



Git



Apache



MariaDB

# Treball amb repositori git i desplegament inicial

Grup 3

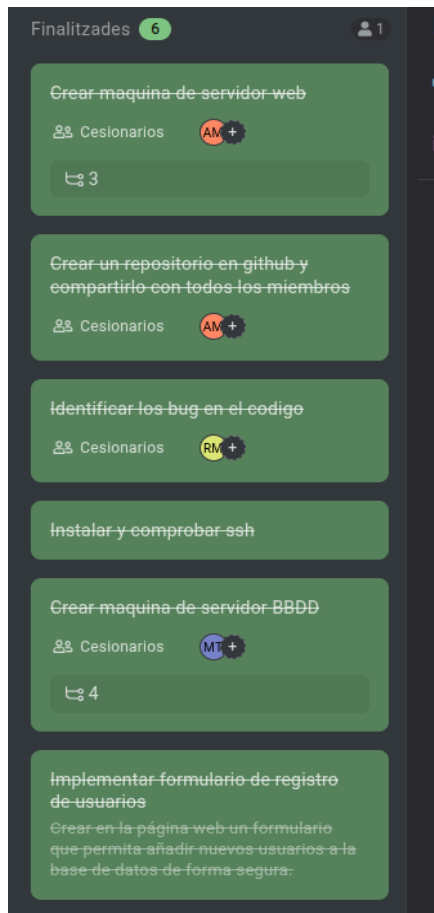
Mario Alaez  
Trishan Mizhquiri  
Manel Reyes

📅 13/10/2025

# INDEX

<b>0. Organització</b>	<b>3</b>
<b>1. Correcció codi</b>	<b>4</b>
1.1. db.php	4
1.2 index.php	4
1.3 add.php	6
1.4 delete.php	7
1.5 edit.php	8
<b>2. Configuració instal·lació servidor web més php</b>	<b>11</b>
2.1. Configuración adaptador	11
2.2. Instal·lació de serveis	11
2.3. Ús de la tecnologia git	12
2.4. Configuració de l'arxiu db.php	12
2.5. Ajustar permisos	13
<b>3. Configuració d'un servidor de bases de dades amb MariaDB en Debian</b>	<b>14</b>
3.1. Configuració de la xarxa	14
3.1.1. Afegiment d'una interfície de xarxa addicional	14
3.1.2. Configuració de la interfície estàtica	14
3.1.3. Reinici del servei de xarxa	14
3.2. Actualització del sistema	15
3.3. Instal·lació de MariaDB	15
3.3.1. Accés al monitor de MariaDB	15
3.3.2. Creació de la base de dades	15
3.3.3. Configuración del servidor de BBDD	16
3.3.4. Creació de taules per la BBDD	16
3.3.5. Creació de l'usuari i permisos	16
<b>4. Arquitectura i Desplegament</b>	<b>17</b>
4.1 Desplegament	17
4.2. Arquitectura	18

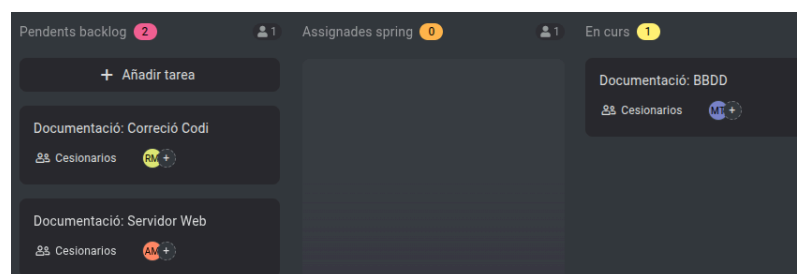
## 0. Organització



Ens hem organitzat amb el proofhub. Hem dividit les tasques per a cada persona. Una persona s'encarregava de corregir el codi per després en una maquina de prova amb tots el serveis configurats fer el test per veure si tot anava bé. Per després compartir-lo amb el resta del equipo.

Per una altra part, les dues persones s'encarregaven en crear en dues màquines, el servidor web amb apache, i l'altre server amb la BBDD. Ho hem fet així, perquè considerem que evitem que algú és quedes sense fer res i evitar errors que puguin afectar completament al servidor.

Una vegada aconseguït tot i la base de dades funcionant, es feia el pull a la servidor web amb la versió final.



