INSTITUTO DEL SUR

CURSO: **LENGUAJE DE PROGRAMACIÓN WEB III**

TEMA: **Lenguajes de programación de lado del servidor - PHP**

INSTITUTO DEL SUR

Lenguajes de programación de lado del servidor - PHP

PHP

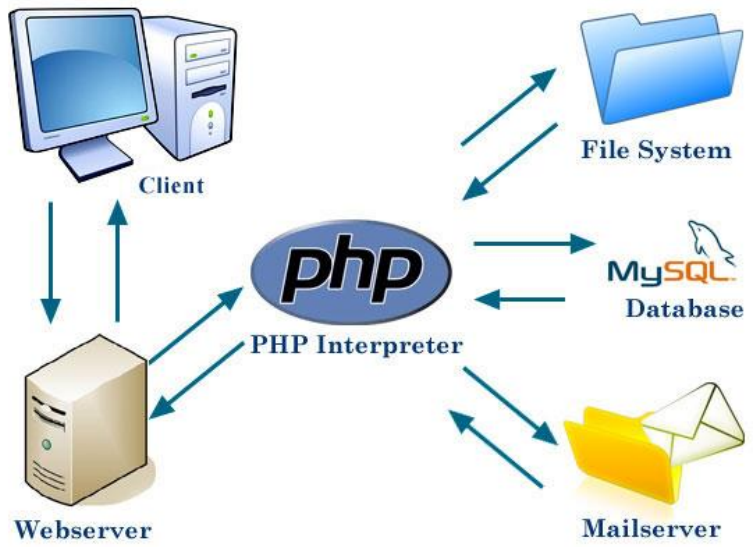
Para crear páginas web hasta ahora hemos visto (**HTML, CSS, JavaScript y Jquery**) en estos lenguajes el código fuente es interpretado por el navegador web para saber cómo debe renderizar (mostrar) la página web e interactuar con los elementos que la componen.

Por lo tanto, cuando nos conectamos a una página web a través de Internet el servidor web en el que se encuentra alojada enviará a nuestro navegador web el código fuente necesario para visualizarla.

PHP

Sin embargo, el código PHP será procesado o interpretado únicamente en el **servidor web** para realizar diversas operaciones (como conectar a **una base de datos**, entre otras muchas cosas) pudiendo también devolver código fuente que pueda ser interpretado por el navegador web.

El lenguaje PHP está registrado bajo **licencia Open Source**, con lo cual puedes descargarlo y usarlo de forma totalmente gratuita.



¿Qué necesito?

Para procesar una página en PHP se debe tener instalado en el ordenador la última versión de PHP y el servidor web **Apache**, debiendo realizar las configuraciones necesarias en este último para que interprete el código PHP.

Asimismo, para acceder a una base de datos (como por ejemplo **MySQL o PostgreSQL**), deberás instalarla también en tu ordenador y configurar PHP para que pueda acceder a ella.

Sin embargo, te recomendamos otra opción más rápida y sencilla: descargar los paquetes **LAMP**, **XAMPP, WAMP o MAMP** que contienen Apache, MySQL y PHP ya configurados y preparados para ser usados.

Incluir código PHP en una página web

Los archivos que contengan código fuente en lenguaje **PHP** deben tener extensión **.php**, y el código PHP se incluirá entre las etiquetas **<?php** y **?>**, y cada instrucción en código PHP debe finalizar con punto y coma (al igual que en JavaScript y otros lenguajes de programación).

SINTAXIS BÁSICA DE PHP

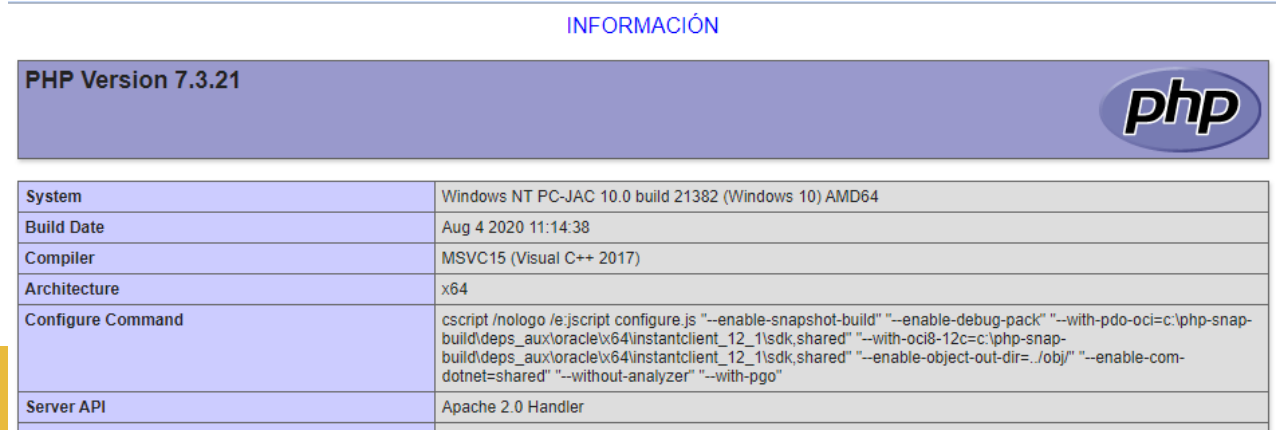
Ejemplo:

```
<HTML>
  <HEAD>
    <TITLE>Mi primer programa en PHP</TITLE>
  </HEAD>
  <BODY>
    <?PHP print("<P>Hola mundo</P>");?>
  </BODY>
</HTML>
```

