

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. UN APLICATIVO	(PG. 1)
2. PHP	(PG. 2)
3. SQL	(PG. 8)
4. PRESENTACIÓN DE LA BASE DE DATOS	(PG. 8)
5. CONCLUSIONES	(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.

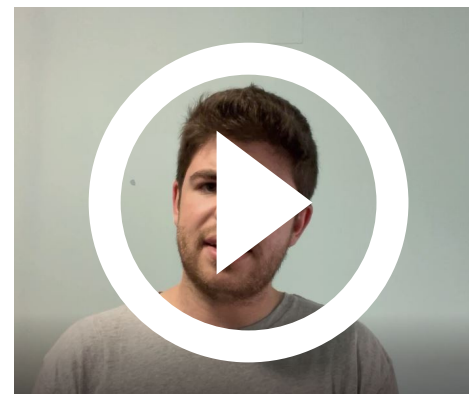
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <form action="valida-envio.php" method="get">
    <!-- <form 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 /
Fuente: Elaboración Propia

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	<code>and</code>	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	<code>or</code>	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	<code>xor</code>	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)	<code>&&</code>	Exactamente igual que <code>and</code>	<code>\$n1 != \$n2 && \$n1 != \$n3</code>
OR (2)	<code> </code>	Exactamente igual que <code>or</code>	<code>\$n1 == \$n2 \$n1 == \$n3</code>
Negación	<code>!</code>	Cambia el valor del elemento al que se le aplica es decir, si <code>\$n1</code> 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 solo 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 / Fuente: Elaboración Propia