# 1. Correcció codi

## 1.1. db.php

En la primera línia, es veu que el nom del servidor està mal escrit, posa localhost i es localhost.

```
$servername = "localhost";
$username = "usuario_crud";
// MEJORA: Es recomendable usar un usuario específico para la app en lugar de 'root'.
// Deja la contraseña vacía "" si usas una instalación por defecto de XAMPP/WAMP.
$password = "pirineus";
$dbname = "crud_db";

// Crear conexión usando el paradigma orientado a objetos de mysqli
$conn = new mysqli($servername, $username, $password, $dbname);

// Establecer el charset a utf8mb4 para soportar caracteres especiales y emojis
$conn->set_charset("utf8mb4");

// Verificar si la conexión ha fallado
if ($conn->connect_error) {
    // die() detiene la ejecución del script y muestra un mensaje de error.
    // Es más seguro no mostrar detalles del error en producción.
    die("Connexió fallida: " . $conn->connect_error);
}
?>
```

## 1.2 index.php

- Correcció (HTML): Hi havia dues etiquetes <table> niades. S'ha eliminat l'etiqueta supèrflua.

```
<!-- CORRECCIÓN: Había dos etiquetas <table> anidadas. Se ha dejado solo una. -->
<table border="1">
```

- Correcció (HTML): El mètode del formulari era method="posts". S'ha corregit a l'estàndard method="post".

```
<!-- CORRECCIÓN: El método del formulario debe ser "post" (en minúsculas y singular).
<form action="add.php" method="post">
    Nom: <input type="text" name="name" required>
    Email: <input type="email" name="email" required>
    <button type="submit">Afegir</button>
</form>
```

- Millora (Usabilitat): S'ha afegit un onclick='return confirm(...)' a l'enllaç d' "Eliminar". Això mostra un quadre de diàleg de confirmació per evitar que els usuaris esborrin registres per accident.
- Millora (Seguretat): Com s'ha esmentat abans, totes les dades que venen de la base de dades (\$row['id'], \$row['name'], etc.) s'emboquen amb htmlspecialchars() per prevenir atacs XSS

```
// MEJORA: Comprobamos si la consulta devolvió alguna fila.
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        echo "<tr>
            <!-- MEJORA (Seguridad): Usamos htmlspecialchars() para prevenir ataques
            <!-- Esto convierte caracteres especiales como < > & en entidades HTML,
            <td>" . htmlspecialchars($row['id']) . "</td>
            <td>" . htmlspecialchars($row['name']) . "</td>
            <td>" . htmlspecialchars($row['email']) . "</td>
            <td>
                <a href='edit.php?id=" . htmlspecialchars($row['id']) . "'>Editar</a>
            <!-- MEJORA (Usabilidad): Se añade una confirmación en JavaScript pa
            <a href='delete.php?id=" . htmlspecialchars($row['id']) . "' onclick=
            </td>
        </tr>
```

- Millora (Robustesa): S'ha afegit una comprovació (if (\$result->num\_rows > 0)) per mostrar un missatge amigable ("No s'han trobat usuaris.") quan la taula està buida, en lloc de mostrar una taula sense files.

### Codi final amb altres millores.

```
<?php
// Incluimos la conexión a la base de datos
include 'db.php';
?>
<!DOCTYPE html>
<html lang="ca">
<head>
    <meta charset="UTF-8">
    <title>CRUD mínim</title>
</head>
<body>
    <h1>Llista d'usuaris</h1>

    <!-- CORRECCIÓN: Había dos etiquetas <table> anidadas. Se ha dejado solo una. -->
    <table border="1">
        <tr>
            <th>ID</th>
            <th>Nom</th>
            <th>Email</th>
            <th>Accions</th>
        </tr>
    <?php
    // MEJORA: Es buena práctica seleccionar solo las columnas que necesitas.
    $result = $conn->query("SELECT id, name, email FROM users");

    // MEJORA: Comprobamos si la consulta devolvió alguna fila.
    if ($result->num_rows > 0) {
        while ($row = $result->fetch_assoc()) {
            echo "<tr>
                <!-- MEJORA (Seguridad): Usamos htmlspecialchars() para prevenir ataques XSS (Cross-Site Scripting). -->
                <!-- Esto convierte caracteres especiales como < > & en entidades HTML, evitando que se ejecute código malicioso. -->
                <td>" . htmlspecialchars($row['id']) . "</td>
                <td>" . htmlspecialchars($row['name']) . "</td>
                <td>" . htmlspecialchars($row['email']) . "</td>
                <td>
                    <a href='edit.php?id=" . htmlspecialchars($row['id']) . "'>Editar</a> |
                <!-- MEJORA (Usabilidad): Se añade una confirmación en JavaScript para evitar borrados accidentales. -->
                <a href='delete.php?id=" . htmlspecialchars($row['id']) . "' onclick='return confirm(\"Estás seguro que vols eliminar aquest usuari?\");'>Eliminar</a>
                </td>
            </tr>";
        }
    } else {
        echo "No s'han trobat usuaris.";
```

```

    } else {
        // MEJORA: Se muestra un mensaje si no hay usuarios en la base de datos.
        echo "<tr><td colspan='4'>No s'han trobat usuaris.</td></tr>";
    }
?>
</table>

<h2>Afegir usuari</h2>
<!-- CORRECCIÓ: El método del formulario debe ser "post" (en minúsculas y singular). -->
<form action="add.php" method="post">
    Nom: <input type="text" name="name" required>
    Email: <input type="email" name="email" required>
    <button type="submit">Afegir</button>
</form>
</body>
</html>

<?php
// MEJORA: Es buena práctica cerrar la conexión a la base de datos al final del script.
$conn->close();
?>

