanco y negro basado en los datos de OpenStreetMap y creado por: http://maps.stamen.com. Para incluirlo en el código se inicializa la variable map que es donde se guarda el mapa y al que se añade una *titleLayer* con el mapa base elegido. (Figura 8)

```
function init(){
  // inicializa el mapa leaflet
  map = new L.Map('map', {
    center: [41.40,2.18],
    zoom: 12,
    minZoom: 10,
    maxZoom: 16
  })

  //define la capabase y la añade al mapa
  L.tileLayer(
    'http://tile.stamen.com/toner-background/{z}/{x}/{y}.png',
    { attribution: 'Stamen'
  }).addTo(map);
```

Fig. 8. Fragmento del código en el que se inicializa el mapa leaflet, definiendo el centro y el nivel de zoom, y al que se añade la capa base creada por Stamen.

En la versión actual de CartoDB, la 2.0, no hay soporte multi-capa para los tiles aunque si lo incluirá la versión 2.1 que saldrá en los próximos meses [2], así que en la aplicación se han creado tres capas CartoDB que se van añadiendo a un *array* de capas que permita más adelante encender o apagar las mismas. (Ver figura 9).

Una de las formas más fáciles de añadir una capa CartoDB es incluir el enlace de la visualización que se creó en el *dashboard* de CartoDB, el mapa aparecerá con la *query*, css e *infowindows* seleccionadas en desde la web:

cartodb.createVis('map','http:// <USER NAME> .cartodb.com/api/v1/viz/ <NOMBRE TABLA>/viz.json');

El documento Viz.JSON contiene toda la información del mapa creado, pero lo realmente interesante de CartoDB es que permite personalizar las visualizaciones a través de las APIs. Desde el código se puede seleccionar la tabla, la *query*, y el estilo CSS que aparecerá en el mapa que creas, y que no tiene porque ser el mismo que se creó desde el *dashboard*. De esta manera la visualización se crea cada vez que cargas la página donde está incluido ese mapa.

En la figura 9 se puede ver como se crea la primera capa del mapa, que es la que muestra los barrios, en un mapa de coropletas con distintos colores en función de su índice RFD. (Ver la figura 10 con el código CartoCSS utilizado).

```
//define un array de layers
var layers=[];

//crea capa con la tabla 'renta barrios08_11': layers[0]
var layer0Url =
  'http://mac.cartodb.com/api/v1/viz/renta_barris_08_11_e/viz.json';

//define la query y el estilo CartoCSS iniciales para la capa[0] (renta_b
var layer0Options = {
  query: "SELECT * FROM {{table_name}}",
  tile_style: "#{{table_name}}{[ rfd_" + selected_year + " <= 250]{polygo
};
//carga la capa CartoDB
cartodb.createLayer(map, layer0Url, layer0Options)
  .on('done', function(layer) {
    layer.infowindow.set('template', $('#infowindow_template0').html());
    map.addLayer(layer);
    layers.push(layer);//mete la layer[0] al array de capas
    createSelector(layer);
    //añade un evento de click que llama a la funcion que dibuja el
    //gráfico pasandole como parámetro el cartodb_id
    layer.on('featureClick', function(e, pos, latlng, data) {
      $('#chartContainer').removeClass('hide');
      charts(data.cartodb_id);
    });
  }).on('error', function() {
    //log the error
  });
```

Fig. 9. Fragmento de código que muestra cómo crear el array de layers y como se van añadiendo la capas que se necesiten.

```
//define la query y el estilo CartoCSS iniciales para la capa[0]
var layer0Options = {
  query: "SELECT * FROM {{table_name}}",
  tile_style: "#{{table_name}}{
    [ rfd_" + selected_year + " <= 250]{polygon-fill: #B10026;}
    [ rfd_" + selected_year + " <= 180]{polygon-fill: #E31A1C;}
    [ rfd_" + selected_year + " <= 140]{polygon-fill: #FC4E2A;}
    [ rfd_" + selected_year + " <= 110]{polygon-fill: #FD8D3C;}
    [ rfd_" + selected_year + " <= 80]{polygon-fill: #FEB24C;}
    [ rfd_" + selected_year + " <= 70]{polygon-fill: #FED976;}
    [ rfd_" + selected_year + " <= 60]{polygon-fill: #FFFFB2;}
    [ rfd_" + selected_year + " <= 45]{polygon-fill: #FFFFCC;}
    line-color: #FFF;
    line-opacity: 1;
    line-width: 1;
    polygon-opacity: 0.8;}"
};
```

Fig. 10. Fragmento de código que muestra como se ha realizado el mapa de coropletas a partir de la tabla que contiene las geometrías de los barrios y los índices RFD de los distintos años y mediante CartoCSS.

