



## MAPES DINÀMICS ON-LINE: BASES DE DADES RELACIONALS, PERL I SVG

**Alexandre Nobajas i Ganau**

Edinburgh Earth Observatory, Intitute of Geography. University of Edinburgh

### Resum

La cartografia distribuïda en xarxa ha anat creixent en complexitat i accessibilitat amb el pas del temps, fins que en els darrers anys la irrupció dels gegants de l'Internet (Google, Microsoft o Yahoo!) ha ocasionat una eclosió de la cartografia *on-line* que ha arribat al públic en general. Aquesta popularització ha fet que es demandi cada cop més informació geogràfica distribuïda, però de vegades els professionals que l'han de subministrar no tenen la formació tècnica suficient per a satisfer els requeriments, pel que cal idear maneres innovadores i senzilles per a crear aquests mapes. El que es proposa és una manera de crear mapes en xarxa que canviïn quan la base de dades sigui modificada, i per fer-ho es recorrerà al llenguatge de programació Perl i a l'estàndard SVG.

Paraules clau: Cartografia distribuïda, bases de dades relacionals, Perl, SVG.

### Abstract

Distributed cartography has grown in complexity and accessibility together with the Internet, but it has not been until recent times, with the irruption of the Internet giants – i.e. Google, Microsoft or Yahoo! – that internet mapping has become broadly known by the public. This quick popular acceptance has created a big demand for on-line maps, but sometimes the professionals who have to supply them do not have the required technical skills to create them, so innovative and simple solutions have to be created to help them. This paper explores a way to create dynamic maps which change each time the database is updated and involves the use of relational databases, Perl and SVG.

Key words: Distributed cartography, relational databases, Perl, SVG

### Resumen

La cartografía distribuida en red ha crecido en complejidad y accesibilidad con el paso del tiempo, hasta que en los últimos tiempos la irrupción de los gigantes de Internet (Google, Microsoft o Yahoo!) ha ocasionado una eclosión de la cartografía *on-line* que ha alcanzado al público en general. Esta popularización ha ocasionado que cada vez se demande más información geográfica distribuida, pero a veces los profesionales que la han de suministrar no tienen la formación técnica suficiente para satisfacer los requerimientos, por lo que se deben idear maneras innovadoras y sencillas para crear estos mapas. Lo que se propone es una manera de crear mapas en red que cambien cuando la base de datos sea modificada, y para hacerlo se usará el lenguaje de programación Perl y el estándar SVG.

Palabras clave: Cartografía distribuida, bases de datos relacionales, Perl, SVG

### 1. Introducció

Tot i que la cartografia en xarxa ja porta força temps entre nosaltres, no ha estat fins a temps recents que ha arribat amb força al gruix de la població i això ha obert tot un

camp per a que geògrafs, cartògrafs, ambientòlegs i tothom que treballi en el territori se'n pugui aprofitar. L'ús de mapes a la xarxa cada cop està més estès i és utilitzat per més usuaris, cosa que està democratitzant la creació, us i modificació de mapes. L'entrada de Google al mercat de la cartografia digital i a la seva distribució, l'any 2005, va ser una revolució a nivell internacional que va suposar una popularització de l'ús de mapes interactius a tot el món. De fet, la capacitat de veure imatges aèries d'arreu del món amb una definició impensable fins fa poc va canviar la manera que tenim de pensar en la informació geogràfica. Tot i això, molts científics, empreses i usuaris tenen la necessitat de crear els seus mapes per a tal de satisfer les seves necessitats.

És per això que cal crear un sistema per a que els professionals de la cartografia en format paper o que no tinguin grans coneixements tècnics puguin distribuir els seus mapes amb facilitat per Internet i arribar a la major quantitat d'usuaris possible. La comunicació que es presenta vol explicar una manera senzilla, ràpida i barata de crear mapes dinàmics utilitzant sistemes de codi obert i estàndards sempre que sigui possible. L'objectiu és descriure de manera entenedora per als professionals neòfits en la cartografia digital un sistema que permet crear mapes dinàmics publicables a la xarxa, pensant en les possibilitats didàctiques que aquest sistema ofereix degut a la seva relativa senzillesa.

