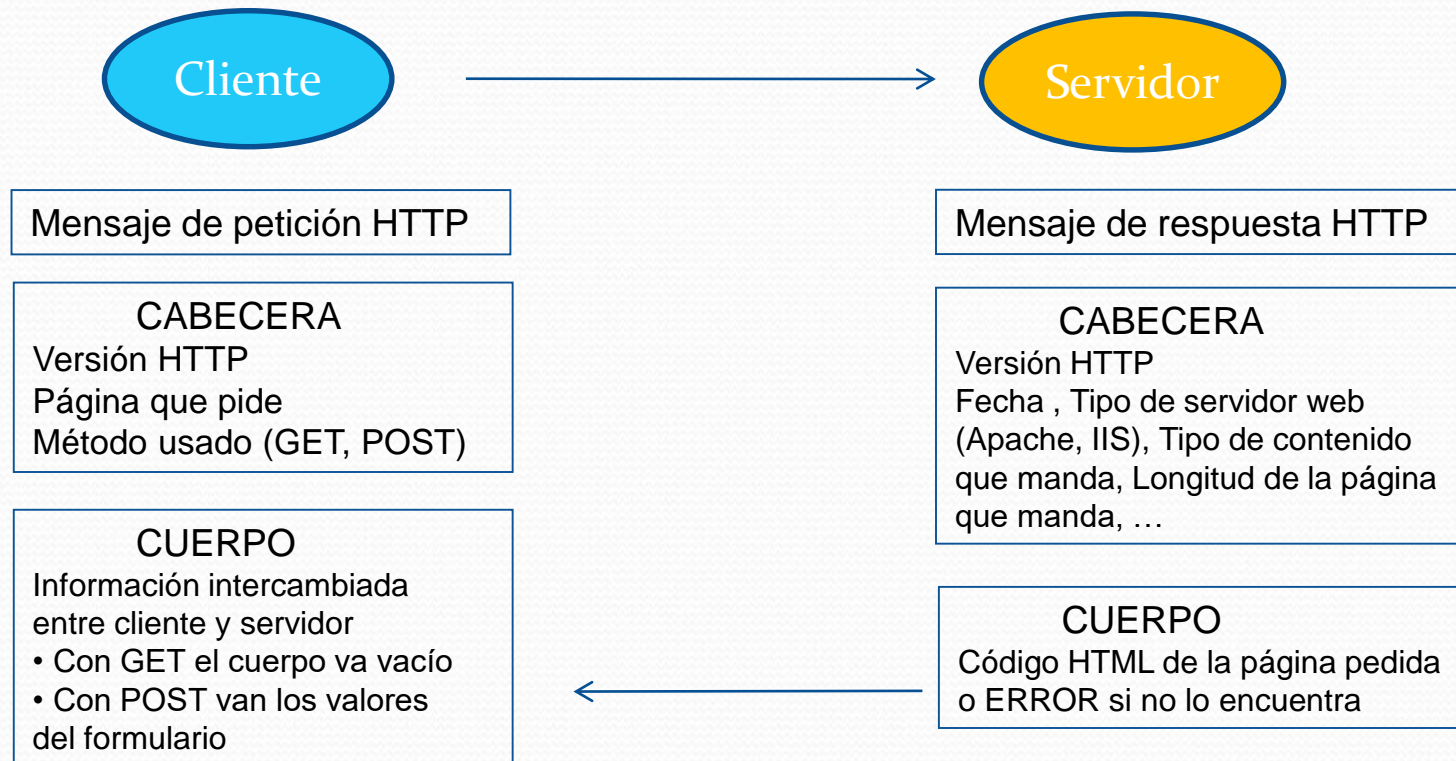




U.T. 3: Formularios.

