

## PHP : Recollida, validació i manipulació de dades

---

### Superglobal \$\_SERVER

Conté informació sobre les variables d'entorn del servidor i les peticions HTTP. Aquesta informació és útil per obtenir detalls sobre l'entorn del servidor, les capçaleres de la petició, i altres dades relacionades amb el funcionament del servidor web.

```
<?php
    echo "<h1>Informació del servidor</h1>";
    echo "<p>Navegador: ".htmlspecialchars($_SERVER['HTTP_USER_AGENT'])."</p>";
    echo "<p>Referent: ".htmlspecialchars($_SERVER['HTTP_REFERER'])."</p>";
    echo "<p>Mètode petició: ".htmlspecialchars($_SERVER['REQUEST_METHOD'])."</p>";
    echo "<p>URI de la petició: ".htmlspecialchars($_SERVER['REQUEST_URI'])."</p>";
    echo "<p>Nom del servidor: ".htmlspecialchars($_SERVER['SERVER_NAME'])."</p>";
    echo "<p>Port del servidor: ".htmlspecialchars($_SERVER['SERVER_PORT'])."</p>";
    echo "<p>Fitxer actual: ".htmlspecialchars($_SERVER['SCRIPT_FILENAME'])."</p>";
    echo "<p>Script actual: ".htmlspecialchars($_SERVER['PHP_SELF'])."</p>";
    echo "<p>String consulta: ".htmlspecialchars($_SERVER['QUERY_STRING'])."</p>";
    echo "<p>Host HTTP: ".htmlspecialchars($_SERVER['HTTP_HOST'])."</p>";
    echo "<p>IP del client: ".htmlspecialchars($_SERVER['REMOTE_ADDR'])."</p>";
    echo "<p>Protocol: ".htmlspecialchars($_SERVER['SERVER_PROTOCOL'])."</p>";
?>
```

### Sanejament de les dades

Utilitzar **htmlspecialchars()** o funcions similars és una pràctica fonamental per protegir les teves aplicacions web contra atacs maliciosos que exploten vulnerabilitats de tipus XSS. Sempre que mostris dades que poden contenir codi HTML o JavaScript, és important escapar-les adequadament per garantir la seguretat dels teus usuaris i evitar que es pugui executar codi maliciós al navegador.

### Formularis i superglobals \$\_GET i \$\_POST

Els formularis permeten als usuaris introduir dades a través d'una interfície web i enviar aquestes dades al servidor per a la seva manipulació o emmagatzematge.

El formulari s'envia al servidor mitjançant un mètode HTTP:

- GET : Les dades s'envien com a paràmetres a l'URL.
- POST : Les dades s'envien en el cos de la sol·licitud HTTP.

Utilitzarem POST per obtenir més seguretat al treballar amb dades sensibles i utilitzarem GET si les dades no són sensibles i volem que els enllaços puguin ser compartits fàcilment, permetent que altres persones accedeixin a les mateixes dades.

El fitxer PHP destinatari es defineix a l'atribut «action» i el mètode a l'atribut «method».

## Validació

Validar les dades d'un formulari és essencial per garantir que el que s'ha rebut sigui vàlid i que compleixi els requisits esperats. Això ajuda a prevenir errors, vulnerabilitats de seguretat, i assegurar-se que les dades que es processen o emmagatzemen són correctes.

Cal revisar les dades tant a la part client com a la part servidor.

```
<?php
    $errors = []; // Array per registrar els errors

    // Verifica que nom està definit
    // si està definit elimina espais en blanc innecessaris amb trim()
    // sinó retorna una cadena buida
    $nom = isset($_POST['nom']) ? trim($_POST['nom']) : '';
    if (empty($nom)) {
        $errors[] = "El nom és obligatori.";
    }

    // Validació del correu electrònic
    $email = isset($_POST['email']) ? trim($_POST['email']) : '';
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $errors[] = "El correu electrònic no és vàlid.";
    }

    // Validació de l'edat: convertir a enter i comprovar rang vàlid
    $edat = isset($_POST['edat']) ? (int)$_POST['edat'] : 0;
    if ($edat < 0 || $edat > 120) {
        $errors[] = "L'edat ha de ser un valor entre 0 i 120.";
    }

    // Comprovar si hi ha errors i informar del resultat
    if (empty($errors)) {
        echo "Dades vàlides.";
    } else {
        foreach ($errors as $error) {
            echo "<p>$error</p>";
        }
    }
?>
```