```

### 1.3 add.php

- Correcció (Sintaxi SQL): La consulta preparada original era VALUES (\*, ?), la qual cosa és incorrecta. S'ha corregit a VALUES (?, ?), fent servir un marcador de posició per cada valor a inserir.
- Correcció (Seguretat): S'ha implementat correctament la sentència preparada per evitar injecció SQL, vinculant les variables \$name i \$email amb bind\_param("ss", ...). El "ss" indica que ambdues variables són de tipus string (cadena de text).

**Codi final amb altres millores**

```

<?php
// Incluimos la conexión a la base de datos
include 'db.php';

// MEJORA: Verificamos que los datos se envían por el método POST para evitar errores.
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Recogemos los datos del formulario
    $name = $_POST['name'];
    $email = $_POST['email'];

    /*
    * CORRECCIÓN (Seguridad y Sintaxis):
    * 1. Usamos sentencias preparadas para prevenir INYECCIÓN SQL. Esto es CRÍTICO.
    * Nunca se deben insertar variables directamente en una consulta SQL.
    * 2. La sintaxis SQL era incorrecta. "VALUES (*, ?)" se ha corregido a "VALUES (?, ?)",
    * usando un marcador de posición (?) por cada valor que se va a insertar.
    */
    $stmt = $conn->prepare("INSERT INTO users (name, email) VALUES (?, ?)");

    // Vinculamos las variables a los marcadores de posición.
    // "ss" significa que ambas variables ($name, $email) son de tipo string.
    $stmt->bind_param("ss", $name, $email);

    // Ejecutamos la consulta preparada
    $stmt->execute();

    // MEJORA: Cerramos la sentencia para liberar recursos.
    $stmt->close();
}

// Cerramos la conexión a la base de datos
$conn->close();

// Redirigimos al usuario de vuelta a la página principal
header("Location: index.php");
// exit() asegura que no se ejecute más código después de la redirección.
exit;
?>

```

## 1.4 delete.php

Correcció (Sintaxis i seguretat): La sintaxi SQL es incorrecta, “DELETE \* FROM” s’ha canviat a “DELETE FROM”. A més, s'utilitza una sentència per prevenir injecció SQL.

***codi final amb millores i correcció***

```

<?php
include 'db.php';

// Verificamos que se ha pasado un 'id' por la URL.
if (isset($_GET['id'])) {
    $id = (int)$_GET['id']; // Convertimos el ID a entero por seguridad.

    /*
     * CORRECCIÓN (Sintaxis y Seguridad):
     * 1. La sintaxis SQL era incorrecta. "DELETE * FROM" se ha cambiado a "DELETE FROM".
     * 2. Se usa una sentencia preparada para prevenir inyección SQL, la forma más segura.
     */
    $stmt = $conn->prepare("DELETE FROM users WHERE id = ?");
    $stmt->bind_param("i", $id); // "i" indica que el ID es de tipo integer.
    $stmt->execute();
    $stmt->close();
}

// Cerramos la conexión
$conn->close();

// Redirigimos al usuario a la página principal
header("Location: index.php");
exit;
?>

```

## 1.5 edit.php

Correcció (Sintaxis i seguretat): La sintaxis de UPDATE es incorrecta. “where name=?” se ha cambiado por “SET name = ?” També, s'utilitza una sentencia preparada per l'actualització, prevenint l'inyecció SQL.

```

/*
 * CORRECCIÓN (Sintaxis y Seguridad):
 * 1. La sintaxis de UPDATE era incorrecta. "where name=?" se ha cambiado por "SET name = ?"
 * 2. Se usa una sentencia preparada para la actualización, previniendo inyección SQL.
 */
$stmt = $conn->prepare("UPDATE users SET name = ?, email = ? WHERE id = ?");

// "ssi" indica los tipos de las variables: string, string, integer.
$stmt->bind_param("ssi", $name, $email, $id);
$stmt->execute();
$stmt->close();