FORMULARIOS: Protocolo HTTP

- HTTP es un protocolo sin estado. El cliente inicia la comunicación:



FORMULARIOS: Protocolo HTTP

Ejemplo de mensaje de respuesta HTTP

Servidor

```
HTTP/1.1 200 OK
Date: Thu, 01 Jan 2001 06:32:56 GMT
Server: Apache/1.3.14 (Unix) PHP/3.0.14
Last-Modified: Tue, 04 Apr 2000 13:54:53 GMT
Content-Length: 465
Connection: close
Content-Type: text/html
```

CABECERA del
mensaje

Línea vacía

```
<HTML>
<HEAD><TITLE>Dpto. O.E.I. - U.P.M.</TITLE>
</HEAD>
<BODY BGCOLOR="#000000" VLINK="#FFFFFF">
...
```

CUERPO del
mensaje

FORMULARIOS: Acceso desde PHP

- ❑ Los Formularios **no** forman parte de PHP, sino del lenguaje estándar de Internet, **HTML**.
 - ❑ Puesto que se utiliza el protocolo HTTP, estamos limitados por su interfaz: sólo se puede utilizar alguno de los comandos del protocolo para establecer la comunicación:
 - ❑ Se utilizan dos comandos del protocolo: GET o POST
 - ❑ Dos tipos diferentes de peticiones, según atributo **method** del **<FORM>**:
 - ❑ Peticiones GET (método GET de HTTP)
 - ❑ Peticiones POST (método POST de HTTP)
- `<FORM ACTION="lo_que_sea.php" METHOD="post/get">`
- ❑ Al pulsar el botón de envío el navegador construye la petición adecuada
-

FORMULARIOS: Acceso desde PHP

☐ Peticiones GET (método GET de HTTP):

- ☐ Los parámetros se le añaden a la URL tras un signo de “?” y se concatenan con &.
- ☐ En el servidor, los valores se guardan en el array asociativo `$_GET`.

`http://site/procesa.php?name1=value1&name2=value2&name3=value3`

Reglas de codificación URL

- ☐ RFC 1738
 - ☐ Los caracteres especiales se traducen:
 - ☐ Espacios en blanco se traducen a “+”, y después a %2B
 - ☐ La # se traduce por %23
 - ☐ Los caracteres especiales se envían con el formato %NN (NN: valor hexadecimal de carácter).
 - ☐ **Son caracteres especiales:**
 - ☐ Ñ,ñ,á,.. (no tienen un carácter US ASCII asociado)
 - ☐ Los peligrosos:<,>,",#
 - ☐ Los reservados con significado especial: /,@,?,&
-

FORMULARIOS: Acceso desde PHP

☐ Peticiones **POST** (método POST de HTTP)

- ☐ Los parámetros se envían en el **cuerpo del mensaje no en la** cadena de solicitud (query string).
 - ☐ En el servidor, los valores se guardan en el array asociativo **\$_POST**.
 - ☐ Como con GET, los caracteres especiales se traducen a ASCII.
 - ☐ Es necesario indicar el tipo de codificación en el <form> con el atributo **enctype**
 - ☐ **application/x-www-form-urlencoded** (Por defecto).
 - ☐ NO PERMITE ENVIAR ARCHIVOS
 - ☐ **multipart/form-data**.
 - ☐ PERMITE ENVIAR ARCHIVOS
-

FORMULARIOS: Acceso desde PHP

- ❑ GET vs POST
 - ❑ Problemas GET
 - ❑ No se puede enviar información binaria (archivos, imágenes, etc.) => necesario el método POST.
 - ❑ Los datos se ven en la URL del navegador.
 - ❑ Problemas POST
 - ❑ Rompe la funcionalidad del botón “Atrás” del navegador
 - ❑ El botón actualizar repite la operación
 - ❑ Principios generales
 - ❑ GET implica “obtener” información.
 - ❑ POST implica “realizar” una acción con un “efecto secundario”.
 - ❑ Mejor POST para procesar formularios
-

FORMULARIOS: Acceso desde PHP

Por tanto:

- ❑ Acceso a los valores introducidos en los formularios a través de arrays globales:
 - ❑ **`$_GET`**: parámetros enviados mediante GET o en la URL
 - ❑ **`$_POST`**: parámetros enviados mediante POST
 - ❑ **`$_REQUEST`**: la unión de `$_GET` y `$_POST` (también `$_COOKIES`)
-

FORMULARIOS: Acceso desde PHP

❑ Desde PHP se puede acceder a los datos de un formulario HTML.

