



# Desarrollo de Aplicaciones con Tecnologías Web

Código: IFCD0210

Familia profesional: Informática y Comunicaciones

Área profesional: Desarrollo

Nivel de cualificación profesional: 3

# Relación de módulos formativos y de unidades formativas:

**MF0491\_3:** Programación web en el entorno cliente.(180 horas)

- **UF1841:** Elaboración de documentos web mediante lenguajes de marcas. (60 horas)
- **UF1842:** Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión. (90 horas)
- **UF1843:** Aplicación de técnicas de usabilidad y accesibilidad en el entorno cliente (30 horas)


**MF0492\_3:** Programación web en el entorno servidor. (240 horas)

- **UF1844: Desarrollo de aplicaciones web en el entorno servidor. (90 horas)**
- **UF1845:** Acceso a datos en aplicaciones web del entorno servidor. (90 horas)
- **UF1846:** Desarrollo de aplicaciones web distribuidas. (60 horas)

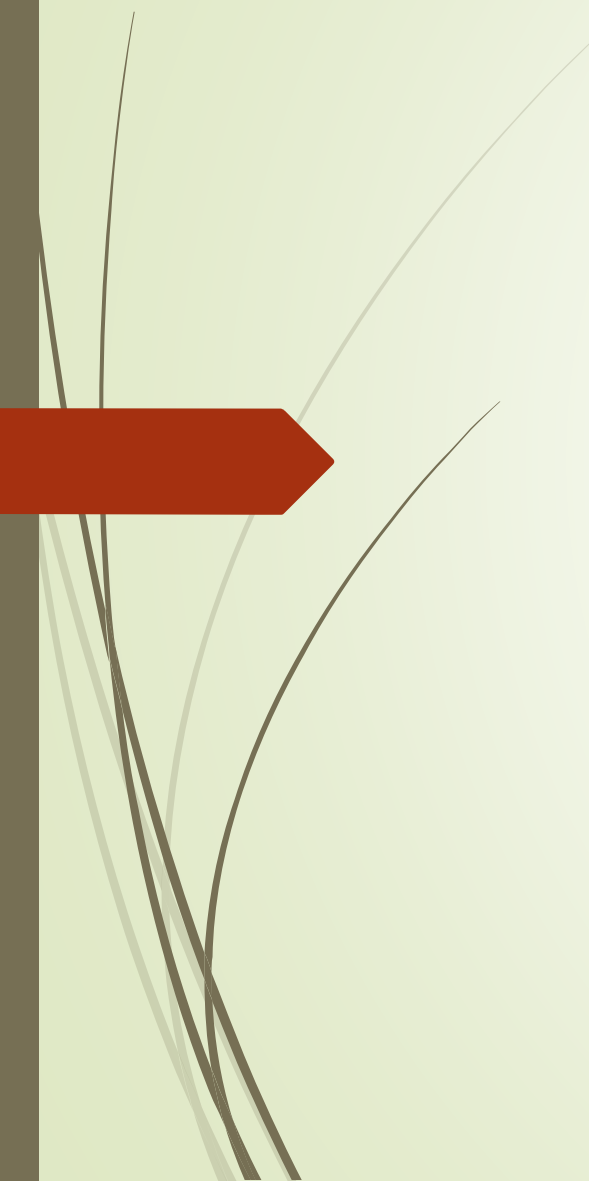
**MF0493\_3:** Implantación de aplicaciones web en entornos internet, intranet y extranet. (90 horas)

**MP0391:** Módulo de prácticas profesionales no laborales de desarrollo de aplicaciones con tecnología web. (80 horas)

# UF1844: Desarrollo de aplicaciones web en el entorno servidor.

- 
1. *Introducción al lenguaje PHP y Bases de Datos*
  2. *Mi primer programa en PHP.*
  3. *Qué es una variable: Concepto y características.*
  4. *Tipos de datos.*
  5. *Cálculos en PHP: expresiones aritméticas y operadores..*
  6. *Condicionales: IF.*
  7. *Condicionales: ELSE y ELSEIF.*
  8. *Trabajando con múltiples opciones; Switch.*
  9. *Bucles: dando vueltas...*
  10. *Arrays.*
  11. *Introducción a las funciones en PHP*
  12. *Funciones que trabajan con variables.*
  13. *Funciones que trabajan con cadenas*
  14. *Funciones que trabajan con arrays*
  15. *Funciones que trabajan con fechas*
  16. *Funciones definidas por el usuario*

# UF1844: Desarrollo de aplicaciones web en el entorno servidor.

- 
- 17.** *Enviando información con PHP*
  - 18.** *Acceso a archivos*
  - 19.** *Cookies*
  - 20.** *Sesiones*
  - 21.** *Bases de Datos MySQLi y PHP*
  - 22.** *Conexión con la Base de Datos*
  - 23.** *Insertar registros en una base de datos desde la web*
  - 24.** *Consultas a una base de datos MySQLi*
  - 25.** *Actualizar tablas*
  - 26.** *Eliminar registros de una tabla.*
  - 27.** *Trabajando con varias tablas.*

4.- Lenguajes de programación de aplicaciones web en el lado servidor

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 1.- Introducción al lenguaje PHP y Bases de Datos (I)

Software necesario:

1. Editor de código:
  - Microsoft Visual Studio

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 1.- Introducción al lenguaje PHP y Bases de Datos (II)

PHP **no se puede ejecutar** en tu ordenador directamente en el navegador con haces con HTML o CSS

PHP se ejecuta en un **servidor web**

Soluciones:

- Publicar la web en un **servidor web real**
- Instalar el servidor en nuestro ordenador y **trabajar de forma local**

**Ejemplos:** Xampp, Wamp Server, Lamp...



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 1.- Introducción al lenguaje PHP y Bases de Datos (III)

#### WAMP / XAMPP

- Servidor independiente de código libre
- Gratuito
- Instala un servidor apache en nuestro ordenador de forma muy sencilla
- Windows, Mac, Linux

Wamp → <http://www.wampserver.com/en/#download-wrapper>

Xampp → [www.apachefriends.org/es/](http://www.apachefriends.org/es/)

Recomendación: debemos tener instalado cualquier navegador para ver nuestras aplicaciones. No ocurre como con html y css que debemos comprobar en varios navegadores si la ejecución es la que esperamos.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 2.- Mi primer programa en PHP (I).

#### Antes de empezar:

- la función básica de PHP es **producir código HTML**
- **Html** solo nos permite insertar elementos **estáticos**
- **PHP** nos permite analizar diferentes situaciones y actuar en función de los **resultados**. Esto nos permite crear página **dinámicas**.

Ej: catalogos, busquedas, tiendas online, etc...