```



## Codi final y altres millores

```
<?php
include 'db.php';

$user = null; // Variable para almacenar los datos del usuario a editar

// --- PARTE 1: OBTENER DATOS DEL USUARIO PARA MOSTRARLOS EN EL FORMULARIO ---
// Verificamos si se ha pasado un 'id' por la URL (método GET)
if (isset($_GET['id'])) {
    $id = (int)$_GET['id']; // Convertimos el ID a entero por seguridad.

    /*
     * CORRECCIÓN (Seguridad): Se usa una sentencia preparada para evitar inyección SQL
     * al obtener los datos del usuario.
     */
    $stmt = $conn->prepare("SELECT * FROM users WHERE id = ?");
    $stmt->bind_param("i", $id); // "i" indica que el ID es de tipo integer.
    $stmt->execute();
    $result = $stmt->get_result(); // Obtenemos el resultado
    $user = $result->fetch_assoc(); // Lo guardamos en un array asociativo
    $stmt->close();
}

// --- PARTE 2: ACTUALIZAR LOS DATOS CUANDO SE ENVÍA EL FORMULARIO ---
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $id = (int)$_POST['id'];
    $name = $_POST['name'];
    $email = $_POST['email'];

    /*
     * CORRECCIÓN (Sintaxis y Seguridad):
     * 1. La sintaxis de UPDATE era incorrecta. "where name=?" se ha cambiado por "SET name = ?".
     * 2. Se usa una sentencia preparada para la actualización, previniendo inyección SQL.
     */
    $stmt = $conn->prepare("UPDATE users SET name = ?, email = ? WHERE id = ?");

    // "ssi" indica los tipos de las variables: string, string, integer.
    $stmt->bind_param("ssi", $name, $email, $id);
    $stmt->execute();
    $stmt->close();

    // Redirigimos al index
    header("Location: index.php");
    exit;
}
```

```

}
?>

<!DOCTYPE html>
<html lang="ca">
<head>
    <meta charset="UTF-8">
    <title>Editar usuari</title>
</head>
<body>
    <h1>Editar usuari</h1>

    <?php if ($user): // MEJORA: Solo mostramos el formulario si el usuario fue encontrado ?>
    <form method="post">
        <input type="hidden" name="id" value="<?= htmlspecialchars($user['id']) ?>">
        Nom:
        <!-- MEJORA (Seguridad): Usamos htmlspecialchars para prevenir XSS al mostrar los datos existentes. -->
        <input type="text" name="name" value="<?= htmlspecialchars($user['name']) ?>" required>
        Email:
        <input type="email" name="email" value="<?= htmlspecialchars($user['email']) ?>" required>
        <button type="submit">Desar</button>
    </form>
    <?php else: // MEJORA: Mensaje si el usuario con ese ID no existe. ?>
        <p>No s'ha trobat l'usuari. <a href="index.php">Tornar a la llista</a>.</p>
    <?php endif; ?>
</body>
</html>

```

## 2. Configuració instal·lació servidor web més php

### 2.1. Configuración adaptador

El primer pas és configurar l'adaptador de xarxa ASIXC2-ITB15

```
GNU nano 2.9.4 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp1s0
iface enp1s0 inet dhcp

auto enp2s0
iface enp2s0 inet static
    address 192.168.1.1
    netmask 255.255.255.0
```

```
isard@debian:~$ ifconfig enp2s0
enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:0c:0d:9a brd ff:ff:ff:ff:ff:ff
    altname enx5254000c0d9a
    inet 192.168.1.1/24 brd 192.168.1.255 scope global enp2s0
        valid_lft forever preferred_lft forever
    inet6 fe80::5054:ff:fe0c:d9a/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
isard@debian:~$
```

Una vegada comprovem que funcione i fem ping el següent pas es instal·lar apache y php

```
-bash: pinh: no s'ha trobat l'ordre
isard@debian:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data:
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=4.23 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=1.76 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=1.63 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=1.53 ms
```

### 2.2. Instal·lació de serveis

**sudo apt install apache2 php libapache2-mod-php php-mysql -y**

```
isard@debian:~$ sudo apt install apache2 php libapache2-mod-php php-mysql
apache2 ja està en la versió més recent (2.4.65-2).
php ja està en la versió més recent (2:8.4+96).
libapache2-mod-php ja està en la versió més recent (2:8.4+96).
php-mysql ja està en la versió més recent (2:8.4+96).
Els paquets següents s'han instal·lat automàticament i ja no són necessaris:
  linux-image-6.12.38+deb13-amd64 linux-image-6.12.41+deb13-amd64
Empreu «sudo apt autoremove» per a suprimir-los.

Resum:
  S'està actualitzant: 0, s'està instal·lant: 0, S'està suprimint: 0, no s'està actualitzant: 0
isard@debian:~$
```

