

# Laboratorio II / Laboratorio de Software II 2022

---

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



**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);
?>
```

*Handwritten annotations:*

- A circle around `$cliente` with a line pointing to the text `array`.
- An arrow from `implode` to the semicolon in the array definition, labeled *pegamento* (glue).
- An arrow from `explode` to the semicolon in the string, labeled *elemento del array* (array element).
- An arrow from `explode` to the resulting array, labeled *string (cadena)* (string).

# Ejemplo implode y explode

← → ↻ ⓘ localhost/Ejemplos-Clase05/02-explode-implode/

**implode y explode**

**Arreglo original:** *array*

Array ( [DNI] => 44126354 [Apellido] => Rivas [Nombre] => Mariana [FechaNac] => 25-11-2002 )

**Datos unidos:** *implode*

44126354;Rivas;Mariana;25-11-2002

**Nuevo arreglo:** *explode*

Array ( [0] => 44126354 [1] => Rivas [2] => Mariana [3] => 25-11-2002 )

# Funciones para Cadenas

---

- `$cadena2 = trim($cadena1, 'subcadena ');` // elimina espacios en blanco o subcadena, si se indica
- Existen más funciones: <https://www.php.net/manual/es/ref.strings.php>

# Funciones para Cadenas – Ejemplo trim

---

- 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" }';
        es 1 → $producto = trim($producto, '}'); // quita }
        ej 2 → $producto = trim($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.

# Manipular archivos de texto

---

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

# 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: *→ escribe en el 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"/>
<input type="button" value="Guardar"/>	

# Manipular archivos de texto - Ejemplo

- index.php *→ en el index va el formulario*

```
<?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

Caso 1. escribimos en un archivo lo del formulario

- guardar.php

NOTA

recuerda que

en `$_POST['nombre']`



Usa el  
NAME

```
<?php
```

```
require_once '../html/header.html';
if (!empty($_POST['apellido']) && !empty($_POST['nombre']) &&
    !empty($_POST['fechaNac']) && !empty($_POST['correo']))
{
    echo '<p>Estan todos los datos</p>';
    ($linea = implode(';', $_POST); // no dejar espacios!!
    echo '<p>' . $linea . '</p>';
    $carpeta = '../txt/';
    if (!file_exists($carpeta)) {
        echo '<p>No existe la carpeta</p>';
        mkdir($carpeta);
    }
    $nombre = 'inscripciones.txt';
    $archivo = fopen($carpeta.$nombre, 'a'); // no dejar espacios!!
    fputs($archivo, $linea . PHP_EOL);
    fclose($archivo);
    echo '<p>Guardado exitoso</p>';
```

→ si el formulario no está vacío

hacemos la línea q' va a escribir

→ si no existe la carpeta

crea la carpeta

abrimos el archivo

→ escribimos la línea

→ cerramos el archivo

# Manipular archivos de texto - Ejemplo

CASO 2: exploramos un archivo y lo mostramos en pantalla.

- guardar.php

<table> → en este ejemplo lo ponen en una tabla ordenado.

```
<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)) { mientras que no sea el final del archivo
```

```
    $linea = fgets($archivo); obtener las líneas del archivo
```

```
    if ($linea != '') { si la línea es ≠ el vacío
```

```
        $separada = explode(';', $linea); separamos la línea como un ARRAY
```

```
?>
```

```
<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>
```

abrimos el archivo para LECTURA!

} agarramos los elementos del mismo array p/ mostrar en la tabla

# 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

- 1) • En un archivo llamado **misfunciones.php**, cree dos procedimientos:
    - A) • **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.
    - B) • **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.
  - 2) • Cree una página que contenga un formulario con los elementos necesarios, con nombre **index.html**.
  - 3) • Cree otra página llamada **procesar.php** que procesará los datos del formulario y se encargará de llamar al procedimiento **prGuardar**.
  - 4) • 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 → formulario

```
<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 → archivo donde se concatena  
los datos del formulario

```
<?php
    include '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=../html/formulario-ingreso.html'); // redirecciona
    } else {
        include '../html/encabezado.html';
        echo '<p class="msje"> # Faltan Datos!!</p>';
        include '../html/pie.html';
        header ('refresh:3;url=../html/formulario-ingreso.html'); // redirecciona
    }
?>
```



# Manipular archivos de texto - Ejemplo

mis-funciones.php → donde se procesa el archivo y anotar

```
<?php
function prGuardar($datos, $nombArchivo) → función que guarda lo impreso
{
    $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
}
```

# Manipular archivos de texto - Ejemplo

## Continuación mis-funciones.php

```
function prMostrar ($nombArchivo) → función que obtiene lo del txt y lo recupera p/ mostrarlo
{
    $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

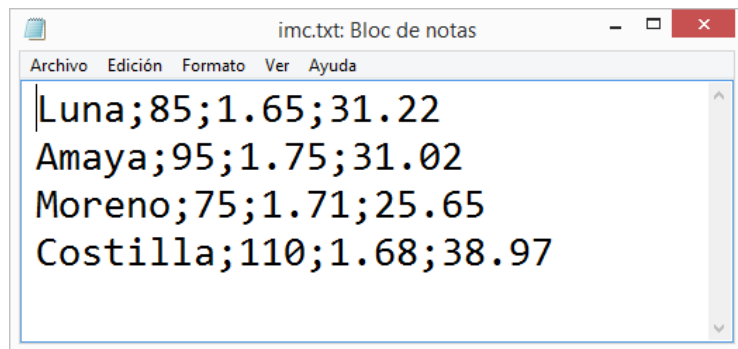
---

mostrar.php

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

# Manipular archivos de texto - Ejemplo

Archivo de texto generado



Mostrado del contenido del archivo

Centro de Estética Kalos	
Índice de Masa Corporal	Mostrar
<b>Pacientes que consultaron</b>	
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

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

Mostrado del contenido del archivo

```
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>';
    }
}
```