## 2. Breu història dels mapes a Internet

La necessitat de publicar mapes i distribuir-los per la xarxa ha existit des dels inicis del ciberespai. Amb el pas del temps diferents tècniques i solucions han anat sorgint, i han anat creixent cada cop més pel que fa a complexitat i funcionalitat (PETERSON, 2003). Aquestes tècniques es poden classificar en tres grans generacions:

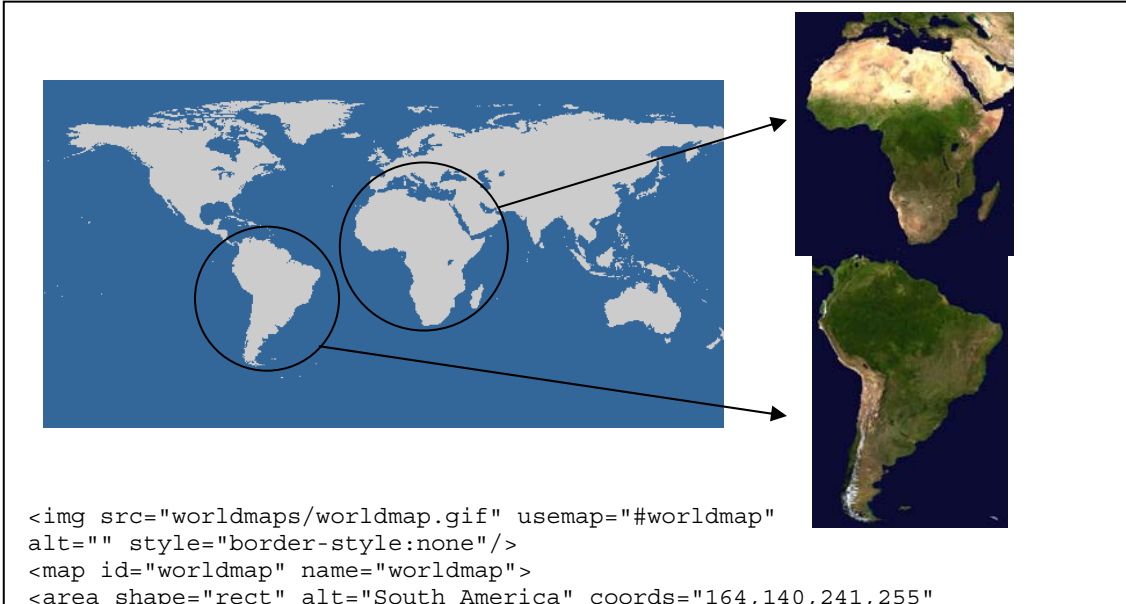
*1<sup>a</sup> Generació.* A principis dels anys noranta, quan la World Wide Web (WWW) va començar a guanyar popularitat i a ser usada per una massa cada cop més gran d'usuaris (CAILLAU; GILLES, 2000), van començar a aparèixer les primeres aplicacions de cartografia i geografia a la Xarxa. Aquestes aplicacions en un inici varen ser simplement imatges en diferents formats com JPEG, GIF, PNG que representaven mapes digitalitzats o dibuixats directament a l'ordinador, però sense cap mena d'interacció possible.

Poc després van sorgir els Imagemaps (Figura 1), que ja permetien un cert nivell d'interacció. Aquests mapes on-line permeten que al fer un clic simple amb el ratolí en una zona prèviament delimitada de la imatge que representa un mapa (hot-spot) enllaci amb una altra URL, però el nivell d'interacció és extremadament limitat i només són útils en aplicacions puntuals. A més a més, no deixen de ser imatges a les que se les hi defineixen regions clicables, pel que el sistema de coordenades sempre és referit a la imatge.

*2<sup>a</sup> Generació.* Tot i la millora que van representar respecte a les imatges estàtiques, els Imagemaps no van ser capaços de satisfer les necessitats dels web màsters. Això va ocasionar que sorgís un nou tipus d'Imagemaps, els Imagemaps Dinàmics. Respecte als Imagemaps originals van incorporar noves funcionalitats, com ara un rudimentari sistema de zoom o la millora del sistema de coordenades de la imatge, ja que aquestes són processades programàticament i s'envien a través de la URL. Un exemple de

coordenades enviades a través de la URL és [www.servidor.com/imagemap?164,387](http://www.servidor.com/imagemap?164,387), on les xifres són les coordenades enviades al servidor i que enllaçaran amb un altre Imagemap o pàgina web. Aquestes coordenades continuen sent referenciades respecte a la cantonada superior esquerra de la imatge i la unitat de mesura és el píxel, tot i que amb coneixements de programació es pot transformar el sistema de coordenades per a fer-lo geogràfic.