En el código de la Figura 9 se ha añadido un evento de *click* que hace que se muestre un gráfico con la evolución del índice RFD en el barrio seleccionado. El gráfico se ha realizado con HighCharts (http://www.highcharts.com/) que es una librería de *JavaScript* que permite incluir gráficos dinámicos páginas web.

Para conseguir que el grafico se cree con los datos específicos del barrio seleccionado hay que realizar una petición getJSON al servidor de CartoDB donde se encuentra la tabla a la que se hace la consulta. (Figura 11)

```
//la funcion charts recibe por parámetro el cartodb_id
//del barrio seleccionado en el mapa
//y hace una peticion getJson a la tabla
function charts(data_id) {
    var campo = data_id;
    var url =
        "http://mac.cartodb.com/api/v1/sql?q=
        SELECT nbarri,rfd_2008,rfd_2009,rfd_2010, rfd_2011
        FROM renta_barris_08_09
        WHERE cartodb_id="+campo+""
    console.log(url);
    var cartoItems = [];
    $.getJSON(url, function(data) {
        $.each(data.rows, function(key, val) {
            cartoItems.push(val.rfd_2008);
            cartoItems.push(val.rfd_2009);
            cartoItems.push(val.rfd_2010);
            cartoItems.push(val.rfd_2011);
            barrio= val.nbarri;
        });
        console.log(cartoItems);
        console.log(barrio);
    creaChart(cartoItems,barrio);
    });
};
```

Figura 11. Fragmento de código donde se muestra como se realiza la petición a la tabla y recibe los datos en formato JSON que se introducen en un *array* para pasárselos a la función que crea el gráfico con *HightCharts*

En respuesta a esa petición el servidor devuelve los datos de la consulta realizada en formato JSON (*JavaScript Object Notation*, formato para el intercambio de datos):

{"time":0.002,
 "total_rows":1,
 "rows":
     [{"nbarri":"la Trinitat Vella",
       "rfd_2008":76.4,
       "rfd_2009":69.9,
       "rfd_2010":58.1,
       "rfd_2011":61.8}
       ]}

Cada vez que se selecciona un barrio aparece un grafico como el de la figura 12:

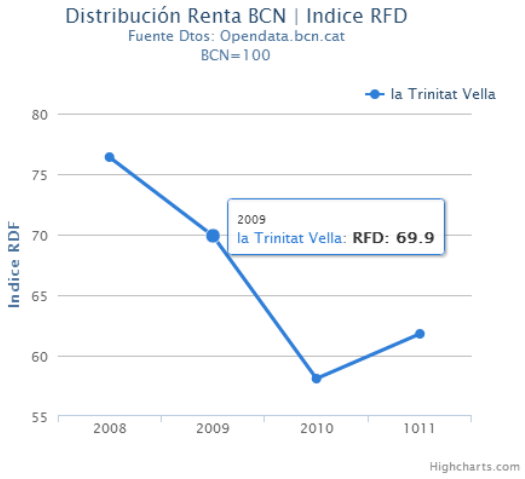

Fig. 12. Grafico con la evolución del índice RFD del barrio de la Sagrera entre los años 2008 a 2011. HightCharts permite que haya una interacción y que al pasar el cursor sobre el grafico veamos un *tooltip* con los valores de ese punto.

La capa con las estaciones de metro, que son elementos puntuales, se ha representado como un mapa de intensidad en el que en función del nivel de zoom el mapa muestra un aspecto otro. En la figura 13 se puede ver que si el mapa tiene poco zoom de un primer vistazo puedes ver que hay zonas de color azul (que indican rentas más bajas) y zonas de color verde (que indican rentas más altas). Cuanto más te acerques en el mapa se verán con más detalle los puntos que indican las estaciones de metro, cada uno con un valor de renta (índice RFD) diferente, es entonces cuando se puede seleccionar sobre la estación que interese y ver la información.

