```cpp
#include <iostream>
#include <winsock2.h>
#include <WS2tcpip.h>

bool esOpcion(std::string buffer);

int main() {
    std::string ip /*= "192.168.1.34"*/;
    int puerto /*= 5005*/;
    std::cout << "Ingrese la dirección IP del servidor: ";
    std::cin >> ip;
    std::cout << "Ingrese el puerto del servidor: ";
    std::cin >> puerto;
    std::cin.ignore();

    WSADATA wsData;
    if (WSAStartup(MAKEWORD(2, 2), &wsData) != 0) {
        std::cerr << "Error al inicializar Winsock" << std::endl;
        return -1;
    }

    SOCKET clientSocket = socket(AF_INET, SOCK_STREAM, 0);
    if (clientSocket == INVALID_SOCKET) {
        std::cerr << "Error al crear el socket del cliente" << std::endl;
        WSACleanup();
        return -1;
    }

    sockaddr_in serverAddr;
    serverAddr.sin_family = AF_INET;
    serverAddr.sin_port = htons(puerto);
    serverAddr.sin_addr.s_addr = inet_addr(ip.c_str());

    if (connect(clientSocket, (sockaddr*)&serverAddr, sizeof(serverAddr)) == SOCKET_ERROR) {
        std::cerr << "Error al conectar al servidor" << std::endl;
        closesocket(clientSocket);
        WSACleanup();
        return -1;
    }

    std::cout << "Conectado al servidor! Ingrese Usuario|Contrasena" << std::endl;

    char buffer[1024];
    std::string rolUsuario;
    bool registro = false;
    int contador = 0;
    while (true) {
        std::cout << "Cliente: ";
        std::cin.getline(buffer, sizeof(buffer));
        send(clientSocket, buffer, strlen(buffer), 0);
        std::string opcion = buffer;

        if (contador >0 && !esOpcion(opcion) && strlen(buffer) != 1024){
            system("cls");
        }
        memset(buffer, 0, sizeof(buffer));
        int bytesReceived = recv(clientSocket, buffer, sizeof(buffer), 0);
        if (opcion == "b" && buffer[0] == 'N') {
            system("cls");
        }

        while (bytesReceived == 1024) {
//            std::cout << buffer << std::endl;
            std::cout.write(buffer, strlen(buffer) - 1);
            send(clientSocket, ".", 1, 0);
            memset(buffer, 0, sizeof(buffer));
            bytesReceived = recv(clientSocket, buffer, sizeof(buffer), 0);
            registro = true;
        }

        if (bytesReceived == SOCKET_ERROR) {
            std::cerr << "Error al recibir datos del servidor" << std::endl;
            break;
        } else if (bytesReceived == 0) {
            std::cout << "Servidor desconectado" << std::endl;
            break;
        }
        if(!registro){
            std::cout << "Servidor: " << buffer << std::endl;
        } else {
            std::cout << buffer << std::endl;
            registro = false;
        }
        contador++;
    }


    closesocket(clientSocket);
    WSACleanup();

    return 0;
}

bool esOpcion(std::string buffer){
    bool retorno = false;
    if(buffer == "1" || buffer == "2" || buffer == "3" || buffer == "4" || buffer == "a" || buffer == "b"){
        retorno = true;
    }
    return retorno;
}
```