

Laboratorio II / Laboratorio de Software II 2024

Clase Teórica 05: Funciones para cadenas, header, Manejo de Archivos de texto, Subir archivos



Docente: Myriam Ruiz
Licenciada en Informática
Profesora en Computación y Matemática
Programadora Universitaria



Funciones para Cadenas

- **implode** (**string** \$pegamento , **array** \$partes) : **string**
 - Concatena elementos de un array, agregando un texto de pegamento entre cada componente, que puede ser cualquier caracter. Devuelve un string
- **explode** (**string** \$delimitador , **string** \$cadena) : **array**
 - Separa una cadena y asigna las partes en las componentes de un array. Habrá tantas componentes como texto quede separadp por el delimitador. Devuelve un array

Ejemplo implode y explode

```
<article>
  <?php
    $cliente = ['DNI' => 44126354, 'Apellido' => 'Rivas',
                'Nombre' => 'Mariana', 'FechaNac' => '25-11-2002'
                ];
    echo '<h3>Arreglo original: </h3>';
    print_r($cliente);
    echo '<hr>';
    $linea = implode(';', $cliente); // une agregando en medio ;
    echo '<h3>Datos unidos: </h3><p>' . $linea . '</p>';
    echo '<hr>';
    $arregloCliente = explode(';', $linea); // separa en cada ;
    echo '<h3>Nuevo arreglo: </h3>';
    print_r($arregloCliente);
  ?>
```

Ejemplo implode y explode

← → ↻ ⓘ localhost/Ejemplos-Clase05/02-explode-implode/
implode y explode
Arreglo original:
Array ([DNI] => 44126354 [Apellido] => Rivas [Nombre] => Mariana [FechaNac] => 25-11-2002)
Datos unidos:
44126354;Rivas;Mariana;25-11-2002
Nuevo arreglo:
Array ([0] => 44126354 [1] => Rivas [2] => Mariana [3] => 25-11-2002)

Funciones para Cadenas

- **trim**: Elimina espacio en blanco (u otro tipo de caracteres) del inicio y el final de la cadena
- **trim**(string \$cadena, string \$character_mask = " \t\n\r\0\x0B"): string

\$character_mask:

- " " (ASCII 32 (0x20)), espacio simple.
- "\t" (ASCII 9 (0x09)), tabulación.
- "\n" (ASCII 10 (0x0A)), salto de línea.
- "\r" (ASCII 13 (0x0D)), retorno de carro.
- "\0" (ASCII 0 (0x00)), el byte NUL.
- "\x0B" (ASCII 11 (0x0B)), tabulación vertical.

Funciones para Cadenas

- Forma de uso:

`$cadena2 = trim($cadena1, 'subcadena ');` // elimina espacios en blanco o subcadena, si se indica

- Existen más funciones: <https://www.php.net/manual/es/ref.strings.php>
- Se deja para conocer, pero se espera que usen sólo las vistas en clase

Funciones para Cadenas – Ejemplo trim

- Ejemplo: Cada vez que un usuario ingresa un dato para nuestro script deja espacios antes y después del dato, lo que dificulta poder búscalos luego, ya que no hay coincidencias por los espacios extra que tiene. Además, a veces agrega una llave } por error porque se encuentra cerca del Enter.
- Realice un script que le permita limpiar el texto ingresado.

Funciones para Cadenas – Ejemplo trim

```
<main>
  <section>
    <article>
      <?php
        $producto = ' monitor 20" }';
        $producto = trim($producto, '}'); // quita }
        $producto = trim($producto); // quita blancos
        if ($producto == 'monitor 20"'){
          echo '<p>Ingreso limpiado con éxito</p>';
        } else {
          echo '</p>No coincide, ingresó con blancos</p>';
        }
      ?>
```


Otras funciones necesarias – Para Redireccionar

- Hasta ahora, si un usuario no había ingresado los datos como se esperaba, sólo le mostrábamos un mensaje de error y a lo sumo hacíamos un enlace para volver a la página con el formulario (no todos lo hacían, no se pedía).
- Sin embargo, para un usuario esa forma de trabajo es tediosa, porque tiene que estar volviendo manualmente a la página anterior. Es por eso que a partir de esa clase usaremos la función **header()**, para poder redireccionar a otra página de manera automática.

Otras funciones necesarias - Redireccionar

- **header** ('refresh:tiempo; url=php/página.php');
- Redirecciona hacia otra página
- **refresh** recibe el tiempo en segundos, que queremos que espere antes de redireccionar
- **url**: establece hacia qué pagina queremos que nos redireccione. Debe tener la ruta relativa, como en un enlace.