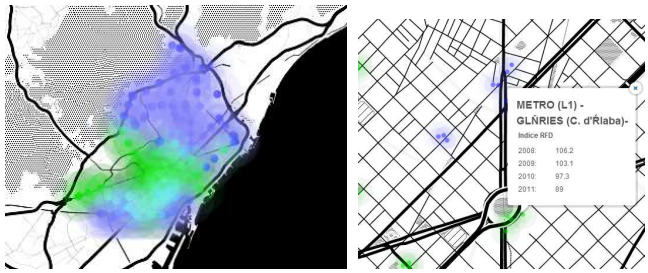

Figura 13. Visualización de la capa de las estaciones de metro en diferentes niveles de zoom.

Para conseguir este efecto parecido a un *heatmap* se han creado tres grupos de marcadores para un mismo punto con CartoCSS, y cada uno de ellos tiene un color, una opacidad y un tamaño diferente (ver la Figura 14 donde se muestra el código utilizado).

```
//crea una capa con la tabla de las estaciones, 'metro_rfd': layer
var layer2Url =
  'http://mac.cartodb.com/api/v1/viz/metro_rfd/viz.json';
var layer2Options = {
  query: "SELECT * FROM {{table_name}}",
  tile_style: "#{{table_name}}{
    [ rfd_"+selected_year+" <= 100]
      {first/marker-fill: #9595FF;
      second/marker-fill: #5353FF;
      third/marker-fill: #0000FF}
    [ rfd_"+ selected_year+" > 100]
      {first/marker-fill: #008000;
      second/marker-fill: #0FFF0F;
      third/marker-fill: #00E100;
    }
    first/marker-opacity: 0.05;
    first/marker-width: 50;
    first/marker-line-width: 0;
    first/marker-placement: point;
    first/marker-allow-overlap: true;
    first/marker-comp-op: lighten;
```

Figura 14. Fragmento con el código de CartoCSS con el que se crea el mapa de intensidad de la capa de las estaciones de metro

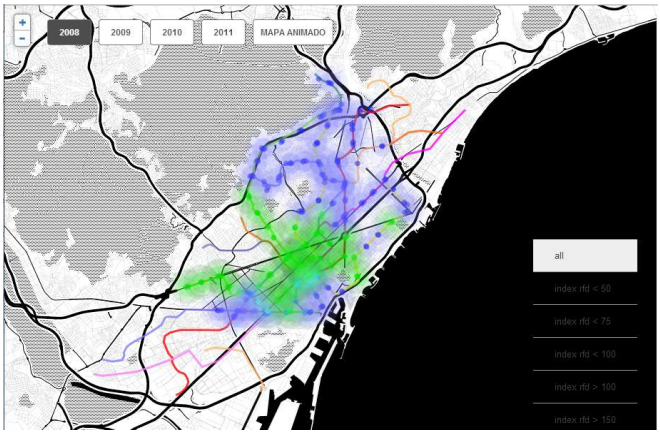

Fig. 15. Vista general de la aplicación web donde se han seleccionado las capas de las estaciones (*heatmap*) y de las líneas de metro y se ha apagado la capa con los barrios. Los colores del *heatmap* están en función del índice rfd del barrio en el que esta localizada la estación: verde si está por encima de 100 azul si está por debajo de ese valor medio.

La visualización de la capa con las estaciones de metro no es un *heatmap* propiamente dicho porque los valores que muestran las geometrías tipo punto no son realmente valores puntuales, sino que se corresponden con el valor general del barrio (polígono) al que pertenece la estación. La utilización de estos elementos puntuales se justifica porque añaden otro tipo de información al mapa, una visión global de Barcelona que indica si esa zona tiene una renta por encima de la media o por debajo.

