

INSTITUTO TECNOLÓGICO SUPERIOR DE CHICONTEPEC.

INGENIERIA EN SISTEMAS COMPUTACIONALES.

ASIGNATURA:

Programación web.

ALUMNO:

Flor Hernández Cruz.

DOCENTE:

Ingeniero Efrén Flores Cruz.

UNIDAD 2:

HTML, XML y CSS.

TRABAJO:

Reporte de unidad 2.

SEMESTRE:

8º

FECHA DE ENTREGA:

31 marzo 2020.

1.- Realizar resumen sobre PHP, APACHE Y MYSQL.

PHP son las siglas de Hypertext Pre-Processor que al traducirlo al español significa "Lenguaje de Programación Interpretado". Este lenguaje es el contenido dinámico que visualizamos en las páginas web. Todo código PHP es invisible para el usuario, porque todas las interacciones que se desarrollan en este lenguaje son por completo transformadas para que se puedan ver imágenes, variedad de multimedia y los formatos con los que somos capaces de interactuar añadiendo o descargando información de ellos.

La internet ofrece una variedad de formas de comunicación que muchas de ellas no conocemos. Fue creado en el año 1994, bajo la premisa de ser un código libre destinado a la comunidad desarrolladora de programas que alimentarían a la red de internet.

Las interfaces gráficas en las que se maneja son denominadas bibliotecas Qt y GTK, con esto, los lenguajes de programación estructurada más comunes como el C y Perl Pueden desarrollar aplicaciones y programas para la red sencillos de interpretar y compatibles con cualquier tipo de Servidor y sistema operativo vigente aún en el mercado. PHP actúa fuertemente con archivos de toda clase, desde lo más básico como .docx, .PDF, .JPG hasta los flash que son animaciones cargadas en la red. PHP es el lenguaje de programación de aplicaciones interactivas más fácil de usar y por más de 20 años no se ha reemplazado.

Scribe

PHP es un lenguaje de programación de lado del Servidor gratuito e independiente de plataforma.

Un lenguaje del lado del Servidor es aquel que se ejecuta en el Servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que conectan en el servidor pueden realizar accesos a la BD, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como página resultante contiene únicamente código HTML, es compatible con todos los navegadores. PHP se escribe dentro del código html



APACHE

Apache HTTP Server es un software de servidor web gratuito y de código abierto para plataformas Unix con el cual se ejecutan el 46% de los sitios web de todo el mundo. Es mantenido y desarrollado por la Apache Software Foundation.

Le permite a los propietarios de sitios web servir contenido en la web. Uno de los servidores web más antiguos y confiables, la primera versión lanzada el

en el año 1995. Cuando alguien quiere visitar un sitio web, ingresa el nombre de dominio en la barra de direcciones de su navegador. Luego, el servidor web envía los archivos solicitados actuando como un repartidor virtual.

Apache no es un servidor físico, sino un software que se ejecuta en un servidor. Su trabajo es establecer una conexión entre un servidor y los navegadores de los visitantes del sitio web (Firefox, Google Chrome, Safari) mientras envían archivos entre ellos (estructurado cliente-servidor). Apache es un software multiplataforma, por lo cual funciona tanto en servidores Unix como en Windows.

Ventajas de Apache

- 1.- Código abierto y gratuito, incluso para uso comercial.
- 2.- Software confiable y estable
- 3.- Parches de seguridad regulares y actualizados con frecuencia.
- 4.- Flexible debido a que su estructura está basada en módulos.
- 5.- Fácil de configurar
- 6.- Multiplataforma
- 7.- Viene listo para trabajar con sitios de WordPress
- 8.- Enorme comunidad y soporte fácilmente disponible en caso de cualquier problema.

MySQL

MySQL es el sistema de gestión de BD relacional más extendido en la actualidad al estar basado en código abierto, desarrollado por MySQL AB, adquirido por Sun Microsystems en 2008 y esta su vez comprada por Oracle Corporation en 2010, la cual ya era dueña de un motor propio InnoDB para MySQL.

MySQL es un Sistema de gestión de base de datos que cuenta con una doble licencia, es de código abierto.

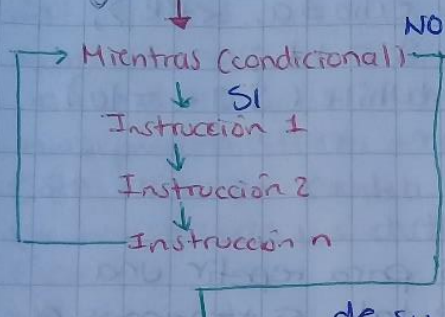
Características MySQL

- 1.- Arquitectura cliente y servidor: Clientes se comunican entre sí de manera diferenciada para un mejor rendimiento.
- 2.- Compatibilidad con SQL: Lenguaje generalizado dentro de la industria. Si se ha trabajado con otros SGBD se puede migrar a MySQL.
- 3.- Vistas: Se puede configurar vistas personalizadas del mismo modo para hacerlo en otras BD SQL.
- 4.- Procedimientos almacenados: Incrementa la eficacia de implementación.
- 5.- Desencadenantes: Permite a los demás poder automatizar ciertas tareas dentro de la BD.
- 6.- Transacciones: Actuación de diversas operaciones en la BD como un dispositivo.