#### Estructura de código PHP:

```
<?php  
... código php ...  
?>
```

```
<?php ... ?>  
<? ... ?>  
<script lenguaje = "php"> ... </script>
```



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 2.- *Mi primer programa en PHP (II).*

- Todo lo que está contenido entre las etiquetas de apertura y cierre, debe coincidir con las reglas de sintaxis de PHP....
- ... no se envía al navegador hasta que no ha sido procesado.
- El navegador nunca va a procesar código PHP.
- El servidor es el que lo procesa y entrega al navegador el resultado de su trabajo.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 2.- *Mi primer programa en PHP (III).*

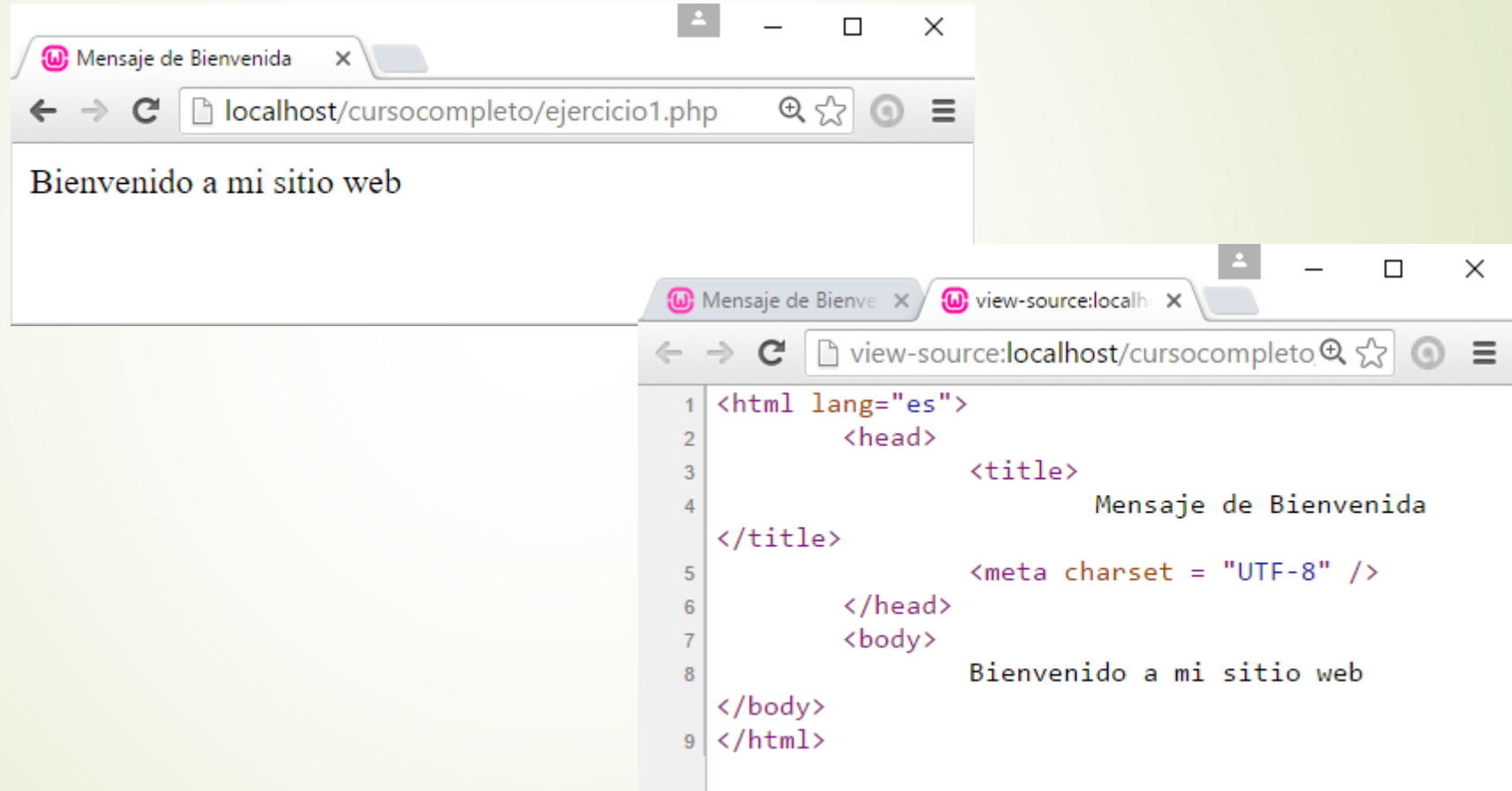
```
<html lang="es">
  <head>
    <title>
      <?php echo "Mensaje de Bienvenida"; ?>
    </title>
    <meta charset = "UTF-8" />
  </head>
  <body>
    <?php
      echo "Bienvenido a mi sitio web";
    ?>
  </body>
</html>
```

Opciones: `echo 'Mensaje de Bienvenida';`  
`echo ("Mensaje de Bienvenida");`

4.- Lenguajes de programación de aplicaciones web en el lado servidor

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 2.- *Mi primer programa en PHP (IV).*



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 2.- *Mi primer programa en PHP (V).*

Los caracteres de escape ( \n y \t )

```
<?php
echo "primera linea\n";
echo "segunda linea<br />";
echo "tercera linea";
?>
```

Se verá de la siguiente forma en el código HTML:

```
primera linea
segunda linea<br>tercera linea
```

Mientras que en el navegador se verá así:

```
primera linea segunda linea
tercera línea
```

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 2.- *Mi primer programa en PHP (VI).*

#### Comentarios en PHP: código claro

```
<?php  
    // Esto es un comentario  
?>
```

*Si deseamos recurrir a los comentarios de varias líneas, la notación sería la siguiente, similar a Java y C:*

```
<? php  
    /*  
    Esto es un comentario  
    multilínea  
    usando la misma sintaxis  
    utilizada en Java y C  
    */  
?>
```

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 3.- *Qué es una variable: Concepto y características (I).*

**Variable:** especie de **contenedor en el que se almacena el valor que nos interesa**, y que puede cambiar cuando sea necesario.

#### **Nombres de variables:**

- debe empezar siempre por una letra o un guion bajo (\_)
- empiezan por el símbolo \$
- pueden contener letras números y \_
- es sensible a mayúsculas y minúsculas

Ej: \$a = 20;

PHP es mucho más flexible que otros lenguajes de programación, y es que ***no se requiere que las variables se declaren antes de su uso.***



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 3.- Qué es una variable: Concepto y características (II).

**Variables dinámicas:** son nombres de variables que se pueden establecer y usar de forma variable.

```
<?php
    $a = "hola";
    $$a = "mundo";
?>
```

Se han definido y almacenado dos variables en el árbol de símbolos de PHP: \$a, que contiene "hola", y \$hola, que contiene "mundo".

```
<?php
    echo "$a ${$a}";
?>
```

produce el mismo resultado que:

```
<?php
    echo "$a $hola";
?>
```

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 4.- Tipos de datos (I).

- Una variable puede contener **diferentes tipos de valores**, cada uno de los cuales tiene un comportamiento y utilidad diferente.
- En PHP **no es necesario definir el tipo** de variable antes de utilizarla
- Las variables **se declaran y definen de forma automática** cuando les asignamos un valor

Los **tipos de variables** mas importantes:

- **Booleanos**
- **Números (enteros y decimales)**
- **Matrices (arrays)**
- **Cadenas**

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 4.- Tipos de datos (II).

#### **Booleanos: Si o No**

Los tipos de datos booleanos se utilizan para indicar valores que solo admiten dos estados

**:verdadero (true)**

**:falso (false)**

Ej. Si queremos saber si un alumno esta dado de alta en un curso:

***\$alta = true;***

Ej. Queremos saber si un alumno ha pagado o no su matricula:

***\$pagado = false;***

Hay muchas formas de utilizarlas y de controlar su estado, se verá a lo largo del curso en multiples ejemplos

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 4.- Tipos de datos (III).

#### **Números:** Enteros y Decimales

Otro tipo de datos es el **número entero, positivo o negativo**, cuyo valor del máximo (absoluto) puede variar dependiendo del sistema operativo que se ejecuta en PHP, pero que por lo general es, aproximadamente 2 mil millones.

También podemos usar **números decimales** (a veces nos referiremos a ellos como "doble" o "real"). Para separar los decimales no usaremos la coma, si no el punto.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

#### Tipos de datos

##### Ejercicio4-3

Define una variable de los siguientes tipos: integer, double, string y boolean.

