

Conociendo PHP



*Introducción a uno de
los lenguajes de
programación web
más usados en
Internet*

*LSI. Angel Veloz Rodríguez, Mgs.
angel.velozr@ug.edu.ec*

Audiencia

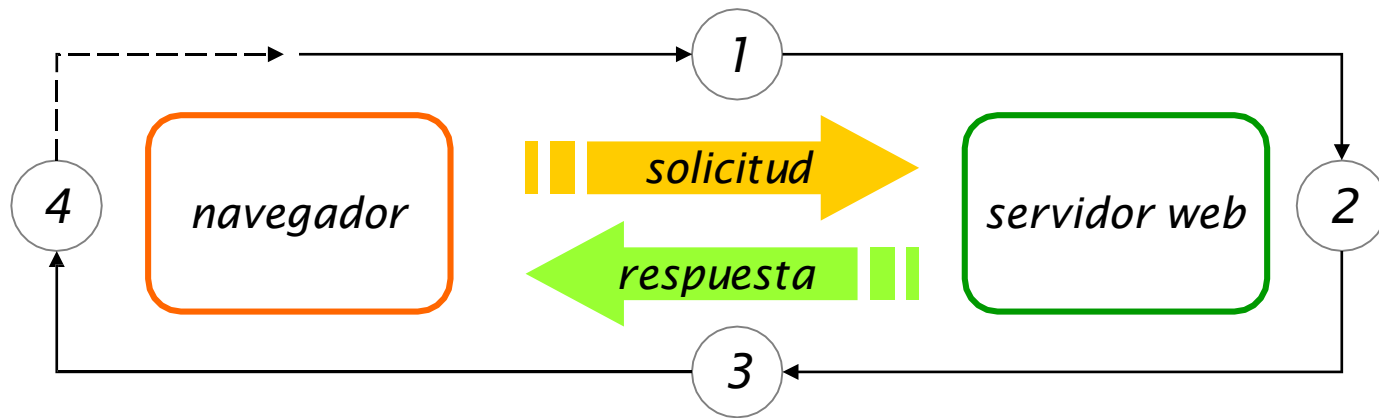
- *Web*
- *Programación*



¿Qué es PHP?

*Lenguaje de programación
para generar páginas web*

El ciclo web



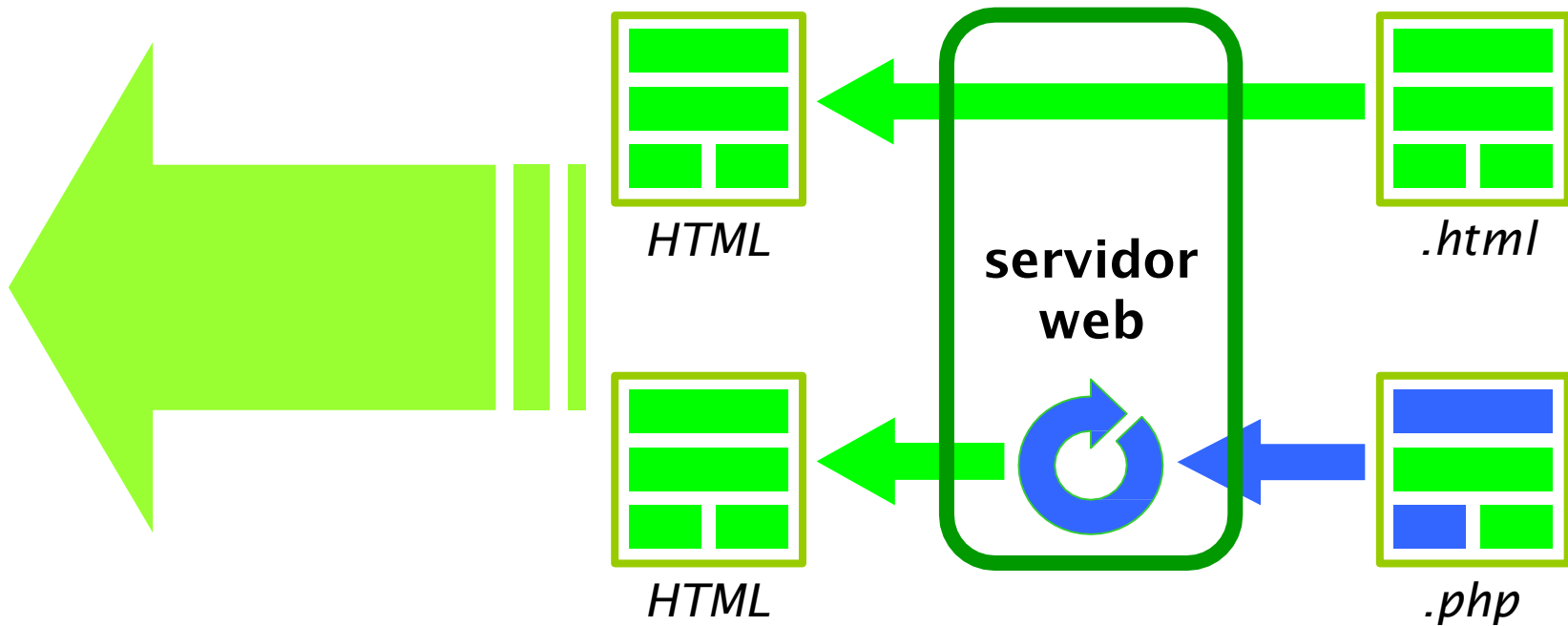
presentación



proceso

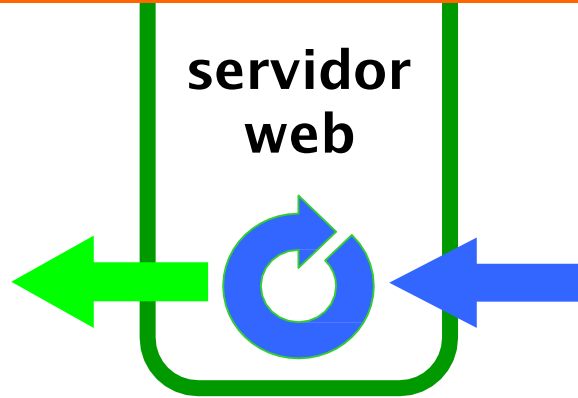
respuesta web

El contenido del documento .html es pasado tal cual a la respuesta HTML



Un documento dinámico (como .php) es procesado para obtener HTML

respuesta web



en el proceso, un documento dinámico puede involucrar diversos recursos



base de datos



email



archivos



gráficos



etc

¿Por qué PHP?

- *Es ampliamente soportado por los servicios de hosting*
- *Es libre*
- *Un número significativo de sites se ha escrito en PHP (Wikipedia , YouTube, Yahoo, Facebook, WordPress...)*
- *Muchos paquetes importantes están escritos en PHP (WordPress, OS Commerce, Joomla, Drupal...)*
- *Es simple, directo, flexible...*
- *Ha evolucionado para soportar programación orientada a objetos*



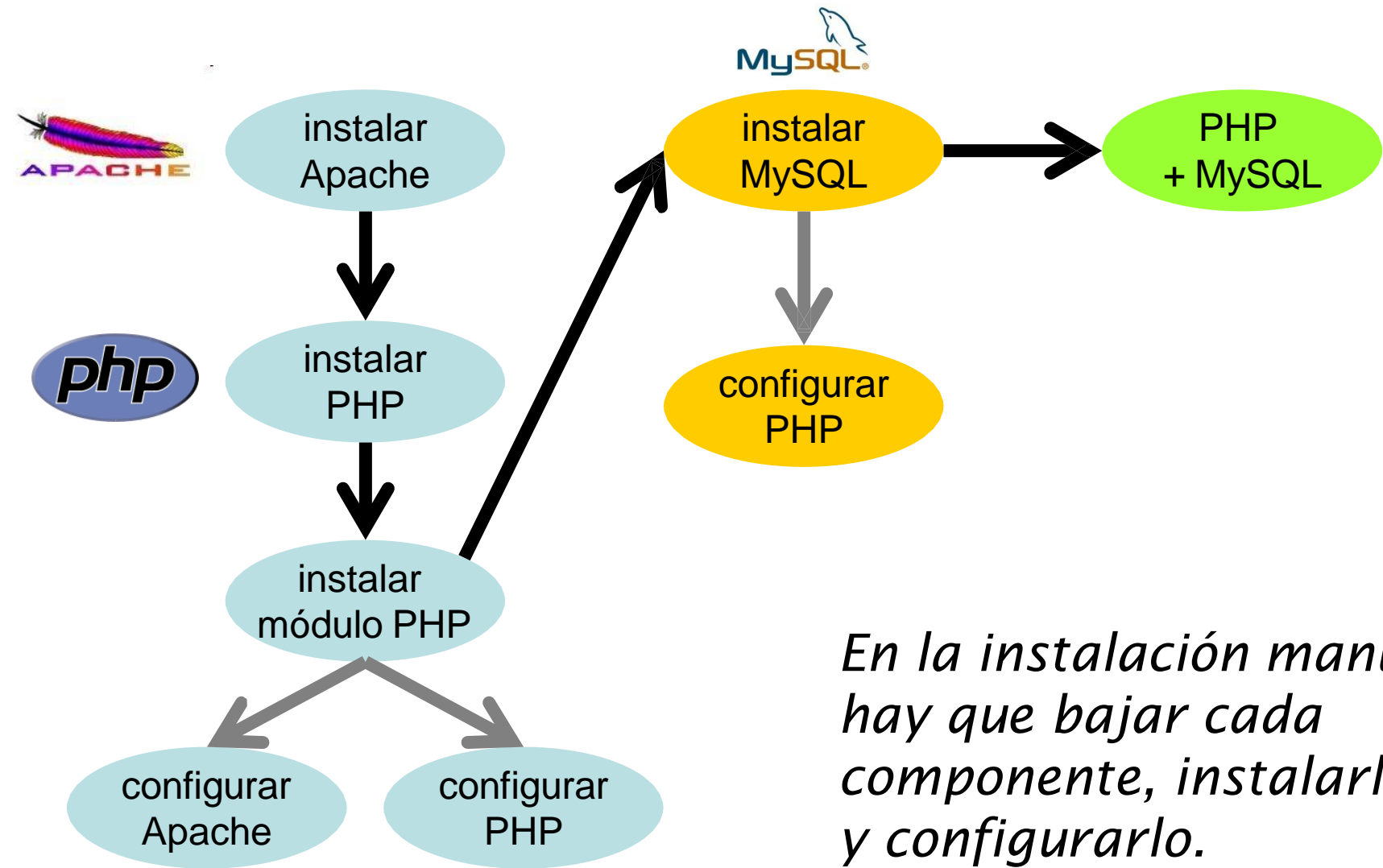
WIKIPEDIA
The Free Encyclopedia



Conociendo PHP

para empezar

instalando PHP



En la instalación manual hay que bajar cada componente, instalarlo y configurarlo.