Otras funciones necesarias - Redireccionar

- Ejemplo: Hacer el control de un ingreso de stock. Si falta un dato, debe redireccionar a la página index.php.

```
<?php
    // Comprueba si los campos no estan vacios
    if (!empty($_POST['producto']) && !empty($_POST['cantidad'])
        && !empty($_POST['precio'])) {
        $producto = $_POST['producto'];
        $cantidad = $_POST['cantidad'];
        $precio = $_POST['precio'];
        // aqui se debe almacenar...
    } else {
        // Si algun campo esta vacio, redirige de nuevo al formulario de ingreso
        header('refresh:2 ; url = ../index.php'); ←
    }
?>
```

Manipular archivos de texto

- Un archivo de texto se puede:
 - Abrir
 - Escribirle
 - Leerlo
 - Cerrarlo
 - Eliminarlo

Uso de la función header() - Recomendación

- Revisar todo el código (incluso las funciones), para no hacer echo de html antes de header().
- Esto se debe a que algunos hostings dan error si se hace de otra forma (se verá más adelante).
- Los require de html deben hacerse luego del header, por ejemplo:

```
} else {  
  
    header("refresh:3;url=../index.php");  
    require_once('../html/header.html');  
    echo '<p>Usuario o contraseña incorrectos</p>';  
  
}
```

Manipular archivos de texto

- Abrir archivo:

```
fopen(  
    string $filename,  
    string $mode,  
    bool $use_include_path = false,  
    resource $context = ?  
): resource
```

Manipular archivos de texto

- Abrir: `$archivo = fopen ($ubicacionArchivo , 'modo')`

modo	Descripción
'r'	Apertura para sólo lectura; inicia desde el principio del archivo.
'r+'	Apertura para lectura y escritura; inicia desde el principio del archivo.
'w'	Apertura para sólo escritura; inicia al principio del archivo y sobrescribe lo que tenga. Si el archivo no existe intenta crearlo.
'w+'	Apertura para lectura y escritura; inicia al principio del archivo y sobrescribe lo que tenga. Si el archivo no existe intenta crearlo.
'a'	Apertura para sólo escritura; inicia al final del archivo. Si el archivo no existe intenta crearlo.
'a+'	Apertura para lectura y escritura; inicia al final del archivo. Si el archivo no existe intenta crearlo.

Manipular archivos de texto

- Agregar líneas a un archivo:

- fwrite o fputs:

fputs(resource \$archivo, string \$data, [int \$length = null]): int|false

- Al finalizar, fputs devuelve cantidad de bytes escritos o false si no pudo escribir

- Leer líneas de un archivo, hasta marca de fin de línea

fgets(resource \$archivo, [int \$length = null]): string|false

- Cerrar archivo: **fclose**(resource \$archivo): bool

Manipular archivos de texto - Usos

- Agregar líneas a un archivo:
 - fwrite o fputs: `fputs($archivo, $texto . PHP_EOL);`
 - PHP_EOL: agrega marca de fin de línea
- Leer líneas de un archivo, hasta marca de fin de línea
 - `$dato = fgets($archivo [, cantBytes]);` // `cantBytes` es opcional
- Cerrar archivo: `fclose ($archivo)`

Manipular archivos de texto

- Comprobar si es el final del archivo:
 - `feof (resource $archivo) : bool`
 - Devuelve true o false
- Vamos a utilizar feof para avanzar por el archivo mientras no sea el final de este. Por lo tanto lo usaremos:

```
while (!feof($archivo)) {  
}
```

Otras funciones necesarias - Carpetas

- La primera vez que guardemos datos en un archivo de texto, deberemos indicar donde se encuentra ubicado.
- Si la carpeta en la que queremos guarda no existe, php nos mostrará un error. Ante eso, tenemos 2 opciones, 1) Crear siempre manualmente la carpeta donde estará el archivo, o 2) que nuestro código se encargue de controlar que la carpeta existe y sino es así crearla.
- Para ello usaremos las funciones `file_exists()` y `mkdir()`

Otras funciones necesarias - Carpetas

- `file_exists` (`string` \$nombreCarpeta) : `bool`
 - Controla si existe una carpeta o archivo, incluida la ruta de la carpeta. Devuelve true si existe y false si no existe.
 - Por lo tanto usaremos `!file_exists($carpeta)`, para controlar el caso de que no exista.
- `mkdir` (`string` \$ruta) : `bool`
 - Crea una carpeta en la ruta indicada. Tenga en cuenta que se seguirá la ruta desde donde esté el código php que la invoca, ingresando a una carpeta o saliendo, tal cual como sucede con los enlaces.

Manipular archivos de texto - Ejemplo


- Se desea crear un script que permita cargar los datos para suscripción y almacenarlos en un archivo de texto. Para ellos:
- 1. Crear un formulario en un archivo de nombre index.php. Los datos cargados en el formulario deberán enviarse al archivo procesar.php.
- 2. En procesar.php se controlará que los datos que llegaron no sean vacíos, y se procederá a guardarlos en un archivo de texto de nombre suscripciones.txt (que estará en la carpeta txt).

