Forwarder:

```
import org.json.JSONObject;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.Socket;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Forwarder implements Runnable {
    private int puerto;
    private String mensaje;
    private String host;

    public Forwarder(int puerto, String host) {
        this.puerto = puerto;
        this.host = host;
    }

    public void marshal(int[] votos){
        JSONObject sampleObject = new JSONObject();
        sampleObject.put("Votos1", votos[0]);
        sampleObject.put("Votos2", votos[1]);
        sampleObject.put("Votos3", votos[2]);
        this.mensaje = sampleObject.toString();
    }

    public void setMensaje(String mensaje) {
        this.mensaje = mensaje;
    }

    @Override
    public void run() {
        DataOutputStream out;
        try {
            //Creo el socket para conectarme con el cliente
            Socket sc = new Socket(host, puerto);
            out = new DataOutputStream(sc.getOutputStream());
            //Envio un mensaje al cliente
            out.writeUTF((String)mensaje);
            sc.close();

        } catch (IOException ex) {
            Logger.getLogger(Forwarder.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }
}
```

Receiver:

```java
import org.json.JSONObject;
import java.io.DataInputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Observable;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Receiver extends Observable implements Runnable {

    private int puerto;

    public Receiver(int puerto) {
        this.puerto = puerto;
    }

    @Override
    public void run() {
        ServerSocket servidor = null;
        Socket sc = null;
        DataInputStream in;

        try {
            //Creamos el socket del servidor
            servidor = new ServerSocket(puerto);

            //Siempre estara escuchando peticiones
            while (true) {
                //Espero a que un cliente se conecte
                sc = servidor.accept();
                //System.out.println("Cliente conectado");
                in = new DataInputStream(sc.getInputStream());
                //Leo el mensaje que me envia
                String mensaje = in.readUTF();

                this.setChanged();
                this.notifyObservers(mensaje);
                this.clearChanged();

                //Cierro el socket
                sc.close();
```

```
            }

        } catch (IOException ex) {
            Logger.getLogger(Receiver.class.getName()).log(Level.SEVERE,
null, ex);
        }

    }
    public int[] unmarshal(String votos){
        JSONObject jsonVotos = new JSONObject(votos);
        Object votos1 = jsonVotos.get("Votos1");
        Object votos2 = jsonVotos.get("Votos2");
        Object votos3 = jsonVotos.get("Votos3");
        int[] arrayVotos = {(int)votos1, (int)votos2, (int)votos3};
        return arrayVotos;
    }

}
```