LAMP, WAMP, ...



Linux



APACHE

A



MySQL

M



php

P



Windows



APACHE

A



MySQL

M



php

P



Mac



APACHE

A



MySQL

M



php

P

- Típicamente, PHP se instala para trabajar junto con Apache y MySQL
- Cuando la instalación aparece en Linux, se la llama LAMP
- Cuando la instalación aparece en Windows, se la llama WAMP
- Y así por el estilo...

XAMPP



- Hay varios proyectos que difunden paquetes que contienen esas instalaciones típicas.
- Usarlos puede facilitar la instalación, y el día a día con el entorno de desarrollo.

- XAMPP es uno de tales paquetes. No sólo provee PHP, sino también Perl y Python y está disponible tanto para Linux, como para Windows, Mac OS X y Solaris



XAMPP

Editores, IDEs



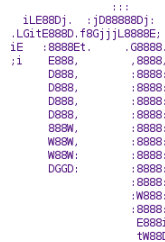
```
...
iLE88Dj. :jD88888Dj:
.LG1tE888D.f8GjjL8888E:
iE :8888Et. :88888.
:1 E888. :8888.
D888. :8888.
D888. :8888.
D888. :8888.
888W. :8888.
W88W. :8888.
D88D. :8888.
:8888.
W888.
E8881
tw88D
```



aptana

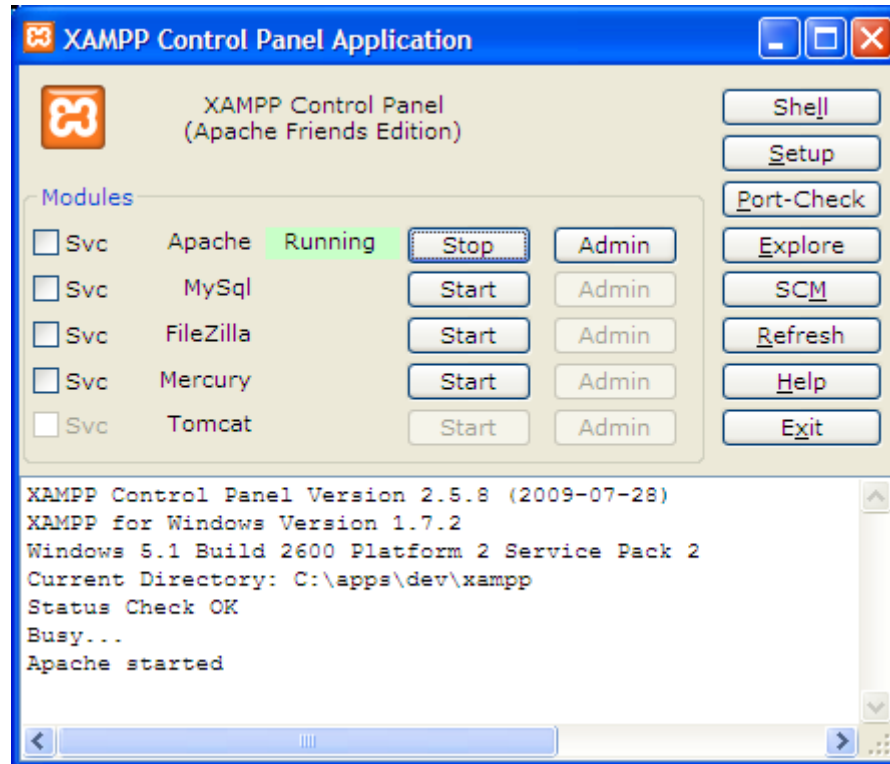
- Igual que con HTML, CSS, Javascript y otros lenguajes web, es posible trabajar usando un simple editor de texto, como Notepad, aunque contar con un entorno de desarrollo puede facilitar mucho las cosas.
- Hay editores de texto muy útiles para programación, como Notepad++, jEdit, ...
- Y entornos de desarrollo muy buenos, como Dreamweaver, Aptana, Eclipse, Zend Studio...

Editores, IDEs



- *Estos son los que suelo usar*
- **Notepad** *cuando no hay nada más a la mano*
- **Notepad++** *es excelente, ligero, con FTP y una buena utilidad para convertir Unicode*
- **Dreamweaver** *es excelente para el diseño HTML, tiene buen soporte para PHP*
- **Aptana** *esta basada en Eclipse, pero tiene mucho mejor soporte para PHP, HTML, Javascript...*
- **Nano**, *cuando estoy Linux y no hay nada más a la mano*

Empezando



El panel de control de XAMPP, desde donde podemos iniciar y detener sus servicios

Conociendo PHP

lo básico

¡Hola Mundo!

`hola_mundo.php`

```
<?php  
echo '¡Hola Mundo!';  
?>
```

- *El código php se escribe entre etiquetas <?php y ?>*
- *Para escribir algo en la página, podemos usar echo*
- *Las cadenas se escriben entre apóstrofes 'así', ó entre comillas "así"*
- *El nombre del archivo tiene la extensión php*

¡Hola Mundo!

hola_mundo.php

```
<?php  
echo '¡Hola Mundo!';  
?>
```

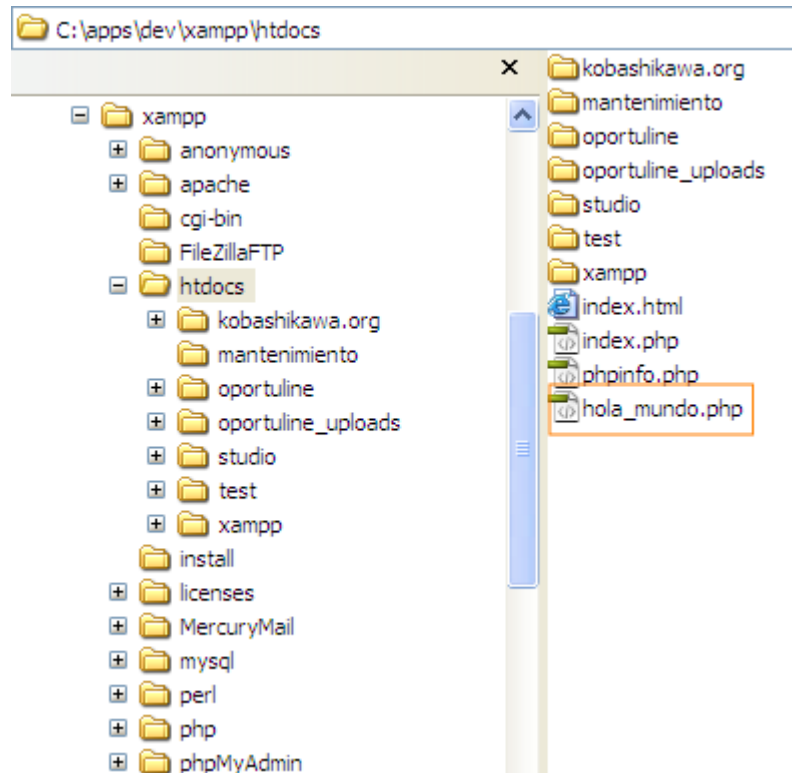


htdocs

...¡Hola Mundo!



htdocs



htdocs/index.php



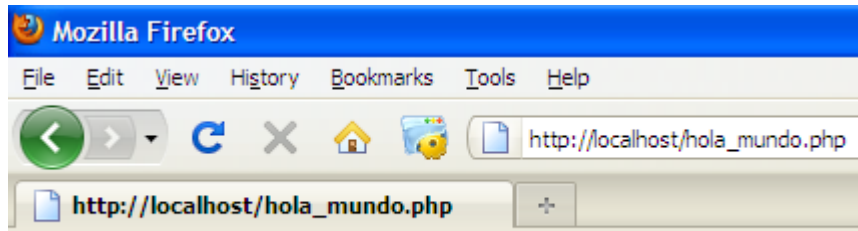
<http://localhost/index.php>

htdocs/hola_mundo.php



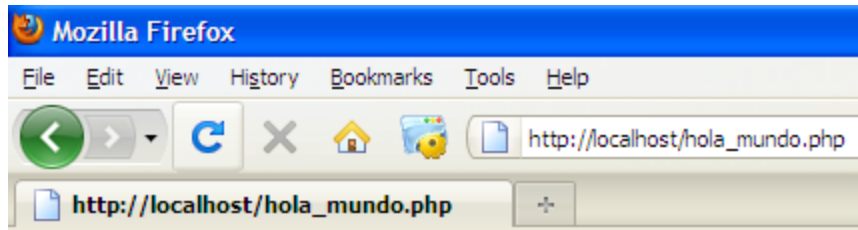
http://localhost/hola_mundo.php

...¡Hola Mundo!



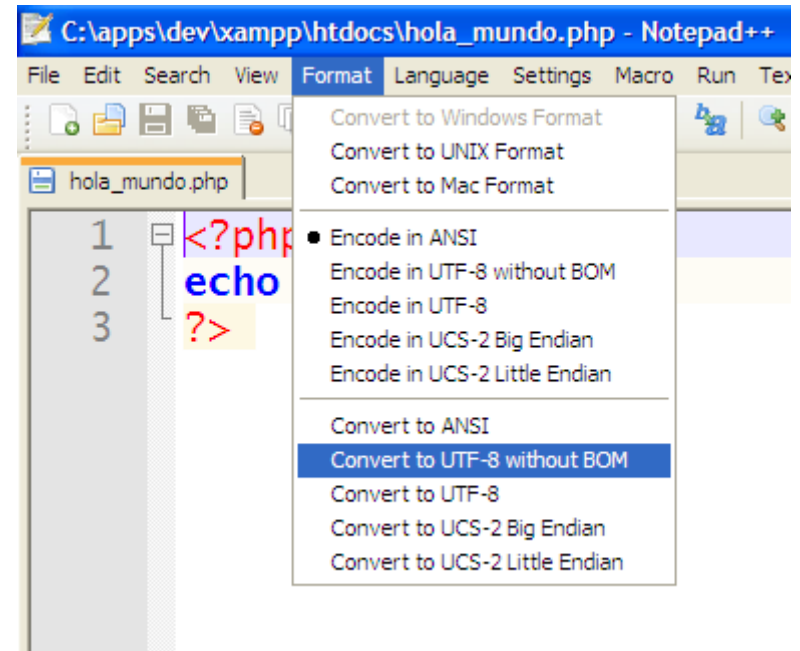
Hola Mundo!

unicode guardado como ASCII



¡Hola Mundo!