## 2.3. Ús de la tecnologia git

Una vegada instal·lat tot el següent pas es crear els directoris corresponents fem un `mkdir -p /var/www/html/app` y utilitzarem l'ordre `git clone` per clonar tot el repositori i utilitzarem la comanda `mv` per moure els arxius php dintre de la carpeta `app`

```
isard@debian:~$ sudo git clone https://github.com/MarioAlaez-ITB2425/G3_MarioAlaez_TrishanMizhquiri_ManelReyes.git
S'està clonant a «G3_MarioAlaez_TrishanMizhquiri_ManelReyes»...
remote: Enumerating objects: 43, done.
remote: Counting objects: 100% (43/43), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 43 (delta 11), reused 21 (delta 6), pack-reused 0 (from 0)
S'estan rebent objectes: 100% (43/43), 13.96 KiB | 477.00 KiB/s, fet.
S'estan resolent les diferències: 100% (11/11), fet.
isard@debian:~$
```

```
isard@debian:~$ ls -la /var/www/html/app/
total 28
drwxr-xr-x 2 www-data www-data 4096 6 d'oct. 16:46 .
drwxr-xr-x 3 root     root     4096 6 d'oct. 15:56 ..
-rwxr-xr-x 1 www-data www-data 1359 6 d'oct. 15:36 add.php
-rwxr-xr-x 1 www-data www-data  937 6 d'oct. 16:46 db.php
-rwxr-xr-x 1 www-data www-data  733 6 d'oct. 15:36 delete.php
-rwxr-xr-x 1 www-data www-data 2537 6 d'oct. 15:36 edit.php
-rwxr-xr-x 1 www-data www-data 2499 6 d'oct. 15:36 index.php
isard@debian:~$ _
```

## 2.4. Configuració de l'arxiu db.php

Ara el que fa falta es modificar l'arxiu `db.php` i hem de posar el nom de la base de dades l'usuari la password i la ip de la màquina que te la BBDD

```
<?php
// Fichero de conexión a la Base de Datos

// CORRECCIÓN: El nombre del servidor era "locahost" y ha sido corregido a "localhost".
$servername = "192.168.1.2";
$username = "root";
// MEJORA: Es recomendable usar un usuario específico para la app en lugar de 'root'.
// Deja la contraseña vacía "" si usas una instalación por defecto de XAMPP/WAMP.
$password = "root";
$dbname = "Error404_TeamNotFound";

// Crear conexión usando el paradigma orientado a objetos de mysqli
$conn = new mysqli($servername, $username, $password, $dbname);

// Establecer el charset a utf8mb4 para soportar caracteres especiales y emojis
$conn->set_charset("utf8mb4");

// Verificar si la conexión ha fallado
if ($conn->connect_error) {
    // die() detiene la ejecución del script y muestra un mensaje de error.
    // Es más seguro no mostrar detalles del error en producción.
    die("Connexió fallida: " . $conn->connect_error);
}
?>
```

## 2.5. Ajustar permisos

Ya per últim s'han ajustat uns permisos amb les comandes

**sudo chown -R www-www-data /var/www/html/app**

**sudo chmod -R 755 /var/www/html/app**

```
isard@debian:/var/www/html$ sudo chown -R www-data:www-data /var/www/html/app/  
isard@debian:/var/www/html$ sudo chmod -R 755 /var/www/html/app/  
isard@debian:/var/www/html$ _
```

### 3. Configuració d'un servidor de bases de dades amb MariaDB en Debian

#### 3.1. Configuració de la xarxa

##### 3.1.1. Afegiment d'una interfície de xarxa addicional

Des del panell d'administració d'IsardVDI, s'ha afegit una segona interfície de xarxa del tipus ASIXC2-ITB15 a la màquina virtual. Un cop afegida, s'ha verificat la disponibilitat de les dues interfícies mitjançant el comandament: **ip a**

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:78:3c:b8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.79/22 metric 100 brd 192.168.123.255 scope global dynamic enp1s0
        valid_lft 3578sec preferred_lft 3578sec
    inet6 fe80::5054:ff:fe78:3cb8/64 scope link
        valid_lft forever preferred_lft forever
3: enp2s0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 52:54:00:5f:42:db brd ff:ff:ff:ff:ff:ff
```

##### 3.1.2. Configuració de la interfície estàtica

S'ha editat el fitxer de configuració de xarxa: **sudo nano /etc/network/interfaces** s'hi ha inserit la configuració següent:

```
# The secondary network interface
auto enp2s0
iface enp2s0 inet static
    address 192.168.1.2
    netmask 255.255.255.0
```

##### 3.1.3. Reinici del servei de xarxa

Un cop guardada la configuració, s'ha reiniciat el servei de xarxa per aplicar els canvis:

**sudo systemctl restart networking**

S'ha verificat novament la configuració amb **ip a** per assegurar que la interfície **enp2s0** disposa de l'adreça IP 192.168.1.2/24.