Otra de las funcionalidades incluidas en la aplicación web es que permite al usuario realizar una selección de los barrios que quiere visualizar en función de su índice RFD. Cuando se crea la capa hay que incluir un *createSelector* (layer), que permite después cambiar la consulta (*query*) realizada a esa tabla a partir de un evento de *click*. (Figura 16)

```
//seleccion por query al index_rfd
$options.click(function(e) {

  // get the area of the selected layer
  var $li = $(e.target);
  var indice = $li.attr('data');

  // deselect all and select the clicked one
  $options.removeClass('selected');
  $li.addClass('selected');

  // create query based on data from the layer
  var query =  "SELECT * FROM {{table_name}}";
  if(indice !== 'all') {

    query = "SELECT * FROM {{table_name}} where rfd_"+ selected_year +" <" +indice ;

    if(indice == '100' || indice == '150' || indice == '200') {
      query = "SELECT * FROM {{table_name}} where rfd_"+ selected_year +" >" +indice ;
    }
  }
```

Figura 16. Fragmento de código donde se muestra como se puede cambiar la consulta que se realiza a la tabla (vía SQL API) para que se muestren los barrios con más o menos índice RFD.

Desde la aplicación web se puede seleccionar que consulta se desea visualizar. La consulta se aplica sobre los datos de la tabla pertenecientes al año que este seleccionado en ese momento. (Figura 17)

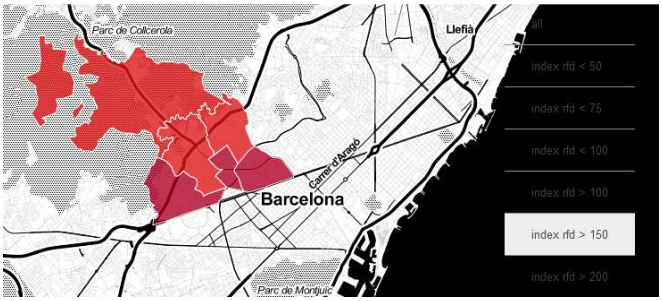

Figura 17. Vista de la aplicación web donde se ha seleccionado los barrios que tienen un RFD_{2008}>150

Finalmente se ha incluido dentro del mapa un enlace al mismo mapa, pero que contiene una pequeña animación de las capas para que vaya pasando de un año a otro sin tener que seleccionarlo con un *click*. En CartoDB han desarrollado una librería específica para realizar mapas en movimiento llamada Torque [https://github.com/CartoDB/torque] pero debido a que no se disponía del tiempo necesario para aprender su funcionamiento se ha descartado su utilización. En su lugar se ha creado un conjunto de funciones que van cambiando el año seleccionado y la *query* con la que se crean las capas cada cierto tiempo (en este caso cada 1.5sg). Ver código en la figura 18.

```
var timeRetard = 1500;

function cambiaAño(){
  $('.year').removeClass('selected');
  $('#'+selected_year).addClass('selected');
}

function draw2008 (){
  selected_year = 2008;
  cambiaAño();
  updateQuery0();
  updateQuery2();
  setTimeout(draw2009, timeRetard);//llama a draw2009 en 1'5 sg
}
function draw2009 (){
  selected_year = 2009;
  cambiaAño();
  updateQuery0();
  updateQuery2();
  setTimeout(draw2010, timeRetard);//
}
function draw2010 (){
  selected_year = 2010;
  cambiaAño();
  updateQuery0();
  updateQuery2();
  setTimeout(draw2011, timeRetard);//
}
function draw2011 (){
  selected_year = 2011;
  cambiaAño();
  updateQuery0();
  updateQuery2();
  setTimeout(draw2008, timeRetard);
}
```

Figura 18. Fragmento de código que realiza una pequeña animación actualizando la *query* en función del año mediante un retardo en la llamada a la función (*SetTimeOut*)