unicode guardado como UTF-8

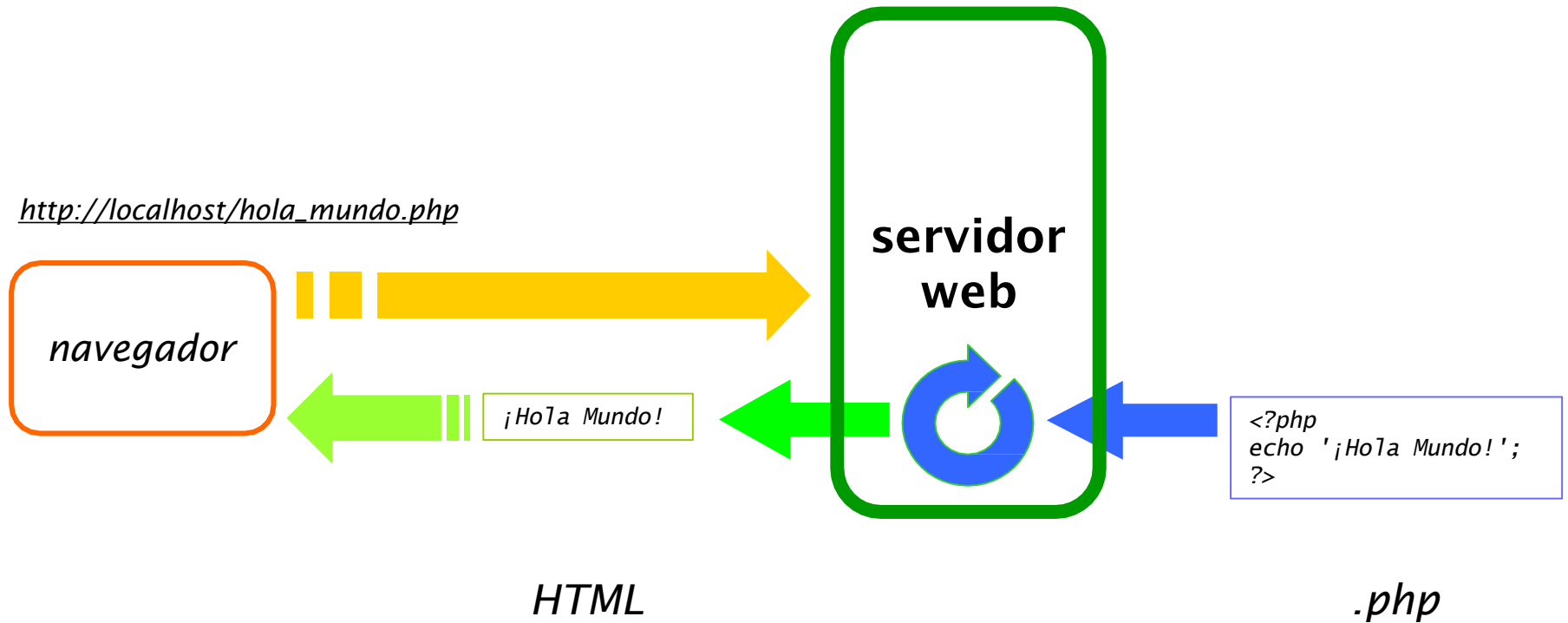


*En Notepad++ es fácil
comprobar y cambiar el encode*



*Si usamos unicode, tener cuidado de
guardar el archivo en UTF-8 (sin BOM)*

...¡Hola Mundo!



...¡Hola Mundo!

`hola_mundo.php`

```
<html>
<body>
  <h1>
    <?php echo '¡Hola Mundo!';?>
  </h1>
</body>
</html>
```

- *Un documento php es básicamente html con código inserto*
- *Los bloques php se reemplazan por su resultado*

...¡Hola Mundo!

```
<html>  
<body>  
  <h1>  
    ¡Hola Mundo!  
  </h1>  
</body>  
</html>
```


Cuadrados

cuadrados.php

```
<html>
<body>
  <h1>Cuadrados</h1>
  <table border="1">
    <?php for ($i=1; $i<=10; $i++) { ?>
      <tr>
        <td><?php echo $i;?></td>
        <td><?php echo $i*$i;?></td>
      </tr>
    <?php } ?>
  </table>
</body>
</html>
```

...Cuadrados

```
<html>
<body>
  <h1>Cuadrados</h1>
  <table border="1">
    <tr>
      <td>1</td>
      <td>1</td>
    </tr>
    <tr>
      <td>2</td>
      <td>4</td>
    </tr>
    ...
    <tr>
      <td>10</td>
      <td>100</td>
    </tr>
  </table>
</body>
</html>
```

 <http://localhost/cuadrados.php>

Cuadrados

1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

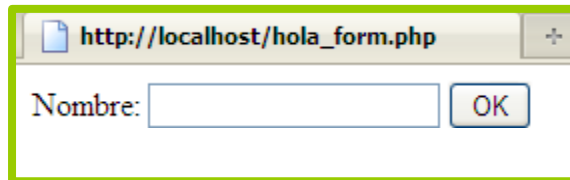
Conociendo PHP

formularios

Hola Tú

hola_form.php

```
<html>
<body>
  <form action="hola.php" method="post">
    Nombre:
    <input type="text" name="nombre"/>
    <input type="submit" value="OK"/>
  </form>
</body>
</html>
```



A screenshot of a web browser window. The address bar shows the URL `http://localhost/hola_form.php`. The page content displays the text "Nombre:" followed by a text input field and an "OK" button. The entire browser window is outlined with a green border.

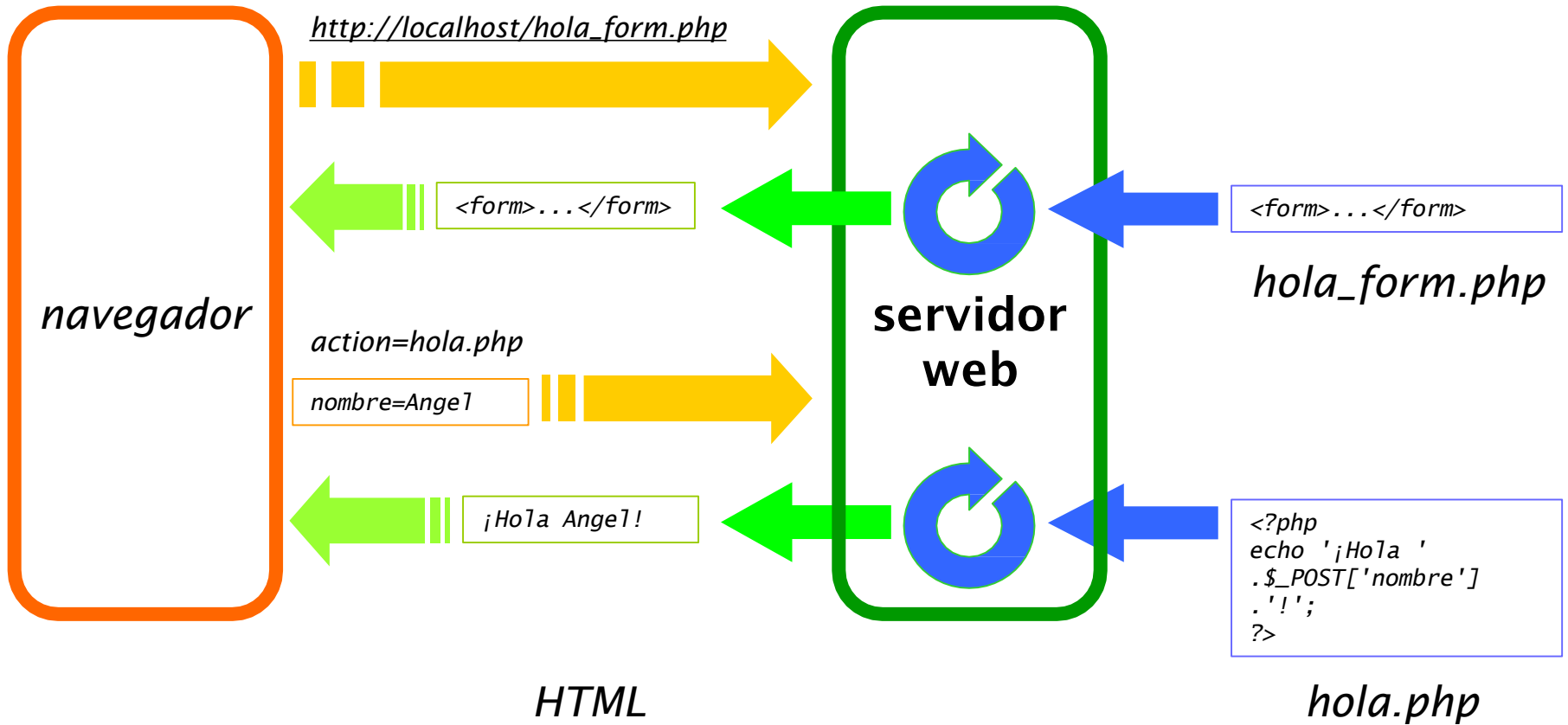
...Hola Tú

hola.php

```
<html>
<body>
<h1>
  <?php echo '¡Hola ' . $_POST['nombre'] . '!' ;?>
</h1>
</body>
</html>
```



