Algunas posibilidades

Existen diferentes posibilidades de creación de ficheros PDF mediante el uso de funciones de PHP. La distribución de PHP incluye funciones para la creación de ficheros usando las bibliotecas PDFlib de *Thomas Merz* que están disponibles en http://www.pdflib.com/.

La distribuciones de PHP para Windows incluyen estas librerías restringiendo su uso a actividades no comerciales. Cuando se trata de Linux/Unix la compilación y utilización de esta biblioteca requiere *su compra*. Por esa razón son muchos los hostings que **no incluyen** entre sus servicios la posibilidad de uso de esta librería.

Una solución alternativa y **gratuita** es el uso de **la clase FPDF** que permite generar documentos PDF directamente desde PHP sin necesidad de utilización de ninguna librería externa.

En http://www.fpdf.org podrás encontrar las últimas versiones de esta clase así como información y documentación relacionada con su utilización.

Dado el carácter **freeware** de esta *clase* y las posibilidades que ofrece para ser modificada y complementada, será la que utilizaremos para desarrollar los contenidos de este tema.

Creación de ficheros PDF

La creación de ficheros PDF requiere que el script incluya la clase **fpdf.php**. Para ello bastará con que contenga la siguiente instrucción:

```
include("fpdf.php")
```

Si el fichero fpdf.php estuviera en un directorio distinto del actual habría de escribirse la ruta completa del mismo en la instrucción anterior.

Además de esto, para poder usar la tipografía propia de la clase será necesario asignar a una **constante**, con nombre **FPDF_FONTPATH** (el nombre ha de mantenerse ya que es que usa en la *clase* **fpdf**), la ruta absoluta hasta el directorio que contiene las fuentes y que en nuestro caso será el direcotorio **fontsPDF** tal como comentamos a la derecha.

La asignación de valores a la

Usando las librerías PDFLib

Ver índice | Búsqueda rápida | Página anterior | Página siguiente

Hemos de *descomentar* la línea que dice:

```
;extension=php_pdf.dll
```

y como *descomentar* no es otra cosa que quitar el punto y coma que va delante de una línea, habremos de dejarla así:

```
extension=php_pdf.dll
```

Una vez guardados los cambios habrá que –como siempre en estos casos– volver a iniciar el servidor Apache.

Para comprobar si está activa la modificación del php.ini que comentamos a la izquierda- bastaría con visualizar el *famosísimo* info.php y comprobar que aparece lo siguiente:

pdf

PDF Support	enabled
PDFlib GmbH Version	4.0.2
Revision	$Revision: 1.112.2.7 $

Si es así, ya estamos en disposición de empezar a trabajar con este nuevo tipo de funciones PHP.

La visualización de ficheros con extensión PDF requiere tener instalado el programa Adobe Acrobat Reader. En caso de que nos dispusieras de él puedes descargar desde el enlace incluido en este párrafo.

Desde enlace, http://www.pdflib.com/products/pdflib/download/index.html tienes acceso a la descarga de estas librerías y también a la documentación relativa a la forma de utilización de las mismas.

> **¡Cuidado!**
>
> Para el desarrollo de los contenidos de este curso no utilizaremos la librería PDFLib. Por esa razón no es necesario que efectuemos los cambios de configuración descritos en el párrafo anterior.
> La clase FPDF -que será la que utilizaremos- **no requiere** ninguna modificación en la configuración de php.ini.

La *clase* FPDF

Al realizar la descarga del fichero fpdf16.zip desde http://www.fpdf.org podremos observar que el zip incluye tres directorios: **doc**, **font** y **tutorial** y una serie de ficheros entre los que podremos encontrar **fpdf.php** que es el que realmente contiene la clase.

Hemos incluido, en el directorio **cursophp** de estos materiales, el fichero **fpdf.php** de esa distribución y también el directorio **font** que hemos renombrado como **fontspdf** al efecto de facilitar la identificación.

El resto de los ficheros, son meramente informativos, no los hemos incluido ya que no van a resultarnos necesarios para el uso de esta clase. Si tratáramos de utilizar esta clase en un servidor remoto deberíamos transferir al mismo, con estos o diferentes nombres, los mismos ficheros y directorios que hemos incluido junto con estos materiales.

constante la haremos mediante la sintaxis:

define('FPDF_FONTPATH','*ruta*')

El constructor FPDF

La clase incluida en el fichero **fpdf.php** tiene por nombre **FPDF**. Por tanto su uso requerirá la creación de un nuevo objeto mediante la sintaxis:

$objeto= **new FPDF()**

Al crear el nuevo objeto se ejecuta siempre el *constructor* (recuerda que un constructor es una función con idéntico nombre que la clase que lo contiene -en este caso FPDF- que se ejecuta de forma automática el momento en que es creado un nuevo objeto) que utiliza tres parámetros: *orientacion*, *unidad de medida* y *formato*.

