

# DOCUMENTACIÓ PHP INDIVIDUAL 2N SPRINT



***IES Enric Valor de Pego***  
***Curs escolar 2023-2024***  
***Desenvolupament d'aplicacions web***  
***2n curs***

***Professorat participant***  
***Vicent Jordà***

***Alumne participant***  
***Noah Calafat***



**IES Enric Valor**  
C/ Duanes, 17  
03780 - Pego  
966 409 960  
03007613@gva.es



**GENERALITAT  
VALENCIANA**  
Conselleria d'Educació,  
Cultura i Esport



**UNIÓ EUROPEA**

Fons Social Europeu

L'FSE inverteix en el teu futur

# Índex

<b>Introducció</b>	<b>2</b>
<b>Funcionament de les pàgines web</b>	<b>3</b>
Pàgina Index	3
Pàgina de creació d'una factura i enviament per a validar.	4
Codi que hem utilitzat per a obtindre informació als selectcs sobre la nostra base de dades.	4
Pàgina validació per part del servidor usant el mètode post.	7
Pàgina visualització de les nostres factures.	9
Pàgina de factura individual.	11
Excepcions usades al nostre codi.	14
<b>Conclusió</b>	<b>15</b>
<b>Webgrafia</b>	<b>16</b>



# Introducció

Aquest document té com a objectiu presentar i documentar tots el treball realitzat al segon sprint del curs per part de l'assignatura d'entorn servidor.

Aquest document tindrà documentació gràfica en gran part del seu contingut, per la meua part veurem tot el contingut tindrà relació amb l'apartat de factures, com un sistema de gestió per visualitzar totes les factures que es disposen a la nostra base de dades, com la manipulació i creació d'aquestes mateixa a la nostra futura aplicació.





# Funcionament de les pàgines web

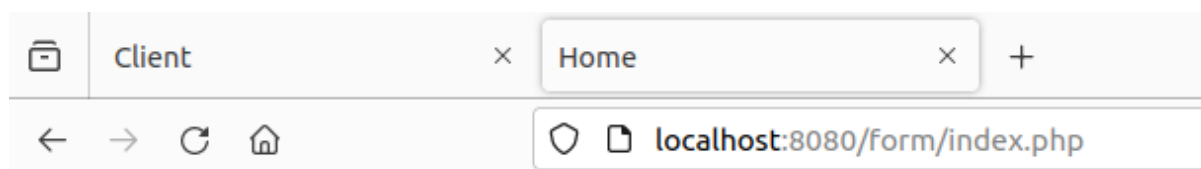
## Pàgina Index

Com es pot visualitzar aquesta esta es la nostra pàgina índex, la qual ens permet interaccionar amb les nostres pàgines de Client, Catàleg i Verificació de factures.

Al Sprint passat, els meus companys de l'equip número 3 explicaren el funcionament de les distintes pàgines Client i Catàleg, resumidament mostren informació de forma aleatoria, pero al futur mostraran informació de la nostra base de dades.

Hui explicaré y documentare el funcionament de les nostres pàgines de verificació de factures, la qual ens permetrà:

1. Navegar entre les factures existents a la base de dades.
2. Crear, modificar y eliminar factures de la nostra base de dades.
3. Veure informació detallada sobre les nostres factures.
4. Validar per part del servidor que les nostres factures s'envien adequadament.



## BHEC

[Inici](#)  
[Client](#)  
[Catàleg](#)  
[Verificar factures](#)



## Pàgina de creació d'una factura i enviament per a validar.

Una vegada hem realitzat el clic per a accedir a la nostra pàgina ens apareix el següent formulari HTML, el qual ens permetrà inserir informació per a crear una nova factura, aquesta pàgina serà la form-factura.php.

Ací podrem inserir tota la informació relacionada amb la factura que volem crear, la informació es classifica de la següent forma:

- Un input per inserir el preu de la factura.
- Un input de tipus date per introduir la data de creació de la factura.
- Un Select el qual obté un llistat dels clients de la nostra base de dades.
- Un Select el qual aconseguix un llistat de les comandes de la nostra base de dades.

Preu

Data de la Factura

Cliente:

Comanda:

Codi que hem utilitzat per a obtindre informació als select sobre la nostra base de dades.

El primer és saber com ens connectem a la nostra base de dades, jo compte amb un fitxer anomenat **config-base-dades.php** aquest ens permet poder fer ús de la nostre base de dades a altres fitxers php sols cridant al nostre fitxer amb el **require\_once** **'../classesPhp/config-base-dades.php'** cosa que ens permetrà fer ús d'eixe mateix fitxer a la nostra nova pàgina PhP.

