

<b>4. Xeración dinámica de páxinas web .....</b>	<b>2</b>
4.1 Xestión de formularios mediante a linguaxe PHP .....	2
Os formularios.....	2
Botón enviar .....	3
Caixa de texto, caixa de contrasinal e área de texto.....	4
Casiña de verificación.....	4
Botón de radio .....	5
Menú .....	5
Control oculto .....	5
Selector de arquivos .....	6
Imaxe .....	7
Verificación de formularios con PHP .....	8
Existencia dos controis .....	8
Seguridade nas entradas de datos .....	9
Saída de datos .....	11
Validación dos datos dos formularios con PHP .....	13

## 4. Xeración dinámica de páxinas web

### 4.1 Xestión de formularios mediante a linguaxe PHP

#### Os formularios

Un **formulario** é un conxunto de controis (botóns, caixas de texto, casillas de verificación etc) que **permiten ao usuario introducir datos e envialos ao servidor web para o seu procesamento**.

#### Estrutura do formulario

A **etiqueta** que delimita un formulario é a etiqueta `<form> ... </form>`. Os atributos máis importantes desta etiqueta son:

- **action**: contén o nome do axente que procesará os datos remitidos ao servidor (por exemplo, un script PHP). O axente especificarase mediante unha dirección absoluta ou relativa do ficheiro que procesará a información.
- **method**: define a maneira de enviar os datos ao servidor. Os posibles valores poden ser:
  - **get**: os valores enviados engádense á dirección indicada no atributo action.
  - **post**: os valores envíanse de forma separada.

Tanto get como post estudaranse na seguinte actividade sobre a persistencia da información entre distintos documentos web.

#### Controis

A etiqueta `<form>` pode conter no seu interior calquera elemento típico de páxina web (parágrafos, imaxes, DIVs, listas, táboas, etc), ademais das etiquetas das etiquetas que crean os controis. As etiquetas que crean os **controis nos formularios** son: `<input/>`, `<button/>`, `<select>`, `<optgroup>`, `<option>` e `<textarea>`. Ademais, pódense **estruturar** os **controis** coas **etiquetas** `<fieldset>` e `<legend>`. A etiqueta `<label>` permite mellorar a accesibilidade aos controis. O navegador envía unicamente os datos dos controis contidos no formulario.

Os **atributos comúns dos controis** son: name,value, disabled, readonly, tabindex e accesskey:

#### Atributos

- **name**: identifica ao control. O formulario envía ao servidor os datos dos controis que teñen establecido o atributo name, polo tanto, si un control non ten establecido o seu atributo name, simplemente ignórase.
- **value**: permite establecer o valor inicial dun control. Cada control o utiliza dunha maneira diferente. O único control sen atributo value é `<textarea>`.
- **disabled**: permite deshabilitar o control. Unha vez deshabilitado, o control no pode coller o foco.
- **readonly**: permite que o control non sexa modificable, aínda que o control pode coller o foco.
- **tabindex**: permite controlar a orde na que o foco pasa dun elemento a outro mediante o tabulador. Os valores de tabindex serán números naturais (cero incluído), non necesariamente consecutivos. Os controles visitaranse en orde de menor a maior número.

- **accesskey**: permite definir teclas de acceso (atallos de teclado). Accederanse coa combinación de teclas alt+shift+tecla. As veces poden entrar en conflito con combinacións de teclas xa definidas polo navegador ou o sistema operativo.

Exemplo de formulario	
<pre>&lt;form action="exemplo.php"&gt; &lt;p&gt;Nome: &lt;input type="text" name="nome" /&gt;&lt;/p&gt; &lt;p&gt;&lt;input type="submit" value="Enviar" /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<p>Nome: <input type="text" value="María"/></p> <p><input type="submit" value="Enviar"/></p>

**\$\_REQUEST**

O programa que recibe os datos os garda automaticamente nunha **matriz \$\_REQUEST**. Mediante a chamada á **función print\_r(\$\_REQUEST)** poderase mostrar o contido da devandita matriz. A función print\_r mostra información sobre unha variable de forma lexible para os humanos.

O seguinte exemplo mostra o que escribiría o programa PHP exemplo.php se recibise a información do formulario do exemplo anterior. Compre ter en conta que, para que se mostren as eñes e as tildes, a documento web que contén o formulario debe ter a seguinte etiqueta: `<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />`, e o ficheiro exemplo.php usará a instrución `header('Content-Type: text/html; charset=UTF-8')` encargada de enviar un encabezado HTTP onde se especificará que usará a codificación de caracteres UTF-8 (salvo que exemplo.php contivese un documento HTML completo co cal usárase a etiqueta meta tamén, pero non é o caso).

Facer exemplo

Exemplo de visualización do contido de \$_REQUEST	
<pre>&lt;?php header('Content-Type: text/html; charset=UTF-8'); print_r(\$_REQUEST); echo "&lt;br/&gt;O valor recibido do campo nome é: ".\$_REQUEST["nome"]; ?&gt;</pre>	<p>Array ( [nome] =&gt; María ) O valor recibido do campo nome é: María</p>

Seguidamente explicarase o uso dos diferentes controis, indicando o código HTML do control no formulario, o código do script PHP do arquivo que procesa os datos, e o resultado de mostrar tanto o valor da matriz \$\_REQUEST como unha sentencia echo onde se acceda ao valor recibido do control.

### Botón enviar

**type="submit"**

Este control pódese crear coa etiqueta `<input>` ou coa etiqueta `<button>`. En ámbolos casos o valor enviado é o contido do atributo value, ou ben o contido da etiqueta no caso de `<input>`.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt; &lt;p&gt;&lt;input type="submit" value="Submit" /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php header('Content-Type: text/html; charset=UTF-8'); print_r(\$_REQUEST); ?&gt;</pre>	Array ( )
<pre>&lt;form action="exemplo.php"&gt; &lt;p&gt;&lt;input type="submit" value="Enviar" name="boton2" /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php header('Content-Type: text/html; charset=UTF-8'); print_r(\$_REQUEST); echo "&lt;br/&gt;O valor recibido do control boton2 é: ".\$_REQUEST["boton2"]; ?&gt;</pre>	Array ( [boton2] => Enviar ) O valor recibido do control boton2 é: Enviar
<pre>&lt;form action="exemplo.php"&gt; &lt;p&gt;&lt;button type="submit"&gt;Submit&lt;/button&gt;&lt;/p&gt;</pre>	<pre>&lt;?php header('Content-Type: text/html; charset=UTF-8');</pre>	Array ( )