```
isard@debian:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:0c:64:b6 brd ff:ff:ff:ff:ff:ff
    altname enx5254000c64b6
    inet 192.168.123.191/22 brd 192.168.123.255 scope global dynamic noprefixroute enp1s0
        valid_lft 3514sec preferred_lft 2940sec
    inet6 fe80::cca4:7094:4baa:8ff7/64 scope link
        valid_lft forever preferred_lft forever
3: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:7b:df:66 brd ff:ff:ff:ff:ff:ff
    altname enx5254007bdf66
    inet 192.168.1.2/24 brd 192.168.1.255 scope global enp2s0
        valid_lft forever preferred_lft forever
```

## 3.2. Actualització del sistema

Abans de procedir amb la instal·lació de paquets, s'ha actualitzat el sistema per garantir la disponibilitat de les darreres actualitzacions de seguretat i estabilitat: **sudo apt update**

```
isard@debian:~$ sudo apt update
Obj:1 http://security.debian.org/debian-security trixie-security InRelease
Obj:2 http://deb.debian.org/debian trixie InRelease
Bai:3 http://deb.debian.org/debian trixie-updates InRelease [47,3 kB]
S'ha baixat 47,3 kB en 0s (172 kB/s)
```

## 3.3. Instal·lació de MariaDB

S'ha instal·lat el servidor de bases de dades MariaDB mitjançant el gestor de paquets d'APT: **sudo apt install mariadb-server -y**

```
isard@debian:~$ sudo apt install mariadb-server
mariadb-server ja està en la versió més recent (1:11.8.3-0+deb13u1).
El paquet següent s'ha instal·lat automàticament i ja no és necessari:
  linux-image-6.12.38+deb13-amd64
```

La versió instal·lada és la **11.8.3-MariaDB**, proporcionada pels repositoris oficials de Debian 13 (Trixie).

### 3.3.1. Accés al monitor de MariaDB

Un cop finalitzada la instal·lació, s'ha accedit al monitor de MariaDB com a superusuari: **sudo mariadb**

```
isard@debian:~$ sudo mariadb
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 48
Server version: 11.8.3-MariaDB-0+deb13u1 from Debian -- Please help get to 10k stars at https://github.com/MariaDB/Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> _
```

### 3.3.2. Creació de la base de dades

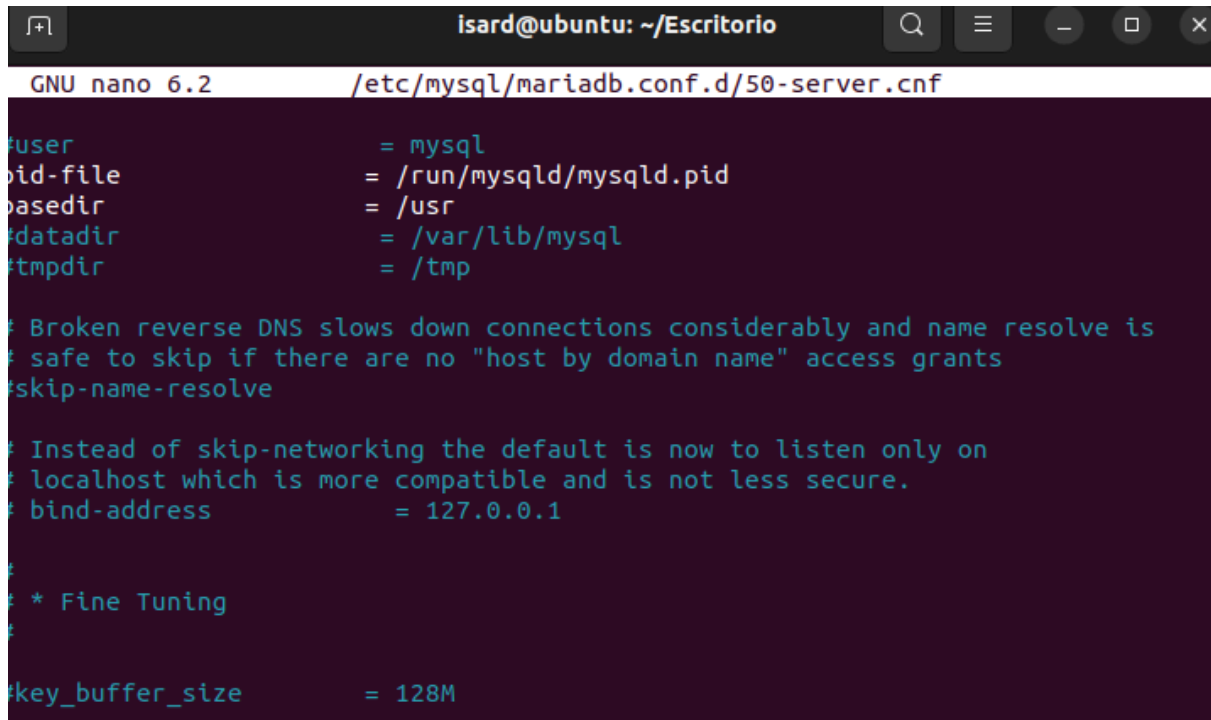
Dins de l'entorn interactiu de MariaDB, s'ha creat una base de dades amb un nom identificatiu per al projecte: **CREATE DATABASE Error404\_TeamNotFound;**

Aquesta base de dades servirà com a repositori central per a les dades de l'aplicació associada al projecte.

