



Técnico Desarrollador de Software

MODULO IX PHP



GRETEL ITZEL GONZÁLEZ GARCÍA

gretelgonzalezg@gmail.com

Objetivo General

- ▶ Que los alumnos aprendan a programar en lenguaje de PHP para la creación de paginas web dinámicas en el entorno de desarrollo de Eclipse y que aprendan a realizar conexión a bases de datos mediante MySQL.



Reglas

- ▶ NO redes sociales
 - ▶ NO se permite entrar con alimentos o bebidas
 - ▶ Mantener el área de trabajo limpia
 - ▶ Celular el silencio
-
- ▶ Receso de 30 minutos
 - ▶ A las 2:00 pm

Evaluación

- ▶ Practicas 50%
- ▶ Examen 10%
- ▶ Proyecto 40%

MANUAL:

- ▶ Técnico desarrollador en software,
Modulo IX, PHP

FALTAS:

- ▶ 2 faltas pierde derecho a 2da oportunidad
- ▶ Se podrá justificar 1 falta



Temario

- ▶ Interfaz de eclipse-PHP
- ▶ Crear un proyecto en PHP en eclipse
- ▶ Crear un archivo en PHP
- ▶ Mi primer página en PHP
- ▶ Vista previa de la pagina PHP
- ▶ Código fuente PHP
- ▶ Trabajando con formularios PHP
- ▶ Conectar con Mysql

¿Qué es PHP?

- ▶ PHP (Hypertext Preprocessor): Preprocesador de Hipertexto.
- ▶ Lenguaje de código abierto.
- ▶ Lenguaje de script del lado del servidor.
- ▶ Los script de PHP están incrustados en los documentos HTML y el servidor los interpreta y ejecuta antes de servir las paginas al cliente.
- ▶ Es transparente; es decir el cliente no ve el código PHP sino los resultados que produce.



¿Cómo funciona?

- ▶ A diferencia de Java o Javascript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que se tengan en el servidor por ejemplo una base de datos.
- ▶ El programa PHP se ejecuta en el servidor y el resultado en el navegador

¿Qué se puede hacer con PHP?

- ▶ Esta enfocado principalmente a la programación de script del lado del servidor, por lo que se pueden hacer cosas como recopilar información en formularios.
- ▶ Generar paginas web con contenido dinámico
- ▶ Enviar y recibir cookies.

¿Qué es Eclipse?

- ▶ Programa: es un software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar Aplicaciones de clientes Enriquecidos.
- ▶ Desarrollado por IBM en noviembre de 2001
- ▶ Comunidad: es una comunidad de personas y organizaciones que deseen colaborar en el desarrollo del software libre. Sus proyectos se centran en la construcción de una plataforma de desarrollo abierta formada por marcos extensibles, herramientas y tiempos de ejecución para la construcción, despliegue y gestión de software.
- ▶ Fundación: la fundación Eclipse es una organización sin fines de lucro, corporación que apoya a miembros que realizan proyectos de Eclipse y ayuda a cultivar tanto una comunidad de código abierto y un ecosistema de productos y servicios complementarios.
- ▶ Versión mas actual es: "LUNA"

XAMPP

Es un programa que nos ayuda como servidor independiente de plataforma, software libre.

Contiene los siguientes programas:

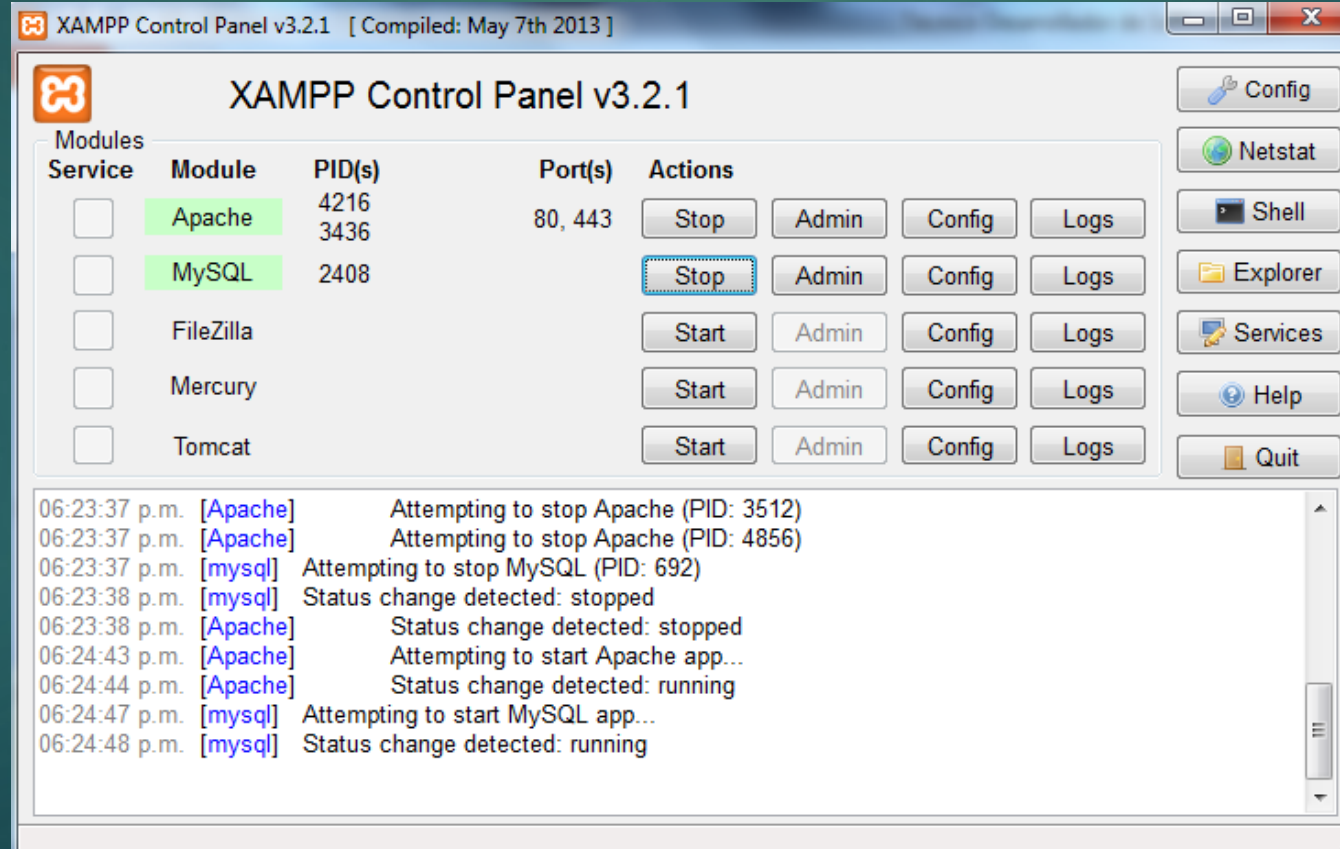
- ▶ **APACHE**: servidor web
- ▶ Interprete para los lenguajes de script PHP y Perl
- ▶ **MySQL**: programa gestor de bases de datos
- ▶ **phpMyAdmin**: interfaz visual para interactuar con esas bases de datos.



XAMPP

Control Panel de XAMPP

- ▶ Aquí encenderás APACHE y MySQL para poder trabajar con PHP y bases de datos, solo es necesario dar clic en el botón “**start**” que se encuentra del lado derecho de la palabra Apache y si no existe ningún problema la palabra Apache se subrayara de color verde.



Sintaxis

- ▶ `<? Php` Declara el comienzo del código de PHP

- ▶ `?>` Declara el cierre de código PHP

- ▶ Comentarios:

Una sola línea se representan con `//`

Mas de una línea se inician con `/*` y se termina con `*/`

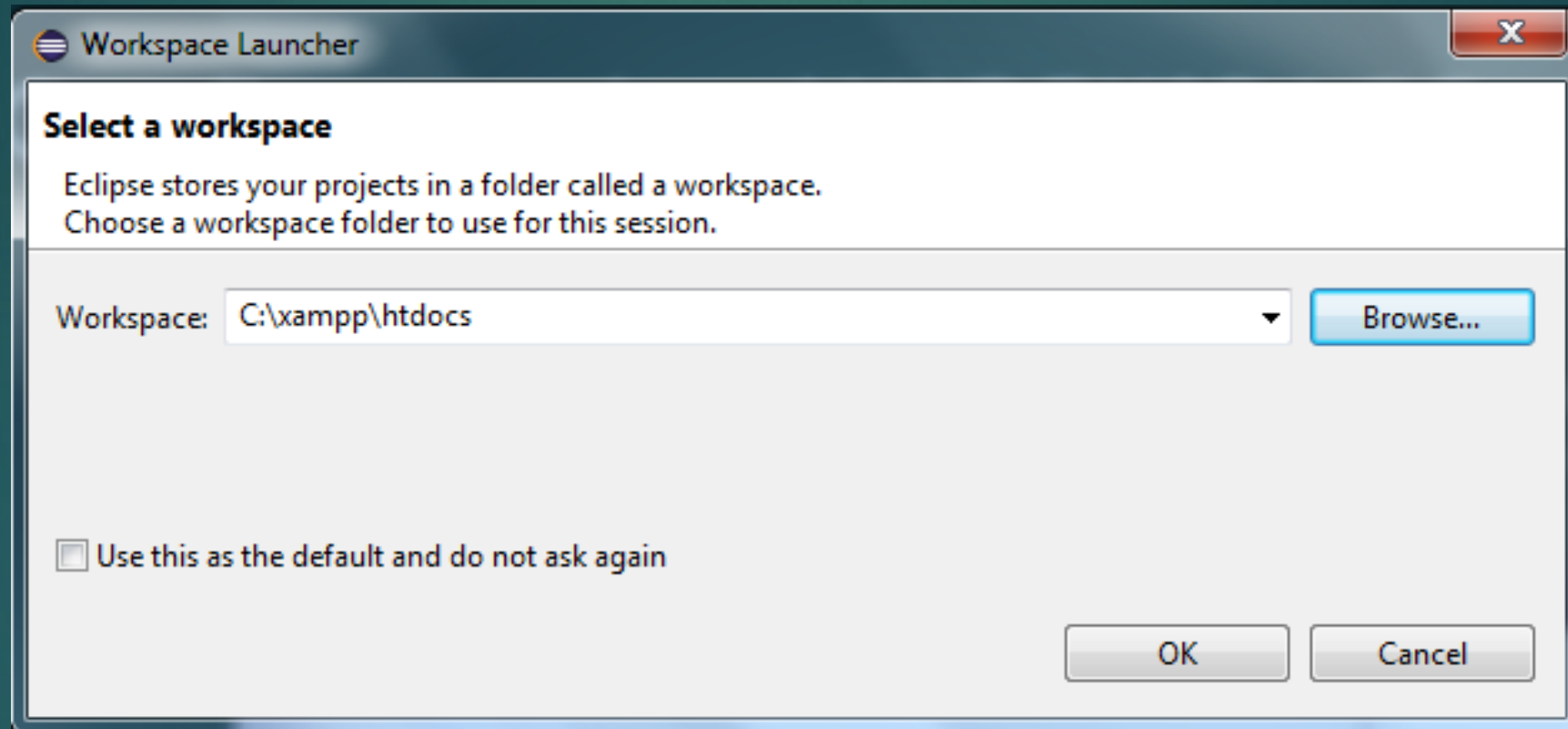
Antes de comenzar a programar

- ▶ Recuerda siempre antes de empezar encender el servidor **APACHE** en el XAMPP, ya que sin este no podrás visualizar tu código PHP.
- ▶ De igual manera para trabajar con bases de datos tienes que iniciar **MySQL** en el XAMPP



Configurando el WORKSPACE

- ▶ Al iniciar Eclipse por primera vez aparece una ventana en la cual nos pide que elijamos una dirección para nuestra carpeta de trabajo. Por default eclipse hace una carpeta llamada workspace en nuestros documentos.
- ▶ Es importante que nosotros demos clic en el botón “**Browse...**” y busquemos el directorio donde se encuentra nuestra carpeta de XAMPP demos doble clic y elijamos la carpeta llamada “**htdocs**”
- ▶ Esta carpeta llamada **htdocs** es donde se guardaran nuestros proyectos y documentos dentro del servidor , por eso es importante hacer esta configuración para poder visualizar nuestros archivos en el navegador.



- ▶ De no realizar esta configuración, nos podría salir el siguiente error.

Ya que al momento de correr nuestro archivo este no se encuentra en el servidor

Objeto no localizado!

El URL solicitado no ha sido localizado en este servidor. Si usted tecleó el URL manualmente, por favor revise su ortografía y vuélvalo a intentar.

Si usted cree que esto es un error del servidor, por favor comuníquese al [administrador del portal](#).

Error 404

[localhost](#)

Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.15

¿Cómo crear un proyecto en eclipse?