</form>	print_r(\$_REQUEST); ?>	
<form action="exemplo.php"> <button type="submit" name="boton3" value="enviado">Enviar</button> </form>	<?php header('Content-Type: text/html; charset=UTF-8'); print_r(\$_REQUEST); echo " O valor recibido do control boton3 é: " . \$_REQUEST["boton3"]; ?>	Array ( [boton3] => enviado ) O valor recibido do control boton3 é: enviado
<form action="exemplo.php"> <button type="submit" name="boton4" value="enviado">Enviar</button> </form>	<?php header('Content-Type: text/html; charset=UTF-8'); print_r(\$_REQUEST); echo " O valor recibido do control boton4 é: " . \$_REQUEST["boton4"]; ?>	Array ( [boton4] => enviado ) O valor recibido do control boton4 é: enviado

## Caixa de texto, caixa de contrasinal e área de texto

Este control envíase sempre, e o valor enviado é o contido da caixa.

type="text"  
"password"  
"textarea"

Código do control no formulario	Código de exemplo.php	Información recibida
<form action="exemplo.php"> <p><input type="text" name="caixaTexto1" /></p> <p><input type="text" name="caixaTexto2" value="Texto e elixir" /></p> <p><input type="password" name="caixaPassword1" /></p> <p><input type="password" name="caixaPassword2" value="ollomol" /></p> <p><textarea rows="4" cols="20" name="areaDeTexto1"></textarea></p> <p><textarea rows="4" cols="20" name="areaDeTexto2">Calquera texto</textarea></p> <p><input type="submit" value="Submit" /></p> </form>	<?php header('Content-Type: text/html; charset=UTF-8'); print_r(\$_REQUEST); echo " O valor recibido do control caixaTexto1 é: " . \$_REQUEST["caixaTexto1"]; echo " O valor recibido do control caixaTexto2 é: " . \$_REQUEST["caixaTexto2"]; echo " O valor recibido do control caixaPassword1 é: " . \$_REQUEST["caixaPassword1"]; echo " O valor recibido do control caixaPassword2 é: " . \$_REQUEST["caixaPassword2"]; echo " O valor recibido do control areaDeTexto1 é: " . \$_REQUEST["areaDeTexto1"]; echo " O valor recibido do control areaDeTexto2 é: " . \$_REQUEST["areaDeTexto2"]; ?>	Array ( [caixaTexto1] => Rúa do Orzán 10 [caixaTexto2] => Texto e elixir [caixaPassword1] => kungfu [caixaPassword2] => ollomol [areaDeTexto1] => Primeira liña do texto [areaDeTexto2] => Calquera texto ) O valor recibido do control caixaTexto1 é: Rúa do Orzán 10 O valor recibido do control caixaTexto2 é: Texto e elixir O valor recibido do control caixaPassword1 é: kungfu O valor recibido do control caixaPassword2 é: ollomol O valor recibido do control areaDeTexto1 é: Primeira liña do texto O valor recibido do control areaDeTexto2 é: Calquera texto

## Casiña de verificación

type="checkbox"

Este control envíase soamente cando se marca a casiña. O valor enviado é “on” se a casiña non ten definido o atributo value, ou o valor do devandito atributo se está definido.

Código do control no formulario	Código de exemplo.php	Información recibida
<form action="exemplo.php"> <p><input type="checkbox" name="casiña1" /></p> <p><input type="checkbox" name="casiña2" /></p> <p><input type="checkbox" name="casiña3" value="Tres" /></p> <p><input type="submit" value="Submit" /></p> </form>	<?php header('Content-Type: text/html; charset=UTF-8'); print_r(\$_REQUEST); echo " O valor recibido do control casiña2 marcada é: " . \$_REQUEST["casiña2"]; echo " O valor recibido do control casiña3 marcada é: " . \$_REQUEST["casiña3"]; ?>	Array ( [casiña2] => on [casiña3] => Tres ) O valor recibido do control casiña2 marcada é: on O valor recibido do control casiña3 marcada é: Tres

## Botón de radio

type="radio"

Este control envíase soamente cando se marca algún dos botóns de radio que forman o control. O valor enviado é “on” se o botón non ten definido o atributo value, ou o valor do atributo value se este está definido.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt;   &lt;p&gt;&lt;input type="radio" name="radio1" /&gt;   &lt;input type="radio" name="radio1" /&gt;&lt;/p&gt;   &lt;p&gt;&lt;input type="radio" name="radio2" /&gt;   &lt;input type="radio" name="radio2" /&gt;&lt;/p&gt;   &lt;p&gt;&lt;input type="radio" name="radio3" /&gt;   &lt;input type="radio" name="radio3" /&gt;&lt;/p&gt;   &lt;p&gt;&lt;input type="radio" name="radio4"   value="Un" /&gt; &lt;input type="radio"   name="radio4" value="Dous" /&gt;&lt;/p&gt;   &lt;p&gt;&lt;input type="radio" name="radio5"   value="Un" /&gt; &lt;input type="radio"   name="radio5" value="Dous" /&gt;&lt;/p&gt;   &lt;p&gt;&lt;input type="submit" value="Submit"   /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html;   charset=UTF-8');   print_r(\$_REQUEST);   echo "&lt;br/&gt;O valor recibido do control   radio2 é: ".\$_REQUEST["radio2"];   echo "&lt;br/&gt;O valor recibido do control   radio3 é: ".\$_REQUEST["radio3"];   echo "&lt;br/&gt;O valor recibido do control   radio4 é: ".\$_REQUEST["radio4"];   echo "&lt;br/&gt;O valor recibido do control   radio5 é: ".\$_REQUEST["radio5"];   ?&gt;</pre>	<pre>Array ( [radio2] =&gt; on [radio3] =&gt; on [radio4] =&gt; Un [radio5] =&gt; Dous ) O valor recibido do control radio2 é: on O valor recibido do control radio3 é: on O valor recibido do control radio4 é: Un O valor recibido do control radio5 é: Dous</pre>

## Menú

select - option

Este control envía sempre a opción elixida. O valor enviado é o contido da etiqueta option seleccionada se a opción elixida non ten definido o atributo value, ou o valor do atributo value se está definido para esa opción.

