|  |
| --- |
| **puc migrar cap a password\_hash() tenint en compte que la base de dades és mariadb?** |

Sí, **pots migrar perfectament cap a password\_hash() encara que la teva base de dades sigui MariaDB**, perquè:

👉 **password\_hash() i password\_verify() són funcions de PHP**, no de la base de dades.  
➡️ Això vol dir que el **hash es genera i comprova en PHP**, i **MariaDB només guarda la cadena resultant** com un simple CHAR(255) o VARCHAR(255).

**✅ Com migrar correctament**

**1. 🔧 Canvia el camp pwdhash a un tipus més gran (opcional però recomanat)**

Els hashes generats per password\_hash() poden arribar als **255 caràcters**. Modifica la teva taula així:

ALTER TABLE personal MODIFY pwdhash VARCHAR(255) NOT NULL;

**2. 🔒 Elimina el camp password (opcional però recomanat)**

Per seguretat, si ja tens el hash, aquest camp és innecessari:

ALTER TABLE personal DROP COLUMN password;

**3. 🧑‍🍳 Inserció d'usuaris nous des de PHP**

$nom = "Kevin José CUI";

$rol = "cuiner";

$username = "cuiner2";

$password = "cuiner2";

$host = "localhost";

$pwdhash = password\_hash($password, PASSWORD\_DEFAULT); // genera el hash segur

$sql = "INSERT INTO personal (nom, rol, username, pwdhash, host) VALUES (?, ?, ?, ?, ?)";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("sssss", $nom, $rol, $username, $pwdhash, $host);

$stmt->execute();

**4. 🔐 Login (verificació de la contrasenya)**

$username = $\_POST['username'];

$password = $\_POST['password'];

$host = $\_POST['host'];

$sql = "SELECT \* FROM personal WHERE username = ? AND host = ? AND es\_actiu = 1";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("ss", $username, $host);

$stmt->execute();

$result = $stmt->get\_result();

if ($usuari = $result->fetch\_assoc()) {

if (password\_verify($password, $usuari['pwdhash'])) {

echo "Benvingut " . $usuari['nom'];

// aquí pots crear la sessió o el que necessitis

} else {

echo "Contrasenya incorrecta.";

}

} else {

echo "Usuari o host incorrectes.";

}

**🧽 Bonus: neteja de dades antigues**

Si tenies usuaris antics amb ed25519\_password() i vols regenerar els seus pwdhash, pots fer un petit script per fer-ho un cop (si tens la contrasenya en clar, o saps regenerar-la):

$sql = "SELECT id\_personal, username FROM personal";

$result = $conn->query($sql);

while ($row = $result->fetch\_assoc()) {

$id = $row['id\_personal'];

$username = $row['username'];

$newHash = password\_hash($username, PASSWORD\_DEFAULT); // si el password és el mateix que el username

$update = $conn->prepare("UPDATE personal SET pwdhash = ? WHERE id\_personal = ?");

$update->bind\_param("si", $newHash, $id);

$update->execute();

}

Et puc ajudar a fer aquesta migració pas a pas, si vols. T'interessaria automatitzar l'actualització de tots els **pwdhash antics?**

|  |
| --- |
| **com hauria de fer un 'insert into personal values....' en DML ?** |

En DML (**Data Manipulation Language**), una instrucció INSERT INTO serveix per inserir dades dins una taula, i la forma correcta depèn de si indiques o no els noms de les columnes.

**✅ Forma Recomanada (explícita amb columnes)**

Sempre és millor indicar les columnes (especialment quan hi ha AUTO\_INCREMENT, DEFAULT, etc.):

INSERT INTO personal (nom, rol, username, pwdhash, host)

VALUES ('Kevin José CUI', 'cuiner', 'cuiner2', 'HASH\_GENERAT', 'localhost');

🔐 Substitueix 'HASH\_GENERAT' pel hash que hauries generat prèviament amb password\_hash() des de PHP.