En la capa de las estaciones que se muestra en el mapa animado se ha cambiado ligeramente el CartoCSS para que note algo más los cambios. No se cambian los colores, pero se juega con las opacidades y el tamaño del *marker* en función de su índice RFD. Ver figura 19 con el código CSS:

```
//crea una capa con la tabla 'metro_rfd': layers[1]
var layer2Url = 'http://mac.cartodb.com/api/v1/viz/metro_rfd/viz.
var layer2Options = {
  query: "SELECT * FROM {{table_name}}",
  tile_style: "#{{table_name}}{
    [ rfd_"+selected_year+" <= 55]
     {first/marker-fill: #9595FF;
       second/marker-fill: #5353FF;
       third/marker-fill: #0000FF;
       first/marker-opacity: 0.12;
       first/marker-width: 65;
       second/marker-opacity: 0.20;
       second/marker-width:35;
       third/marker-opacity: 0.9;
       third/marker-width:9;}
    [ rfd_"+selected_year+" <= 75]
     {first/marker-fill: #9595FF;
       second/marker-fill: #5353FF;
       third/marker-fill: #0000FF;
       first/marker-opacity: 0.07;
       first/marker-width: 55;
       second/marker-opacity: 0.10;
       second/marker-width:30;
       third/marker-opacity: 0.7;
       third/marker-width:8;}
    [ rfd_"+selected_year+" <= 100]
     {first/marker-fill: #9595FF;second/marker-fill: #5353FF;thi
```

Fig. 19. Fragmento de código donde se cambia el tamaño y la opacidad de las estaciones en función de su valor del índice RFD.

IV. ANÁLISIS

Una vez se ha obtenido el mapa web con la visualización de los datos se procede a realizar el análisis de esos datos.

Para cuantificar los ingresos de que disponen los residentes de un territorio para destinarlos al consumo y al ahorro, se usa una variable estadística conocida como Renta Familiar Disponible. Este índice se calcula sumando los ingresos de los asalariados, o los beneficios de las empresas, y restando las cotizaciones a la seguridad social y los impuestos (sean sobre la renta de las personas físicas, de las sociedades o del patrimonio). [3]

El Ayuntamiento de Barcelona publica un índice RFD anual para cada uno de los 73 barrios en los que está distribuida la ciudad. Este Índice RFD se construye a partir de los datos de Renta Familiar Bruta Disponible y de la Renta Familiar Bruta Disponible per cápita de Barcelona, a los que se les añaden una serie variables que hacen referencia al nivel educativo (tasa de titulados), situación laboral (tasa de paro), evolución del parque de vehículos (nº turismos por cada 1000 habitantes), y el precio de la vivienda de segunda mano. [4]

A partir de estos datos se calcula un índice RFD para cada barrio de la ciudad tomando como base que la media de Barcelona es RFD= 100.

A. Distribución espacial del índice RFD por barrios

Desde que a mediados del siglo XIX la burguesía de Barcelona comenzó a elegir la zona alta (por entonces alejada de la ciudad) para veranear, las clases acomodadas siguen eligiendo esos barrios para vivir. Esta percepción se confirma con los datos estadísticos, los barrios con las rentas más elevadas se sitúan en la zona alta de la ciudad: Pedralves, Sant

Gervasi, Sarrià, Vallvidrera o las Tres Torres. (Figura 20)

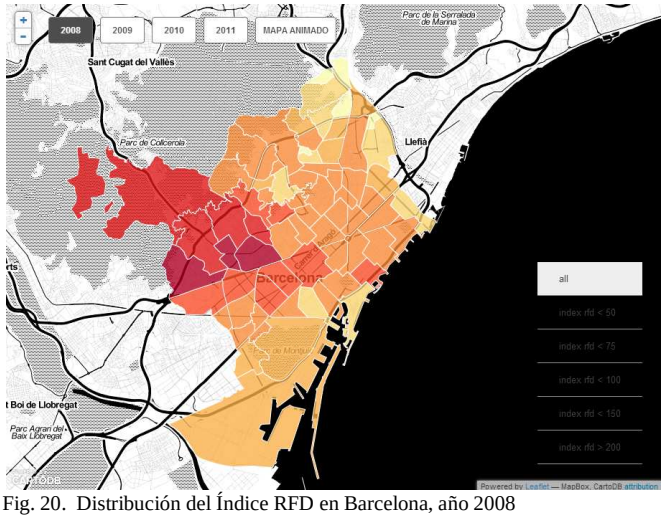

Fig. 20. Distribución del Índice RFD en Barcelona, año 2008

Durante el periodo estudiado se puede observar que hay una gran polaridad en la distribución de la renta durante estos años de crisis económica. Mientras que en un barrio pobre como Can Peguera, que en 2008 tenía un índice de 52.8, la pobreza no ha dejado de aumentar, llegando a un RFD_{2011}=34 (ver Figura 21), en un barrio rico como Pedralbes la riqueza no ha parado de aumentar, pasando de 194.7 en 2008 a 241.5 en 2011 (figura 22).