Incluir código PHP en una página web

Los archivos **.php** pueden contener código fuente en otros lenguajes de programación web (HTML, CSS y JavaScript):

```
<html>
   <head>
      <title>Ejemplo usando PHP</title>
      <style type="text/css">
         p { color:blue; }
      </style>
   </head>
   <body>
      <p style="text-align:center">INFORMACIÓN</p>
      <?php phpinfo(); ?>
   </body>
</html>
```



INFORMACIÓN

PHP Version 7.3.21

System	Windows NT PC-JAC 10.0 build 21382 (Windows 10) AMD64
Build Date	Aug 4 2020 11:14:38
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x64
Configure Command	cscript /nologo /e:jscript configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler

Mostrar texto en el navegador web

Si deseamos que desde el código PHP se devuelva texto o código fuente en otros lenguajes de programación web usaremos **echo** o **print()** , como puedes ver en el siguiente ejemplo:

```
<html>
  <head>
    <title>Ejemplo usando PHP</title>
    <style type="text/css">
        p { color:blue; }
    </style>
  </head>
  <body>

    <?php
      echo "<p style='text-align:center;text-decoration:underline;'>INFORMACIÓN</p>";
      phpinfo();
    ?>

  </body>
</html>
```

Operadores en PHP

Aritméticos:

- Incremento: ++
- Decremento: --
- Multiplicación: *
- División: /
- Resto: %
- Suma: +
- Resta: -

Asignación:

- Asignación: =
- Suma y asignación: +=
- Resta y asignación: -=
- Multiplicación y asignación: *=
- División y asignación: /=
- Módulo (resto) y asignación: %=

Operadores en PHP

Comparación:

- Menor que: <
- Mayor que: >
- Menor o igual a: <=
- Mayor o igual a: >=
- Igual a: ==
- Exactamente igual (mismo tipo de dato y valor): ===
- Diferente de: !=
- Estrictamente diferente de: !==

Condicional:

- Ternario: ?: . Ej.: ((50 == 100) ? echo "Es 50" : echo "No es 50");

Lógicos:

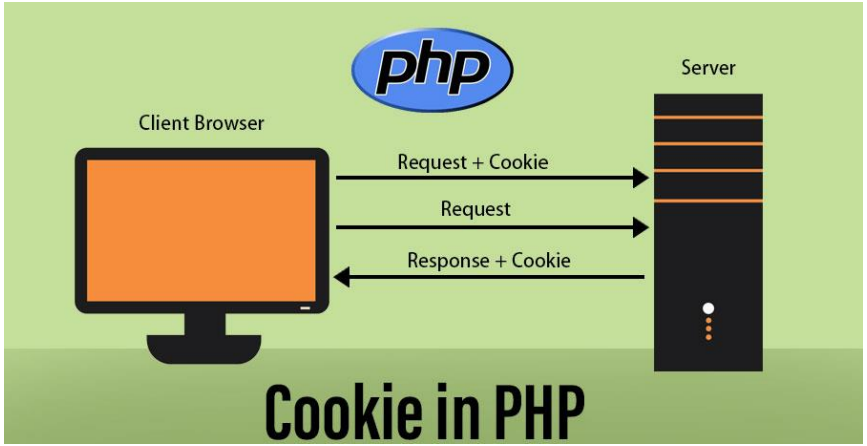
- Negación (NOT): !
- Conjunción (AND): &&
- Disyunción (OR): ||

PHP

Cookies

Cookies en PHP

Una Cookie es un pequeño archivo de texto que podemos guardar en el navegador web del usuario con diversos fines, como el registrar ciertas preferencias de mismo respecto a nuestra página web para cargarlas cada vez que acceda a ella.



Cookies en PHP

Para crear una Cookie en PHP, antes de enviar ningún otro dato al navegador web usaremos la función de PHP **setcookie()** indicando el **nombre** de la misma, su **valor** y el **tiempo** de vida en formato **UNIX Timestamp** (una vez transcurrido dicho periodo será eliminada por el navegador web):

```
<?php
    setcookie("nombre", 1, time() + (60*2) );
    // Crea una Cookie con un tiempo de vida de 2 minutos
?>
```

Cookies en PHP

Observa que para indicar el tiempo de vida como usamos la función **time()**, que devuelve el **número de segundos desde el 1 de Enero de 1970**.