A continuación, imprímelas en la página, una por línea.

##### Ejercicio4-3-2

En primer lugar definiremos tres variables con números enteros.

A continuación, crearemos una cadena en la que incorporaremos estas tres variables, por ejemplo, ("Las notas del examen han sido...").



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 4.- Tipos de datos (IV).

#### **Matrices:** Qué son?

Podemos considerar una matriz como una variable compleja, que **contiene una serie de valores en lugar de uno solo**, cada uno de los cuales se caracteriza por **una clave, o índice que lo identifica de manera única**.

Por ejemplo, podemos definir una matriz llamada \$color donde almacenaremos los colores que deseemos:



Definición:

```
$color = array ('blanco','rojo','azul','verde','amarillo');
```

Llamada:

```
echo $color[1]; // devolverá rojo ya que los arrays se enumeran desde 0
```



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 4.- Tipos de datos (V).

#### **Cadenas:** Qué son?

Una cadena es **cualquier conjunto de caracteres**, sin limitación normalmente, **contenida dentro de un par de comillas** dobles o simples. Las cadenas delimitadas por comillas simples son la forma más simple.

Definición:

```
$cadena1 = "Hola";
```

```
$cadena2 = " Mundo";
```

Llamada:

```
echo $cadena1.$cadena2;
```

```
echo $cadena1." ".$cadena2;
```

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

#### Variables

#### Ejercicio4-5

*Vamos a definir dos variables, una con un nombre y otra con una edad y los insertaremos en la frase "Mi nombre es \_\_\_\_ y tengo \_\_\_\_ años"*

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 5.- Cálculos en PHP: expresiones aritméticas y operadores..

**Los Operadores** nos permiten *realizar operaciones aritméticas* tradicionales, y además, **manipular los contenidos de nuestras variables**.

#### Tipos de Operadores

- de asignación
- de concatenación
- aritméticos
- de incremento/decremento
- de comparación
- Lógicos

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 5.- Cálculos en PHP: expresiones aritméticas y operadores..

#### Operador de asignación (=)

Un ejemplo de operador de asignación lo tendríamos en la expresión

```
$nombre = "Juan";
```

El símbolo = se utiliza para asignar a la variable \$nombre el valor 'Juan'.

En general, podemos decir que con el **operador de asignación** tomamos lo que está a la derecha del signo de igual y le asignaremos el valor que lo sigue.

- =
- +=
- -=
- \*=
- /=
- %=
- .=

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 5.- Cálculos en PHP: expresiones aritméticas y operadores..

#### Operador de concatenación (.)

Las cadenas son una sucesión de caracteres: números, letras y símbolos.

Las cadenas van entre comillas dobles o simples, de ello depende el modo de trabajar con las variables.

**`$precio=150;`**

**`$oferta="El precio es $precio"; -> El precio es 150`**

**`$oferta='El precio es $precio'; -> El precio es $precio`**

Para concatenar usamos el punto (.)

**`$cadena1='esto es cadena 1';`**

**`$cadena2=' esto es cadena 2';`**

**`$cadena=$cadena1.$cadena2; -> esto es cadena 1 esto es cadena 2`**

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 5.- Cálculos en PHP: expresiones aritméticas y operadores..

#### Operadores aritméticos (+, -, \*, / , %)

Permiten realizar operaciones aritméticas con los datos: **suma, resta, división, multiplicación o módulo.**

```
<?php
    $a = 5;
    $b = 9;

    echo "Suma: ";
    echo $a + $b;

    echo "<br/>Resta: ";
    echo $a - $b;

    echo "<br/>Multiplicación: ";
    echo $a * $b;

    echo "<br/>División: ";
    echo $a / $b;

    echo "<br/>Módulo: ";
    echo $a % $b;

?>
```



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 5.- Cálculos en PHP: expresiones aritméticas y operadores..

#### Operadores aritméticos (+, -, \*, / , %)

Prioridad de operadores: operadores del mismo nivel, se hace de izda a dcha. Multiplicación y división tiene prioridad sobre suma y resta. Se cambia con ().

```
<?php
```

```
$a = 4;
```

```
$b = 5;
```

```
$c = 3;
```

```
echo $a-$b*$c;
```

```
echo '<br/>';
```

```
echo ($a-$b)*$c;
```

```
?>
```

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

### Variables y cálculos

#### Ejercicio5-4

*Crear una aplicación que calcule el 25% de 325 y lo muestre en pantalla*

#### Ejercicio5-4-2

*Crear una aplicación que calcule el IVA del 21% de un producto que vale 43.90 y lo mostraremos por pantalla.*

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 5.- Cálculos en PHP: expresiones aritméticas y operadores..

#### Operadores de incremento y decremento (++ , --)

Si necesitas aumentar o disminuir una variable en una unidad, una buena solución es usar los operadores de incremento y decremento.

*Incrementar ++*

`++$a`    `$a++`

*Decrementar --*

`--$a`    `$a--`

Dependiendo de dónde se coloque puede variar el resultado de las operación.

```
<?php
    $a = 15;
    $b = 15;

    echo ++$a;
    echo '<br/>';
    echo $b++;

?>
```

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

### Operaciones aritméticas

#### Ejercicio5-5

*Vamos a definir dos variables numéricas enteras, \$a y \$b. Asígnales el valor que quieras.*

*Realiza los cálculos y muestra en pantalla su suma, su resta, su multiplicación y su división.*

*A continuación, incrementa las dos variables en una unidad y muestra en pantalla el resultado.*

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 6.- Condicionales: IF (I)

#### Qué son y para que sirven??

Las estructuras de "control de flujo" nos presentan un aspecto fundamental de la programación: **la capacidad de realizar diferentes operaciones, y si lo deseamos, ejecutarlas varias veces, sobre la base de la evaluación de determinadas condiciones.**

La instrucción **if** nos **permite realizar o no** ciertas **porciones de código**, dependiendo de que se cumpla o no una condición.

***if (condición) instruccion***

Después del **if** se indica en la **expresión** entre paréntesis que **debe ser evaluada** (condición), en el **sentido booleano**, es decir, su valor se considera **verdadero o falso**.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 6.- Condicionales: IF (II)

Si la condición deriva en varias instrucciones, estas deben ir entre llaves {instrucciones}

```
if ($nombre == 'Juan') {  
    echo "Hola Juan!<br>";  
    echo "¿Quieres ver tus nuevas fotos?<br />";  
}  
echo "Hola a todo el mundo!"; // esta instrucción sale siempre
```

Cuidado con lo siguiente:

```
$a = 7;  
if ($a = 4) echo '$a es igual a 4!'; // también probar if ($a = 0)  
$a = 7;  
if ($a == 4) echo '$a es igual a 4!';
```



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

#### Condicionales: if

##### Ejercicio6-1

*Una variable llamada \$precio contendrá el precio de un artículo.*

*Crea una estructura condicional para que en caso de que el precio sea mayor o igual a 100, aparezca en pantalla el rótulo "Este artículo es muy caro!"*

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 6.- Condicionales: IF (III)

**Operadores de comparación: (<, >, ==, ===, <=, >=, !=)**

!=	diferente
===	idénticos
>	mayor
> =	mayor o igual
<	menor
<=	menor o igual

La comparación con idéntico (===) debe coincidir el valor y el tipo de dato para que cumpla la condición de verdadero.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 6.- Condicionales: IF (IV)

#### Operadores de comparación: (<, >, ==, ===, <=, >=, !=)

En la comparación con cadenas, debemos tener en cuenta que se consideran menores los que vienen primero en orden alfabético (“a” es la mas pequeña y “z” la mayor) .