Una vegada incorporat el require\_once 'del nostre fitxer' sols necessitarem crear una variable, la qual en totes les pàgines hem anomenat \$pdo i assignar-li un valor, el qual serà la nostra funció **obtenerConexionPDO()**.



```
require_once '../classesPhP/config-base-dades.php';

$pdo = obtenerConexionPDO();
```

Una vegada declarada la nostra variable, podem utilitzar-la per fer qualsevol tipus d'acció a la nostra base de dades, considere que aquesta és una bona pràctica per a tindre el codi de forma més organitzada.

```
// Preparar la meua consulta SQL per a obtindre informació de la meua base de dades
$stmt = $pdo->prepare("SELECT * FROM usuario");

// Es per a fer que la consulta guardada en la variable stmt s'execute
$stmt->execute();

// La variable usuarios emmagatzemarà tota la informació que retorne la consulta com un "arreglo de files" -> Preguntar a classe
$usuarios = $stmt->fetchAll(PDO::FETCH_ASSOC);

// Segona consulta
$stmt2 = $pdo->prepare("SELECT * FROM comanda");
$stmt2->execute();
$comandes = $stmt2->fetchAll(PDO::FETCH_ASSOC);
```

Posteriorment, crearem dues variables anomenades **stmt** i **stmt2** les quals ens permeten emmagatzemar informació resultant de les nostres consultes SQL, posterior a realitzar la consulta, utilitzarem l'execute per executar-les i posteriorment usarem el **fetchAll()** que retorna un array indexat pels noms de les columnes del conjunt de resultats.

Una vegada obtenida la informació de la nostra base de dades sols necessitarem mostrar-la als nostres selects HTML:

```
<div>
  <label for="cliente">Cliente:</label>
  <select id="cliente" name="cliente" required>
    <option selected disabled="disabled" hidden="true">seleccione el client</option>
    <?php foreach ($usuarios as $usuari) : ?>
      <option value="<?=$usuari['DNI']; ?>"><?=$usuari['nombre'] ?></option>
    <?php endforeach; ?>
  </select>
</div>
<div>
  <label for="comanda">Comanda:</label>
  <select id="comanda" name="comanda" required>
    <option selected disabled="disabled" hidden="true">selecciona la comanda</option>
    <?php foreach ($comandes as $comanda) : ?>
      <option value="<?=$comanda['id']; ?>"><?=$comanda['matricula_v'] ?></option>
    <?php endforeach; ?>
  </select>
</div>
```



**IES Enric Valor**  
C/ Duanes, 17  
03780 - Pego  
966 409 960  
03007613@gva.es



**GENERALITAT  
VALENCIANA**  
Conselleria d'Educació,  
Cultura i Esport



Fons Social Europeu

L'FSE inverteix en el teu futur

**UNIÓ EUROPEA**

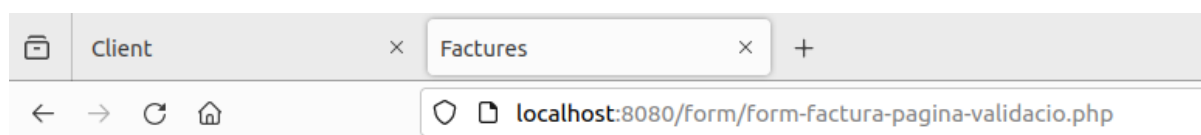
Per a mostrar la informació als selectes hem fet **2 bucles** el qual llig les **comandes** i **usuaris** de la nostra base de dades 1 per 1 i mostra únicament la informació sol·licitada en pantalla, en aquests casos el **nom d'usuari** i la **matrícula** del vehicle, i al valor del nostre select, guardarà l'**id** de la nostra comanda i el **DNI** del nostre usuari.



## Pàgina validació per part del servidor usant el mètode post.

En aquesta pàgina veurem la informació que ha segut enviada mitjançant el formulari HTML, en cas que algun camp estiga buit (sense informació), es mostrarà tots aquells camps del formulari que no han segut emplenats, i directament es cancel·larà la inserció de la factura creada a la base de dades, ja que el que voldrem és que no li falte ningun camp per emplenar al nostre formulari.

