

HERRAMIENTA BASADA EN SOFTWARE LIBRE PARA EL ANÁLISIS Y GEOLOCALIZACIÓN DE ENFERMEDADES TRANSMITIDAS POR VECTORES (ETV) EN EL DEPARTAMENTO DEL PUTUMAYO

Miguel Evangelista Villota Escandón

Programa de Maestría en Software Libre Convenio UNAB-UOC, Universidad Autónoma de Bucaramanga – UNAB,
Bucaramanga, Colombia
meve82@gmail.com

Abstract— Este documento presenta un resumen general de todo el proceso llevado a cabo para el desarrollo de un sistema basado en software libre que permite el análisis y geolocalización de enfermedades transmitidas por vectores (ETV) en el Departamento del Putumayo. El artículo tiene especial énfasis en la descripción del código fuente que se implementó para la alimentación de datos, para la generación y procesamiento de mapas en línea y para las opciones de transferencia de información.

I. INTRODUCCION

Es común encontrar propuestas exitosas que simplifican el trabajo de las personas y que solucionan problemas a cualquier nivel sin importar la cantidad de datos existentes, así como el tipo de usuario que lo requiere, todo esto enmarcado dentro de los avances tecnológicos que cada día mejoran y que permiten la utilización de herramientas capaces de agilizar y visualizar de manera eficiente y de forma segura los resultados que un proceso de datos genera.

Entre los avances tecnológicos derivados de las TIC's encontramos los Sistemas de Información Geográficos comúnmente llamados SIG, los cuales permiten espacializar información buscando optimizar flujos de trabajo y reducir costos operativos de manera innovadora, todo esto centrado en tres objetivos generales: captura, análisis y presentación de información.

Todos estos avances integrados correctamente se convierten en herramientas poderosas para el análisis y procesamiento de información de cualquier tipo, logrando respuestas eficientes a cualquier clase de preguntas.

El Departamento del Putumayo localizado en el Sur de la Amazonia Colombiana, no puede ser ajeno a estos beneficios y más aún, teniendo en cuenta que por su geolocalización y diversos climas, es muy común que existan diversos tipos de males propios de la zona (selva húmeda) como enfermedades transmitidas por vectores clasificadas como parasitarias (leishmaniasis, toxoplasmosis, malaria) y enfermedades producidas por virus (dengue, fiebre amarilla, hepatitis).



Fig. 1 Localización del Departamento del Putumayo en Colombia

El presente artículo plantea una solución que combina las potencialidades de SIG y de Bases de Datos de tal forma que permita un análisis en tiempo real de la densidad de eventos relacionados con enfermedades transmitidas por vectores y que permitirá a las autoridades de salud tomar decisiones favorables para los habitantes de las zonas afectadas.

II. ESTADO ACTUAL DE VIGILANCIA DE ETVs

A nivel Mundial las diferentes organizaciones de la Salud, sin importar su origen, geolocalización y poder económico, tienen como uno de sus objetivos la vigilancia epidemiológica, que se convierte en una estrategia especial para la detección de enfermedades y para poder llevar a cabo programas de prevención, control y erradicación de las mismas. De la misma forma y dependiendo de los resultados de la vigilancia, asignan el estatus “libre de enfermedad o infección”.

La vigilancia epidemiológica brinda datos para apoyar el proceso de análisis de riesgos para fines de salud animal y pública y de paso apoyar y justificar la aplicación o no de las políticas sanitarias de una administración en Salud.

Además permite determinar la aparición o distribución de enfermedades o infecciones.

En Colombia la vigilancia epidemiológica viene liderada por el Instituto Nacional de Salud, a través del sistema Nacional de Vigilancia en Salud Pública – SIVIGILA-, para lo cual se han establecido una serie de protocolos que permiten a los profesionales de la salud reportar la información de acuerdo a los parámetros que SIVIGILA exige, de tal manera que permita su análisis y posterior publicación para la toma de decisiones.

Para realizar las acciones de vigilancia epidemiológica, se requiere agrupar los padecimientos o eventos epidemiológicos alrededor de un período de tiempo determinado. Este período es generalmente de una semana y se le conoce como semana epidemiológica; a su vez, a la división de los 365 días del año en semanas epidemiológicas, se le conoce como calendario epidemiológico, el cual es un instrumento de estandarización de la variable tiempo para los fines de la vigilancia epidemiológica.