Orientación, medidas y formato

Al crear un nuevo objeto pueden incluirse los valores de todos o parte de estos parámetros. En ese caso la sintaxis sería:
$obj=new FPDF(ort,unds,tam)

La *orientacion*(ort) permite dos valores: **P** (**normal** ó *Portrait*) y **L** (**apaisado** ó *Landscape* en denominación inglesa). El valor por defecto es **P** (**normal**).

El *unds* (unidad de medida a utilizar) permite dos valores: **in** (**pulgadas**),**pt** (**puntos**), **mm** (**milímetros**) y también **cm** (**centímetros**). El valor por defecto es **mm** (**milímetros**).

Recuerda que **una pulgada** equivale a **25,4 milímetros** y que **un punto** equivale a **1/72 pulgadas** (0,35277 mm.).

Respecto a los formatos, los valores preestablecidos son los siguientes:

Valor	Dimensiones (mm.)
A4	210x297
A5	148,5x210
A3	297x420
Legal	215,9x355,6
Letter	215,9x279,4

Pueden crearse documentos PDF con dimensiones distintas a las anteriores. Para ello sería necesario crear un array con las dimensiones pretendidas

$medidas=array(ancho,alto)

Los tipos de letra

Por defecto el directorio **font** que se incluye en la distribución de http://www.fpdf.org -que nosotros hemos renombrado como **fontspdf**- incluye las tipografías: *Courier*, *Helvetica* y *Times*, (permitiendo utilizar el nombre *Arial* como sinónimo de *Helvetica*) y las fuentes de símbolos: *Symbol* y *Zapfdingbats*.

Sin embargo, es posible usar cualquier otra tipografía. En páginas siguientes veremos la forma en que pueden generarse e incluirse nuevos tipos de letra mediante la utilidad **makefont** incluida en el propio directorio **fontspdf**.

El primer PDF

Como primer ejemplo crearemos un fichero con dos páginas *en blanco*.

Una vez creado el objeto se **añaden las páginas** mediante la función (incluida en la clase FPDF) **AddPage()** que debe ser invocada mediante la sintaxis:

$objeto->AddPage();

```
<?
# incluimos la clase fpdf que está en este mismo directorio
include("fpdf.php");
# y definimos la constante FPDF_FONTPATH como la ruta absoluta
# hasta el directorio que contiene las fuentes tipográficas
define('FPDF_FONTPATH',$_SERVER['DOCUMENT_ROOT'].'/cursophp/fontspdf/');
# creamos  un nuevo objeto (MiPDF) utilizando la clase FPDF
$MiPDF=new FPDF();
# creamos una página en blanco
$MiPDF->Addpage();
# creamos una segunda página en blanco
$MiPDF->Addpage();
# visualizamos el documento
$MiPDF->Output();
?>
```

ejemplo132.php

Dimensionado y orientación de documentos PDF

En este ejemplo establecemos dimensiones, unidades de medida y orientación del documento. Comprobaremos que podemos cambiar la orientación (de normal a apaisada o viceversa) de cada una las páginas del documento.

Las modificaciones de orientación de las páginas pueden realizarse incluyendo el tipo de orientación (L ó P) de esa página concreta en la llamada al método AddPage(). La sintaxis sería: *$objeto->AddPage('L')* para el caso de apaisado ó *$objeto->AddPage('P')* si se tratara de orientación *normal*.

Cuando se añade una página sin especificar la orientación en **AddPage()** la nueva página tendrá la orientación indicada en la llamada al constructor del nuevo objeto (*$objeto= new FPDF('orientacion','unidades','tamaño')*).

```
<?
# incluimos la clase fpdf que está en este mismo directorio
include("fpdf.php");
# y definimos la constante FPDF_FONTPATH como la ruta absoluta
# hasta el directorio que contiene las fuentes tipográficas
define('FPDF_FONTPATH',$_SERVER['DOCUMENT_ROOT'].'/cursophp/fontspdf/');
/* vamos a configurar el documento como apaisado (P), utilizando
 las pulgadas como unidad de medida y unas dimensiones "no estandar"
 de 10 x 20 pulgadas */

# creamos un array con las dimensiones (ancho y alto);
$dimensiones=array (10,20);
# creamos  un nuevo objeto (MiPDF) utilizando la clase FPDF
# incluyendo en este caso los valores a utilizar por el constructor
$MiPDF=new FPDF('P','in',$dimensiones);
# creamos una página en blanco. Incluimos, para esta primera página
# un cambio de orientación respecto a la inicial
```

e incluir esa variable como parámetro -puedes verlo en el ejemplo de la derecha- en la creación del nuevo objeto.

El método Output()

La clase FPDF incluye la función Output para poder visualizar o guardar el documento creado. Utiliza dos parámetros: *nombre del fichero PDF* y *destino del documento*. La sintaxis podría ser la siguiente:

$obj->Output(nomb,dest)

Puede asignarse cualquier nombre, con o sin extensión, tal como puede verse en el ejemplo. Cuando no se indica el nombre, la función asigna por defecto el de **doc.pdf**.

