

Generación de reportes utilizando gráficas animadas con FusionCharts.php

Ángel González Santillán

Instituto Tecnológico de Tuxtepec

santillan18@hotmail.es

María Isabel Hernández Zágada

Instituto Tecnológico de Tuxtepec

Clara Aguirre Roque

Instituto Tecnológico de Tuxtepec

Juán Verdeja Cruz

Instituto Tecnológico de Tuxtepec

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 radica en que ahí se emplean herramientas para generar gráficas y que para que sean llamativas en los resultados que se desean reflejar se utilizan técnicas de gráficas (aunque para este caso gráficas animadas) que toman la información de las bases de datos bajo cualquier sistema, es muy importante la generación de éste tipo de gráficas 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 una gráfica existen múltiples opciones, pero a su vez surgen inconvenientes sobre el costo que cada una tiene para utilizarla así como

su rendimientos es por ello que se presenta la utilización de un script de java que permite basándose en ejemplos ilustrativos su utilización y que permita favorecer la toma de decisiones al momento de generar gráficas animadas.

Palabras clave: *Librerías, FusionCharts.php*

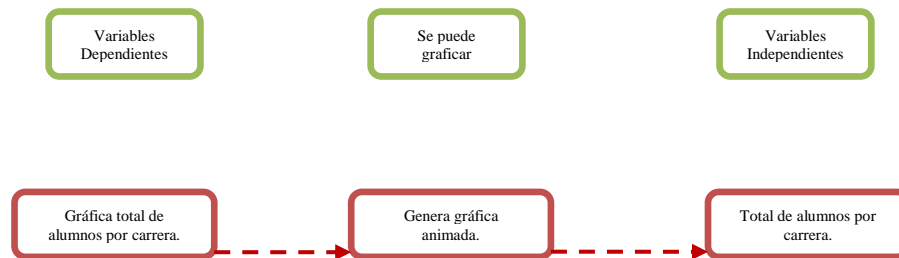
Introducción

Hoy en día existen muchas librerías que se utilizan en la generación de reportes para gráficas pero al momento de hablar de gráficas animadas a pesar de que existe diversas librerías para proyectar la información tomada de una base de datos es muy complejo en ocasiones entender su funcionamiento debido a que a pesar de que existen pequeños y sencillos tutoriales para saber cómo se utilizan es aún complejo pues sólo se explica de forma muy general su utilización pero no se observa con un sistema puesto en marcha su funcionamiento para éste caso se utilizará una librería de software libre ya que permite se pueda modificar al gusto de uno en base a las necesidades que más le convengan a uno dependiendo las necesidades que en ese momento se tengan 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 sólo que es lo que hace la librería 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 como es el caso de la librería que se utilizará *FusionCharts.php*, 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 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 como *FusionCharts.php* 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, 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 al contar con la información que se presenta en éste artículo donde por medio de un ejemplo práctico se observa la utilización de dicha librería (*FusionCharts.php*).

Metodología.

El desarrollo de éste proyecto tiene un enfoque cuantitativo porque se pretende intencionalmente demostrar como utilizando (correctamente) la librería *FusionCharts.php* se pueden generar gráficas animadas en tiempo real tomadas de una base de datos y explicar detalladamente cómo se alimenta, cómo se utiliza la librería de software libre (*FusionCharts.php*) tomando información de la base de datos que trabaja en tiempo real y dicha información sea transformada a gráficas animadas; Las condiciones detectadas en el planteamiento del problema y antecedentes serán consideradas, para que estas condiciones sean utilizadas y saber si sabiendo utilizar dicha librería permite tomar la información y graficarla, al mismo tiempo que esas gráficas servirán para la toma de decisiones para éste proyecto en específico, siendo éste proyecto muy específico pues se detectó la variable a medir : Graficación de total de alumnos por carrera siendo ésta la variable dependiente e independientes a medir que sería total de alumnos por carrera a graficar:



El proyecto de investigación está basado en estudios correlacionales debido a que se pretende responder a preguntas de investigación que asocia variables utilizando un patrón predecible que en este caso es la librería *FusionCharts.php*. Como por ejemplo si ¿Es posible generar con facilidad una gráfica animada en tiempo real tomando información de una base de datos utilizando la librería *FusionCharts.php* apoyándose en ejemplos demostrativos desarrollados en ésta investigación? 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 la variable dependiente aquí planteada con la independiente.

Para medir las variables a correlacionar, la información procederá de la aplicaciones generadas (programadas) para probar si por medio de una buena explicación de la librería *FusionCharts.php* se puede generar una gráfica en tiempo real tomada de una base de datos y comprobar por medio de los formatos de salida (gráficas animadas en éste caso será por carreras) si es posible o no efectuar tal aplicación. Para el caso de éste proyecto la muestra está definida en la unidad de análisis como la librería *FusionCharts.php* que permite salvo su buen uso y aplicación generar gráficas animadas.