Si usas el navegador web **Firefox** podrás **ver información sobre las cookies** utilizando la barra de herramientas de la extensión **Web Developer**, complemento indispensable para todo aquel que le guste o se dedique profesionalmente a crear páginas web.

Comprobar si existe una Cookie

Para comprobar si existe una Cookie usaremos el array asociativo **$_COOKIE** del siguiente modo:

```
<?php
    if( isset( $_COOKIE['nombre']) )
    {
        echo "<p>La cookie ha sido creada</p>";
    }
    else
    {
        echo "<p>La cookie no existe, la creamos</p>";

        // Crea una Cookie con un tiempo de vida de 2 minutos
        setcookie("nombre", 1, time() + (60*2) );
    }

    echo "<a href='02_cookies_comprobar.php'>Haz clic aquí para recargar la página</a>";
?>
```

Comprobar si están activadas las Cookies

En el siguiente ejemplo hemos creado la función **comprobarCookies()** con la que podremos comprobar si las Cookies están activas en el navegador web del usuario.

Tal y como se ha comentado, será necesario recargar la página web o cargar otra para que se actualice **$_COOKIE** (y poder entonces usar nuestra función), por lo tanto en el ejemplo hemos usado la función de PHP **header()** para cargar otra página.

```
<?php
    setcookie("nombre", 1, time() + (60*2) );
    header("Location: 03_cookies_activas_2.php");
?>
```

```
<?php
    function comprobarCookies()
    {
        $activas = false;

        if( isset($_COOKIE['nombre']) )
            $activas = true;

        return $activas;
    }

    // ---------------

    if( comprobarCookies() == true )
        echo("Las Cookies están activas");
    else
        echo "Las Cookies están desactivadas";
?>
```

Obtener el valor de una Cookie

Para **leer** el valor de una Cookie tan sólo hemos de acceder al valor contenido en el array asociativo **$_COOKIES**, como mostramos en el siguiente ejemplo:

```
<?php
    setcookie("nombre", "informaticapc.com", time() + (60*2) );
    header("Location: 04_cookies_leer_2.php");
?>
```

```
<?php
    if( isset($_COOKIE['nombre']) )
        echo "El valor de la Cookie 'nombre' es [".$_COOKIE['nombre']."]";
    else
        echo "No existe la Cookie";
?>
```

Eliminar Cookies en PHP

Para **eliminar** una Cookie en PHP tan sólo debemos volver a crearla indicando una fecha anterior a la actual.

```
<?php
    setcookie("nombre", "informaticapc.com", time() + (60*2) );
    header("Location: 05_cookies_eliminar_2.php");
?>

<?php
    if( isset($_COOKIE['nombre']) )
    {
        echo "Eliminamos la Cookie";
        setcookie("nombre", "", time() - 1 );
    }
    else
    {
        echo "No existe la Cookie";
    }

    echo "<p><a href='05_cookies_eliminar_2.php'>Haz clic para recargar la página y comprobar si existe la Cookie</a></p>";
?>
```

INSTITUTO DEL SUR

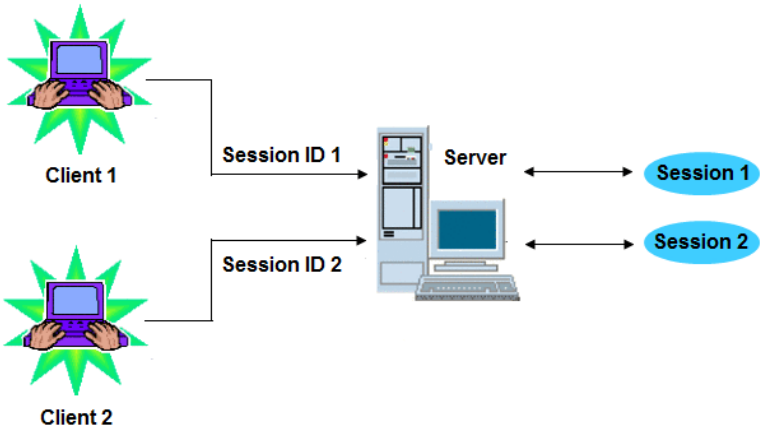
PHP

Sesiones y variables de sesión

Sesiones y variables de sesión

Al crear una variable en PHP ésta se encontrará disponible durante la ejecución de la página contenida en un archivo .php, eliminándose automáticamente una vez finalizado el **script**.

Sin embargo, en ocasiones necesitaremos que determinada información esté disponible en diferentes páginas en PHP y en posteriores accesos a las mismas: para ello podemos usar **variables de sesión**.



Iniciar y eliminar una sesión en PHP

Para **crear una variable de sesión** en primer lugar deberemos iniciar una sesión en PHP.