❑ Fichero **uno.html**:

```
<html><body>
<form action="dos.php" method="POST">
  Edad: <input type="text" name="edad">
  <input type="submit" value="aceptar">
</form>
</body></html>
```

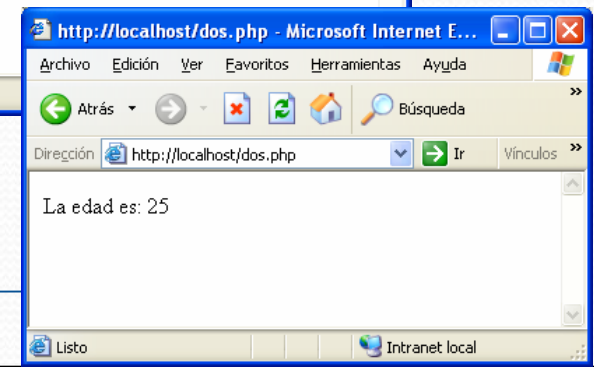
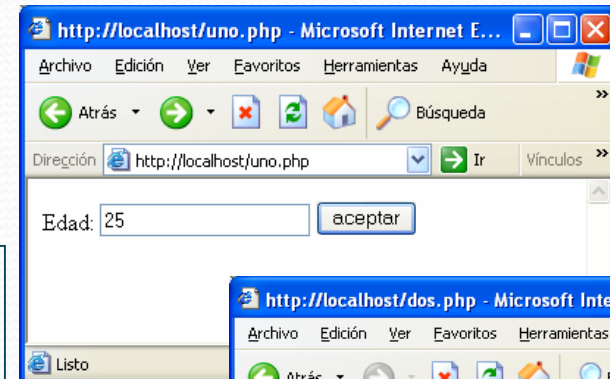
Página a la que se llamará al hacer click en submit

Estos valores deben ser iguales. Si no, no podremos recuperar la información.

❑ Fichero **dos.php**

```
<?php
echo "La edad es:". $_REQUEST['edad'];
?>
```

Podría ser \$_POST



FORMULARIOS: Acceso desde PHP

- ❑ Acceso a los diferentes tipos de elementos de entrada de formulario
 - ❑ Elementos de tipo **INPUT**
 - ❑ TEXT
 - ❑ RADIO
 - ❑ CHECKBOX
 - ❑ BUTTON
 - ❑ FILE
 - ❑ HIDDEN
 - ❑ PASSWORD
 - ❑ SUBMIT / RESET
 - ❑ Elemento **SELECT**
 - ❑ Simple / múltiple
 - ❑ Elemento **TEXTAREA**
-

FORMULARIOS: Acceso desde PHP

□ TEXT

Introduzca la cadena a buscar:

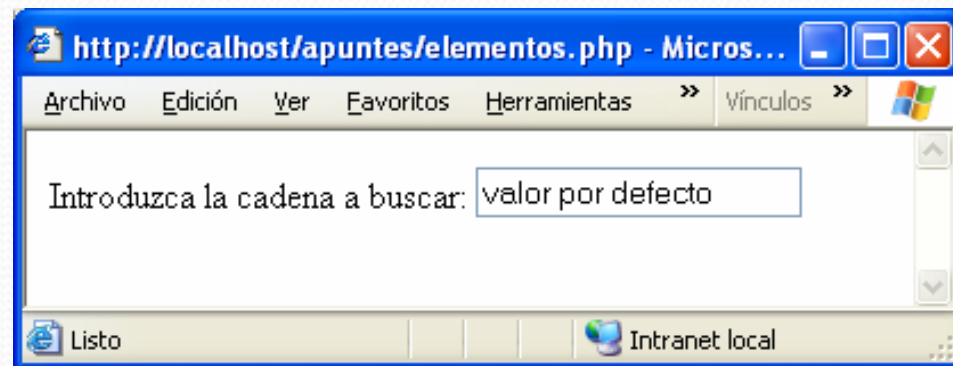
```
<INPUT TYPE="text" NAME="cadena" VALUE="valor por defecto" SIZE="20">
```

```
<?PHP
```

```
$cadena = $_REQUEST['cadena'];
```

```
echo $cadena;
```

```
?>
```