Manipular archivos de texto - Ejemplo

- index.php

← → ↻ ⓘ localhost/Ejemplos-Clase05/03-guardar-archivo/

Formulario

Apellido:	<input type="text"/>
Nombre:	<input type="text"/>
Fecha de Nacimiento:	<input type="text" value="31/03/2000"/> 
Correo:	<input type="text" value="correo@empresa.com"/>

Manipular archivos de texto - Ejemplo

- index.php

```
<?php
    require_once 'html/header.html';
?>

    <section>
        <article>
            <form action="php/guardar.php" method="post">
                <label for="apel">Apellido: </label>
                <input type="text" name="apellido" id="apel" maxlength="50">
                <label for="nom">Nombre: </label>
                <input type="text" name="nombre" id="nom" maxlength="20">
                <label for="fech">Fecha de Nacimiento: </label>
                <input type="date" name="fechaNac" id="fech" value="2000-03-31">
                <label for="cor">Correo: </label>
                <input type="email" name="correo" id="cor" placeholder="correo@empresa.com">
                <input type="submit" value="Guardar">
            </form>
```

Manipular archivos de texto - Ejemplo

- guardar.php

```
1  <?php
2      require_once '../html/header.php';
3      if (!empty($_POST['apellido']) && !empty($_POST['nombre']) &&
4          !empty($_POST['fechaNac']) && !empty($_POST['correo'])
5      ) {
6          echo '<p>Estan todos los datos</p>';
7          $linea = implode(';', $_POST); // no dejar espacios!!
8          echo '<p>' . $linea . '</p>';
9          $carpeta = '../txt/';
10         if (!file_exists($carpeta)) {
11             echo '<p>No existe la carpeta</p>';
12             mkdir($carpeta);
13         }
14         $nombre = 'inscripciones.txt';
15         $archivo = fopen($carpeta.$nombre, 'a'); // no dejar espacios!!
```


Manipular archivos de texto - Ejemplo

- guardar.php

```
15     $archivo = fopen($carpeta.$nombre,'a'); // no dejar espacios!!
16     if ($archivo) { // verifica que se haya abierto el archivo
17         $control = fputs($archivo, $linea . PHP_EOL);
18         if ($control) { // verifica que se haya podido guardar
19             echo '<p>Guardado exitoso</p>';
20             fclose($archivo);
21         } else {
22             '<p>No se pudo guardar los datos</p>';
23         }
24     } else {
25         '<p>No se pudo abrir el archivo</p>';
26     }
27     ?>
28     <table>
```

Manipular archivos de texto - Ejemplo

- guardar.php

```
<table>
  <thead>
    <th>Apellido</th><th>Nombre</th><th>Fecha</th><th>Correo</th>
  </thead>
  <tbody>
<?php
  // Mostrar archivo
  $archivo = fopen($carpeta.$nombre,'r'); // no dejar espacios!!
  while (!feof($archivo)) {
    $linea = fgets($archivo);
    if ($linea != '') {
      $separada = explode(';', $linea);
    }
    ?>
    <tr>
      <td><?php echo $separada[0]; ?></td>
      <td><?php echo $separada[1]; ?></td>
      <td><?php echo $separada[2]; ?></td>
      <td><?php echo $separada[3]; ?></td>
    </tr>
```

Manipular archivos de texto - Ejemplo

- guardar.php

```
</tr>
<?php
    }
}
?>
</tbody>
</table>
<?php
    } else {
        echo '<p>Faltan datos</p>';
        header ('refresh:3; url=../index.php');
    }

require_once '../html/footer.html';

?>
```

Manipular archivos de texto - Ejemplo

- En un archivo llamado **misfunciones.php**, cree dos procedimientos:
 - **prGuardar**, que reciba el nombre, peso y estatura de una persona, y el nombre del archivo donde guardar, y cuya tarea sea calcular el índice de masa corporal utilizando la fórmula 1 y guardar todo en un archivo de texto, llamado **imc.txt**, de **modo** que se puedan agregar líneas en distintos momentos.
 - **prMostrar**, que recibe el nombre de un archivo, y cuya tarea será abrir el archivo generado por el otro procedimiento y mostrar línea a línea su contenido.
- Cree una página que contenga un formulario con los elementos necesarios, con nombre **index.html**.
- Cree otra página llamada **procesar.php** que procesará los datos del formulario y se encargará de llamar al procedimiento **prGuardar**
- Crear una última página llamada **mostrar.php** que llame al procedimiento **prMostrar**.
- Fórmula 1: $IMC = \text{peso corporal [kg]} / (\text{altura [m]})^2$

Manipular archivos de texto - Ejemplo

index.html

```
<main>
  <section>
    <article>
      <form action="../../../php/procesar.php" method="post">
        <label for="ap">Apellido</label>
        <input type="text" name="apellido" id="ap">
        <label for="pe">Peso (en Kg)</label>
        <input type="number" step="0.01" name="peso" min="0" id="pe">
        <label for="es">Estatura (en metros)</label>
        <input type="number" step="0.01" name="estatura" min="0" id="es">
        <fieldset>
          <input type="submit" value="Guardar">
          <input type="reset" value="Limpiar">
        </fieldset>
      </form>
```