Para **Iniciar, identificar y eliminar sesiones** usaremos las siguientes funciones de PHP:

- **session_start()**: se coloca al principio de cada archivo .php (incluso antes de las funciones **include()** o **require()**) para **iniciar o reanudar la sesión**. Devuelve **true** si la sesión se ha iniciado correctamente o **false** en caso contrario.

- **session_id()**: asigna un identificador a una sesión (se recomienda que sea corto y que contenga **sólo caracteres alfanuméricos**). Si no asignamos un identificador, se asignará uno automáticamente. En caso de ser usada, debe estar antes de **session_start()**.

Iniciar y eliminar una sesión en PHP

- **session_name()**: asigna un nombre a una sesión (se recomienda que sea corto y que contenga **sólo caracteres alfanuméricos**). Si no indicamos un nombre, se usará por defecto el definido en el archivo de configuración de PHP (por defecto es **PHPSESSID**). **IMPORTANTE**: en caso de usarse deberá estar definida en cada archivo .php, y situada antes de **session_start()**.

- **session_destroy()**: elimina la información de la sesión actual, pero sin eliminar las variables ni las Cookies de sesión (el código para borrar dicha información deberá ir justo antes de llamar a esta función). Devuelve **true** si se ha eliminado la sesión o **false** en caso de error.

Iniciar y eliminar una sesión en PHP

Los pasos a seguir para crear y finalizar una sesión en PHP son:

```
<?php
   session_id();       // Uso no imprescindible. Si se usa debe estar antes de 'session_start()'
   session_start();    // Iniciar la sesión
   ...
   session_unset();    // Borrar las variables de sesión
   setcookie(session_name(), 0, 1 , ini_get("session.cookie_path"));    // Eliminar la cookie
   session_destroy();  // Destruir la sesión
?>
```

Crear y eliminar variables y arrays de sesión

Para crear una variable o array de sesión en PHP usaremos el array asociativo **$_SESSION** del siguiente modo:

Si deseamos eliminar una variable de sesión disponemos de la función de PHP **unset()**. Si lo que deseamos es **eliminar todas las variables de sesión** usaremos **session_unset()**.

```php
<?php
    // Iniciar la sesión
    session_start();

    // Variables de sesión:
    $_SESSION['sesion_iniciada'] = true;
    $_SESSION['nombre'] = "PEDRO";

    // Variable de sesión en array:
    $_SESSION['aDatos'] = array();
    $_SESSION['aDatos']['nombre'] = "MARTA";

    echo "Nombre: ".$_SESSION['nombre']."<br />";
    echo "Nombre (en array): ".$_SESSION['aDatos']['nombre']."<p />";

    // Borrar el contenido de las variables y arrays de sesión:
    session_unset();
    //$_SESSION = array();      // También podemos hacerlo así:

    echo "<p>Variables y arrays de la sesión borrados</p>";

    if( isset($_SESSION['nombre']) == false )
    echo "Nombre no definido.<br />";

    if( isset($_SESSION['aDatos']['nombre']) == false )
        echo "Nombre (en array) no definido";
?>
```

Propagar objetos en una sesión de PHP

También es posible propagar Objetos en una sesión teniendo en cuenta que deberemos incluir la definición del mismo en cada una de las páginas (o bien en un archivo accesible desde todas ellas).

```php
<?php
   class Persona
   {
      public $nombre    = null;
      public $apellidos = null;

      function Persona()
      {
      }

      function getNombre() {
         return $this->nombre;
      }

      function setNombre( $nombre ) {
         $this->nombre = $nombre;
      }

      function getApellidos() {
         return $this->apellidos;
      }

      function setApellidos( $apellidos ) {
         $this->apellidos = $apellidos;
      }
   }
?>
```

Propagar objetos en una sesión de PHP

```php
<?php
    require_once("Persona.php");

    // Iniciar la sesión
    session_start();

    // Crear una instancia del objeto:
    $objPersona = new Persona();
    $objPersona->setNombre("MARTINA");
    $objPersona->setApellidos("MARRERO MEDINA");

    // Variables de sesión:
    $_SESSION['usuario'] = $objPersona;

    echo "PÁGINA PRINCIPAL<br />";
    echo "================<p />";

    // Mostrar información del objeto en la sesión:
    echo "Nombre: [".$_SESSION['usuario']->getNombre()."]<br/>";
    echo "Apellidos: [".$_SESSION['usuario']->getApellidos()."]<p/>";

    echo "<a href='03_sesion2.php'>Ir a la otra página</a><br/>";
?>
```

```php
<?php
    require_once("Persona.php");

    session_start();  // Continuar la sesión

    if( isset($_SESSION['usuario']) == true )
    {
        echo "COMPROBAR LOS VALORES<br />";
        echo "======================<p />";

        print_r( $_SESSION );

        echo "<p />";

        // Mostrar información del objeto en la sesión:
        echo "Nombre: [".$_SESSION['usuario']->getNombre()."]<br/>";
        echo "Apellidos: [".$_SESSION['usuario']->getApellidos()."]<p/>";
    }
    else
    {
        echo "<p>No has pasado por la página principal</p>";
    }

    echo "<a href='03_sesion1.php'>Volver a la página principal</a>";
?>
```