**Figura 1. Exemple d'Imagemap.**



```

<map id="worldmap" name="worldmap">
<area shape="rect" alt="South America" coords="164,140,241,255"
href="worldmaps/samerica.jpg"/>
<area shape="rect" alt="Africa" coords="265,93,387,215"
href="worldmaps/africa.jpg"/>
```

Al clicar o bé a Sud-Amèrica o bé a l'Àfrica al mapamundi de l'esquerra, s'enllaça a imatges de satèl·lit de cada continent respectivament. A sota es pot observar el codi HTML que converteix la imatge GIF en un Imagemap. Font: Elaboració pròpia.

3<sup>a</sup> Generació. No va ser fins a la tercera generació que es va iniciar la revolució tant pel que fa a la tecnologia com a la popularització del seu ús. Aquesta generació va començar l'any 1998 i ha anat evolucionant fins arribar a l'actualitat, on les possibilitats d'interacció, de personalització i de transmissió de dades han crescut d'una manera sense precedents. Aquesta generació ha introduït els "slippy maps" o mapes lliscants, que permeten fer zoom i desplaçar-se pel mapa d'una manera més ràpida i la seva personalització mitjançant les APIs (Aplicación Programming Interface) basades en JavaScript o .NET. Tot i aquest increment de sofisticació respecte les generacions anteriors i l'ús de tecnologia Ajax la base que la tecnologia continuen seguint els Imagemaps.

Un dels primers sistemes que va usar aquesta nova manera de mostrar mapes en xarxa va ser el servei MS TerraServer, establert l'any 1998, però va passar força desapercebut i no va ser fins l'entrada del cercador Google al món de la cartografia *on-line* que aquest tipus de serveis es van popularitzar. Aquesta irrupció es va produir l'any 2005 i va suposar que tota la població, i no només els especialistes, tinguessin accés a una base de dades cartogràfica de tot el món amb uns nivells de resolució i d'interacció

inimaginables fins al moment. Posteriorment, amb la publicació de la seva API, Google va permetre que es poguessin personalitzar els mapes d'una manera relativament fàcil i que apareguessin els “mashups”, aplicacions que combinen imatgeria de GoogleMaps amb d'altres fonts de dades externes. Una de les principals virtuts de GoogleMaps, entre d'altres, és que la interfície és intuïtiva i personalitzable, cosa que ha facilitat a la seva difusió.

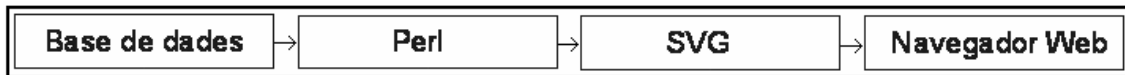
### 3. Mapes dinàmics

Els concepte mapa dinàmic es pot referir a dos tipus de serveis *on-line*. Per una banda hi ha els mapes dinàmics en el sentit que són interactius i permeten moure's a través del mapa mitjançant funcions com el lliscament o el zoom. En aquesta categoria hi entren per exemple els Imagemaps dinàmics o els mapes *on-line* de tercera generació com ara GoogleMaps o Terraserver. D'altra banda hi ha els mapes dinàmics en el sentit que es dibuixen des d'una base de dades i es mostren directament al navegador web, pel que qualsevol canvi en la base de dades és automàticament actualitzat al mapa. Això fa que el procés d'actualització de la informació geogràfica del mapa sigui instantani i que no calgui actualitzar la informació dues vegades, a la base de dades i al mapa, el que suposa un estalvi en temps i diners. Les aplicacions d'aquest tipus de mapes són innumerables, però qualsevol mapa on la informació representada canviï ràpidament se'n pot beneficiar. A més a més se'ls hi poden donar les mateixes funcionalitats interactives de zoom i lliscament que als altres mapes dinàmics, creant llavors uns mapes més complexos i interactius. A aquests mapes creats des de bases de dades se'ls pot fins i tot combinar amb serveis de tercera generació si l'aplicació desitjada ho requereix.