**❌ Forma NO recomanada (omplint tots els camps)**

Aquesta només funciona si **inclous tots els camps** en el mateix ordre de definició, inclòs el id\_personal (auto\_increment), que no és recomanable fer manualment:

INSERT INTO personal

VALUES (NULL, 'Kevin José CUI', 'cuiner', 'cuiner2', 'cuiner2', 'HASH\_GENERAT', 'localhost', TRUE);

Aquesta forma no és segura perquè:

* Si canvies l'ordre o afegeixes columnes a la taula, fallarà.
* Estàs inserint la contrasenya en clar ('cuiner2'), cosa que no hauries de fer.
* Estàs tocant el id\_personal, que hauria de generar-se automàticament.

**🧑‍🍳 Exemple correcte i segur**

Si ja tens el hash generat a PHP:

$password = "cuiner2";

$pwdhash = password\_hash($password, PASSWORD\_DEFAULT);

// En SQL:

INSERT INTO personal (nom, rol, username, pwdhash, host)

VALUES ('Kevin José CUI', 'cuiner', 'cuiner2', '$pwdhash', 'localhost');

**Vols que et generi un script complet en PHP que faci aquest INSERT amb password\_hash()?**

|  |
| --- |
| **si** |

Perfecte! Aquí tens un **script complet en PHP** que fa un INSERT INTO personal utilitzant password\_hash() de forma segura, amb connexió a MariaDB via mysqli.

**✅ Script PHP: inserir un usuari nou**

<?php

// Configura la connexió a la base de dades

$host = "localhost";

$dbname = "el\_teu\_nom\_de\_base\_de\_dades";

$user = "el\_teu\_usuari";

$pass = "la\_teva\_contrasenya";

// Crear connexió

$conn = new mysqli($host, $user, $pass, $dbname);

// Comprovar connexió

if ($conn->connect\_error) {

die("Connexió fallida: " . $conn->connect\_error);

}

// Dades del nou usuari

$nom = "Kevin José CUI";

$rol = "cuiner"; // 'cuiner', 'cambrer' o 'administrador'

$username = "cuiner2";

$password = "cuiner2";

$host\_user = "localhost";

// Generar hash segur de la contrasenya

$pwdhash = password\_hash($password, PASSWORD\_DEFAULT);

// Preparar consulta

$sql = "INSERT INTO personal (nom, rol, username, pwdhash, host)

VALUES (?, ?, ?, ?, ?)";

$stmt = $conn->prepare($sql);

if (!$stmt) {

die("Error a prepare: " . $conn->error);

}

$stmt->bind\_param("sssss", $nom, $rol, $username, $pwdhash, $host\_user);

// Executar

if ($stmt->execute()) {

echo "Usuari inserit correctament.";

} else {

echo "Error a l'inserció: " . $stmt->error;

}

// Tancar connexió

$stmt->close();

$conn->close();

?>

**🔐 Consells extres**

* Pots adaptar-ho fàcilment per rebre les dades via $\_POST des d’un formulari.
* Si vols controlar duplicats (username + host), pots capturar errors de tipus duplicate key i avisar a l'usuari.

Vols també un petit formulari HTML per inserir usuaris amb aquest script?

|  |
| --- |
| **si** |

Genial! Aquí tens un petit **formulari HTML** juntament amb el mateix script PHP adaptat per rebre les dades via POST.

**📝 1. Formulari HTML (formulari\_personal.html)**

<!DOCTYPE html>

<html lang="ca">

<head>

<meta charset="UTF-8">

<title>Alta de Personal</title>

</head>

<body>

<h2>Afegir nou membre del personal</h2>

<form action="inserir\_personal.php" method="POST">

<label>Nom:</label><br>

<input type="text" name="nom" required><br><br>

<label>Rol:</label><br>

<select name="rol" required>

<option value="cuiner">Cuiner</option>

<option value="cambrer">Cambrer</option>

<option value="administrador">Administrador</option>

</select><br><br>

<label>Nom d'usuari:</label><br>

<input type="text" name="username" required><br><br>

<label>Contrasenya:</label><br>

<input type="password" name="password" required><br><br>

<label>Host:</label><br>

<input type="text" name="host" value="localhost" required><br><br>

<input type="submit" value="Afegir">

</form>

</body>

</html>

**💾 2. Script PHP (inserir\_personal.php)**

<?php

// Connexió

$host = "localhost";

