

Implementación de un servidor de bases de datos

1. Instalación y configuración del servidor de bases de datos.

- **Instalar MySQL:** Con comando `apt install mysql-server`.
- **Verificar que está activo:** Con comando `systemctl status mysql`.
- **Configurarlo para que se inicie automáticamente al encender.**

2. Creación y gestión de bases de datos y usuarios.

- **Crear una base de datos llamada codearts_db.**
- **Crear los siguientes usuarios con los siguientes permisos:**
 - ☐ `admin_db` (Acceso total a la base de datos).
 - ☐ `dev_user` (Acceso de solo lectura y escritura).
 - ☐ `intern_user` (Acceso solo de lectura).
- **Asignar permisos específicos a cada usuario:** `GRANT PRIVILEGES`.

3. Conexión de una aplicación web a la base de datos.

- **Instalar Apache.**
- **Crear un script de conexión:** Que permita leer y escribir datos en la base de datos.
- **Realizar pruebas con consultas:** Con `SELECT`, `INSERT`, `UPDATE` y `DELETE` desde la aplicación web.

4. Seguridad y respaldo de datos.

- **Configurar accesos seguros restringiendo conexiones remotas:** `bind-address`.
 - **Implementar autenticación segura con políticas de contraseñas:** `mysql_secure_installation`.
 - **Crear un sistema de copias de seguridad automáticas:** `mysqldump`
 - **Registrar intentos de acceso fallidos y auditoría de consultas con logs del sistema:** `/var/log/mysql.log`.
-

1. Instalación y configuración del servidor de bases de datos.

Para instalar MySQL lo haremos con el comando:

sudo apt install mysql-server

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: ~  
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo apt install mysql-server  
[sudo] password for AZDINFARISSE:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... Hecho  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes adicionales:  
  libcgi-fast-perl libcgi-pm-perl libevent-pthreads-2.1-7t64 libfcgi-bin libfcgi-perl  
  libfcgi0t64 libhtml-template-perl libmecab2 libprotobuf-lite32t64 mecab-ipadic  
  mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common  
  mysql-server-8.0 mysql-server-core-8.0  
Paquetes sugeridos:  
  libipc-sharedcache-perl mailx tinyca  
Se instalarán los siguientes paquetes NUEVOS:  
  libcgi-fast-perl libcgi-pm-perl libevent-pthreads-2.1-7t64 libfcgi-bin libfcgi-perl  
  libfcgi0t64 libhtml-template-perl libmecab2 libprotobuf-lite32t64 mecab-ipadic  
  mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common  
  mysql-server mysql-server-8.0 mysql-server-core-8.0  
0 actualizados, 18 nuevos se instalarán, 0 para eliminar y 10 no actualizados.  
Se necesita descargar 29,2 MB de archivos.  
Se utilizarán 242 MB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [Y/n] s  
Des:1 http://es.archive.ubuntu.com/ubuntu noble/main amd64 mysql-common all 5.8+1.1.0build1  
[6.746 B]  
Des:2 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-core-8.0 amd
```

Ahora verificamos que el servicio está activo con el comando:

systemctl status mysql

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: ~  
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ systemctl status mysql  
● mysql.service - MySQL Community Server  
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)  
   Active: active (running) since Mon 2025-06-30 10:16:38 CEST; 1min 19s ago  
 Process: 5380 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0)  
 Main PID: 5388 (mysqld)  
   Status: "Server is operational"  
    Tasks: 37 (limit: 4605)  
  Memory: 363.7M (peak: 377.8M)  
     CPU: 811ms  
   CGroup: /system.slice/mysql.service  
           └─5388 /usr/sbin/mysqld  
lines 1-11/11 (END)
```

Lo siguiente es habilitar el servicio para que se inicie automáticamente al encender el servidor. Utilizaremos el comando:

sudo systemctl enable mysql

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: ~  
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo systemctl enable mysql  
[sudo] password for AZDINFARISSE:  
Synchronizing state of mysql.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install enable mysql  
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$
```

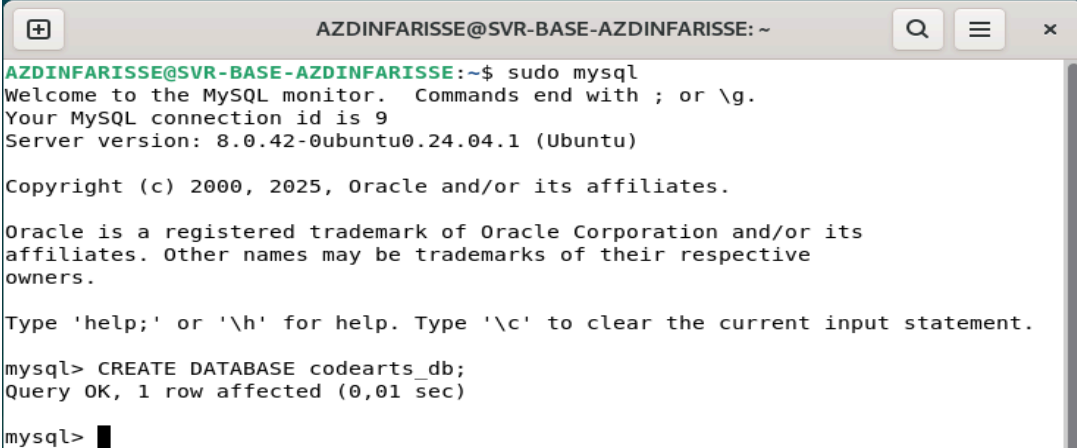
2. Creación y gestión de bases de datos y usuarios.

Para crear una base de datos primero accedemos a MySQL como root. Para ello utilizaremos el comando:

Sudo mysql

Y ahora crearemos la base de datos con el comando:

CREATE DATABASE codearts_db



```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: ~
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.42-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

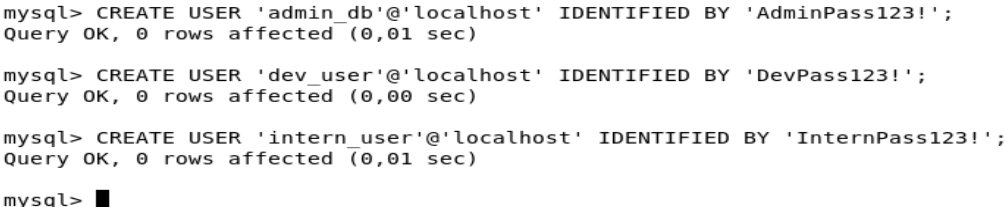
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE codearts_db;
Query OK, 1 row affected (0,01 sec)

mysql>
```

Ahora vamos a crear los usuarios. Para crearlos en mysql sería:

CREATE USER 'admin_db'@'localhost' IDENTIFIED BY 'AdminPass123!';
CREATE USER 'dev_user'@'localhost' IDENTIFIED BY 'DevPass123!';
CREATE USER 'intern_user'@'localhost' IDENTIFIED BY 'InternPass123!';



```
mysql> CREATE USER 'admin_db'@'localhost' IDENTIFIED BY 'AdminPass123!';
Query OK, 0 rows affected (0,01 sec)

mysql> CREATE USER 'dev_user'@'localhost' IDENTIFIED BY 'DevPass123!';
Query OK, 0 rows affected (0,00 sec)

mysql> CREATE USER 'intern_user'@'localhost' IDENTIFIED BY 'InternPass123!';
Query OK, 0 rows affected (0,01 sec)

mysql>
```

Ahora para asignar los permisos a estos usuarios con GRANT PRIVILEGES, se usarían los siguientes comandos:

Usuario admin_db (Acceso total): **GRANT ALL PRIVILEGES ON codearts_db.* TO 'admin_db'@'localhost';**

Usuario dev_user (lectura y escritura): **GRANT SELECT, INSERT, UPDATE, DELETE ON codearts_db.* TO 'dev_user'@'localhost';**

Usuario intern_user (sólo lectura): **GRANT SELECT ON codearts_db.* TO 'intern_user'@'localhost';**

Lo siguiente es aplicar los cambios realizados con el comando:

FLUSH PRIVILEGES;

Y verificamos que los permisos se han concedido correctamente con:

SHOW GRANTS FOR 'dev_user'@'localhost';

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,00 sec)

mysql> SHOW GRANTS FOR 'dev_user'@'localhost';
+-----+
| Grants for dev_user@localhost |
+-----+
| GRANT USAGE ON *.* TO `dev_user`@`localhost` |
| GRANT SELECT, INSERT, UPDATE, DELETE ON `codearts_db`.* TO `dev_user`@`localhost` |
+-----+
2 rows in set (0,00 sec)

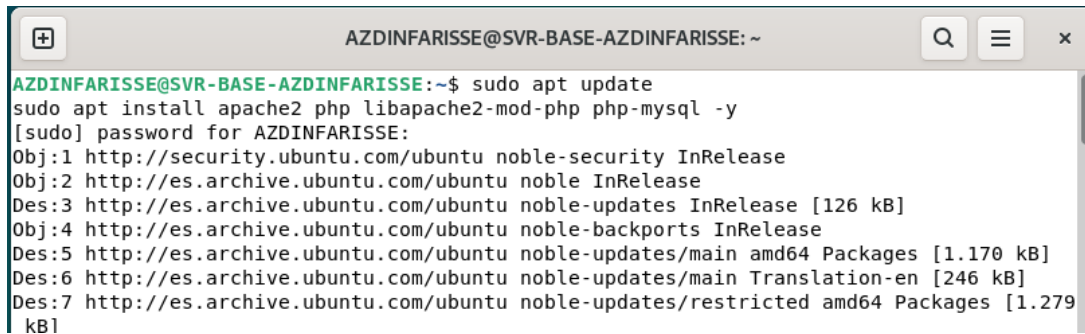
mysql> █
```

