Nombre de la práctica	Práctica1: Creación de usuarios			No.	1
Asignatura:	Taller de Base de datos	Carrera:	Ingeniería en Sistemas Computacionales	Duración de la práctica (Hrs)	

NOMBRE DEL ALUMNO: Vanesa Hernández Martínez
GRUPO: 3501

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):
Actividades en aula de clases y en equipo personal

III. Material empleado:

- Laptop
- Navicat
- Docker

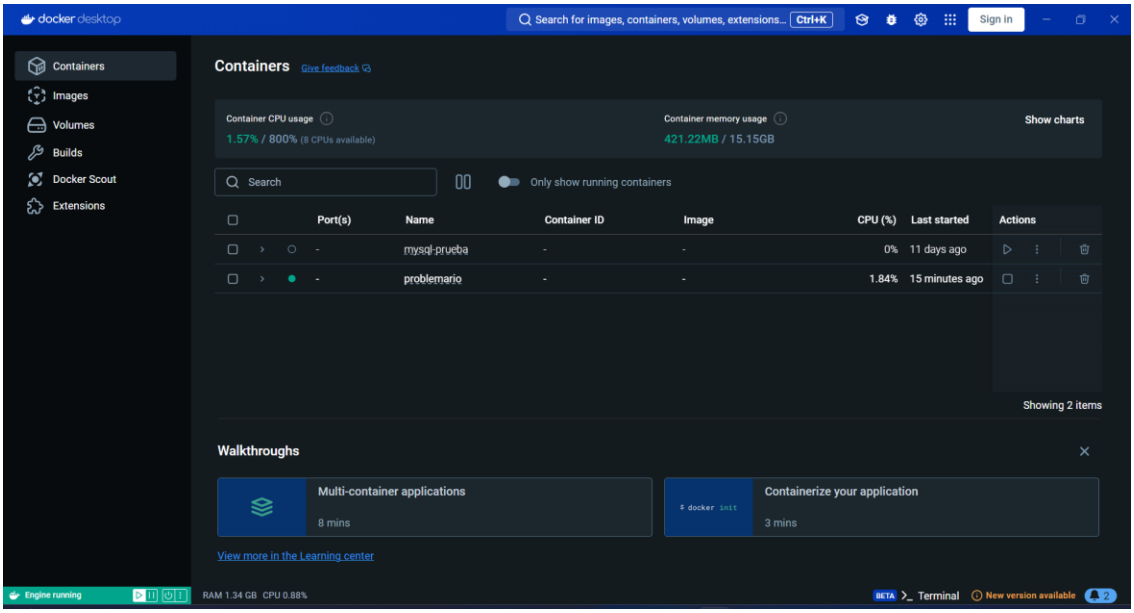
1. Para generar los problemas del tema creamos un nuevo contenedor, para ello creamos un archivo Docker-compose.yaml en donde configuramos todos los datos de la conexión:

docker-compose.yaml

```yaml
services:
  database:
    image: mysql:8.0
    ports:
      - 3308:3306
    volumes:
      - ./mysql:/var/lib/mysql
    environment:
      MYSQL_ROOT_PASSWORD: C413b  # Cambia esta contraseña por una segura
      MYSQL_DATABASE: problemario    # Nombre de la base de datos inicial
      MYSQL_USER: root        # Usuario inicial de MySQL
      MYSQL_PASSWORD: C413b


  flask:
    build: ./app
    ports:
      - 5000:5000
    depends_on:
      - database
```

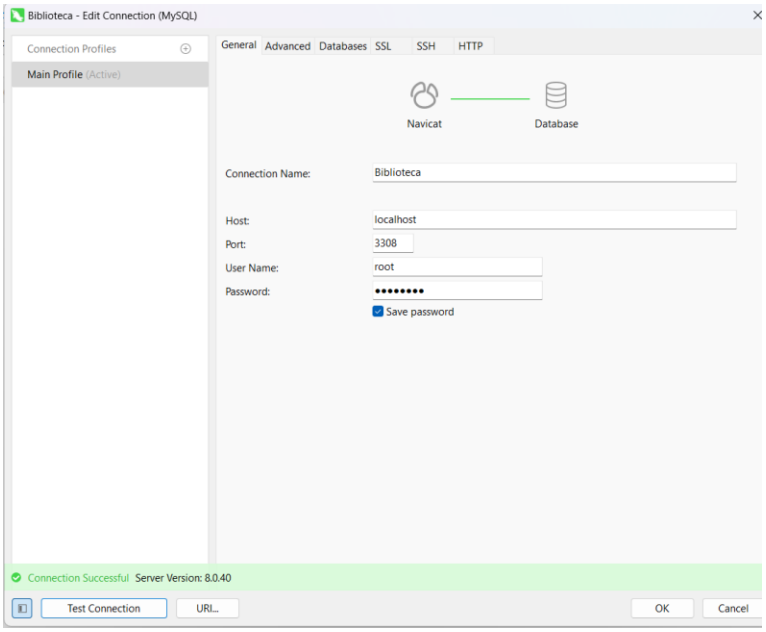
2. Abrimos Docker para revisar qure el servicio este corriendo.