Esta gran desigualdad existente en la ciudad ha ido aumentando de manera espectacular, incidiendo además en otros aspectos no económicos, como la esperanza de vida. Un dato muy esclarecedor: "los vecinos de los barrios ricos de Barcelona viven ocho años más que los del Raval (con una renta familiar de 64,2), 81 años frente a 73". [5]

En 2011 la diferencia entre el barrio con la renta más alta y el de la renta más baja es de 207 puntos, mientras que en el año 2008 era de 164 puntos.

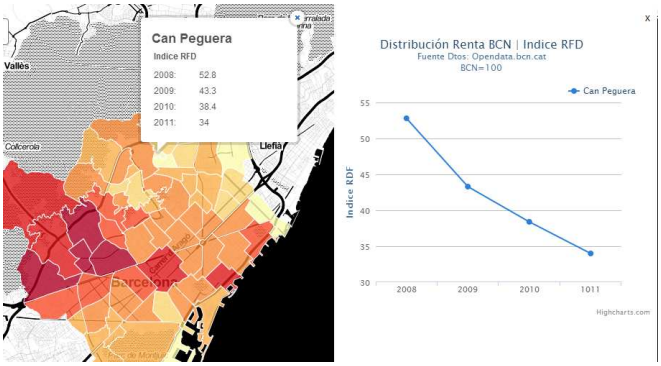

Fig. 21. Mapa con la distribución de la renta en 2011, en el que se ha seleccionado el barrio de Can Peguera, que tiene el RFD_{2011}=34, el más bajo de toda la ciudad

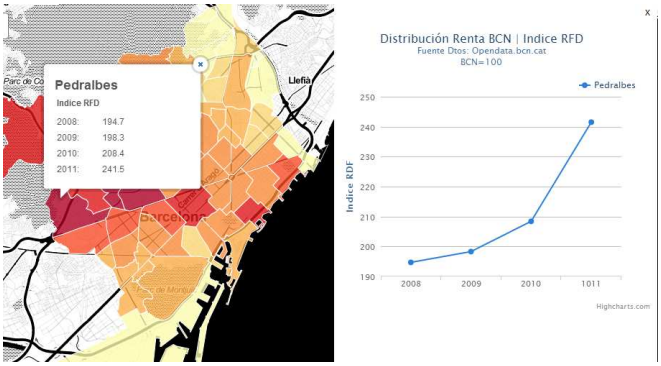

Fig. 22. Mapa con la distribución de la renta en 2011, en el que se ha seleccionado Pedralbes, que tiene el RFD_{2011}=241.5, el más alto de toda la ciudad

Si realizamos una comparativa de los barrios con índice RFD<100 en 2008 y 2011 se ve cómo ha aumentado el número de barrios con rentas por debajo de la media. En 2008 había 50 barrios, y en 2011 ya eran 55. (Figura 23)

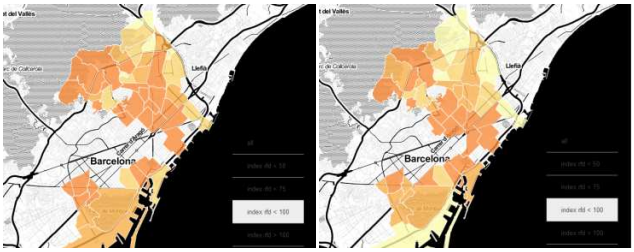

Fig. 23. Imagen de la derecha: Barrios con RFD <100 en 2008 (50 en total), Imagen de la izquierda, misma selección pero en el año2011 (55 barrios en total)