Se o menú admite selección múltiple, entón o nome do menú debe acabar con corchetes ([ ]) e se envía como un array de tantos elementos como se elixise.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt;   &lt;select name="menu1"&gt;     &lt;option selected="selected"&gt;&lt;/option&gt;     &lt;option&gt;&lt;/option&gt;   &lt;/select&gt;   &lt;br/&gt;   &lt;select name="menu2"&gt;     &lt;option selected="selected"&gt;Opción     1&lt;/option&gt;     &lt;option&gt;Opción 2&lt;/option&gt;   &lt;/select&gt;   &lt;br/&gt;   &lt;select name="menu3"&gt;     &lt;option selected="selected"     value="Un"&gt;Opción 1&lt;/option&gt;     &lt;option value="Dous"&gt;Opción 2&lt;/option&gt;   &lt;/select&gt;   &lt;br/&gt;   &lt;select name="menu4[]" size="3"   multiple="multiple"&gt;     &lt;option selected="selected"&gt;Opción     1&lt;/option&gt;     &lt;option&gt;Opción 2&lt;/option&gt;     &lt;option&gt;Opción 3&lt;/option&gt;   &lt;/select&gt;   &lt;p&gt;&lt;input type="submit" value="Submit"   /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html;   charset=UTF-8');   print_r(\$_REQUEST);   echo "&lt;br/&gt;O valor recibido do control   menu1 é: ".\$_REQUEST["menu1"];   echo "&lt;br/&gt;O valor recibido do control   menu2 é: ".\$_REQUEST["menu2"];   echo "&lt;br/&gt;O valor recibido do control   menu3 é: ".\$_REQUEST["menu3"];   echo "&lt;br/&gt;O valor recibido do control   menu4 é o array: ";   print_r(\$_REQUEST["menu4"]);   ?&gt;</pre>	<pre>Array ( [menu1] =&gt; [menu2] =&gt; Opción 1 [menu3] =&gt; Dous [menu4] =&gt; Array ( [0] =&gt; Opción 1 [1] =&gt; Opción 3 ) ) O valor recibido do control menu1 é: O valor recibido do control menu2 é: Opción 1 O valor recibido do control menu3 é: Dous O valor recibido do control menu4 é o array: Array ( [0] =&gt; Opción 1 [1] =&gt; Opción 3 )</pre>

## Control oculto

type="hidden"

Este control envíase sempre e o valor enviado é o valor do atributo value.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt;   &lt;input type="hidden" name="oculto1" /&gt;   &lt;input type="hidden" name="oculto2" value="Calquera cousa" /&gt;   &lt;p&gt;&lt;input type="submit" value="Submit" /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html; charset=UTF-8');   print_r(\$_REQUEST);   echo "&lt;br/&gt;O valor recibido do control oculto1 é: " . \$_REQUEST["oculto1"];   echo "&lt;br/&gt;O valor recibido do control oculto2 é: " . \$_REQUEST["oculto2"]; ?&gt;</pre>	<pre>Array ( [oculto1] =&gt; [oculto2] =&gt; Calquera cousa ) O valor recibido do control oculto1 é: O valor recibido do control oculto2 é: Calquera cousa</pre>

## Selector de arquivos

type="file"

O selector de arquivos permite **enviar un ficheiro** desde o cliente ata o servidor. O valor enviado é o nome do arquivo elixido.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt;   &lt;input type="file" name="arquivo1" /&gt;   &lt;p&gt;&lt;input type="submit" value="Submit" /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html; charset=UTF-8');   print_r(\$_REQUEST);   echo "&lt;br/&gt;O valor recibido do control arquivo1 é: " . \$_REQUEST["arquivo1"]; ?&gt;</pre>	<pre>Array ( [arquivo1] =&gt; datos.txt ) O valor recibido do control arquivo1 é: datos.txt</pre>

Para que este control envíe toda a información, o formulario debe ter o **atributo enctype** co valor **multipart/form-data** (un tipo de codificación que han de usar os formularios que envían ficheiros) e ser enviado co **método POST**. A información almacénase entón no **array \$\_FILES** (pero non no array **\$\_REQUEST**).

php.ini

Para poder recibir ficheiros é necesario **ter establecidas** a seguintes **directivas** no arquivo de configuración PHP **php.ini**:

- **file\_uploads**: pode tomar un dos valores “on” ou “off”. Permite que haxa ou non cargas de ficheiros.
- **uploads\_max\_filesize**: tamaño máximo do ficheiro que se pode subir.
- **upload\_tmp\_dir**: directorio temporal onde se gardan os ficheiros cargados.
- **post\_max\_size**: tamaño máximo dos datos enviados polo método post.

Revisar configuración ficheiro

Ao editar o ficheiro php.ini, as tres primeiras variables da anterior lista toman os seguintes valores:

```

C:\xampp\php\php.ini - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Macro  Ejecutar  Plugins  Ventana  ?

907  ;;;;;;;;;;;;;;;;;
908  ; File Uploads ;
909  ;;;;;;;;;;;;;;;;;
910
911  ; Whether to allow HTTP file uploads.
912  ; http://php.net/file-uploads
913  file_uploads = On
914
915  ; Temporary directory for HTTP uploaded files (will use system default if not
916  ; specified).
917  ; http://php.net/upload-tmp-dir
918  upload_tmp_dir = "C:\xampp\tmp"
919
920  ; Maximum allowed size for uploaded files.
921  ; http://php.net/upload-max-filesize
922  upload_max_filesize = 2M
923
924  ; Maximum number of files that can be uploaded via a single request
925  max_file_uploads = 20
926

```

## \$\_FILES

Cando PHP recibe un ficheiro, o almacena no directorio especificado na directiva `upload_tmp_dir` e asigna elementos ao array `$_FILES["nomeControlFicheiro"]`. O devandito **array** é **asociativo** en contén os seguintes elementos:

- `$_FILES["nomeControlFicheiro"]["name"]`: nome que tiña o ficheiro cargado no lado de cliente.
- `$_FILES["nomeControlFicheiro"]["type"]`: tipo MIME do arquivo cargado.
- `$_FILES["nomeControlFicheiro"]["size"]`: tamaño do ficheiro cargado.
- `$_FILES["nomeControlFicheiro"]["tmp_name"]`: nome do arquivo cargado no directorio temporal do servidor.
- `$_FILES["nomeControlFicheiro"]["error"]`: código de erro (toma o valor cero se non se produce ningún erro).

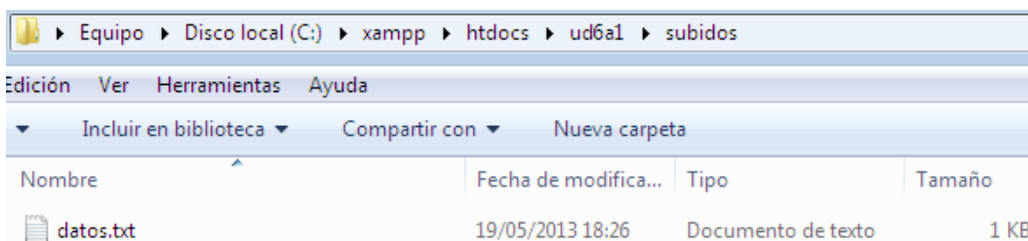
Como o ficheiro do directorio temporal desaparecerá ao rematar o script PHP, é necesario **copiar o arquivo a outro lugar**. Para isto usárase a función **`move_uploaded_file($orixe, $destino)`**, onde `$orixe` é o nome do ficheiro cargado no directorio temporal do servidor (normalmente usárase directamente `$_FILES["nomeControlFicheiro"]["tmp_name"]`) e `$destino` o nome do ficheiro que conterá a copia.

