



**CICLOFORMATIVO DE GRADOS SUPERIOR-TÉCNICO EN  
ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN REDES**

**IMPLANTACIÓN APLICACIONES WEB**

**EJERCICIOS PHP**

**1. VARIABLES, COMENTARIOS Y VISUALIZACIÓN DE CONTENIDOS EN NAVEGADOR:**

**Crea dos variables cuyo nombre sea “var1” y “var2” e imprímelas en la página. Pon un comentario con el tipo de dato que contienen:**

Para imprimir contenido en la página usamos la instrucción **echo**.

**2. BUCLE FOR y OPERACIONES:**

**Escribe un programa que imprima por pantalla los cuadrados (el número multiplicado por sí mismo) de los 20 primeros números naturales:**

Necesitas implementar un bucle que se repita 20 veces utilizando la instrucción **for**.

**3. CONDICIONAL IF Y OPERADORES DE COMPARACIÓN**

**Modifica el ejercicio anterior para mostrar por pantalla si el cuadrado calculado es un número par o impar:**

Utiliza la instrucción **if()...else** para filtrar. Recuerda que podemos calcular el resto de una división con **%**. Ej:  $4\%2 \rightarrow 0$  y  $5\%2 \rightarrow 1$ .

#### 4. BUCLE WHILE Y OPERADORES DE ASIGNACIÓN:

**Escribe un programa que calcule los 20 primeros números de la sucesión de fibonnaci. Utiliza el bucle WHILE.**

A diferencia del for, en el bucle while solamente podemos pasar una condición, en este caso, que n sea menor o igual a 20.

Para simplificar el código, podemos multiplicar también de la siguiente formas:

- `$j = $j * $i`      #Tradicional
- `$j *= $i.`      # \*= asigna a la variable de almacenamiento (variable j) el valor de multiplicar esa misma variable (valor de j) por la otra variable pasada (valor de i).

#### 5. CONDICIONALES, BUCLES Y VARIABLES GET:

**Generar un programa que muestre en el navegador la tabla de multiplicar de un numero pasada por el parámetro GET de la URL:**

Durante una transacción http con un servidor php, se genera un conjunto de variables globales que podemos utilizar para gestionar la sesión específica del usuario. Para gestión de variables por solicitud, tenemos las variables `$_GET[]` y `$_POST[]` que en función del método utilizado para enviar la solicitud contendrán información o no.

#### COSAS A TENER EN CUENTA:

##### ¿EXISTE \$\_GET?

**isset(\$variable)** → Se utiliza en PHP para saber si una variable está definida y no es nula (NULL)

En este caso, queremos saber si la variable “tabla” se ha pasado por el parámetro get, por tanto, colocamos:

```
isset($_GET["tabla"])
```

##### ¿ES UN NÚMERO?

**is\_numeric(\$variable)** → Se utiliza en PHP para comprobar si una variable es numérica o no.

## 6. ARRAYS Y BUCLES

**Crear un array llamado meses y que almacene el nombre de los doce meses del año. Recorrerlo con FOR para mostrar por pantalla los doce meses.**

**count(\$meses)** → En PHP la función count() se usa para contar el número de elementos que hay en un array. La función devuelve el número de elementos del array como un número entero, en este caso, 12.

Si quisiéramos introducir con parámetro GET el dato que queremos leer:

```
$mes = $_GET["mes"];  
echo $meses[$mes];
```

## 7. EL BUCLE FOREACH

**Igual que el ejercicio anterior pero usando el bucle foreach:**

El bucle foreach es el que más se utiliza para recorrer este tipo de array donde los datos no son uniformes.

**foreach(\$array as \$value)** → Recorre el array \$array como variables \$value

## 8. PROFUNDIZANDO EN LOS ARRAYS

Hacer un programa que tenga un array de 8 números enteros y resolver lo siguiente con él:

- Recorrerlo y mostrarlo
- Ordenarlo y mostrarlo
- Mostrar su longitud
- Buscar un número concreto en el vector

Si quisiéramos ordenar de manera descendente, usaríamos **rsort(\$numeros)**

```
#Ordenar en descendente:
rsort($numeros);
foreach ($numeros as $num) {
    echo $num."<br>";
}
```

En PHP, la función `array_search()` se usa para buscar un valor en un array. Recuerda que el operador `!` (not) se usa para negar una expresión booleana.

## 9. DETECTAR EL NAVEGADOR WEB:

**Escribe un programa que muestre la dirección IP del usuario que visita nuestra web y si usa Firefox le mostramos un mensaje positivo:**

**\$\_SERVER** → Es una variable superglobal en PHP que contienen información sobre el servidor web y el entorno en el que se está ejecutando el script actual.

Algunas de las claves más comunes en la variable `$_SERVER` son:

`$_SERVER['DOCUMENT_ROOT']`: La ruta completa del directorio raíz del servidor web.

`$_SERVER['HTTP_HOST']`: El nombre de host del servidor web.

`$_SERVER['PHP_SELF']`: El nombre del archivo del script actual.

`$_SERVER['REMOTE_ADDR']`: La dirección IP del cliente que está accediendo a la página.

`$_SERVER['REQUEST_METHOD']`: El método utilizado para enviar la solicitud HTTP (GET, POST, PUT, DELETE, etc.).

`$_SERVER['SERVER_NAME']`: El nombre del servidor web que está sirviendo la página.

`$_SERVER['SERVER_SOFTWARE']`: La información sobre el software del servidor web utilizado.

## 10. RELLENAR UN ARRAY

**Escribe un programa que añada valores a un array mientras que su longitud sea menor a 50 y después que se muestre la información del array por pantalla**

La función `var_dump` no se ve estéticamente correcto si utilizamos XAMPP.

## 11. ARRAYS MULTIDIMENSIONALES

**Crea un array con el contenido de la siguiente tabla y genera una tabla html a partir del array:**

Frutas	Deportes	Idiomas
Manzana	Fútbol	Español
Naranja	Tenis	Inglés
Sandía	Baloncesto	Francés
Fresa	Béisbol	Italiano