

1.2: Lenguajes de Programación: Web con PHP

27 Septiembre 2022



Educar en la Verdad
para ser libres



1.4: Lenguajes de Programación: Web con PHP

PHP es un lenguaje de guiones de propósito general, pero diseñado para el desarrollo de páginas web dinámicas utilizando código embebido dentro del lenguaje de marcas.

Su sintaxis está basada en la de C / C++, y por lo tanto es muy similar a la de Java. Aunque lo puedes hacer de otras formas, los delimitadores recomendados para incluir código PHP dentro de una página web son `<?php` y `?>`

El código se ejecuta por un entorno de ejecución con el que se integra el servidor web (normalmente utilizando Apache con el módulo **mod_php**). La configuración tanto del servidor web Apache, como de PHP, se realiza por medio de ficheros de configuración.

El de Apache es **httpd.conf**, y el de PHP es **php.ini**. Este fichero, **php.ini**, puede encontrarse en distintas ubicaciones. La función **phpinfo()** nos informa, entre otras muchas cosas, del lugar en que se encuentra almacenado el fichero **php.ini** en nuestro ordenador.

El fichero de configuración (**php.ini**) es leído al arrancar PHP. En las versiones en que PHP funciona como módulo de servidor, esto sucede únicamente cuando se inicia el servidor. En las versiones CGI y CLI, esto ocurre en cada ejecución.





1.4: Lenguajes de Programación: Web con PHP

Dependiendo de cómo se integre PHP con Apache, los cambios que realicemos en nuestro fichero de configuración se aplicarán en un momento o en otro.

Si como es nuestro caso, utilizamos **mod_php** para ejecutar PHP como un módulo de Apache, las opciones de configuración de PHP se aplicarán cada vez que se reinicie Apache.

Por tanto, no te olvides de hacerlo cada vez que hagas cambios en **php.ini**. Por ejemplo: **sudo /etc/init.d/apache2 restart** o también **sudo service apache2 restart**.

En la documentación de PHP se incluye una lista completa de las directivas que se pueden utilizar en **php.ini**.
<http://php.net/manual/es/ini.core.php>



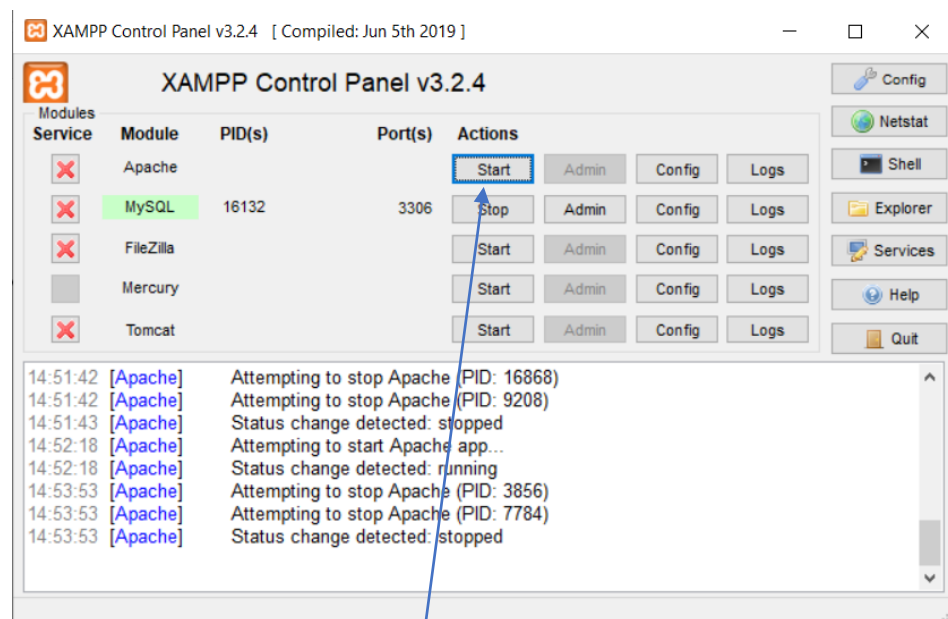
1.4: Lenguajes de Programación: Web con PHP

Algunas de las directivas más utilizadas que figuran en el fichero **php.ini** son:

- **short_open_tags**. Indica si se pueden utilizar en PHP los delimitadores cortos `<? y ?>`. Es preferible no usarlos, pues puede causarnos problemas si utilizamos páginas con XML. Para prohibir la utilización de estos delimitadores con PHP le asignamos a esta directiva el valor Off.
- **max_execution_time**. Permite que puedas ajustar el número máximo de segundos que podrá durar la ejecución de un script PHP. Evita que el servidor se bloquee si se produce algún error en un script.
- **error_reporting**. Indica qué tipo de errores se mostrarán en el caso de que se produzcan. Por ejemplo, si haces **error_reporting = E_ALL**, te mostrará todos los tipos de errores. Si no quieres que te muestre los avisos pero sí otros tipos de errores, puedes hacer **error_reporting = E_ALL & ~E_NOTICE**.
- **file_uploads**. Indica si se pueden o no subir ficheros al servidor por HTTP.
- **upload_max_filesize**. En caso de que se puedan subir ficheros por HTTP, puedes indicar el límite máximo permitido para el tamaño de cada archivo. Por ejemplo, **upload_max_filesize = 1M**.

1.4: Lenguajes de Programación: Web con PHP

Reiniciamos Apache usando el panel de control de XAMPP para que sus cambios surtan efecto.





1.4: Lenguajes de Programación: Web con PHP

Ejemplo #1 Nuestro primer script de PHP: *hola.php*

Creamos carpeta en XAMP . htdocs

FileZillaFTP

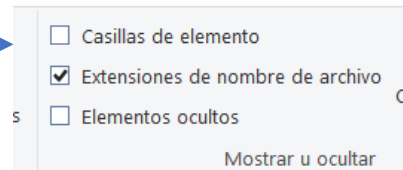
htdocs

img

index.php

Le ponemos el nombre que creamos un nuevo documento

la extensión



y le quitamos .txt

una vez creado le cambiamos

```
<?php
```

```
echo "<p>Hola Mundo</p>"
```

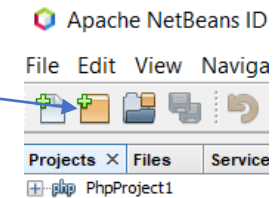
```
?>
```



1.4: Lenguajes de Programación: Web PHP con NetBeans Apache

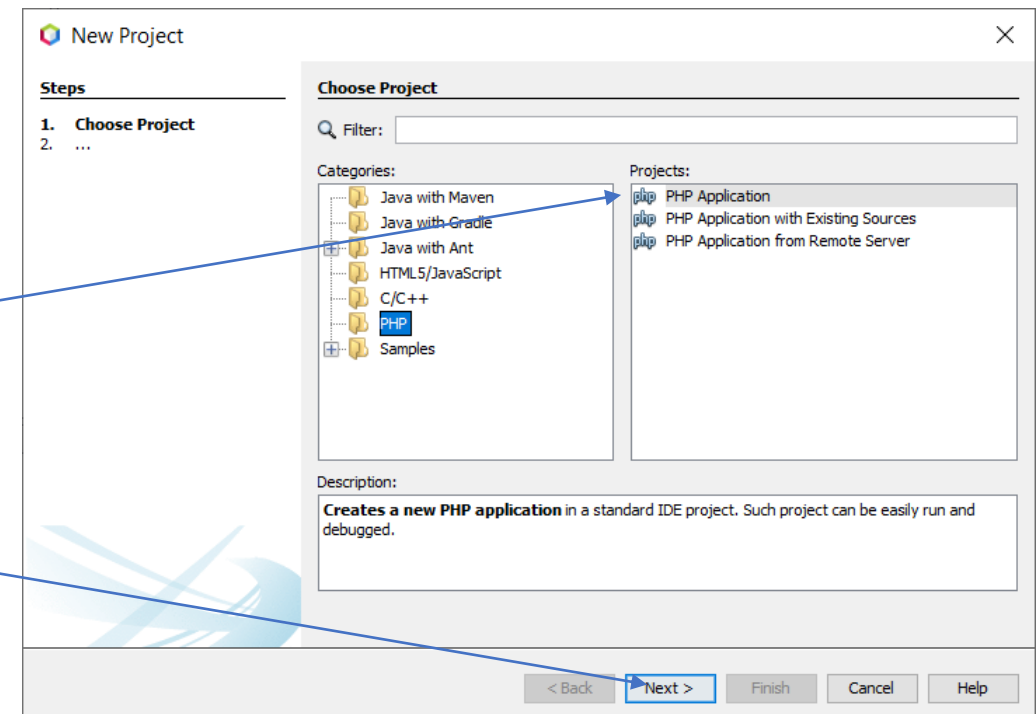
Procedemos a la descarga de NetBeans Apache <https://netbeans.apache.org/> y JDK <https://www.oracle.com/technetwork/es/java/javase/downloads/index.html>