No **exemplo** que segue, envíase un **ficheiro datos.txt** que será almacenado nunha **subcartafol "subidos"** dentro do subdirectorio de `htdocs` onde está se atopa o script PHP.

Facer exemplo:  
Form\_Type\_File

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php" enctype="multipart/form-data" method="post"&gt;   &lt;input type="file" name="arquivo2" /&gt;   &lt;p&gt;&lt;input type="submit" value="Submit"   /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html;   charset=UTF-8');   print_r(\$_FILES);   echo "&lt;br/&gt;O nome do arquivo recibido no   control arquivo2 é:   " . \$_FILES["arquivo2"]["name"];    move_uploaded_file(\$_FILES["arquivo2"]["tmp   _name"],   "subidos/" . \$_FILES["arquivo2"]["name"]);   ?&gt;</pre>	<pre>Array ( [arquivo2] =&gt; Array ( [name] =&gt;   datos.txt [type] =&gt; text/plain [tmp_name] =&gt;   C:\xampp\tmp\phpD1B7.tmp [error] =&gt; 0 [size]   =&gt; 403 ) ) O nome do arquivo recibido no control arquivo2 é: datos.txt</pre>

Unha vez executado, a carpeta `subidos` conterá o ficheiro, tal como amosa a imaxe.



## Imaxe

`type="image"`

O control de tipo imaxe inserta unha **imaxe** que **funciona como un botón**. Ao facer clic nun punto da imaxe, é como facer un submit, e tanto enviaranse as coordenadas do punto no que se pulsou, como os valores dos outros controis do formulario. Definindo o atributo `value` enviarase tamén o nome do control xunto co valor do atributo.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php" enctype="multipart/form-data" method="post"&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html;</pre>	<pre>Array ( [imaxe1_x] =&gt; 112 [imaxe1_y] =&gt; 120 [imaxe1] =&gt; imaxeGNU )</pre>

<pre>&lt;input type="image" name="imaxe1" alt="GNU" src="gnu.jpg" value="imaxeGNU"/&gt; &lt;/form&gt;</pre>	<pre>charset=UTF-8); print_r(\$_REQUEST); echo "&lt;br/&gt;O valor da coordenada X do control imaxe1 onde se pulsou é: ".\$_REQUEST["imaxe1_x"]; echo "&lt;br/&gt;O valor da coordenada Y do control imaxe1 onde se pulsou é: ".\$_REQUEST["imaxe1_y"]; echo "&lt;br/&gt;O valor do control imaxe1 é: ".\$_REQUEST["imaxe1"]; ?&gt;</pre>	<p>O valor da coordenada X do control imaxe1 onde se pulsou é: 112</p> <p>O valor da coordenada Y do control imaxe1 onde se pulsou é: 120</p> <p>O valor do control imaxe1 é: imaxeGNU</p>
---	---	--

## Verificación de formularios con PHP

Cando un usuario introduce datos nun formulario pode ocorrer que non o faga de xeito adecuado, por exemplo, que non cubra todos os campos obrigatorios. No devandito exemplo sería necesario recargar o formulario para que o usuario enchese todos os controis obrigatorios, deste xeito preverase que o usuario intente acceder a un documento web sen pasar polo formulario.

Aínda que **existe a posibilidade de verificar o formulario usando Javascript no lado do cliente, evitando así tanto o envío de información pola rede para a súa validación**, como a carga de procesamento no servidor, pódese facer a verificación de formularios con PHP.

### Verificación

En xeral faranse unha serie de **comprobacións á hora de verificar un formulario**:

- Existencia dos **controis**.
- **Seguridade** nas **entradas** de datos.
- **Saída** de datos.

## Existencia dos controis

Os controis para os que se definiu o parámetro name aparecerán no array **\$\_REQUEST**. Cando o usuario non introduciase **ningún valor nun control**, este control aparecerá no array **\$\_REQUEST** cunha **cadea valeira asociada**. No caso dos controis de **campo de texto**, **controlárase cunha sentencia if** se o control contén ou non unha cadea valeira.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt; &lt;p&gt;Nome:&lt;input type="text" name="nome" /&gt;(*)&lt;/p&gt; &lt;p&gt;&lt;input type="submit" value="Submit" /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php header('Content-Type: text/html; charset=UTF-8'); echo "&lt;pre&gt;"; print_r(\$_REQUEST); echo "&lt;/pre&gt;"; echo "&lt;br/&gt;O valor do control nome é: ".\$_REQUEST["nome"]; if (\$_REQUEST["nome"]=="") print"&lt;p&gt;É obrigatorio introducir o nome&lt;/p&gt;"; else print"&lt;p&gt;O nome introducido é: \$_REQUEST[nome]&lt;/p&gt;" ?&gt;</pre>	<p>Array</p> <pre>(   [nome] =&gt; )</pre> <p>O valor do control nome é: É obrigatorio introducir o nome</p> <p>Array</p> <pre>(   [nome] =&gt; Víctor )</pre> <p>O valor do control nome é: Víctor O nome introducido é: Víctor</p>

Existencia  
`isset(variable)`

No caso dos controis como as **casinhas de verificación**, ou os **botóns de radio**, soamente están definidos no array **\$\_REQUEST** no caso en que se marcasen no formulario. Polo



tanto, ao **tratar de acceder ao elemento** correspondente no array `$_REQUEST`, **si no se marcou**, producirase un **erro**. O xeito de resolver este problema é comprobando a existencia do elemento no array `$_REQUEST` mediante o **uso da función `isset()`**, que admite como **argumento unha variable**, e devolve **verdadeiro se existe**, ou **falso se non existe**.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt;   &lt;p&gt;Desexo recibir información: &lt;input   type="checkbox" name="informacion" /&gt;&lt;/p&gt;   &lt;p&gt;Modo de pago:&lt;br/&gt; Efectivo &lt;input   type="radio" name="modoPago"   value="efectivo" /&gt;&lt;br/&gt; Tarxeta &lt;input   type="radio" name="modoPago"   value="tarxeta" /&gt;&lt;/p&gt;   &lt;p&gt;&lt;input type="submit" value="Submit"   /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html;   charset=UTF-8');   echo"&lt;pre&gt;";   print_r(\$_REQUEST);   if (isset(\$_REQUEST["informacion"]))     print"&lt;p&gt;Desexa recibir información&lt;/p&gt;";   else     print"&lt;p&gt;Non desexa recibir   información&lt;/p&gt;";   if (isset(\$_REQUEST["modoPago"]))     print"&lt;p&gt;O modo de pago elixido é:   \$_REQUEST[modoPago]&lt;/p&gt;";   else     print"&lt;p&gt;Debe elixir un modo de   pago&lt;/p&gt;";   ?&gt;</pre>	<pre>Array ( ) Non desexa recibir información</pre>
		<pre>Debe elixir un modo de pago</pre>
		<pre>Array (   [informacion] =&gt; on ) Desexa recibir información</pre> <pre>Debe elixir un modo de pago</pre> <pre>Array (   [informacion] =&gt; on   [modoPago] =&gt; tarxeta ) Desexa recibir información</pre> <pre>O modo de pago elixido é: tarxeta</pre>