FORMULARIOS: Acceso desde PHP

☐ RADIO

Sexo:

```
<INPUT TYPE="radio" NAME="sexo" VALUE="M" CHECKED>Mujer
```

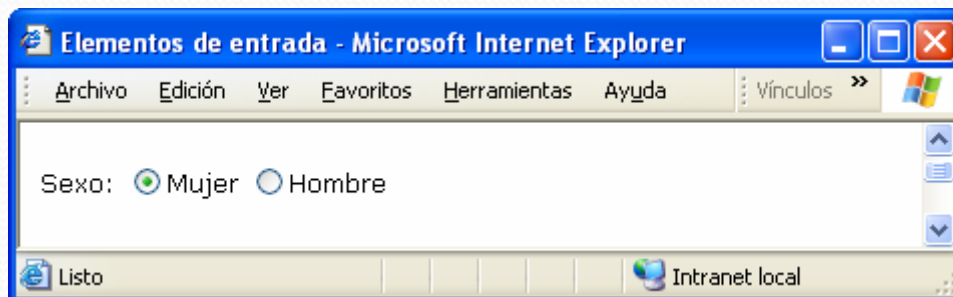
```
<INPUT TYPE="radio" NAME="sexo" VALUE="H">Hombre
```

```
<?PHP
```

```
    $sexo = $_REQUEST['sexo'];
```

```
    echo ($sexo);
```

```
?>
```



- ☐ Los botones radio se llaman igual para que si se elije uno se desmarquen los otros.
-

Formularios. Controles

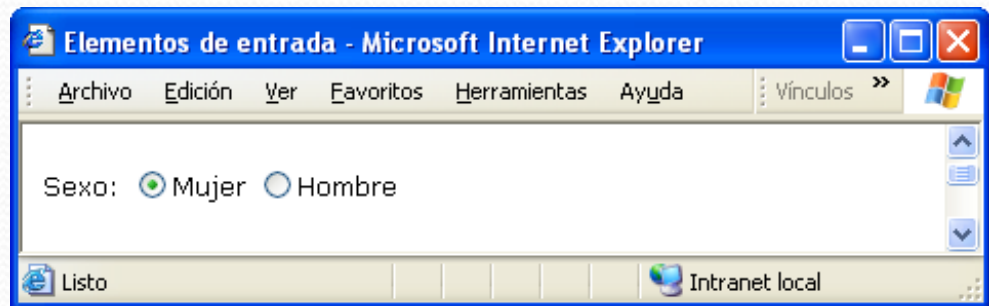
- **RADIO**

Si no se marca no envía nada.

Si está marcado envía:

- ON si no tiene value
- El value si lo tiene asignado

.



Formularios. Controles

- **CHECKBOX**

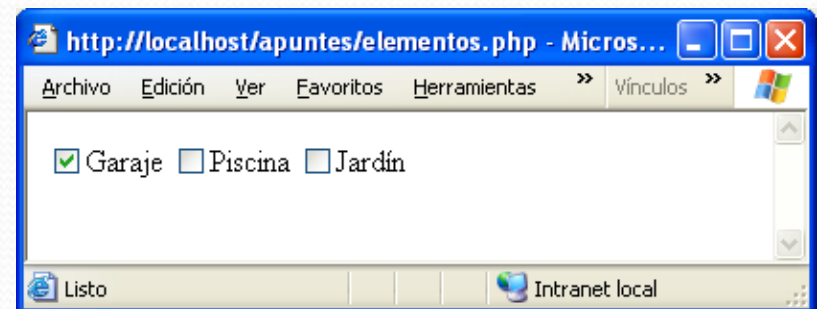
Si no se marca no envía nada.

