

## Lucha de clases. Comparativa de rendimiento de clases FPDF vs CLASSEZPDF mediante ejemplos ilustrativos

**Ángel González Santillán**

Departamento de Sistemas y Computación, Instituto Tecnológico de Tuxtepec  
[santillan18@hotmail.es](mailto:santillan18@hotmail.es)

**Horacio Maldonado Hernández**

Departamento de Sistemas y Computación, Instituto Tecnológico de Tuxtepec  
[lacho\\_m\\_h@hotmail.com](mailto:lacho_m_h@hotmail.com)

### Resumen

En las carreras de ingeniería en informática se imparten las materias de comercio electrónico (TWG-1304), programación Web avanzada (TWG-1305) tópicos de base de datos (IFF-1026), desarrollo de aplicaciones Web (IFD-1010) y Taller de base de datos (AEH-1063), de las cuales la importancia de generar reportes en formato PDF tomadas de las bases de datos bajo cualquier sistema, es muy importante por el impacto que pudiesen tomar al momento de procesar la información y tomar decisiones; sin embargo, al momento de elegir alguna de las librerías disponibles para la elaboración de un reporte. PDF existen o surgen inconvenientes sobre el mejor desempeño considerando que como toda aplicación tiene sus limitantes en su utilización, por lo que un análisis comparativo basados en ejemplos ilustrativos permitirá favorecer la toma de decisiones sobre una u otra librería para su uso y aplicación a un sistema cualquiera.

**Palabras clave:** librerías, fpdf, class.ezpdf.

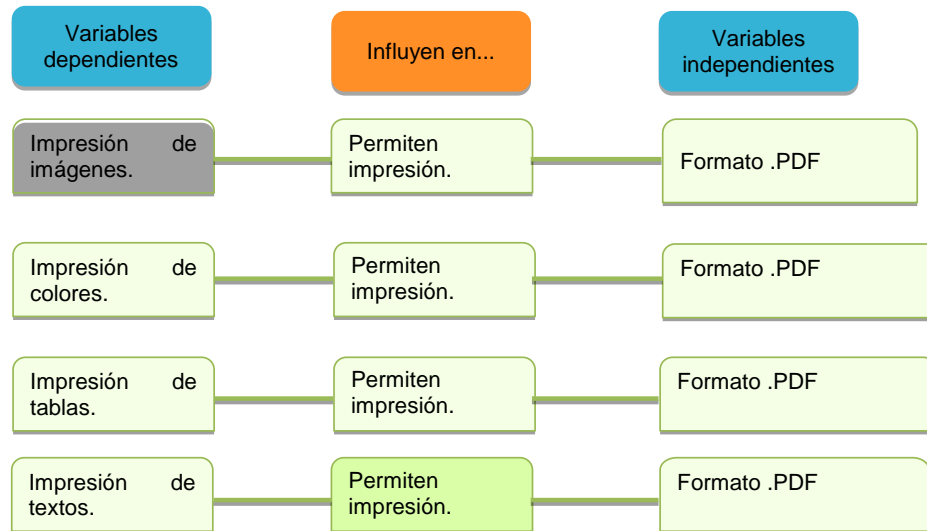
### Introducción

Hoy en día existen muchas librerías que se utilizan en la generación de reportes en formato .pdf como la fpdf, tcpdf, class.ezpdf así como dompdf , todas ellas son de software libre y se pueden modificar al gusto de uno en base a las necesidades que más le convengan a uno dependiendo las necesidades que en ese momento tenga uno ya sea al momento de dar clases (como docente) y requiera basarse en un ejemplos para demostrar su utilización, o si tenga la necesidad de dar a conocer no solo que es lo que hace una librería u otra si no denotar los alcances y limitaciones, o también si desea demostrar en donde se pueden aplicar, en qué casos, en qué tipo de sistemas en tiempo real y como interacciona con las bases de datos, eso es para