## Seguridade nas entradas de datos

Pode ocorrer que o usuario intente **inserir código HTML nun control do formulario**, o que pode xerar comportamentos inesperados e riscos de seguridade. Por exemplo, nun campo de texto onde se solicita o nome, o usuario podería introducir a cadea “<strong>Pilar</strong>”. Cando se mostrase por pantalla o elemento correspondente do array `$_REQUEST`, o nome aparecería en negra. Outro **posible risco** nun campo de texto, sería a **inxección de código para atacar ao servidor**. Un exemplo sinxelo disto, sería inxectar a seguinte cadea en JavaScript “<body onload=“alert('Ola')”>”, aínda que neste caso soamente afectaría ao propio usuario amosándolle unha ventá emerxente no momento en que se tratase de mostrar o elemento correspondente do array `$_REQUEST`.

Para **eliminar** todas as **etiquetas HTML** nunha cadea, utilízase a **función `strip_tags($cadea)`**. O **inconveniente** desta función é que **elimina calquera subcadea que comece por “<”**, polo tanto non debería utilizarse no caso en que se desexa conservar ese carácter.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt;   &lt;p&gt;Nome:&lt;input type="text" name="nome"   /&gt;(&lt;*)&lt;/p&gt;   &lt;p&gt;&lt;input type="submit" value="Submit"   /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html;   charset=UTF-8');   echo"&lt;pre&gt;";   print_r(\$_REQUEST);   echo"&lt;/pre&gt;";   echo "&lt;br/&gt;O valor do control nome é:   ".strip_tags(\$_REQUEST["nome"]);   ?&gt;</pre>	<pre>Array (   [nome] =&gt; Pilar )  O valor do control nome é: Pilar</pre>

Outra acción importante a aplicar sobre a entrada de datos é a **eliminación dos espazos en branco iniciais e finais**. Isto farase aplicando á cadea a función **`trim()`**. Débese ter en conta que unha cadea que contén soamente espazos, aínda que ao mostrarse en pantalla

Seguridade  
`strip_tag(cadea)`  
`trim(cadea)`

no se vexa ningún carácter, esta cadea porén non está valeira, polo tanto cando se faga a comparación coa cadea "" na sentencia if, a condición sería falsa. Véxase o seguinte exemplo cando se introducen soamente espazos no campo de texto.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt;   &lt;p&gt;Nome:&lt;input type="text" name="nome"   /&gt;(*)&lt;/p&gt;   &lt;p&gt;&lt;input type="submit" value="Submit"   /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html;   charset=UTF-8');   echo"&lt;pre&gt;";   print_r(\$_REQUEST);   echo"&lt;/pre&gt;";   //Sen usar a función trim()   if (\$_REQUEST['nome'] == "") {     print "&lt;p&gt;Non se escribiu ningún     nome&lt;/p&gt;";   } else {     print "&lt;p&gt;O seu nome é ".     trim(\$_REQUEST['nome']) . "&lt;/p&gt;";   }   //Usando a función trim()   if (trim(\$_REQUEST['nome']) == "") {     print "&lt;p&gt;Non se escribiu ningún     nome&lt;/p&gt;";   } else {     print "&lt;p&gt;O seu nome é ".     trim(\$_REQUEST['nome']) . "&lt;/p&gt;";   } ?&gt;</pre>	<pre>Array (   [nome] =&gt; ) O seu nome é  Non se escribiu ningún nome</pre>

Pódense **combinar as dúas funcións** de seguridade vistas ata o de agora nunha mesma instrución: **trim(strip\_tags(\$\_REQUEST[control]))**. O que se pode facer é definir unha variable no código para dada un dos controis, de xeito que logo sexa máis sinxelo manexar a variable que chamar ao array \$\_REQUEST cada vez. Aínda que os campos de texto sempre existen, no seguinte exemplo úsase a función isset sobre un control de texto para indicar o método xeral de comprobación dos controis.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt;   &lt;p&gt;Nome:&lt;input type="text" name="nome"   /&gt;(*)&lt;/p&gt;   &lt;p&gt;Enderezo:&lt;input type="text"   name="endereço" /&gt;&lt;/p&gt;   &lt;p&gt;&lt;input type="submit" value="Submit"   /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html;   charset=UTF-8');   echo"&lt;pre&gt;";   print_r(\$_REQUEST);   echo"&lt;/pre&gt;";   if (isset(\$_REQUEST['nome']))  \$nome=trim(strip_tags(\$_REQUEST['nome']));   else     \$nome="";   if (isset(\$_REQUEST['endereço']))  \$endereço=trim(strip_tags(\$_REQUEST['ende rezo']));   else     \$endereço="";   if (\$nome == "")     print "&lt;p&gt;Non se escribiu ningún     nome&lt;/p&gt;";   else     print "&lt;p&gt;O seu nome é \$nome&lt;/p&gt;";   if (\$endereço == "")     print "&lt;p&gt;Non se ningún endereço&lt;/p&gt;";   else     print "&lt;p&gt;O seu endereço é     \$endereço&lt;/p&gt;"; ?&gt;</pre>	<pre>Array (   [nome] =&gt;   [endereço] =&gt; Rúa Real 10 ) Non se escribiu ningún nome  O seu endereço é Rúa Real 10</pre>

Notación  
(condición)?  
verdadeiro:  
falso;

Facer exemplo

Outro xeito de facer o anterior é usando a **notación (condición)? verdadeiro : falso;** en troques da sentencia if.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt;   &lt;p&gt;Nome:&lt;input type="text" name="nome"   /&gt;(*)&lt;/p&gt;   &lt;p&gt;Enderezo:&lt;input type="text"   name="endereço" /&gt;&lt;/p&gt;   &lt;p&gt;&lt;input type="submit" value="Submit"   /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html;   charset=UTF-8');   echo"&lt;pre&gt;";   print_r(\$_REQUEST);   echo"&lt;/pre&gt;";   \$nome= (isset(\$_REQUEST['nome']))     ? trim(strip_tags(\$_REQUEST['nome']))     : "";   \$endereço=   (isset(\$_REQUEST['endereço']))     ?   trim(strip_tags(\$_REQUEST['endereço']))     : "";   if (\$nome == "")     print "&lt;p&gt;Non se escribiu ningún   nome&lt;/p&gt;";   else     print "&lt;p&gt;O seu nome é \$nome&lt;/p&gt;";   if (\$endereço == "")     print "&lt;p&gt;Non se ningún endereço&lt;/p&gt;";   else     print "&lt;p&gt;O seu endereço é   \$endereço&lt;/p&gt;";   ?&gt;</pre>	<pre>Array (   [nome] =&gt;   [endereço] =&gt; Rúa Real 10 ) Non se escribiu ningún nome O seu endereço é Rúa Real 10</pre>

## Saída de datos

Unha vez recollidos os **datos**, cando sexan **mostrados** posteriormente nunha páxina web, **cómpre ter en conta** algúns **casos especiais**. Analizaranse dous deles, cando se trata dos caracteres **&** e **"** para despois aplicar unha solución xeral.