Abrimos NetBeans Apache y creamos un nuevo proyecto (New Project).
utilizar botón derecho ratón (New Project)



También podemos

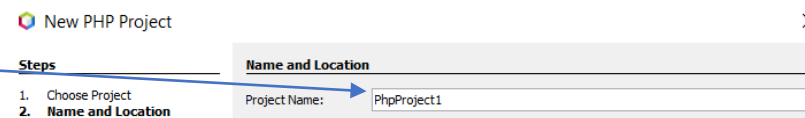
Seleccionamos carpeta PHP + PHP Application y siguiente



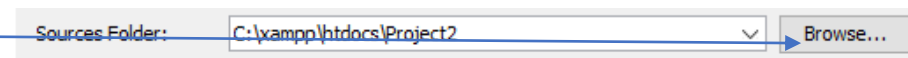


1.4: Lenguajes de Programación: Web PHP con NetBeans Apache

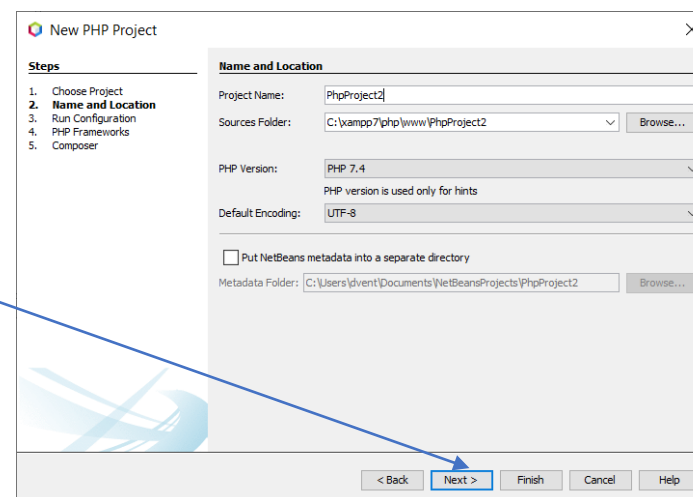
Ponemos el nombre del proyecto que queremos



Buscamos la carpeta donde guardaremos el proyecto



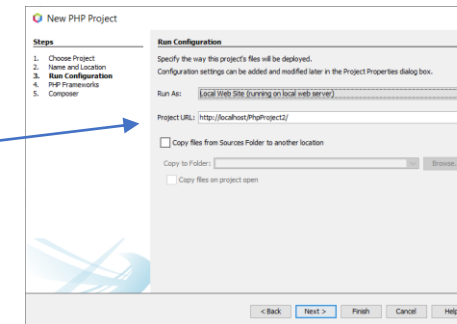
Damos siguiente (Next)



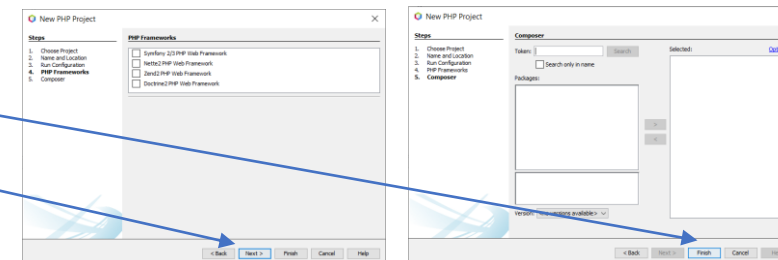
1.4: Lenguajes de Programación: Web PHP con NetBeans Apache

En esta parte debemos fijarnos en la parte de Project URL.
Tenemos que tener en cuenta el puerto de XAMPP y
Poner el mismo, en caso contrario no funcionará
`http://localhost/PhpProject2/`

localhost:80, localhost:8080.....



En estos dos pasos no hacemos nada Next y Finish





1.4: Lenguajes de Programación: Web PHP con NetBeans Apache

¿Cómo es la estructura básica?