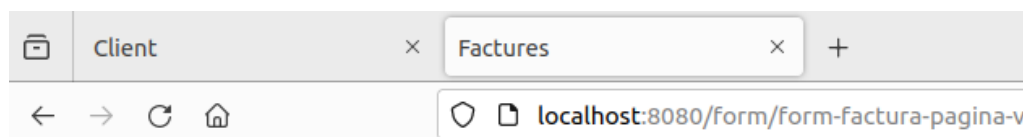
Els missatges d'errors a causa del buit al nostre formulari es mostraran de la següent forma:



## Información Factura

- El camp 'Preu' és obligatori.
- El camp 'Data de la factura' és obligatori.
- El camp 'Codi de Client' és obligatori.
- El camp 'Codi de comanda' és obligatori.

Fem ús de la validació mitjançant empty per saber si el valor del formulari està buit, en cas que no hi haguera cap camp amb algun buit, la informació de la nostra factura es mostraria de la següent forma:



Factura insertada con éxito.

## Información Factura

Preu	Data de la factura	Codi de Client	Codi de comanda
777777	2023-11-01	98765432B	1





Tota la informació introduïda al formulari adoptara una estructura de taula per a permetre a l'usuari poder analitzar la informació de forma més senzilla.

```
if (empty($price)) {
    $errors[] = "El camp 'Preu' és obligatori.";
}

if (empty($facDate)) {
    $errors[] = "El camp 'Data de la factura' és obligatori.";
}

if (empty($clienteId)) {
    $errors[] = "El camp 'Codi de Client' és obligatori.";
}

if (empty($comandaId)) {
    $errors[] = "El camp 'Codi de comanda' és obligatori.";
}

if (empty($errors)) {
    try {
        $factura = new Factura();
        $factura->setPreu($price);
        $factura->setData($facDate);
        $factura->setClient($clienteId);
        $factura->setComanda($comandaId);

        $stmt = $pdo->prepare("INSERT INTO factura (precio, fecha, dni_usuario, id_comanda) VALUES (?, ?, ?, ?)");
        $stmt->execute([ $factura->getPreu(), $factura->getDataAsString(), $factura->getClient(), $factura->getComanda()]);

        echo "Factura insertada con éxito.";
    }
}
```

Cada vegada que falte un camp de la nostra factura per introduir, aquest emetrà un missatge que serà emmagatzemat en una variable, el qual posteriorment serà llegit en nostre **HTML** mitjançant un **foreach** i mostrara en pantalla cada missatge que hi ha dins del nostre array d'errors utilitzant les etiquetes **<li>**.

```
<?php if (!empty($errors)): ?>
    <ul>
        <?php foreach ($errors as $error): ?>
            <li><?= $error; ?></li>
        <?php endforeach; ?>
    </ul>
<?php elseif (isset($factura)): ?>
```



## Pàgina visualització de les nostres factures.

Tornant al nostre formulari HTML, podem veure que en la part inferior tindrem una opció molt interessant, anomenada Pàgina modificació de factures.

Preu

Data de la Factura

Cliente:

Comanda:

### [Pàgina modificació de factures](#)

Una vegada hem accedit a aquesta pàgina s'ens mostrara de la següent forma:

### Listado de Facturas

- [Factura ID: 1](#)
  - Precio: 15000
  - Fecha: 2023-01-20
  - DNI Usuario: 12345678A
  - ID Comanda: 1
- [Factura ID: 2](#)
  - Precio: 18000
  - Fecha: 2023-02-10
  - DNI Usuario: 98765432B
  - ID Comanda: 1
- [Factura ID: 5](#)
  - Precio: 8888
  - Fecha: 2023-11-08
  - DNI Usuario: 98765432B
  - ID Comanda: 2
- [Factura ID: 6](#)
  - Precio: 2635
  - Fecha: 2023-11-08
  - DNI Usuario: 98765432B
  - ID Comanda: 2
- [Factura ID: 8](#)
  - Precio: 111
  - Fecha: 2023-11-01
  - DNI Usuario: 98765432B
  - ID Comanda: 1
- [Factura ID: 11](#)
  - Precio: 91919191
  - Fecha: 2023-09-14
  - DNI Usuario: 12345678A
  - ID Comanda: 1
- [Factura ID: 12](#)
  - Precio: 3333333
  - Fecha: 2023-11-10
  - DNI Usuario: 98765432B
  - ID Comanda: 2



Aquesta pàgina ens mostrarà un **llistat** de totes les factures que existeixen a la nostra base de dades, mostrant la informació concreta de les seues respectives factures, en un futur li donarem un aspecte molt més elegant, apart de mostrar la informació de la respectiva factura i classificar aquestes per el seu **ID**, podrem accedir a aquestes factures realitzant un clic al id de la factura.

