

TRABAJO DE MÓDULO

**DEL PHP AL SQL
UF184-UF184-UF1846**

**UNA PROPUESTA DE APLICACIÓN
BASADA EN LOS MÓDULOS
FORMATIVOS**

**BELÉN SERRANO
JAUME LOZANO**

ÍNDICE

1. HTML, UN LENGUATGE DE MARCAT	(PG. 1)
2. CSS, L'ESTIL IMPORTA	(PG. 3)
3. JAVASCRIPT, MOU-TE!	(PG. 5)
4. WORDPRESS, UN SISTEMA D'ÈXIT	(PG. 8)
5. TANCAMENT	(PG. 10)

UN APLICATIVO

UN FORMULARIO, LO QUE VES Y LO QUE NO

Desde el equipo que forman los autores del documento se ha propuesto la creación de una agencia de viajes. Es por eso que se ha iniciado un estudio para valorar la viabilidad y estudiar los intereses de los potenciales clientes así como la situación del mismo mercado. Por eso, con el objetivo de poder recoger esa información se ha impulsado el siguiente formulario. Se trata de un formulario elaborado con lenguaje PHP y que está formado por los "Inputs" es decir, las partes que lo componen, básicos para su funcionamiento.

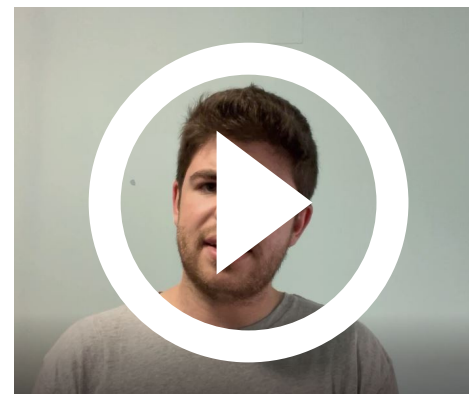
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <form action="valida-envio.php" method="get">
    <!-- <form action="valida-envio.php" method="post"> PARA ENVIAR POR POST-->
    <input type="text" name="nombre" placeholder="Nombre" id="">
    <input type="email" name="email" placeholder="Email" id="">
    <input type="submit" name="submit" value="Enviar formulario" >
  </form>

  <!-- Segundo formulario de prueba -->
  <form action="valida-envio.php" method="post">
    <input type="text" name="nombre" placeholder="Nombre" id="">
    <input type="email" name="email" placeholder="Email" id="">
    <input type="submit" name="submit2" value="Enviar formulario2" >
  </form>
</body>
</html>
```

Figura 1. Fuente: Elaboración Propia

Formulario creado con lenguaje PHP /
Fuente: Elaboración Propia

Lo que se visualiza, es desde el punto de vista de la programación, un elemento que forma parte del back-end, en la figura 1, y del front-end en la figura dos, la parte que se visualiza. Es importante tener en cuenta las diferencias existentes entre el front-end y el back-end. Esta diferenciación forma parte también de la clasificación que dentro de las especializaciones de programación se utiliza para definir a los profesionales. Por ejemplo, una determinada empresa, podría requerir los servicios de un programador "back-end", que se centrará en la creación de la estructura software web que no se visualiza por parte en la parte del cliente/usuario.



Video explicativo de las diferencias entre
back-end y front-end /
Font: Elaboración propia

PHP, DEL CLIENTE AL SERVIDOR

PHP es un lenguaje de programación de páginas de contenido dinámico. Es un lenguaje del lado del servidor. Desde el servidor se lee el lenguaje php, y se mandan los datos transformados en lenguaje HTML. PHP se incorpora al lenguaje normal de la página (HTML) mediante scripts.

Fue creado originalmente por Rasmus Lerdorf en 1995. Actualmente, va por la versión 8.1.7, y trabaja conjuntamente con otros programas como son la base de datos MySQL y el servidor Apache.

Para crear páginas con PHP debemos convertir nuestro ordenador en un servidor local, de manera que podamos ver los resultados de nuestro trabajo antes de subirlo a la red.

Tener un servidor local en el ordenador requiere tener instalados varios programas, los cuales trabajan conjuntamente. Hasta no hace muchos años debían instalarse por separado y configurarlos después.

Por suerte hoy en día tenemos el paquete integrado XAMPP que contiene ya todos los programas y los instala y configura de una manera sencilla.

Una vez instalado, al abrir el programa nos encontramos con un panel de control. Deben estar marcados ahí los programas "Apache" y "MySQL" para que XAMPP funcione correctamente.