Manipular archivos de texto - Ejemplo

procesar.php

```
<?php
    include_once 'mis-funciones.php';
    $nombArchivo = 'imc.txt';
    if (!empty($_POST['apellido']) &&
        !empty($_POST['peso']) &&
        !empty($_POST['estatura'])) {
        $datos = implode(';', $_POST); // concatena lo recibido del formulario
        prGuardar($datos, $nombArchivo);
        header ('refresh:0;url=../php/formulario-ingreso.php'); // redirecciona
    } else {
        header ('refresh:3;url=../php/formulario-ingreso.php'); // redirecciona
        include_once '../php/encabezado.php';
        echo '<p class="msje"> # Faltan Datos!!</p>';
        include_once '../php/pie.php';
    }
?>
```

Manipular archivos de texto - Ejemplo

mis-funciones.php

```
<?php
function prGuardar($datos, $nombArchivo)
{
    $separados = explode(';', $datos); // separa los datos concatenados
    // calcula índice de masa corporal
    $imc = $separados[1] / ($separados[2]*$separados[2]);
    $imc = round($imc, 2); // redondea a 2 decimales
    // concatena los datos con el $imc y el delimitador de final de línea
    $datos .= ';' . $imc . PHP_EOL; // PHP_EOL fin de línea
    $ubicacion = '../txt/';
    if (!file_exists($ubicacion)) { // controla si no existe la carpeta
        mkdir($ubicacion); // crea la carpeta si no existe
    }
    $archivo = fopen($ubicacion.$nombArchivo, 'a'); // abre para sólo escritura
    fputs($archivo, $datos); // se guarda $datos en el archivo de texto
    fclose($archivo); // se cierra el archivo
}
```

Hacer los controles de diapo 25

Manipular archivos de texto - Ejemplo

Continuación mis-funciones.php

```
function prMostrar ($nombArchivo)
{
    $ubicacion = '../txt/';
    $archivo = fopen($ubicacion.$nombArchivo, 'r'); // abre para sólo lectura
    while (!feof($archivo)) { // mientras no sea el final del archivo
        $linea = fgets($archivo); // obtiene una línea del archivo
        if ($linea != '') { // controla que no esté vacía
            $separado = explode(';', $linea); // separa línea en un array
            echo '<p>Nombre: ' . $separado[0] . '</p>';
            echo '<p>IMC: ' . $separado[3] . '</p>';
            echo '<hr>'; // hace una línea
        }
    }
    fclose($archivo); // se cierra el archivo
}
```

?>

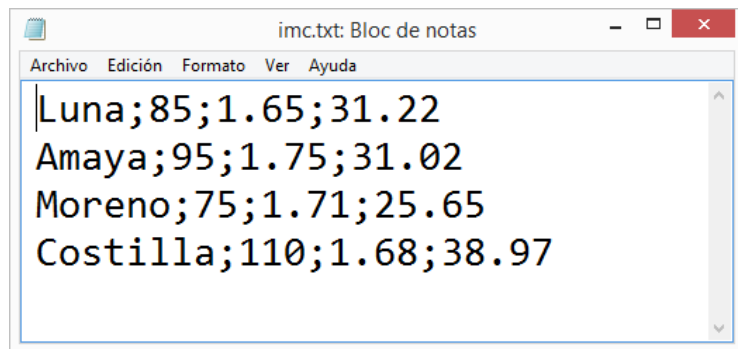
Manipular archivos de texto - Ejemplo

mostrar.php

```
<?php
    include 'mis-funciones.php';
    include '../php/encabezado.php';
    echo '<h2>Pacientes que consultaron</h2>';
    prMostrar('imc.txt');
    include '../php/pie.php';
?>
```

Manipular archivos de texto - Ejemplo

Archivo de texto generado



Mostrado del contenido del archivo

Centro de Estética Kalos	
Índice de Masa Corporal	Mostrar
Pacientes que consultaron	
Nombre: Luna IMC: 31.22	
Nombre: Amaya IMC: 31.02	
Nombre: Moreno IMC: 25.65	
Nombre: Costilla IMC: 38.97	

Manipular archivos de texto - Ejemplo

Otra forma de Mostrar

Mostrado del contenido del archivo

Pacientes que

Nombre	IMC
Luna	31.22
Amaya	31.02
Moreno	25.65
Costilla	38.97
Ruiz	43.42
Corrales	23.03

Nombre	IMC
Luna	31.22
Amaya	31.02
Moreno	25.65
Costilla	38.97
Ruiz	43.42
Corrales	23.03

```
function prMostrarTabla ($nombArchivo)
{
    echo '<table>';
    echo '<thead><th>Nombre</th><th>IMC</th></thead>';
    echo '<tbody>';
    $ubicacion = '../txt/';
    $archivo = fopen($ubicacion.$nombArchivo, 'r'); // abre para sólo lectura
    while (!feof($archivo)) { // mientras no sea el final del archivo
        $linea = fgets($archivo); // obtiene una línea del archivo
        if ($linea != '') { // controla que no esté vacía
            $separado = explode(';', $linea); // separa línea en un array
            echo '<tr>';
            echo '<td>' . $separado[0] . '</td>';
            echo '<td>' . $separado[3] . '</td>';
            echo '</tr>';
        }
    }
    echo '</tbody>';
    echo '</table>';
    fclose($archivo); // se cierra el archivo
}
```