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 la

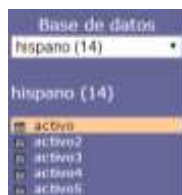
librería (*FusionCharts.php*) de generación de reportes en cuestión en conjunto con la facilidad de los ejemplos demostrativos aquí empleados para el buen uso y manejo de la librería antes mencionada, por lo que claramente está definida la forma en cómo se seleccionará la muestra. El objetivo es que al considerar la selección antes mencionada se sometan a un proceso de evaluación para de esa forma medir la variable del proyecto aquí planteadas y dar respuesta a la hipótesis de forma contundente basada en la librería utilizada en la generación de gráficas animadas.

FusionCharts.php

Ésta librería

Ahora se intentará por medio de ejemplos ilustrativos demostrar su buen uso y aplicación de la librería queriendo generar gráficas para un sistema escolar que despliega el número de alumnos que existen en cada carrera.

Para ello se parte de que se cuenta con una base de datos y sus respectivas tablas, la base de datos se llama hispano y las tablas de donde se tomarán los datos se llaman activo, activo2, activo3, activo4, activo5 como se observa:



Lo primero y más importante que debemos saber es que se debe establecer la conexión ante todo pues si se trata de generar gráficas en tiempo real conectadas a base de datos remotas debemos especificar los siguientes argumentos:

```
<?php require_once('../..../Connections/Activos3.php'); ?>
```

Se coloca esta línea por qué se hace el llamado a un archivo externo que contiene los datos de la conexión (cuando se coloca ../..../ Indica que se encuentra esa carpeta y archivo en dos subniveles por arriba de la carpeta principal), como se puede observar ese archivo externo está en la carpeta **Connections** y dentro de esa carpeta hay una que se llama **Activos3.php** y en ese archivo se tienen los siguientes argumentos:

```
$hostname_Activos3 = "localhost";  
$database_Activos3 = "hispano";  
$username_Activos3 = "root";  
$password_Activos3 = "";  
$Activos3 = mysql_pconnect($hostname_Activos3, $username_Activos3, $password_Activos3) or trigger_error(mysql_error(),E_USER_ERROR);
```

Para este caso se especifica que nuestro servidor se **localhost**, la base de datos se llama **hispano** (como se indicó anteriormente), el nombre de usuario es **root** (el nombre de usuario es definido por el administrador del servidor web cuando se crea la base de datos y lo mismo sucede con la contraseña) y para éste caso la contraseña **no tiene**.

Una vez hecho esto también se debe especificar la ruta de donde se tomará la librería que es el tema en cuestión y que nos permitirá usarla posteriormente para generar las gráficas animadas deseadas, y los argumentos son los siguientes:

```
require once("../..../JS/FusionCharts.php");
```

Como se observa la librería **FusionCharts.php** se encuentra dentro de la carpeta **JS** que está dos subniveles por arriba de la carpeta principal (en este caso la carpeta principal se llama hispano).



Una vez hecho esto se tiene todo listo para poder trabajar, lo primero que se debe de hacer es inicializar las variables que van a obtener el dato numérico que se necesita para graficar, como se comentó al inicio que las gráficas son generadas en tiempo real en base a una conexión de la base de datos (ya se explicó anteriormente el tema de la conexión) se debe hacer una consulta a la base de datos y en base a los resultados de esa consulta se obtendrán los datos que se desean graficar y con eso se obtiene un dato de la base de datos en tiempo real, a continuación se colocan los siguientes argumentos:

```

$query = "SELECT count(serialemp) AS total1 FROM activo ";
$result1 = mysql_query($query) or die(mysql_error());
$total1 = mysql_fetch_array($result1);

$query = "SELECT count(serialemp) AS total2 FROM activo ";
$result2 = mysql_query($query) or die(mysql_error());
$total2 = mysql_fetch_array($result2);

$query = "SELECT count(serialemp) AS total3 FROM activo ";
$result3 = mysql_query($query) or die(mysql_error());
$total3 = mysql_fetch_array($result3);

$query = "SELECT count(serialemp) AS total4 FROM activo ";
$result4 = mysql_query($query) or die(mysql_error());
$total4 = mysql_fetch_array($result4);
    
```

Como se observa en el código anterior se hace una consulta (**select**) para que cuente (**count**) el total de los campos existentes en **serialemp** (en este caso tiene los números de control) que es una campo que forma parte de la tabla **activo** y todo ello lo almacene en la variable **total1**, toda esta consulta (**\$query**.- ejecuta un conjunto de resultados como un objeto) se almacena el **\$result1** utilizando **mysql_query**(devuelve valor verdadero) y por último **Mysql_fecht_array(\$result1)** almacena en la variable

`$valores1` el resultado de la consulta inicial, se debe utilizar `Mysql_fetch_array` para poder recuperar los resultados de la consulta anterior en un array asociativo y de esa forma poder manipular la información derivada de la consulta, de la misma forma se efectúa el mismo procedimiento para las tablas activo3, activo4, activo5.