...Hola Tú

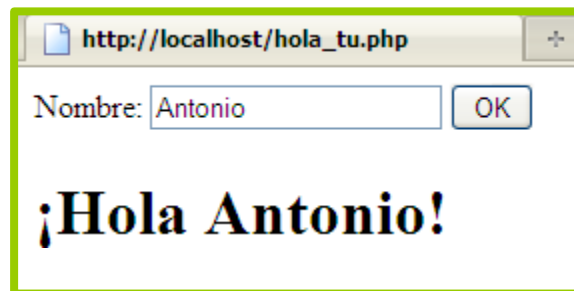


...Hola Tú

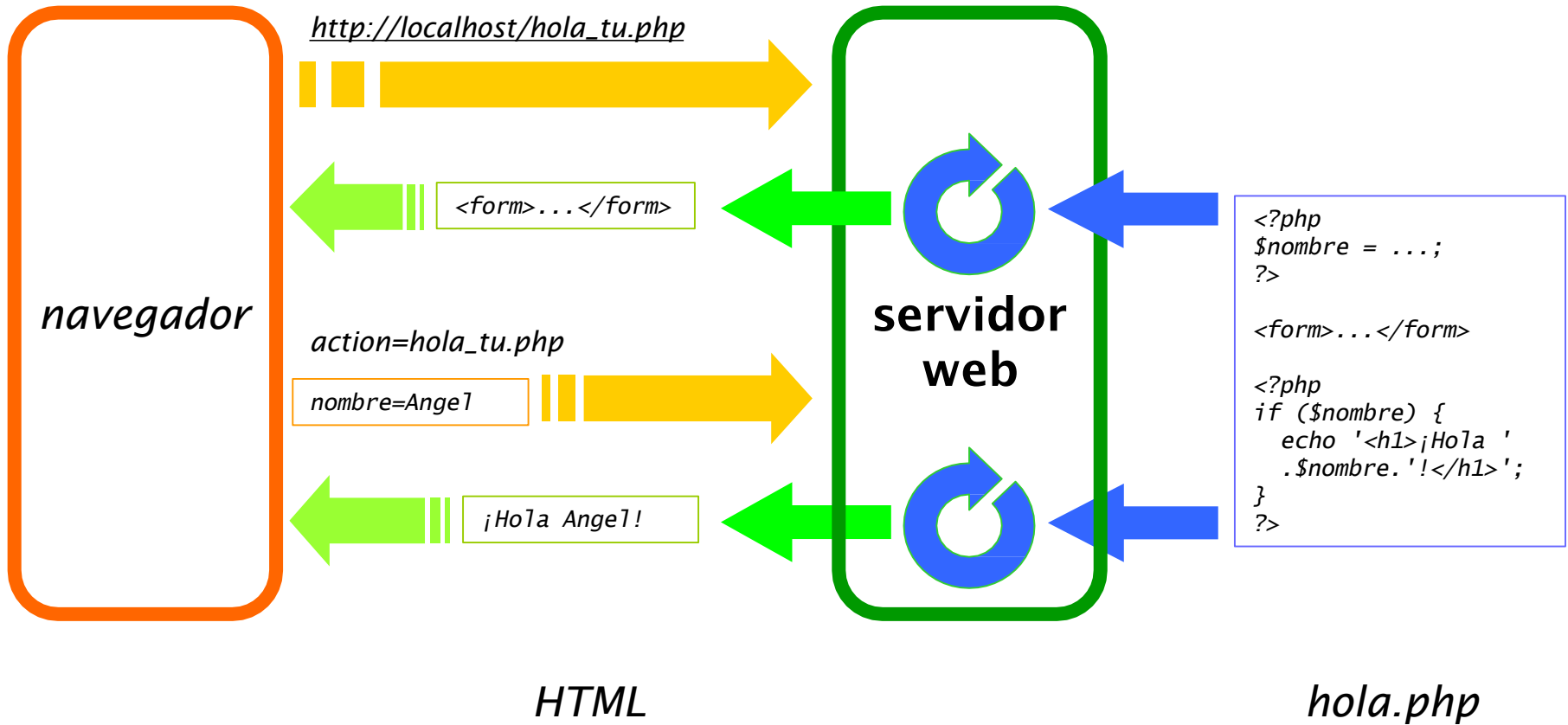
hola_tu.php

versión de una página

```
<html>
<body>
  <?php
    $nombre = isset($_POST['nombre'])?$_POST['nombre']:'';
  ?>
  <form action="hola_tu.php" method="post">
    Nombre:
    <input type="text" name="nombre"
      value="<?php echo $nombre;?>" />
    <input type="submit" value="OK">
  </form>
  <?php if ($nombre) {
    echo '<h1>¡Hola '.$nombre.'!</h1>';
  }?>
</body>
</html>
```



...Hola Tú



Conociendo PHP

un poco más

Bloques básicos

```
$x = 1+1;
```

```
if ($x>0) {  
  ...  
}
```

```
if ($x>0) {  
  ...  
} else {  
  ...  
}
```

```
switch ($x) {  
  case 1:  
    ...  
    break;  
  ...  
  default  
    ...  
    break;  
}
```

```
for ($i=0; $i<10; $i++) {  
  ...  
}
```

```
while ($i<10) {  
  ...  
}
```

```
do {  
  ...  
} while ($i<10);
```

asignación

decisión

- *if*
- *if/else*
- *switch*

repetición

- *for*
- *while*
- *do-while*

Arrays

```
$x = 1;  
$a = array(2, 3, $x, 'Hola');
```

```
(  
    [0]=>2  
    [1]=>3  
    [2]=>1  
    [3]=>'Hola'  
)
```

a[3]

'Hola'

```
foreach ($a as $item) {  
    echo $item.'  
}
```

```
2  
3  
1  
'Hola'
```

```
foreach ($a as $indice=>$item) {  
    echo $indice.': '.$item.'  
}
```

```
0: 2  
1: 3  
2: 1  
3: 'Hola'
```

Funciones

```
function hola_mundo() {  
    echo '¡Hola Mundo!';  
}
```

```
hola_mundo();
```

```
function hola($x) {  
    echo '¡Hola '.$x.'!';  
}
```

```
hola('Mundo');
```

```
function saludo($y) {  
    return '¡Hola '.$y.'!';  
}  
function hola($x) {  
    echo saludo($x);  
}
```

```
hola('Mundo');
```

```
¡Hola Mundo!
```

Clases

```
class Hola() {  
    function saludo($y) {  
        return '¡Hola '.$y.'!';  
    }  
    function hola($x) {  
        echo $this->saludo($x);  
    }  
}
```

```
$h = new Hola();  
$h->hola('Mundo');
```

¡Hola Mundo!

include

cabecera.php

```
echo '<h1>Conociendo PHP</h1>';
```



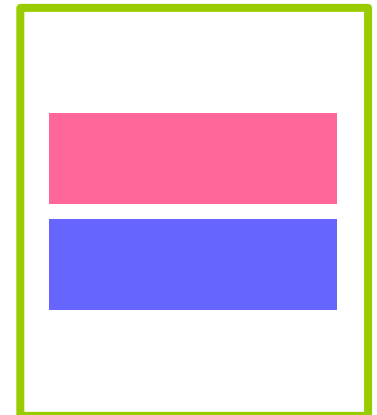
_hola_mundo.php

```
echo 'Hola Mundo';
```



hola_mundo.php

```
<html>
<body>
  <?php include 'cabecera.php';?>
  <?php include '_hola_mundo.php';?>
</body>
</html>
```



require

funciones.php

```
function hola() {  
    return '¡Hola Mundo!';  
}
```



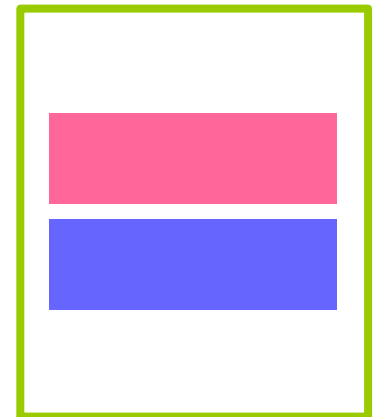
_hola_mundo.php

```
echo hola();
```



hola_mundo.php

```
<html>  
<body>  
    <?php require('funciones.php');?>  
    <?php include '_hola_mundo.php';?>  
</body>  
</html>
```

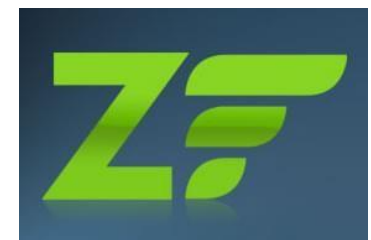


include & require

- *include() y require() son similares, la diferencia es que si no existe el archivo a incluir, include() emitirá una alarma, mientras que require() terminará el programa.*
- *Un mismo archivo se puede incluir más de una vez.*
- *Para prevenir duplicaciones en las inclusiones existen **include_once()** y **require_once()**.*

Frameworks

- *Un framework es una forma de organizar el trabajo.*
- *Luego de un tiempo desarrollando, uno empieza a notar ciertos patrones de organización. Aún sin proponérselo, va surgiendo un framework.*
- *Algunos frameworks muy difundidos en la comunidad PHP son CakePHP, CodeIgniter, Zend, Symfony...*
- *Un framework estandarizado obliga a seguir ciertas convenciones de organización, pero pone a nuestra disposición código ampliamente probado y optimizado.*



Referencias

- *PHP site*
 - <http://php.net>
- *W3Schools - PHP*
 - <http://www.w3schools.com/PHP/default.asp>
- *CakePHP*
 - <http://cakephp.org>
- *CodeIgniter*
 - <http://codeigniter.com>
- *Zend Framework*
 - <http://framework.zend.com/>
- *Symphony*
 - <http://www.symfony-project.org/>
- *Coquette Icons*
 - <http://dryicons.com/free-icons/preview/coquette-icons-set/>



Conociendo PHP



*Introducción a uno de
los lenguajes de
programación web
más usados en
Internet*

*LSI. Angel Veloz Rodríguez, Mgs.
angel.velozr@ug.edu.ec*



*preguntas,
comentarios*





INTERACCIÓN CON BASE DE DATOS

DESARROLLO DE
APLICACIONES WEB

CONCEPTOS BÁSICOS DE PHP Y MYSQL



PHP es un lenguaje de programación de código abierto que está especialmente indicado para la creación y desarrollo web y que se ha hecho muy popular, en gran medida, por su capacidad para poder ser incrustado en HTML. Su denominación es un acrónimo recursivo de “PHP: Hypertext Preprocessor”.



MySQL es una base de datos relacional. Está considerada como la mayor base de datos de código abierto del mundo para entornos de desarrollo web y cuenta con licencia doble GPL/Licencia comercial por parte de Oracle.



SQL hace referencia a StructuredQueryLanguage y es una base de datos relacional porque permite obtener datos de varias fuentes pero a partir de una consulta única.

CONCEPTOS BÁSICOS DE PHP Y MYSQL ...

- Como la gran mayoría de bases de datos, MySQL presenta seis tipos de objetos:
 - **Tablas:** estructura física donde se almacenan los datos.
 - **Vistas:** tablas formadas a partir de otras tablas como resultado de una consulta SQL.
 - **Funciones:** operaciones que el sistema realiza sobre los datos y que son imprescindibles para poder operar con la base de datos.
 - **Índices:** como en la vida real, constituyen un instrumento que facilita el acceso a los registros de una tabla en una base de datos, por lo que se suelen emplear en los campos que presentan mayores búsquedas para agilizar las mismas.
 - **Procesos almacenados:** programas que se almacenan en la base de datos para que sean ejecutados directamente. Por ejemplo, enviar una felicitación de cumpleaños a todos los usuarios cada vez que cumplen años.
 - **Triggers o disparadores:** permiten crear, editar o borrar tablas en una base de datos y se ejecutan sólo bajo ciertas condiciones.

CONCEPTOS BÁSICOS DE PHP Y MYSQL

...

- PHP está focalizado básicamente en la programación de scripts del lado del servidor, así que puede hacer todo lo que haga cualquier otro programa CGI como enviar y recibir cookies, generar páginas con contenidos dinámicos, recopilar formularios, etcétera.
- PHP admite la inmensa mayoría de servidores web, y puede utilizarse sobre todos los principales sistemas operativos del momento, incluyendo Windows, macOS, Linux, RISC OS, muchas variantes de Unix, etcétera.
- MySQL también es compatible con los principales sistemas operativos como Windows, Mac OS X y Linux; para su instalación será necesario descargar el paquete adecuado, WAMP, MAMP o LAMP respectivamente.

¿QUÉ ES PHPMYADMIN?



- **phpMyAdmin** es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando un navegador web.
- Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 72 idiomas.

¿QUÉ ES UN CRUD?