```
<html>
<head>
  <title>Prueba de PHP</title>
</head>
<body>
  <?php echo '<p>Hola Mundo</p>'; ?>
</body>
</html>
```

Ejemplo de script básico de PHP

ELEMENTOS PHP

Existen una serie de elementos que constituyen y estructuran el lenguaje PHP. Entre ellos, encontramos las **variables**, que son los elementos que se definen en el código y pueden cambiar, las **constantes**, que tal como indica el nombre, mantienen su valor durante el documento así como los **operadores** y **funciones** de diferentes tipologías. En el caso de las funciones, pueden llegar a ser definidas por el usuario.

**VARIABLES: \$TEXTO = "HOLA MUNDO";
PUEDEN SER: N° ENTEROS, REALES, CADENAS DE TEXTO, ARRAYS, OBJETOS**

CONSTANTES: DEFINE():DEFINE("CONSTANTE1","UNA CONSTANTE")

OPERADORES DE ASIGNACIÓN:

- `+=` : `$a += $b;` // igual que `$a = $a + $b;`
- `-=` : `$a -= $b;` // igual que `$a = $a - $b;`
- `*=` : `$a *= $b;` // igual que `$a = $a * $b;`
- `/=` : `$a /= $b;` // igual que `$a = $a / $b;`
- `%=` : `$a %= $b;` // igual que `$a = $a % $b;`

OPERADORES CONDICIONALES

Nombre	Signo	Descripción	Ejemplo
igual	<code>==</code>	Devuelve 1 si los dos elementos son iguales	<code>\$n1 == \$n2</code>
idéntico	<code>===</code>	Devuelve 1 si los dos elementos son iguales, pero además los datos deben ser del mismo tipo.	<code>\$n1 === \$n2</code>
No igual	<code>!=</code>	Devuelve 1 si los dos elementos son distintos	<code>\$n1 != \$n2</code>
No idéntico	<code>!==</code>	Devuelve 1 si los dos elementos son distintos, pero además los datos deben ser del mismo tipo	<code>\$n1 !== \$n2</code>
Mayor que	<code>></code>	Devuelve 1 si el primer valor es mayor que el segundo.	<code>\$n1 > \$n2</code>
Menor que	<code><</code>	Devuelve 1 si el primer valor es menor que el segundo.	<code>\$n1 < \$n2</code>
Mayor o igual que	<code>>=</code>	Devuelve 1 si el primer valor es mayor o igual que el segundo.	<code>\$n1 >= \$n2</code>
Menor o igual que	<code><=</code>	Devuelve 1 si el primer valor es mayor o igual que el segundo.	<code>\$n1 <= \$n2</code>

ELEMENTOS PHP

OPERADORES LÓGICOS

Nombre	Signo	Descripción	Ejemplo
AND	and	Devuelve 1 sólo si las dos condiciones se cumplen, si no es así devuelve una cadena vacía.	<code>\$n1 != \$n2 and \$n1 != \$n3</code>
OR	or	Devuelve 1 si una de las dos condiciones se cumple o si se cumplen las dos. Sólo si las dos no se cumplen devuelve cadena vacía.	<code>\$n1 == \$n2 or \$n1 == \$n3</code>
XOR	xor	Devuelve 1 sólo cuando una de las dos condiciones se cumple. Si no se cumple ninguna o se cumplen las dos devuelve cadena vacía.	<code>\$n1 == \$n2 xor \$n1 == \$n3</code>
AND (2)	&&	Exactamente igual que and	<code>\$n1 != \$n2 && \$n1 != \$n3</code>
OR (2)	 	Exactamente igual que or	<code>\$n1 == \$n2 \$n1 == \$n3</code>
Negación	!	Cambia el valor del elemento al que se le aplica es decir, si \$n1 es verdadero (1) lo cambia a falso (""), y viceversa	<code>!\$n1</code>

FUNCIONES: SON TROZOS DE CÓDIGO QUE NO SE EJECUTAN CUANDO SE LEEN, SINO QUE SE GUARDAN Y SE PUEDEN EJECUTAR MÁS TARDE CON UNA LLAMADA A LA FUNCIÓN DESDE CUALQUIER PARTE DEL CÓDIGO.