```
MariaDB [(none)]> CREATE DATABASE Error404_TeamNotFound;
```

### 3.3.3. Configuración del servidor de BBDD

El que fem és configurar mysql de tal forma que el servidor escolta totes les IPs



```
isard@ubuntu: ~/Escritorio
GNU nano 6.2 /etc/mysql/mariadb.conf.d/50-server.cnf

#user                        = mysql
#pid-file                    = /run/mysqld/mysqld.pid
#basedir                    = /usr
#datadir                    = /var/lib/mysql
#tmpdir                     = /tmp

# Broken reverse DNS slows down connections considerably and name resolve is
# safe to skip if there are no "host by domain name" access grants
#skip-name-resolve

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
# bind-address               = 127.0.0.1

#
# * Fine Tuning
#

#key_buffer_size            = 128M
```

### 3.3.4. Creació de taules per la BBDD

Ara hem de crear la taula que utilitza la pàgina web

```
MariaDB [Error404_TeamNotFound]> CREATE TABLE users (
->     id INT AUTO_INCREMENT PRIMARY KEY,
->     name VARCHAR(100) NOT NULL,
->     email VARCHAR(100) NOT NULL
-> );
Query OK, 0 rows affected (0,023 sec)
```

### 3.3.5. Creació de l'usuari i permisos

El següent pas es crear l'usuari root per a la xarxa 192.168.1.0 i li donem permisos d'administrador

```
MariaDB [Error404_TeamNotFound]> CREATE USER 'root'@'192.168.1.%' IDENTIFIED BY 'root';
Query OK, 0 rows affected (0,004 sec)

MariaDB [Error404_TeamNotFound]> GRANT ALL PRIVILEGES ON Error404_TeamNotFound.* TO 'root'@'192.168.1.%'
;
Query OK, 0 rows affected (0,002 sec)