- **CRUD** es el acrónimo de "Crear, Leer, Actualizar y Borrar" (del original en inglés: Create, Read, Update and Delete), que se usa para referirse a las funciones básicas en bases de datos o la capa de persistencia en un software.
- En programación, crear, leer, actualizar y borrar (con el acrónimo CRUD) son las cuatro funciones básicas de la persistencia de Bases de Datos. Términos alternativos son usados a veces cuando se definen las cuatro funciones básicas de CRUD, como “recuperar” en vez de “leer”, “modificar” en vez de “actualizar” o “destruir” en vez de “borrar”.
- CRUD se usa también a veces para describir convenciones de interfaz de usuario que facilita la vista, búsqueda y modificación de la información; a menudo se usa en programación de formularios (forms) e informes (reports)..

PRÁCTICA

- Con base a los archivos proporcionados del diseño de una Agenda, desarrollemos un CRUD para realizar estas tareas.

CRUD : Agenda

Guardar Actividad

Actividad	Descripción	Creada	Acción
Levantarme	Salir de la cama	2020-09-01 15:46:33	 
Desayuno	Tomar Desayuno	2020-09-01 15:46:54	 
Trabajo	Ir al trabajo	2020-09-01 15:48:32	 
Hora de salida	Regresar a casa	2020-09-01 15:49:06	 
Cenar	Cenar con mi novia	2020-09-01 15:49:47	 
Acostarme	Dormir hasta mañana	2020-09-01 15:50:59	 

Copyright © 2020 CRUD : Agenda. Todos los derechos reservados

DESARROLLO
DE
APLICACIONES
WEB

MODELO-VISTA- CONTROLADOR (MVC)



El patrón de diseño (de software) MVC se encarga de separar la lógica de negocio de la interfaz de usuario y es el más utilizado en aplicaciones web, frameworks, etc.



Facilita la funcionalidad, mantenibilidad, y escalabilidad del sistema, de forma cómoda y sencilla, a la vez que ayuda a no mezclar lenguajes de programación en el mismo código, el conocido “código espagueti”.

¿QUÉ ES EL PATRÓN MVC?

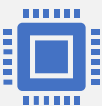
MVC DIVIDE LAS APLICACIONES EN TRES NIVELES DE ABSTRACCIÓN:



Modelo: es la lógica de negocios. Es decir las clases y métodos que se comunican directamente con la base de datos.



Vista: es la encargada de mostrar la información al usuario, de forma gráfica y legible.

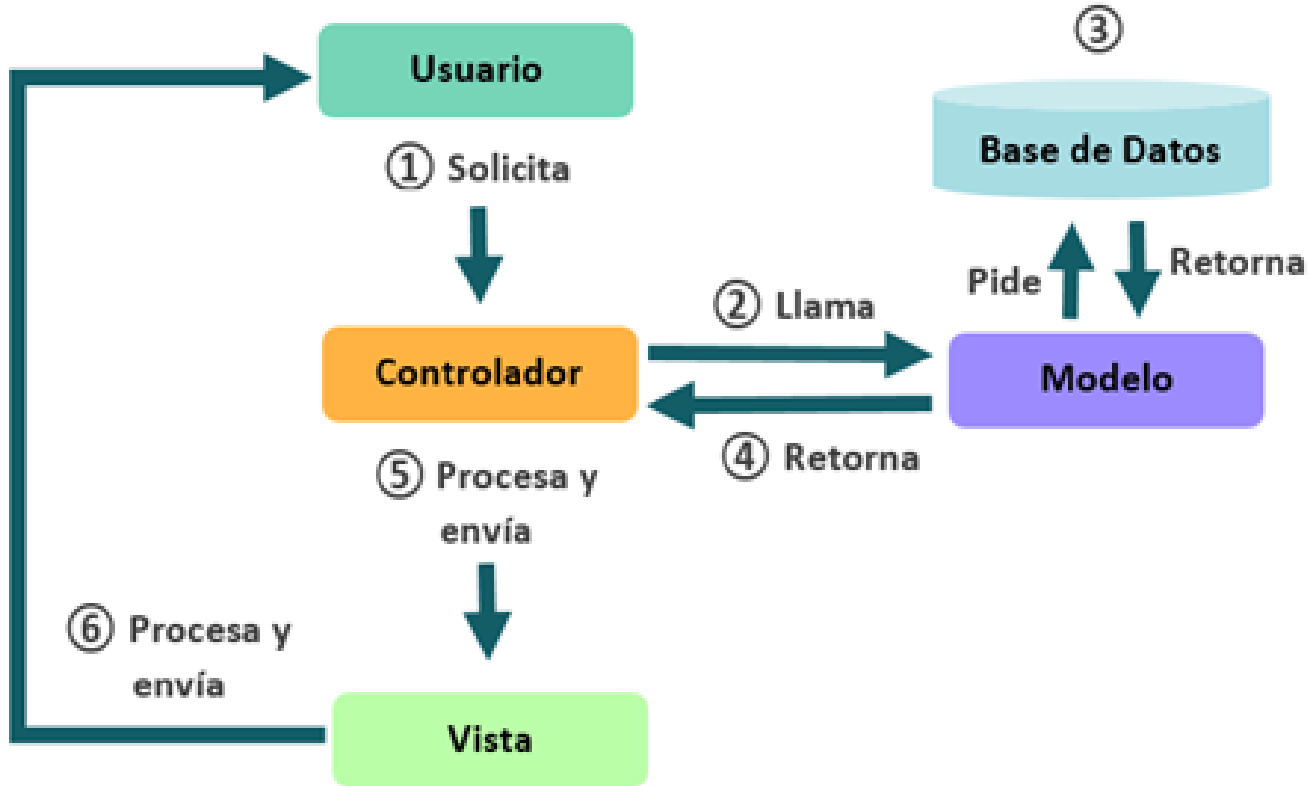


Controlador: es el intermediario entre la vista y el modelo, se encarga de controlar las interacciones del usuario en la vista, pide los datos al modelo y los devuelve de nuevo a la vista para que esta los muestre al usuario. Es decir las llamadas a clases y métodos, y los datos recibidos de formularios.

¿CÓMO FUNCIONA EL MVC?

El funcionamiento básico del patrón MVC, puede resumirse en:

1. El usuario realiza una petición.
2. El controlador captura la petición.
3. Hace la llamada al modelo correspondiente.
4. El modelo será el encargado de interactuar con la base de datos.
5. El controlador recibe la información y la envía a la vista.
6. La vista muestra la información.



ESQUEMA DE FUNCIONAMIENTO DEL MVC

DESARROLLO
DE
APLICACIONES
WEB

MVC (MODELO VISTA CONTROLADOR) EN PHP

¿CÓMO IMPLEMENTAR EL MVC EN PHP?

- Para implementar el MVC en PHP es necesario crear una estructura de ficheros parecida a esta:



EJEMPLO DEL USO DEL MVC CON PHP

■ # index.php

```
<?php
require_once("db/db.php");
require_once("controllers/personas_controller.php");
?>
```

■ # db.php

```
<?php
class Conectar{
    public static function conexion(){
        $conexion=new mysqli("localhost", "root", "", "mvc");
        $conexion->query("SET NAMES 'utf8'");
        return $conexion;
    }
}
?>
```

EJEMPLO DEL USO DEL MVC CON PHP

- # model/personas_model.php

```
<?php
class personas_model{
    private $db;
    private $personas;

    public function __construct(){
        $this->db=Conectar::conexion();
        $this->personas=array();
    }
    public function get_personas(){
        $consulta=$this->db->query("select * from personas;");
        while($filas=$consulta->fetch_assoc()){
            $this->personas[]=$filas;
        }
        return $this->personas;
    }
}
?>
```

EJEMPLO DEL USO DEL MVC CON PHP

- # controller/personas_controller.php

```
<?php
//Llamada al modelo
require_once("models/personas_model.php");
$per=new personas_model();
$datos=$per->get_personas();

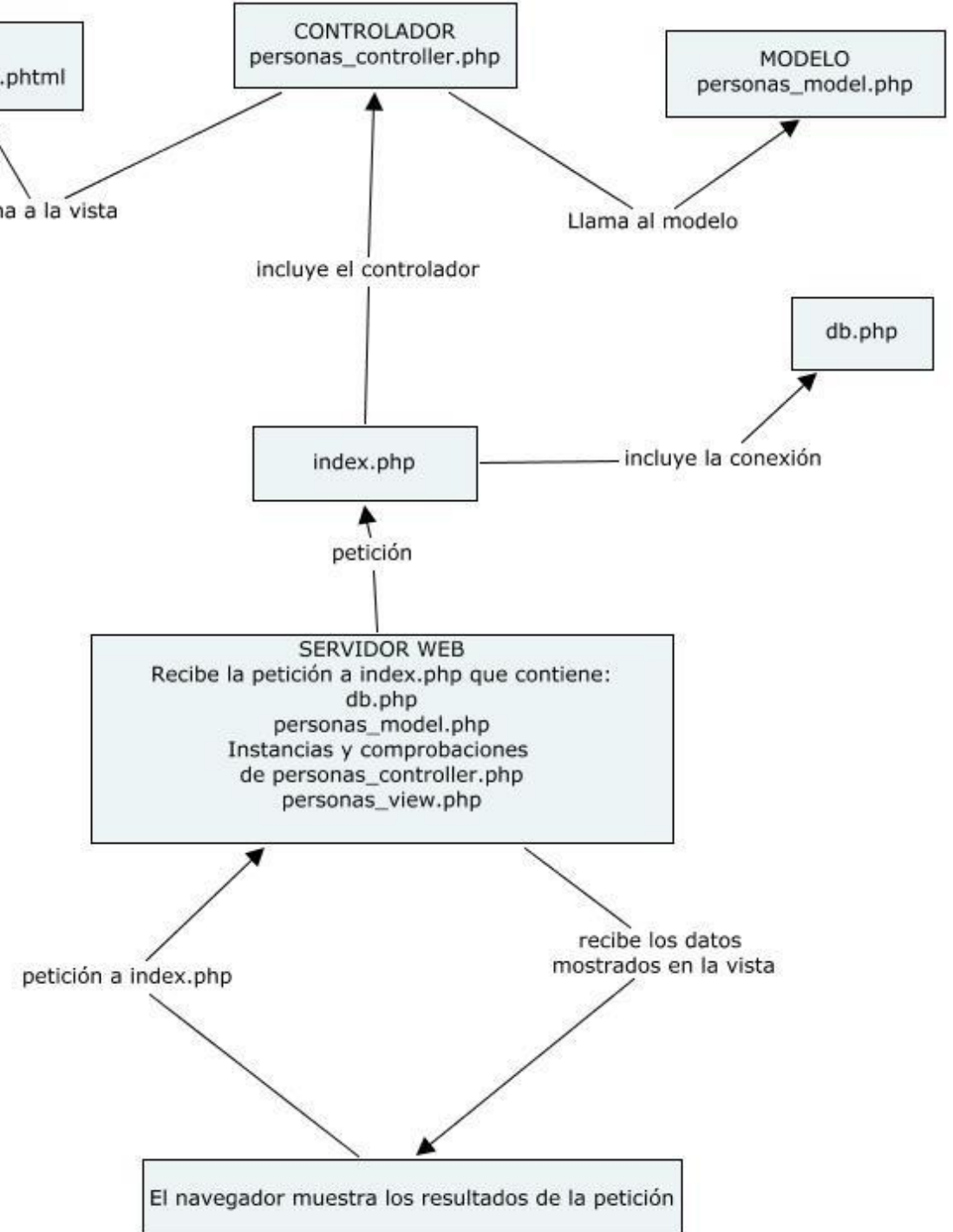
//Llamada a la vista
require_once("views/personas_view.phtml");
?>
```