Para la configuración de una sesión se utilizarán las que se conoce como variables de sesión. Son unos componentes que permiten la configuración de estas sesiones, que, de forma predeterminada, durarán hasta que el usuario cierra el navegador, o hasta el día siguiente, a partir de las 00:00 horas. Es importante remarcar que estas variables de sesión disponen de información sobre solo un usuario y en ese sentido, están disponibles 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
/ Fuente: Elaboración Propia

```
<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
/ Fuente: Elaboración Propia

PRESENTACIÓN DE LA BASE DE DATOS

Una vez se obtuvieron las primeras respuestas del formulario, recuperando la idea de análisis de la situación expresada al 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 permitía 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.

The screenshot shows a Google Form titled "Preferencias en vacaciones". It includes a greeting "Hola," and a note: "Esta es una encuesta para un trabajo académico. Os pedimos que de manera anónima contestéis a las preguntas que se formulan a continuación." The form has several sections: "Código Postal (CP)" with a text input field, "Fecha de nacimiento" with a date picker, and a "Texto de respuesta corto" field.

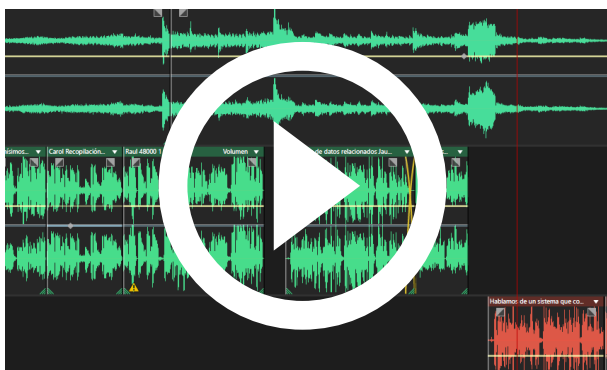
Formulario para la creación de la bases de datos. / Fuente: Elaboración propia

Marca temporal	Código P	Fecha de nacimiento	Gasto en vacaciones de	Ingresos netos anuales	Preferencia de tipo de tui	Preferencia de destino	¿Gastas más dinero en vacac
7/4/2022 12:14:50	17300	21/2/1999	Poco (100-300)	6000	Ciudades	Comunitario (UE)	Intento hacer turismo en époc
7/4/2022 12:20:34	08830	9/1/1975	Poco (100-300)	4900	De playa	Local	
7/4/2022 12:27:27	08905	6/5/1972	Bastante (600-1200)	70000	De playa	Nacional	
7/4/2022 12:27:44	08630	11/6/1972	Normal (300-600)	50000	De montaña / excursione	Nacional	NO
7/4/2022 12:30:18	08759	23/3/1974	Poco (100-300)	40099	Segundas residencias pa	Nacional	500
7/4/2022 12:31:08	08940	9/11/2000	Poco (100-300)	6.600	De montaña / excursione	Nacional	No
7/4/2022 12:33:33	08630	15/8/1974	Normal (300-600)	30000	De playa	Comunitario (UE)	No
7/4/2022 12:35:20	08980	17/1/1977	Mucho (más de 1200)	60000	Ciudades	Comunitario (UE)	NO
7/4/2022 12:59:08	08860	20/5/1972	Normal (300-600)	9000	De montaña / excursione	Local	
7/4/2022 13:23:55	08850	26/4/1972	Normal (300-600)	29000	De ciudad / urbano	Extranjero (UK, EUA, y o	No
7/4/2022 13:38:05	08850	5/1/1970	Normal (300-600)	15000	Segundas residencias pa	Nacional	No
7/4/2022 14:30:57	08860	4/7/2022	Mucho (más de 1200)	1000	De playa	Extranjero (UK, EUA, y otros	terceros países)
7/4/2022 14:37:50	08755	7/9/1974	Normal (300-600)	1500	De playa	Comunitario (UE)	
7/4/2022 14:40:12	08737	1/5/1986	Normal (300-600)	Puede variar de 12.000 a	De playa	Local	No
7/4/2022 15:40:31	08905	24/11/1972	Mucho (más de 1200)	18000	De montaña / excursione	Comunitario (UE)	No
7/4/2022 15:45:51	08755	1/7/1972	Mucho (más de 1200)	35000	De playa	Nacional	Sí. Suelo salir más
7/4/2022 16:10:54	08905	8/9/1972	Poco (100-300)	40000	De playa	Nacional	

Base de datos generada / Fuente: Elaboración Propia

¿BASE DE DATOS?

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.



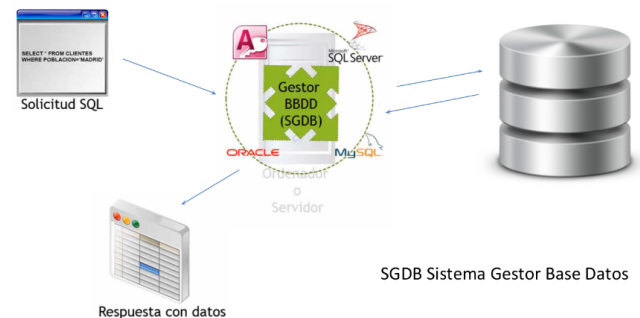
Archivo de audio sobre las bases de datos. / Fuente: Elaboración propia

EL LENGUAJE SQL

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 TRUNCATE DROP ALTER	SELECT DELETE UPDATE INSERT	GRANT REVOKE	COMMIT ROLLBACK SAVEPOINT



COMANDO* + CLÁUSULAS* + OPERADORES + FUNCIONES

FROM + WHERE + GROUP BY + HAVING + ORDER BY

Partimos de la base de datos generada a través de Google Forms 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:

Si quisiéramos visualizar cuánta gente prefiere la playa, ordenado por código postal.

La consulta sería: **SELECT * FROM `Respuestas de formulario 1`**

WHERE `Preferencia de tipo de turismo` ="De playa" ORDER BY "Código Postal (CP)"

Marca temporal	Código	Fecha de nac	Gasto en vacaciones	Ingresos	Preferencia de tipo	Preferencia de destino	¿Gastas más
04/07/2022 16:10:54	08905	08/09/1972	Poco (100-300)	40000	De playa	Nacional	
04/07/2022 15:45:51	08755	01/07/1972	Mucho (más de 1200)	35000	De playa	Nacional	Sí. Suelo salir m
04/07/2022 12:33:33	08630	15/08/1974	Normal (300-600)	30000	De playa	Comunitario (U	No
04/07/2022 14:37:50	08755	07/09/1974	Normal (300-600)	1500	De playa	Comunitario (U	
04/07/2022 14:30:57	08860	04/07/2022	Mucho (más de 1200)	1000	De playa	Extranjero (UK,	
04/07/2022 12:33:33	08630	15/08/1974	Normal (300-600)	30000	De playa	Comunitario (U	No
04/07/2022 12:27:27	08905	06/05/1972	Bastante (600-1200)	70000	De playa	Nacional	
04/07/2022 12:20:34	08830	09/01/1975	Poco (100-300)	4900	De playa	Local	

Queremos lanzar un promoción de viaje de playa, para un perfil que viva en Barcelona o alrededores y cuyo presupuesto anual sea superior es (600-1200).

La consulta sería: **SELECT * FROM `Respuestas de formulario 1`**

WHERE `Gasto en vacaciones de verano (ocio) por persona` ="Bastante (600-1200)"

Código Postal (CP)	Fecha de naci	Gasto en vacaciones	Ingresos netos anua	Preferencia de tipo	Preferencia de dest
38905	6/5/1972	Bastante (600-1200)	70000	De playa	Nacional

Las consultas que lancemos a la base de datos nos van a permitir obtener información, que podremos transformar en conocimiento y a partir de ahí tomar las decisiones más adecuadas para nuestro negocio.

CONCLUSIONES

UNA REFLEXIÓN FINAL

Tal y como se ha explicado a lo largo del documento y como se resuelve evidente, el conocimiento sobre lo que es una base de datos no es solo recomendable sino fundamental. Para nosotros y el teórico proyecto empresarial que ha orientado parte del trabajo, la base de datos es fundamental y la vía de acceso a esta información es igual de importante. Mientras se amplían las horas que las personas dedican a las plataformas o redes sociales, sus dueños y administradores alimentan sus modelos de negocio basándose en a las infinitas extensiones de datos que no solo tienen un impacto en económico sino también social y político.

Periodistas, analistas, sociológicos, politólogos y estadísticos han corroborado una vez detrás de otra la utilización de perfiles generados en bases de datos para dirigir campañas de marketing político. En ese sentido, la sensibilización con el lenguaje de programación SQL que nos permite dirigir consultas es un potencial que habilita al usuario presentarse más consciente cuando es el target de cualquier campaña dirigida.

Además, es fundamental conocer el mecanismo y la automatización que existe detrás de una solicitud de un cliente o usuario para conocer una determinada información. Esta consulta, puede producirse después de la interacción con un formulario. De hecho, es una de las funciones de estos aplicativos que están programados a través del lenguaje PHP.

Ser conscientes del mecanismo que hay detrás del botón de "envío" es un elemento clave que apodera al usuario y que le permite conocer no solo a quien enviará la información y en que condiciones, sino también que tecnologías, y que funciones van a procesar su información y con que mecanismos acabarán llegando a los destinatarios. Es tan importante una base de datos para una administración pública, como también lo es para una empresa. La clave de una base de datos no son, por lo tanto, los datos, sino la capacidad que aportan sus relaciones. Y la explotación de estas vinculaciones son la explicación del éxito de aquellos/as que han sabido detectarlo.



Belén Serrano / Jaume Lozano
Castelldefels, 2022.