el caso de docentes que deseen aplicarlo en cuales quiera de los casos mencionados anteriormente sin embargo sería de mucha utilidad también para alguien que se dedica a programas sistemas y al momento de la toma de decisiones requiere de la utilización de formatos de salida por reportes de impresión por lo que conocer de forma rápida por medio de ejemplos que es lo que sí se puede hacer y lo que no con una librería y otra permea los tiempos y mejora las apariencias en sus reportes, pues es muy común que en la creación de un sistema web cuando se está programando se lleven a cabo varias pruebas para ver qué es lo que hace y cual librería o clase convendría más para lo que uno está buscando o simplemente porque se tiene problemas con la famosa salida de los acentos, por ello contar con información oportuna para su correcto cumplimiento es de vital importancia para toda la comunidad (Cuerpo directivo, docente, no docente , público involucrado en el área y alumnos) permitiendo obtener múltiples beneficios haciendo con ello una manera más eficiente y cómoda permitiendo ahorro de tiempo.

## Metodología

El desarrollo de éste proyecto tiene un enfoque cuantitativo porque se pretende intencionalmente acotar por medio de la comparación de librerías de impresión en formato .PDF los informes o reportes ; Las condiciones detectadas en el planteamiento del problema y antecedentes son únicas, siendo éste proyecto muy específico pues se detectaron las variables a medir como: Impresión de imágenes, Impresión de colores, impresión de tablas, impresión de Textos, por lo que las variables independientes y dependientes a medir se muestran como siguen:



El proyecto de investigación está basado en estudios correlacionales debido a que se pretende responder a preguntas de investigación que asocian variables utilizando un patrón predecible para un grupo de librerías de impresión en formato .PDF como por ejemplo si ¿Es posible imprimir reportes de igual manera utilizando las 2 librerías FPDF VS CLASS.EZPDF? Por lo que de esta manera se pretende no solo dar respuesta a las preguntas de hipótesis anteriormente planteadas sino también evaluar el grado en que se asocia las variables con la aplicación de las librerías FPDF VS CLASS.EZPDF.

Para medir las variables a correlacionar, la información procederá de las aplicaciones generadas para probar cada una de las librerías y comprobar por medio de los formatos de salida .PDF si es posible o no efectuar tal aplicación. Para el caso de este proyecto la muestra está definida en la unidad de análisis como las 2 librerías de generación de reportes .PDF.

Para la selección de la muestra se consideró la muestra no probabilística por que para ésta investigación se requiere de comprobar lo que se puede hacer y no con las librerías de generación de reportes en cuestión, por lo que claramente está definida la forma en cómo se seleccionará la muestra. El objetivo es que al considerar la selección antes mencionada se sometan a un proceso de evaluación para de esa forma medir las variables del proyecto aquí planteadas y dar respuesta a la hipótesis de forma contundente basada en 2 de las librerías utilizadas en la generación de reportes.

## FPDF.

Para comenzar se verá ¿qué es la librería fpdf?, se puede mencionar que es una librería que está escrita en php y permite tomar información de un sistema, tomar información de una base de datos y plasmarla en un archivo .PDF considerando características de imagen, color, tipos de letras, márgenes entre otras características como se encontró en (MAIDANA, Esteban, fpdflibrary, <http://www.fpdf.org/>, 20 Mayo 2013):

FPDF es una clase escrita en PHP que permite generar documentos PDF directamente desde PHP, es decir, sin usar la biblioteca PDFlib. La F de FPDF significa Free (gratis y libre): puede usted usarla para cualquier propósito y modificarla a su gusto para satisfacer sus necesidades.