B. Distribución lineal del índice RFD y su relación con las líneas de metro

Si observamos la distribución de los barrios con mayores rentas de Barcelona (Figura 24) se puede comprobar que son zonas donde prácticamente no pasan las líneas de metro de la ciudad. En estos barrios abundan las viviendas unifamiliares y edificios de lujo de cuatro vientos [5], y tienen una densidad de población muy por debajo de la media. En el distrito de Sarrià-Sant Gervasi la densidad está en torno a los 7.000 hab/m^2, mientras que la media de Barcelona se sitúa en torno los 16.000 hab/km^2. [http://www.bcn.cat/estadistica/castella/index.htm]

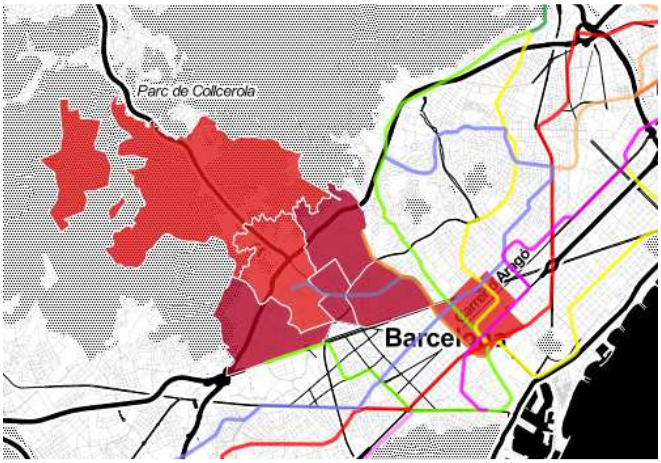

Fig. 24. Selección de los barrios con RFD_{2011}>150 que muestra la relación entre la ausencia de líneas de metro en los barrios con rentas más altas debido a que en estos barrios predominan las casas unifamiliares.

C. Distribución del índice RFD y su relación con el porcentaje de población con estudios superiores:

Aunque no se ha incluido como capa en la aplicación web desarrollada, si que se pueden visualizar en el *dashboard* de CartoDB los datos de población con estudios superiores para el año 2011 por barrios, observándose una equivalencia entre las zonas con mayor tasa de estudios y las de índice RFD más alto (Figura 25).

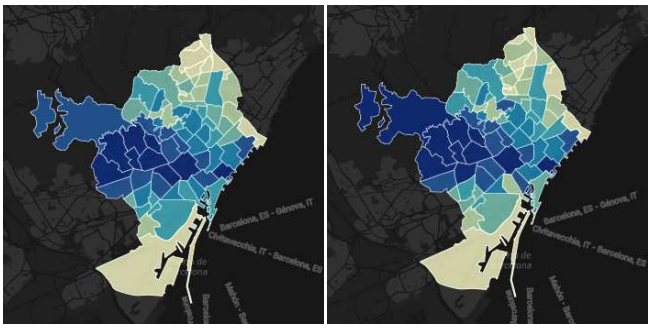

Fig. 25. Equivalencia casi total entre la distribución de los barrios por porcentaje de población con estudios superiores (izquierda) y la distribución en función del índice RFD (derecha) para el año 2011. Los tonos más oscuros indican los valores más elevados en ambas imágenes.

V. CONCLUSIÓN

Una vez finalizado el presente proyecto se puede afirmar que el uso de servicios GIS en la nube, como es el caso de CartoDB, es un gran aliado a la hora de trabajar con datos espaciales. Permite realizar análisis espaciales con PostGIS de manera fácil, y sin tener que realizar ningún tipo de instalación de software. CartoDB es una herramienta muy fácil de utilizar desde el *dashboard* de su página web, y relativamente simple si se quieren usar sus APIs en el desarrollo de aplicaciones web, como ha sido el caso de este proyecto. Se dispone de abundante documentación y tutoriales (tanto en su web como en su cuenta de GitHub) llenos de ejemplos que resultan de gran ayuda cuando se dan los primeros pasos en el desarrollo de una aplicación con CartoDB.

La gran ventaja de usar este tipo de herramientas es que fácilmente se pueden publicar los mapas obtenidos en la web para que cualquiera pueda verlos, e interactuar con ellos. No siendo necesario que ese usuario final tenga conocimientos sobre SIG, ni que sepa usar los clásicos SIG de escritorio (como ArcGIS o gvSIG), para que obtenga la información que busca al ver el mapa.