La importancia de la división y sobre todo de la utilización de las semanas epidemiológicas, radica en que permite la comparación de eventos epidemiológicos sucedidos en determinado año o período dentro de un año, con los de años previos. Facilita asimismo, la comparación entre países, dado que se trata de una metodología epidemiológica adoptada oficialmente en el ámbito internacional.

Las semanas epidemiológicas inician en domingo y terminan en sábado; para designar la primera semana epidemiológica del año, se ubica el primer sábado de enero que incluya en los días inmediatamente precedentes, cuatro o más días del mes de enero, aun cuando esa primera semana se inicie en diciembre.

III. HERRAMIENTAS DE SOFTWARE LIBRE UTILIZADAS PARA EL DESARROLLO DEL SISTEMA

La selección está enfocada al servidor web, el cual debe contar con soporte para datos espaciales y ser de uso libre, y el servidor más completo y ajustado a estos requerimientos básicos es Mapserver. Se valora también los lenguajes de programación a utilizar tanto de servidor como del lado del cliente, optando por utilizar PHP (para servidor) dada su robustez, documentación y facilidad de uso y del lado del cliente HTML, JavaScript, y jQuery ya que son los lenguajes de mayor uso en la actualidad para aplicativos web y está más que demostrado su poder.

Otro aspecto importante es el de la representación espacial de los objetos creados (puntos), el cual debe ser compatible con las herramientas tecnológicas antes descritas, también tiene que contar con amplia documentación y sobre todo brindar el servicio sin costo alguno y entre los mejores está Google Maps, porque es un servicio de Google que ofrece imágenes vía satélite de todo el planeta, combinadas, en el caso de algunos países, con mapas de sus ciudades, lo que unido a sus posibilidades de programación abierta ha dado lugar a diversas utilidades ofrecidas desde numerosas páginas web.

Un aspecto importante es el costo de uso de esta tecnología, ya que los sitios web y las aplicaciones que utilizan el API de Google Maps pueden generar de forma gratuita hasta 25.000 cargas de mapas al día por cada servicio y en caso de sobrepasar este límite de

solicitudes, se requerirá la adquisición de licencias del API de Google Maps for Business.

Es muy importante tener en cuenta los formatos universales que pueden adoptarse para el intercambio geoespacial de información y que sean compatibles 100% con Google Maps, uno de estos formatos lo provee GeoJson ya que maneja un estándar mundial, se acopla sin problemas a PHP y permite representar fácilmente geometrías de tipo punto, línea, polígono, multipunto, multilínea y multipolígono entre otras.

IV. ANÁLISIS, DESARROLLO Y PUBLICACIÓN DEL SISTEMA

A. Modelo Relacional Base de Datos

Información básica: Tipo de vector, Fecha del evento, Observaciones, coordenadas de latitud y longitud, municipio y el punto geométrico. Toda esta información se suportó en PostgreSQL, distribuidos en tres tablas:

Puntos: Registro de los casos de enfermedades transmitidas por vectores que se reportaran por medio del sistema, tiene un campo (geom) que contiene una geometría de tipo punto.

Vector: Lista de vectores que transmiten las enfermedades

Putumayo: Almacena todos los shapes de los 12 Municipios del Putumayo. Tiene un campo (geom).

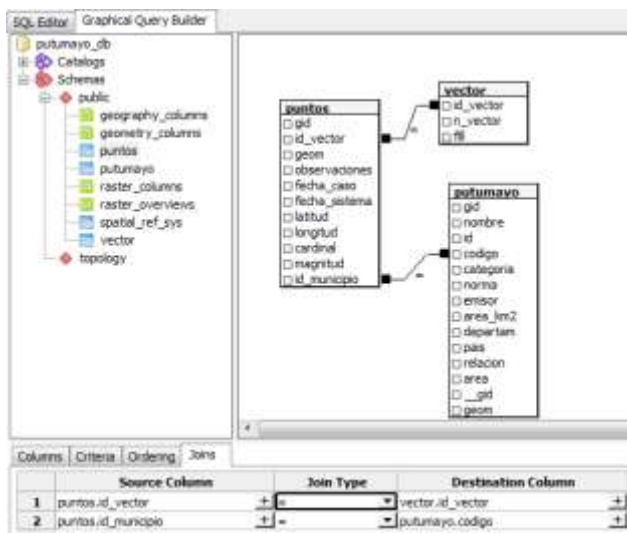


Fig. 2 Modelo Básico de la Base de Datos en PostgreSQL

El desarrollo del sistema estuvo enfocado en la facilidad de uso y optimización para la velocidad de captura, procesamiento y presentación de datos.