3. Conexión de una aplicación web a la base de datos:

Ahora vamos a instalar Apache y PHP con soporte para MySQL con el comando: **ç**

sudo apt update

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



```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: ~
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo apt update
sudo apt install apache2 php libapache2-mod-php php-mysql -y
[sudo] password for AZDINFARISSE:
Obj:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu noble InRelease
Des:3 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Obj:4 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease
Des:5 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1.170 kB]
Des:6 http://es.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [246 kB]
Des:7 http://es.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [1.279
kB]
```

Y reiniciamos Apache para que detecte PHP con:

sudo systemctl restart apache2

Ahora vamos a crear un script que nos permita leer y escribir datos en la base de datos, creando un archivo llamado **db_test.php** en:

sudo nano /var/www/html/db_test.php

Una vez creado este archivo, le añadiremos el siguiente código dentro:

```

<?php
$host = 'localhost';
$db = 'codearts_db';
$user = 'dev_user'; // puedes usar 'admin_db' si prefieres pruebas completas
$pass = 'DevPass123!';
$charset = 'utf8mb4';

$dsn = "mysql:host=$host;dbname=$db;charset=$charset";

try {
    $pdo = new PDO($dsn, $user, $pass);
    echo "✅ Conexión exitosa a la base de datos.<br>";

    // Crear tabla de prueba si no existe
    $pdo->exec("CREATE TABLE IF NOT EXISTS personas (
        id INT AUTO_INCREMENT PRIMARY KEY,
        nombre VARCHAR(50),
        edad INT
    )");

    echo "✅ Tabla creada o ya existente.<br>";

    // INSERTAR datos
    $pdo->exec("INSERT INTO personas (nombre, edad) VALUES ('Ana', 30)");

    // CONSULTAR
    $stmt = $pdo->query("SELECT * FROM personas");
    while ($row = $stmt->fetch()) {
        echo "👤 " . $row['nombre'] . " - " . $row['edad'] . " años<br>";
    }

    // ACTUALIZAR
    $pdo->exec("UPDATE personas SET edad = 31 WHERE nombre = 'Ana'");

    // ELIMINAR
    $pdo->exec("DELETE FROM personas WHERE nombre = 'Ana'");

    echo "✅ Pruebas de SELECT, INSERT, UPDATE y DELETE completadas.";

} catch (PDOException $e) {
    echo "❌ Error de conexión: " . $e->getMessage();
}
?>

```

Ahora comprobamos entrando en nuestro navegador a:
http://localhost/db_test.php

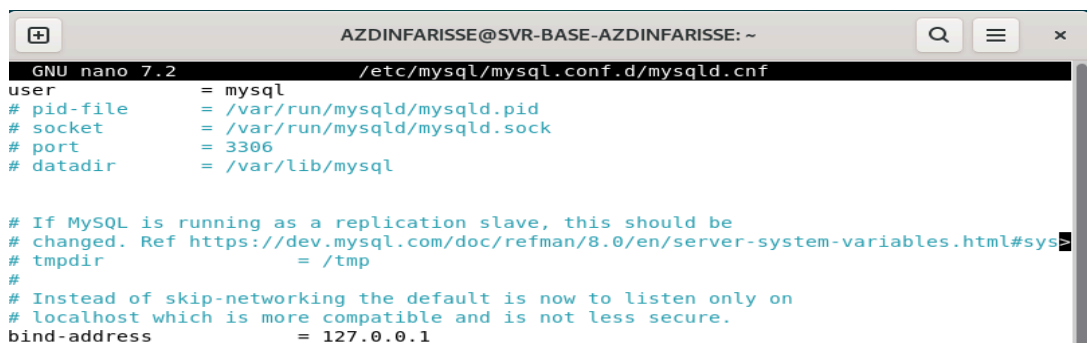


4. Seguridad y respaldo de datos.

Para configurar accesos seguros restringiendo conexiones remotas, lo haremos editando el archivo `mysqld.cnf` con el comando:

`sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf`

y buscaremos la línea **bind-address** y la editaremos para que tenga la ip **127.0.0.1** la cual solo permite conexiones locales.



y reiniciamos el servicio con:

`sudo systemctl restart mysql`

Ahora vamos a ejecutar el comando **`sudo mysql_secure_installation`** y respondemos a la pregunta de la validación de contraseñas que sí, y decimos que queremos la opción de seguridad media.