Es esa interactividad que se le ofrece al usuario final la que resulta más positiva del uso de estos servicios en la nube. Y la que a la larga hará que estas herramientas prácticamente sustituyan a los SIG de escritorio.

Quizás la única limitación que se ha encontrado a CartoDB durante la realización de este artículo hace referencia a las proyecciones. Según la documentación de su web, al subir datos a CartoDB el sistema detecta automáticamente el SRID original y realiza la trasformación a WGS84, que es el sistema que utiliza por defecto. El problema es que la proyección ED50 (que todavía es muy común en España) no está incluida en su librería (ver Figura 26). Por esa razón el shape con los barrios se hubo de transformar con ArcGis a WGS84 antes de poder subirlo a CartoDB.

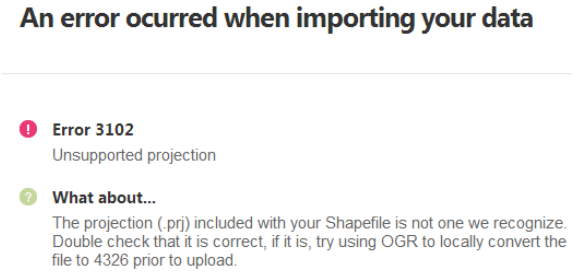

Fig. 26. Error en CartoDB al intentar importar un shape en ED5031N

La aplicación web realizada para este proyecto cumple el objetivo inicial de conseguir una visualización que permita detectar los cambios observados en la renta de las familias que viven en los distintos barrios de Barcelona durante estos últimos años de crisis económica. Y como ha afectado de manera muy diferente a unos barrios y a otros, aumentando de manera significativa la distancia en las rentas de los barrios más pobres a los más ricos.

Como propuesta de mejora para futuros trabajos como este sería conveniente contar con datos sobre el índice RFD anteriores a 2008 para observar una evolución mucho más completa. En este proyecto no ha sido posible ya que la actual división de barrios de Barcelona data de 2007 [4], y en la web del ayuntamiento no se dispone de información extrapolable a la división actual. También sería conveniente mejorar la animación realizada en la aplicación, ya que se notan demasiado los saltos cuando carga las capas con los distintos años, seguramente utilizando librerías como torque o D3 se solucionarían estos problemas.

Las futuras líneas de trabajo pueden ir encaminadas a seguir actualizando la aplicación conforme el ayuntamiento publique los datos del índice RFD en los años venideros. Incluir más criterios de visualización, por ejemplo, algún formulario que permitiera introducir consultas SQL simples. También se podían incluir otro tipo de datos, como el cierre total o parcial de ambulatorios, o esperanza de vida en cada barrio.

AGRADECIMIENTOS

Gracias a Anna Muñoz por su ayuda y consejos durante la realización de este proyecto.

REFERENCIAS

[1] Poverty in Chicago as a Transit Map <http://www.ctapovertymap.org/>

[2] Blog de CartoDB, "*CartoDB 2.1 is almost here*" <http://blog.cartodb.com/post/53932512302/cartodb-2-1-is-almost-here>

[3] R. Madariaga; J. C. Martori; y R. Oller. *Distribución espacial y desigualdad de la renta salarial en el Área Metropolitana de Barcelona*. Scripta Nova. Revista Electrónica de Geografía y Ciencias Sociales. [En línea]. Barcelona: Universidad de Barcelona, 20 de junio de 2012, vol. XVI, nº 405. <http://www.ub.es/geocrit/sn/sn-405.htm>. [ISSN: 1138-9788].

[4] M. J. Calvo, X. Güell, J. Salabert, "*Distribució territorial de la Renda Familiar per càpita a Barcelona"* , Ajuntament de Barcelona <http://w150.bcn.cat/publicacions/sites/default/files/fitxers/documents_gestio/rfd.pdf> [ISBN: 978-84-9850-005-9] Enero 2007

[5] C. Blanchar. "*Crece la Barcelona pobre*" artículo publicado en El País el 6 Enero 2013, <http://ccaa.elpais.com/ccaa/2013/01/05/catalunya/1357414914_291755.html >