## Processament de formularis

Passos a seguir:

1. Comprovar el mètode de la sol·licitud: Abans de processar les dades, verifica el mètode HTTP utilitzat per enviar les dades. Els mètodes comuns són GET i POST. Pots usar **`$_SERVER["REQUEST_METHOD"]`**.
2. Recollir les dades del formulari: Utilitza els superglobals **`$_POST`** o **`$_GET`**, segons el mètode d'enviament, per obtenir les dades enviades pel formulari.
3. Validar les Dades: Comprova que les dades compleixin els requisits esperats, com formats, rangs i tipus.
4. Netejar les dades: Utilitza **`htmlspecialchars()`** per escapar els caràcters especials.
5. Processar les dades: Ja podem utilitzar les dades, per exemple, les podem emmagatzemar en una base de dades, generar i enviar correus electrònics personalitzats o informes, etc. De moment, simplement les mostrarem.
6. Informació a l'usuari: Mostra missatges d'error si les dades no són vàlides o confirmació si el procés ha estat completat amb èxit.

Exemple pràctic:

Formulari a la part client

<https://github.com/xbaubes/DesenvolupamentWeb/blob/main/Backend/PHP/PHP-dades/forms/formulari.php>

Processament del formulari al servidor

<https://github.com/xbaubes/DesenvolupamentWeb/blob/main/Backend/PHP/PHP-dades/forms/processaFormulari.php>

## Superglobal \$\_SESSION

Permet emmagatzemar informació per a ser utilitzada en diverses pàgines durant la sessió d'un usuari. Les dades emmagatzemades en \$\_SESSION persisteixen mentre la sessió estigui activa (fins que l'usuari tanqui el navegador o la sessió sigui destruïda).

Abans de fer servir \$\_SESSION, has d'iniciar sessió usant `<?php session_start(); ?>` al principi de la pàgina PHP.

Pots assignar-li valors tal com ho faries amb un array associatiu:

```
<?php $_SESSION['usuari'] = 'Quimet'; ?>.
```

Pots accedir a qualsevol dada emmagatzemada en la sessió des de qualsevol pàgina on s'hagi iniciat la sessió:

```
<?php echo $_SESSION['usuari']; ?>.
```

Pots buidar les dades de la sessió amb `<?php session_unset(); ?>`.

Pots eliminar la sessió i les seves dades amb `<?php session_destroy(); ?>`.

Aquí pots comprovar com es poden usar les dades guardades a la sessió a dues pàgines diferents:

<https://github.com/xbaubes/DesenvolupamentWeb/blob/main/Backend/PHP/PHP-dades/session/sessionCreation.php>

<https://github.com/xbaubes/DesenvolupamentWeb/blob/main/Backend/PHP/PHP-dades/session/sessionUnset.php>

## ACTIVITATS

1. S'ha de completar un formulari web que validi les dades dels usuaris i mostri els camps buits i els errors en funció de les opcions seleccionades.

La pàgina HTML base ja està creada i inclou un formulari amb camps per introduir el nom, correu electrònic, missatge i selecció de preferències:

<https://github.com/xbaubes/DesenvolupamentWeb/blob/main/Backend/PHP/PHP-dades/formulariActivitat/formBase.php>.

El formulari s'ha d'enviar a la mateixa pàgina, aquest és el comportament per defecte.

La validació de dades ha de passar un filtre previ a la part client amb HTML o JavaScript, aquí però ens centrarem en la part servidor i la validació en PHP.

Validacions de les dades del formulari:

- Cap camp del formulari pot estar buit.
- El camp «Nom» ha de contenir entre 3 i 20 caràcters.
- El camp «Correu electrònic» ha de tenir un format vàlid.