Fig. 3 Mapa de navegación del Sistema

El redimensionamiento y diseño de las pantallas están a cargo de jQuery Mobile y la marca <meta name="viewport" content="width=device-width"/> de HTML.

B. Evaluación de Coordenadas

Cuando el usuario se encuentra ingresando eventos, el sistema utiliza funciones espaciales para verificar que dicho punto se encuentre en el Departamento.

Se efectúan las conversiones para obtener grados en formato decimal y se ejecuta la consulta espacial basadas en st_within y GEOMETRYFROMTEXT con el SIRD 4326.

```

<code>
</code>

```

Fig. 4 Código fuente utilizado para la evaluación de puntos.

C. Almacenamiento de registros

El almacenamiento del punto registrado se ejecuta en segundo plano y retorna un mensaje al usuario dependiendo del tipo de resultado obtenido, que puede ser error o correcto almacenamiento.

```
<?php
require_once("../conector/codebase/config.php");
require("../conector/codebase/db_postgre.php");
$lat_gra = $_REQUEST['lat_gra'];$lat_min = $_REQUEST['lat_min'];$lat_seg = $_REQUEST['punto_card'] = $_REQUEST['punto_card'];$lon_gra = $_REQUEST['lon_gra'];$lon_min = $_REQUEST['lon_seg'] = $_REQUEST['lon_seg'];$lat_min = $lat_min/60;$lat_seg = $lat_seg/3600;
$latitud = ($lat_gra + $lat_min + $lat_seg);$lon_min = $lon_min/60;$lon_seg = $lon_s
$longitud = ($lon_gra + $lon_min + $lon_seg)*-1;if($punto_card==2) $latitud = ($lat_gr
$fecha = explode("-", $_REQUEST['fecha']);
$fecha = $fecha[0]. "-" . $fecha[1]. "-" . $fecha[2];

$consultasql = "insert into puntos (id_vector, geom, observaciones, fecha_caso, fecha
('' . $_REQUEST['vector'] . "', GEOMETRYFROMTEXT('POINT (' . $longitud . " " . $latitud
'fecha' . " . $fecha . " . " . $_REQUEST['municipio'] . " . " . $latitud . " . " . $lon

$result = pg_query($sig_db, $consultasql);

if($result)
    echo "OK";
else
    echo "SIN MUNICIPIO";
?>
```

Fig. 5 Código fuente para el almacenamiento de un caso de enfermedad.

D. Creación del archivo GeoJson

El archivo GeoJson que se genera para la representación de puntos a través de Google Maps y que puede ser utilizado desde otras aplicaciones utiliza un ciclo para acceder el resultado de una consulta y generar línea a línea el contenido del archivo. Primero se define la estructura en un arreglo y luego se alimenta cada una de las propiedades del mismo y por último se genera utilizando la función `json_encode()`.

```
$result = pg_query($sig_db, $consultasql);
$geojson = array(
    'type' => 'FeatureCollection',
    'features' => array()
);

while ($row = pg_fetch_assoc($result)) {
    $properties = $row;
    $feature = array(
        'type' => 'Feature',
        'geometry' => array(
            'type' => 'Point',
            'coordinates' => array(
                $row['longitud'],
                $row['latitud']
            )
        ),
        'properties' => $properties
    );
    array_push($geojson['features'], $feature);
}

echo json_encode($geojson, JSON_NUMERIC_CHECK);
$conn = NULL;
```

Fig. 5 Código fuente para la generación del archivo GeoJson

E. Creación del archivo XML

El archivo XML se genera para el posible intercambio de datos con otras entidades. Se crea utilizando un ciclo para acceder el resultado de una consulta y generar línea a línea el contenido del archivo.

```
$dom = new DOMDocument("1.0");
$node = $dom->createElement("Datos");
$parnode = $dom->appendChild($node);
header("Content-type: text/xml");

$result = pg_query($sig_db, $consultasql);

while ($row = pg_fetch_assoc($result)) {
    $node = $dom->createElement("Punto");
    $newnode = $parnode->appendChild($node);
    $newnode->setAttribute("gid", $row['gid']);
    $newnode->setAttribute("municipio", $row['nombre']);
    $newnode->setAttribute("vector", $row['n_vector']);
    $newnode->setAttribute("latitud", $row['latitud']);
    $newnode->setAttribute("longitud", $row['longitud']);
    $newnode->setAttribute("fecha_caso", $row['fecha_caso']);
    $newnode->setAttribute("observaciones", $row['observaciones']);
}
echo $dom->saveXML();
?>
```