$dbname = "el\_teu\_nom\_de\_base\_de\_dades";

$user = "el\_teu\_usuari";

$pass = "la\_teva\_contrasenya";

$conn = new mysqli($host, $user, $pass, $dbname);

if ($conn->connect\_error) {

die("Error de connexió: " . $conn->connect\_error);

}

// Recollida de dades

$nom = $\_POST['nom'] ?? '';

$rol = $\_POST['rol'] ?? '';

$username = $\_POST['username'] ?? '';

$password = $\_POST['password'] ?? '';

$host\_user = $\_POST['host'] ?? 'localhost';

// Validació bàsica (opcionalment pots fer més control)

if (!$nom || !$rol || !$username || !$password || !$host\_user) {

die("Falten dades.");

}

// Generar hash

$pwdhash = password\_hash($password, PASSWORD\_DEFAULT);

// Inserció

$sql = "INSERT INTO personal (nom, rol, username, pwdhash, host)

VALUES (?, ?, ?, ?, ?)";

$stmt = $conn->prepare($sql);

if (!$stmt) {

die("Error a prepare: " . $conn->error);

}

$stmt->bind\_param("sssss", $nom, $rol, $username, $pwdhash, $host\_user);

if ($stmt->execute()) {

echo "Usuari inserit correctament.";

} else {

if ($conn->errno === 1062) {

echo "Error: Ja existeix un usuari amb aquest username i host.";

} else {

echo "Error a l'inserció: " . $stmt->error;

}

}

$stmt->close();

$conn->close();

?>

**🧪 Vols que aquest formulari validi que el username no existeix abans d’enviar el formulari (via AJAX)?**

|  |
| --- |
| **si** |

Perfecte! Ara t'ensenyaré com fer una **validació en temps real amb AJAX** per comprovar si el username ja existeix **abans d’enviar el formulari**. Això millora molt l’experiència d’usuari.

**📁 Estructura dels fitxers**

projecte/

│

├── formulari\_personal.html

├── inserir\_personal.php

└── comprovar\_username.php ← (Nou fitxer per AJAX)

**📝 1. formulari\_personal.html (actualitzat amb AJAX i jQuery)**

Inclourem jQuery i un espai per missatges de validació:

<!DOCTYPE html>

<html lang="ca">

<head>

<meta charset="UTF-8">

<title>Alta de Personal</title>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

</head>

<body>

<h2>Afegir nou membre del personal</h2>

<form id="formulari" action="inserir\_personal.php" method="POST">

<label>Nom:</label><br>

<input type="text" name="nom" required><br><br>

<label>Rol:</label><br>

<select name="rol" required>

<option value="cuiner">Cuiner</option>

<option value="cambrer">Cambrer</option>

<option value="administrador">Administrador</option>

</select><br><br>

<label>Nom d'usuari:</label><br>

<input type="text" name="username" id="username" required><br>

<span id="missatge-username" style="color: red;"></span><br><br>

<label>Contrasenya:</label><br>

<input type="password" name="password" required><br><br>

<label>Host:</label><br>

<input type="text" name="host" value="localhost" required><br><br>

<input type="submit" value="Afegir">

</form>

<script>

$(document).ready(function() {

$('#username').on('input', function() {

const username = $(this).val();

const host = $('input[name="host"]').val();

if (username.length > 2) {

$.post('comprovar\_username.php', { username: username, host: host },

function(data) {

if (data === 'existeix') {

$('#missatge-username').text('Aquest username ja existeix!');

$('input[type="submit"]').prop('disabled', true);

} else {

$('#missatge-username').text('');

$('input[type="submit"]').prop('disabled', false);

}

});

} else {

$('#missatge-username').text('');

$('input[type="submit"]').prop('disabled', false);

}

});

});

</script>

</body>

</html>

**🆕 2. comprovar\_username.php**

Aquest fitxer repeteix part de la connexió a la base de dades i comprova si el username + host ja existeixen.