FPDF tiene otras ventajas: funciones de alto nivel. Esta es una lista de sus principales características:

- Elección de la unidad de medida, formato de página y márgenes

- Gestión de cabeceras y pies de página

- Salto de página automático

- Salto de línea y justificación del texto automáticos

- Admisión de imágenes (JPEG, PNG y GIF)

- Colores

- Enlaces

- Admisión de fuentes TrueType, Type1 y codificación

- Compresión de página

Existen varias versiones 10 para ser exactos:

- v1.7 (18/06/2011)

- v1.6 (03/08/2008)

v1.53 (31/12/2004)

v1.52 (30/12/2003)

v1.51 (03/08/2002)

v1.5 (28/05/2002)

v1.41 (13/03/2002)

v1.4 (02/03/2002)

v1.31 (12/01/2002)

v1.3 (03/12/2001)

Entre los ejemplos clásicos de generación de reportes .PDF como se encontró en (MAIDANA, Esteban, fpdflibrary, <http://www.fpdf.org/>, 20 Mayo 2013) :

```
<?php
require('fpdf.php');

$pdf = new FPDF();
$pdf->AddPage();
$pdf->SetFont('Arial','B',16);
$pdf->Cell(40,10,'¡Hola, Mundo!');
$pdf->Output();
?>
```

Que genera un reporte (documento .pdf) con la leyenda hola mundo. <sup>¡Hola, Mundo!</sup> . Después de incluir el fichero de la clase, creamos el objeto FPDF. El constructor FPDF() se usa aquí con sus valores por defecto: las páginas son de tamaño a4 alargado y la unidad de medida es el milímetro. Se podría haber declarado explícitamente con:

```
$pdf = new FPDF('P','mm','A4');
```

Es posible usar el formato apaisado(L), otros formatos de página (como Carta y Legal) y otras unidades de medida (pt, cm, in).

Por el momento no hemos creado ninguna página, así que añadiremos una con AddPage(). El origen de coordenadas está en la esquina superior izquierda y la posición actual está por defecto situada a 1 cm de los bordes; los márgenes pueden cambiarse con SetMargins().

Antes de que podamos imprimir texto, es obligatorio escoger una fuente con SetFont(), si no, el documento no será válido. Escogemos Arial en negrita de tamaño 16:

```
$pdf->SetFont('Arial','B',16);
```

Podríamos haber especificado itálica con I, subrayado con U o normal con una cadena vacía (o cualquier combinación de las anteriores). Observe que el tamaño de la fuente se determina en puntos, no en milímetros (ni en cualquier otra unidad establecida por el usuario); es la única excepción. Las otras fuentes incorporadas son Times, Courier, Symbol y ZapfDingbats.

Ahora podemos imprimir una celda con Cell(). Una celda es una superficie rectangular, con borde si se quiere, que contiene texto. Se imprime en la posición actual. Especificamos sus dimensiones, el texto (centrado o alineado), si queremos dibujar o no los bordes, y dónde se ubicará la posición actual después de imprimir la celda (a la derecha, debajo o al principio de la siguiente línea). Para añadir el borde, deberemos hacer esto:

```
$pdf->Cell(40,10,';Hola, Mundo!',1);
```

Para añadir una nueva celda a continuación, con el texto centrado e ir a la siguiente línea, escribiremos:

```
$pdf->Cell(60,10,'Hecho con FPDF.',0,1,'C');
```

Nota: el salto de línea puede provocarse también mediante Ln(). Este método le permite especificar además la altura del salto.

Finalmente, el documento se cierra y se envía al navegador con Output(). También podríamos haberlo guardado en un fichero pasando como parámetro el nombre del archivo.

Cuidado: en caso de que el PDF se envíe al navegador, nada más debe enviarse, ni antes ni después (el más mínimo espacio en blanco o retorno de carro también cuenta). Si se envía algún dato antes, obtendrá el mensaje de error: "Some data has already been output, can't send PDF file". Si se envía después, su navegador puede que muestre únicamente una página en blanco.

