

Anexo III - Creación de imágenes dinámicas

Puede ser presentada en la página web sin necesidad de ser almacenada previamente en el servidor, su contenido que puede ser modificado en cada instante (gráficos con valores actuales de *base de datos*). Se utiliza formato PNG y JPG. Hay que comporbar configuración del php.ini. Es un script con las instrucciones para la creacion de la imagen.

Visualizar una imagen dinámica desde una página web: Se invoca el fichero que contiene el script desde la etiqueta clásica de inserción de imágenes de HTML

```

```

donde imgxx.php será el nombre del script que genera la imagen.

La primera etiqueta del script ha de ser (según sea jpeg o png):

```
Header("Content-type: image/jpeg") o Header("Content-type: image/png")
```

Después hay que:

- **Crear la imagen:** `$nom = imagecreate(anc,al)` crea una imagen del tamaño indicado en *anc* y *al* (pixels) se recoge `$nom`, a partir de ese instante dibujamos sobre ella (es el lienzo).(si no doy color en jpeg es por defecto negro, en png hay que especificarlo sino no se ve).
- Dibujarla:
- **Enviarla al navegador** para que pueda ser visualizada:
 - o Si creamos un JPG: `Imagejpeg($nom)`
 - o Si creamos un PNG: `Imagepng($nom)`
- **Borrarla** (conveniente): `Imagedestroy($nom)`

1.1 Dibujar la imagen

- **Creando colores:** Se puede crear y almacenar un color mediante la siguiente función:

```
$color=imagecolorallocate ($nom,R,G,B).
```

R, G y B enteros comprendidos entre 0 y 255 *intensidad* de las luces roja, verde y azul.

- **Aplicar colores de fondo**

```
Imagefill($nom,x,y,$col)
```

`$nom`: *variable* que contiene la imagen, *x* e *y* son las coordenadas del punto de la imagen a partir del cual se aplica el relleno y `$col` el color.

- **Rectángulos sin relleno**

```
imagerectangle($nom, x0, y0, x1, y1, $col)
```

x₀, *y₀* coordenadas vértice superior izquierdo y *x₁*, *y₁* inferior derecho. (0,0) es la esquina superior izquierda de la imagen.

- **Rectángulos con relleno**, dibujo el rectángulo y luego lo relleno con:

```
imagefilledrectangle( $nom, x0, y0, x1, y1, $col)
```

- **Polígonos con relleno**

Primero hay que crear un array con las coordenadas de losvértices.

```
$v=array(x0, y0, x1, y1,... xn, yn )
```

Luego se usa la función:

```
imagefilledpolygon($nom, $vert, nº vert , $col)
```

\$vert es el array, nº vert es el número de vértices y \$col es el color de relleno.

- **Polígonos sin relleno**

Su funcionamiento es idéntico al anterior, el nombre de la función es:

```
imagepolygon($nom, $vert, nº vert , $col)
```

- **Elipses, circunferencias y arcos**

```
imagearc($nom, Xc, Yc , a, b, Gi, Gf, $col)
```

Xc e Yc *coordenadas del centro* de la elipse, a *longitud eje horizontal de la elipse*, b *longitud del eje vertical*, Gi es el *punto inicial del arco (grados sexagesimales, en el sentido agujas reloj)*, Gf es el *punto final del arco*.

- **Dibujando sobre una imagen de fondo:**

Basta con crear una variable indicando el *path* y el *nombre de la imagen*, por ejemplo: \$b="./images/cruz.jpg" (ojo formato vieja y nueva iguales) y sustituir la instrucción de *creación de imagen* \$nom = imagecreate(x,y) por:

```
$nom= imagecreatefrompng ($b) o $nom= imagecreatefromjpeg ($b)
```

- **Trazando segmentos:**

```
imageline($nom,x0, y0,x1,y1,$col)
```

x0, y0 e x1, y1 son las coordenadas los

- **Crear transparencias:** (sólo en png)

```
imagecolortransparent ($nom , $col).
```

\$color es el *color que pretendemos hacer transparente*.

1.2 Insertando textos:

Varias funciones:

- **imagechar (\$im, n, x, y, \$txt, \$col)** Inserta el *primer carácter de la cadena con orientación horizontal*.

- n es un número *entre UNO y CINCO* (tamaño de la letra de menor a mayor).
- x e y coordenadas de esquina superior izquierda del carácter a representar.
- \$txt es la cadena de la que se extraerá el *primer carácter*.

- **imagecharup (\$im, n, x, y, \$txt, \$col)** Similar al anterior, pero *orientación vertical*.

- **imagestring (\$im, n, x, y, \$txt, \$col)** Similar a imagechar se inserta *la cadena completa*.

- **imagestringup (\$im, n, x, y, \$txt, \$col)** *cadena completa, orientación vertical*

1.2.1 Tipos de letra

Todas las funciones anteriores utilizan siempre la fuente predefinida por PHP y sólo permiten los cinco tamaños que hemos podido ver en los ejemplos.

1.2.2 Utilizando fuentes TrueType

Para usar estas funciones es preciso que estén instaladas las librerías FreeType y conocer *el path* de directorio que contiene las fuentes TrueType.

Instalación de fuentes

Se crea un *subdirectorio* llamado fuentes (dentro del directorio donde estén las páginas). En él se incluyen las fuentes a utilizar.

Escribiendo con fuentes TrueType

```
Imagettfttext($nom, tam, ang, x, y, $col, $fuente, $txt)
```

- tam: número entero que indica el tamaño de la fuente.
- ang Inclinação del texto (grados sexagesimales) (ang=0, texto en horizontal).
- x e y son las coordenadas del punto de inicio de la esquina inferior izquierda del texto.
- \$fuente cadena de texto que contiene el path y el nombre de la fuente.
- \$txt la variable que contiene el texto a insertar.

Colocando textos

Con esta función podemos determinar las dimensiones de una caja de texto (el rectángulo imaginario que rodea el texto).

```
$rec= ImageTTFBBox (tam, ang, $fuente, $txt)
```

La variable \$rec recoge un array escalar cuyos valores son las coordenadas de las cuatro esquinas de la *caja de texto*. Los índices de cada elemento de ese array son los siguientes:

Inferior izquierdo \$rec[0], \$rec[1]

Inferior derecho \$rec[2], \$rec[3]

Superior derecho \$rec[4], \$rec[5]

Superior izquierdo \$rec[6], \$rec[7]

Las coordenadas del *vértice inferior izquierdo* son siempre (0,0).

Los puntos situados *por encima* del (0,0) tienen *ordenada* negativa.

Las *abscisas* de los puntos situados *a la izquierda* del (0,0) son negativas.

1.2.3 Guardando imágenes

Las imágenes *no se guardan en servidor*. Si se quiere guardarlas hay añadir a las etiquetas Imagepng(\$nombre) o Imagejpeg(\$nombre) la extensión del fichero que vamos de guardar. Así por ejemplo:

```
Imagepng($nombre, "mi_imagen.png") o Imagejpeg($nombre, "mi_imagen.jpg")
```