<?php

$host = "localhost";

$dbname = "el\_teu\_nom\_de\_base\_de\_dades";

$user = "el\_teu\_usuari";

$pass = "la\_teva\_contrasenya";

$conn = new mysqli($host, $user, $pass, $dbname);

if ($conn->connect\_error) {

die("Error: " . $conn->connect\_error);

}

$username = $\_POST['username'] ?? '';

$host\_user = $\_POST['host'] ?? '';

$sql = "SELECT id\_personal FROM personal WHERE username = ? AND host = ?";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("ss", $username, $host\_user);

$stmt->execute();

$stmt->store\_result();

if ($stmt->num\_rows > 0) {

echo "existeix";

} else {

echo "lliure";

}

$stmt->close();

$conn->close();

?>

**✅ Resultat**

* Quan l’usuari escriu un username, el sistema comprova si existeix i mostra un missatge.
* El botó d’enviament es bloqueja automàticament si el nom d’usuari no és vàlid.

Vols que afegim també una comprovació que la **contrasenya sigui forta (mínim caràcters, números, etc.)** abans d'enviar?

|  |
| --- |
| **si** |

Perfecte! Ara afegirem una **validació de la contrasenya en temps real al formulari**, per assegurar-nos que sigui prou forta **abans de permetre l'enviament**.

**🎯 Requisits per a la contrasenya:**

* Mínim **8 caràcters**
* Almenys **una lletra majúscula**
* Almenys **una lletra minúscula**
* Almenys **un número**

**📝 Actualització de formulari\_personal.html amb validació de contrasenya**

Afegim el control just després del camp de contrasenya:

<label>Contrasenya:</label><br>

<input type="password" name="password" id="password" required><br>

<span id="missatge-password" style="color: red;"></span><br><br>

I afegim el codi jQuery per validar la contrasenya:

<script>

function validarPassword(password) {

// Regex: mínim 8 caràcters, almenys una minúscula, una majúscula i un número

const regex = /^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d).{8,}$/;

return regex.test(password);

}

$(document).ready(function() {

function comprovarSubmit() {

const usernameError = $('#missatge-username').text() !== '';

const password = $('#password').val();

const passwordValid = validarPassword(password);

if (usernameError || !passwordValid) {

$('input[type="submit"]').prop('disabled', true);

} else {

$('input[type="submit"]').prop('disabled', false);

}

}

$('#username').on('input', function() {

const username = $(this).val();

const host = $('input[name="host"]').val();

if (username.length > 2) {

$.post('comprovar\_username.php', { username: username, host: host },

function(data) {

if (data === 'existeix') {

$('#missatge-username').text('Aquest username ja existeix!');

} else {

$('#missatge-username').text('');

}

comprovarSubmit();

});

} else {

$('#missatge-username').text('');

comprovarSubmit();

}

});

$('#password').on('input', function() {

const password = $(this).val();

if (!validarPassword(password)) {

$('#missatge-password').text('La contrasenya ha de tenir almenys

8 caràcters, una majúscula, una minúscula i un número.');

} else {

$('#missatge-password').text('');

}

comprovarSubmit();

});

});

</script>

**✅ Resultat:**

* **El formulari només es pot enviar si:**
  + El username no existeix
  + La contrasenya compleix els requisits de seguretat
* **Missatges d’error dinàmics** avisen l’usuari abans d’enviar.

Vols que també validem que el **camp "Nom" no contingui números** o que **tots els camps estiguin plens abans de validar res més**?

|  |
| --- |
| **si** |

Perfecte! Anem a reforçar la validació del formulari:

**🎯 Noves validacions que afegirem:**

1. ✅ **El camp “Nom”** no pot contenir números ni caràcters especials.
2. ✅ Tots els camps han d’estar **plens** abans de validar el username o la contrasenya.