3. Dentro de la terminal de Visual colocamos la siguiente instrucción para descargar la imagen y hacer posible la conexión.

docker compose up --build -d

4. Una vez se haya inicializado todo de manera correcta creamos una nueva conexión en Navicat con los datos que especificamos en Docker-compose.yaml, si al hacer testing esta nos muestra una línea de color verde esto indica que la conexión se hizo de manera correcta.



Ejercicio 1: Crear un usuario básico

Crea un usuario llamado biblioteca_usuario con la contraseña password123 . Este usuario debe tener acceso limitado para conectarse solo desde localhost .

- **Instrucción:**

  CREATE USER 'biblioteca_usuario'@'localhost' IDENTIFIED BY 'password123';

- **Funcionamiento:**

Dentro de la misma terminal de visual use el siguiente comando para abrir una terminal dentro del contenedor:

docker exec -it problemario-database-1 bash

Dentro del contenedor, utilice el cliente de MySQL para conectarme

mysql -u biblioteca_usuario -p

Si el usuario es correcto me pedirá la contraseña y me dará acceso a mysql

```
PS C:\Users\vanes\OneDrive\Escritorio\Problemario> docker exec -it problemario-database-1 bash
bash-5.1# mysql -u biblioteca_usuario -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.40 MySQL Community Server - GPL

Copyright (c) 2000, 2024, 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>
```

Ejercicio 2: Crear un usuario para acceso remoto

Crea un usuario llamado usuario_remoto con la contraseña remote123 que pueda conectarse desde cualquier dirección IP (%).

- **Instrucción:**

CREATE USER 'usuario_remoto'@'%' IDENTIFIED BY 'remote123';

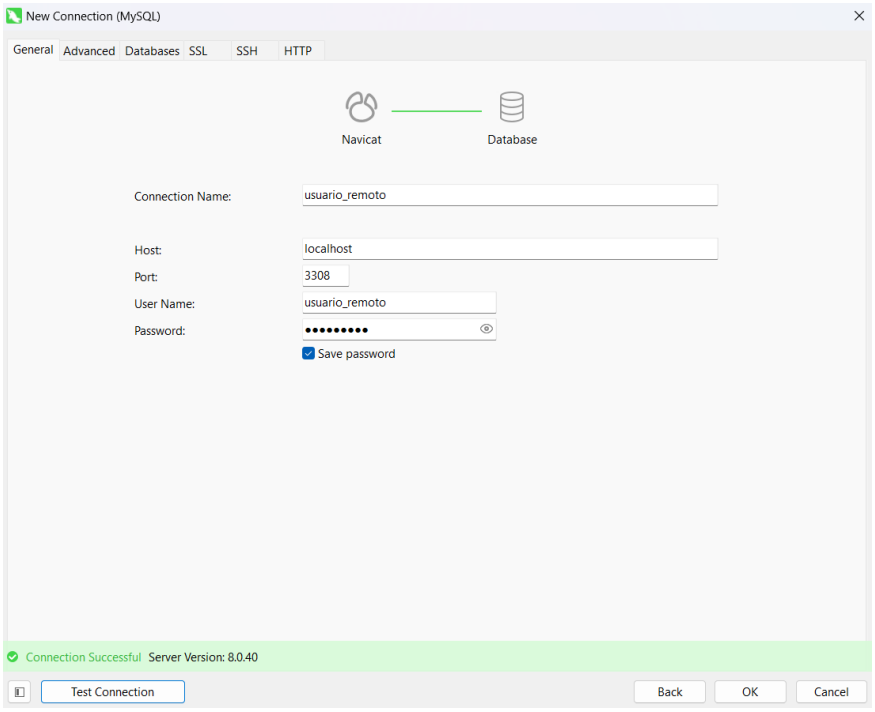
- ✓ 'usuario_remoto': Es el nombre del usuario que estamos creando.
- ✓ '%': Permite que el usuario se conecte desde cualquier dirección IP.
- ✓ 'remote123': Es la contraseña del usuario.