Las minúsculas son “mayores” que las mayúsculas y todas las letras son “mayores” que los números.



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

### Operadores de comparació

#### Ejercicio6-2

*Vamos a crear dos variables numéricas, \$a y \$b, con las cifras que desees. A continuación, realiza las siguientes tareas:*

- *Haz que ambas variables aparezcan en pantalla con sus correspondientes valores*
- *Si \$a es mayor que \$b, haremos que aparezca en pantalla el mensaje "A es mayor que B"*
- *Si \$b es mayor que \$a, el mensaje que aparecerá es "B es mayor que A".*
- *Si ambas son iguales, se mostrará el mensaje "A y B son iguales".*

#### Ejercicio6-2-2

*Una variable guardará un valor numérico entero. El programa deberá mostrar un mensaje si el número es par y otro si es impar.*

*También debes prever el caso de que el número no sea un entero positivo.*

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 6.- Condicionales: IF (V)

#### Operadores lógicos: ( AND &&, OR ||, XOR )

Nos permiten **trabajar con varias comparaciones, combinando los posibles valores** que pudiesen tener estas.

OR o	\$A > \$B    \$A > \$C    \$C <= \$D	Sólo debe cumplirse una
AND &&	\$A == \$B AND \$C > \$A AND \$B > \$E	Deben cumplirse todas
XOR	\$A > \$B XOR \$A == \$C	Si uno es verdadero el otro debe ser falso

```
<?php
```

```
$a = 1;
```

```
$b = 8;
```

```
$c = 3;
```

```
$d = 1;
```

```
if ($a == $d and $b > $c and $c > $d) { echo "Es Verdadero!!"; }
```

```
if ($a == $d xor $d > $c) { echo "Es Verdadero!!"; }
```

```
?>
```



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 7.- Condicionales: ELSE y ELSEIF (I).

**ELSE:** establece que instrucciones se ejecutarán en el caso de que NO se cumpla la condición establecida por el If.

```
if ($a > $b) { echo 'a es mayor que b'; }  
else { echo 'a NO es mayor que b'; }
```

Debe colocarse inmediatamente después de la llave de cierre del “true”.

```
<?php  
    $nota = 6;  
    if ($nota <5) {  
        echo "estás suspendido";  
    } else {  
        echo "estás APROBADO!!";  
    }  
?>
```



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 7.- Condicionales: ELSE y ELSEIF (II).

Las instrucciones if pueden ser **anidadas una dentro de otras**, lo que nos permite crear código de una cierta complejidad. Ejemplo:

```
if ($nombre == 'Juan') {  
    if ($apellido == 'Lopez') {  
        print "Juan Lopez, bienvenido a nuestra página";  
    } else {  
        print "Tenemos un nuevo Juan entre nosotros!";  
    }  
} else {  
    print "Hola $nombre!";  
}
```

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

#### If-Else

##### Ejercicio7-1

La variable \$num contendrá un valor entero entre 1 y 3. Deberemos crear una estructura condicional que haga aparecer en pantalla el nombre del número con letras (por ejemplo, si el número es 3, deberá verse "tres").

##### Ejercicio7-2

Partimos de tres variables numéricas enteras, a las que llamaremos \$a, \$b y \$c. Debemos elaborar un programa que indique cual es el mayor y cual el menor.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.



### If-Else

#### Ejercicio7-3

Para realizar este ejercicio, en primer lugar deberemos conocer una nueva función matemática que no hemos tratado en el tema.

La función `rand()` genera un número aleatorio. Admite dos parámetros, que serán el número inicial y final entre los que queremos ese número al azar.

Usando esta función, vamos a crear un número entre 1 y 100, es decir, `rand(1,100)`; Crearemos una estructura condicional que nos diga cuantos dígitos tiene ese número, uno, dos o tres.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 7.- Condicionales: ELSE y ELSEIF (III).

**Elseif:** podemos indicar una segunda condición que **se evaluará** solo en el caso en que la **anterior sea falsa**.

**Elseif** se ejecutará...

- Cuando no haya alguna **respuesta verdadera** en el **if**...
- ...o en un posible **elseif anterior**
- Si llegamos al final de los **elseif**, y **ninguno ha sido verdadero**, se ejecutará lo que está dentro del **else** final.



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 7.- Condicionales: ELSE y ELSEIF (IV).

```
if ($nombre == 'Juan') {  
    print "Bienvenido Juan!";  
} elseif ($apellido == 'Lopez') {  
    print "Bienvenido, señor Lopez";  
} else {  
    print "Hola $nombre!";  
}
```

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

#### If-Else-Elseif

#### Ejercicio7-4

Desarrollar una aplicación que muestre el resultado de la nota obtenida por un alumno en función de la siguiente tabla:

- < 5 Suspendido
- >= 5 Aprobado
- >=6 Bien
- >=7 Notable
- >=9 Sobresaliente
- ==10 Matricula de Honor



## 8.- Trabajando con múltiples opciones; Switch.

If, Elself y Else se vuelve muy confuso cuando tenemos múltiples opciones. Por eso tenemos esta instrucción que simplifica el proceso de elección, cuando tenemos múltiples opciones: ***nos permite evaluar diversos valores y ejecutar un código determinado en cada caso.***

```
switch (<condición>) {  
    case <valor 1>:  
        <código>  
        break;  
    case <valor 2>:  
        <código>  
        break;  
    ....  
    default:  
        <código>;  
        break;  
}
```

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.



### Switch

#### Ejercicio8-1

Utilizando switch, crearemos un programa que en base al valor de una variable numérica que puede estar entre 1 y 7, muestre el nombre del día de la semana asociado. Por ejemplo 1 -> Domingo, 2 -> Lunes.

#### Ejercicio8-1-2

La variable \$color guardará uno de los colores del arco iris, a tu elección (son rojo, naranja, amarillo, verde, azul, añil y violeta)

En función del color que se haya guardado en la variable, el programa mostrará un texto representativo (no importa su contenido, solo es para practicar)

## 8.2- Operadores ternarios.

Se componen de ***tres expresiones***: el valor devuelto son la segunda o tercera de estas expresiones, dependiendo de si la primera es verdadera o falsa. Se puede considerar una manera muy concisa de una sentencia if.

***\$tipo = (\$altura >= 200) ? 'alto' : 'normal';***

Condición

Si se cumple...

Si no se cumple...

***if (\$altura >= 200) {  
    \$tipo = 'alto';  
} else {  
    \$tipo = 'normal';  
}***

## 9.- Bucles: dando vueltas...

Nos permite realizar ciertas tareas de **forma repetitiva**



Existen 4 tipos en Php:

For, While, Do.... While y Foreach

## 9.- Bucles: FOR

Permite **realizar varias veces** una misma tarea.

```
for (expr1; expr2; expr3) {  
    sentencia;  
}
```

- La primera expresión (**expr1**) es evaluada (ejecutada) una vez incondicionalmente al comienzo del bucle.
- En el **comienzo de cada iteración**, se evalúa **expr2**. Si se evalúa como TRUE, el bucle continúa y se ejecutan la/s sentencia/s anidada/s. Si se evalúa como FALSE, finaliza la ejecución del bucle.
- Al **final de cada iteración**, se evalúa (ejecuta) **expr3**.

Cuidado con los bucles sin fin



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

#### FOR

#### Ejercicio9-2

Crear una tabla de conversión de divisas, de euros a pesetas. En concreto, la tabla mostrará la conversión desde 1 hasta 10 euros.