Manipular archivos de texto - Ejemplo

Dado el siguiente archivo:

```
laboratorios.csv: Bloc de notas
Archivo Edición Formato Ver Ayuda
Laboratorio A;Block 3;1er Piso;labA.jpg
Laboratorio FOMECE;Block 3;1er Piso;FOMECE.jpg
Laboratorio B;Block 2;Planta Baja;laboB.jpeg
Laboratorio Redes;Block 2;Planta Baja;redes.jpg
Línea 1, columna 1 100% Windows (CRLF) UTF-8
```

Mostrado del contenido del archivo
como tabla

Nombre	Lugar	Piso	Imagen
Laboratorio A	Block 3	1er Piso	
Laboratorio FOMECE	Block 3	1er Piso	
			

Manipular archivos de texto - Ejemplo

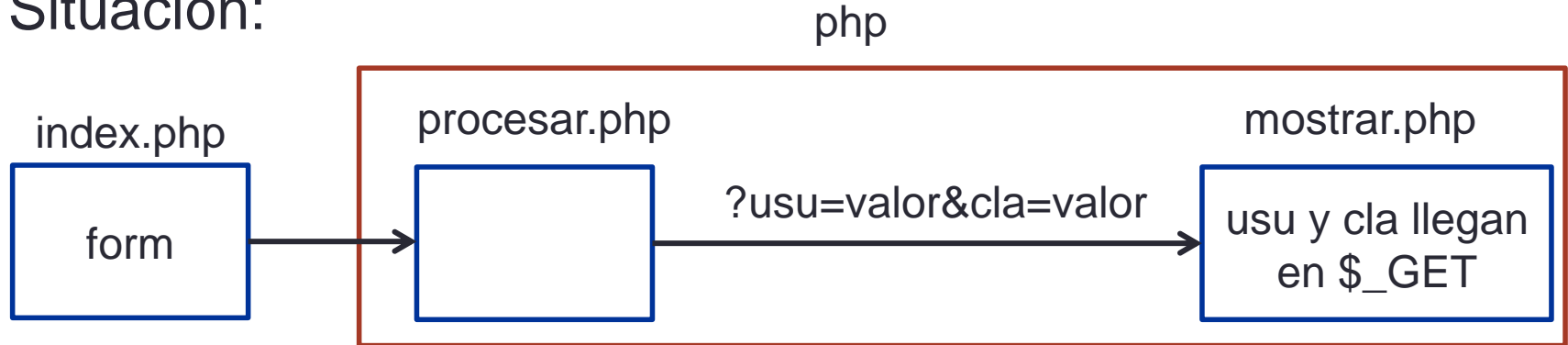
Código:

```
<table>
<thead><th>Nombre</th><th>Lugar</th><th>Piso</th><th>Imagen</th></thead>
<tbody>

<?php
$ubicacion = 'texto/';
$nombre = 'laboratorios.csv';
$sarchivo = fopen($ubicacion.$nombre, 'r'); // abre para sólo lectura
while (!feof($sarchivo)) { // mientras no sea el final del archivo
    $linea = fgets($sarchivo); // obtiene una línea del archivo
    if ($linea != '') { // controla que no esté vacía
        $separado = explode(';', $linea); // separa línea en un array
        echo '<tr>';
        echo '<td>' . $separado[0] . '</td>';
        echo '<td>' . $separado[1] . '</td>';
        echo '<td>' . $separado[2] . '</td>';
        echo '<td> </td>';
        echo '</tr>';
    }
}
```

Pasar datos por la url

- Aprovechando que ya conocemos como se envían variables como el method get de formularios, pasaremos variables agregándolas a una url.
- Eso es válido al hacer enlaces o redireccionamientos con header
- Situación:



Pasar datos por la url

- index.php

```
<main>
  <form action="php/procesar.php" method="post">
    <section>
      <label for="lblUsu">Usuario:</label>
      <input type="text" id="lblUsu" name="usuario">
    </section>
    <section>
      <label for="lblClave">Clave:</label>
      <input type="password" id="lblCla" name="clave">
    </section>
    <input type="submit" value="Ingresar">
  </form>
</main>
```

Pasar datos por la url

- procesar.php

```
<?php
    if (!empty($_POST['usuario']) && !empty($_POST['clave'])) {
        $usuario = $_POST['usuario'];
        $clave = $_POST['clave'];
        header('refresh:3; url=mostrar.php?usu=' . $usuario . '&' . 'cla=' $clave);
        include 'header.php';
        echo '<p>El usuario es válido</p>';
    } else {
        header('refresh:3; url=../index.php');
        include 'header.php';
        echo '<p>El usuario no válido</p>';
    }
?>
```

* header () debe ir antes de cualquier código html (lo necesitamos así para ciertos hostings). Las instrucciones que siguen se ejecutan, aún cuando ya no se vea la página.