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: ~
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW      Length >= 8
MEDIUM  Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary
        file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 1

Skipping password set for root as authentication with auth_socket is used by default.
If you would like to use password authentication instead, this can be done with the "ALT
```

y a todas las demás preguntas respondemos que sí.

Ahora para crear un sistema de copias de seguridad automáticas lo haremos creando el script **backup_mysql.sh** en:

```
sudo nano /usr/local/bin/backup_mysql.sh
```

Y le añadimos lo siguiente:

```
#!/bin/bash
FECHA=$(date +%F-%H-%M)
DESTINO="/srv/backups"
mkdir -p "$DESTINO"
mysqldump -u root -pEstudios1010 --all-databases >
"$DESTINO"/backup-$FECHA.sql
```

Ahora lo haremos ejecutable con el comando:

```
sudo chmod +x /usr/local/bin/backup_mysql.sh
```

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: ~
GNU nano 7.2 /tmp/crontab.ZwqWSq/crontab
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 3 * * * /usr/local/bin/backup_mysql.sh
[ Wrote 24 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Ahora vamos a ver los logs de acceso y errores con el comando:

sudo tail -f /var/log/mysql/error.log

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo tail -f /var/log/mysql/error.log
[sudo] password for AZDINFARISSE:
2025-06-30T10:37:33.146101Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for
connections. Version: '8.0.42-0ubuntu0.24.04.1' socket: '/var/run/mysqld/mysqld.sock'
port: 3306 (Ubuntu).
2025-06-30T12:06:26.855824Z 0 [System] [MY-013172] [Server] Received SHUTDOWN from user
<via user signal>. Shutting down mysqld (Version: 8.0.42-0ubuntu0.24.04.1).
2025-06-30T12:06:27.952363Z 0 [System] [MY-010910] [Server] /usr/sbin/mysqld: Shutdown c
omplete (mysqld 8.0.42-0ubuntu0.24.04.1) (Ubuntu).
2025-06-30T12:06:28.388273Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0
.42-0ubuntu0.24.04.1) starting as process 13718
2025-06-30T12:06:28.395157Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has st
arted.
2025-06-30T12:06:28.553524Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has en
ded.
2025-06-30T12:06:28.665730Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is se
lf signed.
2025-06-30T12:06:28.665751Z 0 [System] [MY-013602] [Server] Channel mysql_main configure
d to support TLS. Encrypted connections are now supported for this channel.
2025-06-30T12:06:28.679812Z 0 [System] [MY-011323] [Server] X Plugin ready for connectio
ns. Bind-address: '127.0.0.1' port: 33060, socket: /var/run/mysqld/mysqldx.sock
2025-06-30T12:06:28.679867Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for
connections. Version: '8.0.42-0ubuntu0.24.04.1' socket: '/var/run/mysqld/mysqld.sock'
port: 3306 (Ubuntu).
```

Y activamos los logs de consultas generales editando el archivo:

sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf

Y añadiendo al final del archivo lo siguiente:

general_log = 1

general_log_file = /var/log/mysql/mysql.log

Ahora reiniciamos con:

sudo systemctl restart mysql

Y vemos el log con:

sudo less /var/log/mysql/mysql.log

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo less /var/log/mysql/mysql.log
/usr/sbin/mysqld, Version: 8.0.42-0ubuntu0.24.04.1 ((Ubuntu)). started with:
Tcp port: 3306 Unix socket: /var/run/mysqld/mysqld.sock
Time          Id Command      Argument
2025-06-30T12:54:09.298860Z      0 Execute    CREATE TABLE performance_schema.innodb_r
edo_log_files(
  `FILE_ID` BIGINT NOT NULL COMMENT 'Id of the file.',
  `FILE_NAME` VARCHAR(2000) NOT NULL COMMENT 'Path to the file.',
  `START_LSN` BIGINT NOT NULL COMMENT 'LSN of the first block in the file.',
  `END_LSN` BIGINT NOT NULL COMMENT 'LSN after the last block in the file.',
  `SIZE_IN_BYTES` BIGINT NOT NULL COMMENT 'Size of the file (in bytes).',
  `IS_FULL` TINYINT NOT NULL COMMENT '1 iff file has no free space inside.',
  `CONSUMER_LEVEL` INT NOT NULL COMMENT 'All redo log consumers registered on smaller leve
ls than this value, have already consumed this file.'
)engine = 'performance_schema'
```