



HOTELES.COM

# Pràctica 2

**Computació Orientada al Web**  
**Curs 2024**

**Fet per:**  
**Maria Pérez Pullès**

## Introducció

L'objectiu d'aquesta pràctica va ser afegir les funcionalitats de servidor i gestionar una base de dades usant PhpMyAdmin i MySQL, basant-se en la pàgina web desenvolupada en la pràctica anterior.

## Apartat 2 Secció 2

### Implementació de les funcionalitats de servidor (PHP)

En aquest apartat, he implementat la funcionalitat de servidor, simulant una cerca d'allotjaments. Per això, ha estat necessari crear dues pàgines (client.php i servidor.php). En el fitxer client.php (antic Apartat2.html), he afegit un formulari horitzontal que conté els camps destinació, entrada, sortida i hostes i un botó que s'utilitza per a enviar el formulari. A continuació estan il·lustrades les funcionalitats del codi més rellevants:

```
<form action="http://localhost/MariaPerezPulles_P2_COW/Apartado1/servidor.php"
target="_blank" method="POST">
    <div class="form-group">
        <div class="input-group mb-2">
            <div class="input-group-prepend">
                <div class="input-group-text">
                    
                </div>
            </div>
            <input type="text" class="form-control" placeholder="Introduce una
ciudad, pueblo o lugar de interés"
                name="destination" required>
            </div>
            <div class="form-row">
                <div class="form-group col-md-6">
                    <input type="date" class="form-control" placeholder="Fecha
entrada" name="checkin">
                </div>
                <div class="form-group col-md-6">
                    <input type="date" class="form-control" placeholder="Fecha
salida" name="checkout">
                </div>
            </div>
            <div class="form-group">
                <div class="input-group mb-2">
                    <div class="input-group-prepend">
                        <div class="input-group-text">
```

```

        
        </div>
        </div>
        <select id="inputState" class="form-control"
placeholder="Número de personas" name="guests">
        <option selected disabled>Número de personas</option>
        <option>1</option>
        <option>2</option>
        <option>3</option>
        <option>4</option>
        <option>5</option>
        <option>6</option>
        <option>7</option>
        <option>8</option>
        </select>
    </div>
</div>
<div>
    <button type="submit" class="btn search-button btn-lg
btn-block">Buscar</button>
</div>
</form>

```

Els atributs **action** i **method** de l'etiqueta `<form>` van ser utilitzats per direccionar el formulari cap a un fitxer handler i definir el mètode del formulari, respectivament. El control de les dades va ser realitzat en el costat del client i en el costat del servidor. Per part del client van ser utilitzats els atributs `minlength` i `maxlength` per definir un nombre mínim i un nombre màxim de caràcters. També van ser utilitzats els atributs **min** i **max** per definir el valor mínim i el màxim acceptat per al nombre d'hostes.

Per part del servidor, s'ha implementat la funció `check_dates()` per validar les dates de la cerca i s'ha utilitzat la funció `htmlspecialchars()` del PHP per evitar la injecció de codi en la variable destinació. A més, van ser utilitzats regex per a les variables destinació i hostes, comprovats per mitjà de la funció `preg_match()` del PHP.

També es van usar les excepcions per mitjà de les comandes `try`, `catch` i `throw` per redirigir l'usuari a altres pàgines, notificant el tipus d'error específic que ha ocorregut.

```

<?php

function check_dates($d1, $d2) {
    return (strtotime($d2) > strtotime($d1) && strtotime($d1) >=
c    strtotime(date("Y/m/d")));
}
try {
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $dest = htmlspecialchars($_POST["destination"]);
        $checkin = $_POST["checkin"];
    }
}

```

```
$checkout = $_POST["checkout"];
$guests = $_POST["guests"];
$regex_dest = "/^[a-zA-Z -]{1,20}$/"; // Max: 20
characters

$regex_guests = "/^[1-9]{1,2}$/"; // Max: 100 guests

if (!preg_match($regex_dest, $dest)) {
    throw new Exception('Destination field must contain
1 to 20 characters!');
}
elseif (!preg_match($regex_guests, $guests)) {
    throw new Exception('Guests number must be between 1
and 20!');
}
elseif (!check_dates($checkin, $checkout)) {
    throw new Exception('Checkin date must be greater or
equal than today!');
}

} else {
    throw new Exception('Method requested was not a POST!');
}
} catch(Exception $e){
    die('Message: '.$e->getMessage());
}

?>
```

## Apartat 2 Secció 3

### Gestió d'una base de dades usant PhpMyAdmin 4

En aquest apartat s'ha realitzat la connexió amb la base de dades i implementat dues queries necessàries per proveir les noves funcionalitats. La primera query va implementar la funcionalitat de mostrar el resultat de la cerca d'allotjaments disponibles en un destí específic. Per això, va rebre els paràmetres enviats pel formulari de la secció anterior. La connexió va ser realitzada utilitzant la llibreria PDO de PHP de la següent forma:

```
$conn = new PDO("mysql:host=$servername;dbname=world", $username, $password);  
... $rows = $conn->query("SELECT * FROM cities WHERE name = $dest");
```

El resultat de la query (\$rows) va ser mostrat en la pàgina servidor.php utilitzant cards. La funció `foreach()` de PHP va ser responsable per omplir-les de la següent forma:

```
<?php  
  
if ($rows->rowCount() > 0) {  
    echo '<div class="container">';  
    echo '<div class="row">';  
  
    foreach ($rows as $row) {  
        echo '<div class="col-md-4 mb-4">';  
        echo '<div class="card mx-auto" style="width: 18rem;">';  
        // si nay img añadila  
        // echo '';  
        echo '<div class="card-body">';  
        echo '<h5 class="card-title">' . $row['name'] . '</h5>';  
        echo '<p class="card-text">' . $row['district'] .  
</p>';  
        echo '<p class="card-text">Country: ' .  
$row['country_code'] . '</p>';  
        echo '<p class="card-text">Population: ' .  
$row['population'] . '</p>';  
        echo '</div>';  
        echo '</div>';  
        echo '</div>';  
    }  
  
    echo '</div>';  
    echo '</div>';  
} else {  
    echo 'No destinations were found. Please, try another  
destination!';
```

```
}
```

La segona query va implementar la funcionalitat d'afegir un nou client a la base de dades. Per això, va ser creat un formulari amb els camps nom, email i password que es pot accedir des del botó de Sign up del capçal de la pàgina principal. El formulari de la pàgina [sign\\_up.php](#) envia les dades a la pàgina [new\\_user.php](#), que és responsable per realitzar la connexió amb la base de dades i realitzar la query representada a continuació:

```
// Insert new user
$name_quo = $conn->quote($name); $email = $conn->quote($email); $pwd =
$conn->quote($pwd) $rows = $conn->exec("INSERT INTO students (id, name,
email, password) VALUES ($id, $name_quo, $email, $pwd)");
```

El mètode [quote\(\)](#) va ser utilitzat per codificar les variables que s'usen en la query per evitar injecció en el SQL i el mètode [exec\(\)](#) va ser utilitzat per efectuar la query d'inserció del nou client.