

Servicio Registro

Código

cliente.py

```
import socket

def cliente_registro():
    HOST = "79.72.63.217"
    PUERTO = 5000

    try:
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as cliente:
            cliente.connect((HOST, PUERTO))

            # Introducir el nombre
            nombre = input("Introduce tu nombre: ").strip()
            mensaje = f"SOY {nombre}\n"

            # Enviar el mensaje al servidor
            cliente.sendall(mensaje.encode())

            # Recibir la respuesta
            respuesta = cliente.recv(1024).decode().strip()
            print("Servidor:", respuesta)

    except ConnectionRefusedError:
        print("No se pudo conectar al servidor. Asegúrate de que está en ejecución.")
    except Exception as e:
        print(f"Error: {e}")

if __name__ == "__main__":
    cliente_registro()
```

ServidorRegistro.java

```
package registro;

import java.io.*;
import java.net.*;
import java.sql.*;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;

public class ServidorRegistro {
    private static final int PUERTO = 5000;
    private static final String DB_URL = "jdbc:mysql://localhost:3306/registro_db";
    private static final String DB_USER = "registro";
    private static final String DB_PASSWORD = "1234";

    public static void main(String[] args) {
        try (ServerSocket servidor = new ServerSocket(PUERTO)) {
            System.out.println("Servidor de registro escuchando en el puerto " + PUERTO);

            while (true) {
                try (Socket cliente = servidor.accept();
```

```

        BufferedReader entrada = new BufferedReader(new
InputStreamReader(cliente.getInputStream()));
        PrintWriter salida = new PrintWriter(cliente.getOutputStream(), true) {

            System.out.println("Cliente conectado desde " + cliente.getInetAddress());

            String mensaje = entrada.readLine().trim().toLowerCase();
            if (mensaje.startsWith("soy ")) {
                String nombre = mensaje.substring(4).trim();
                if (usuarioExiste(nombre)) {
                    registrarUsuario(nombre, cliente.getInetAddress().getHostAddress());
                    salida.println("Registro realizado correctamente");
                } else {
                    salida.println("Nombre desconocido");
                }
            } else {
                salida.println("Protocolo incorrecto");
            }

            System.out.println("Conexión cerrada con " + cliente.getInetAddress());

        } catch (IOException e) {
            System.err.println("Error en la comunicación con el cliente: " +
e.getMessage());
        }
    } catch (IOException e) {
        System.err.println("Error al iniciar el servidor: " + e.getMessage());
    }
}

private static boolean usuarioExiste(String nombre) {
    String sql = "SELECT id FROM usuarios WHERE user = ?";
    try (Connection conn = DriverManager.getConnection(DB_URL, DB_USER, DB_PASSWORD);
        PreparedStatement pstmt = conn.prepareStatement(sql)) {

        pstmt.setString(1, nombre);
        ResultSet rs = pstmt.executeQuery();
        return rs.next();

    } catch (SQLException e) {
        System.err.println("Error al verificar usuario: " + e.getMessage());
        return false;
    }
}

private static void registrarUsuario(String nombre, String ip) {
    String sql = "INSERT INTO registro (id_user, ip, fechahora) " +
        "VALUES ((SELECT id FROM usuarios WHERE user = ?), ?, ?)";
    try (Connection conn = DriverManager.getConnection(DB_URL, DB_USER, DB_PASSWORD);
        PreparedStatement pstmt = conn.prepareStatement(sql)) {

        pstmt.setString(1, nombre);
        pstmt.setString(2, ip);
        pstmt.setString(3, LocalDateTime.now().format(DateTimeFormatter.ofPattern("yyyy-MM-dd
HH:mm:ss"))));

        pstmt.executeUpdate();
        System.out.println("Registro guardado: " + nombre + " desde " + ip);

    } catch (SQLException e) {
        System.err.println("Error al registrar usuario: " + e.getMessage());
    }
}
}

```

Yo lo voy a instalar directamente en Oracle.
Creamos la base de datos con los siguientes comandos:

```
CREATE DATABASE registro_db;  
USE registro_db;  
CREATE TABLE usuarios ( id INT AUTO_INCREMENT PRIMARY KEY, user  
VARCHAR(100) UNIQUE NOT NULL );  
CREATE TABLE registro ( id INT AUTO_INCREMENT PRIMARY KEY, id_user  
INT, ip VARCHAR(45) NOT NULL, fechahora DATETIME NOT NULL, FOREIGN  
KEY (id_user) REFERENCES usuarios(id) );
```

Creo algunos usuarios de inicio dentro de la base de datos:

```
INSERT INTO usuarios (user) VALUES ('oscar'), ('adolfo'),  
('marcos');
```

Y creamos un usuario para realizar la conexión a la base de datos.

```
CREATE USER 'registro'@'localhost' IDENTIFIED BY '1234';  
  
GRANT ALL PRIVILEGES ON registro_db.* TO 'registro'@'localhost';  
  
FLUSH PRIVILEGES;
```

Antes de continuar, necesitaremos instalar el conector de mysql para java, lo haremos con los siguientes comandos:

```
wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-j\_9.2.0-1ubuntu24.10\_all.deb  
sudo dpkg -i mysql-connector-j_9.2.0-1ubuntu24.10_all.deb  
sudo apt-get -f install
```

Una vez creada la base de datos e instalado el conector, compilamos el código en el servidor para obtener el .class. Pero para ejecutarlo, primero tendremos que parar el servicio de saludo, ya que estaremos usando el mismo puerto 5000:

```

ubuntu@w7:~/PSP/T3/PSP303_ServicioRegistro$ javac registro/ServidorRegistro.java
ubuntu@w7:~/PSP/T3/PSP303_ServicioRegistro$ ls registro/
ServidorRegistro.class  ServidorRegistro.java
ubuntu@w7:~/PSP/T3/PSP303_ServicioRegistro$ java registro.ServidorRegistro
Error al iniciar el servidor: Address already in use
ubuntu@w7:~/PSP/T3/PSP303_ServicioRegistro$ sudo systemctl stop saludo
ubuntu@w7:~/PSP/T3/PSP303_ServicioRegistro$ java registro.ServidorRegistro
Servidor de registro escuchando en el puerto 5000

```

Al ejecutarlo con el comando *java registro.ServidorRegistro*, nos denegará el registro al poner el nombre ya que no estará haciendo uso del conector y no se podrá conectar a la base de datos, así que podremos hacer uso del conector con el siguiente comando:

```

java -cp "/usr/share/java/mysql-connector-java-9.2.0.jar:."
registro.ServidorRegistro

```

De esta forma, observamos que se pueden ya hacer registros correctamente:

```

PS C:\Users\oscar\Documents\GitHub\DAM2\ProgramacionServiciosProcesos\T3\PSP303_ServicioRegistro> python3 .\registro.py
¿Quién eres? > soy oscar
Servidor: Registro realizado correctamente
PS C:\Users\oscar\Documents\GitHub\DAM2\ProgramacionServiciosProcesos\T3\PSP303_ServicioRegistro>

ubuntu@w7:~/PSP/T3/PSP303_ServicioRegistro$ java -cp "/usr/share/java/mysql-connector-java-9.2.0.jar:." registro.ServidorRegistro
Servidor de registro escuchando en el puerto 5000
Cliente conectado desde /79.117.146.253
Registro guardado: oscar desde 79.117.146.253
Conexión cerrada con /79.117.146.253

```

Creación del servicio

Para crear el servicio, hacemos:

```

sudo vim /etc/systemd/system/registro.service

```

Y escribimos lo siguiente:

```
[Unit]
Description=Servidor de registro en Java
After=network.target

[Service]
User=ubuntu
WorkingDirectory=/home/ubuntu/PSP/T3/PSP303_ServicioRegistro/
ExecStart=/usr/bin/java -cp /usr/share/java/mysql-connector-java-9.2.0.jar:/home/ubuntu/PSP/T3/PSP303_ServicioRegistro
registro.ServidorRegistro
SuccessExitStatus=143
Restart=always
RestartSec=5
StandardOutput=journal
StandardError=journal

[Install]
WantedBy=multi-user.target
```