Per a obtindre la informació de totes les factures hem fet ús d'una funció anomenada **getAllFacturas()** la qual es troba dins de la nostra classe **FacturaCRUD**, la qual està directament relacionada amb la nostra base de dades.

```
$facturaCRUD = new FacturaCRUD($pdo);  
$facturas = $facturaCRUD->getAllFacturas();
```

Per a mostrar totes les factures hem fet ús d'un **foreach** al nostre codi HTML, separant aquestes per **<li>**, **<div>**, **<a>** i **<ul>**.

```
<ul>  
  <?php foreach ($facturas as $factura): ?>  
    <li>  
      <div>  
        <a href="factura-detalles.php?id=<?= $factura->getNumFactura() ?>">  
          Factura ID: <?= $factura->getNumFactura() ?>  
        </a>  
      </div>  
      <ul>  
        <li>Precio: <?= $factura->getPreu() ?></li>  
        <li>Fecha: <?= $factura->getDataAsString() ?></li>  
        <li>DNI Usuario: <?= $factura->getClient() ?></li>  
        <li>ID Comanda: <?= $factura->getComanda() ?></li>  
      </ul>  
    </li>  
  <?php endforeach; ?>  
</ul>
```



## Pàgina de factura individual.

Una vegada hem accedit a una factura, aquesta mostrara una informació o un altra depenent dels paràmetres que siguen passats pel queryString de la pàgina web:



## Detalle de Factura

- Factura ID: 12
- Precio: 3333333
- Fecha: 2023-11-10
- DNI Usuario: 98765432B
- ID Comanda: 2

### Editar Factura

<input type="text" value="3333333"/>	<input type="text" value="10 / 11 / 2023"/>	<input type="text" value="98765432B"/>	<input type="text" value="2"/>	<button>Actualizar Factura</button>
--------------------------------------	---	--	--------------------------------	-------------------------------------

Una vegada dins de la pàgina també podrem modificar la informació de la nostra factura, aquesta pàgina, inserira també directament els valors que te la factura cada columna de la nostra taula factura de la base de dades en els inputs de modificació de la factura, per facilitar saber a l'usuari quin camp o informació està modificant.



Factura actualizada con éxito.

## Detalle de Factura

- Factura ID: 12
- Precio: 11111111
- Fecha: 2023-11-23
- DNI Usuario: 98765432B
- ID Comanda: 1

### Editar Factura

<input type="text" value="11111111"/>	<input type="text" value="23 / 11 / 2023"/>	<input type="text" value="98765432B"/>	<input type="text" value="1"/>	<button>Actualizar Factura</button>
---------------------------------------	---	--	--------------------------------	-------------------------------------



Una vegada modificada la informació de la nostra factura, aquesta s'actualitzara automàticament en la nostra pàgina web, mostrant la nova informació immediatament.

```
<?php
require '../classesPhp/Factura.php';
require '../classesPhp/FacturaCRUD.php';
require_once '../classesPhp/config-base-dades.php';

$pdo = obtenerConexionPDO();

$facturaCRUD = new FacturaCRUD($pdo);

if (isset($_GET['id'])) {
    $facturaId = $_GET['id'];
    $factura = $facturaCRUD->getFacturaById($facturaId);

    if (isset($_POST['updateFactura'])) {
        // Procesar el formulario de actualización
        $factura->setPreu($_POST['precio']);
        $factura->setData($_POST['fecha']);
        $factura->setClient($_POST['dni_usuario']);
        $factura->setComanda($_POST['id_comanda']);

        if ($facturaCRUD->updateFactura($facturaId, $factura)) {
            echo "Factura actualizada con éxito.";
        } else {
            echo "Error al actualizar la factura.";
        }
    }
} else {
    echo 'xd';
}
?>
```

Per modificar la informació he fet ús dels mètodes dins de la meua classe FacturaCRUD, els quals ens permeten la modificació de la informació de la nostra base de dades de forma molt més senzilla.



Els canvis realitzats amb la corresponent factura es veuran reflexats en totes les pàgines que obtinguen informació d'eixa factura, ja que actua a nivell de base de dades.

