

Cartografía automática en Internet インターネット上の言語地図の自動作成

Antonio Ruiz Tinoco
アントニオ・ルイズ・ティノコ

言語地図は大変便利な研究道具ではあるが、その作成は大変時間がかかるものである。作成しても高価で、一般の研究者・学生に届かないこともよくある。言語地図が出版されてから、データの追加、削除、訂正などは大変困難である。また、高い出版の費用のために長年の調査の結果が発表されないで他の研究者の手に届かない。しかし、現在、インターネットでのプログラミング、データベースの発達のために、言語地図は容易に作成できるようになり、共同研究などにも大変に役立ち、研究の結果を発表する方法にもなる。ここでは、ApacheというWWWサーバ、PHPという使いやすいプログラミング言語、とMySQLというデータベース管理用プログラムを利用して具体的にどのようにすれば言語地図が作成できるかを試みて、実際にスペイン語の語彙の共同研究 (Varilex) で得られたデータを使い、その最初の結果を報告する。

0. Introducción

Un atlas lingüístico es el resultado de largos años de trabajo de campo de un investigador o casi siempre de un grupo de investigadores. Son de gran utilidad para la investigación de cualquier aspecto lingüístico de la variación, particularmente en el campo de la Dialectología. Sin embargo, suelen ser instrumentos voluminosos, casi siempre costosos y a veces de poca difusión. Desgraciadamente, bastantes trabajos de campo no han podido publicarse debido a los altos costos de publicación, o tienen una circulación reducida.

Afortunadamente, gracias al uso de las bases de datos relacionales (MySQL, PostgreSQL, etc.) , lenguajes de programación fáciles de aprender

(PHP4, Ruby, etc.) y otros programas que permiten confeccionar gráficos (GD Graphics Library, etc.) en la actualidad es posible la creación de sistemas automáticos para representar gráficamente a través de Internet el contenido de las bases de datos de los trabajos de campo sin necesidad de un conocimiento técnico demasiado especializado.

En este trabajo, queremos mostrar las posibilidades que nos ofrece Internet para la preparación de un sistema automático que reproduzca gráficamente sobre la marcha el resultado de las consultas a una base de datos. Como ejemplo, utilizaremos la base de datos de Varilex¹ de variación léxica del español, que hemos venido preparando desde hace algunos años. Los gráficos no se preparan de antemano y no están almacenados de forma estática para bajarlos, sino que se van creando sobre la marcha según las características de la consulta, es decir, son generados dinámicamente.

1. Entorno del sistema

Desde el punto de vista del usuario, el sistema debe ser multiplataforma, es decir, compatible con la mayoría de los sistemas operativos existentes, como son las diferentes versiones de Windows², Mac OS³, y las principales versiones de Unix⁴, sin olvidarnos de Linux⁵. Afortunadamente, el entorno que presentamos más adelante es compatible con estos sistemas operativos mencionados y solamente es necesario un navegador como Internet Explorer⁶ o Netscape⁷ para acceder a la base de datos y obtener todas las páginas necesarias del atlas en forma de gráfico creado automáticamente. Ambos navegadores se pueden obtener sin costo

1 Se puede ver información adicional en las páginas del proyecto en:

<http://varilex.call.sophia.ac.jp> y <http://gamp.c.u-tokyo.ac.jp/~ueda/varilex/>

2 Cf.: <http://www.microsoft.com/windows/default.asp>

3 Cf.: <http://www.apple.com>

4 Sistema operativo desarrollado originalmente por la empresa Bell Laboratories en 1969.

5 Sistema operativo similar a Unix, originalmente creado por Linus Torvalds. Para más información consultar <http://www.linux.org/>

6 Cf.: <http://windowsupdate.microsoft.com/?IE>

7 Cf.: <http://home.netscape.com/>

alguno.

Como hardware, solamente necesitamos un servidor de características normales. Nosotros hemos diseñado nuestro sistema utilizando uno de tipo PC en Linux, y creemos que es suficiente para nuestros objetivos. Aconsejamos que no se quede demasiado justo de memoria debido al uso continuado de gráficos que se van creando sobre la marcha.

Para el desarrollo del sistema, hacen falta varias herramientas que presentamos a continuación y que se pueden obtener en general por licencia GNU/GPL⁸ sin costo alguno para uso académico, con excepción del sistema operativo Windows⁹, que no es absolutamente necesario.