Tratar caracte-  
res especiais

O carácter **&** non pode ser escrito sen máis nun documento web xa que representa o **comezo dunha entidade HTML** (por exemplo, &lt;). O seguinte exemplo, aínda que funcione, **xeraría un documento HTML non válido**.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt;   &lt;p&gt;Nome:&lt;input type="text" name="nome"   /&gt;(*)&lt;/p&gt;   &lt;p&gt;&lt;input type="submit" value="Submit"   /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html;   charset=UTF-8');   echo"&lt;pre&gt;";   print_r(\$_REQUEST);   echo"&lt;/pre&gt;";   \$nome= (isset(\$_REQUEST['nome']))     ? trim(strip_tags(\$_REQUEST['nome']))     : "";   if (\$nome == "")     print "&lt;p&gt;Non se escribiu ningún   nome&lt;/p&gt;";   else     print "&lt;p&gt;O seu nome é \$nome&lt;/p&gt;";   ?&gt;</pre>	<pre>Array (   [nome] =&gt; Xoán &amp; Neves ) O seu nome é Xoán &amp; Neves</pre>

str\_replace(  
reemprazaCad,  
conCad,enCad)

A **solución** sería **substituír** co carácter **&** **pola entidade HTML correspondente**, é dicir, **&amp;**. Para isto utilizarase a función **str\_replace()**. O anterior exemplo quedaría da seguinte maneira.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt;   &lt;p&gt;Nome:&lt;input type="text" name="nome"   /&gt;(*)&lt;/p&gt;   &lt;p&gt;&lt;input type="submit" value="Submit"   /&gt;&lt;/p&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html;   charset=UTF-8');</pre>	<pre>Array (   [nome] =&gt; Xoán &amp; Neves</pre>

<pre> /&gt;&lt;/p&gt; &lt;/form&gt; </pre>	<pre> echo"&lt;pre&gt;"; print_r(\$_REQUEST); echo"&lt;/pre&gt;"; \$nome= (isset(\$_REQUEST['nome'])) ? trim(strip_tags(\$_REQUEST['nome'])) : ""; \$nome= str_replace("&amp;", "&amp;amp;", \$nome); if (\$nome == "")     print "&lt;p&gt;Non se escribiu ningún nome&lt;/p&gt;"; else     print "&lt;p&gt;O seu nome é \$nome&lt;/p&gt;"; ?&gt; </pre>	<pre> ) O seu nome é Xoán &amp; Neves </pre>
--	---	--

Cando a cadea que se introduce no campo de texto contén o carácter "&quot;", o código HTML que se poida xerar incluíndo esa cadea non será correcto. No seguinte exemplo, ao intentar construír código HTML na sentencia print, o resultado non é o esperado xa que non imprime toda a cadea. Isto é debido a que xera a cadea <p>Corrixe: <input type="text" value="Antonio "Toño" /></p>, e esas dúas " consecutivas provocan o mal funcionamento.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre> &lt;form action="exemplo.php"&gt; &lt;p&gt;Nome:&lt;input type="text" name="nome" /&gt;(*)&lt;/p&gt; &lt;p&gt;&lt;input type="submit" value="Submit" /&gt;&lt;/p&gt; &lt;/form&gt; </pre>	<pre> &lt;?php     header('Content-Type: text/html; charset=UTF-8');     echo"&lt;pre&gt;";     print_r(\$_REQUEST);     echo"&lt;/pre&gt;";     \$nome= (isset(\$_REQUEST['nome']))     ? trim(strip_tags(\$_REQUEST['nome']))     : "";     if (\$nome == "")         print "&lt;p&gt;Non se escribiu ningún nome&lt;/p&gt;";     else         print "&lt;p&gt;Corrixe: &lt;input type='text' value='\"\$nome' /&gt;&lt;/p&gt;"; ?&gt; </pre>	<pre> Array (     [nome] =&gt; Antonio "Toño" )  Corrixe: <input type="text" value="Antonio"/> </pre>

A solución ao problema anterior é **substituír** o carácter "&quot;" pola entidade HTML &quot; unha vez máis utilizando a función str\_replace().

Código do control no formulario	Código de exemplo.php	Información recibida
<pre> &lt;form action="exemplo.php"&gt; &lt;p&gt;Nome:&lt;input type="text" name="nome" /&gt;(*)&lt;/p&gt; &lt;p&gt;&lt;input type="submit" value="Submit" /&gt;&lt;/p&gt; &lt;/form&gt; </pre>	<pre> &lt;?php     header('Content-Type: text/html; charset=UTF-8');     echo"&lt;pre&gt;";     print_r(\$_REQUEST);     echo"&lt;/pre&gt;";     \$nome= (isset(\$_REQUEST['nome']))     ? trim(strip_tags(\$_REQUEST['nome']))     : "";     \$nome= str_replace("&amp;quot;", "&amp;quot;", \$nome);     if (\$nome == "")         print "&lt;p&gt;Non se escribiu ningún nome&lt;/p&gt;";     else         print "&lt;p&gt;Corrixe: &lt;input type='text' value='\"\$nome' /&gt;&lt;/p&gt;"; ?&gt; </pre>	<pre> Array (     [nome] =&gt; Antonio "Toño" )  Corrixe: <input type="text" value="Antonio Toño"/> </pre>

htmlspecialchars  
(cadea, parámetros...)

A **solución xeral** tanto para os devanditos caracteres como para outros como ' ou < e >, é o uso da función **htmlspecialchars()**. Esta función fai todas as substitucións dunha soa vez. Dado que a versión 5.4.0 de PHP modificou o xogo de caracteres predeterminado para a devandita función (de ISO-8859-1 a UTF-8) para que a función ISO-8859-1.

