



INTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE COMPUTO



Nombre de los alumnos: SANDOVAL MUÑOZ EMANUEL

Matriculas: 2022670104

Maestra: BAUTISTA ROSALES SANDRA IVETTE

Materia: APLICACIONES PARA COMUNICACIONES EN RED

Grupo: 6CM4

Cliente

```
import java.io.*;
import java.net.Socket;

public class Cliente implements Serializable {
    private String nombre; // Excluded from serialization 4 usages
    private int edad;      // Excluded from serialization 4 usages
    private double salario; 4 usages
    private String email; 4 usages
    private boolean casado; 4 usages
    private float altura; // Excluded from serialization 4 usages

    public Cliente(String nombre, int edad, double salario, String email, boolean casado, float altura) { 1 usage
        this.nombre = nombre;
        this.edad = edad;
        this.salario = salario;
        this.email = email;
        this.casado = casado;
        this.altura = altura;
    }

    public String getNombre() { 1 usage
        return nombre;
    }

    public void setNombre(String nombre) { no usages
        this.nombre = nombre;
    }

    public int getEdad() { 1 usage
        return edad;
    }

    public void setEdad(int edad) { no usages
        this.edad = edad;
    }
}
```

```

public double getSalario() { 1 usage
    return salario;
}

public void setSalario(double salario) { no usages
    this.salario = salario;
}

public String getEmail() { 1 usage
    return email;
}

public void setEmail(String email) { no usages
    this.email = email;
}

public boolean isCasado() { return casado; }

public void setCasado(boolean casado) { no usages
    this.casado = casado;
}

public float getAltura() { 1 usage
    return altura;
}

public void setAltura(Float altura) { no usages
    this.altura = altura;
}

```

```

public static void main(String[] args) {
    try {
        System.out.println("Sin serializar:");
        Cliente cliente = new Cliente( nombre: "Emanuel", edad: 21, salario: 2000.2, email: "mariovsgoku12@gmail.com", casado: true, altura: 1.75f);
        System.out.println(cliente);

        // Serializing the client
        FileOutputStream fos = new FileOutputStream( name: "cliente.ser");
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        oos.writeObject(cliente);
        oos.close();
        fos.close();

        // Sending the serialized file via socket (assuming server is already listening)
        File file = new File( pathname: "cliente.ser");
        Socket socket = new Socket( host: "localhost", port: 1234);
        FileInputStream fileStream = new FileInputStream(file);
        DataOutputStream dos = new DataOutputStream(socket.getOutputStream());

        //Serialization
        FileInputStream fis = new FileInputStream( name: "cliente.ser");
        ObjectInputStream ois = new ObjectInputStream(fis);
        Cliente deserializedCliente = (Cliente) ois.readObject();
        ois.close();
        fis.close();

        // Send the file name and size before the file content
        dos.writeUTF(file.getName());
        dos.writeLong(file.length());

        byte[] buffer = new byte[4096];
        int bytesRead;
        while ((bytesRead = fileStream.read(buffer)) != -1) {
            dos.write(buffer, off: 0, bytesRead);
        }
    }
}

```

```

        dos.flush();
        //print deserializacion
        System.out.println("Serializado:");
        System.out.println("Nombre: " + deserializedCliente.getNombre());
        System.out.println("Edad: " + deserializedCliente.getEdad());
        System.out.println("Salario: " + deserializedCliente.getSalario());
        System.out.println("Email: " + deserializedCliente.getEmail());
        System.out.println("Casado: " + deserializedCliente.isCasado());
        System.out.println("Altura: " + deserializedCliente.getAltura());
        fileStream.close();
        dos.close();
        socket.close();
        System.out.println("Archivo enviado");
    } catch (Exception e) {
        e.printStackTrace();
    }
}

@Override
public String toString() {
    return "Nombre: " + nombre +
        ", Edad: " + edad +
        ", Salario: " + salario +
        ", Email: " + email +
        ", Casado: " + casado +
        ", Altura: " + altura;
}
}