1.1. Sistema operativo

Todas las pruebas efectuadas hasta el momento se han realizado en Linux y parcialmente con las versiones 98 y 2000 de Windows. No hemos comprobado el funcionamiento en otros sistemas operativos, ya que ello no afecta la forma de uso final y nos consta que son compatibles por numerosos usuarios de estos sistemas que participan en grupos de discusión a través de Internet. En nuestro caso y por comodidad propia para aprovechar los recursos disponibles, nuestro servidor funciona con Linux, al que nos conectamos a través de telnet¹⁰ y FTP¹¹ para enviar los programas generalmente preparados en Windows. Veremos algunos detalles en los siguientes apartados.

⁸ Cf.: Las características del proyecto GNU de software libre se pueden consultar en: <http://www.gnu.org/>

⁹ Usamos Windows parcialmente ya que disponemos de este sistema operativo para otros usos y no queremos dedicar un ordenador adicional, pero no es absolutamente necesario.

¹⁰ Uno de los principales servicios de Internet. Básicamente se utiliza para establecer conexión con un ordenador remoto, como si estuviera en una red local.

¹¹ FTP (File Transfer Protocol), Protocolo de Transferencia de Archivos, sirve para transmitir archivos de todo tipo a través de la Red. Utilizamos el software libre FFFTP ver. 1.85, que se puede obtener en <http://www.vector.co.jp> porque permite fácilmente la transmisión de datos en japonés gracias a la función de cambio de códigos, lo cual nos resulta útil para otros proyectos. Existen muchos otros programas de libre distribución con características similares.

1.2. Servidor web

A nuestro juicio, el servidor web¹² más fiable en plataforma Linux es Apache¹³ y afortunadamente encontramos detallados manuales de administración como los de Charles Aulds (2000). Utilizamos la versión 1.3.22, ya que las versiones más adelantadas están en fase experimental. Existe una gran cantidad de información tanto bibliográfica como directamente en la red. También existen versiones para los principales sistemas operativos, incluido Windows. Además, tiene la ventaja de que los lenguajes de programación que presentamos en el siguiente párrafo se integran perfectamente en forma de módulos, lo que reduce el tiempo de respuesta del servidor no siendo necesario el uso de cgi-bin en absoluto.

Otro servidor web muy utilizado en entorno Windows es Xitami¹⁴.

1.3. Lenguaje de programación

Hay varias posibilidades para escoger un lenguaje de programación, como son PERL¹⁵, Ruby¹⁶ y otros más. Preferimos el lenguaje PHP4¹⁷, lenguaje de programación del lado del servidor¹⁸, porque creemos que, además de ser multiplataforma se integra mejor al servidor web, y los *scripts* se pueden incluir sin mayor problema dentro del protocolo HTML, por lo que resulta relativamente fácil preparar documentos interactivos. Además, originalmente es un lenguaje diseñado para ser utilizado en documentos hipertexto, fácil de aprender y de revisar (*debugging*). La comunidad de usuarios cada vez es mayor y siempre está dispuesta a ayudar desinteresadamente en los posibles dudas de uso. También hay que tener en

12 También llamado servidor WWW. Integra la mayoría de los servicios de Internet con la ayuda del protocolo HTML, y nos da acceso a los documentos hipertexto. También permite aplicaciones interactivas a través de la red. El uso más frecuente es a través de un navegador.

13 Cf.: <http://httpd.apache.org>

14 Cf.: <http://www.xitami.com/>

15 Cf.: <http://www.perl.com>

16 Cf.: <http://www.ruby-lang.org>

17 Cf.: <http://www.php.net>

18 Al ser un lenguaje del lado del servidor, el usuario no tiene que instalar ni descargar ningún programa aparte para utilizarlo.

cuenta que la velocidad de respuesta es tal vez la más alta entre los lenguajes de programación tipo *script*, sobre todo cuando se combina con *Zend Optimizer* y *Zend Accelerator*¹⁹. PHP funciona en la realidad como un módulo de Apache y está muy bien documentado en introducciones con CD-ROM incluido como las de Julie C. Meloni (2000), textos más avanzados como Gerken, T. & Rastchiller, T. (2000) y hasta gruesos manuales como Converse T. & Park J. (2000).

Otra gran ventaja adicional es la capacidad de conexión con casi todas las bases de datos disponibles.

1.4. Base de datos

Creemos que un tipo estándar de base de datos como SQL²⁰ es el más adecuado para nuestro propósito. Y para gestionar la base de datos que presentaremos más adelante es necesario instalar un programa de gestión de base de datos. Los programas de distribución libre más conocidos son MySQL²¹ y PostgreSQL²². Preferimos el uso de MySQL por su velocidad y robustez. Hay bastante bibliografía sobre el uso de MySQL, sistema usado por nosotros, como Judith Bowman et al. (1996), Paul Dubois et al. (1999), Randy J. Yarger et al. (1999) y muchos más aparecidos recientemente.