- ▶ Cuando decimos un proyecto se refiere a la creación de una carpeta específica donde nosotros almacenaremos nuestros archivos.
- ▶ Podemos encontrar las instrucciones de como crear un proyecto en la pagina 8 del manual “Técnico desarrollador en Software” módulo IX PHP

¿Cómo crear un archivo PHP?

- ▶ Un archivo hace referencia al lugar donde escribiremos nuestro código fuente de PHP.
- ▶ Estos archivos los almacenaremos dentro de carpetas para tener un mejor control de nuestro proyecto y también nos ayudara para tener una mejor estructura.
- ▶ Podemos encontrar las instrucciones de como crear un proyecto en la pagina 9 del manual “Técnico desarrollador en Software” módulo IX PHP



```
<?php
```

```
//Mi prier script
```

```
    echo "Hola mundo";
```

```
?>
```



Mi primer Script “Hola Mundo”

- ▶ Crearemos un proyecto llamado “**Clase-1**”
- ▶ Ahora crearemos un archivo llamado “**Hello Word**”
- ▶ Y del lado derecho en el área de trabajo teclearemos el siguiente código

```
<html>
  <head><title>PHP interaccion con HTML</title></head>
  <body>
    <?php
      echo "Hola mundo";
    ?>
  </body>
</html>
```

Imprimir datos en pantalla con la sentencia echo y print()

- ▶ Print(): es una función de PHP que indica al software que escriba o imprima en pantalla lo que se encuentra dentro de sus paréntesis.
- ▶ echo: es una sentencia que también sirve para imprimir en pantalla.

La diferencia de estas 2 sentencias únicamente es su sintaxis, ya que ambas sentencias funcionan de igual manera y tienen el mismo propósito.

- ▶ Ambas pueden ser utilizadas con comidas dobles o simples, pero a diferencia de **echo**, la función **print()** el texto que deseemos mostrar en pantalla se debe poner entre paréntesis.

```
<html>
  <head>
    <title>Imprimir datos en pantalla</title>
  </head>
  <body>
    <?php
      echo "Bienvenidos al Modulo 9 - PHP, <br>";
      echo 'Tecnico Desarrollador en Software, <br>';
      print ("Facultad de Ingenieria Mecanica y Electrica;<br>");
      print ('Universidad Autonoma de Nuevo León');
    ?>
  </body>
</html>
```

Ejercicio practico

- ▶ Realizar un código que imprima en pantalla los siguientes datos:

“Técnico desarrollador de software”

“Modulo 9”

“Nombre completo”

“Matricula”

- ▶ El código deberá de contener las 2 sentencias para imprimir datos vistas anteriormente

Variables

- ▶ Las variables son espacios de memoria que sirven para almacenar datos que pueden variar durante el progreso del programa, en PHP deben estar precedidas por signo de dólar (\$), y le asignamos contenido con el signo igual (=).
- ▶ Pueden ser con valor String, valor numérico, booleanos

```
<html>
  <head>
    <title>Variables</title>
  </head>
  <?php
    header('Content-Type: text/html; charset=UTF-8');

    $boolean1 = true;
    $boolean2 = false;
    $entrada = "INPUT";
    $salida = "OUTPUT";
    $edad = 22;
    $nombre = "Gretel";
    echo "Hola mi nombre es ".$nombre. " y tengo " . $edad. " años";
    echo "<br>";
    echo "<br>";
    echo "Haremos la tabla de verdad de la puerta logica AND";
    echo "<br>";
    echo "<br>";
  ?>
  <body>
    <table border="2px">
      <tr>
        <td colspan="2"><?php echo $entrada ?></td>
        <td colspan="2"><?php echo $salida ?></td>
      </tr>
      <tr>
        <td>A</td>
        <td>B</td>
        <td colspan="2">A AND B</td>
      </tr>
      <tr>
```

```
        <td>0</td>
        <td>0</td>
        <td colspan="2">0</td>
    </tr>
    <tr>
        <td>0</td>
        <td>1</td>
        <td colspan="2">0</td>
    </tr>
    <tr>
        <td>1</td>
        <td>0</td>
        <td colspan="2"><?php echo $boolean2 ?></td>
    </tr>
    <tr>
        <td>1</td>
        <td>1</td>
        <td colspan="2"><?php echo $boolean1?></td>
    </tr>
</table>
</body>
</html>
```