Recuerda que la equivalencia Euro pesetas era:  $1\text{€} = 166.386 \text{ pts.}$

#### Ejercicio9-2-2

Vamos a crear un programa que construya una estructura en HTML para diseñar una tabla con 20 filas y 10 columnas.

Para realizar esta tarea recurriremos a dos bucles for. Dentro de cada una de las celdas de la tabla puedes colocar cualquier letra, para que sea más visible.



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 9.- Bucles: WHILE

Su sintaxis requiere la palabra clave **while** y **entre paréntesis**, la **condición** a ser evaluada, y **dentro de las llaves**, el **código debe ser repetido hasta que se cumpla esta condición**.

```
while (expr) {  
    sentencia;  
}
```

**No proporciona instrucciones para inicializar y incrementar el contador:** tenemos que poner estas declaraciones en el flujo general del código,

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 9.- Bucles: Do..... WHILE

Es similar al **while** , la condición es evaluada tras finalizar el bloque repetitivo, lo que implica que se ejecute, al menos, una vez.

```
Do {  
    sentencia;  
} while (expr)
```

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

#### For, While, Do...while

#### Ejercicio9-6

Crear tres programas distintos que realicen la misma tarea, para practicar un poco: uno con for, otro con while y el tercero con do/while.

El programa en cuestión debe ofrecernos la tabla de multiplicar del 2.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 9.- Bucles: Break y Continue

En determinados casos podría interesarnos **salir del bucle** aunque la condición todavía **no esté verificada**.

- **Break sirve para salir del bucle**
- **Continue impide que se ejecuten algunas instrucciones** de un bucle sin interrumpir la ejecución de las iteraciones siguientes.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 10.- Arrays (I).

Un **array** es una **lista de valores**.

Cada elemento de esa lista se identifica **con un nombre**, que será común para todos ellos y un **índice** que será el elemento que los diferencia.

Forma de declararlos:

```
$color = array('verde','rojo','amarillo','gris');
```

Esto es un array escalar, cada elemento tiene un valor y un índice numérico que lo identifica (empezando por 0):

**0 → verde**

**1 → rojo**

**2 → amarillo**

**3 → gris**

Para hacer referencia a los elementos de un array:

**\$color[2] → ej: echo \$color[2];** mostraría **amarillo**

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 10.- Arrays (II).

Agregar nuevos valores:

```
$color[ ] = 'naranja';
```

asigna el valor numérico de **forma automática**, el siguiente libre (4 en nuestro caso). Si lo deseamos asignar de forma manual:

```
$color[7] = 'purpura';
```

si asignamos valores a un array en índices NO consecutivos, NO existen.



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

#### Arrays

##### Ejercicio10-1

Definir un array que contenga los nombres de los días de la semana. Mostrar en pantalla el nombre del primer y último elemento.

##### Ejercicio10-2

Crear un array compuesto por treinta números diferentes, que serán las ventas diarias de nuestra tienda. A partir de estos datos, calcular el promedio de ventas y mostrarlo por pantalla. Puedes usar un bucle for para ayudarte en la suma.

##### Ejercicio10-3

Partir de un array compuesto por 10 elementos numéricos enteros. Construir un programa que encuentre el más alto y el más bajo de ellos y los muestre por pantalla.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 10.- Arrays (III) Asociativos.

Este tipo de array admite como **índice** tanto **números** como **nombres**, en cuyo caso tendremos que indicarlo nosotros.

**`$a['índice']=valor;`**

El índice también puede ser una variable creada previamente.

**`$a[$índice]=valor;`**

Se pueden combinar índices numéricos con índices alfabéticos en el mismo array.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

### Arrays asociativos

#### Ejercicio10-4

Vamos a crear un array asociativo en el que se guarden las contraseñas de cinco usuarios de un sistema.

A cada uno de los usuarios se accederá por su nombre.

Deberemos hacer que se muestre en pantalla la contraseña de uno de esos usuarios.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 10.- Arrays (IV) Bidimensionales y Multidimensionales.

Una matriz multi-dimensional es ***una matriz en la que uno o más elementos de las matrices es a su vez otra matriz.***

Ej: Suponga que desea recopilar datos personales de más de una persona, y que por cada persona registrará su nombre, fecha de nacimiento y ciudad de residencia.

```
$persona = array(
    array('nombre' => 'Juan', 'apellido' => 'Lopez', 'fecha_nacimiento' => '1973/06/15',
        'residencia' => 'Madrid'),
    array('nombre' => 'Maria', 'apellido' => 'Flores', 'fecha_nacimiento' => '1968/04/05',
        'residencia' => 'Sevilla'),
    array('nombre' => 'Ana', 'apellido' => 'Garrido', 'fecha_nacimiento' => '1964/11/26',
        'residencia' => 'Barcelona'));

echo $persona[0]['apellido']; // imprime 'Lopez'
echo $persona[1]['residencia']; // imprime 'Sevilla'
echo $persona[2]['nombre']; // imprime 'Ana'
```

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 10.- Recorrer Arrays (V): Foreach.

Recorre los diferentes tipos de arrays de una manera controlada.

Dos sintaxis diferentes:

- Para **índices** numéricos

El bucle asigna a la variable \$valor el valor del elemento actual y en la siguiente iteración, devuelve el siguiente valor. **ejemplo36**

```
foreach (expresion as $valor)  
{  
  //acciones  
}
```

- Para **arrays asociativos**

Igual que antes pero además se asocia cada clave del array a la variable \$clave. **ejemplo36-1**

```
foreach (expresion as $clave=>$valor)  
{  
  //acciones  
}
```



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

### Arrays y Foreach

#### Ejercicio10-5

Crear un array con 10 notas, recorrerlo y calcular su promedio. En función de la nota, sacar en pantalla si está suspenso, aprobado, notable o sobresaliente.

#### Ejercicio10-6

Crear una agenda de contactos en un array multidimensional y recorrerlo para mostrar su contenido con Foreach dentro de una tabla, con un encabezado (Nombre-Teléfono-Tipo de contacto).



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 11.- Introducción a las funciones en PHP (I)

Una función es un **conjunto de instrucciones que tienen un propósito**, llevar a cabo ciertas operaciones. Las funciones nos **evitan repeticiones de código**: sólo necesitaremos llamar a la función apropiada, y proporcionarle los parámetros que necesita para su ejecución.

Las funciones pueden estar **integradas en el lenguaje o ser definidas por el usuario**. En ambos casos, el modo de recuperación es el mismo.

La **sintaxis básica** con la que se llama a una función es muy simple:

***funcion();***

Si la función devuelve un valor, podemos indicar la variable en la que lo almacenaremos:

***\$valor = funcion();***

También podemos utilizarla dentro de una expresión:

***\$ejemplo = (10 \* funcion()) - otro\_numero();***

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 11.- Introducción a las funciones en PHP (II)

Las funciones las podemos agrupar, básicamente, en las siguientes áreas:

- Funciones que trabajan con ***variables***.
- Funciones que trabajan con ***cadenas***
- Funciones que trabajan con ***arrays***
- Funciones que trabajan con ***fechas***
- Funciones ***definidas por el usuario***

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 12.- Funciones que trabajan con variables (I).

#### **empty(valor)**

- Comprueba *si la variable que pasamos está vacía o no*.
- Devuelve un valor booleano (true o false).

```
$a = 0;  
$b = empty($a);
```

true 

#### **isset(valor)**

- Indica si una *variable esta indefinida o contiene un valor*
- Devuelve un valor booleano (true o false).

```
$a = 5;  
$b = isset($a);
```

true 

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 12.- Funciones que trabajan con variables (II).