Si marcado envía:

- ON sino tiene value
- El value si lo tiene asignado

Dos formas de manejarlos:

- Nombre diferentes a cada opción
- Array de opciones



Formularios. Controles

Extras:

```
<INPUT TYPE="checkbox" NAME="garaje" CHECKED>Garaje  
<INPUT TYPE="checkbox" NAME="piscina" >Piscina  
<INPUT TYPE="checkbox" NAME="jardin" >Jardín  
<br><br>
```

```
<?php  
echo "<br><br>";  
if($_REQUEST["garaje"]== "on");  
echo "garaje <br>";  
if ($_REQUEST["piscina"]== "on");  
echo "piscina <br>";  
if ($_REQUEST["jardin"]== "on");  
echo "jardin <br>";  
?>
```

FORMULARIOS: Acceso desde PHP

☐ HIDDEN

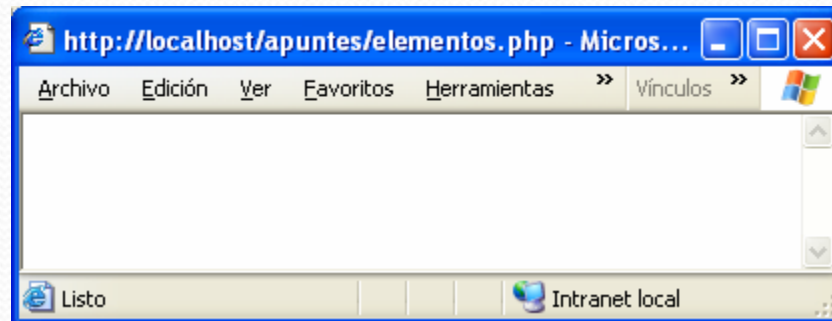
```
<INPUT TYPE='hidden' NAME='username' VALUE="$usuario">
```

```
<?PHP
```

```
    $username = $_REQUEST['username'];
```

```
    echo $username;
```

```
?>
```



FORMULARIOS: Acceso desde PHP

☐ SELECT Simple

Color:

```
<SELECT NAME="color">  
<OPTION VALUE="rojo" SELECTED>Rojo</OPTION>  
<OPTION VALUE="verde">Verde</OPTION>  
<OPTION VALUE="azul">Azul</OPTION>  
</SELECT>
```

```
<?PHP
```

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

```
    echo $color;
```

```
?>
```



FORMULARIOS: Acceso desde PHP

☐ SELECT Múltiple

☐ Valores vectoriales de un formulario:

- ☐ Select múltiples o botones de comprobación con el mismo nombre
- ☐ Si no se indica nada, sólo se tiene acceso a un valor
- ☐ En el código HTML hay que añadir “[]” al nombre del control
- ☐ Devuelve un array, con count() podemos conocer su tamaño

Idiomas:

```
<SELECT MULTIPLE SIZE="3" NAME="idiomas[]">
<OPTION VALUE="ingles" SELECTED>Inglés</OPTION>
<OPTION VALUE="frances">Francés</OPTION>
<OPTION VALUE="aleman">Alemán</OPTION>
<OPTION VALUE="holandes">Holandés</OPTION>
</SELECT>
```

```
<?PHP
```

```
$idiomas = $_REQUEST['idiomas'];
foreach ($idiomas as $idioma)
    echo "$idioma<BR>\n";?>
```



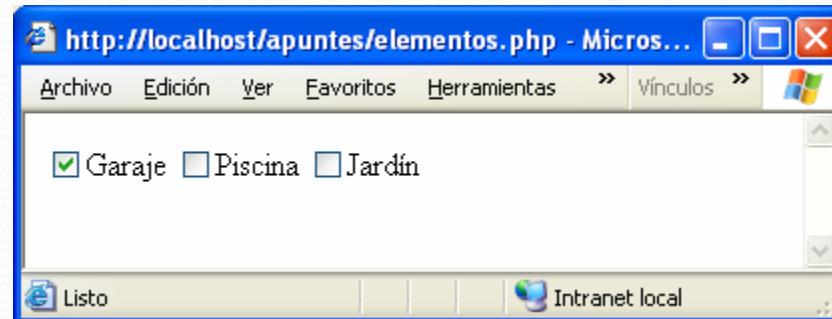
FORMULARIOS: Acceso desde PHP

☐ CHECKBOX

- ☐ botones de comprobación con el mismo nombre → usar **arrays []**

```
<INPUT TYPE="checkbox" NAME="extras[]" VALUE="garaje" CHECKED>Garaje  
<INPUT TYPE="checkbox" NAME="extras[]" VALUE="piscina">Piscina  
<INPUT TYPE="checkbox" NAME="extras[]" VALUE="jardin">Jardín
```

```
<?PHP  
$extras = $_REQUEST['extras'];  
foreach ($extras as $extra)  
    echo "$extra<BR>\n";  
?>
```



FORMULARIOS: Acceso desde PHP

□ TEXTAREA

Comentario:

```
<TEXTAREA COLS="50" ROWS="4" NAME="comentario">
```

Este libro me parece ...