- **Funcionamiento:**

Desde la máquina remota o desde Navicat, configura una nueva conexión con los siguientes datos:

- Host: La dirección IP o el nombre del servidor donde está ejecutándose MySQL.
- Usuario: usuario_remoto.
- Contraseña: remote123.
- Puerto: 3308.

Prueba la conexión haciendo clic en Test Connection y si esta es de color verde ya se creó la conexión con el usuario remoto.



Ejercicio 3: Usuario con restricciones de contraseña

Crea un usuario llamado usuario_seguro con la contraseña seguro123 que expire en 90 días y permita máximo 5 intentos fallidos de inicio de sesión.

- ## Instrucción:

Ejecuta el siguiente comando para crear al usuario sin las restricciones avanzadas inicialmente:

CREATE USER 'usuario_seguro'@'%' IDENTIFIED BY 'seguro123';

Después se configurará la duración de la contraseña con:

ALTER USER 'usuario_seguro'@'%' PASSWORD EXPIRE INTERVAL 90 DAY;

Y por último este código para limitar el intento de accesos fallidos:

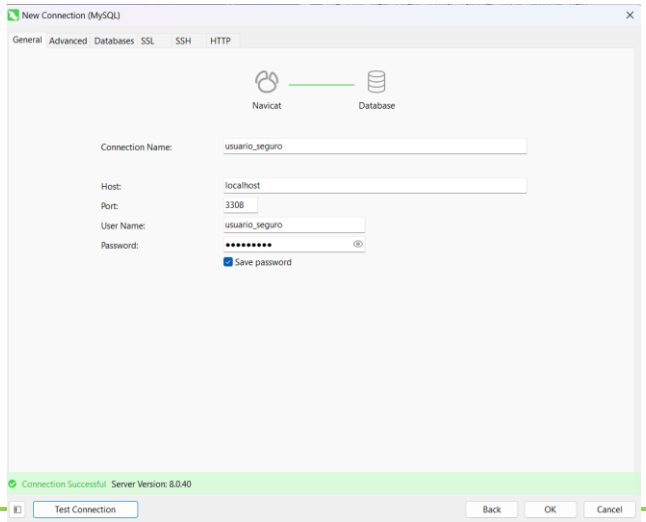
SET GLOBAL max_connections = 5;

- ## Funcionamiento:

Desde la máquina remota o desde Navicat, configura una nueva conexión con los siguientes datos:

- Host: La dirección IP o el nombre del servidor donde está ejecutándose MySQL.
- Usuario: usuario_seguro.
- Contraseña: seguro123.
- Puerto: 3308.

Prueba la conexión haciendo clic en Test Connection y si esta es de color verde ya se creó la conexión con el usuario seguro.



Ejercicio 4: Crear varios usuarios a la vez

Crea tres usuarios:

1. admin_biblioteca con acceso total desde localhost.

2. lector con acceso limitado desde 192.168.0.100 .

3. editor con acceso desde cualquier IP.

- ### Instrucción:

1. **CREATE USER 'admin_biblioteca'@'localhost' IDENTIFIED BY 'password_admin';**

   localhost: Limita el acceso únicamente a esa dirección IP.

2. **CREATE USER 'lector'@'192.168.0.100' IDENTIFIED BY 'password_lector';**

   192.168.0.100: Limita el acceso únicamente a esa dirección IP.

3. **CREATE USER 'editor'@'%' IDENTIFIED BY 'password_editor';**

   @'%': Permite acceso desde cualquier IP.

Ejercicio 5: Verificación de usuarios creados

Consulta la lista de usuarios existentes en MySQL.

- ### Instrucción:

SELECT user, host FROM mysql.user;

- mysql.user: Es la tabla interna de MySQL donde se almacenan los usuarios y sus privilegios.
- user: Es el nombre de usuario.
- host: Es la dirección IP o el nombre del host desde donde ese usuario puede conectarse.

- ## Funcionamiento:

user	host
editor	%
root	%
usuario_remoto	%
usuario_seguro	%
vane	%
lector	192.168.0.100
admin_biblioteca	localhost
biblioteca_usuario	localhost
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
root	localhost

Conclusión

Los ejercicios realizados demostraron cómo gestionar usuarios en MySQL, desde su creación hasta la implementación de restricciones de seguridad. Aprendimos a:

1. Configurar accesos básicos y remotos según las necesidades.
2. Aplicar políticas de seguridad como expiración de contraseñas y límites de intentos fallidos.
3. Crear y gestionar usuarios en masa para diferentes roles.
4. Verificar configuraciones mediante consultas.

En general, administrar usuarios de manera eficiente refuerza la seguridad y garantiza un control adecuado del acceso a la base de datos.