Es posible desarrollar encabezados y pies de página como se encontró en (MAIDANA, Esteban, fpdflibrary, <http://www.fpdf.org/>, 20 Mayo 2013):

### **Cabecera, pie, salto de página e imagen**

Aquí tenemos un ejemplo de dos páginas con cabecera, pie de página y logotipo:

```
<?php
require("fpdf.php");

class PDF extends FPDF
{
    // Cabecera de página
    function Header()
    {
        // Logo
        $this->Image('logo_pb.png',10,8,33);
        // Arial Bold 15
        $this->SetFont('Arial','B',15);
        // Movernos a la derecha
        $this->Cell(80);
        // Título
        $this->Cell(30,10,'Title',1,0,'C');
        // Salto de línea
        $this->Ln(20);
    }

    // Pie de página
    function Footer()
    {
        // Posición: a 1,5 cm del final
        $this->SetY(-15);
        // Arial italic 8
        $this->SetFont('Arial','I',8);
        // Número de página
        $this->Cell(0,10,'Page '.$this->PageNo().'/{nb}',0,0,'C');
    }

    // Creación del objeto de la clase heredada
    $pdf = new PDF();
    $pdf->AliasNbPages();
    $pdf->AddPage();
    $pdf->SetFont('Times','',12);
    for($i=1;$i<=40;$i++)
        $pdf->Cell(0,10,'Imprimiendo línea número '.$i,0,1);
    $pdf->Output();
}>
```

Este ejemplo hace uso de los métodos Header() y Footer() para procesar las cabeceras y pies de páginas. Se llaman automáticamente. Ya existen en la clase FPDF original, pero no hacen nada. Por ello, tenemos que heredar la clase y sobrescribirlos.

El logotipo se imprime con el método Image() especificando su esquina superior izquierda y su anchura. La altura se calcula automáticamente respetando las proporciones de la imagen.

Para imprimir el número de página, se le pasa un valor nulo (null) como ancho de la celda. Eso significa que la celda se extenderá hasta el margen derecho de la página; puede ser útil centrar el texto. El número actual de la página se devuelve por el métodos PageNo(); mientras que el número total de páginas se obtiene mediante un valor especial de {nb} que será sustituido cuando se cierre el documento (suponiendo que usted antes utilizara AliasNbPages()).

Observe el uso del método SetY () que le permite especificar la posición en una ubicación absoluta respecto del origen de coordenadas de la página, empezando por el principio o por el final.

Otra característica interesante se usa en el ejemplo: el salto automático de página. Tan pronto como una celda cruza el límite máximo de la página (a 2 cm del final, por defecto), se ejecuta un salto y se recupera la fuente. Aunque la cabecera y el pie usan su propia fuente (Arial), el cuerpo del documento continua con Times. Este mecanismo automático de recuperación también se

aplica a los colores y al ancho de línea. El límite que fuerza los saltos de página puede establecerse con `SetAutoPageBreak ()`.

Para este ejemplo tendría un reporte de impresión .PDF como sigue:



Title

Imprimiendo línea número 1

Imprimiendo línea número 2

Imprimiendo línea número 3

Terminando hasta 40.

Es posible desarrollar un script que imprima texto desde archivos .txt externos al archivo .php manipulando colores y bordes en los encabezados así como su justificación como se encontró en (MAIDANA, Esteban, fpdflibrary, <http://www.fpdf.org/>, 20 Mayo 2013):

### **Saltos de línea y colores**

Continuemos con un ejemplo que imprime párrafos justificados. También ilustra el uso de colores.



