main.js

```javascript
const readline = require('readline');

const {log, biglog, errorlog, colorize} = require("./out");

const cmds = require("./cmds");

const net = require("net");

net.createServer(socket => {

    console.log("Se ha conectado un cliente desde " + socket.remoteAddress);

    // Mensaje inicial
    biglog(socket, 'CORE Quiz', 'green');

    const rl = readline.createInterface({
        input: socket,
        output: socket,
        prompt: colorize("quiz > ", 'blue'),
        completer: (line) => {
            const completions = 'h help add delete edit list test p play credits q quit'.split(' ');
            const hits = completions.filter((c) => c.startsWith(line));
            // show all completions if none found
            return [hits.length ? hits : completions, line];
        }
    });

    socket
    .on("end" , () => {rl.close(); })
    .on("end" , () => {rl.close(); });
    rl.prompt();

    rl
        .on('line', (line) => {

            let args = line.split(" ");
            let cmd = args[0].toLowerCase().trim();

            switch (cmd) {
                case '':
                    rl.prompt();
                    break;

                case 'help':
                case 'h':
                    cmds.helpCmd(socket,rl);
                    break;

                case 'quit':
                case 'q':
                    cmds.quitCmd(socket,rl);
                    break;

                case 'add':
                    cmds.addCmd(socket,rl);
                    break;

                case 'list':
                    cmds.listCmd(socket,rl);
                    break;

                case 'show':
                    cmds.showCmd(socket,rl, args[1]);
                    break;

                case 'test':
                    cmds.testCmd(socket, rl, args[1]);
                    break;

                case 'play':
                case 'p':
                    cmds.playCmd(socket,rl);
                    break;

                case 'delete':
                    cmds.deleteCmd(socket,rl, args[1]);
                    break;

                case 'edit':
                    cmds.editCmd(socket,rl, args[1]);
                    break;

                case 'credits':
                    cmds.creditsCmd(socket,rl);
                    break;

                default:
                    log(socket,`Comando desconocido: '${colorize(cmd, 'red')}'`);
                    log(socket,`Use ${colorize('help', 'green')} para ver todos los comandos disponibles.`);
                    rl.prompt();
                    break;
            }
        })
        .on('close', () => {
            log(socket,'Adios!');
            //process.exit(0);
        });

})

.listen(3030);
```