Indicar que un archivo Php normalmente suele contener HTML y partes de código Php. Es por esto que antes de aventurarnos a de estudiar Php tengamos conocimientos de HTML.

En ocasiones oiremos decir que Php es un lenguaje “embebido” en HTML, es decir, incrustado en código HTML. Todos los fragmentos de código Php deben comenzar con la siguiente cadena de caracteres:

`<?php` ,y finalizar con `?>`

Los símbolos `<?php` y `?>` se dice que son los delimitadores del código Php. Normalmente, una página Php básica sería algo así:

```
*Sin título: Bloc de notas
Archivo Edición Formato Ver Ayuda
<!DOCTYPE html>
<html>
<head>
<title>Ejemplo de una web con PHP - aprenderaprogramar.com</title>
<meta charset="utf-8">
</head>
<body>
<?php
/* Aquí vendría el código PHP */
/* Este código PHP podría ser una consulta a base de datos */
/* Y además podríamos mostrar esos datos */
?>
</body>
</html>
```



1.4: Lenguajes de Programación: Web PHP con NetBeans Apache

En el anterior código podemos identificar los siguientes elementos:

- La primera línea es una etiqueta relativa a la versión/variante de HTML que declaramos usar. Esta primera etiqueta no es estrictamente necesaria. Sirve únicamente para indicar qué estándar de HTML es el que declaramos usar. `<!DOCTYPE html>`
- Las etiquetas de apertura y cierre de html que delimitan el código HTML. `<html> </html>`
- Las etiquetas de apertura y cierre head que delimitan la cabecera del código HTML. `<head> </head>`
- Las etiquetas body que delimitan el cuerpo del código HTML y que es donde suele ir el contenido central de un documento HTML (texto, imágenes, formularios, etc.). `<body> </body>`



1.4: Lenguajes de Programación: Web PHP con NetBeans Apache

- Dentro de las etiquetas body, además de código HTML, podemos tener código PHP embebido. En este caso hemos incluido únicamente un fragmento de código PHP, pero en el cuerpo del documento HTML podríamos incluir diversos fragmentos de código HTML mezclados con diversos fragmentos de código PHP. Cada fragmento de código PHP tiene que ir escrito con sus delimitadores/etiquetas.
- *Todo lo que salga de esas etiquetas será ignorado por el analizador*

```
<?php
```

```
/* Aquí vendría el código PHP */
```

```
/* Este código PHP podría ser una consulta a base de datos */
```

```
/* Y además podríamos mostrar esos datos */
```

```
echo "XXXXXXXXXX";
```

```
?>
```



1.4: Lenguajes de Programación: Web PHP con NetBeans Apache

PHP también permite la etiqueta de apertura abreviada `<?` (la cual está desaconsejada debido a que sólo está disponible si se habilita con la directiva `short_open_tag` del fichero de configuración `php.ini`, o si PHP se configuró con la opción `--enable-short-tags`).

Si un fichero **contiene solamente código de PHP**, se puede omitir la etiqueta de cierre de PHP al final del mismo.

```
<?php
/* Aquí vendría el código PHP */
/* Este código PHP podría ser una consulta a base de datos */
/* Y además podríamos mostrar esos datos */
echo "XXXXXXXXXX";
```



1.4: Lenguajes de Programación: Web PHP con NetBeans Apache

PHP: Mi primer programa PHP, crearemos el siguiente archivo mediante el uso de NetBeans Apache y XAMPP.

```
<html>
<head>
  <meta charset="UTF-8">
  <title></title>
</head>
<body>
  <?php

  echo 'Mi primer programa php';

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

Para identificar nuestra lengua
incorporaremos Lang=es
<html Lang=es>

Codificación de caracteres para Unicode



1.4: Lenguajes de Programación: PHP (STRING o CADENA)

PHP: Una cadena o string en PHP, es una serie seguida de caracteres, donde un carácter es igual a un byte. Sin embargo antes de que puedas usar tu primera cadena o string en PHP tienes que crearla primeramente. Hay distintas formas de hacer esto.

La forma más fácil de definir una cadena es cerrarla entre comillas dobles `$string = "valor"` , y todos los caracteres entre las comillas se percibirán como caracteres para el código PHP.

Sencillamente podemos decir que las strings o cadenas son variables que contienen texto. Por ejemplo, una cadena que contiene un nombre se define de la siguiente manera

```
$nombre = "Jhon";  
echo $nombre;
```

Sin embargo, también puedes usar comillas simples (`'`) y PHP también lo interpretará como una string o cadena de texto.

```
$nombre = "Jhon";  
echo $nombre;
```



1.4: Lenguajes de Programación: PHP (STRING o CADENA)

PHP: Debéis tomar en cuenta, que si escribís vuestro texto después del comando echo usando comillas dobles (" "), PHP interpretará lo que habéis escrito, como por ejemplo las variables, y si usas comillas simples, PHP mostrará en pantalla exactamente todo lo que habéis escrito. Aquí os muestro un ejemplo:

```
$nombre = "Alonso";  
echo "Tu nombre es $nombre" . "\n";  
//Esto te mostrará en pantalla la frase: Tu nombre es Jhon  
echo 'Tu nombre es $nombre';  
// Esto te mostrará en pantalla la frase: Tu nombre es $nombre
```

Tener en cuenta el ejemplo anterior a la hora de mostrar tu texto en la pantalla, recordar que las comillas dobles interpretarán lo que escribís, y que las simples van a mostrar en pantalla tal cual la oración que habéis escrito.

También podemos formatear cadenas fácilmente usando variables. Aquí hay un ejemplo:

```
$nombre = "Alonso";  
$introduccion = "Hello $nombre";  
echo $introduccion;
```




1.4: Lenguajes de Programación: PHP (STRING o CADENA)

PHP: Una cadena o string en PHP, es una serie seguida de caracteres, donde un carácter es igual a un byte. Sin embargo antes de que puedas usar tu primera cadena o string en PHP tienes que crearla primeramente. Hay distintas formas de hacer esto.

La forma más fácil de definir una cadena es cerrarla entre comillas dobles `$string = "valor"` , y todos los caracteres entre las comillas se percibirán como caracteres para el código PHP.

Sencillamente podemos decir que las strings o cadenas son variables que contienen texto. Por ejemplo, una cadena que contiene un nombre se define de la siguiente manera

```
$nombre = "Jhon";  
echo $nombre;
```

Sin embargo, también puedes usar comillas simples (`'`) y PHP también lo interpretará como una string o cadena de texto.

```
$nombre = "Jhon";  
echo $nombre;
```





1.4: Lenguajes de Programación: PHP (STRING o CADENA)

PHP: Es posible poder concatenar cadenas usando el operador de punto (.). Como ejemplo:

```
$primer_nombre = "John";  
$segundo_nombre = "Doe";  
$nombre = $primer_nombre . " " . $segundo_nombre;  
echo $nombre;
```

Si deseamos medir la longitud de una cadena, podemos utilizar la función strlen:

(PHP strlen(): Obtener la longitud de una cadena string)
\$int: Longitud del \$string pasado, devuelve 0 si el string esta vacio.
\$string: Cadena que queremos medir en longitud.

```
$string = "La longitud de esta cadena es de 47 caracteres.";  
echo strlen($string);
```



1.4: Lenguajes de Programación: PHP (STRING o CADENA)

PHP: Para contar el número de palabras contenidas en una cadena podemos usar la función `str_word_count()` de PHP que te contará automáticamente el número de palabras en una cadena. Aquí hay un ejemplo de esta función.

```
echo str_word_count ("Este es mi código PHP"); // Salida en pantalla: 5
```

Si queremos invertir una cadena podemos usar la función `strrev()`. He aquí un ejemplo:

```
echo strrev ("Este es mi código PHP"); // Salida en pantalla: PHP ogidoc im se etsE
```

En el caso que nosotros queramos buscar un texto específico dentro de una cadena, utilizaremos la función `strpos()` de PHP, que busca un texto específico dentro de una cadena.

En el caso de encontrar una coincidencia, la función devuelve la posición del carácter de la primera coincidencia. Si no se encuentra ninguna coincidencia queda en este caso en blanco

El siguiente ejemplo busca el texto “PHP” en la cadena “Este es mi código PHP”:

```
echo strpos ("Este es mi código PHP", "PHP"); // El resultado que saldrá en pantalla será 19
```



1.4: Lenguajes de Programación: PHP (STRING o CADENA)

Si lo deseamos también podemos **reemplazar el texto** dentro de una cadena la función **str_replace()** de PHP reemplaza algunos caracteres con algunos otros caracteres en una cadena. El siguiente ejemplo reemplaza el texto “código” por “lenguaje”:

```
echo str_replace ("código", "lenguaje", "Este es código PHP"); //Se mostrará en pantalla Este es lenguaje PHP
```



1.4: Lenguajes de Programación: PHP (Unir y dividir)

Podemos unir matrices para formar cadenas o dividir cadenas en matrices de cadenas.

Por ejemplo, para dividir una cadena con una lista de frutas separadas por una coma, usamos la función `explode`:

```
$frutas = "manzana,banana,naranja";  
$lista_de_frutas = explode(",", $frutas);  
echo "La segunda fruta en la lista es $lista_de_frutas[1]";
```

“La segunda fruta en la lista es banana”

Ahora, para unir una matriz a una sola cadena separada por comas, usamos la función `implode`:

```
$lista_de_frutas = ["manzana","banana","naranja"];  
$frutas = implode(",", $lista_de_frutas);  
echo "Las frutas son $frutas";
```

“Las frutas son manzana,banana,naranja”



1.4: Lenguajes de Programación: PHP (STRING o CADENA)

Ejercicio propuesto:

Separa la cadena que contiene la lista de números y crea una nueva matriz llamada lista_de_numeros.

38,42,58,48,33,59,87,17,20,8,98,14,62,66,14,62,97,66,74,78,66,2,79,29,72,6,3,71,46,68,48,4,12,52,66,48,14,
39,63,69,81,61,21,77



1.4: Lenguajes de Programación: PHP (STRING o CADENA)

Solución ejercicio propuesto

```
<?php
$numeros = "38,42,58,48,33,59,87,17,20,8,98,14,62,66,14,62,97,66,74,78,66,2,79,29,72,6,3,71,46,68,48,4,12,52,66,48,14,39,63,69,81,61,21,77";
$lista_de_números= explode(",", $numeros);

// TODO: dividir la variable $numeros en un array
// Llamar a $lista_de_numeros

print_r ($lista_de_numeros);
?>
```

```
[0] => 38 [1] => 42 [2] => 58 [3] => 48 [4] => 33 [5] => 59 [6] => 87 [7] => 17 [8] => 20 [9] => 8 [10] => 98 [11] => 14 [12] => 62 [13] => 66 [14] => 14
[15] => 62 [16] => 97 [17] => 66 [18] => 74 [19] => 78 [20] => 66 [21] => 2 [22] => 79 [23] => 29 [24] => 72 [25] => 6 [26] => 3 [27] => 71 [28] => 46
[29] => 68 [30] => 48 [31] => 4 [32] => 12 [33] => 52 [34] => 66 [35] => 48 [36] => 14 [37] => 39 [38] => 63 [39] => 69 [40] => 81 [41] => 61 [42] =>
21 [43] => 77
```



1.4: Lenguajes de Programación: PHP (STRING o CADENA)

Ejercicio propuesto:

Verifica si en nuestra cadena existe la palabra SEGUNDO en la frase “CES JUAN PABLO”

```
//Determinar si existe o no una palabra en un string
$existe_cadena = FALSE;
$posicion_cadena = strpos("CES JUAN PABLO", 'SEGUNDO');
if($posicion_cadena !== FALSE){$existe_cadena = TRUE;}
echo "CES JUAN PABLO ";
echo $existe_cadena ? 'true' : 'false';
echo "<br/>";
```

Con este pequeño código `$dia_ingles = date('D');` tendremos el día en la variable `$dia_ingles`.

Debemos de mostrar el día en español por pantalla, con el siguiente mensaje: "El día de la semana es: XXXX".

Recordemos, para introducir a las sentencias **if y else**, debemos ser conocedores que **if** es una sentencia condicional, en la cual si una condición resulta ser verdadera, se ejecutara el código escrito dentro del **if**. El uso de **else** por otro lado es opcional, y se utiliza para ejecutar un código específico en caso de la una condición sea falsa y no se ejecute el **if**.





Bibliografía

- Vicente, J. (2013) “Desarrollo Web en entorno servidor”, editorial: Ibergaceta publicaciones, S.L. ISBN:978-84-1545-292-8
- Comesaña, L. (2017) “Desarrollo Web en entorno servidor”