### 4. Mètode per a crear mapes dinàmics

Els mapes dinàmics generats a partir d'una base de dades es poden crear utilitzant múltiples tècniques i tecnologies. El problema és que moltes d'aquestes tècniques només estan l'abast d'experts en informàtica i d'alguns enginyers, pel que d'altres comunitats interessades en la cartografia però que manquen d'una formació tècnica suficient es poden veure privades de crear-los. És per això que cal un sistema que permeti crear mapes a partir de bases de dades que sigui senzill, didàctic i fàcilment modificable. D'aquesta manera es pot assolir un doble objectiu, crear mapes dinàmics d'una manera simple i tenir un mètode per a ensenyar els fonaments de les bases de dades, la programació i la cartografia distribuïda a la gent sense un vessant tècnic. A més a més, cal que la tecnologia emprada sigui el més accessible possible tant pel que fa al seu preu com a la seva distribució, tot i que això no hauria de menyscar la qualitat del producte final.

El mètode proposat combina l'ús de les bases de dades relacionals amb el llenguatge de dibuix vectorial escalable SVG, mentre que la comunicació entre ambdós elements es realitza mitjançant la utilització del llenguatge de programació Perl. Com es pot veure a la figura 2, l'esquema de funcionament del mètode proposat és directe, cosa en facilita la comprensió i que fa que el seu aprenentatge no suposi un gran repte per a qui no tingui grans coneixements d'informàtica o de programació.

**Figura 2. Esquema del funcionament del sistema de creació de mapes dinàmics proposat.**

Font: Elaboració pròpia.

#### 4.1 Base de dades

Com s'ha esmentat prèviament, els mapes dinàmics es nodreixen de la informació que els prové de les bases de dades. De bases de dades n'hi ha de diferents tipus depenent del model en que es basin. Cronològicament segons l'any d'aparició, les bases de dades es divideixen en jeràrquiques, en xarxa, relacionals o orientades a objectes, tot i que en l'actualitat majoritàriament s'usen les dues últimes. Tot i que les bases de dades orientades a objectes són les més modernes encara no han demostrat ser capaces de desplaçar a les bases de dades relacionals del mercat ni han aconseguit ser les preferides pel gruix d'usuaris. A més a més, el model relacional és el que s'usa més tant a l'acadèmia com a la indústria, cosa que fa que el programari disponible a la major part d'institucions principalment funcioni amb el model relacional. És doncs, a causa de la seva gran implantació arreu del món, a la seva popularitat, a la seva versatilitat i a la gran quantitat de programari que existeix per a gestionar-les que el mètode proposat per a crear mapes dinàmics inclou l'ús de bases de dades relacionals.

Al mercat existeixen diferents sistemes de gestió de bases de dades relacionals, però no tots tenen suport per a bases de dades espacials, que són les idònies per a crear mapes dinàmics. Dels que en tenen, els més populars i que més s'ajusten a les necessitats que presenta dibuixar mapes des d'una base de dades són Oracle, MySQL i PostgreSQL. Mentre que Oracle és software de propietari, tant MySQL com PostgreSQL són software que, tot i tenir llicències diferents, es poden obtenir de franc des de les seves respectives pàgines web. Tots tres sistemes de gestió tenen extensions espacials que permeten emmagatzemar dades de tipus espacial, com per exemple coordenades. Això fa que, a diferència d'altres mètodes per a representar mapes en xarxa descrits anteriorment, aquest mètode permeti respectar les coordenades geogràfiques, crear punts, línies, polilínies, polígons o multipolígons i fins i tot ajustar el sistema de projecció adequat a les necessitats del programador o de l'usuari.

El mètode, la versatilitat i les capacitats de creació de les bases de dades espacials difereix entre els tres sistemes, però el funcionament de base és similar i un cop es domina un sistema adaptar-se a un dels altres dos és relativament fàcil. L'elecció entre un fabricant i un altre depèn dels recursos econòmics disponibles, les necessitats o el familiaritzat que s'estigui amb un sistema en concret. En l'exemple de codi de l'epígraf 5 s'usa com a gestor de la base de dades Oracle perquè tot i ser un gestor de pagament és el més potent, és el que té un millor suport tècnic i és un sistema utilitzat per moltes institucions, tant públiques com privades, tot i que, com ja s'ha dit, canviar a un altre gestor no hauria de suposar cap problema.