Como ya se tiene el resultado de la consulta en las variables antes mencionadas que fueron tomados esos datos de la base de datos por medio de la conexión previamente establecida, ahora se procede a almacenar esa información en una variable php para que después se pueda utilizar como se observa en los siguientes argumentos:

```
$intTotalAnio1 = $valores1["total1"];  
$intTotalAnio2 = $valores2["total2"];  
$intTotalAnio3 = $valores3["total3"];  
$intTotalAnio4 = $valores4["total4"];
```

La variable php es `$intTotalAnio1` y en ella se vacían los resultados de la primera consulta (`$valores1["total1"]`) de la misma manera aplica para las demás consultas efectuadas para cada tabla.

La sentencia `$strXML` sirve para concatenar los parámetros finales para el gráfico, sin embargo se debe inicializar primero como se observa a continuación:

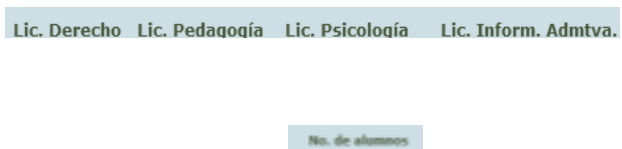
```
$strXML = "";
```

Una vez que se inicializó se procede a capturar información en dicha variable como se observa:

```
$strXML = "<chart caption = 'Lic. Derecho' Lic. Pedagogía' Lic. Psicología' Lic. Inform. Edmva.' bgColor='#FFFFFF' baseFontSize='12' showValues='1' dataName='No. de alumnos'";
```

Con la sentencia `chart caption` se colocan los encabezados o títulos que llevarán las gráficas o en este caso las barras (animadas) que se formarán, para éste ejemplo se hace

referencia a los nombre de las carreras de que está compuesto el sistema escolar (lic. derecho, lic. pedagogía, lic. psicología y lic. Informática Administrativa), por último se colocan las sentencias que darán color (**bgcolor**), tamaño de letra (**basefontsize**), desplegar los valores (**showvalues**) que éste caso fue uno solo pues es una sola línea de títulos la que se captura, y en el parámetro del lado del eje de la X (**xAxisName**) la leyenda No. de alumnos, todo esto se puede apreciar en la siguiente salida:

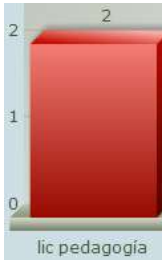


Como se puede observar a pesar de estar en lugares distintos las leyendas tienen las mismas características de tamaño y color pues están como se observó anteriormente sobre la misma sentencia **\$strXML**.

Ahora se procederá a dibujar las barras (animadas) que desplegarán los datos del total de alumnos por carrera (que se hizo en la sentencia de la consulta utilizando un count) utilizando los siguientes argumentos:

```
$nada='';
$strXML .= "<set label = '$nada' value = '". $intTotalAnio1.'" color = 'EA1000' />";
$strXML .= "<set label = '$nada' value = '". $intTotalAnio2.'" color = '0000FF' />";
$strXML .= "<set label = '$nada' value = '". $intTotalAnio3.'" color = '6D8D16' />";
$strXML .= "<set label = '$nada' value = '". $intTotalAnio4.'" color = 'FFBA00' />";
```

La variable **\$nada** de php sirve para poder almacenar alguna leyenda o dato numérico que se desee desplegar (p.e **\$nada=Lic. Pedagogía;**), por ejemplo se pudo haber colocado el nombre de cada carrera debajo de cada barra por ejemplo:



Pero se prefirió que las leyendas se colocaran arriba (como se explicó anteriormente) por lo que la variable \$nada queda vacía, al quedar vacía simplemente no se despliega leyenda alguna por debajo de cada barra (como se observará más adelante) .