Facer exemplo

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt;   &lt;p&gt;Nome:&lt;input type="text" name="nome"   /&gt;(*)&lt;/p&gt;   &lt;p&gt;&lt;input type="submit" value="Submit"   /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html;   charset=UTF-8');   echo"&lt;pre&gt;";   print_r(\$_REQUEST);   echo"&lt;/pre&gt;";   \$nome= (isset(\$_REQUEST['nome']))   ?   htmlspecialchars(trim(strip_tags(\$_REQUEST[   'nome'])), ENT_QUOTES, "ISO-8859-1")   : "";   if (\$nome == "")     print "&lt;p&gt;Non se escribiu ningún   nome&lt;/p&gt;";   else     print "&lt;p&gt;Corrixe: &lt;input type='text'   value='\"\$nome\"'/&gt;&lt;/p&gt;";   ?&gt;</pre>	<pre>Array (     [nome] =&gt; Antonio '&lt;"Toño"&gt;' )</pre> <p>Corrixe: <input "="" type="text" value="Antonio "/></p>
		<pre>Array (     [nome] =&gt; Antonio '"Toño"' )</pre> <p>Corrixe: <input toño""="" type="text" value="Antonio "/></p>

## Validación dos datos dos formularios con PHP

Unha vez recibido un dato, normalmente é necesario **comprobar** se o **dato** é do **tipo esperado** (número, texto, email, data, etc) **para** poder **procesalo** sen erros. Tal como no caso da verificación, a validación de formularios tamén se pode levar a cabo no lado do cliente utilizando Javascript. Neste caso estudarase cómo facer a **validación de validación**.

Cómpre ter en conta que algunha das funcións que se van a tratar a continuación son soamente soportadas por versións recentes de PHP, polo tanto, se o código a desenvolver puidese ser implantado en servidores con versións antigas de PHP, debería ser necesario **comprobar** se o **servidor soporta unha función** antes de usala. Para isto PHP conta ca función **function\_exists()**. Véxase un exemplo de uso desta función.

Facer exemplo

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt;   &lt;p&gt;Idade:&lt;input type="text" name="idade"   /&gt;(*)&lt;/p&gt;   &lt;p&gt;&lt;input type="submit" value="Submit"   /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html;   charset=UTF-8');   echo"&lt;pre&gt;";   print_r(\$_REQUEST);   echo"&lt;/pre&gt;";   \$idade= (isset(\$_REQUEST['idade']))   ? \$_REQUEST['idade']   : "";   if (\$idade == "")     print "&lt;p&gt;Non se escribiu ningún   nome&lt;/p&gt;";   else{     if (function_exists("filter_var")){       if (filter_var(\$idade,       FILTER_VALIDATE_INT))         print "&lt;p&gt;A idade é o número       \$idade&lt;/p&gt;";       else         print "A idade ten que ser un       número enteiro";     }else       print "A función       &lt;strong&gt;filter_var()&lt;/strong&gt; non está       dispoñible neste servidor";     }   ?&gt;</pre>	<pre>Array (     [idade] =&gt; 35 anos )</pre> <p>A idade ten que ser un número enteiro</p>
		<pre>Array (     [idade] =&gt; 35 )</pre> <p>A idade é o número 35</p>

## Funcións is\_

Existe un **conxunto de funcións is\_** a partir da versión 4 de PHP e posteriores. **Son funcións booleanas que determinan se o valor que se lles pasa como parámetro é dun tipo determinado, devolvendo verdadeiro ou falso.** Este grupo de funcións recóllense na seguinte táboa:

	Función	Tipo de datos	Alias (función equivalente)
existencia	isset(\$valor)	Definida	
	is_null(\$valor)	NULL	
números	is_bool(\$valor)	Booleano	
	is_numeric(\$valor)	Número (pode ter signo, parte decimal, e pode estar expresado en notación decimal, exponencial ou hexadecimal).	
	is_int(\$valor)	Enteiro	is_integer(\$valor) is_long(\$valor)
	is_float(\$valor)	Float	is_double(\$valor) is_real(\$valor)
cadeas	is_string(\$valor)	Cadea	
outros	is_scalar(\$valor)	Escalar (entero, float, cadea ou booleano)	
	is_array(\$valor)	Array	
	is_callable(\$valor)	Función	
	is_object(\$valor)	Obxecto	
	is_resource(\$valor)	Recurso	

Estas funcións comprobán o tipo para valores de variables independentemente da súa orixe, por exemplo poden ser valores almacenados nun array. Cando os valores veñen dun formulario, estes son sempre cadeas, polo que as funcións `is_int()` e `is_float()` devolverán sempre false. No caso dos formularios, usárase a función **is\_numeric()** para comprobar se o dato é un número.

No exemplo que segue amósase o uso dalgunhas destas funcións.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt;   &lt;p&gt;Estatura:&lt;input type="text"   name="estatura" /&gt;&lt;/p&gt;   &lt;p&gt;&lt;input type="submit" value="Submit"   /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html;   charset=UTF-8');   echo"&lt;pre&gt;";   print_r(\$_REQUEST);   echo"&lt;/pre&gt;";   \$estatura= (isset(\$_REQUEST['estatura']))     ? \$_REQUEST['estatura']     : "";   if (is_numeric(\$estatura))     print "&lt;p&gt;Escribiuse o número   \$estatura.&lt;/p&gt;";   else     print "&lt;p&gt;Non se escribiu ningún número   no campo estatura.&lt;/p&gt;";   ?&gt;</pre>	<pre>Array (     [estatura] =&gt; 1X70 ) Non se escribiu ningún número no campo estatura.</pre>
		<pre>Array (     [estatura] =&gt; 1.70 ) Escribiuse o número 1.70</pre>

## Funcións ctype\_

**As funcións ctype\_ son un conxunto de funcións booleanas que devolven verdadeiro se todos os caracteres dunha cadea son dun determinado tipo.** Están dispoñibles desde a versión PHP 4.0.4 e posteriores.

Función	Tipo de datos
ctype_alnum(\$valor)	Alfanuméricos
ctype_alpha(\$valor)	Alfabéticos (maiúsculas ou minúsculas, con acentos, ñ, Ç, etc)
ctype_cntrl(\$valor)	Caracteres de control (salto de liña, tabulador, etc)
ctype_digit(\$valor)	Díxitos
ctype_graph(\$valor)	Caracteres imprimibles (excepto espazos)
ctype_lower(\$valor)	Minúsculas
ctype_print(\$valor)	Caracteres imprimibles
ctype_punct(\$valor)	Signos de puntuación (caracteres imprimibles que non son alfanuméricos nin espazos en branco)
ctype_space(\$valor)	Espazos en branco (espazos, tabuladores, saltos de liña, etc)
ctype_upper(\$valor)	Maiúsculas
ctype_xdigit(\$valor)	Díxitos hexadecimais