Visualització dels errors:

- En cas que hi hagi errors, es mostraran missatges d'error informant-ne.
- Els missatges d'error es mostraran en el color seleccionat mitjançant les opcions de radiobuttons.
- Els camps que continguin errors han de tenir un contorn del mateix color que el missatge d'error (vermell, verd o blau).
- Pots usar el següent fitxer CSS:

<https://github.com/xbaubes/DesenvolupamentWeb/blob/main/Backend/PHP/PHP-dades/formulariActivitat/formStyles.css>

Quan es rebin les dades a la pàgina s'han de mostrar novament al camp corresponent. És a dir els camps com els input s'han d'emplenar amb les dades rebudes.

Detalls tècnics a tenir en compte:

- El checkbox s'envia com un array. Pots usar la funció **in\_array( )** per verificar si un valor existeix dins d'un array.
- Per mostrar un valor a un element HTML com un input s'ha d'usar l'atribut **value**.
- Per marcar per defecte una casella de selecció HTML usem l'atribut **checked**.
- El valor del radiobutton i el checkbox és estàtic, per tal no cal aplicar tècniques de depuració de dades, en els altres camps sí.
- Per saber la longitud d'un string pots usar **strlen ( )**, aquí tens un llistat amb totes les funcions aplicables a strings: <https://www.php.net/manual/es/ref.strings.php>.
- Inspeccioneu des del navegador per veure quin codi HTML es genera.

Validació del formulari completada i en funcionament.

Aquí pots veure com els camps que s'han introduït correctament es guarden i s'insereixen novament als camps corresponents mentre que els que no són correctes es descarten i s'escriu un missatge informatiu i es marca el contorn del camp respectiu. El color amb el que es mostren els errors és el seleccionat als camps radiobutton.

## Envia les teves dades

Nom:

Correu electrònic:

Missatge:

Color errors: ☐ Vermell  
☒ Verd  
☐ Blau

Pokémon simpàtics: ☐ Bulbasaur  
☐ Charmander  
☐ Squirtle



## Envia les teves dades

El 'Correu electrònic' no és vàlid.

El camp 'Missatge' està buit.

No has seleccionat cap Pokémon.

Nom:

Correu electrònic:

Missatge:

Color errors: ☐ Vermell  
☒ Verd  
☐ Blau

Pokémon simpàtics: ☐ Bulbasaur  
☐ Charmander  
☐ Squirtle

## 2. Sistema d'autenticació bàsic amb PHP.

Crea una pàgina de connexió, gestiona la sessió de l'usuari i mostra una pàgina de benvinguda un cop l'usuari s'ha autenticat correctament.

**login.php:** Pots usar el següent formulari com a base:

<https://github.com/xbaubes/DesenvolupamentWeb/blob/main/Backend/PHP/PHP-dades/sessioActivitat/login.php>

- Has de comprovar les dades autoenviades per mètode POST.
  - Si el valor del camp «username» és «admin» i el del camp «password» és «password» crea una sessió per guardar el nom de l'usuari i immediatament redirigeix l'usuari a la pàgina «welcome.php»:  
`<?php header("Location: welcome.php"); exit; ?>`.
  - Si les credencials no són correctes, has d'indicar-ho amb un missatge a la pàgina web.
- Si accedeixes a la pàgina i tens una sessió activa has de redirigir a «welcome.php»

**welcome.php:**

- Ha de comprovar si l'usuari està autenticat revisant si existeix una sessió.
  - Si està autenticat, mostra el nom d'usuari usant el valor de la sessió.
  - Si no ho està, redirigeix a «login.php».
- Aquesta pàgina ha de contenir un enllaç a «logout.php» per eliminar la sessió.

**logout.php:**

- Elimina la sessió de l'usuari i immediatament redirigeix a «login.php».
- És una pàgina intermediària, no ha de contenir HTML.

## BIBLIOGRAFIA I WEBGRAFIA

«PHP.net». <https://www.php.net/manual/es/index.php>

«w3Schools». <https://www.w3schools.com/php/>



Autor: Xavier Baubés Parramon

Aquest document es llicència sota Creative Commons versió 4.0.  
Es permet compartir i adaptar el material però reconeixent-ne l'autor original.