```
function nombreFuncion($argumento1,$argumento2) {  
    <!-- // ...instrucciones de la función ...// -->  
    return $resultado;  
}
```

IF

```
if ( /*condición*/ ) { /*código a ejecutar si se cumple la condición*/ }  
else { /*código a ejecutar si la condición no se cumple*/ }
```

ELSEIF

```
if ( /*primera condición*/ )  
{ /*código a ejecutar si se cumple la primera condición*/ }  
elseif ( /*segunda condición*/ )  
{ /*código para la segunda condición*/ }  
elseif ( /*tercera condición*/ )  
{ /*código para la tercera condición*/ }  
.....  
else { /*código cuando no se cumple ninguna de las condiciones anteriores*/ }
```

CONDICIONALES:

SWITCH

```
switch ($variable) {  
    case "valor1":  
        /*código a ejecutar para el valor1;*/  
        break;  
    case "valor2":  
        /*código a ejecutar para el valor2;*/  
        break;  
    case "valor3":  
        /*código a ejecutar para el valor3;*/  
        break;  
    .....  
    default:  
        /*código si la variable no tiene ningún valor de los anteriores;*/  
}
```

ELEMENTOS PHP

WHILE

```
$i=1; //variable de control del bucle, en estado inicial. while ($i<=10)
{ //Condición: la variable debe ser menor o igual que 10. echo "7 x $i = ".7*$i." ";
  //en pantalla línea de la tabla. $i++; //aumentamos la variable en una unidad. }
```

BUCLES:

DO... WHILE

```
$n=0;
<!-- //Número de vueltas -->
$i=1;
<!-- //variable de control -->
do { echo "$i: Esta frase se repetirá $n veces";
  <!-- //texto en pantalla -->
  $i++;
  <!-- //aumentar en una unidad la variable de control -->
} while ($i<=$n)
<!-- //parar el bucle cuando llegue al número de vueltas. -->
```

FOR

```
for ($i=1 ; $i<=10 ; $i++) {
  //ini.:en 1; cond.:<=10; act.:número siguiente. $n= $i*7;
  //multiplicar echo "7 x $i = $n ";
  //sacar en pantalla }
```

FOREACH

```
foreach ($array as $valor) { echo "$valor, "; }
```

COOKIES, SE ACORDARÁN DE TI

Una cookie es un archivo que se crea automáticamente para poder guardar un dato o una variable. Una vez creado para utilizarlo en otra página no tenemos más que abrirlo en esa otra página.

- La cookie se guarda como un archivo en el navegador del usuario. Siempre que el usuario no haya desactivado el uso de cookies en el navegador.
- Las cookies tienen fecha de caducidad.
- El usuario debe utilizar el mismo navegador para recuperarlas.
- Se puede acceder a una cookie desde cualquier página del sitio, y durante el tiempo que esté activa, siempre que se sepa su nombre.
- Una cookie sólo almacena texto y no puede ocupar más de un Kb.

INSERTAR COOKIES

utilizamos la función
`set_cookie()`.
`set_cookie($nombre,$v
alor,$caducidad);`

ACCEDER COOKIES

Utilizamos el nombre
de la cookie para
verla en pantalla.
`echo
$_COOKIE['nombre'];`

BORRAR COOKIES

Se borra
automáticamente al
caducar.
También se puede
borrar
sobrescribiendo:
`set_cookie()`, pero
poniendo sólo el
primer argumento, es
decir, el nombre:
`set_cookie('nombre');`

LAS SESIONES

Mientras una persona entra en nuestra página web y se dispone a rellenar el formulario se puede afirmar que se está generando una sesión. Estas se pueden definir como la manera de almacenar información que posteriormente se utilizará para mejorar el funcionamiento de determinadas páginas. Una información, que en este caso, y a diferencia de las cookies, se almacenará en el navegador y no en el dispositivo.

```
<?php  
  
session_start();  
  