INSTITUTO DEL SUR

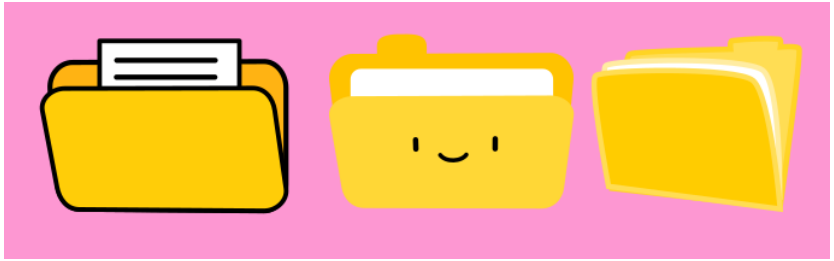
PHP

Archivos y carpetas en PHP

Manejo de archivos y carpetas en PHP

El lenguaje PHP dispone de una gran cantidad de funciones para realizar todo tipo de operaciones con archivos y carpetas, tanto básicas (como crear archivos y carpetas, modificar, eliminar...) como otras más avanzadas (para obtener y asignar permisos, crear enlaces simbólicos, etc.).

En esta sesión si bien no explicaremos todas ellas, sí veremos cómo realizar algunas de las tareas que más usualmente se realizan sobre archivos y carpetas en PHP.



Comprobar si existe un archivo o directorio

Si necesitamos averiguar si existe un archivo o carpeta utilizaremos la función de PHP **file_exists()**, la cual nos devolverá **true** si existe **false** en caso contrario:

```
<?php
    if( file_exists("datos.txt") == true )
        echo "<p>El archivo existe</p>";
    else
        echo "<p>El archivo no se ha encontrado</p>";


    if( file_exists("miCarpeta") == true )
        echo "<p>El directorio existe</p>";
    else
        echo "<p>El directorio no existe</p>";


    if( file_exists("./miCarpeta/") == true )
        echo "<p>El directorio existe</p>";
    else
        echo "<p>El directorio no existe</p>";
?>
```

Obtener información de un archivo

Algunas funciones que nos devolverán información sobre un archivo son:

- **filesize()**: devuelve el tamaño de un archivo en bytes.
- **filetype()**: devuelve el tipo de fichero de que se trata: **file, dir, block, char, fifo, link o unknown**.
- **filemtime()**: devuelve la fecha y hora de la última modificación del archivo, en formato **UNIX Timestamp**.
- **fileperms()**: devuelve la mascara de permisos (en sistemas UNIX).
- **stat()** y **fstat()**: devuelven información sobre un archivo abierto con **fopen()**.
- La diferencia entre ellas es que a **stat()** se le pasa como parámetro el nombre del archivo, y a **fstat()** el recurso obtenido con **fopen()**.

Obtener información de un archivo

Otras funciones con las que podemos obtener información sobre los archivos son:

- **is_executable()**: devuelve **true** si el archivo es ejecutable.
- **is_readable()**: comprueba si existe un archivo se puede leer.
- **is_writable()**: comprueba si existe un archivo y se puede escribir en él.
- **is_file()**: indica si el nombre pasado como parámetro es un archivo.
- **is_link()**: indica si el nombre pasado como parámetro es un enlace simbólico.
- **is_dir()**: indica si el nombre pasado como parámetro es un directorio.

Crear, abrir y cerrar un archivo en PHP

Para **crear un archivo en PHP** usaremos la función **fopen()** indicando el nombre del mismo como primer parámetro (pudiendo indicar la ruta) y el modo de creación en el segundo, pudiendo ser:

- **r**: abre el archivo en modo sólo lectura, colocando el puntero del archivo al principio del mismo.
- **r+**: apertura en modo lectura y escritura, colocando el puntero del archivo al principio del mismo.
- **w**: apertura del archivo en modo sólo escritura, colocando el puntero del archivo al principio del mismo: si el archivo no existe se creará, y si existe su contenido será borrado.
- **w+**: Lo mismo que el anterior pero abre el archivo en modo lectura y escritura.
- **a**: abre el archivo en modo sólo escritura, colocando el puntero del archivo al final del mismo: si el archivo no existe será creado.
- **a+**: Lo mismo que el anterior pero abre el archivo en modo lectura y escritura.
- **x**: abre el archivo en modo sólo escritura, quedando el puntero del archivo al principio del mismo: si el archivo no existía se crea, y si existía se devolverá **false** generándose también un error de tipo **E_WARNING**.
- **x+**: igual que la opción anterior pero abre el archivo en modo lectura y escritura.
- **c**: abre el archivo en modo sólo escritura, quedando el puntero del archivo al principio del mismo: si el archivo no existía se creará. Si existe no se eliminará su contenido (como sucede con **w**) ni se devolverá un error (como con **x**).
- **c+**: lo mismo que la opción anterior pero crea y abre el archivo en modo lectura y escritura.