De tota manera, el que sí comparteixen tots els gestors de bases de dades relacionals esmentats és en que usen com a codi de comunicació entre l'usuari i la base de dades el llenguatge SQL. Aquest és un llenguatge molt semblant a l'anglès, cosa que fa que sigui força fàcil d'aprendre i que a més és força versàtil. També té a favor que s'ha convertit en un estàndard *de facto* i la major part de programes de bases de dades el fan anar amb

versions més o menys modificades. Per tant, per a poder crear mapes dinàmics seguint aquest mètode cal tenir com a mínim nocions de SQL, cosa que fa que pugui servir com a mètode didàctic per ensenyar el funcionament de les bases de dades a les comunitats que no hi tinguin experiència.<sup>1</sup>

#### 4.2 Perl

Per a poder connectar la base de dades amb el sistema gràfic que acaba dibuixant el mapa cal un llenguatge de programació que permeti personalitzar els requeriments de l'usuari. El llenguatge triat per a fer-ho ha estat Perl perquè manca de moltes de les limitacions arbitràries que d'altres llenguatges tenen i per tant és un dels llenguatges de programació més fàcils d'aprendre. La gran flexibilitat que ofereix permet connectar tots els elements fàcilment amb uns nivells de rendiment més que acceptables. Per als programadors més experimentats Perl pot resultar una mica frustrant perquè prima la versatilitat, és a dir la facilitat d'ús, a la que el codi sigui elegant o minimalista. Tot i això com que el mètode proposat està enfocat als usuaris poc experimentats en la programació, Perl és el llenguatge ideal a causa de la seva potència i facilitat d'aprenentatge. A més a més, el llenguatge Perl és un sistema de codi obert, estable, multiplataforma, àmpliament usat per a crear aplicacions en l'entorn *web* a causa de la seva extensió CGI. El que el fa més adient però és que té el sistema de connexió amb bases de dades DBI, que facilita molt les coses a l'hora d'administrar-ne la connexió i l'administració i això fa que també en faciliti la selecció de dades.<sup>2</sup>

#### 4.3 SVG

El mètode triat per a dibuixar el mapa al navegador web és SVG. Aquest és un estàndard XML obert creat pel W3C per a representar gràfics vectorials, text i ràsters en dues dimensions. La seva gran compatibilitat permet integrar-los en pàgines *web*, tot i que en l'actualitat encara presenta algunes limitacions per a ser interpretat pels navegadors web. Així doncs dels tres navegadors més usats només un, l'Opera a partir de la versió 9 n'inclou suport nadiu. L'Internet Explorer exigeix l'ús d'un plug-in que es pot descarregar de franc des de la pàgina web d'Adobe, mentre que el Mozilla Firefox, tot i tenir-ne suport parcial des de la versió 1.5, no serà plenament compatible fins a la versió 3.0 (ara en període de proves i desenvolupament). Un dels principals avantatges d'aquest format és que al ser un sistema vectorial no es perd definició al fer-hi zoom (funció integrada dins els navegadors, pel que no cal programar-la) i sol ocupar menys espai i ser més ràpid de carregar que un arxiu d'espai ràster en brut, cosa que és important quan es tracta amb Internet i amples de banda limitats. Cal dir que tot i ser un tipus d'arxiu relativament nou i estar encara en període d'expansió, companyies punteres mundials com Adobe, Canon, Corel, IBM, Kodak, Nokia, Opera o Sun participen en el seu desenvolupament, cosa que garanteix la seva qualitat, i el fet de ser un estàndard en certifica la seva continuïtat a curt i mig termini. A més a més, com s'ha indicat abans, tot i el seu caràcter vectorial pot incorporar també gràfics ràster i text, fet que li atorga una gran versatilitat, que unida a la seva interactivitat i a la seva capacitat d'animar els gràfics el fan un mètode molt atractiu pel seu ús en cartografia distribuïda. Dit això, cal afegir que és un sistema força fàcil de programar i amb poca estona es poden obtenir resultats satisfactoris i de qualitat, com mostra la Figura 3.<sup>3</sup>