EJEMPLO DEL USO DEL MVC CON PHP

- El controlador debe tener siempre esta estructura llamada al modelo y debajo a la vista, si hubiera mas modelos y vistas se sigue haciendo así con todos.
- Los ficheros de la vista según el estándar de Zend Framework debemos usar .phtml, pero se podría sin ningún problema usar la extensión .php

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <title>Personas</title>
  </head>
  <body>
    <?php
      foreach ($datos as $dato) {
        echo $dato["nombre"]."<br/>";
      }
    ?>
  </body>
</html>
```


ESQUEMA DEL EJEMPLO





TAREA: PRÁCTICA DE MVC CON PHP

- Realizar la práctica de la implementación del modelo básico de MVC con PHP propuesto en esta clase.
- Reportar las novedades y comentarios de ésta implementación.
- Tarea individual.



DESARROLLO
DE
APLICACIONES
WEB



APLICACIONES
WEB ÁGILES
UTILIZANDO
FRAMEWORKS

INTRODUCCIÓN

- Los proyectos se ven afectados por las limitaciones de tiempo, costo, alcance, calidad, recursos, capacidades organizativas y otras limitaciones que los hacen difíciles de planificar, ejecutar, administrar y finalmente tener éxito.

TIPOS DE PROYECTOS



- En el eje horizontal tenemos la experiencia y nuestro conocimiento sobre una herramienta; en el eje vertical se plasma la claridad de los requerimientos.

MANIFIESTO ÁGIL

- El manifiesto Ágil surge el 17 de febrero del 2001, cuando se reunieron diecisiete críticos del desarrollo de software, y acuñaron el término “metodología Ágil” para definir los métodos que estaban surgiendo como alternativa a las metodologías formales.
- El manifiesto Ágil está conformado por 12 principios asociados a 4 aspectos o pilares.

ASPECTOS O PILARES DEL MANIFIESTO

- A los individuos y su interacción, por encima de los procesos y las herramientas.
- El software que funciona, por encima de la documentación detallada.
- La colaboración con el cliente, por encima de la negociación contractual.
- La respuesta al cambio, por encima del seguimiento de un plan.



PRINCIPIOS DETRÁS DEL MANIFIESTO ÁGIL

1. La mayor prioridad es satisfacer al cliente a través de la entrega temprana y continua de software útil.
2. Bienvenidos los cambios a los requerimientos, incluso los tardíos.
3. Liberar frecuentemente software funcionando, desde un par de semanas a un par de meses, con preferencia por los periodos más cortos.
4. Los responsables del negocio y los desarrolladores deben trabajar juntos diariamente durante el proyecto.
5. Construir los proyectos alrededor de individuos motivados. Proporcionar el ambiente y el soporte que necesiten, y confiar en que conseguirán realizar el trabajo.
6. La conversación directa es el método más eficiente y efectivo de transmitir información, tanto al equipo como dentro de éste.



PRINCIPIOS DETRÁS DEL MANIFIESTO ÁGIL ...

7. El software funcionando es la medida de progreso.
8. Los procesos ágiles promueven el desarrollo sostenible.
9. La atención continua a la excelencia técnica y al buen diseño incrementan la agilidad.
10. La simplicidad - el arte de maximizar la cantidad de trabajo no hecho - es esencial.
11. Las mejores arquitecturas, requerimientos y diseños emergen de los equipos auto-organizados.
12. En intervalos regulares, el equipo reflexiona sobre cómo volverse más efectivo, entonces afina y ajusta su comportamiento como corresponde.



¿QUÉ ES AGILIDAD?

- “Agilidad es la capacidad de crear y responder al cambio con el fin de obtener ganancias en un entorno empresarial turbulento”
- “La agilidad es la capacidad de equilibrar la flexibilidad y estabilidad”



¿CÓMO DEBEMOS VER A LA AGILIDAD?

- En cualquier tipo de disciplina de gestión, ser ágil es una cualidad, por lo tanto esto debe ser una meta que se debe tratar de alcanzar.
- La gestión de proyectos Agile especialmente, implica la adaptabilidad durante la creación de un producto, servicio o cualquier otro resultado.

¿POR QUÉ METODOLOGÍAS ÁGILES?



- El 80% de todos los proyectos emplearán Métodos Ágiles en los próximos años (Gartner).
- Casi tres cuartas partes (71%) de las organizaciones informan que utilizan enfoques ágiles a veces, a menudo o siempre.

GESTIÓN DE PROYECTOS TRADICIONAL

- Ventajas: Orden lógico.
- Desventaja: Asume predictibilidad

Requirements

Analysis

Design

Coding

Testing

Operation

GESTIÓN DE PROYECTOS ÁGIL

- **Scrum** es un marco de trabajo de adaptación iterativa e incremental, rápido, flexible y eficaz diseñado para ofrecer un valor significativo de forma rápida en todo el proyecto.
- Scrum es:
 - Ligero.
 - Fácil de entender.
 - Extremadamente difícil de llegar a dominar.

■ Usos de Scrum


- Scrum fue desarrollado inicialmente para gestionar y desarrollar productos. Desde principios de los años 90.
- Scrum se ha usado para desarrollar software, hardware, software embebido, redes de funciones interactivas, vehículos autónomos, escuelas, gobiernos, mercadeo, también para gestionar la operación de organizaciones y casi todo lo que usamos en nuestra vida diaria, como individuo y como sociedad.
- Scrum demostró ser especialmente efectivo en la transferencia iterativa e incremental de conocimiento.
- Scrum se usa ahora ampliamente para productos, servicios y gestión de la organización matriz.

CONCLUSIONES


- **Mejora del tiempo de respuesta** para implementar los requerimientos más elementales de un sistema de software.
- **Rápido aprendizaje en el uso de la herramienta.** En general, los programadores con experiencia en lenguajes web tienen una curva de aprendizaje muy rápida para lograr utilizar los aspectos generales y aún particulares.
- **Alto nivel de confianza y reducción de los tiempos de prueba.** Al disponer de una herramienta estable, los desarrollos tradicionales obtenidos automáticamente no necesitan pruebas exhaustivas del software, lo que produce mejoras en términos de calidad de desarrollo de software.
- **Auditoría y control de cambios.** Es posible que el administrador del sistema defina de manera dinámica la información que será auditada por la aplicación web, como elemento adicional dentro del sistema desarrollado. Además, dentro de las atribuciones del administrador estará la definición de aquellos datos considerados indispensables para el sistema, sobre los cuales un usuario estándar con bajos permisos no podrá realizar determinadas operaciones.

CONCLUSIONES

- **Versatilidad para la incorporación de código particular por el programador.** De esta forma, es posible incorporar fácilmente la funcionalidad propia a cada sistema. Si bien las posibilidades de la herramienta está en continua evolución, siempre hay características puntuales que se necesita del software. Los Frameworks están pensados para incorporar de una manera sencilla código específico.
- **Los Frameworks permiten acortar plazos en el ciclo de vida de desarrollo de software,** no sólo en la fase de programación sino también en la de diseño y de prueba. Esto genera expectativas positivas para el desarrollo de aplicaciones que requieran ser orientadas a la web.



DESARROLLO
DE
APLICACIONES
WEB



FRAMEWORKS
PARA
DESARROLLO
WEB



¿QUÉ ES UN FRAMEWORK?

- Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.
- En el desarrollo de software, un entorno de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software.
- Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.
- Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

01

Acelerar el
proceso de
desarrollo,

02

reutilizar código
ya existente y

03

promover buenas
prácticas de
desarrollo como el
uso de patrones.

OBJETIVOS PRINCIPALES DE UN FRAMEWORK

CARACTERÍSTICAS DE LOS FRAMEWORKS

La **Autenticación** mediante login y password que permite restringir el acceso y el tipo de permiso.

El **Acceso a los datos** en archivos txt, xml por ejemplo mediante interfaces que integran la base de datos.

Abstracción de URLs y Sesiones ya que el framework se encarga de manejarlas.

Internacionalización que permite la inclusión de varios idiomas en el desarrollo.

Controladores fácilmente adaptables a las necesidades del proyecto que gestionan las peticiones y/o eventos.

- Compatibilidad de Lenguajes
- Transparencia de proyectos de plataforma a plataforma
- Portabilidad de Arquitectura
- Integración con múltiples dispositivos.
- Desarrollo de aplicaciones de manera más sencilla, ya que cuenta con los componentes necesarios incluidos.
- Reutilización de Código
- Maneja Política de diseño uniforme y organizado.

VENTAJAS MÁS COMUNES

TIPOS DE FRAMEWORKS

Para aplicaciones web

- Son aquellos Frameworks que se utilizan específicamente para la creación de proyectos online. Desde el diseño web de una página hasta los servicios web más específicos.
- Dentro de estos Frameworks existen otros tipos, dependiendo del lenguaje de programación utilizado. Sin embargo, nada impide que se pueda utilizar un Framework originalmente pensado en un lenguaje de programación, en otro diferente.
- Para un buen programador en muchos casos es más fácil adaptar un lenguaje a otro, que modificar un proyecto con diferentes objetivos.

TIPOS DE FRAMEWORKS

...

Para aplicaciones en general

- Permite complementar la estructura de una aplicación para un sistema operativo.
- Por ejemplo, Microsoft ha desarrollado el .NET Framework que ayuda a los programadores a reutilizar estructuras ya elaboradas.
- Este Framework ya viene instalado en su sistema operativo, el popular Windows.

TIPOS DE FRAMEWORKS

...

Para tecnología AJAX

- La tecnología AJAX permite que el usuario haga solicitudes al servidor sin que sea necesario recargar una página después de cada nueva solicitud.
- De esta forma, las informaciones van surgiendo a medida que son solicitadas sin la necesidad de que la página quede recargándose.
- Existen Frameworks específicos para esta tecnología, permitiendo la reutilización de un código ya elaborado.

TIPOS DE FRAMEWORKS

...

De gestión de contenidos

- A estos Frameworks también se les conoce como CMF, que significa, Content Manager Framework y facilita la programación de aplicaciones de un Sistema de Gestión de Contenidos, popularmente conocido como CMS, por ejemplo, WordPress.
- Existe una gran variedad de CMF de acuerdo con la plataforma para la que será creada la aplicación.

TIPOS DE FRAMEWORKS

...

De Multimedia

- Esta interfaz facilita el trabajo de los programadores que trabajan con video, audio e imagen y colabora con la creación de las aplicaciones multimedias en general, pudiendo servir para proyectos más complejos, como videoconferencias y conversores de medios.



Es un framework diseñado para apoyar el desarrollo de sitios web dinámicos, aplicaciones web y servicios web.



Este tipo de frameworks intenta aliviar el exceso de carga asociado con actividades comunes usadas en desarrollos web.



Por ejemplo, muchos framework proporcionan bibliotecas para acceder a bases de datos, estructuras para plantillas y gestión de sesiones, y con frecuencia facilitan la reutilización de código.

¿QUÉ ES UN FRAMEWORK WEB?

DEFINICIÓN DE FRAMEWORK WEB

- Un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web.

PATRÓN MVC



Para comprender como trabajan los frameworks Web existentes es imprescindible conocer el patrón MVC.



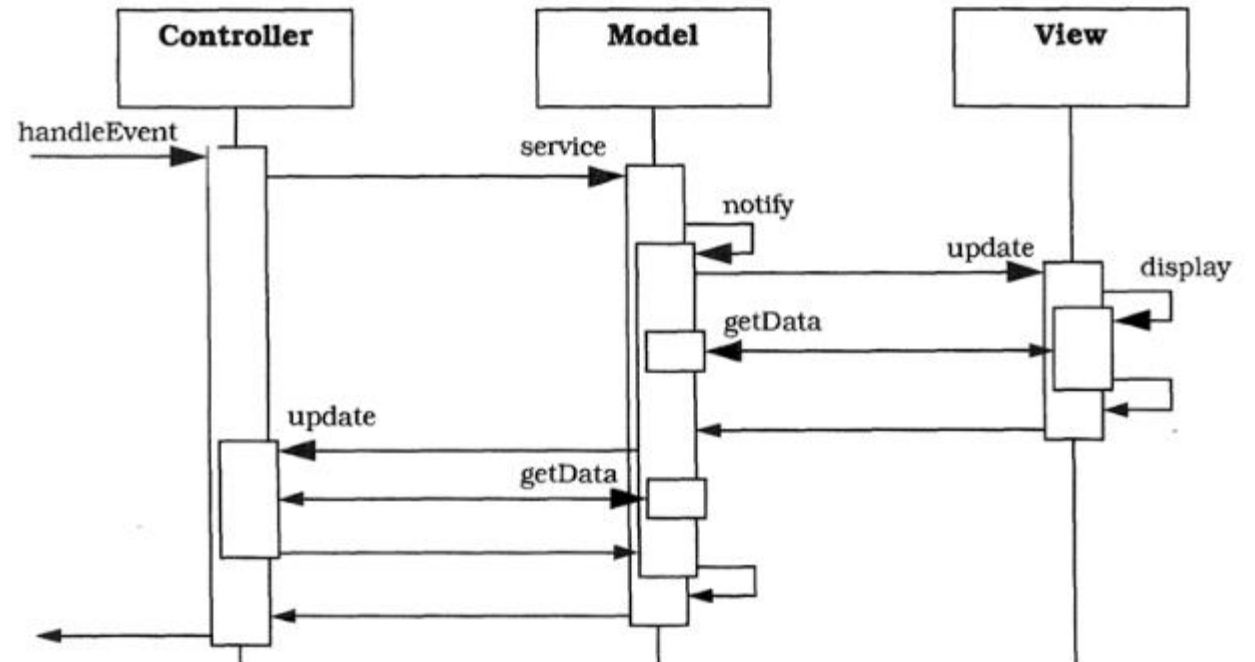
El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios.



La mayoría, por no decir todos, de los frameworks para Web implementan este patrón.

PATRÓN MODELO-VISTA-CONTROLADOR

- Este patrón organiza la aplicación en tres modelos separados,
- el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio,
- el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información,
- el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema.



- El tiempo que se gasta en crear una aplicación empezando de cero es totalmente innecesario y solo traería retraso en la ejecución de nuestra estrategia y una mayor inversión.
- El uso de un Framework optimiza la elaboración de un proyecto, finalizándolo más rápido, dando acceso a recursos que puede ser que no conozcas y utilizando un código más limpio.

¿POR QUÉ LOS FRAMEWORKS SON IMPORTANTES?

¿CUÁNTOS FRAMEWORKS EXISTEN?

Existen muchos Framework por eso más que difícil es casi imposible cuantificarlos y su uso va de acuerdo a las necesidades de cada proyecto y al gusto del desarrollador; algunos de ellos pueden ser:

- **.NET**: El Framework de Microsoft.
- **Ruby on Rails (RoR)**: es el Framework de aplicaciones web de código abierto de un lenguaje de programación llamado Ruby.
- **Genesis**: Framework para WordPress
- **Zengrid**: Framework para Joomla
- **PhoneGap**: Permite crear aplicaciones móviles multiplataforma a partir de tecnologías web como HTML5, CSS3 y Javascript.
- **JQuery Mobile**: basado en HTML5 optimizado para todas las plataformas de dispositivos móviles.
- **Titaniun**: Framework Open Source con licencia Apache, genera aplicaciones nativas para iPhone, Android y Blackberry.

¿CÓMO ELEGIR EL MEJOR FRAMEWORK PARA TUS OBJETIVOS?



La mejor forma para elegir el Framework indicado es conocer en detalles el proyecto, ya que el tipo de Framework utilizado dependerá de estas informaciones.



Los Frameworks son recursos extremadamente útiles tanto en la creación de una aplicación o software. Lo importante es tener bien claros los objetivos para poder elegir el Framework más indicado para el proyecto.



Este es un recurso cada vez más utilizado por los programadores que buscan optimizar su trabajo independientemente del lenguaje de programación con el que trabajen.



Cada día que pasa surgen nuevos lenguajes de programación, cada uno con diferentes características que se adaptan al tipo de aplicación que se quiera realizar.

REPORTE DE LABORATORIO DE FRAMEWORKS PARA PHP

Instrucciones


- Elaborar un reporte de investigación sobre Frameworks de PHP.
- Trabajo grupal
- Este reporte debe incluir la prueba y utilización del Framework asignado a su grupo.

El reporte de Laboratorio debe contener:

- Carátula con integrantes del grupo
- Antecedentes del Framework (empresas u organizaciones que las mantienen o les dan soporte, breve historia) .
- Instalación (con capturas de pantallas de la prueba y comentadas)
- Configuración (con capturas de pantallas de la prueba y comentadas)
- Estructura de directorios (con capturas de pantallas de la prueba y comentadas)
- Generadores de CRUD (Generar un CRUD, con capturas de pantallas de la prueba y comentadas)
- Incluir sus propias conclusiones y recomendaciones.

LISTADO DE FRAMEWORKS:

1. Laravel
2. Symfony
3. CodeIgniter
4. Yii 2
5. Phalcon
6. CakePHP
7. Zend Framework
8. Slim
9. FuelPHP
10. PHPixie
11. Zikula
12. Kohana




DESARROLLO
DE
APLICACIONES
WEB



AJAX /
ASYNCHRONOUS
JAVASCRIPT AND
XML

¿QUÉ ES AJAX? / JAVASCRIPT ASÍNCRONO Y XML

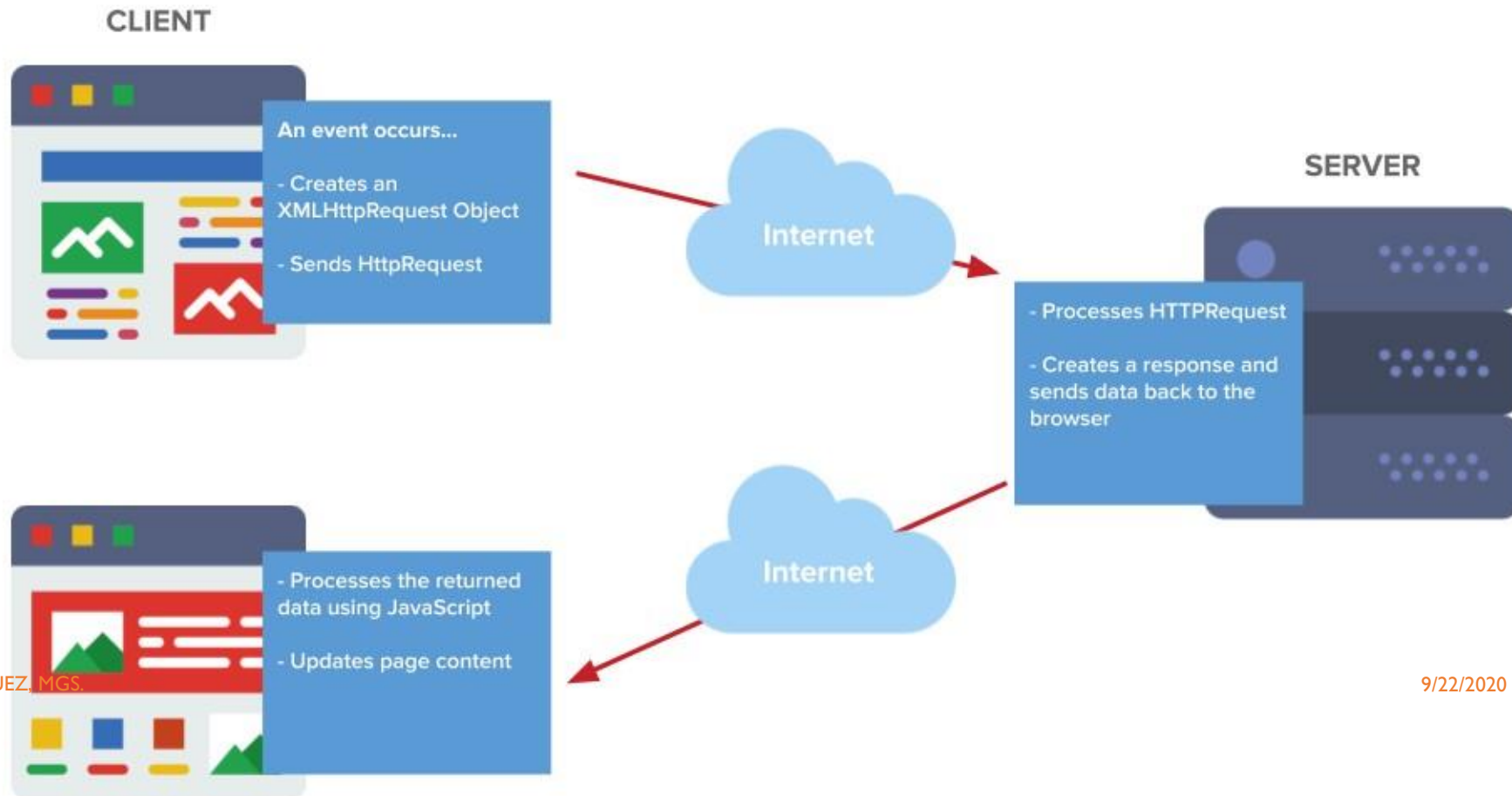
- AJAX es una técnica para la creación de páginas web rápidas y dinámicas.
- AJAX permite que las páginas web se actualicen de forma asincrónica mediante el intercambio de pequeñas cantidades de datos con el servidor en segundo plano.
- Esto significa que es posible actualizar partes de una página web, sin necesidad de recargar la página entera.
- Páginas web clásicas, (que no utilice AJAX) deben recargar la página entera si el contenido debe cambiar.
- Ejemplos de las aplicaciones que utilizan AJAX: Google Maps, Gmail, Youtube, Facebook.

- 
- Leer datos de un servidor web, después de que se haya cargado la página
 - Actualizar una página web sin recargar la página
 - Envíe datos a un servidor web, en segundo plano

AJAX ES EL SUEÑO DE UN DESARROLLADOR, PORQUE PUEDE:

¿CÓMO FUNCIONA AJAX?

Ajax



- AJAX se basa en estándares de Internet, y utiliza una combinación de:
 - **Objeto XMLHttpRequest** (para intercambiar datos de forma asíncrona con el servidor)
 - **JavaScript / DOM** (a visualización / interactúan con la información)
 - **CSS** (al estilo de los datos)
 - **XML** (a menudo utilizado como formato para la transferencia de datos)

AJAX SE BASA EN ESTÁNDARES DE INTERNET

EL OBJETO XMLHTTPREQUEST

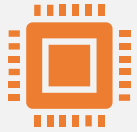
- Todos los navegadores modernos admiten el objeto XMLHttpRequest.
- El objeto XMLHttpRequest se puede utilizar para intercambiar datos con un servidor web detrás de escena.
- Esto significa que es posible actualizar partes de una página web, sin recargar toda la página.

CREAR UN OBJETO XMLHTTPREQUEST

- Todos los navegadores modernos (Chrome, Firefox, IE7 +, Edge, Safari, Opera) tienen un objeto XMLHttpRequest integrado .
- Sintaxis para crear un objeto XMLHttpRequest:

Ejemplo