Crear, abrir y cerrar un archivo en PHP

Como puedes ver en el siguiente ejemplo, **fopen()** nos devuelve un puntero al archivo:

```
<?php
   $archivo = fopen("datos.txt", "w+b");    // Abrir el archivo, creándolo si no existe

   if( $archivo == false )
     echo "Error al crear el archivo";
   else
     echo "El archivo ha sido creado";

   fclose($archivo);   // Cerrar el archivo
?>
```

Desplazamiento dentro de un archivo

Algunas funciones de PHP que debes conocer puesto que te resultarán útiles para averiguar la posición del **puntero a un archivo** así como para desplazarlo dentro del mismo, son:

- **rewind()**: situa el puntero del archivo al principio del mismo
- **ftell()**: nos devolverá la posición actual del puntero al archivo dentro del mismo.
- **fseek()**: nos permite desplazarlo hacia una posición exacta.
- **feof()**: informa de si el puntero al archivo se encuentra al final del mismo.

Insertar texto en un archivo

Para insertar texto en un archivo usaremos las funciones **fwrite()** o **fputs()**, las cuales devolverán **false** en caso de error.

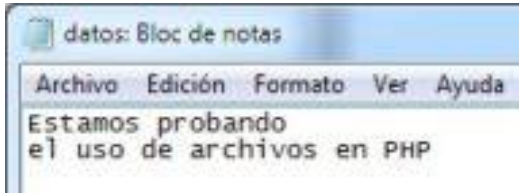
Cuando se escribe en un archivo los datos previamente son almacenados en un **buffer de escritura** (espacio en memoria RAM). El tamaño del buffer suele ser de 8K, y tal y como indica el manual de PHP:

Si hay dos **procesos esperando a escribir en un archivo**, cada uno se pausará después de escribir 8K de información para permitir que el otro escriba.

Con la función de PHP **set_file_buffer()** es posible definir el tamaño del buffer.

Insertar texto en un archivo

Usando la función **fflush()** podemos forzar que se escriban en el archivo los cambios pendientes en el buffer de escritura. Cuando se llama a **fclose()** se escribirán también los cambios pendientes.

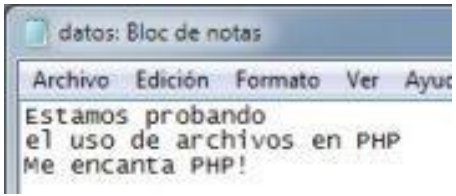



```php
<?php
   // Abrir el archivo, creándolo si no existe:
   $archivo = fopen("datos.txt","w+b");

   if( $archivo == false ) {
     echo "Error al crear el archivo";
   }
   else
   {
      // Escribir en el archivo:
      fwrite($archivo, "Estamos probando\r\n");
      fwrite($archivo, "el uso de archivos ");
      fwrite($archivo, "en PHP");

      // Fuerza a que se escriban los datos pendientes en el buffer:
      fflush($archivo);
   }

   // Cerrar el archivo:
   fclose($archivo);
?>
```

Insertar texto en un archivo

También podemos insertar texto en un archivo a partir de una cadena de texto usando la función de PHP **file_put_contents()**:

```
<?php
    $cadena = file_get_contents("datos.txt");

    $cadena .= "\r\nMe encanta PHP!";
    file_put_contents("datos.txt", $cadena);
?>
```

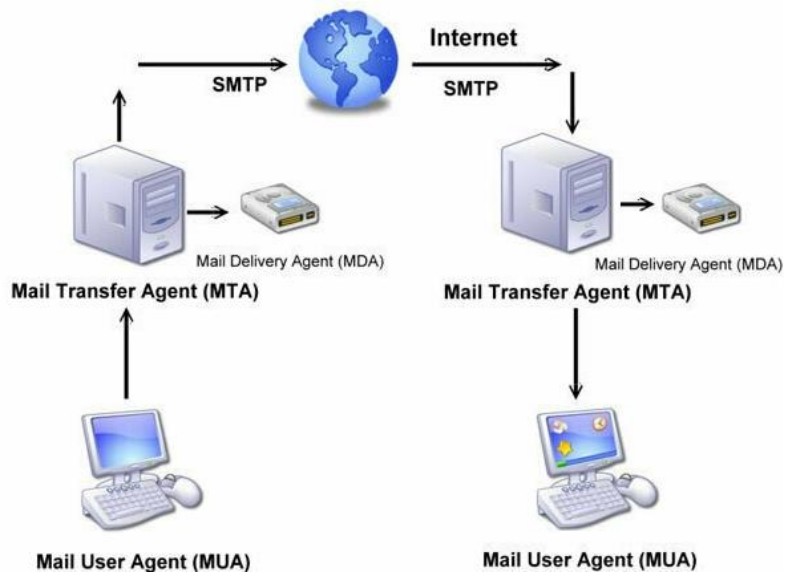


PHP

Envío de correo electrónico

Envío de correo electrónico en PHP

El enviar correo electrónico desde PHP es un proceso muy sencillo, aunque primero deberás realizar los cambios en el archivo de configuración de PHP **php.ini** , también es posible usar la función de PHP **ini_set()** para no tener que modificar el archivo.



Enviar un E-Mail desde PHP como texto plano

Los correos electrónicos pueden ser enviados como **texto plano** o en **formato HTML** (pudiendo con ello insertar imágenes, dar formato al texto, etc.). A continuación te mostramos un sencillo ejemplo de cómo **enviar un EMail con PHP** en texto plano.

```
<html>

   <head>
      <title>Enviar E-Mail desde PHP | informaticapc.com</title>
   </head>

   <body>
      <form name="frmMail" id="frmMail" action="01_mail_texto_plano.php"
method="post">

         Nombre: <input type="text" name="txtNombre" id="txtNombre" />
         EMail: <input type="text" name="txtMail" id="txtMail" />

         <p>Mensaje:</p>
         <textarea name="txtMensaje" id="txtMensaje" rows="10"
cols="60"></textarea><p/>

         <input type="submit" name="btnSubmit" value="Enviar" />

      </form>
   </body>

</html>
```

Enviar un E-Mail desde PHP como texto plano

```
<?php
    /*
        En caso de que no puedas enviar los correos electrónicos y no puedas o no quieras
        editar el archivo de configuración 'php.ini', descomenta las siguientes líneas con
        las que modificamos la configuración en tiempo de ejecución. Si es necesario, modifica
        el valor adecuado.
    */
    //ini_set('SMTP', "localhost");
    //ini_set('smtp_port', 25);
    //ini_set('sendmail_from', "postmaster@localhost.com");
    //ini_set('display_errors', "On");    // Mostrar los errores (usar sólo durante las pruebas)

    // Comprobar si llegaron los datos requeridos:
    if(  !empty($_POST) &&
         (isset($_POST['txtNombre'])  && !empty($_POST['txtNombre']))  &&
         (isset($_POST['txtMail'])    && !empty($_POST['txtMail']))    &&
         (isset($_POST['txtMensaje']) && !empty($_POST['txtMensaje']))
      )
    {
        $mensaje = "Mensaje de: ".$_POST['txtNombre'].PHP_EOL;
        $mensaje .= "EMail: ".$_POST['txtMail'].PHP_EOL.PHP_EOL;
        $mensaje .= $_POST['txtMensaje'];

        // Indicar cabecera con el nombre del remitente. Si no indicamos la dirección de correo puede que
        // no se realice el envío a a otros servicios como Hotmail o Yahoo
        $cabecera = "From: TU_NOMBRE <TU_CUENTA_DE_EMAIL@TU_SERVIDOR.com>";

        // IMPORTANTE: debes sustituir la dirección de correo por aquella en que deseas recibir el EMail:
        $ok = mail( trim($_POST['txtMail']), "Mensaje de prueba", $mensaje, $cabecera );

        if( $ok == true )
            echo "<p>El E-Mail ha sido enviado</p>";
        else
            echo "<p>ERROR al enviar el E-Mail</p>";

        echo "<p>Haz <a href='01_mail_texto_plano.html'>clic para volver al formulario</a></p>";
    }
    else
    {
        $html  = "<html>";
        $html .= "<head>";

        // Después de cuatro segundos de mostrarse esta página web de error se redirigiría a la URL especificada.
        $html .= "<meta http-equiv='refresh' content='4;url=01_mail_texto_plano.html'>";
```

Enviar un E-Mail desde PHP como texto plano

```
        // Después de cuatro segundos de mostrarse esta página web de error se
redirigiría a la URL especificada.
        $html .= "<meta http-equiv='refresh'
content='4;url=01_mail_texto_plano.html'>";

        $html .= "</head>";
        $html .= "<body>";
        $html .= "No han llegado todos los datos. En unos segundos será
redirigido a la página principal.";
        $html .= "</body>";
        $html .= "</html>";

        echo $html;
    }
?>
```

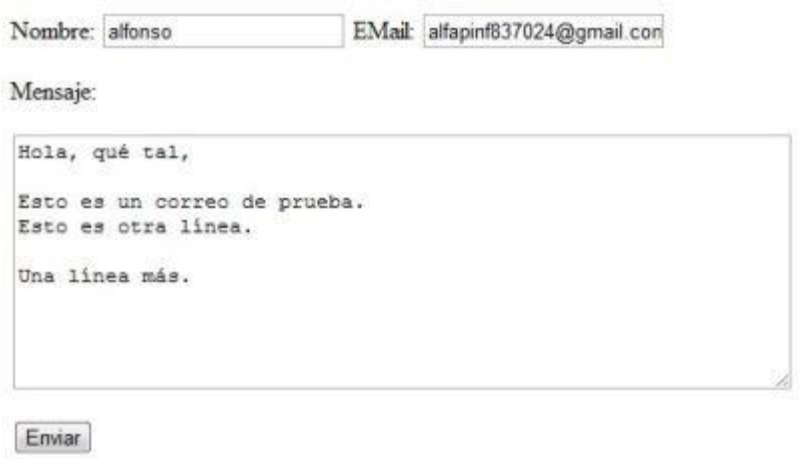
Enviar un E-Mail desde PHP como texto plano

Observa que hemos usado la función **ini_set()** para no tener que modificar el archivo de configuración.