## El siguiente

```
<?php
require('fpdf.php');

class PDF extends FPDF
{
function Header()
{
    global $title;

    // Arial bold 15
    $this->SetFont('Arial','B',15);
    // Calculamos ancho y posición del título.
    $w = $this->GetStringWidth($title)+6;
    $this->SetX((210-$w)/2);
    // Colores de los bordes, fondo y texto
    $this->SetDrawColor(0,80,180);
    $this->SetFillColor(230,230,0);
    $this->SetTextColor(220,50,50);
    // Ancho del borde (1 mm)
    $this->SetLineWidth(1);
    // Título
    $this->Cell($w,9,$title,1,1,'C',true);
    // Salto de línea
    $this->Ln(10);
}

function Footer()
{
    // Posición a 1,5 cm del final
    $this->SetY(-15);
    // Arial itálica 8
    $this->SetFont('Arial','I',8);
    // Color del texto en gris
    $this->SetTextColor(128);
    // Número de página
    $this->Cell(0,10,"Página ".$this->PageNo()." de ".$this->PageCount(),0,0,'C');
}
}
```

```
function ChapterTitle($num, $label)
{
    // Arial 12
    $this->SetFont('Arial','',12);
    // Color de fondo
    $this->SetFillColor(200,220,255);
    // Título
    $this->Cell(0,6,"Capítulo $num : $label",0,1,'L',true);
    // Salto de línea
    $this->Ln(4);
}

function ChapterBody($file)
{
    // Leemos el fichero
    $txt = file_get_contents($file);
    // Times 12
    $this->SetFont('Times','',12);
    // Imprimimos el texto justificado
    $this->MultiCell(0,5,$txt);
    // Salto de línea
    $this->Ln();
    // Cita en itálica
    $this->SetFont('','I');
    $this->Cell(0,5,"(fin del extracto)");
}

function PrintChapter($num, $title, $file)
{
    $this->AddPage();
    $this->ChapterTitle($num,$title);
    $this->ChapterBody($file);
}
}
```

```
$pdf = new PDF();
$title = '20000 Leguas de Viaje Submarino';
$pdf->SetTitle($title);
$pdf->SetAuthor('Julio Verne');
$pdf->PrintChapter(1,'UN RIZO DE HUIDA','20k_c1.txt');
$pdf->PrintChapter(2,'LOS PROS Y LOS CONTRAS','20k_c2.txt');
$pdf->Output();
?>
```

El método GetStringWidth() le permite determinar la longitud de una cadena en el tipo de letra actual, y se usa aquí para calcular la posición y ancho del borde que rodea al título. Después se establecen los colores (mediante SetDrawColor(), SetFillColor() y SetTextColor()) y el borde de la línea se establece en 1 mm (en contra de los 0,2 por defecto) con SetLineWidth(). Finalmente, imprimimos la celda (el último parámetro a 1 indica que debe colorearse el fondo).

El método usado para imprimir los párrafos es MultiCell(). Cada vez que la línea llega al extremo derecho de la celda o aparece un carácter de fin de línea, se ejecuta un salto de línea y se crea automáticamente otra celda debajo de la actual. El texto se encuentra justificado por defecto. Se definen dos propiedades del documento: título (SetTitle()) y autor (SetAuthor()). Las propiedades pueden verse de dos maneras. La primera es abrir el documento directamente con Adobe Reader, vaya al menú Archivo y seleccione la opción Propiedades del documento. La segunda, también disponible desde el plug-in, es hacer clic con el botón derecho y seleccione Propiedades del documento. Para éste ejemplo tendría la siguiente salida de impresión de .pdf:

20000 Leguas de Viaje Submarino

Capítulo 1 : UN RIZO DE HUIDA

The year 1866 was marked by a bizarre développement, an unexplained and downright inexplicable phenomenon that surely no one has forgotten. Without getting into those rumors that upset civilians in the seaports and deranged the public mind even far inland, it must be said that professional seamen were especially alarmed. Traders, shipowners, captains of vessels, skippers, and master mariners from Europe and America, naval officers from every country, and at their heels the various national governments on these two continents, were

20000 Leguas de Viaje Submarino

In every big city the monster was the latest rage; they sang about it in the coffee houses, they ridiculed it in the newspapers, they dramatized it in the theaters. The tabloids found it a fine opportunity for launching all sorts of hoaxes. In those newspapers short of copy, you saw the reappearance of every gigantic imaginary creature, from "Moby Dick," that dreadful white whale from the High Arctic regions, to the stupendous kraken whose tentacles could manueve a 500-ton craft and drag it into the ocean depths. They even reprinted reports from ancient times: the views of Aristotle and Pliny accepting the existence of such monsters, then the Norwegian stories of Bishop Pontoppidan, the narratives of Paul Egede, and finally the reports of Captain Harrington -- whose good faith is above suspicion--in which he claims he saw, while aboard the Castilian in 1857, one of those enormous serpents that, until then, had frequented only the seas of France's old extremist newspaper, The Constitutionalist.