O seguinte exemplo amosa a comprobación de se o valor recibido do campo idade é numérico ou non.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt;   &lt;p&gt;Idade:&lt;input type="text" name="idade" /&gt;&lt;/p&gt;   &lt;p&gt;&lt;input type="submit" value="Submit" /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html;   charset=UTF-8');   echo"&lt;pre&gt;";   print_r(\$_REQUEST);   echo"&lt;/pre&gt;";   \$idade= (isset(\$_REQUEST['idade']))     ? \$_REQUEST['idade']     : "";   if (ctype_digit(\$idade))     print "&lt;p&gt;Escribiuse o número   \$idade.&lt;/p&gt;";   else     print "&lt;p&gt;Non se escribiu ningún número   no campo idade.&lt;/p&gt;";   ?&gt;</pre>	<p>Array (   [idade] =&gt; 20 ) Escribiuse o número 20.</p> <p>Array (   [idade] =&gt; 20a ) Non se escribiu ningún número no campo idade.</p>

## Funcións filter

As **funcións filter** creáronse como unha extensión de PHP PECL (PHP Extension Community Library), pero pasaron a formar parte de PHP a partir da versión 5.2.0. A función filter que interesa neste caso é **filter\_var()** que recibe como parámetro **un valor e un filtro**. Esta función devolverá verdadeiro se o valor correspóndese co tipo indicado polo filtro, e falso en caso contrario. O filtros dispoñibles para a devandita función son os que se amosan na seguinte táboa.

Filtro	Tipo de datos
FILTER_VALIDATE_INT	Entero
FILTER_VALIDATE_BOOLEAN	Booleano
FILTER_VALIDATE_FLOAT	Float
FILTER_VALIDATE_REGEXP	Expresión regular

FILTER_VALIDATE_URL	URL
FILTER_VALIDATE_EMAIL	Dirección de correo
FILTER_VALIDATE_IP	IP

No **exemplo** que segue móstrase o funcionamento da función **filter\_var()** para validar dous campos estatura e email dun formulario.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt;   &lt;p&gt;Estatura:&lt;input type="text"   name="estatura" /&gt;&lt;/p&gt;   &lt;p&gt;Email:&lt;input type="text" name="email"   /&gt;&lt;/p&gt;   &lt;p&gt;&lt;input type="submit" value="Submit"   /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html;   charset=UTF-8');   echo"&lt;pre&gt;";   print_r(\$_REQUEST);   echo"&lt;/pre&gt;";   \$estatura= (isset(\$_REQUEST['estatura']))     ? \$_REQUEST['estatura']     : "";   \$email= (isset(\$_REQUEST['email']))     ? \$_REQUEST['email']     : "";   if (filter_var(\$estatura,   FILTER_VALIDATE_FLOAT))     print "&lt;p&gt;Escribiuse o número real   \$estatura.&lt;/p&gt;";   else     print "&lt;p&gt;Non se escribiu ningún número   real no campo estatura.&lt;/p&gt;";   if (filter_var(\$email,   FILTER_VALIDATE_EMAIL))     print "&lt;p&gt;Escribiuse o enderezo de email   \$email.&lt;/p&gt;";   else     print "&lt;p&gt;Non se escribiu ningún   enderezo de email no campo email.&lt;/p&gt;"; ?&gt;</pre>	<pre>Array (   [estatura] =&gt; 1.70   [email] =&gt; anton@xunta.es ) Escribiuse o número real 1.70. Escribiuse o enderezo de email anton@xunta.es.</pre>
		<pre>Array (   [estatura] =&gt; 1X70   [email] =&gt; anton@xunta.es ) Non se escribiu ningún número real no campo estatura. Escribiuse o enderezo de email anton@xunta.es.</pre>
		<pre>Array (   [estatura] =&gt; 1X70   [email] =&gt; antonxunta.es ) Non se escribiu ningún número real no campo estatura. Non se escribiu ningún enderezo de email no campo email.</pre>

Funcións  
explode (*char*,  
*cadea*),  
checkdate  
(*array*)

Cando en algún dos campos se introduce unha data, para poder validala, unha forma sinxela é mediante o uso de dúas funcións: **explode()** e **checkdate()**. Supoñendo unha data almacenada nunha cadea, cun formato determinado dd/mm/aaaa (debería informarse deste formato ao usuario no formulario para que introduza así a data), aplicaráselle a función **explode** indicando / como carácter de separación, obtendo así un array de tres elementos: día, mes e ano. Comprobarase que o devandito array ten tres elementos, e a continuación usárase a función **checkdate()** pasándolle como parámetros o mes, o día e o ano (nesa orde). Desta maneira, a función checkdate devolverá verdadeiro se a data é válida, e falso no caso contrario. Adicionalmente, poderíanse facer outras comprobacións se, por exemplo, desexárase que o ano fose posterior ao 2000. O seguinte exemplo ilustra este proceso de validación de datas.

Código do control no formulario	Código de exemplo.php	Información recibida
<pre>&lt;form action="exemplo.php"&gt;   &lt;p&gt;Data de nacemento:&lt;input type="text"   name="dataNacemento" /&gt;&lt;/p&gt;   &lt;p&gt;&lt;input type="submit" value="Submit"   /&gt;&lt;/p&gt; &lt;/form&gt;</pre>	<pre>&lt;?php   header('Content-Type: text/html;   charset=UTF-8');   echo"&lt;pre&gt;";   print_r(\$_REQUEST);</pre>	<pre>Array (   [dataNacemento] =&gt; 24/05/2013 ) Escribiuse a data de nacemento 24/05/2013.</pre>



<p>Data de nacemento: <input type="text" value="dd/mm/aaaa"/></p> <p><input type="button" value="Submit"/></p>	<pre> echo"&lt;/pre&gt;"; \$dataNacemento= (isset(\$_REQUEST['dataNacemento'])) ? \$_REQUEST['dataNacemento'] : ""; if (\$dataNacemento!=""){     \$dataArray=explode("/", \$dataNacemento);     if (count(\$dataArray)==3){         if(checkdate(\$dataArray[1], \$dataArray[0], \$dataArray[2]))             print "&lt;p&gt;Escribiuse a data de nacemento \$dataNacemento.&lt;/p&gt;";         else             print "&lt;p&gt;Algún dos valores día, mes e ano non é válido.&lt;/p&gt;";         }else             print "&lt;p&gt;A data introducida non ten un formato válido.&lt;/p&gt;";         }else             print "&lt;p&gt;Non se escribiu ningunha no campo data de nacemento.&lt;/p&gt;"; ?&gt; </pre>	<p>Array</p> <pre> (     [dataNacemento] =&gt; 24-05-2013 ) </pre> <p>A data introducida non ten un formato válido.</p>
		<p>Array</p> <pre> (     [dataNacemento] =&gt; 24/17/2013 ) </pre> <p>Algún dos valores día, mes e ano non é válido.</p>