**Figura 3. Mapa dels municipis de l'Espanya Peninsular i les Balears en format SVG**

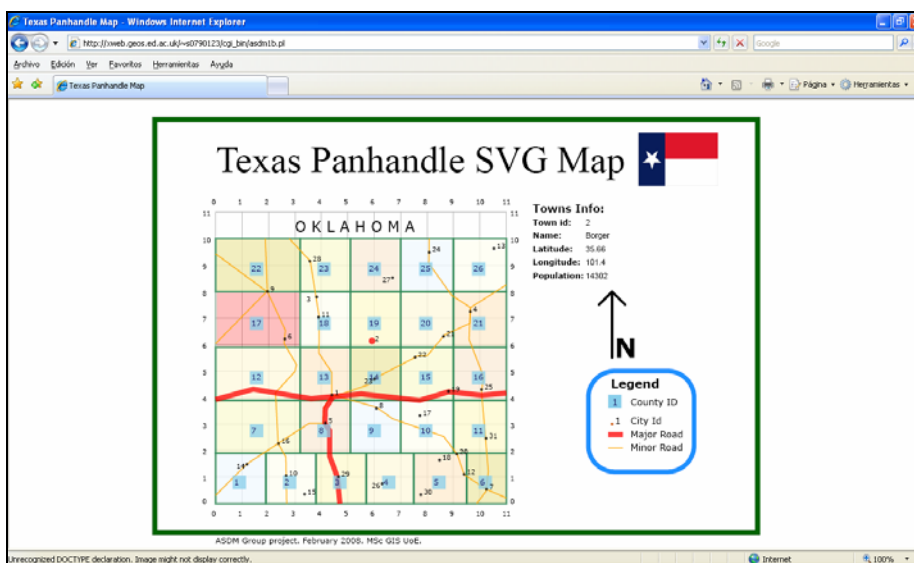


Font: Elaboració pròpia a partir de dades de l'Instituto Geográfico Nacional

## 5. Exemple de codi

Per a crear un mapa directament des de la base de dades amb el mètode que es descriurà a continuació no calen milers de línies de codi, amb tenir un esquelet bàsic a partir del qual afegir els elements desitjats n'hi ha prou. En aquest cas, les necessitats del servidor tampoc són difícils d'assolir, amb tenir-hi instal·lat Perl i la base de dades n'hi ha prou<sup>4</sup>. L'exemple mostrat a la Figura 4 ha estat realitzat seguint el mateix esquelet amb algunes ampliacions per augmentar-ne la interactivitat i donar més informació contextual. A continuació es comentarà cada part del codi per a explicar cada part per separat.

**Figura 4. Exemple de mapa de la zona nord de l'estat de Texas dibuixat amb directament des de la bases de dades amb el mètode descrit**



Font: Elaboració pròpia a partir de dades del U.S. Census Bureau



**a)** Comunicar que s'està treballant amb Perl i invocar els mòduls necessaris per a que el sistema funcioni. La primera línia s'ha de posar sempre per a indicar que s'està treballant amb Perl. Les altres línies de codi són per a invocar els mòduls necessaris, com el DBI, el que permet treballar amb bases de dades.

```
#!/usr/local/bin/perl
use lib qw ( /opt/perl);
use CGI::Carp qw(fatalsToBrowser);
use DBI;
```

**b)** Obrir una connexió amb la base de dades i indicar al navegador les versions que s'utilitzaran en el document

```
$ENV{'TNS_ADMIN'}='/usr/local/oracle_10/network/admin';
print "Content-Type: image/svg+xml\n\n";

my $dbh=DBI->connect ('dbi:Oracle:geogsp10','usuari','password')
    ||die "Database connection not made: $DBI::errstr";

print qq(<?xml version="1.0" standalone="no"?>);
print qq(<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" );
print qq("http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">);
```

**c)** Crear l'espai on serà dibuixat el SVG. Aquest pas és molt important ja que, usant un paral·lelisme, indica la mida la tela que s'usarà per a pintar el mapa. Si les dimensions es defineixen com a percentatge enlloc d'en píxels es pot ajustar la mida del mapa a la resolució de pantalla dels usuaris finals.

```
print qq(<svg width="100%" height="100%" viewBox="3 -45 50 50"
preserveAspectRatio="xMidYMid"
xmlns="http://www.w3.org/2000/svg"
version="1.1">>\n);
```