1.5. Librería gráfica

Para la creación dinámica de gráficos, necesitamos una librería²³ para generarlos. Hay varios tipos de librería y creemos que la más adecuada para nuestro objetivo es la llamada *GD Graphics Library*²⁴, creada por Thomas Boutell, que permite la creación dinámica de imágenes en formato

19 Cf.: <http://www.zend.com/zend/products.php>

20 Structured Query Language

21 Cf.: <http://www.mysql.com>

22 Cf.: <http://www.postgresql.org/>

23 Una librería es un conjunto de funciones de un lenguaje de programación. Con el uso de las funciones que se incluyen en las librerías podemos crear algoritmos ya comprobados y no partir de cero en la creación de los programas.

24 Cf.: <http://www.boutell.com/gd/>

JPEG, PNG y WBMP. Permite dibujar líneas, arcos, introducir texto, usar colores e imágenes ya preparadas. Debido a problemas de derechos de autor no soporta el formato GIF, pero no es absolutamente necesario, ya que tanto JPEG como PNG se utilizan de forma general en casi todos los navegadores web.

1.6. Otros

Para la instalación de los programas mencionados puede ser de utilidad el uso de alguna de las siguientes herramientas.

Instaladores como PHPDEV²⁵ y PHPTRIAD²⁶: Utilidades para la instalación casi automática en el sistema operativo Windows de los programas Apache, PHP, MySQL y otros programas adicionales.

También, tanto Nusphere²⁷ como Abriasoftware²⁸ han publicado programas que permiten la instalación que necesitamos en varios sistemas operativos.

Es conveniente la utilización de un editor tanto de texto normal como de HTML para la preparación de la página interfaz que utilizaremos para conectar con la base de datos. Se pueden encontrar muchos programas de distribución libre en Internet. En nuestro caso utilizamos Kajero²⁹ ya que además de los caracteres propios del español, también se pueden utilizar los diferentes códigos del japonés, que solemos utilizar en otros proyectos.

Evidentemente necesitamos un programa de creación o manejo de gráficos. No es necesario ningún programa complejo ya que el único uso será el de la preparación de los gráficos (mapas blancos de las regiones utilizadas) y poder medir en *pixels*³⁰ la situación de los puntos dentro del mapa, como veremos más adelante. En nuestro caso, usamos Paint Shop

25 Cf.: <http://www.firepages.com.au/dev4.htm>

26 Cf.: <http://www.phpgeek.com>

27 Cf.: <http://www.nusphere.com>

28 Cf.: <http://abriasoftware.com>

29 Cf.: <http://www.vector.co.jp/soft/win95/writing/se066723.html>

30 *Pixel* es una palabra derivada de *picture element* y es la unidad básica de programación de los puntos a color de los gráficos.

Pro³¹, pero hay otros muchos programas igualmente útiles para nuestro objetivo.

Finalmente, la estructura del servidor se podría representar como en la Fig. 1.

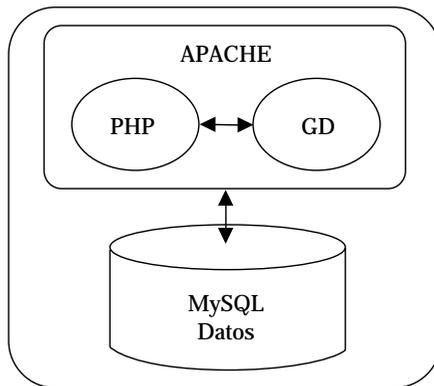


Fig. 1

2. Estructura de la base de datos

Reproducimos a continuación la estructura básica de una parte de la base de datos de variación léxica, Varilex, para ilustrar el funcionamiento del sistema de cartografía. La Fig.2 muestra la tabla *contestaciones* tal como se puede ver en la página del proyecto Varilex, cuya estructura es la siguiente³²:

31 Cf.: <http://www.jasc.com/products/psp/>

32 La estructura de la base de datos Varilex está actualmente en proceso de rediseño, pero ello no afecta básicamente al sistema que presentamos aquí. Para mayor información, consultar la serie de informes anuales publicados por el equipo Varilex.