## Listado de Facturas

- [Factura ID: 1](#)
  - Precio: 15000
  - Fecha: 2023-01-20
  - DNI Usuario: 12345678A
  - ID Comanda: 1
- [Factura ID: 2](#)
  - Precio: 18000
  - Fecha: 2023-02-10
  - DNI Usuario: 98765432B
  - ID Comanda: 1
- [Factura ID: 5](#)
  - Precio: 8888
  - Fecha: 2023-11-08
  - DNI Usuario: 98765432B
  - ID Comanda: 2
- [Factura ID: 6](#)
  - Precio: 2635
  - Fecha: 2023-11-08
  - DNI Usuario: 98765432B
  - ID Comanda: 2
- [Factura ID: 8](#)
  - Precio: 111
  - Fecha: 2023-11-01
  - DNI Usuario: 98765432B
  - ID Comanda: 1
- [Factura ID: 11](#)
  - Precio: 91919191
  - Fecha: 2023-09-14
  - DNI Usuario: 12345678A
  - ID Comanda: 1
- [Factura ID: 12](#)
  - Precio: 11111111
  - Fecha: 2023-11-23
  - DNI Usuario: 98765432B
  - ID Comanda: 1



## Excepcions usades al nostre codi.

Basicament he fet us de excepcions personalitzades, he creat una classe anomenada `baseDadesException` i he utilitzat aquesta funció per a que mostre un missatge de error a la nostra pantalla en cas de que no es puga accedir a la nostra base de dades.

```
class BaseDadesException extends Exception {  
    public function __construct($message = "Error en la base de dades", $code = 0, Exception $previous = null) {  
        parent::__construct($message, $code, $previous);  
    }  
  
    public function __toString() {  
        return __CLASS__ . ": [{".$this->code}]: {".$this->message}";  
    }  
}  
  
try {  
    $pdo = obtenerConexionPDO();  
}  
catch (BaseDadesException $e) {  
    echo "No s'ha pogut accedir a la base de dades: " . $e->getMessage();  
}  
catch (Exception $e) {  
    echo "Excepció genèrica capturada: " . $e->getMessage();  
}
```

Utilitze aquesta excepció per a la connexió amb la meua base de dades, en aquest cas utilitze dos `catch` perquè el que vull és que principal capture la meua excepció, però en cas que l'excepció que es mostrara en pantalla no fora una excepció de la meua classe personalitzada es mostraria que l'excepció capturada és una genèrica.



## Conclusió

*En aquesta documentació hem presentat i documentat el treball realitzat en el segon sprint del curs d'entorn servidor. Hem centrat especialment en l'apartat de factures dins de la nostra aplicació web.*

*Durant aquesta etapa, hem après com interactuar amb les pàgines de client i catàleg, que, tot i mostrar informació aleatòria, estaran connectades a la nostra base de dades en un futur proper.*

*Hem explicat i documentat detalladament el funcionament de les pàgines de verificació de factures, que ens permeten navegar per les factures existents a la base de dades, crear-ne de noves, modificar-les i eliminar-les. També hem destacat la validació per part del servidor per assegurar que les factures s'envien adequadament.*

*La pàgina de creació d'una factura i enviament per validar ens permet inserir la informació necessària per a crear una nova factura, com el preu, la data de creació i la selecció de clients i comandes de la base de dades. Hem explicat com obtenim la informació d'aquests selectes mitjançant consultes a la base de dades.*

*La validació per part del servidor mitjançant el mètode POST ens assegura que tots els camps del formulari estiguin complets abans de la inserció de la factura a la base de dades. En cas de camps buits, es mostren missatges d'error.*

*Hem detallat la pàgina de visualització de les factures, on es mostren totes les factures existents, classificades per ID. A més, hem destacat la possibilitat d'accedir a les factures individuals mitjançant un clic.*

*Les pàgines individuals de les factures mostren la informació concreta de cada factura i permeten la seva modificació. Qualsevol canvi es reflecteix immediatament en la interfície, ja que les dades es gestionen a nivell de base de dades.*

*Finalment, hem explicat l'ús de les excepcions personalitzades, com la `baseDadesException`, que es fan servir per gestionar els errors de connexió a la base de dades de manera controlada.*





**IES Enric Valor**  
C/ Duanes, 17  
03780 - Pego  
966 409 960  
03007613@gva.es



**GENERALITAT  
VALENCIANA**  
Conselleria d'Educació,  
Cultura i Esport



Fons Social Europeu

L'FSE inverteix en el teu futur

**UNIÓ EUROPEA**

*Aquesta documentació serveix com a guia completa per a comprendre el funcionament de les pàgines relacionades amb les factures a la nostra aplicació web, i com s'han implementat les funcionalitats i les mesures de seguretat per a garantir un funcionament adequat i una gestió eficaç de les dades.*

## Webgrafia

*PHP Documentation* [en línia] <<https://www.php.net/docs.php>> [Consulta: 17 octubre 2023]

*PHP PDO: MANUAL* [en línia] <<https://www.php.net/manual/en/book.pdo.php>> [Consulta: 06 octubre 2023] [PHP: PDO - Manual](#)