(fin del extracto)

Es posible desarrollar un script que imprima columnas consecutivas de texto desde archivos .txt o también texto tomado de una base de datos, solo que en éste ejemplo se tomará de un archivo de texto como se viene manejando como se encontró en (MAIDANA, Esteban, fpdflibrary, <http://www.fpdf.org/>, 20 Mayo 2013):

## CLASS.EZPD

Es otra librería que permite la generación de reportes en .PDF como se encontró en (Desarrollo Web, <http://blog.unijimpe.net/generar-pdf-con-php/>, 20 Mayo 2013):

PHP entre sus características primarias tiene la capacidad de generar archivos PDF, tiene funciones específicas para personalizar y crear estos archivos pero son muy difíciles de utilizar, para ello existe una clase llamada R&OS: pdfclass el cual provee métodos muy potentes y simplificados para la creación de archivos PDF.

Se iniciará con un script del siguiente ejemplo:

```
<?php
include ('class.ezpdf.php');
$pdf = new Cezpdf();
$pdf->selectFont('Fonts/Helvetica.afm');
$pdf->ezText('Mi primer pdf en PHP', 30);
$pdf->ezStream();
?>
```

Mi primer pdf en PHP

Para éste ejemplo tendría la siguiente salida de impresión de .pdf: Hemos empleado los métodos: selectFont para seleccionar la fuente con la que escribiremos los textos, ezText que permite agregar un texto con su respectivo tamaño de fuente y finalmente generamos el archivo con la función ezStream.

Ahora colocaremos un script similar pero que contenga la fecha y la hora:

```
<?php
include ('class.ezpdf.php');
$pdf = new Cezpdf('a4');
$pdf->ezText("<b>Factura numero 123123123 </b>\n",20);
$pdf->ezText("Esta es una prueba de pdf\n",12);
$pdf->ezText("\n\n",10);
$pdf->ezText("<b>Fecha:</b> ".date("d/m/Y"),10);
$pdf->ezText("<b>Hora:</b> ".date("H:i:s")."\n",10);
$pdf->ezStream();
?>
```

Factura numero 123123123

Esta es una prueba de pdf

Fecha: 21/05/2013  
Hora: 19:30:04

Para este ejemplo tendría la siguiente salida de impresión de .pdf:

Ahora se desarrollará un script que imprima una tabla con los meses del año numerándolos y en color las líneas de la tabla, para ello se considerará el siguiente código:

```
<?php
include('class.ezpdf.php');
$pdf = new Cezpdf('a4');
$pdf->selectFont('fonts/calibri.afm');

$data[] = array('num'=>1, 'mes'=>'Enero');
$data[] = array('num'=>2, 'mes'=>'Febrero');
$data[] = array('num'=>3, 'mes'=>'Marzo');
$data[] = array('num'=>4, 'mes'=>'Abril');
$data[] = array('num'=>5, 'mes'=>'Mayo');
$data[] = array('num'=>6, 'mes'=>'Junio');
$data[] = array('num'=>7, 'mes'=>'Julio');
$data[] = array('num'=>8, 'mes'=>'Agosto');
$data[] = array('num'=>9, 'mes'=>'Septiembre');
$data[] = array('num'=>10, 'mes'=>'Octubre');
$data[] = array('num'=>11, 'mes'=>'Noviembre');
$data[] = array('num'=>12, 'mes'=>'Diciembre');

$title = array('num'=>'<b>Numero</b>', 'mes'=>'<b>Mes</b>');

$pdf->setText("Meses en PHP</b>");
$pdf->setTable("Listado de Meses",12);
$pdf->setTable($data,$title,"", $options );
$pdf->setText("\n\n",10);
$pdf->setText("<b>Fecha:</b> ".date("d/m/Y"),10);
$pdf->setText("<b>Hora:</b> ".date("H:i:s")."\n\n",10);
$pdf->stream();
?>
```