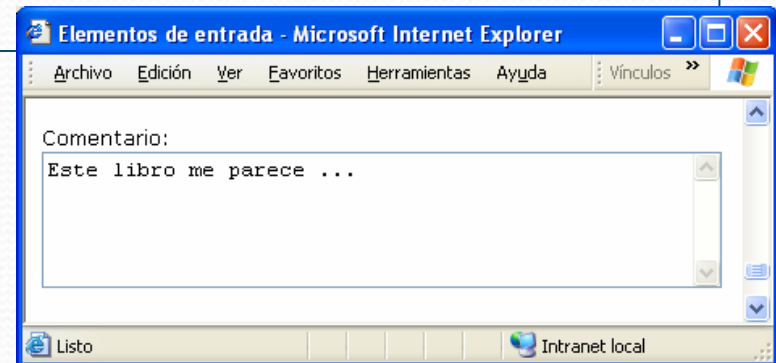
```
</TEXTAREA>
```

```
<?PHP
```

```
    $comentario = $_REQUEST['comentario'];
```

```
    echo $comentario;
```

```
?>
```

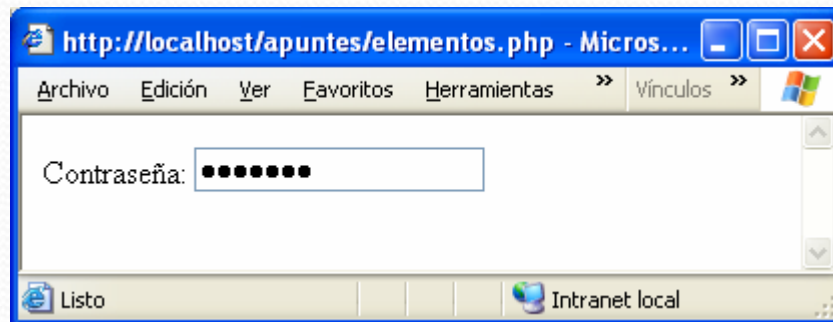


FORMULARIOS: Acceso desde PHP

❑ PASSWORD

Contraseña: <INPUT TYPE="password" NAME="clave">

```
<?PHP
$clave = $_REQUEST['clave'];
echo $clave;
?>
```



FORMULARIOS: Acceso desde PHP

☐ SUBMIT

```
<INPUT TYPE=submit NAME="enviar" VALUE="Enviar datos">
```

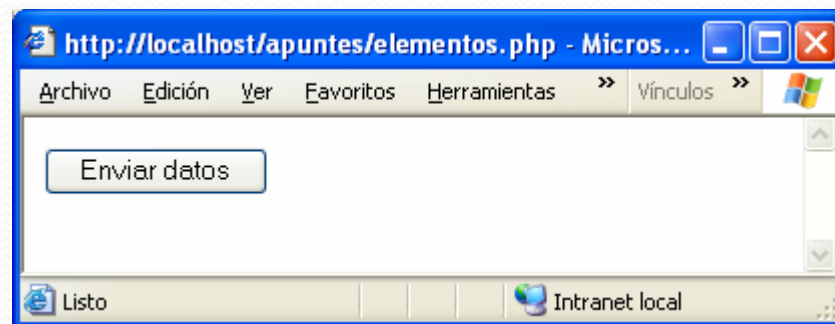
```
<?PHP
```

```
$enviar = $_REQUEST['enviar'];
```

```
if ($enviar)
```

```
    echo "Se ha pulsado el botón de enviar";
```

```
?>
```



FORMULARIOS: Acceso desde PHP

❑ RESET

```
<INPUT TYPE=reset NAME="borrar" VALUE="Limpiar datos">
```

```
<?PHP
```

```
$borrar = $_REQUEST['borrar'];
```

```
if ($borrar)
```

```
    echo "Se ha pulsado el botón de limpiar datos";
```

```
?>
```

