

2. Recogida de datos por teclado mediante formularios

2.1 Recogida de datos en dos pasos

En una página web, los datos se introducen mediante formularios. En la mayoría de las ocasiones tendremos dos páginas: una página con un formulario que recoge los datos y otra página que los procesa.

A continuación tenemos una página con nombre `index.php` (se podría llamar también `index.html` ya que no contiene código PHP):

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
  </head>
  <body>
    Introduce tu nombre:
    <form action="saluda.php" method="get">
      <input type="text" name="nombre"><br>
      <input type="submit" value="Enviar">
    </form>
  </body>
</html>
```

Al pulsar el botón **Enviar**, el contenido del formulario se envía a la página que indicamos en el atributo `action`. La página `saluda.php` tendría el siguiente código:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
  </head>
  <body>
    Hola <?php echo $_GET['nombre'] ?>,
    que tengas un buen día.
  </body>
</html>
```

Observa que la información que se quiere enviar se introduce en un cuadro de texto dentro del formulario (página `index.php`). A esa información la llamamos en este caso `nombre`. Para recoger esa información se utiliza `$_GET['nombre']`.

Cuando los datos que se recogen y se manipulan son números en lugar de cadenas de caracteres, el procedimiento es el mismo.

En el siguiente ejemplo tenemos una aplicación que suma dos números. El programa `index.php` recoge los datos y el programa `suma.php` suma los dos números que recibe y muestra el resultado.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
  </head>
  <body>
    <h3>Calcula la suma de dos números</h3>
    <form action="suma.php" method="get">
      Primer número: <input type="number" name="a"><br>
      Segundo número: <input type="number" name="b"><br>
      <input type="submit" value="Sumar">
    </form>
  </body>
</html>
```

Fíjate que los campos `input` del formulario son de tipo `number`. Se podrían declarar de tipo `text` y el programa funcionaría sin problemas pero se declaran `number` para que el formulario no deje pasar los datos si se introduce algo que no sea un número. Imagínate que el usuario introduce la palabra `hola` en un campo en lugar de un número. Si el `input` es de tipo `text`, la palabra `hola` llegaría a `suma.php` y se intentaría sumar, lo que provocaría un error. Sin embargo, si el `input` es de tipo `number`, el formulario no envía `hola` sino que muestra un mensaje diciéndole al usuario que introduzca un número.

En la medida de lo posible, intenta utilizar siempre el tipo adecuado en los campos `input`; por ejemplo `number`, `color`, `date`, `email`, `url`, etc. de tal forma que los datos que envía el formulario tengan el formato correcto.

A continuación se muestra el fichero `suma.php`.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
  </head>
  <body>
    <?php
      $a = $_GET['a'];
      $b = $_GET['b'];
      echo "La suma de $a mas $b es ", $a + $b;
    ?>
  </body>
</html>
```

Como hemos visto anteriormente, en PHP no hace falta declarar las variables especificando el tipo. Una misma variable puede contener incluso valores de distintos tipos durante su vida útil. Esto hace que no necesitemos ninguna conversión como habría que hacer utilizando otros lenguajes.

2.2 Métodos `get` y `post`

Mediante el atributo `method` de la etiqueta `form` se debe especificar un método de envío; los dos métodos posibles son `get` y `post`. El resultado final es el mismo, la única diferencia radica en que con el método `get` se pueden ver los parámetros que envía el formulario en la barra de direcciones del navegador.