Para este ejemplo tendría la siguiente salida de impresión de .pdf:

Meses en PHP  
Listado de Meses

Numero	Mes
1	Enero
2	Febrero
3	Marzo
4	Abril
5	Mayo
6	Junio
7	Julio
8	Agosto
9	Septiembre
10	Octubre
11	Noviembre
12	Diciembre

### Resultados y discusión

Una vez terminado el análisis, desarrollo e implementación de los scripts de las librerías, se procederá a comprobar las variables por medio del cuestionario mencionado con anterioridad (Cuestionario).

Nombre	Resultados
1. Docentes academia de Lic. Informática e Ing. Sistemas Computacionales.	25
2. Docentes academia de Lic. Informática e Ing. Sistemas Computacionales.	20
3. Docentes academia de Lic. Informática e Ing. Sistemas Computacionales.	30
4. Docentes academia de Lic. Informática e Ing. Sistemas Computacionales.	20
TOTAL	80

95 / 4= 23.75.

Resultado en la escala Likert

20                      25                      30                      35                      40

SI es igual de fácil utilizar las 2 librerías.                      Cubre al 50%                      NO e fácil utilizar las librerías solo 1 de ellas.

Se observa que al desarrollar ejemplos ilustrativos utilizando las 2 librerías de impresión en formato .pdf es igualmente sencillo utilizar una u otra librerías a excepción de tablas y acentuación de caracteres que de alguna forma son un poco complejos.

### **Trabajo a futuro**

Se espera desarrollar scripts que conectados a un base de datos demuestren el manejo de impresión de campos de tablas de base de datos en columnas de tablas dibujadas en forma dinámica en pdf con tamaños de n renglones.

### **Conclusiones**

Al concluir el proyecto se observa que las variables de hipótesis planteadas a una inicio mostraron un resultado positivo considerable al arrojar que es relativamente sencillo trabajar con una u otra solo mostrando diferencias pero no significativas al momento de utilizar impresión de tablas y acentos, pues para imprimir tablas con fpdf y tomando para cada columna de una tabla que imprime información de una base de datos al utilizar la sentencia multicell para la librería fpdf deforma la tabla y no permite su uniformidad pues por naturaleza un multicell brinca a la siguiente línea final respecto de la anterior cosa que con la librería class.ezpdf no sucede y es más fácil imprimir sea del tamaño que sea la información de la base de datos , respecto a los acentos es más fácil tratarlos con la librerías class.ezpdf que con fpdf porque modificando una sola vez la librería class.ezpdf se imprimen los acentos de forma automática cosa que con la librerías fpdf hay que modificar el código con se imprime a excepción de éstos 2 casos es muy similar la utilización para una y para otra.

## Bibliografía

Kenneth E. Kendall. (2005). Análisis y diseño de sistemas. 6th. Ed. School of Business-Camden Camden, New Jersey.

Roberto Hernández Sampieri. (2010). Metodología de la Investigación. 5th. Ed.

StigSæthe.r (2002). Manual de PHP. 2th. Ed.

IanGilfillan ().

C.J. Date (2001). Introducción a los sistemas de base de datos. 7th. Ed. Reading Massachusetts. E.U.A.

Dirección General de Institutos Tecnológicos (1997). Manual de procedimientos para la instalación y operación de las academias en el Sistema Nacional de Institutos Tecnológicos.

## Referencias electrónicas

<http://www.fpdf.org/>

<http://www.kamlov.site90.net/?p=416>