**📝 Codi jQuery actualitzat amb totes les validacions:**

Afegeix això dins del teu <script> al final de formulari\_personal.html:

function validarPassword(password) {

const regex = /^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d).{8,}$/;

return regex.test(password);

}

function validarNom(nom) {

const regex = /^[A-Za-zÀ-ÿ\s\-']+$/; // Accepta lletres amb accents, espais, guions i apòstrofs

return regex.test(nom);

}

$(document).ready(function() {

function comprovarSubmit() {

const nom = $('#nom').val().trim();

const username = $('#username').val().trim();

const password = $('#password').val();

const host = $('input[name="host"]').val().trim();

const usernameError = $('#missatge-username').text() !== '';

const passwordValid = validarPassword(password);

const nomValid = validarNom(nom);

const campsPlens = nom && username && password && host;

// Missatge pel nom

if (!nomValid && nom.length > 0) {

$('#missatge-nom').text('El nom només pot contenir lletres i espais.');

} else {

$('#missatge-nom').text('');

}

// Activar o desactivar el botó

if (usernameError || !passwordValid || !nomValid || !campsPlens) {

$('input[type="submit"]').prop('disabled', true);

} else {

$('input[type="submit"]').prop('disabled', false);

}

}

$('#nom, #username, #password, input[name="host"]').on('input', function() {

comprovarSubmit();

});

$('#username').on('input', function() {

const username = $(this).val();

const host = $('input[name="host"]').val();

if (username.length > 2) {

$.post('comprovar\_username.php', { username: username, host: host },

function(data) {

if (data === 'existeix') {

$('#missatge-username').text('Aquest username ja existeix!');

} else {

$('#missatge-username').text('');

}

comprovarSubmit();

});

} else {

$('#missatge-username').text('');

comprovarSubmit();

}

});

});

**✏️ Actualitza el formulari HTML també:**

Afegim el id="nom" i el span per mostrar missatges:

<label>Nom:</label><br>

<input type="text" name="nom" id="nom" required><br>

<span id="missatge-nom" style="color: red;"></span><br><br>

**✅ Resultat:**

* **El camp Nom** no accepta números.
* **Tots els camps** han d’estar plens.
* El botó **es desactiva automàticament** si hi ha algun error.

Vols que t’ajudi a fer una versió estilitzada amb **Bootstrap 5** també?

|  |
| --- |
| **si** |

Perfecte! Aquí tens el formulari totalment estilitzat amb **Bootstrap 5**, amb totes les funcionalitats anteriors (validació en temps real, contrasenya forta, comprovació d’usuari via AJAX, etc.).

**📦 CDN de Bootstrap 5**

Afegirem Bootstrap 5 via CDN i una mica d’estil per a una aparença neta i moderna.

**📄 formulari\_personal.html amb Bootstrap 5**

<!DOCTYPE html>

<html lang="ca">

<head>

<meta charset="UTF-8">

<title>Alta de Personal</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

</head>

<body class="bg-light">

<div class="container mt-5">

<div class="card shadow-lg rounded-4">

<div class="card-body">

<h3 class="card-title mb-4">Afegir nou membre del personal</h3>

<form id="formulari" action="inserir\_personal.php" method="POST" novalidate>

<div class="mb-3">

<label for="nom" class="form-label">Nom complet</label>

<input type="text" class="form-control" name="nom" id="nom" required>

<div id="missatge-nom" class="form-text text-danger"></div>

</div>

<div class="mb-3">

<label for="rol" class="form-label">Rol</label>

<select class="form-select" name="rol" required>

<option value="">Selecciona un rol</option>

<option value="cuiner">Cuiner</option>

<option value="cambrer">Cambrer</option>

<option value="administrador">Administrador</option>

</select>

</div>

<div class="mb-3">

<label for="username" class="form-label">Nom d'usuari</label>

<input type="text" class="form-control" name="username" id="username" required>

<div id="missatge-username" class="form-text text-danger"></div>

</div>

<div class="mb-3">

<label for="password" class="form-label">Contrasenya</label>

<input type="password" class="form-control" name="password" id="password" required>