**d)** Posar el títol del mapa i transformar la tela per a que les coordenades comencin al marge inferior esquerra i així transformar les coordenades de la imatge i convertir-les en geogràfiques. A més a més es pot incrementar o reduir la mida de la imatge escalant-la. El "print qq" que hi ha davant de moltes línies de codi és per a que Perl llegeixi SVG sense problemes.

```
print qq(<title>Posar el títol aquí</title>);
print qq(<g transform="scale(2.95,-2.95)">\n);
```

**e)** Aquest és un mètode per a crear una graella per a les coordenades de la imatge.

```
my $grid=0;
while ($grid <= 11){
    print qq(<line x1="0" y1="$grid" x2="11" y2="$grid" stroke="black"
stroke-width="0.01" opacity="1" />\n);
    print qq(<line x1="$grid" y1="0" x2="$grid" y2="11" stroke="black"
stroke-width="0.01" opacity="1" />\n);
    my $text=$grid-0.15;
    print qq(<text x="$text" y="0.48" font-family="Verdana" font-size="0.25"
fill="black" transform="scale(1,-1)">$grid</text>\n);
    print qq(<text x="-0.5" y="- $text" font-family="Verdana" font-
size="0.25" fill="black" transform="scale(1,-1)">$grid</text>\n);
    print qq(<text x="$text" y="-11.30" font-family="Verdana" font-
size="0.25" fill="black" transform="scale(1,-1)">$grid</text>\n);
    print qq(<text x="11.15" y="- $text" font-family="Verdana" font-
size="0.25" fill="black" transform="scale(1,-1)">$grid</text>\n);
    $grid++; }
```

**f)** Per dibuixar, per exemple, divisions rectangulars (per exemple eixamples, comptats americans o camps de conreu) el mètode inferior seria l'adient. A la base de dades només caldria emmagatzemar els vèrtexs en columnes estàndard, pel que no és necessari usar les extensions espacials. És un "loop" que dibuixa tots els polígons que hi hagi a la base de dades i a més els pinta depenent de l'identificador desitjat





```
my $sth = $dbh->prepare("SELECT lowx,lowy,hix-lowx,hiy-lowy,identificador FROM
taula");
$sth->execute;

while (@data=$sth->fetchrow_array()){
print qq( <rect );
print qq(x="$data[0]" y="$data[1]" width="$data[2]" height="$data[3]");

#CHANGE COLOR DEPENDING ON THE COUNTY
if (($data[4] == "1") || ($data[4] == "3") || ($data[4] ==
"5") || ($data[4] == "7"))
{ $colour = "aliceblue"; }
elsif ($data[4] == "2" || ($data[4] == "4") || ($data[4] ==
"6") || ($data[4] == "8"))
{ $colour = "ivory"; }
else
{
$colour = "antiquewhite";}
```

**g)** Si es volen dibuixar formes més complexes cal utilitzar les extensions espacials de la base de dades, en aquest cas Oracle Spatial. L'exemple serveix per a dibuixar objectes lineals, com ara carreteres, línies elèctriques o rius, però adaptant mínimament el codi també es poden dibuixar punts, polígons i multipolígons. A l'igual que el punt anterior l'exemple està dissenyat per a funcionar com a "loop", de manera que, en aquest cas, totes les línies de la base de dades serien dibuixades automàticament i si una canviés el mapa canviaria conseqüentment.

```
#DRAW ROAD 1
my $sth = $dbh->prepare("select t.x, t.y, id_carretera from carreteres,
table(sdo_util.getvertices(carreteres.geom)) t ")
||die "Database connection not made: $DBI::errstr";
$sth->execute;
print qq( <polyline points=");
while (@data=$sth->fetchrow_array()){
print qq($data[0],$data[1] );}
print qq(" stroke-width="0.05");
print qq( fill="none" stroke="orange" stroke-dasharray="
opacity="0.75"/>\n);
```

**h)** Per acabar, cal tancar la transformació iniciada al punt "d" i tancar la declaració del SVG

```
print qq(</g>\n);
print qq(</svg>\n);
```

## 6. Conclusions

El sistema proposat per a construir mapes dinàmics que s'ajustin a les necessitats de cada usuari partir d'una base de dades es caracteritza per la seva senzillesa i potència. Tot i no ser un mètode que serveixi per competir amb els gegants de la cartografia *on-line*, com Google o Microsoft, la seva gran versatilitat i el fet de crear-lo des de zero permet que pugui complir amb moltes de les necessitats que altres sistemes no permeten satisfer. Al tenir un esquema de funcionament tant lineal i senzill, es pot utilitzar com a mètode didàctic per a que els estudiants aprenguin a usar bases de dades relacionals, SQL, programació en Perl i SVG i publicació dels mapes al web.