2.- Realizar ejercicios en php sobre bucles =

WHILE

Es la estructura iterativa más sencilla en lo que se refiere a uso. Su uso es para mostrar datos de una tabla en MySQL o filtrar de un determinado rango de números y/o fechas de nuestra BD



1.- Al ejecutar instrucción WHILE en PHP se comprueba su condición

2.- Si la condición se cumple se ejecuta el contenido del bucle hasta llegar a lo último

3.- Al terminar el último instrucción

de su contenido se vuelve arriba y comprueba con la

4.- Cuando la condición del bucle deje de cumplirse el bucle termina

Ejemplo:

```
<? php
$numero = 5 //ejemplo de factorial número 5
$factorial = 1;
while ($numero > 1) {
    $factorial = $factorial * $numero;
    $numero = $numero - 1;
}
echo 'El factorial de ' . $numero . ' es ' . $factorial.
?>
```


DO WHILE

Garantiza la ejecución de las sentencias interna al menos una vez, dado que la validez de la expresión a evaluar se efectúa hasta el final de la sentencia

Ejemplo:

```
Do {  
    Sentencias  
} WHILE (expresión)  
  
<?php  
$a = 1;  
do {  
    echo $a. "<br>";  
    $a++;  
} while ($a <= 10)  
?>
```

FOR

Uno de los ciclos más usados para repetir una secuencia de instrucciones, sobre todo si tenemos el límite de cantidad exacta de veces que se quiere repetir la instrucción.

Ejemplo imprime números de 1-15

```
<?php
```

```
for($i=1; $i<15; $i++) {  
    echo $i; // Imprime ciclos  
    echo "<br>";  
}  
?>
```

FOREACH

Es una estructura especializada en recorrer los elementos que contiene una variable, es por eso que está especialmente indicado para ver o recorrer todos los elementos de un array.

Su sintaxis simplifica la forma de recorrer un array, ya que además de calcular automáticamente el número de elementos que tiene el array (número de repeticiones), nos proporciona una variable (2 de los arrays asociativos). Esta variable cambia con cada repetición, de manera que va mostrando los distintos elementos del array.

Ejemplo

```
<?php
```

```
$Semana = array ("lunes", "martes", "miercoles",  
"jueves", "viernes", "sabado", "domingo");
```

```
foreach ($Semana as $dia) {  
    echo "$dia, ";  
}
```

```
?>
```

CONDICIONAL IF

Permite ejecutar un fragmento de código si la expresión provista junto con ella se evalúa como verdadero.

Ejemplo:

```
<?php
```

```
$age = 50;
```

```
if ($age > 30)
```

```
{ echo "Your age is greater than 30! ";
```

```
}
```

```
?>
```


ELSE

Si if es falsa no hará nada, la mayoría de las veces, también desea ejecutar un fragmento de código diferente si la expresión se evalúa falsa, ahí es donde la declaración else participa

```
if (expresión)
{
    // código q se ejecuta si if es verdad
}
else
{
    // código si if es falso
}
```

If-else Tiene más de 2 opciones para elegir

```
< ? php
$age = 50;
if ($age < 30)
{
    echo "Your age is less than 30!";
}
elseif ($age > 30 & $age < 40)
{
    echo "Your age is between 30 and 40!";
}
else
{
    echo "Your age is greater than 50!";
}
? >
```

5.- Resumen del funcionamiento del

Método POST Y GET

Existen 2 métodos con los que el navegador envía información al servidor.

- * Método HTTP GET. Información se envía de forma visible

- * Método HTTP POST. Información se envía de forma no visible

Antes de que el navegador envíe la información proporcionada, la codifica mediante URL encoding, dando como resultado un Query String. Esta codificación es un esquema de keys y valores separados por un ampersand &.

Key1 = value1 & Key2 = value2 & Key3 = value3...

Los espacios y otros caracteres no alfanuméricos se sustituyen. Una vez que la información es codificada, se envía al servidor.

— Método HTTP GET —

El método get envía la información codificada del usuario en el header del HTTP request, directamente en la URL. La página web y la información codificada se separan por un interrogante ? :

www.ejemplo.com/index.htm?Key1=value1&key2=Value3...

- * El método GET envía la información en la propia URL estando limitada a 2000 caracteres

- * La información es visible por lo que con este método nunca se envía información sensible.

- * No se pueden enviar datos binarios (archivos, imágenes).

- * En PHP los datos se administran con el array asociativo \$_GET.

Scribe

Ejemplo:

```
<html?
<body>
<form action = "formget.php" method = "get">
Nombre: <input type = "text" name = "nombre"> <br>
Email: <input type = "text" name = "email"> <br>
<input type = "Submit" Value = "Enviar">
</form>
Hola <?php isset($_GET["nombre"]) ? print
$_GET["nombre"] : "" ; ?> <br>
Tu email es: <?php isset($_GET["email"]) ?
print $_GET["email"] : "" ; ?>
</body>
</html>
```

La url que resulta de hacer click en Submit es de la forma: `formget.php? nombre=peter& email=peter%40ejemplo.com`
En este caso @ es un carácter especial y se codifica

MÉTODO HTTP POST

Con el método HTTP POST también se codifica la información, pero ésta se envía a través del body del HTTP Request, por lo que no aparece en la URL

- * El método POST no tiene límite de cantidad de información a enviar
- * La información proporcionada no es visible, por lo que se puede enviar información sensible.

- * Se puede usar para enviar **texto normal** así como **datos binarios** (archivos, imágenes...).
- * PHP proporciona el array asociativo `$_POST` para acceder a la información enviada.

```
<html>
<body>
<form action = "formpost.php" method = "post">
  Nombre = <input type = "text" name = "nombre"> <br>
  Email: <input type = "text" name = "email"> <br>
  <input type = "submit" value = "Enviar">
</form>
Hola <?php isset($_POST["nombre"])? print $_POST
["nombre"]:" " ?><br>
Tu email es: <?php isset($_POST["email"]) print
$_POST["email"]:" " ; ?>
</body>
</html>
```