<div id="missatge-password" class="form-text text-danger"></div>

</div>

<div class="mb-3">

<label for="host" class="form-label">Host</label>

<input type="text" class="form-control" name="host" value="localhost" required>

</div>

<button type="submit" class="btn btn-primary w-100">Afegir</button>

</form>

</div>

</div>

</div>

<script>

function validarPassword(password) {

const regex = /^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d).{8,}$/;

return regex.test(password);

}

function validarNom(nom) {

const regex = /^[A-Za-zÀ-ÿ\s\-']+$/;

return regex.test(nom);

}

$(document).ready(function () {

function comprovarSubmit() {

const nom = $('#nom').val().trim();

const username = $('#username').val().trim();

const password = $('#password').val();

const host = $('input[name="host"]').val().trim();

const usernameError = $('#missatge-username').text() !== '';

const passwordValid = validarPassword(password);

const nomValid = validarNom(nom);

const campsPlens = nom && username && password && host;

$('#missatge-nom').text(

nomValid || nom.length === 0 ? '' : 'El nom només pot contenir lletres i espais.');

if (!validarPassword(password) && password.length > 0) {

$('#missatge-password').text(

'Contrasenya: mínim 8 caràcters, una majúscula, una minúscula i un número.');

} else {

$('#missatge-password').text('');

}

const disable = usernameError || !passwordValid || !nomValid || !campsPlens;

$('button[type="submit"]').prop('disabled', disable);

}

$('#nom, #username, #password, input[name="host"]').on('input', function () {

comprovarSubmit();

});

$('#username').on('input', function () {

const username = $(this).val();

const host = $('input[name="host"]').val();

if (username.length > 2) {

$.post('comprovar\_username.php', {username: username, host: host}, function (data) {

if (data === 'existeix') {

$('#missatge-username').text('Aquest username ja existeix!');

} else {

$('#missatge-username').text('');

}

comprovarSubmit();

});

} else {

$('#missatge-username').text('');

comprovarSubmit();

}

});

});

</script>

</body>

</html>

**✅ Amb aquest formulari ja tens:**

* Estil modern i adaptat a dispositius.
* Validacions en temps real:
  + Nom sense números ni símbols.
  + Contrasenya segura.
  + Username únic (AJAX).
* Botó **activat/desactivat automàticament**.

Vols que també afegim un **missatge bonic de confirmació** quan s’inserta correctament? (amb modal, toast o alerta?)

|  |
| --- |
| **si, amb modal** |

Perfecte! Ho farem amb **Bootstrap Modal** perquè quedi elegant:  
quan l'usuari envia el formulari i la inserció té èxit, obrirem una **finestra modal de confirmació**.

**✏️ Com ho farem:**

1. Afegirem una **modal oculta** al teu HTML.
2. Farem que, després de fer submit, aparegui la modal **sense recarregar la pàgina**.
3. Si vols, també podem redirigir després d'uns segons.

**📄 Actualitzem formulari\_personal.html**

Afegim la **modal de confirmació**:

<!-- Modal de confirmació -->

<div class="modal fade" id="modalConfirmacio" tabindex="-1" aria-labelledby="modalConfirmacioLabel" aria-hidden="true">

<div class="modal-dialog modal-dialog-centered">

<div class="modal-content rounded-4 shadow">

<div class="modal-header bg-success text-white rounded-top-4">

<h5 class="modal-title" id="modalConfirmacioLabel">Inserció completada</h5>

<button type="button" class="btn-close btn-close-white" data-bs-dismiss="modal"

aria-label="Tanca"></button>

</div>

<div class="modal-body">

El nou personal s'ha afegit correctament!