Fig 6. Código fuente para la generación del archivo XML

F. Integración del archivo GeoJson con Google Maps

La integración del archivo GeoJson con Google Maps se realiza de la siguiente manera:

Primero debe crearse el objeto tipo Google Maps

```
map = new google.maps.Map(document.getElementById('map-canvas'), {
    zoom: 2,
    center: new google.maps.LatLng(0,0),
    mapTypeId: google.maps.MapTypeId.HYBRID | });

var script = document.createElement('script');
```

Figura 7. Código fuente para la creación del objeto Google Maps

Luego, asignamos como fuente de datos para Google Maps, el archivo GeoJson generado, pasando los parámetros sobre los cuales debe realizarse la consulta.

```
script.src = "phpseajax/json.php?num=< echo $_REQUEST['num']?&vector=< echo $_REQUEST['vector']?&desde=< echo $_REQUEST['desde']?&hasta=< echo $_REQUEST['hasta']?&";
```

Fig 8. Código fuente para asociar GeoJson y Google Maps

Y por último viene la creación de las marcas para cada uno de los puntos de acuerdo a los registros devueltos por GeoJson.

```

window.eqfeed_callback = function(results)
{
    for (var i = 0; i < results.features.length; i++)
    {
        var coords = results.features[i].geometry.coordinates;
        if(!bounds) var bounds = new google.maps.LatLngBounds();
        var latlng = new google.maps.LatLng(coords[1], coords[0]);
        bounds.extend(latlng);
        var content = results.features[i].properties;
        //console.log(content);
        var marker = new google.maps.Marker(
        {
            position: latlng,
            icon: {
                path: google.maps.SymbolPath.CIRCLE,
                fillColor: content['fill'],
                strokeColor: content['fill'],
                strokeOpacity: 1.0,
                strokeWeight: 0.2,
                scale: 18 //pixels
            },
            title: content['n_vector'],
            map: map
        });

        var content = "<b>Enfermedad: </b>" + content['n_vector'] + "<br><b>Fecha del caso: </b>" + content['fec']
        + content['latitud'] + ", " + content['longitud'] + "<br><b>observaciones: </b>" + content['observaciones']
    }
}

```

Fig 9. Código fuente para la asignación de marcas en Google Maps

G. Mapa en línea con la información consultada

El sistema está en capacidad de generar una consulta y por medio de Google Maps representar los resultados en un mapa, insertando marcas con su respectiva convención que las relaciona directamente con el tipo de vector registrado.

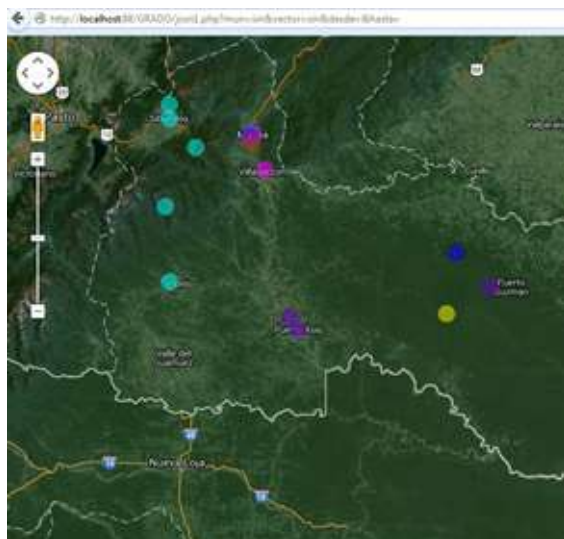


Fig 1. Presentación de resultados en un mapa basado en Google Maps

H. Mapa en línea de densidad de casos

El sistema genera un mapa general en el cual se crean marcas que con la ayuda de gradientes representan la densidad de los casos presentados en un área determinada.

Esta opción solo está disponible para el caso de las consultas que previamente seleccionan un vector .