Base de Datos varilex - tabla contestaciones

Mostrando campos 0 - 30 (45148 total)

SQL-query:
SELECT * FROM contestaciones LIMIT 0, 30

orden	codciu	país	ciudad	informante	forma	respuesta
A001	2	ES	SCO	e155	1	americana
A001	2	ES	SCO	e156	1	americana
A001	2	ES	SCO	e161	1	americana

Fig. 2 tabla contestaciones

orden: indica el orden de los conceptos encuestados. Así A significa que es la primera encuesta de la serie y 001 significa que es el primer concepto tratado en dicha encuesta.

codciu: número de código de la ciudad donde se realizó la encuesta

país: código de dos letras del país

ciudad: código de tres letras de la ciudad

informante: código de referencia del informante

forma: código identificador de formas

respuesta: respuesta obtenida en la encuesta a esa pregunta

Como podemos ver por los campos de la tabla contestaciones, están recogidas las respuestas obtenidas a cada concepto de la encuesta, así como la ciudad y la referencia al informante. Por supuesto, mediante una interfaz como la existente en la actualidad se puede consultar la base y obtener los datos. Sin embargo, si queremos mostrar los resultados en un gráfico necesitamos dotar al sistema con algunos datos simples adicionales, como es la posición de cada ciudad expresada en pixels. Podemos ver algunos datos parciales en la siguiente tabla de la Fig. 2

CIUDAD	X	Y
HAB	445	340
MED	505	505
BOG	540	540
CAR	630	480
MER	565	475
TAC	550	475
VNC	595	450

Fig. 2

Los datos de la tabla significan: HAB (La Habana) está situado en el pixel 445 del eje X y el 340 del eje Y del mapa blanco utilizado como fondo. Así sucesivamente con MED (Medellín), BOG (Bogotá), CAR (Caracas), etc.

Naturalmente, si queremos crear diferentes tipos de atlas donde se representen todas las ciudades encuestadas del mundo hispánico, zonas dialectales, o un país concreto, deberemos preparar otros tantos mapas en blanco que sirvan de fondo. También adjudicaremos unas ordenadas a cada punto. En la fase actual, hemos preparado un mapa que abarca solamente el continente americano. Más adelante preparemos otros mapas con diferentes características que se van creando de forma análoga según el mismo método.

3. Algunos algoritmos básicos para la creación dinámica de gráficos

A continuación veremos algunas partes del algoritmo utilizado en la actualidad con el objeto de mostrar su facilidad de uso y flexibilidad para crear diferentes figuras geométricas y añadir notas sobre una imagen de fondo. También veremos algunos algoritmos utilizados en la interfaz con la base de datos.

Anteriormente hemos mencionado que el lenguaje PHP se ha diseñado para usarlo en los documentos que utilizan el protocolo HTML. Para señalar que una parte del documento es lenguaje PHP solamente es necesario incluirlo dentro de `<?php y ?>` como en los ejemplos siguientes.

Para empezar, hay que transmitirle al servidor que el contenido que queremos que muestre es una imagen, y lo expresamos de la siguiente manera:

```
<?php  
header ("Content-type: image/png");  
?>
```

En este caso estamos indicando que el formato de la imagen será png. También lo podemos cambiar por el formato jpeg.

A continuación tenemos que definir en unidades pixel el tamaño de la imagen que queremos reproducir. Por ejemplo:

```
$imagen = imagecreate(150,320);
```

\$imagen es el nombre de la variable que contiene el espacio dedicado a la imagen, 150 es el ancho y 320 es la altura de la imagen. Por supuesto, podemos cambiar estos parámetros según nos convenga. En nuestro caso, no necesitamos preparar un campo vacío sobre el que mostrar una imagen sino que usamos como fondo una imagen preparada de un mapa blanco, que como prueba llamaremos "hispanoamerica.png" y que contiene un mapa blanco en formato png como se puede ver en tamaño reducido en la fig. 3. Para ello, en vez de la fórmula anterior, nos resulta más conveniente definir la imagen de fondo con la siguiente fórmula:

```
$imagen = imagecreatefrompng("hispanoamerica.png");
```



Fig. 3

El siguiente paso será declarar los colores que vamos a utilizar dentro de variables diferentes, como en el ejemplo siguiente:

```
$rojo = imagecolorallocate($imagen, 255,0,0);  
$azul = imagecolorallocate($imagen, 0,0,255);  
$verde = imagecolorallocate($imagen, 0,255,0);  
$negro = imagecolorallocate($imagen, 0,0,0);
```

De esta manera, ya disponemos de una imagen de fondo y unos colores que podemos usar. Si ahora queremos dibujar una línea azul desde el punto (5,20) al (300, 340), podemos hacerlo fácilmente de la siguiente forma:

```
imageline($imagen, 5, 20, 300, 340, $azul);
```

Para dibujar un rectángulo rojo que vaya desde el punto (5,20) al

(300, 340), utilizamos la función correspondiente:

```
imagefilledrectangle($Imagen, 5, 20, 300, 340, $rojo);
```

Así, en la librería de gráficos de GD hay preparadas funciones para el dibujo de todo tipo de formas geométricas como líneas, rectángulos, círculos, arcos, todo tipo de polígonos, etc. Solamente es necesario utilizar la función adecuada y sustituir los parámetros necesarios por sus valores concretos.

También podemos escribir notas sobre la imagen. Por ejemplo, si definimos:

```
ImageTTFText($Imagen, 16, 0, 620, 20, $azul, "arial.ttf", "Pruebas  
proyecto VARILEX 2001");  
ImageTTFText($Imagen, 16, 0, 680, 40, $azul, "arial.ttf", "Mapa  
de");  
ImageTTFText($Imagen, 16, 0, 755, 40, $rojo, "arial.ttf",  
"camiseta");
```

obtendremos una nota escrita sobre el fondo con la fuente "arial" como se puede ver a continuación en la fig. 4:



Pruebas proyecto VARILEX 2001
Mapa de camiseta

Fig. 4

4. Reflejar el contenido de la base de datos

Hasta ahora hemos visto ejemplos de cómo mostrar imágenes o escribir algunas palabras que iban incluidas previamente en el algoritmo. En este apartado vamos a ver cómo se puede hacer de forma dinámica. Es decir,

lo que necesitamos es obtener el contenido de las palabras o de los lugares donde se usan a través de una interfaz con la base de datos, y de manera automática introducir tales parámetros en las fórmulas mostradas anteriormente para obtener automáticamente el atlas. Para ello, hacemos un uso intensivo de las variables. Por ejemplo, para obtener el gráfico de la fig. 4 hemos escrito dentro del algoritmo "camiseta". El usuario del sistema no tiene acceso al algoritmo y lo que quiere es simplemente introducir la palabra en cuestión en alguna ventanilla preparada en la interfaz. Y eso es lo que hemos hecho exactamente mediante un simple mecanismo de transmisión de datos a través del método POST. Vamos a ver a continuación una parte simplificada de la interfaz preparada para que el usuario introduzca una variable, que en este caso llamaremos \$keyword, y que posteriormente se pasa al programa varilex.php tal como se indica.

```
<form method=post action="varilex.php">
  Experimento de consulta de la base de datos VARILEX <br>
  Introducir una palabra:<br>
  <input type=text name=keyword size=39 maxlength=25>
  <input type=submit VALUE=" BUSCAR">
```

De esta manera, en el programa varilex.php podremos cambiar la línea donde decía:

```
ImageTTFText($imagen, 16, 0, 755, 40, $rojo, "arial.ttf",
"camiseta");
```

por otra que contenga en su lugar el nombre de la variable y que nos servirá para cuantas veces se consulte la base de datos:

```
ImageTTFText($imagen, 16, 0, 755, 40, $rojo, "arial.ttf",
"$keyword");
```

A continuación necesitamos preparar una conexión con la base de datos y preguntarle en qué ciudades se utiliza tal forma léxica, es decir, la introducida en la ventana preparada en la interfaz y que se entregará al programa como la variable \$keyword. Tal pregunta en la versión MySQL del lenguaje SQL, con la palabra “camiseta” se podrá hacer de la siguiente manera:

```
SELECT distinct respuesta, ciudades.ciudad, X, Y
FROM contestaciones, ciudades
WHERE contestaciones.ciudad = ciudades.ciudad
AND respuesta LIKE '%camiseta%';
```

Si hacemos esta consulta a través de una interfaz tradicional como la que hay preparada también en la página del proyecto, podremos obtener el siguiente resultado de la fig. 5. Dentro del programa, hay que modificar la forma de hacer la pregunta para que forme una variable que irá tomando diferentes valores según las respuestas obtenidas de la base de datos, como mostramos a continuación:

```
$result = mysql_query ("SELECT DISTINCT respuesta,
ciudades.ciudad, país, X, Y FROM contestaciones, ciudades
WHERE contestaciones.ciudad = ciudades.ciudad AND respuesta
LIKE '$keyword'");
```

respuesta	ciudad	X	Y
camiseta	STI	562	387
camiseta	NYK	475	140
camiseta	NOR	353	275
camiseta	MON	249	319
camiseta	AGS	239	358
camiseta	MEX	260	390
camiseta	MED	505	505
camiseta	TAC	550	475
camiseta	PAZ	610	737
camiseta	MTV	741	921

Fig. 5

De esta forma, al obtener todos los parámetros como valores de las variables indicadas, podemos dibujar las siguientes imágenes donde no hemos introducido constantes sino las variables³³ mismas. Al mismo tiempo, aprovechamos para anotar al lado del cuadrado rojo tanto el nombre del país como el de la ciudad en cuestión, separados por un guión tal como se puede observar en el algoritmo.

```
while ( $row = mysql_fetch_array($result) )
{
    imagefilledrectangle($imagen, "$row[X]", "$row[Y]", "$row[X]" +
    $Sancho, "$row[Y]" + $Sancho, $rojo);
    imagestring($imagen, 5, "$row[X]" + $Sancho*1.5, "$row[Y]",
    "$row[páís]" . "-" . "$row[ciudad]", $azul);
}
```

Mediante este método de hacer preguntas a la base de datos, podremos almacenar en la variable \$row los datos pedidos y después

³³ A la variable Sancho, que indicaría el ancho del cuadrado que queremos dibujar, le damos previamente un valor arbitrario, simplemente definiendo de antemano Sancho = 8;

utilizarlos por partes. Así $Srow[X]$ es el valor obtenido para X, $Srow[Y]$ es el obtenido para Y, " $Srow[X]$ " + $\$ancho$, es el valor de X al que hemos añadido el valor de $\$ancho$, es decir 8 pixels, que es el ancho del cuadrado, $Srow[Y]$ + $\$ancho$, nos da el valor de Y para el otro extremo del cuadrado rojo en forma análoga y finalmente $Srow[pais]$ nos da el código del país y $Srow[ciudad]$ el de la ciudad. El resultado de una parte del gráfico obtenido, que se puede ver en el navegador y guardar si fuera necesario, es el que se muestra en la fig. 6.



Fig. 6

5. Conclusiones

Cada vez es más frecuente el uso en Internet de técnicas de programación de parte del servidor en conexión con bases de datos y generación dinámica de imágenes, los programas son más accesibles y aumenta la información disponible sobre estas técnicas. Por otra parte, la confección tradicional de atlas lingüísticos es un proceso lento y laborioso y el resultado no siempre llega a la mayoría de los estudiosos ni de los estudiantes, excepto en contadas excepciones. Las técnicas antes citadas, aunque necesitan ser aprendidas, nos proporcionan un medio muy eficaz no solamente de ordenar los datos como con cualquier otro campo de estudio, sino también la posibilidad de coordinar la investigación independientemente

del lugar de residencia de cada uno de los miembros del grupo de investigación. Además de poder poner a disposición de la comunidad científica los resultados de las investigaciones en menor tiempo que por la publicación en papel, es más fácil reeditar y reordenar los datos con puntos de vista diferentes, mediante la simple confección de una nueva interfaz o mediante preguntas diferentes a las bases de datos compartidas.

En esta ocasión hemos presentado las posibilidades que nos ofrecen estas técnicas y en un futuro muy cercano esperamos desarrollar un sistema completo de creación automática de atlas lingüísticos para la base de datos del proyecto Varilex.

6. Referencias

- Aulds, Charles. 2000. *Linux Apache Web Server Administration (Linux Library)*, Sybex.
- Bowman, Judith S.; Emerson, Sandra L; Darnovsky, Marcy: *The Practical Sql Handbook : Using Structured Query Language*, Addison-Wesley Pub, 1996
- Converse, Tim; Park, Joyce. 2000. *PHP4 Bible*. IDG Books Worldwide.
- Dubois, Paul; Widenius Monty: *MySQL*, New Riders Publishing, 1999
- Equipo Varilex (Coord. Ueda, Hiroto; Takagaki, Toshihiro; Ruiz Tinoco, Antonio). *VARILEX, Variación léxica del español en el mundo*. Vols. 1-9.
- Gerken, Till and Rastchiller, Tobias. 2000. *Web Application Development with PHP*. New Riders.
- Meloni, Julie C. 2000. *PHP Fast & Easy Web Development*, Prima Publishing.
- Yarger, Randy Jay; Reese, George & King, Tim: *MySQL & mSQL*, O'Reilly & Associates, 1999.