#### *is\_null(valor)*

- Comprueba **si la variable es igual a NULL** (error si se ejecuta var no definida).
- Devuelve un valor booleano (true o false).

#### *is\_integer(valor)*

- Comprueba si una **variable es de tipo entero**.
- Devuelve un valor booleano (true o false).

```
$a = 5.3;  
$b = is_integer($a); false 
```

#### *is\_float(valor)*

- Comprueba si una **variable es de tipo float o decimal**.
- Devuelve un valor booleano (true o false).

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 12.- Funciones que trabajan con variables (III).

#### ***is\_numeric(valor)***

- Comprueba **si la variable contiene un valor numérico**.
- Devuelve un valor booleano (true o false).

```
$a = "casa";  
$b = is_numeric($a);
```

false 

#### ***is\_array(valor)***

- Comprueba si una **variable es una matriz**.
- Devuelve un valor booleano (true o false).

#### ***is\_string(valor)***

- Comprueba si una **variable es una cadena**.
- Devuelve un valor booleano (true o false).



### 13.- Funciones que trabajan con cadenas (I)

Estas son las funciones que vamos a estudiar:

- Strlen
- Substr
- Trim
- Explode
- Mayúsculas y Minúsculas
  - strtolower
  - strtoupper
  - ucfirst
  - ucwords



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 13.- Funciones que trabajan con cadenas (II)

#### ***strlen(cadena)***

- Devuelve la longitud en caracteres de una cadena.
- Devuelve un número entero.

ver ejemplo45 y ejemplo46

```
$cadena = "casa";  
echo strlen($cadena);
```

4 ✓

#### ***substr(cadena)***

- Devuelve parte de una cadena dada. Argumentos:
  - cadena a manipular (primera posición: 0)
  - número: posición donde se empezará a extraer
  - número: cantidad de caracteres que serán extraídos (si no se pone, extrae hasta el final)

nota: existen **strpos()** y **strstr()**

```
$cadena = "En un lugar de la Mancha";  
$subcadena = substr($cadena, 6, 5);  
echo $subcadena;
```

lugar

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 13.- Funciones que trabajan con cadenas (II)

#### ***trim(cadena)***

- **Elimina** los **espacios** en blanco que se puede encontrar al principio y/o al final de la cadena.
- Si utilizo un segundo argumento, puedo eliminar los caracteres que ponga en el. ***trim(cadena, "caracteres")***;

ver ejemplo48

#### ***str\_replace(cadena1, cadena2, cadena3)***

- Sustituye la primera cadena con la segunda, dentro de la tercera.
  - distingue mayúsculas y minúsculas.

```
str_replace('g', 'p', 'gato');
```

pato



### 13.- Funciones que trabajan con cadenas (III)

<b>strtolower</b>	aplica letras <b>minúsculas</b>
<b>strtoupper</b>	aplica letras <b>mayúsculas</b>
<b>ucfirst</b>	<b>mayúscula</b> en la primera letra de la cadena
<b>ucwords</b>	<b>mayúscula</b> en la primera letra de cada <b>palabra</b>

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 13.- Funciones que trabajan con cadenas (IV)

#### ***explode(caracter, cadena, numero)***

- ***Transforma una cadena en un array.***
  - ***caracter*** que vamos a usar para diferenciar cada uno de los elementos del array
  - ***cadena*** que vamos a transformar en array
  - ***numero máximo de elementos*** que el array puede contener

```
$cadena = 'uno,dos,tres,cuatro,cinco';  
$array = explode(',', $cadena);  
echo $array[1];
```

dos

**Nota:** existe la función inversa → **implode()**

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra. Funciones de Cadenas

#### Ejercicio13

Partiremos de la siguiente cadena de caracteres:

**\$texto = "uno-dos-tres-cuatro-cinco";**

Crear una página que, a partir de esta cadena, muestre una lista con los elementos de la misma que resultan de partirla usando como carácter separador el guion '-' de la siguiente forma:

- uno
- dos
- tres
- cuatro
- cinco

Usa las funciones explode y count.



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 14.- Funciones que trabajan con Arrays. (I)

Veremos las más significativas (**count**, **sort**, **in\_array**) y nombraremos otras.

#### **count(array)**

- **Cuenta** el numero de elementos de la matriz.
- Devuelve un entero.

```
$color = array ('blanco', 'negro', 'amarillo', 'verde', 'rojo');  
echo count($color);
```



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 14.- Funciones que trabajan con Arrays. (II)

#### **sort(array)**

```
$color = array ('blanco', 'negro', 'amarillo', 'verde', 'rojo');
```

```
sort($color);    amarillo  
                  blanco  
                  negro  
                  rojo  
                  verde
```

- **Ordena** los elementos de la matriz.
- Los valores se disponen en **orden ascendente**.
- Los **índices** se pierden y **NO** devuelve nada.

Segundo argumento de la función:

- **SORT\_NUMERIC** compara los elementos de forma **numérica**
- **SORT\_STRING**    compara los elementos como **cadena**s
- **SORT\_NATURAL | SORT\_FLAG\_CASE** ordena texto y no tiene en cuenta mayúsculas ni minúsculas.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 14.- Funciones que trabajan con Arrays. (III)

#### *in\_array (valor, array)*

- **Encuentra** un valor dentro de la matriz.
- Devuelve un valor booleano dependiendo si el valor esta o NO dentro de la matriz.

```
$color = array ('blanco', 'negro', 'amarillo', 'verde', 'rojo');
```

```
in_array('negro', $color);
```

✓ TRUE

```
in_array('naranja', $color);
```

✗ FALSE

Hay un tercer parámetro (true, false) que indica que busque exacto el dato y el tipo de dato.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 15.- Funciones que trabajan con fechas. (I)

#### Date

Nos **devuelve una cadena**. Parámetros:

- **Formato** en que se mostrarán los datos.
- **Número entero**: representa una fecha que será convertida al formato establecido.

**timestamp**: es un numero entero que representa los segundos transcurridos desde el 1 de Enero de 1970

**time()** Este es el más simple de todos, ya que proporciona el **timestamp del momento** en el que se ejecuta. Devuelve un **entero (fecha y hora)**.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 15.- Funciones que trabajan con fechas (II)

#### *Date (formato)*

**Y** año con 4 dígitos

**n** número del mes (1-12)

**m** número del mes con 2 dígitos (01-12)

**F** mes textual (de 'January' a 'December')

**M** mes textual con 3 letras ('Jan' a 'Dec')

**d** día del mes en dos dígitos (01-31)

**j** día del mes (1-31)

**w** día de la semana, numérico  
(0 = domingo, 6 = sábado)

**I** día de la semana, en texto  
( 'Sunday' - 'Saturday' )

**D** día de la semana de 3 letras ('Sun' - 'Sat')

**H** hora en 2 dígitos (00-23)

**G** hora (0-23)

**i** minutos en dos dígitos (00-59)

**s** segundos en dos dígitos (00-59)



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 15.- Funciones que trabajan con fechas (II)

#### ***strtotime (fecha)***

Convierte una descripción de fecha/hora textual en Inglés a una fecha (nº segundos desde 01/01/1970)

#### ***mktime (hora, minuto, segundo, mes, día, año)***

Devuelve un entero (fecha y hora) en función de los parámetros.