Ahora, con los siguientes comandos, activamos e iniciamos el servicio:

```
sudo systemctl daemon-reload
sudo systemctl enable registro
sudo systemctl start registro
```

Y como podemos observar, nuestro servicio ya está funcionando:

```
ubuntu@w7:~/PSP/T3/PSP303_ServicioRegistro$ sudo systemctl status registro
● registro.service - Servidor de registro en Java
   Loaded: loaded (/etc/systemd/system/registro.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-01-27 13:39:19 UTC; 5s ago
     Main PID: 257620 (java)
        Tasks: 19 (limit: 4623)
      Memory: 18.0M (peak: 18.2M)
         CPU: 65ms
    CGroup: /system.slice/registro.service
            └─257620 /usr/bin/java -cp /usr/share/java/mysql-connector-java-9.2.0.jar:/home/ubuntu/PSP/T3/PSP303_Serv

Jan 27 13:39:19 w7 systemd[1]: registro.service: Scheduled restart job, restart counter is at 74.
Jan 27 13:39:19 w7 systemd[1]: Started registro.service - Servidor de registro en Java.
Jan 27 13:39:19 w7 java[257620]: Servidor de registro escuchando en el puerto 5000

ubuntu@w7:~/PSP/T3/PSP303_ServicioRegistro$ |
```

Resultado

```
PS C:\Users\oscar\Documents\GitHub\DAM2\ProgramacionServiciosProcesos\T3\PSP303_ServicioRegistro> python3 .\registro.py
● ¿Quién eres? > soy oscar
  Servidor: Registro realizado correctamente
PS C:\Users\oscar\Documents\GitHub\DAM2\ProgramacionServiciosProcesos\T3\PSP303_ServicioRegistro> python3 .\registro.py
● ¿Quién eres? > soy adolfo
  Servidor: Registro realizado correctamente
PS C:\Users\oscar\Documents\GitHub\DAM2\ProgramacionServiciosProcesos\T3\PSP303_ServicioRegistro> |
```

Y podemos comprobar en el servicio que el saludo se ha recibido adecuadamente.

```
ubuntu@w7:~/PSP/T3/PSP303_ServicioRegistro$ sudo systemctl status registro
● registro.service - Servidor de registro en Java
   Loaded: loaded (/etc/systemd/system/registro.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-01-27 13:39:19 UTC; 1min 42s ago
     Main PID: 257620 (java)
       Tasks: 21 (limit: 4623)
      Memory: 39.6M (peak: 40.5M)
         CPU: 814ms
    CGroup: /system.slice/registro.service
            └─257620 /usr/bin/java -cp /usr/share/java/mysql-connector-java-9.2.0.jar:/home/ubuntu/PSP/T3/PSP303_ServicioRegistro

Jan 27 13:39:19 w7 systemd[1]: registro.service: Scheduled restart job, restart counter is at 74.
Jan 27 13:39:19 w7 systemd[1]: Started registro.service - Servidor de registro en Java.
Jan 27 13:39:19 w7 java[257620]: Servidor de registro escuchando en el puerto 5000
Jan 27 13:39:43 w7 java[257620]: Cliente conectado desde /79.117.146.253
Jan 27 13:39:46 w7 java[257620]: Registro guardado: oscar desde 79.117.146.253
Jan 27 13:39:46 w7 java[257620]: Conexión cerrada con /79.117.146.253
Jan 27 13:40:36 w7 java[257620]: Cliente conectado desde /79.117.146.253
Jan 27 13:40:38 w7 java[257620]: Registro guardado: adolfo desde 79.117.146.253
Jan 27 13:40:38 w7 java[257620]: Conexión cerrada con /79.117.146.253
lines 1-19/19 (END)
```