Pasar datos por la url

- mostrar.php

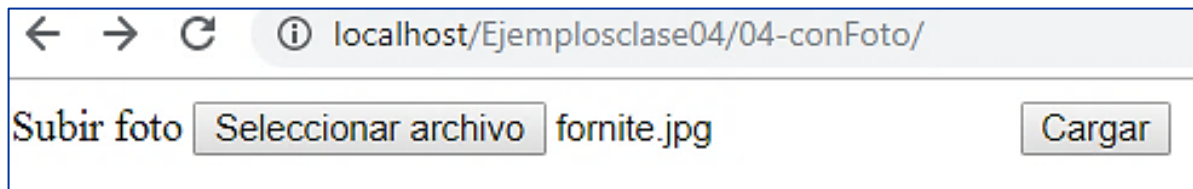
```
<?php
    if(!empty($_GET['usu']) && !empty($_GET['cla'])) {
        include 'header.php';
        echo '<p>Usuario: ' . $_GET['usu'] . '</p>';
        echo '<p>Clave: ' . $_GET['cla'] . '</p>';
    }
?>
```

* Las variables agregadas a la dirección, se guardan en el arreglo
\$_GET

Subir archivos al Servidor

- Novedades en el Formulario para poder subir archivos

```
<form action="php/procesar.php" method="post" enctype="multipart/form-data">  
  <label for="fot">Subir foto</label>  
  <input type="file" name="foto" id="fot">  
  <input type="submit" value='Cargar'>  
</form>
```



A screenshot of a web browser window. The address bar shows the URL 'localhost/Ejemplosclase04/04-conFoto/'. The page content displays a file upload form. It starts with the text 'Subir foto' followed by a file selection button labeled 'Seleccionar archivo'. To the right of this button, the filename 'fornite.jpg' is displayed. Further to the right is a button labeled 'Cargar'.

Formularios – Que pasa al enviar un archivo?

Se debe controlar que no esté vacía cada posición dentro de `$_FILES`, que coincide con los name de los input de tipo file

procesar.php

Suponemos un formulario que sólo permite enviar una imagen:

Se guarda la imagen en `$_FILES`

```
<?php
    if(!empty($_FILES['foto']['size'])) {
        |    var_dump($_FILES);
    }
}
```



← → ↻ ⓘ localhost/Ejemplosclase04/04-conFoto/

Subir foto fornite.jpg

← → ↻ ⓘ localhost/Ejemplosclase04/04-conFoto/php/procesar.php 🔍 ☆ 👤 ⋮

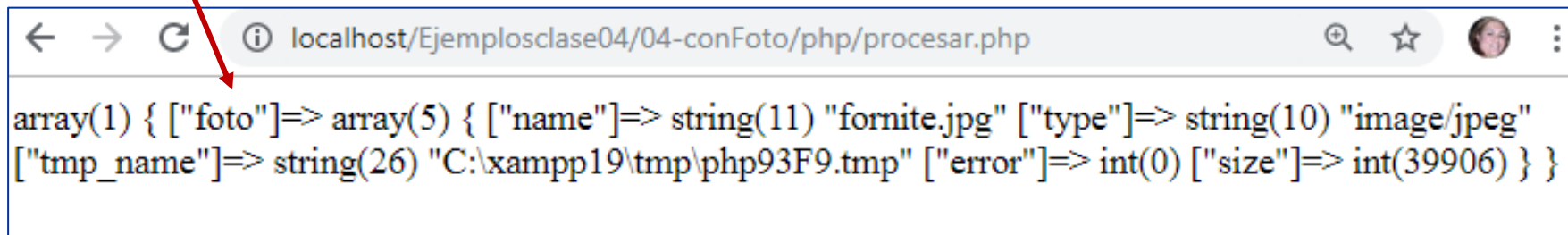
```
array(1) { ["foto"]=> array(5) { ["name"]=> string(11) "fornite.jpg" ["type"]=> string(10) "image/jpeg"
["tmp_name"]=> string(26) "C:\xampp19\tmp\php93F9.tmp" ["error"]=> int(0) ["size"]=> int(39906) } }
```

Subir archivos al Servidor

- procesar.php

name del input

```
<?php
    if(!empty($_FILES['foto']['size'])) {
        var_dump($_FILES);
    }
```



Al enviar el formulario se almacenan todos los datos de la imagen en el arreglo Superglobal `$_FILES`. Tendremos que tomar algunos datos de ese arreglo para poder transferir un archivo al servidor

Subir archivos al Servidor

- Limitar el tipo de archivo a subir (no lo reconocen todos los navegadores)

```
<form action="php/procesar.php" method="post" enctype="multipart/form-data">
  <legend>Cargado de archivos</legend>
  <label for="fot">Subir foto
    <input type="file" name="foto" accept="image/*" id="foto">
  </label>
  <label for="vid">Subir Video
    <input type="file" name="video" accept="video/*" id="vid">
  </label>
  <label for="txt">Subir Curriculum
    <input type="file" name="texto" accept=".doc,.docx,.odt,.pdf" id="txt">
  </label>
  <input type="submit" value='Cargar'>
</form>
```