#### ***checkdate (mes, día, año)***

Comprueba si los valores transmitidos representan una fecha válida. Devuelve un valor booleano.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

### Funciones de Fechas

#### Ejercicio15

Crear una página que muestre un mensaje de bienvenida que dependa de la hora actual, de la siguiente manera:

- Si la hora se encuentra entre las 8 y las 13, mostrará 'Buenos días'
- Si la hora se encuentra entre las 14 y las 20, mostrará 'Buenas tardes'
- Si la hora se encuentra entre las 21 y las 7, mostrará 'Buenas noches'

#### Ejercicio15-1

En primer lugar, deberemos crear una variable que guarde el número de día de hoy.

A continuación, haremos que, si la fecha es o está por debajo del día 10, aparecerá el mensaje "Periodo de pago activo".

Si el día es superior al 10, el mensaje que aparecerá es "Fuera del periodo de pago"



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 16.- Funciones definidas por el usuario. (I)

Podemos crear funciones para resolver **tareas repetitivas**.

Crear función:

```
Function nombre (argumentos) {  
    instrucciones  
}
```

Las funciones PHP **no se ejecutan** al ser creadas, deben ser **llamadas**.

```
function hola()  
{  
    echo("Hola!");  
} hola()
```

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 16.- Funciones definidas por el usuario. (II)

Podemos crear uno o mas **archivos** donde **defina todas** aquella **funciones** que vamos a utilizar en nuestra aplicación.

Para poder **incluir esas funciones** en otros archivos, vamos a utilizar la función include:

**include ('funciones.php');**

Esta es una opción muy interesante, no solo para usarlo con las funciones sino para cualquier archivo. Por ejemplo, para **dividir un sitio web en diferentes partes y reutilizarlas**, como la cabecera, el pie, publicidad, barras laterales, etc.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 16.- Funciones definidas por el usuario. (III)

**Parámetros** en nuestras funciones. Lo habitual es que las funciones precisen de algún **parámetro** para trabajar:

- por valor
- por referencia
- por defecto

Paso de parámetros **por valor**.

```
function hola($nombre){  
  print 'Hola '.$nombre;  
}
```

```
hola('Pedro');
```

**Hola Pedro**

(sigue)

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 16.- Funciones definidas por el usuario. (IV)

#### Paso de parámetros **por valor**. (sigue)

- Si cambiamos el valor del argumento dentro de la función, **no se ve modificado fuera de ella**.
- Los parámetros tienen validez **durante la ejecución** de la función.

```
function saludo($cadena){  
    $cadena = 'HOLA';  
}
```

```
$palabra = 'ADIOS';  
saludo($palabra);  
echo $palabra;
```

**ADIOS**

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 16.- Funciones definidas por el usuario. (V)

Paso de parámetros **por referencia**.

**SI SE MODIFICA LA VARIABLE DENTRO=>SE MODIFICA TB. FUERA**

El cambio del valor de un parámetro dentro de una función **afecta al valor de la variable** original.

En la declaración de la función escribimos “&” antes del parámetro.

```
function saludo(&$cadena)
{
    $cadena = 'HOLA';
}
```

```
$palabra = 'ADIOS';
saludo($palabra);
echo $palabra;
```

**HOLA**



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 16.- Funciones definidas por el usuario. (VI)

#### Paso de parámetros **por defecto**.

Podemos establecer valores por defecto.

Se usan **cuando no se facilitan** alguno o todos los parámetros que precisa la función.

```
function hola ($nombre="amigo"){  
    print 'Hola '.$nombre;  
}
```

```
hola('Pedro');
```

**Hola Pedro**

```
hola();
```

**Hola amigo**

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 16.- Funciones definidas por el usuario. (VII)

Número **variable** de parámetros.

```
function sumar(...$números) {  
    $total = 0;  
    foreach ($números as $n) {  
        $total += $n;  
    }  
}
```

Para saber que versión de PHP tengo, uso la función **phpinfo()**;

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 17.- Enviando información con PHP. (I)

Aprenderemos como enviar información a una aplicación PHP.

- Aportará **gran interactividad** a nuestras aplicaciones
- Requiere que tengamos conocimientos de **HTML**
- Y también conocimientos de **Formularios**: será la principal manera de enviar la información en nuestras aplicaciones.

Dos métodos para enviar información:

- **GET**
- **POST**

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 17.- Enviando información con PHP. (II)

#### El método **GET**

- Permite **enviar datos recogidos a una página**, son enviados usando la misma **url**.
- Cuando la página termine de cargarlos, **serán visibles** para el usuario.

```
<a href="pagina.php?
usuario=pedro&codigopostal=28080">Enviar</a>
```

```
pagina.php
?
usuario=pedro
&
codigopostal=28080
```



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 17.- Enviando información con PHP. (III)

El método **GET** – Recuperar los datos del formulario.

Para recuperar los datos, usaremos la variable global:

**`$_GET`**

En el caso anterior:

```
$usuario = $_GET['usuario'];  
$codigopostal = $_GET['codigopostal'];
```

#### Limitaciones:

- La longitud de la url puede ser considerable y eso podría afectar negativamente al **posicionamiento**.
- La **longitud máxima** de una url es de **2048** caracteres.
- La información enviada será **visible por todo el mundo** y se guardará en el historial del navegador.



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 17.- Enviando información con PHP. (IV)

#### El método **POST**

- Solo funciona enviando la información mediante **formularios**.
- La información enviada va en el **cuerpo de la solicitud de HTTP** (no se ve)
- Si queremos puede ir incluso **encriptada**.

#### Ventajas:

- No tiene ninguna restricción de **tamaño**.
- Acepta **caracteres ASCII** y **datos binarios** (envío de archivos).
- Si se usa bajo el **protocolo HTTPS** los datos viajan de manera segura.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 17.- Enviando información con PHP. (IV)

#### Envío de Archivos

- Envío de datos binarios (documentos o imágenes).
- Se hace mediante un formulario con el **método POST**.
- En el **formulario** hay que especificar que se va a realizar el envío.

```
<form enctype="multipart/form-data"  
action="pagina.php" method="POST" >
```

- El archivo PHP que procese los datos no va a usar la variable \$\_POST

```
$_FILE
```

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

### Enviando información con PHP

#### Ejercicio1

Crearemos un formulario que nos pida la introducción de un nombre de usuario y dos campos para escribir dos veces una contraseña.

A través de una función definida por el usuario, la página que procesa la contraseña debe indicarnos si las contraseñas son iguales o diferentes.

#### Ejercicio2

confeccionar un formulario con dos campos que solicite el nombre de persona y su edad

Estos datos serán recibidos por el archivo del ejercicio que indicará si es mayor de edad (si la edad es mayor o igual a 18)

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

### Enviando información con PHP

#### Ejercicio17-3

Crearemos un formulario en el que incluiremos un campo de texto, para que se introduzca el nombre de una persona y tres controles de tipo radio que permitirán escoger si la persona no tiene estudios, tiene estudios primarios o secundarios.

Otra página, por ejemplo, procesará los datos, mostrando en pantalla el nombre de la persona y el nivel de estudios seleccionado.

#### Ejercicio17-4

Crearemos un formulario que solicite la carga del nombre de una persona y que permita seleccionar una serie de deportes que practica con controles del tipo checkbox (futbol, basket, tennis, voley)

En un archivo aparte procesaremos el formulario y mostraremos el número de deportes que practica.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

### Enviando información con PHP

#### Ejercicio17-5

Confeccionaremos un formulario que solicite el nombre de una persona y un cuadro combinado para realizar una selección de los ingresos mensuales de la persona: de 1-1000, de 1001-3000 o más de 3000

En la página que procesa el formulario mostrar un mensaje si debe pagar impuestos a las ganancias (si supera 3000).