echo 'Bienvenido a tu perfil. Puedes navegar por la plataforma<br />';  
echo date('H:i:s d m Y', $_SESSION['El momento concreto']);  
?>
```

Ejemplo de un inicio de sesión en PHP Elaboración Propia

Para la configuración de una sesión se utilizarán las que se conoce como variables de sesión. Son unos componentes que permiten al congruación de estas sesiones, que, de forma predeterminada durarán hasta que el usuario cierra el nevagdor, o hasta el día siguiente, a partir de las 00:00 horas. Es importante remaracar que estas variables de sesión disponen de información sobre solo un usuario y en ese sentido, están dispoonibles para todas las páginas dentro de un aplicativo. Para el uso y la configuración de las sesiones se utilizarán las siguientes funciones:

SESSION_START()

CON ESTA FUNCIÓN
EMPEZAMOS UNA
SESIÓN

SESSION_DESTROY()

LA FUNCIÓN QUE SE
UTILIZA PARA
CERRAR UNA SESIÓN

\$_SESSION

ESTA ES LA VARIABLE QUE
DEBEMOS CONCRETAR EN
EL SCRIPT PARA
ESTABLECER UNA
VARIABLE DE SESIÓN.
\$_SESSION ES UNA
VARIABLE GLOBAL.

EL VIAJE DE LOS DATOS: POST Y GET

Una de las funcionalidades clave de PHP es la creación de aplicativos como es el caso de los formularios, tal y como se ha mostrado con anterioridad. Para que estos se ejecuten se debe preparar su estructura, que se puede hacer tanto utilizando PHP como HTML y por otro lado su mecanismo básico de envío.

Este último mecanismo será el que envíe aquellas respuestas que se hayan seleccionado en el formulario. Existen dos formas de envío en función de las preferencias de los creadores. Por un lado el método **GET**, en el que los datos se guardarían en un array asociativa en la variable \$GET. En este caso no sería recomendable utilizarlo para web ya que la información completada se nos mostraría en la dirección después del envío correspondiente.

Sería óptimo utilizar el método POST ya que nos ofrece más seguridad y además tendremos la opción de que en caso de no introducir los valores necesarios en el aplicativo se nos vuelva a redireccionar al formulario. Las dos opciones se muestran a continuación:

```
<head>
<body>
  <form action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF'])
?>" method="get">
    <input type="text"
      name="uno de los inputs" placeholder="??">
  </form>
</body>
</html>
```

Ejemplo de envío por el método GET

```
<head>
<body>
  <form action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF'])
?>" method="post">
    <input type="text"
      name="uno de los inputs" placeholder="??">
  </form>
</body>
</html>
```

Ejemplo de envío por el método POST

PRESENTACIÓN DE LA BASE DE DATOS (PENDIENTE)

PHP es un lenguaje de programación de páginas de contenido dinámico.

Es un lenguaje del lado del servidor. Desde el servidor se lee el lenguaje php, y se mandan los datos transformados en lenguaje HTML.

PHP se incorpora al lenguaje normal de la página (HTML) mediante scripts.

Fue creado originalmente por Rasmus Lerdorf en 1995. Actualmente, va por la versión 5, y trabaja conjuntamente con otros programas como son la base de datos MySQL y el servidor Apache.