El parámetro destino puede tomar uno de los siguientes valores:

I
permite visualizar el fichero directamente en el navegador si el plugin si está disponible. La opción «Guardar como...» asignaría, por defecto, el nombre especificado en el parámetro *nomb*.

D
envía el fichero al navegador y muestra ventana de opción que permite elegir entre *Abrir* o *Descargar*. Si se elige esta última guardará el fichero con el nombre especificado en el parámetro *nomb*.

F
guarda el fichero en el directorio actual del servidor con el nombre asignado en la opción **nomb**.

El método SetDisplayMode()

Este método -sólo afecta a la forma en la se visualiza el documento en la pantalla del cliente- permite configurar dos parámetros: *tamaño* y *forma de visualización*.

El primero puede usar una de estas opciones: *fullpage* (el zoom del visor se adapta de modo que *encaje* la página completa en la pantalla); *fullwidth* (ajusta el zoom de forma que se visualize el documento ajustando el ancho al de la pantalla); *real* (ajusta el zoom de visor al 100%) ó *default* que usa el modo por defecto del visor.

Mediante el segundo parámetro se puede establecer la forma de visualización. La opción *single* muestras las páginas separadas de una en una; *continuous* va mostrándolas una detrás de otra de forma continua; *two* muestras dos columnas (con dos páginas) simultaneamente, y *default* que, como en el caso anterior, usa el

```
$MiPDF->Addpage('L');
# creamos una segunda página en blanco
# en la que, al no incluir el parámetro de orientación
# utilizará el valor utilizado por el constructor.
$MiPDF->Addpage();
# visualizamos el documento
$MiPDF->Output();
?>
```

ejemplo133.php

Un ejemplo un poco más completo

Aunque no tienen excesiva utilidad práctica, vamos a comentar someramente aquí -también los incluimos en el ejemplo- algunos de los métodos, de carácter básicamente informativo, que incluye la clase FPDF.

SetAuthor("*nombre del autor***")** Permite incluir una cadena con el nombre del autor.

SetCreator("*nombre de la aplicación***")** Permite incluir una cadena con el nombre del programa utilizado para crear el documento.

SetTitle("*título del documento***")** Permite incluir una cadena con el título del documento.

SetKeywords("*palabras clave***")** Permite incluir una cadena con una lista de palabras clave separadas por espacios.

Los datos incluidos en el documento mediante los métodos anteriores no son presentados diractamente en el documento. Sólo son visualizables cuando se exploran los *metadatos* del documento.

```
<?
# incluimos la clase fpdf que está en este mismo directorio
include("fpdf.php");
# y definimos la constante FPDF_FONTPATH como la ruta absoluta
# hasta el directorio que contiene las fuentes tipográficas
define('FPDF_FONTPATH',$_SERVER['DOCUMENT_ROOT'].'/cursophp/fontspdf/');
/* vamos a configurar el documento como apaisado (P), utilizando
 los milímetros como unidad de medida y unas dimensiones "no estandar"
 de 140 x 200 milimetros */
# creamos un array con las dimensiones (ancho y alto);
$dimensiones=array (140,200);
# creamos  un nuevo objeto (MiPDF) utilizando la clase FPDF
# incluyendo en este caso los valores a utilizar por el constructor
$MiPDF=new FPDF('P','mm',$dimensiones);
# el método SetAuthor nos permite incluir el nombre del autor
$MiPDF->SetAuthor('Pepe Pérez');
# el método SetCreator nos permite incluir el nombre de la
# aplicacion que genera el pdf
$MiPDF->SetCreator('clase FPDF');
# el método SetTitle nos permite incluir un título
$MiPDF->SetTitle('Pruebas del pdf');
# el método SetKeywords nos permite incluir palabras claves
# separadas por espacios y dentro de una misma cadena
$MiPDF->SetKeywords('palabra1 palabra2');
# el método SetDisplayMode nos permite incluir palabras claves
# separadas por espacios y dentro de una misma cadena
$MiPDF->SetDisplayMode('fullpage','two');
# creamos una página en blanco. Incluimos, para esta primera página
# un cambio de orientación respecto a la inicial
$MiPDF->Addpage('L');
# creamos una segunda página en blanco
# en la que, al no incluir el parámetro de orientación
# utilizará el valor utilizado por el constructor.
$MiPDF->Addpage();
# visualizamos el documento
$MiPDF->Output('donpepito.pdf','I');
?>
```

Los ejemplos siguientes son similares. La única modificación que contienen respecto al código

modo por defecto del visor.

fuente anterior es la correspondiente al segundo parámetro (destino) del método Output.

Destino="I"	Destino="D"	Destino="F"