#### Ejercicio17-6

En el primer archivo, a través de un formulario con un solo campo, el usuario escogerá escribiendo un número cuantas veces desea que se repita el bucle.

Al pulsar el botón "Aceptar", se ejecutará en el archivo ejercicio17-6.php. Su cometido será: mostrar el mensaje "Los bucles son fáciles!" tantas veces como se haya indicado en el formulario. Consejo: usa While para llevar a cabo esta tarea.



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

### Enviando información con PHP

#### Ejercicio17-7

Realizaremos el mismo ejercicio que en la práctica anterior pero en este caso haremos que el formulario y el resultado se muestren en el mismo archivo, ejercicio17-7.php

#### Ejercicio17-8

Crear un formulario que nos permita subir hasta tres archivos al servidor. Podemos subir imágenes y mostrarlas en la página que gestione la subida de los archivos.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 18.- Acceso a archivos. (I)

Acceder a fuentes de datos externas con el fin de **recuperar** o **guardar** información.

Las principales operaciones con archivos son:

- **Lectura**
- **Escritura**
- **Posicionamiento**



## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 18.- Acceso a archivos. (II)

#### Abrir y Cerrar ficheros.

**FOpen**: abre un archivo del servidor local o de una dirección web o de un FTP, para **leerlo o escribirlo**.

#### Argumentos:

- **Nombre** del archivo que deseamos abrir
- **Modo** de **apertura**.

<b>r</b>	Solo lectura
<b>r+</b>	Lectura y escritura
<b>w</b>	Solo escritura Si el archivo especificado no existe, lo crea. Si existe lo borra y lo vuelve a crear
<b>w+</b>	Lectura y escritura Si el archivo especificado no existe, lo crea. Si existe lo borra y lo vuelve a crear

<b>a</b>	Solo escritura Si no existe el archivo lo crea. Si existe empieza a escribir al final del archivo
<b>a+</b>	Lectura y escritura Si no existe el archivo lo crea. Si existe empieza a escribir al final del archivo

## 18.- Acceso a archivos. (IV)

### **FOpen** MODOS DE APERTURA

*La diferencia entre los modos:*

**a**

**a+**

colocan el puntero de lectura/escritura **al final del archivo**

**r**

**r+**

**w**

**w+**

colocan el puntero de lectura/escritura **al principio del archivo**

*Ej: `fopen("archivo.txt", r);` // abre para lectura colocándose al principio del fichero "archivo.txt".*

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### 18.- Acceso a archivos. (V)

**FGets** Leer ficheros.

La función **fgets()** recupera el contenido de **una línea** de un archivo.

Si queremos recuperar hasta la última línea de un archivo, debemos trabajar conjuntamente con un bucle **while**.

**Feof** Final de fichero.

La función **feof()** nos devuelve:

- **true** cuando hemos llegado al final del fichero
- **false** si no lo hemos alcanzado todavía

En nuestras aplicaciones lo usaremos junto con **while** para leer el fichero completo, hasta que sea **true**.



## 18.- Acceso a archivos. (VI)

**Fputs** Escribe una línea en un fichero

```
$fp = fopen("fichero.txt", "w");  
fputs($fp, "Estoy escribiendo en un fichero");  
fclose($fp);
```



Una vez hemos de **terminado** de operar con el fichero escribimos la instrucción

```
fclose($a)
```

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

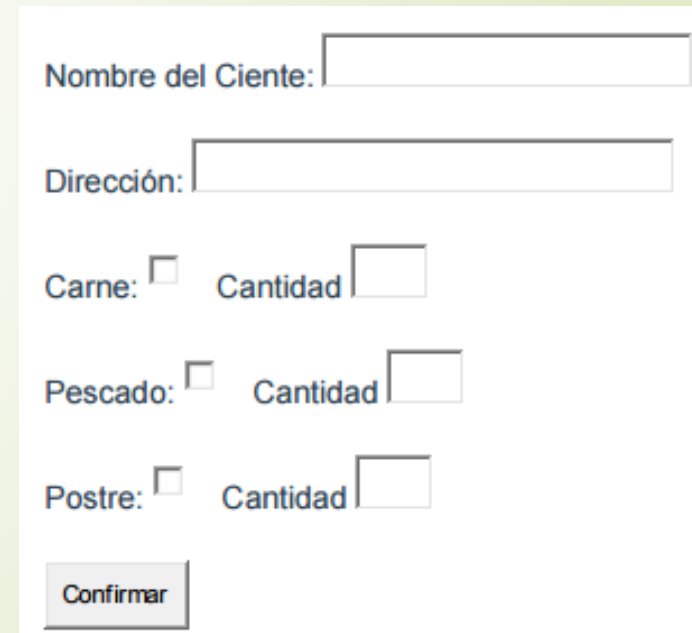
### Manos a la obra.

### Acceso a Archivos

#### Ejercicio18-1

En primer lugar vamos a elaborar un formulario con un pequeño menú de restaurante, similar al que te mostramos a continuación:

- El contenido de este formulario deberá ser grabado en un archivo de texto llamado ***pedidos.txt***
- Para facilitar el trabajo, puedes crear también el archivo pedidos.txt en blanco y darle permisos de escritura.
- El formulario crearlo como ***pedidos.html*** y el alta del pedido como ***pedidos.php***



A screenshot of a web form for restaurant orders. The form is white with a light gray border. It contains the following elements:

- Nombre del Ciente:** A text input field.
- Dirección:** A text input field.
- Carne:** A checkbox followed by the label "Carne:".
- Cantidad:** A text input field.
- Pescado:** A checkbox followed by the label "Pescado:".
- Cantidad:** A text input field.
- Postre:** A checkbox followed by the label "Postre:".
- Cantidad:** A text input field.
- Confirmar:** A button with the text "Confirmar".

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

### Acceso a Archivos

#### Ejercicio18-2

Partiendo del supuesto del ejercicio anterior, tenemos un archivo llamado ***pedidos.txt*** en el que nos han dejado guardado el menú solicitado.

Crear una aplicación que muestre el contenido del archivo de pedido. Llamarlo ***leerpedidos.php***

```
Nombre:Jose Garcia
Dirección:Otra calle que no se, 5, 2º
Cantidad de Carne:2
Cantidad de Pescado:5
Cantidad de Postre:3
-----
```

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

### Acceso a Archivos

#### Ejercicio18-3

En este ejercicio crearemos una aplicación que permita **registrar una queja** en un archivo de texto (**quejas.txt**). Elaboraremos un formulario que permita almacenar el nombre del visitante (objeto HTML de tipo "text"), la descripción de la queja (objeto de tipo "textarea") y también guardaremos en el archivo de texto la fecha y hora de la queja de forma automática.

- El archivo "**quejas1.php**" debe contener ese formulario de ingreso de datos.
- El archivo "**quejas2.php**" debe registrar la información en el archivo de datos.
- Por último el archivo "**quejas3.php**" debe imprimir el archivo con todas las quejas registradas hasta el momento.
- Crearemos un hipervínculo en el archivo "**quejas1.php**" para poder ver todas las quejas registradas.

## UF1844: Desarrollo de aplicaciones web en el entorno servidor.

### Manos a la obra.

### Acceso a Archivos

#### Ejercicio18-4

Vamos a crear un sencillo **contador de visitas** para nuestra página.

Crearemos un archivo de texto llamado, por ejemplo, **contador.txt**. En él guardaremos una cifra.

Cada vez que alguien abra la **principal.php**, donde se encuentra nuestro programa, se sumará una visita a ese contador.

Esta es una forma sencilla de trabajar con archivos, **abrirlos, leerlos, grabarlos y cerrarlos**.