Con **PHP_EOL** insertamos saltos de línea (en el formato adecuado según el sistema operativo usado).

Fíjate también que como primer parámetro de la función **mail()** debes introducir la dirección de correo electrónico del destinatario.

Al cargarse la página web **01_mail_texto_plano.html** veremos el siguiente formulario, en el que escribiremos el texto del mensaje:



Enviar un E-Mail desde PHP como texto plano

Al hacer clic en el botón **Enviar** se procesará el archivo **01_mail_texto_plano.php**, en el que se comprobará si han llegado los datos del formulario y si se introdujo el texto del mensaje para de ser así enviar el E-mail, mostrándose en todo caso el mensaje correspondiente.

Si todo ha ido bien deberías ver el mensaje informando de ello, y en unos pocos minutos (como mucho) llegaría el correo electrónico a la cuenta de correo indicada.

Enviar un E-Mail en formato HTML

A continuación te mostramos un ejemplo de como enviar correos electrónicos en formato HTML o en formato HTML.

```
<html>

   <head>
      <title>Enviar E-Mail desde PHP | informaticapc.com</title>
   </head>

   <body>
      <form name="frmMail" id="frmMail" action="01_mail_texto_plano.php" method="post">

         Nombre: <input type="text" name="txtNombre" id="txtNombre" />
         EMail: <input type="text" name="txtMail" id="txtMail" />

         <p>Mensaje:</p>
         <textarea name="txtMensaje" id="txtMensaje" rows="10" cols="60"></textarea><p/>

         <input type="submit" name="btnSubmit" value="Enviar" />

      </form>
   </body>

</html>
```

Enviar un E-Mail en formato HTML

```
<?php
    /*
        En caso de que no puedas enviar los correos electrónicos y no puedas o quieras
        editar el archivo de configuración 'php.ini', descomenta las siguientes líneas con
        las que modificamos la configuración en tiempo de ejecución. Si es necesario, modifica
        el valor adecuado.
    */
    //ini_set('SMTP', "localhost");
    //ini_set('smtp_port', 25);
    //ini_set('sendmail_from', "postmaster@localhost.com");
    //ini_set('display_errors', "On");    // Mostrar los errores (usar sólo durante las pruebas)

    // Comprobar si llegaron los datos requeridos:
    if(  !empty($_POST) &&
         (isset($_POST['txtNombre'])  && !empty($_POST['txtNombre']))  &&
         (isset($_POST['txtMail'])    && !empty($_POST['txtMail']))                    &&
         (isset($_POST['txtMensaje']) && !empty($_POST['txtMensaje']))
      )
    {
        $mensaje  = "Mensaje de: ".$_POST['txtNombre']."<br />";
        $mensaje .= "EMail: ".$_POST['txtMail']."<p />";
        $mensaje .= nl2br($_POST['txtMensaje']);
```

Enviar un E-Mail en formato HTML

```
// Indicar cabecera con el nombre del remitente. Si no indicamos la dirección de correo
puede que
        // no se realice el envío a a otros servicios como Hotmail o Yahoo
        $headers  ="Content-type: text/html; charset=iso-8859-1;";  // Especifica correo en
formato HTML
        $headers .= "From: TU_NOMBRE
<TU_CUENTA_DE_EMAIL@TU_SERVIDOR.com>";

        // IMPORTANTE: debes sustituir la dirección de correo por aquella en que deseas
recibir el EMail:
        $ok = mail( trim($_POST['txtMail']), "Mensaje de prueba", $mensaje, $headers );

        if( $ok == true )
           echo "<p>El E-Mail ha sido enviado</p>";
        else
           echo "<p>ERROR al enviar el E-Mail</p>";

        echo "<p>Haz <a href='02_mail_html.html'>clic para volver al formulario</a></p>";
    }
    else
    {
        $html  = "<html>";
        $html .= "<head>";
```

Enviar un E-Mail en formato HTML

```
        // Después de cuatro segundos de mostrarse esta página web de error se redirigiría a
la URL especificada.
        $html .= "<meta http-equiv='refresh' content='4;url=02_mail_html.html'>";

        $html .= "</head>";
        $html .= "<body>";
        $html .= "No han llegado todos los datos. En unos segundos será redirigido a la página
principal.";
        $html .= "</body>";
        $html .= "</html>";

        echo $html;
    }
?>
```