- Php distingue entre mayúsculas y minúsculas, por lo que no es lo mismo \$myvar que \$Myvar, estas son variables totalmente distintas.

```
<html>
  <head><title>Variables</title></head>
  <body>
    <?php
      /*aqui se puede ver que PHP distingue ayusculas de inusculas*/

      $myvar = "SEVILLA";
      $Myvar = "MADRID";

      echo $myvar."<br>";
      echo $Myvar."<br>";
    ?>
  </body>
</html>
```

► Realizando operaciones con variables

```
<html>
  <head><title>Variables</title></head>
  <body>
    <?php

      /*En este codigo vereos las variables y los tipos de variables que existen*/

      $num1 = 4;
      $num2 = 4;
      $num3 = 10;
      $result = $num1 + $num2 + $num3;

      echo "El resultado es ".$result

    ?>
  </body>
</html>
```

Ejercicio practico

- ▶ Realizar un código que imprima en pantalla que contenga 3 o mas variables de tipo numérico y muestre en pantalla solo el resultado de estas variables.

Constantes

- ▶ Las constantes son similares a las variables, con la salvedad de que no llevan el signo dólar delante, y sólo la podemos asignar una vez.
- ▶ Para definir una constante usaremos la función **define**:
- ▶ **Concatenar**: método por el cual podemos unir varios datos.
- ▶ Se utiliza el punto (.)
- ▶ Esto se hace con la finalidad de no escribir tanto código.

```
<html>
  <head><title>Concatenacion</title></head>
  <body>
    <?php

        /*Concatenar Constantes*/

        define("saludo", "Hola como estas?<br>");
        define("numero", 40);

        echo saludo.numero;
    ?>
  </body>
</html>
```


Arrays o vectores

- ▶ Nos permiten organizar elementos en forma de lista y podemos acceder a ellos mediante una **clave** «predefinida» o también podemos agregar una clave «personalizada»

▶ `$array = array("elemento1", 2, "elemento3");`

Variable
Nombre del array



Elementos



Ejercicio practico

- Declarar arrays predefinidos

```
<?php
```

```
$array = array("elemento1", 2, "elemento3");
```

```
echo $array[0];
```

```
?>
```

*Clave predefinida del
elemento al que queremos
accesar*

*Claves predefinidas
O posición*

Ejercicio practico

- Declarar arrays personalizados

```
<?php
```

```
$asociativo = array("clave 1" => "elemento1", "clave2" => 2, "clave3" => "elemento3");
```

```
echo $asociativo["clave1"];
```

```
?>
```

Claves personalizadas para acceder a nuestros elementos

Operadores

- ▶ Las operadores en PHP son aquellos que nos permiten realizar alguna operación ya sea matemática, asignación etc.
- ▶ Los tipos de operadores son:
 - ▶ **Aritméticos**: nos permiten realizar operaciones aritméticos como suma, resta, multiplicación, etc. (+ - * /)
 - ▶ **Asignación**: estos nos permiten asignarle un valor a una variable o algún tipo de dato. (=)
 - ▶ **Cadena o concatenación**: nos permite unir o concatenar varios tipos de datos (.)

```
$adicion = 4 + 4;  
$multiplicacion = $adicion * 2;  
  
echo $adicion;  
echo "<br>";  
echo $multiplicacion;
```

Aritméticos

- ▶ $\$a + \b suma
- ▶ $\$a - \b resta
- ▶ $\$a * \b multiplicación
- ▶ $\$a / \b división
- ▶ $\$a++$ incrementa 1 en 1
- ▶ $\$a--$ resta 1 en 1

Comparadores

- ▶ $\$a < \b \$a menor que \$b
- ▶ $\$a > \b \$a mayor que \$b
- ▶ $\$a \leq \b \$a menor o igual
- ▶ $\$a \geq \b \$a mayor o igual que \$b
- ▶ $\$a == \b compara si \$a es igual que \$b
- ▶ $\$a != \b compara si \$a es distinto que \$b

Lógicos

- ▶ $a \text{ AND } b$ verdadero si ambos son verdaderos
- ▶ $a \ \&\& \ b$ verdadero si ambos son verdaderos
- ▶ $a \text{ OR } b$ verdadero si alguno de los dos son verdaderos o si ambos son verdaderos
- ▶ $a \ || \ b$ verdadero si alguno de los dos son verdaderos o si ambos son verdaderos
- ▶ $a \text{ XOR } b$ verdadero si solo uno de los dos es verdadero, pero no ambos