```
var xhttp = new XMLHttpRequest();
```



Por razones de seguridad, los navegadores modernos no permiten el acceso entre dominios.



Esto significa que tanto la página web como el archivo XML que intenta cargar, deben estar ubicados en el mismo servidor.



Si desea utilizar el ejemplo anterior en una de sus propias páginas web, los archivos XML que cargue deben estar ubicados en su propio servidor.

ACCESO A TRAVÉS DE DOMINIOS

Method	Description
<code>new XMLHttpRequest()</code>	Creates a new XMLHttpRequest object
<code>abort()</code>	Cancels the current request
<code>getAllResponseHeaders()</code>	Returns header information
<code>getResponseHeader()</code>	Returns specific header information
<code>open(method, url, async, user, psw)</code>	Specifies the request <i>method</i> : the request type GET or POST <i>url</i> : the file location <i>async</i> : true (asynchronous) or false (synchronous) <i>user</i> : optional user name <i>psw</i> : optional password
<code>send()</code>	Sends the request to the server Used for GET requests
<code>send(string)</code>	Sends the request to the server. Used for POST requests
<code>setRequestHeader()</code>	Adds a label/value pair to the header to be sent

MÉTODOS DEL OBJETO XMLHTTPREQUEST

PROPIEDADES DEL OBJETO XMLHTTPREQUEST

Property	Description
onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
responseText	Returns the response data as a string
responseXML	Returns the response data as XML data
status	Returns the status-number of a request 200: "OK" 403: "Forbidden" 404: "Not Found" For a complete list go to the Http Messages Reference
statusText	Returns the status-text (e.g. "OK" or "Not Found")



AJAX CON PHP

AJAX SE UTILIZA PARA CREAR APLICACIONES MÁS INTERACTIVAS

EJEMPLO DE PHP AJAX

- En este ejemplo, cuando un usuario escribe un carácter en el campo de entrada, se ejecuta una función llamada **showHint()**.
- La función es activada por el evento **onkeyup**.

```
<html>
<head>
<script>
function showHint(str) {
    if (str.length == 0) {
        document.getElementById("txtHint").innerHTML = "";
        return;
    } else {
        var xmlhttp = new XMLHttpRequest();
        xmlhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
                document.getElementById("txtHint").innerHTML = this.responseText;
            }
        };
        xmlhttp.open("GET", "gethint.php?q=" + str, true);
        xmlhttp.send();
    }
}
</script>
</head>
<body>

<p><b>Start typing a name in the input field below:</b></p>
<form>
First name: <input type="text" onkeyup="showHint(this.value)">
</form>
<p>Suggestions: <span id="txtHint"></span></p>
</body>
</html>
```

- En primer lugar, comprueba si el campo de entrada está vacía (`str.length == 0`). Si es así, borra el contenido del marcador de posición `txtHint` y sale de la función.
- Sin embargo, si el campo de entrada no está vacía, hace lo siguiente:
 - Crea un objeto `XMLHttpRequest`
 - Crea la función a ejecutar cuando la respuesta del servidor está listo
 - Envía la solicitud fuera a un archivo PHP (`gethint.php`) en el servidor
 - Observa que el parámetro `q` se añade a la url (`gethint.php? Q = "+ str`)
 - Y la variable `str` mantiene el contenido del campo de entrada

EXPLICACIÓN DEL CÓDIGO

EL ARCHIVO PHP - "GETHINT.PHP"

- El archivo PHP verifica una matriz de nombres y devuelve los nombres correspondientes al navegador:

```
<?php
// Array with names
$a[] = "Anna";           // get the q parameter from URL
$a[] = "Brittany";      $q = $_REQUEST["q"];
$a[] = "Cinderella";
$a[] = "Diana";
$a[] = "Eva";           $hint = "";
$a[] = "Fiona";
$a[] = "Gunda";         // lookup all hints from array if $q is different from ""
$a[] = "Hege";          if ($q !== "") {
$a[] = "Inga";          $q = strtolower($q);
$a[] = "Johanna";       $len=strlen($q);
$a[] = "Kitty";         foreach($a as $name) {
$a[] = "Linda";         if (strpos($q, substr($name, 0, $len))) {
$a[] = "Nina";           if ($hint === "") {
$a[] = "Ophelia";       $hint = $name;
$a[] = "Petunia";       } else {
$a[] = "Amanda";        $hint .= ", $name";
$a[] = "Raquel";        }
$a[] = "Cindy";         }
$a[] = "Doris";         }
$a[] = "Eve";           }
$a[] = "Evita";         }
$a[] = "Sunniva";       }
$a[] = "Tove";          }
$a[] = "Unni";
$a[] = "Violet";
$a[] = "Liza";           // Output "no suggestion" if no hint was found or output correct value
$a[] = "Elizabeth";     echo $hint === "" ? "no suggestion" : $hint;
$a[] = "Ellen";
$a[] = "Wenche";
$a[] = "Vicky";
?>
```

AJAX Y BASE DE DATOS

AJAX SE PUEDE
UTILIZAR PARA LA
COMUNICACIÓN
INTERACTIVA CON
UNA BASE DE
DATOS

EJEMPLO DE BASE DE DATOS AJAX

- El siguiente ejemplo demostrará cómo una página web puede obtener información de una base de datos con AJAX:

id	FirstName	LastName	Age	Hometown	Job
1	Peter	Griffin	41	Quahog	Brewery
2	Lois	Griffin	40	Newport	Piano Teacher
3	Joseph	Swanson	39	Quahog	Police Officer
4	Glenn	Quagmire	41	Quahog	Pilot

Ejemplo

Peter Griffin ▼

Firstname	Lastname	Age	Hometown	Job
Peter	Griffin	41	Quahog	Brewery


```

<html>
<head>
<script>
function showUser(str) {
    if (str == "") {
        document.getElementById("txtHint").innerHTML = "";
        return;
    } else {
        if (window.XMLHttpRequest) {
            // code for IE7+, Firefox, Chrome, Opera, Safari
            xmlhttp = new XMLHttpRequest();
        } else {
            // code for IE6, IE5
            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
        xmlhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
                document.getElementById("txtHint").innerHTML = this.responseText;
            }
        };
        xmlhttp.open("GET","getuser.php?q="+str,true);
        xmlhttp.send();
    }
}
</script>
</head>
<body>

```

```

<form>
<select name="users" onchange="showUser(this.value)">
    <option value="">Select a person:</option>
    <option value="1">Peter Griffin</option>
    <option value="2">Lois Griffin</option>
    <option value="3">Joseph Swanson</option>
    <option value="4">Glenn Quagmire</option>
</select>
</form>
<br>
<div id="txtHint"><b>Person info will be listed here...</b></div>

</body>
</html>

```

EJEMPLO EXPLICADO

- En primer lugar, comprobar si se selecciona persona. Si no se selecciona ninguna persona (`str == ""`), borrar el contenido de `txtHint` y salir de la función. Si se selecciona una persona, haga lo siguiente:
- Crear un objeto `XMLHttpRequest`
- Cree la función a ejecutar cuando la respuesta del servidor está listo
- Enviar la solicitud fuera a un archivo en el servidor
- Tenga en cuenta que un parámetro (`q`) se añade a la URL (con el contenido de la lista desplegable)

```

<!DOCTYPE html>
<html>
<head>
<style>
table {
    width: 100%;
    border-collapse: collapse;
}

table, td, th {
    border: 1px solid black;
    padding: 5px;
}

th {text-align: left;}
</style>
</head>
<body>

<?php
$q = intval($_GET['q']);

$con = mysqli_connect('localhost','peter','abc123','my_db');
if (!$con) {
    die('Could not connect: ' . mysqli_error($con));
}

mysqli_select_db($con,"ajax_demo");
$sql="SELECT * FROM user WHERE id = '".$q."'";
$result = mysqli_query($con,$sql);

```

```

echo "<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Age</th>
<th>Hometown</th>
<th>Job</th>
</tr>";
while($row = mysqli_fetch_array($result)) {
    echo "<tr>";
    echo "<td>" . $row['FirstName'] . "</td>";
    echo "<td>" . $row['LastName'] . "</td>";
    echo "<td>" . $row['Age'] . "</td>";
    echo "<td>" . $row['Hometown'] . "</td>";
    echo "<td>" . $row['Job'] . "</td>";
    echo "</tr>";
}
echo "</table>";
mysqli_close($con);
?>
</body>
</html>

```



Cuando la consulta se envía desde el JavaScript para el archivo PHP, ocurre lo siguiente:



PHP abre una conexión a un servidor MySQL



La persona correcta se encuentra



Se crea una tabla HTML, lleno de datos, y enviado de vuelta al marcador de posición "txtHint"

EXPLICACIÓN DE CÓDIGO PHP



Realizar la práctica de los ejemplos de AJAX con PHP y MySQL propuesto en esta clase.



Reportar las novedades y comentarios de ésta implementación.



Tarea individual.

PRÁCTICA DE AJAX CON PHP Y MYSQL