Para lograr ese efecto se utiliza las librerías de visualización de la api de Google Maps, invocándola como se indica a continuación:

```
<script src="https://maps.googleapis.com/maps/api/js?v=3.exp&signed_in=true&libraries=visualization"></script>
```

Figura 2. Código para cargar librería API de Google

Luego se cargan todas las coordenadas resultantes de la consulta

```

var map, pointarray, heatmap;

var loader = dhtmlx.ajax.postSync("phpsajax/coordenadas.php?mun=? echo $_REQUEST['desde']?>&hasta=? echo $_REQUEST['hasta']?>");
coordendas_reg = loader.xmlDoc.responseText.split("...");

var listaCoordenadas = new Array();

for (i in coordendas_reg)
{
    coord_int = coordendas_reg[i].split(",");
    listaCoordenadas[i] = new google.maps.LatLng(coord_int[1], coord_int[0]);
}

```

Fig 3. Código para cargar coordenadas con Google API

Por último se asignan los puntos a la capa especificando que la visualización de los mismos debe ser con opciones de densidad.

```

var pointArray = new google.maps.MVCArray(listaCoordenadas);
heatmap = new google.maps.visualization.HeatmapLayer({
  data: pointArray
});

heatmap.setMap(map);
changeRadius();

```

Fig 4. Asignación de puntos a una capa con Google API

V. CONCLUSIONES

La representación de información a través de mapas puede brindar ideas más claras respecto a los datos que se consultan, para nuestro caso de estudio puede utilizarse para la toma de decisiones respecto a posibles focos de enfermedades ocurridas en uno o varios Municipios y por un determinado periodo.

Las pruebas de funcionamiento del sistema se efectuaron utilizando el servicio de internet domiciliario, con una velocidad de descarga de 1 Mb y de subida de datos de 512 Kb, obteniendo resultados satisfactorios en cuanto a tiempo de respuesta del servidor, lo que demuestra que si se logró la meta de contar con una herramienta eficiente, robusta y liviana.

Existe gran variedad de software libre que puede utilizarse para el desarrollo de herramientas web o de escritorio enfocadas al registro, análisis y procesamiento de datos espaciales y su posterior representación en mapas almacenados de forma local o a través de un servicio como Google Maps.

Relacionar la información estadística con el área geográfica a la que aplican los datos, facilita su comprensión y asegura en gran medida que el usuario los convierta en conocimiento.

La representación cartográfica o espacial es de gran apoyo para la Epidemiología, ya que ayuda a ubicar los principales focos de problemas en salud y de paso brinda mecanismos de información para que nuestros gobernantes tomen decisiones acertadas que beneficien a la población.

La implementación de un sistema de información igual o mejor al que se obtuvo como resultado de este trabajo, permitirá a cualquier entidad de la salud lo siguiente:

- Determinar la situación de salud en un área determinada
- La generación y análisis de hipótesis de investigación
- Identificación de grupos de alto riesgo a la salud
- Planeación y Programación de actividades
- Monitoreo y evaluación de intervenciones.

RECONOCIMIENTOS

Especiales agradecimientos al Ingeniero JUAN CARLOS GARCIA OJEDA Magister en Ciencias Computacionales, ya que su guía y observaciones fueron fundamentales para lograr el desarrollo de la herramienta expuesta en este artículo.

REFERENCIAS

- [1] NIETO MESA, Marco Oscar. (2010). Guía para elaborar el marco metodológico de un proyecto de investigación. Disponible en <http://es.scribd.com/doc/39980319/GUIA-PARA-ELABORAR-EL-MARCO-METODOLOGICO-DE-UN-PROYECTO-DE-INVESTIGACION>
- [2] MORALES, Víctor. (1994). Planeamiento y análisis de investigaciones. El dorado Ediciones. Caracas. Venezuela.
- [3] J. López Quijano, Domine JavaScript, México: AlfaOmega, 2005.
- [4] A. Vaquero Sanchez y G. Quiroz Vieyra, Diseño de Interfaz de Usuario Para Aplicaciones Windows, Madrid: McGrawHill, 2000.
- [5] J. Senn A., Análisis y Diseño de Sistemas de Información, México: McGrawHill, 1992.
- [6] SAMAJA J. 1993, *Epistemología y Metodología*, EUDEBA, Bs. Aires
- [7] SAMPIERI R. Y OTROS 1991, *Metodología de la investigación*, MC GRAW HILL, México
- [8] L. Herrera, Desarrollo de Aplicaciones Web con Php, Santo Domingo : V Centenario, 2011.
- [9] M. Castaño y M. g. Piattini Velthuis, Concepción y Diseño de Bases de Datos del Modelo E/R, Wilmington: Ra-ma, 1993.