</div>

<div class="modal-footer">

<button type="button" class="btn btn-success" data-bs-dismiss="modal">Acceptar</button>

</div>

</div>

</div>

</div>

**🔥 Ara actualitzem el submit per mostrar la modal:**

Afegim aquest script **després** del $(document).ready(...) que ja tens:

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

<script>

$(document).ready(function() {

$('#formulari').submit(function(e) {

e.preventDefault(); // Evita que es recarregui la pàgina

$.ajax({

type: 'POST',

url: 'inserir\_personal.php', // el teu fitxer PHP que fa l'INSERT

data: $(this).serialize(),

success: function(resposta) {

if (resposta.trim() === 'ok') {

const modalConfirmacio =

new bootstrap.Modal(document.getElementById('modalConfirmacio'));

modalConfirmacio.show();

$('#formulari')[0].reset(); // Neteja el formulari

$('button[type="submit"]').prop('disabled', true); // Torna a desactivar el botó

} else {

alert('Error al desar: ' + resposta);

}

},

error: function() { alert('Error del servidor.'); }

});

});

});

</script>

**📜 Important:**

**En inserir\_personal.php**:

* Si la inserció té èxit, cal que imprimeixis només ok.

Exemple mínim:

<?php

// Connexió a la BBDD...

// Recollir dades...

if ($stmt = $conn->prepare('INSERT INTO personal (...) VALUES (?, ?, ?, ?, ?, ?)')) {

$stmt->bind\_param(...);

if ($stmt->execute()) {

echo 'ok'; // Molt important!

} else {

echo 'Error en l\'inserció.';

}

} else {

echo 'Error de preparació.';

}

?>

**🧹 Resum:**

* El formulari **no recarrega la pàgina**.
* Si tot va bé, **apareix la modal**.
* Si falla, es mostra un **alert** amb l’error.

💬 **Vols també que afegim una redirecció automàtica a una pàgina d’"administrar personal" després de tancar la modal?** (És molt fàcil i queda molt professional!) 🚀  
Voldries això?

|  |
| --- |
| **si** |

Perfecte! Anem a fer-ho **molt polit**:

Quan l'usuari tanqui la **modal de confirmació**, el redirigirem automàticament a una altra pàgina, per exemple, una pàgina com llista\_personal.php.

**🔥 Actualització final:**

Afegim això **just dins** de la funció success de l'$.ajax() (on ja mostrem la modal):

if (resposta.trim() === 'ok') {

const modalConfirmacio = new bootstrap.Modal(document.getElementById('modalConfirmacio'));

modalConfirmacio.show();

$('#formulari')[0].reset();

$('button[type="submit"]').prop('disabled', true);

// Quan es tanqui la modal, redirigim

const modalElement = document.getElementById('modalConfirmacio');

modalElement.addEventListener('hidden.bs.modal', function () {

window.location.href = 'llista\_personal.php'; // <<< aquí poses la pàgina on vols anar

});

}

**📌 Explicació ràpida:**

* Mostrem la **modal** (modalConfirmacio.show()).
* Esperem que **l'usuari tanqui la modal** (clicant el botó o la X).
* Quan es tanqui (hidden.bs.modal), **fem un window.location.href** cap a la nova pàgina (llista\_personal.php o la que vulguis).

**✅ Flux final:**

1. L'usuari omple el formulari.
2. S’envia via AJAX.
3. Si tot va bé ➔ **Modal de confirmació**.
4. Quan tanquen la modal ➔ **Redirecció automàtica** a la pàgina desitjada.

**Vols que també et prepari una petita pàgina de llista\_personal.php que mostri els usuaris en una taula bonica amb Bootstrap? 🚀**

(seria perfecte per tenir el sistema complet!)

|  |
| --- |
| **si** |

|  |
| --- |
| **el pwdhash es genera bé, però no funciona al fer creat user:** |

Aquí tens el motiu pel qual **no funciona** el teu CREATE USER tot i que el *hash* es genera correctament:

**🔴 Problema:**

IDENTIFIED BY **no accepta un hash** com a contrasenya directa.

Quan fas:

CREATE USER 'usuari'@'localhost' IDENTIFIED BY 'alguna\_cosa\_hash';

MySQL **interpreta 'alguna\_cosa\_hash' com una contrasenya en text pla**, no com un hash.