```

Servidor

```
import java.net.*;
import java.io.*;

public class Servidor {
    public static void main(String[] args) {
        try {
            ServerSocket serverSocket = new ServerSocket( port: 1234);
            System.out.println("Servidor listo, esperando conexiones...");

            while (true) {
                Socket clientSocket = serverSocket.accept();
                System.out.println("Conexion establecida desde: " + clientSocket.getInetAddress() + ":" + clientSocket.getPort());

                try (DataInputStream dis = new DataInputStream(clientSocket.getInputStream())) {
                    System.out.println("Waiting to receive file name and size...");
                    String fileName = dis.readUTF();
                    long fileSize = dis.readLong();
                    System.out.println("Recibiendo archivo: " + fileName + " de tamaño " + fileSize + " bytes");

                    // Receiving file
                    try (BufferedOutputStream bos = new BufferedOutputStream(new FileOutputStream(fileName))) {
                        byte[] buffer = new byte[1024];
                        int bytesRead;
                        long totalRead = 0;
                        while (totalRead < fileSize) {
                            bytesRead = dis.read(buffer);
                            totalRead += bytesRead;
                            bos.write(buffer, off: 0, bytesRead);
                        }
                        bos.flush();
                        System.out.println("Archivo " + fileName + " recibido.");
                    }

                    // Deserialize the object from file
                    try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(fileName))) {
                        Cliente cliente = (Cliente) ois.readObject();
                        System.out.println("Objeto Cliente deserializado: ");
                        System.out.println(cliente); // Assuming Cliente class has a proper toString() method
                    }
                }
            }
        }
    }
}
```

```
        } catch (EOFException e) {
            System.out.println("EOFException caught: Client may have sent incomplete data.");
        } finally {
            clientSocket.close();
        }
    }
} catch (Exception e) {
    e.printStackTrace();
}
}
```

Ejecución

Espera la conexión:

```
"C:\Program Files\Java\jdk-18.0.2\bin\java.exe" "-javaagent:C:\Users\emanu\AppData\Local\JetBrains\IntelliJ IDEA 2022.3.3\lib\idea_rt.jar=20430:C:\Users\emanu\AppData\Local\JetBrains\IntelliJ IDEA 2022.3.3\bin" -Dfile.encoding=UTF-8
Servidor listo, esperando conexiones...
```

Envío de datos por el cliente:

```
Sin serializar:
Nombre: Emanuel, Edad: 21, Salario: 2000.2, Email: mariovsgoku12@gmail.com, Casado: true, Altura: 1.75
Serializado:
Nombre: Emanuel
Edad: 21
Salario: 2000.2
Email: mariovsgoku12@gmail.com
Casado: true
Altura: 1.75
Archivo enviado
```

Datos recibidos del servidor

```
"C:\Program Files\Java\jdk-18.0.2\bin\java.exe" "-javaagent:C:\Users\emanu\AppData\Local\JetBrains\IntelliJ IDEA 2022.3.3\lib\idea_rt.jar=20430:C:\Users\emanu\AppData\Local\JetBrains\IntelliJ IDEA 2022.3.3\bin" -Dfile.encoding=UTF-8
Servidor listo, esperando conexiones...
Conexion establecida desde: /127.0.0.1:20439
Waiting to receive file name and size...
Recibiendo archivo: cliente.ser de tamaño 159 bytes
Archivo cliente.ser recibido.
Objeto Cliente deserializado:
Nombre: Emanuel, Edad: 21, Salario: 2000.2, Email: mariovsgoku12@gmail.com, Casado: true, Altura: 1.75
```

Preguntas:

1. ¿Qué características observas al abrir el archivo con el objeto serializado?

Al abrir un archivo con un objeto serializado, observamos que el archivo generalmente está en formato binario y no es legible directamente por humanos. Este archivo es compacto y contiene una estructura detallada que representa tanto los valores de los atributos del objeto como información sobre su tipo y estructura.

2. ¿Qué valores adoptan los atributos seleccionados como no serializables cuando se imprimen con el objeto ya serializado?

Los atributos seleccionados como no serializables generalmente no se incluyen en el archivo serializado. Cuando se intenta imprimir o acceder a estos atributos después de deserializar el objeto, usualmente adoptan valores por defecto o se establecen en None, dependiendo del comportamiento predeterminado del lenguaje de programación utilizado.

Referencias:

Portianko, V. (2024, February 9). *Serialización y deserialización en Java*. CodeGym.

<https://codegym.cc/es/groups/posts/es.112.serializacion-y-deserializacion-en-java>

Ramhead. (2024, March 29). *Cómo funciona la serialización en Java*. JavaRush.

<https://javarush.com/es/groups/posts/es.710.cmo-funciona-la-serializacin-en-java>