```
<html>
<head>
  <title>Prueba de PHP</title>
</head>
<body>
  <?php echo '<p>Hola Mundo</p>'; ?>
</body>
</html>
```

PHP es un lenguaje de programación de páginas de contenido dinámico.

Fue creado originalmente por Rasmus Lerdorf en 1995. Actualmente, va por la versión 5, y trabaja conjuntamente con otros programas como son la base de datos MySQL y el servidor Apache.

Ejemplo de script básico de PHP

LA CREACIÓN DE LA BASE DE DATOS REVISAR

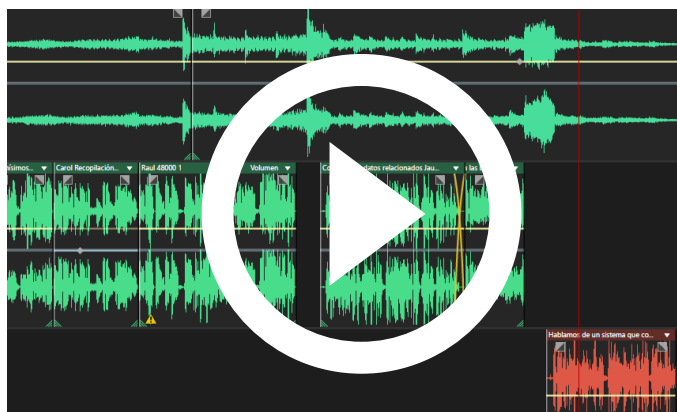
Una vez se obtuvieron las primeras respuestas del formulario, recuperando la idea de análisis de la situación expresada a la inicio, se confeccionó una bases de datos con un número limitado de registros que permite almacenar los primeros datos. Para la recogida esta información se diseñó un formulario a través de la herramienta Google Forms. Esta permitia la creación de la reducida base de datos de manera más segura, rápida y eficaz. En este formulario se definieron preguntas que se podían contestar siguiendo una serie de parámetros como son valores de fecha, de texto corto o largo o bien de selección única de opciones.

A screenshot of a Google Form titled "Preferencias en vacaciones". The form has a light blue header with the title. Below the title, there is a greeting "Hola," and a paragraph explaining that it is a survey for academic work and that responses are anonymous. The form contains two input fields: "Código Postal (CP)" with a red asterisk indicating it is required, and "Fecha de nacimiento" also with a red asterisk. The date field has a calendar icon. On the right side of the form, there is a vertical toolbar with icons for adding, deleting, and duplicating questions, as well as a "Tr" icon and a list icon.

Formulario para la creación de la bases de datos. / Fuente: Elaboración propia

¿BASE DE DATOS? REVISAR

Con el objetivo de explicar algunas de las características o acercar la definición de lo que es una base de datos se ha creado un documento de audio de 2 minutos i 42 segundos en los que se define. Además se incorporan voces que pueden llegar a representar una parte de la sensibilización social con este concepto.



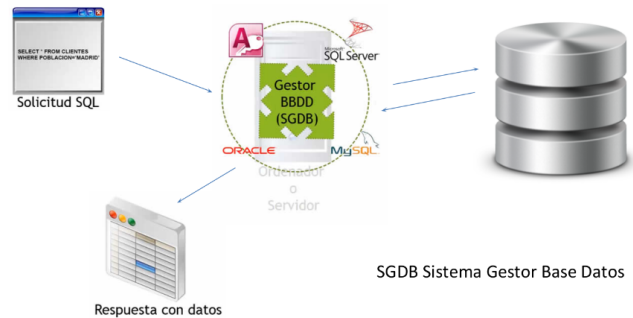
Archivo de audio sobre las bases de datos. / Fuente: Elaboración propia

EL LENGUAJE SQL (REVISAR DEFINICIÓN SQL)

Cuando hablamos de SQL, se trata del Structured Query Language. Lenguaje estructura de consultas. Inventado por IBM (SEQUEL).

PRINCIPALES COMANDOS

DDL	DML	DCL	TCL
CREATE	SELECT	GRANT	COMMIT
TRUNCATE	DELETE	REVOKE	ROLLBACK
DROP	UPDATE		SAVEPOINT
ALTER	INSERT		



COMANDO* + CLÁUSULAS* + OPERADORES + FUNCIONES

FROM + WHERE + GROUP BY + HAVING + ORDER BY

Partimos de una base de datos generada a través de google en la que hemos recogido las preferencias de tipo de vacaciones de una muestra de la población, así como el presupuesto que estarían dispuestos a destinar

A partir de esta base de datos y de las consultas que hagamos podremos tomar decisiones inteligentes para el desarrollo de una app que proporcione un servicio viajes personalizados.

CONSULTAS:

Queremos analizar las preferencias de los destinos de vacaciones.

La consulta sería: *****

Queremos saber el porcentaje de los ingresos que se dedican a las vacaciones.

La consulta sería: *****

Queremos saber qué código postal de Barcelona dedica más dinero a las vacaciones. La consulta sería: *****

Queremos lanzar un promoción de viaje de montaña/rural, para un sector de la población que vive en Barcelona y alrededores y cuyo presupuesto anual en vacaciones es superior a *****. La consulta sería: *****

ÁLGEBRA RELACIONAL, LA BASE DE LAS CONSULTAS (PENDIENTE)

PHP es un lenguaje de programación de páginas de contenido dinámico. Es un lenguaje del lado del servidor. Desde el servidor se lee el lenguaje php, y se mandan los datos transformados en lenguaje HTML. PHP se incorpora al lenguaje normal de la página (HTML) mediante scripts.

Fue creado originalmente por Rasmus Lerdorf en 1995. Actualmente, va por la versión 5, y trabaja conjuntamente con otros programas como son la base de datos MySQL y el servidor Apache.

Ejemplo de script básico de PHP

CONCLUSIONES

UNA REFLEXIÓN FINAL (PENDIENTE)

PHP es un lenguaje de programación de páginas de contenido dinámico. Es un lenguaje del lado del servidor. Desde el servidor se lee el lenguaje php, y se mandan los datos transformados en lenguaje HTML. PHP se incorpora al lenguaje normal de la página (HTML) mediante scripts.

- PHP como elemento imprescindible para saber como funciona la interacción en la web con el usuario a través de aplicativos como el formulario.
- SQL: Imprescindible para saber lo que se ejecuta cuando hacemos una solicitud. Como se llevan a cabo estás.
- SQL: Sensibilización con el número tan grande de posibilidades que ofrece

Fue creado originalmente por Rasmus Lerdorf en 1995. Actualmente, va por la versión 5, y trabaja conjuntamente con otros programas como son la base de datos MySQL y el servidor Apache.

Ejemplo de script básico de PHP



Belén Serrano / Jaume Lozano
Castelldefels, 2022.