Ejercicio final curso java:

Crear un programa ue corra en un servidor que tenga una lista de personal, el cliente debe ser capaz de leer las personas, enviar una persona nueva y modificar una persona.

Al terminar el proceso del saldo del servidor la lista se debe guardar en un archivo como objetos.

```java
package JavaEjercicioFinal;
import java.net.Socket;
import java.util.ArrayList;
import java.util.List;
import java.net.ServerSocket;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;

import javafx.util.Pair;

public class ServerPersonas {
    protected static List <Persona> personasLista= new ArrayList<Persona>();
    public static void main(String[] args) {


        try (ServerSocket serverSocket = new ServerSocket(8090)) {
            System.out.println("Servidor esperando conexiones en el puerto 8090");

            cargarLista();

            while (true) {
                Socket clientSocket = serverSocket.accept();
                System.out.println("Cliente conectado desde " + clientSocket.getInetAddress());

                ObjectInputStream objectInputStream = new ObjectInputStream(clientSocket.getInputStream());
                DataInputStream dis=new DataInputStream(clientSocket.getInputStream());
                DataOutputStream dos=new DataOutputStream(clientSocket.getOutputStream());

                try {
                    String mensaje=dis.readUTF();
                    Persona personaNueva = (Persona) objectInputStream.readObject();

                    //GUARDAR
                    if(mensaje.toLowerCase().equals("guardar")) {
                        if(!verificarRepetido(personaNueva).getKey()) {
                        personasLista.add(personaNueva);
                        System.out.println(personaNueva.toString()+" guardado con exito");
                        dos.writeUTF(personaNueva.toString()+" guardado con exito");
                        }else {System.out.println("id repetido, no se pudo guardar");
                        dos.writeUTF("id repetido, no se pudo guardar");
                        }

                    //MOSTRAR TODO
                    }else if(mensaje.toLowerCase().equals("mostrar")) {
                        dos.writeUTF("lista recibida del servidor: \n"+mostrarLista());
```

```
                    //MODIFICAR
                    }else if(mensaje.toLowerCase().equals("modificar")) {
                        if(verificarRepetido(personaNueva).getKey()) {
                            int indiceAModificar = verificarRepetido(personaNueva).getValue();
                            personasLista.set(indiceAModificar, personaNueva);
                            System.out.println("Persona con id: "+ personaNueva.getId()+" modificada con exito");
                            dos.writeUTF("Persona con id: "+ personaNueva.getId()+" modificada con exito");
                        }else {
                            dos.writeUTF("Persona con id: "+ personaNueva.getId()+" No encontrada.");
                        }

                    //BORRAR
                    }else if(mensaje.toLowerCase().equals("borrar")) {
                        if(verificarRepetido(personaNueva).getKey()) {
                            int indiceAModificar = verificarRepetido(personaNueva).getValue();
                            personasLista.remove(indiceAModificar);
                            System.out.println("Persona con id: "+ personaNueva.getId()+" borrado con exito");
                            dos.writeUTF("Persona con id: "+ personaNueva.getId()+" borrado con exito");
                        }
                    }
                    guardarLista();

                } catch (ClassNotFoundException e) {
                    e.printStackTrace();
                }
                dos.close();
                objectInputStream.close();
                clientSocket.close();

            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
```

```
public static Pair<Boolean, Integer> verificarRepetido(Persona personaAVerificar){
    boolean banderaIdRepetido = false;

    if (personasLista.isEmpty()) {
        return new Pair<>(banderaIdRepetido, 0);
    }

    for (int i = 0; i < personasLista.size(); i++) {
        if (personasLista.get(i).getId() == personaAVerificar.getId()) {
            banderaIdRepetido = true;
            return new Pair<>(banderaIdRepetido, i);
        }
    }

    return new Pair<>(banderaIdRepetido, 0);//retorna un pair donde key es booleano para saber si existe o no la fila
    //y value para saber el indice en el que esta
}
```

```
public static void guardarLista() {
    try {
        ObjectOutputStream objectOutputStream = new ObjectOutputStream(new FileOutputStream("personas.dat"));
        objectOutputStream.writeObject(personasLista);
        objectOutputStream.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

```java
    public static void cargarLista() {
        try {
            ObjectInputStream objectInputStream = new ObjectInputStream(new FileInputStream("personas.dat"));
            personasLista = (List<Persona>) objectInputStream.readObject();
            objectInputStream.close();
        } catch (IOException | ClassNotFoundException e) {
            System.out.println("No se pudo cargar la lista de personas. Se creará una lista vacía.");
            personasLista = new ArrayList<>();
        }
    }
}
```

```java
public static String  mostrarLista(){
    if(!personasLista.isEmpty()) {
         StringBuilder stringBuilder = new StringBuilder();
        System.out.println("\n ***** Lista De Personas *****");
        for (int i = 0; i < personasLista.size(); i ++) {
            System.out.println(personasLista.get(i).toString());
            stringBuilder.append(personasLista.get(i).toString()).append("\n");
        }
        return stringBuilder.toString();
        }else {
            System.out.println("No hay personas para imprimir");
            return "No hay personas para imprimir";
        }


}
```

Cliente:

```java
package JavaEjercicioFinal;
import java.io.*;

public class ClientePersonas {

    public static void main(String[] args) {
        try (Socket clientSocket = new Socket("localhost", 8090)) {
            Persona persona = new Persona(7, "minerva", "123456789");
            interactuarPersona( persona, clientSocket,"guardar");

        } catch (IOException e) {
            e.printStackTrace();

        }
    }
    public static void interactuarPersona(Persona persona, Socket socketDelServer,String accion) throws IOException{
            ObjectOutputStream objectOutputStream = new ObjectOutputStream(socketDelServer.getOutputStream());
            DataOutputStream dos = new DataOutputStream(socketDelServer.getOutputStream());
            DataInputStream dis = new DataInputStream(socketDelServer.getInputStream());
            dos.writeUTF(accion);
            objectOutputStream.writeObject(persona);
            System.out.println(dis.readUTF());
            objectOutputStream.flush();
            objectOutputStream.close();
            dos.close();
            dis.close();


    }

}
```

Pruebas:

```
<terminated> ClientePersonas [Java Application] C:\Program Files\Java\jdk1.8.0_202\bin\javaw.exe  (25 oct. 2023 17:38:19 – 17:38:19) [pid: 18172]
Persona [id=7, nombre=minerva, numeroTelefono=123456789] guardado con exito
```

```
Servidor esperando conexiones en el puerto 8090
Cliente conectado desde /127.0.0.1
Persona [id=7, nombre=minerva, numeroTelefono=123456789] guardado con exito
```

mostrar:

```
try (Socket clientSocket = new Socket("localhost", 8090)) {
     Persona persona = new Persona(7, "minerva", "123456789");
     interactuarPersona( persona, clientSocket,"mostrar");
```

```
lista recibida del servidor:
Persona [id=2, nombre=tilin, numeroTelefono=123456789]
Persona [id=1, nombre=pepe, numeroTelefono=123456789]
Persona [id=3, nombre=pepe, numeroTelefono=123456789]
Persona [id=4, nombre=pepe, numeroTelefono=123456789]
Persona [id=5, nombre=pepe, numeroTelefono=123456789]
Persona [id=6, nombre=pepe, numeroTelefono=123456789]
Persona [id=7, nombre=minerva, numeroTelefono=123456789]
```

```
ServerPersonas [Java Application] C:\Program Files\Java\jdk1.8.0_202\bin\javaw.exe  (25 oct. 2023 17:37:33) [pid: 4324]
Servidor esperando conexiones en el puerto 8090
Cliente conectado desde /127.0.0.1
Persona [id=7, nombre=minerva, numeroTelefono=123456789] guardado con exito
Cliente conectado desde /127.0.0.1

 ***** Lista De Personas *****
Persona [id=2, nombre=tilin, numeroTelefono=123456789]
Persona [id=1, nombre=pepe, numeroTelefono=123456789]
Persona [id=3, nombre=pepe, numeroTelefono=123456789]
Persona [id=4, nombre=pepe, numeroTelefono=123456789]
Persona [id=5, nombre=pepe, numeroTelefono=123456789]
Persona [id=6, nombre=pepe, numeroTelefono=123456789]
Persona [id=7, nombre=minerva, numeroTelefono=123456789]
```

Modificar

```
try (Socket clientSocket = new Socket("localhost", 8090)) {
     Persona persona = new Persona(7, "nezahualcoyotl", "123456789");
     interactuarPersona( persona, clientSocket,"modificar");
```

```
<terminated> ClientePersonas [Java Application] C:\Program Files\Java\jdk1.8.0_202\bin\javaw.exe  (25 oct. 2023 17:41:04 – 17:41:04) [pid: 6040]
Persona con id: 7 modificada con exito
```

```
<terminated> ClientePersonas [Java Application] C:\Program Files\Java\jdk1.8.0_202\bin\javaw.exe  (25 oct. 2023 17:41:27 – 17:41:28) [pid: 15844]
lista recibida del servidor:
Persona [id=2, nombre=tilin, numeroTelefono=123456789]
Persona [id=1, nombre=pepe, numeroTelefono=123456789]
Persona [id=3, nombre=pepe, numeroTelefono=123456789]
Persona [id=4, nombre=pepe, numeroTelefono=123456789]
Persona [id=5, nombre=pepe, numeroTelefono=123456789]
Persona [id=6, nombre=pepe, numeroTelefono=123456789]
Persona [id=7, nombre=nezahualcoyotl, numeroTelefono=123456789]
```

Borrar

```
interactuarPersona( persona, clientSocket,"borrar");
```

```
Persona con id: 7 borrado con exito
```

```
<terminated> ClientePersonas [Java Application] C:\Program Files\Java\jdk1.8.0_202\bin\javaw.exe  (25 oct. 2023 17:42:22 – 17:42:22) [pid: 17460]
lista recibida del servidor:
Persona [id=2, nombre=tilin, numeroTelefono=123456789]
Persona [id=1, nombre=pepe, numeroTelefono=123456789]
Persona [id=3, nombre=pepe, numeroTelefono=123456789]
Persona [id=4, nombre=pepe, numeroTelefono=123456789]
Persona [id=5, nombre=pepe, numeroTelefono=123456789]
Persona [id=6, nombre=pepe, numeroTelefono=123456789]
```