Subir archivos al Servidor

- Cuando subimos 3 archivos sería algo así

Claves dentro
de \$_FILES

'foto' =>

'video' =>

'texto' =>

El dato asociado a las
claves, es un arreglo

name	type	tmp_name	Size
foto.jpg	img/jpeg	Ubicación temporal en servidor	En bytes

name	type	tmp_name	Size
video.mp4	video/mp4	Ubicación temporal en servidor	En bytes

name	type	tmp_name	Size
cv.pdf	application /pdf	Ubicación temporal en servidor	En bytes

Subir archivos al Servidor

- Cómo se podría acceder al name?

Clave dentro
de \$_FILES

→ 'foto' =>

name	type	tmp_name	Size
foto.jpg	img/jpeg	Ubicación temporal en servidor	En bytes

Se accede con `$_FILES['foto']['name']`

Obtendríamos foto.jpg

Subir archivos al Servidor


- Mover archivo hacia el servidor

`move_uploaded_file(string $filename, string $destination): bool`

- `$resultado = move_uploaded_file($origen, $destino)`
 - `$origen` es una lugar temporal, que se obtiene del arreglo `$_FILES`
 - `$destino` es un lugar definitivo dentro del servidor, que incluye carpeta y nombre de archivo.

Subir archivos al Servidor

- Ejemplo: Crear un formulario, para cargar el DNI y el CV de un postulante en una empresa, mandar el archivo hacia guardar.php, el nombre final del archivo será DNI.extensión

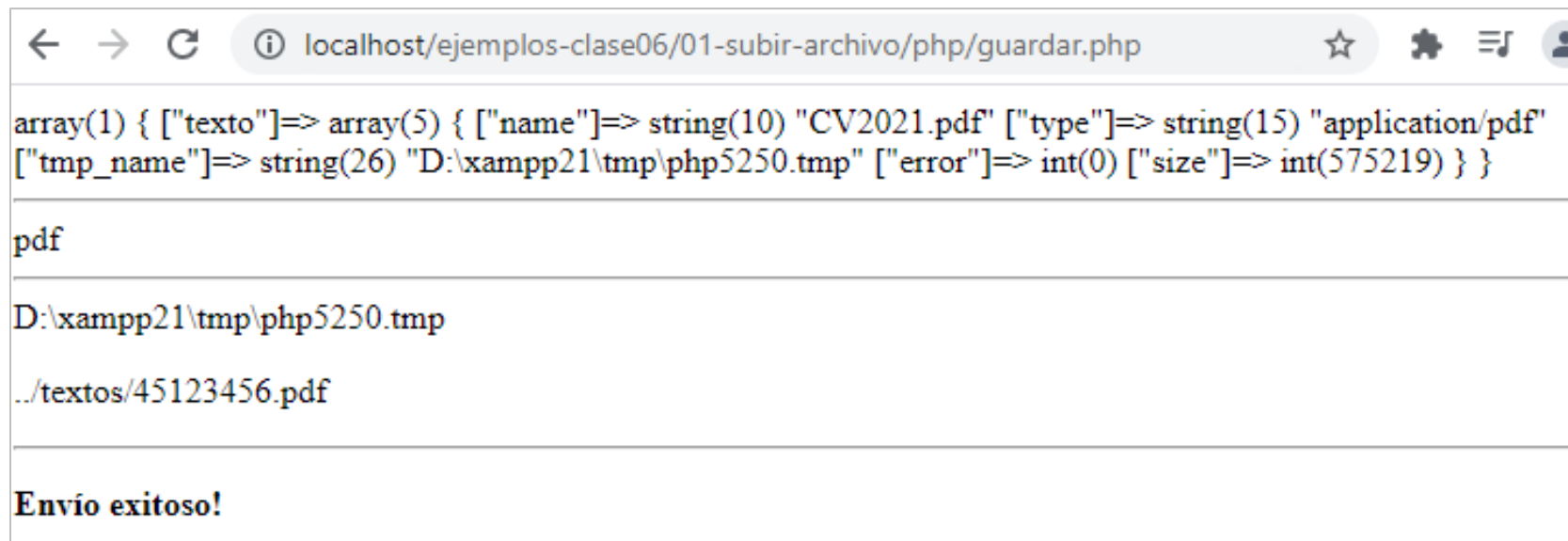


A screenshot of a web browser window showing a form titled "Cargado de archivo". The form is enclosed in a yellow box. It contains two input fields: "DNI (sin puntos)" with the value "45123456" and "Subir Curriculum" with a file selection button labeled "Seleccionar archivo" and the filename "CV2021.pdf". Below these fields is a "Cargar" button. At the bottom of the browser window, there is a contact number: "Contacto: 3816300503".

```
<article>
  <form action="php/guardar.php" method="post" enctype="multipart/form-data">
    <fieldset>
      <legend>Cargado de archivo</legend>
      <label for="docu">DNI (sin puntos) </label>
      <input type="text" name="dni" id="docu" maxlength="10">
      <label for="txt">Subir Curriculum </label>
      <input type="file" name="texto" accept=".doc,.docx,.odt,.pdf" id="txt">
      <input type="submit" value='Cargar'>
    </fieldset>
  </form>
</article>
```