El model proposat és un esquelet per a crear mapes dinàmics i al mateix temps aprendre els fonaments dels sistemes d'informació geogràfica distribuïts en xarxa, però a partir del model descrit es poden crear aplicacions més complexes i fins i tot de qualitat suficient per a distribuir recerca científica o per a crear aplicacions comercials de qualitat per als més diversos usos. Per exemple, afegint unes quantes formes HTML o



mitjançant l'ús de JavaScript, per donar dos exemples, es pot personalitzar i incrementar de manera significativa qualsevol aplicació del model.

## Agraïments

L'autor vol agrair a la "Fundación Obra Social Caja España" el suport econòmic donat per a la realització de la recerca associada a la publicació de la present comunicació.

## Notes

- 1.- Per a tenir més informació sobre les bases de dades relacionals es recomana llegir (DOUGLAS, B; JOSHUA, D, 2001). Per a aprofundir en l'ús de Oracle (GREENWALD, R; STACKOWIAK, R; STERN, J, 2007) , en l'ús de MySQL (DYER, R, 2008) , en l'ús de PostgreSQL (WORSLEY, J; DRAKE, J, 2002) o accedir a la pàgina web oficial de cada gestor de bases de dades.
- 2.- Per a tenir més informació sobre Perl es recomana llegir (DESCARTES, A; BUNCE, T 2000), (FOY, B 2007) o (SIEVER, E; SPAINHOUR, S; PATWARDHAN, N 1998) o accedir a la web [www.perl.org/](http://www.perl.org/) o <http://www.perl.com/>.
- 3.- Per a tenir més informació sobre SVG es recomana llegir (EISENBERG, J, 2002) o accedir a la web [www.w3.org/Graphics/SVG/](http://www.w3.org/Graphics/SVG/), [www.carto.net](http://www.carto.net) o [www.adobe.com/svg/](http://www.adobe.com/svg/)
- 4.- També es podria utilitzar un servidor sense base de dades i usar una base de dades ubicada a l'equip local, però el rendiment no seria tant bo i a més caldria tenir sempre encès tant el servidor com l'equip local per a que el sistema funcionés.

## Bibliografia

- CAILLIAU, R; GILLIES, J (2000). *How the Web Was Born: The Story of the World Wide Web*. Oxford University Press, Oxford.
- DESCARTES, A; BUNCE, T (2000) *Programming the Perl DBI* O'Reilly, Sebastopol, California, USA
- DOUGLAS, B; JOSHUA, D (2001) *Object Storage Fact Books: Object DBMSs and Object-Relational Mapping*. Barry & Associates, Inc.
- DYER, R (2008) *MySQL in a Nutshell* O'Reilly, Sebastopol, California, USA
- EISENBERG, J (2002) *SVG Essentials* O'Reilly, Sebastopol, California, USA
- FOY, B (2007) *Mastering Perl* O'Reilly, Sebastopol, California, USA
- GENNICK, J (2006) *SQL Pocket Guide, Second Edition* O'Reilly, Sebastopol, California, USA
- GREENWALD, R; STACKOWIAK, R; STERN, J (2007) *Oracle Essentials, Fourth Edition* O'Reilly, Sebastopol, California, USA
- KRAAK, M-J; BROWN, A (2001): *Web Cartography – Developments and prospects*, Taylor & Francis, New York
- PETERSON, M (editor) (2003). *Maps and the Internet*. Elsevier Science, Amsterdam.
- SCHWARTZ, R; CHRISTIANSEN, T; WALL, L (1999) *Learning Perl* O'Reilly, Sebastopol, California, USA
- SIEVER, E; SPAINHOUR, S; PATWARDHAN, N (1998) *Perl in a Nutshell* O'Reilly, Sebastopol, California, USA
- SRINIVASAN, S (1997) *Advanced Perl Programming* O'Reilly, Sebastopol, California, USA
- WORSLEY, J; DRAKE, J (2002) *Practical PostgreSQL* O'Reilly, Sebastopol, California, USA