**✅ Solucions possibles:**

**✔️ Opcion 1: Passar la contrasenya en text pla (MySQL la converteix a hash automàticament):**

Si estàs usant PASSWORD() o un SHA-256 tu mateix abans, **no ho facis**. En lloc d'això, fes:

$password = 'la\_contrasenya\_en\_text'; // no el hash

$stmt\_user = "CREATE OR REPLACE USER '$username'@'$host' IDENTIFIED BY '$password' PASSWORD EXPIRE NEVER;";

$conn->query($stmt\_user);

⚠️ El servidor MySQL aplicarà el hash internament amb l’algoritme que tingui per defecte (caching\_sha2\_password en MySQL 8.0).

**❌ NO fer:**

IDENTIFIED BY '$pwdhash' // si $pwdhash és el resultat de password\_hash() o PASSWORD()

Això no funcionarà perquè MySQL no reconeix hashes externs com a valors vàlids per a IDENTIFIED BY.

**🔐 Vols forçar un hash tu?**

Només es pot fer si uses IDENTIFIED WITH 'plugin' AS 'hash', i és complicat i desaconsellat tret que sàpigues exactament quin *plugin d'autenticació* fa servir MySQL (ex. mysql\_native\_password).

Exemple (no recomanat):

CREATE USER 'usuari'@'localhost'

IDENTIFIED WITH mysql\_native\_password AS '\*HASHED\_PASSWORD';

Però \*HASHED\_PASSWORD ha de ser un hash SHA1 **en format MySQL** (comença per \* i 40 caràcters hex).

**✅ Recomanació final:**

Passa la contrasenya **en text pla** i deixa que MySQL la gestioni:

$password = 'secret123';

$username = $conn->real\_escape\_string($username);

$host = $conn->real\_escape\_string($host);

$sql = "CREATE OR REPLACE USER '$username'@'$host' IDENTIFIED BY '$password' PASSWORD EXPIRE NEVER;";

$conn->query($sql);

Vols que t'ajudi a integrar-ho de forma segura, amb validació del formulari?

|  |
| --- |
| **i com obtinc el pwdhash que genera mysql en PHP?** |

Perfecte! Aquí tens un **exemple complet en PHP** que:

1. Crea un usuari MySQL amb una contrasenya donada.
2. Després recupera el **hash de la contrasenya** des de la taula mysql.user (si tens permisos suficients).

**✅ Exemple PHP: crear\_usuari\_i\_obtenir\_hash.php**

<?php

$host = 'localhost';

$admin\_user = 'root';

$admin\_pass = 'la\_teva\_contrasenya\_de\_root';

$conn = new mysqli($host, $admin\_user, $admin\_pass);

if ($conn->connect\_error) {

die("Error de connexió: " . $conn->connect\_error);

}

$username = 'nou\_usuari';

$userhost = 'localhost';

$password = 'contrasenyaSegura123';

// Escapem els valors

$username\_esc = $conn->real\_escape\_string($username);

$userhost\_esc = $conn->real\_escape\_string($userhost);

$password\_esc = $conn->real\_escape\_string($password);

// 1. Crear usuari

$sql\_create = "CREATE OR REPLACE USER '$username\_esc'@'$userhost\_esc' IDENTIFIED BY

'$password\_esc' PASSWORD EXPIRE NEVER;";

if ($conn->query($sql\_create) === TRUE) {

echo "Usuari creat correctament.<br>";

} else {

die("Error en crear l'usuari: " . $conn->error);

}

// 2. Recuperar el hash

$sql\_hash = "SELECT authentication\_string FROM mysql.user WHERE user = ? AND host = ?";

$stmt = $conn->prepare($sql\_hash);

$stmt->bind\_param("ss", $username, $userhost);

$stmt->execute();

$stmt->bind\_result($auth\_hash);

$stmt->fetch();

$stmt->close();

if ($auth\_hash) {

echo "Hash MySQL: <code>$auth\_hash</code>";

} else {

echo "No s'ha pogut obtenir el hash.";

}

$conn->close();

?>

**⚠️ Requisits:**

* El compte que fa la connexió (ex: root) ha de tenir permisos per consultar mysql.user.
* Assegura’t que el mode d’autenticació és compatible (caching\_sha2\_password o mysql\_native\_password).

Vols que també assigni privilegis a aquest usuari automàticament després de crear-lo?

|  |
| --- |
|  |