Subir archivos al Servidor

- Resultado visible de guardar.php



A screenshot of a web browser window. The address bar shows the URL `localhost/ejemplos-clase06/01-subir-archivo/php/guardar.php`. The page content displays the output of a PHP script, which is a JSON array. The array contains one element, which is another array with five key-value pairs: 'texto' (an array), 'name' (a string), 'type' (a string), 'tmp_name' (a string), and 'error' (an integer). Below the JSON output, the file type 'pdf' is shown. Then, the temporary file path `D:\xampp21\tmp\php5250.tmp` is displayed. Finally, the relative file path `../textos/45123456.pdf` is shown. At the bottom of the page, the message **Envío exitoso!** is displayed.

```
array(1) { ["texto"]=> array(5) { ["name"]=> string(10) "CV2021.pdf" ["type"]=> string(15) "application/pdf" ["tmp_name"]=> string(26) "D:\xampp21\tmp\php5250.tmp" ["error"]=> int(0) ["size"]=> int(575219) } }
```

pdf


D:\xampp21\tmp\php5250.tmp

../textos/45123456.pdf

Envío exitoso!

Subir archivos al Servidor

- Cómo queda guardado el archivo:

o vol (D:) > xampp21 > htdocs > Ejemplos-Clase06 > 01-subir-archivo > textos				▼	🔄
Nombre		Fecha de modificación	Tipo	Tamaño	
 45123456.pdf		24/9/2021 21:37	Foxit PDF Reader ...	562 KB	

Subir archivos al Servidor

- Código de guardar.php

```
if (!empty($_FILES['texto']['size']) && !empty($_POST['dni'])) {  
    $tipoArchivo = $_FILES['texto']['type'];  
    if ($tipoArchivo == 'application/msword' || $tipoArchivo = 'application/pdf'  
        || $tipoArchivo = 'application/vnd.oasis.opendocument.text') {  
        var_dump($_FILES); /* sólo en etapa de testeo, para observar lo enviado */  
        $nombre = $_FILES['texto']['name'];  
        echo '<hr>';  
        $ext = explode('.', $nombre);  
        echo $ext[1]; // $ext[1] contiene la extensión del archivo  
        $rutaOrigen = $_FILES['texto']['tmp_name'];  
        echo '<hr>';  
        echo $rutaOrigen;  
        $nuevoNombre = $_POST['dni']; // toma el dni para formar el nuevo nombre  
        $destino = '../textos/' . $nuevoNombre . '.' . $ext[1];  
        echo '<p>' . $destino . '</p>'; /* muestra como queda el nombre */  
        $envio = move_uploaded_file($rutaOrigen, $destino);  
        echo '<hr>';  
        if ($envio) {  
            echo '<p><strong>Envío exitoso!</strong></p>';  
        }  
    }  
}
```

Eliminar archivos del Servidor

- De ser necesario, se puede eliminar un archivo:

```
$ubicacionArchivo = '../img/clase/foto1.jpg';  
unlink($ubicacionArchivo);
```

Subir archivos al servidor

- Tener en cuenta que:
 - Si subimos dos archivos con el mismo nombre, el 2do reemplaza al que estaba antes (pensar como darle nombres distintos a los originales)
 - No todos los navegadores interpretan **accept** en los formularios, por tanto deberá controlarse que sean archivos válidos al procesar los datos
 - Los servidores tienen una limitación en el tamaño del archivo a subir (se puede modificar)

Subir más de un archivo

- Se puede subir varios archivos con distintos input de tipo file
- O se puede subir con un único input de tipo file, utilizando el atributo **multiple** (no lo veremos en esta materia)
 - Además, se deberá indicar que el name del input es un arreglo, colocando **[]** después del nombre

Subir archivos al servidor – Limitación de tamaño

- En el caso de Xampp se puede modificar el tamaño de archivo editando el archivo php.ini.
- Ir a **c:\xampp\php** y abrir el archivo **php.ini**, modificar las líneas:
 - upload_max_filesize
 - post_max_size
 - memory_limit
 - max_execution_time
- Luego de guardar cambios, apagar y encender de nuevo apache

Subir archivos al servidor – Limitación de tamaño

- Se debería agregar un input de tipo hidden (oculto) con un atributo que limita el tamaño (medido en bytes) de subida de archivo, ubicándolo antes del input de tipo file:
- `<input type='hidden' name= 'MAX_FILE_SIZE' value= '6300000'>`
- (6300000 es aproximadamente 6MB)