Ahora la variable \$strXML desplegará la leyenda que podría llevar por debajo de cada barra (animada) que se vaya a desplegar y que como se explicó anteriormente no se colocará nada, después en value se desplegará lo que contenga cada variable \$intTotalAnio1 que es producto del resultado de la consulta hecha anteriormente, pasando ahora a elegir un color para la barra que se dibuja y con eso diferenciar entre una carrera y otra (una barra y otra) el total de alumnos por carrera aquí representados utilizando color="EA1000";

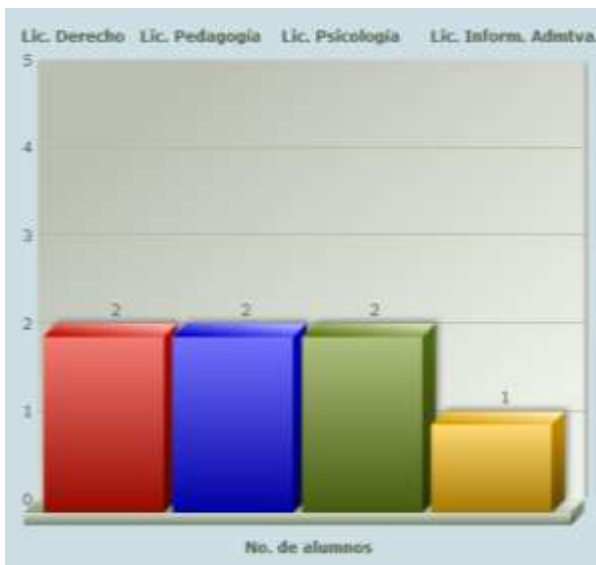
Por último se cierra la etiqueta chart:

```
$strXML .= "</chart>";
```

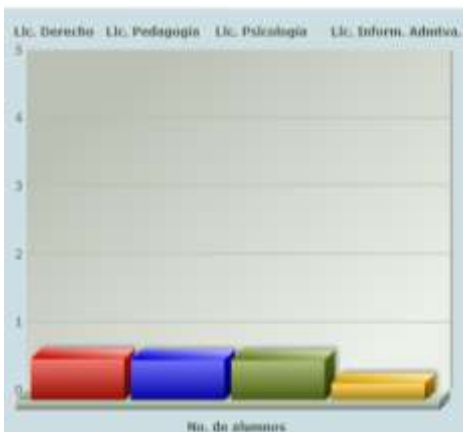
Ya para terminar se renderiza la imagen animada que se va a presentar que es la gráfica en cuestión utilizando los siguientes argumentos:

```
echo("<center>");
echo renderChartHTML("../imagen/Column3D.swf", "", $strXML, "ejemplo", 480, 450, false);
echo("</center>");
```

Como se observa la sentencia echo (“<center>”); permite centrar la gráfica animada que se va a crear (rebderChartHTML) donde se apoya en la carpeta imagen y el archivo Column3D.swf, para éste ejemplo se le da una tamaño de 480 por 450 y con esto se obtiene la siguiente gráfica:



Como se observa aparecen 4 barras cada una de ellas está desplegando las consultas que se explicaron en las líneas anteriores y cada una de ellas contienen colores diferentes como se había mencionado y en la parte superior aparecen los resultados (en número) derivado de la consulta hecha a la base de datos, se puede apreciar muy bien la animación de la gráfica por obvias razones que este es un documento de texto pero se anexa la siguiente imagen:



Aquí se aprecia como se va dando el crecimiento (animación) de cada barra hasta llegar a su límite final en base a la lectura de las carreras tomadas en la base de datos y desplegar su valor final:



Resultados y discusión.

Una vez terminado el análisis, desarrollo e implementación de los scripts de la librería, se procederá a comprobar la variable por medio del siguiente cuestionario:

| | Nombre | Resultados |
|-------|---|------------|
| 1. | Docentes academia de Lic. Informática | 25 |
| 2. | Docentes academia de g. stern ; Comouta onales. | 20 |
| 3. | Alumnos de Lic. InformáLic. | 30 |
| 4. | Alumnos de Ing. Sist. CoLic. | 20 |
| TOTAL | | 80 |

$95 / 4 = 23.75.$

Resultado en la escala Likert

20 25 30 35 40

SI fácil utilizar la librería. **Cubre** al 50% **NO** es fácil utilizar la librería..

Se observa que al desarrollar scripts utilizando éstos ejemplos ilustrativos desarrollados en ésta investigación es igualmente sencillo utilizar FusionCharts.php representar gráficas animadas tomando información de base de datos.

Trabajo a futuro

Se espera desarrollar scripts que conectados a un base de datos demuestren el manejo de generación de gráficas pero que la información que tome de la base de datos sea tomada de consultas hechas por formularios múltiples cuya condición emplee el conectivo y aplicable a dos o más tablas.

Conclusión

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 la librería FusionCharts.php al momento de entender cómo se alimentan sus argumentos de la librería derivados de consultas hechas a la base de datos

Bibliografía

1. Kenneth e. Kendall (2005). Análisis y diseño de sistemas. 6th. Ed. School of Business-Camden Camden, New Jersey.
2. Stig Sæther (2002). Manual de PHP. 2th. Ed.
3. Ian Gilfillan ().
4. C.J. Date (2001). Introducción a los sistemas de base de datos. 7th. Ed. Reading Massachusetts. E.U.A.