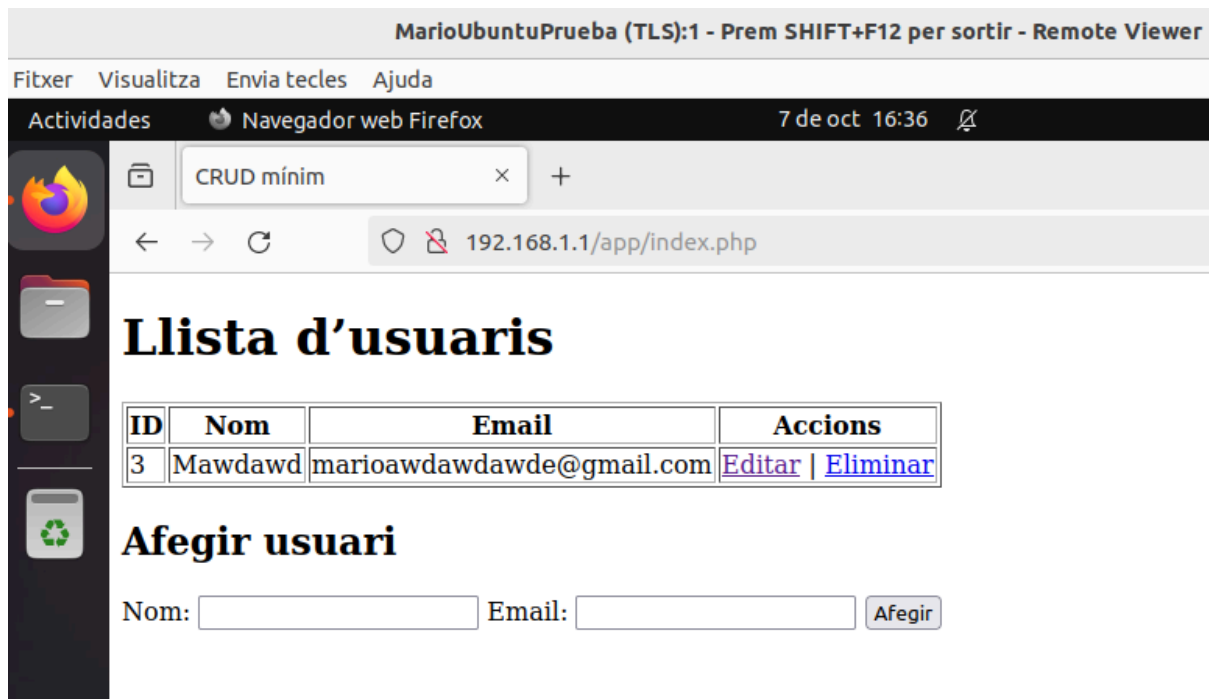
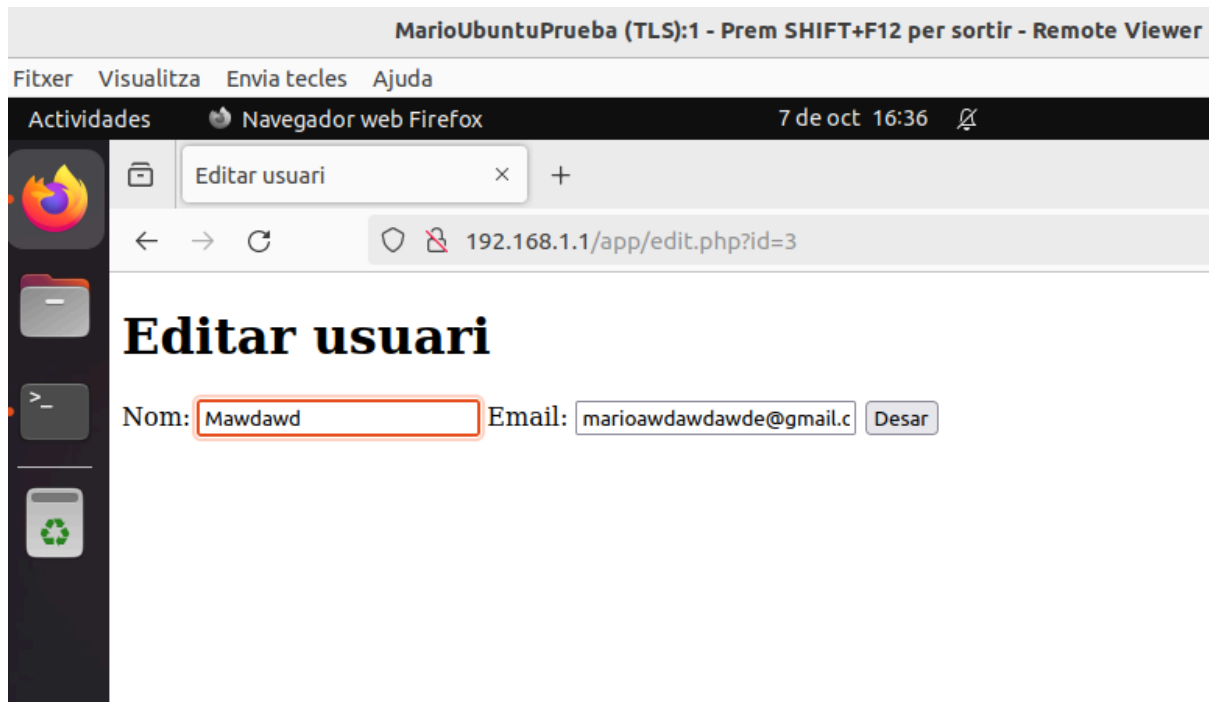
MariaDB [Error404_TeamNotFound]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,001 sec)
```



## 4. Arquitectura i Desplegament

### 4.1 Desplegament

Des del client, connectem amb Firefox al servidor web i, gràcies al PHP, podem modificar la pàgina i afegir dades com usuaris o editar-les.



## 4.2. Arquitectura

La arquitectura utilitzada és:

