```
TcpClient cliente = null;
NetworkStream stream = null;
string servidor = "127.0.0.1";
int porta = 13000;
string msgInicial = string.Empty;
string simboloJogador = "0";
string resposta = string.Empty;
bool jogoIniciado = false;
bool jogoEncerrado = false;
int bytesLidos = 0;
bool ignorar = false;
bool controle = false;

try
{
    cliente = new TcpClient(servidor, porta); // Conectando ao servidor local
    stream = cliente.GetStream();

    Console.WriteLine("Conectado ao servidor do Jogo da Velha!");
```

Inicializa variáveis para conexão e controle do jogo.
Conecta ao servidor usando TcpClient e obtém o NetworkStream para comunicação.
Exibe uma mensagem de confirmação de conexão.

```
    byte[] buffer = new byte[512];

    while (!jogoIniciado)
    {
        // Ler resposta inicial do servidor (esperando outro jogador ou início do jogo)
        bytesLidos = stream.Read(buffer, offset: 0, size: buffer.Length);
        msgInicial = Encoding.UTF8.GetString(buffer, index: 0, count: bytesLidos).Trim();

        if (msgInicial == "1" && !ignorar)
        {
            Console.WriteLine("Aguardando outro jogador...");
            simboloJogador = "X";
            ignorar = true;
        }
        else if (msgInicial == "0")
        {
            Console.WriteLine("Iniciando Partida...");
            jogoIniciado = true;
        }
    }
```

Cria um buffer para leitura de dados do servidor.
Enquanto o jogo não estiver iniciado, lê a resposta do servidor:
Se a resposta é "1", indica que está esperando outro jogador e ajusta o símbolo do jogador.
Se a resposta é "0", inicia o jogo.

```
// Recebe mensagem do Servidor
bytesLidos = stream.Read(buffer, offset: 0, size: buffer.Length);
resposta = Encoding.UTF8.GetString(buffer, index: 0, count: bytesLidos);

// Exibe o jogo da velha
Console.WriteLine(resposta);
ExibirJogo(resposta);

if (simboloJogador == "O")
{
    controle = true;
}
```

Recebe o estado inicial do tabuleiro do servidor e o exibe.
Se o jogador é "O", define controle como true.

```
// Loop do jogo
while (!jogoEncerrado)
{
    if (controle)
    {
        Console.WriteLine("Aguarde a jogada do oponente...");
    }

    // Recebe mensagem do Servidor
    bytesLidos = stream.Read(buffer, offset: 0, size: buffer.Length);
    resposta = Encoding.UTF8.GetString(buffer, index: 0, count: bytesLidos);

    if (resposta.Length == 1)
    {
        while (true)
        {
            // Pede para o usuário digitar a jogada
            Console.WriteLine("Sua vez! Digite sua jogada (1-9):");

            // Envia a jogada para o Servidor
            string jogada = Console.ReadLine();
            byte[] msg = Encoding.UTF8.GetBytes(jogada);
            stream.Write(msg, offset: 0, size: msg.Length);

            // Receber a confirmação da jogada
            bytesLidos = stream.Read(buffer, offset: 0, size: buffer.Length);
            resposta = Encoding.UTF8.GetString(buffer, index: 0, count: bytesLidos).Trim();
```

```
            if (resposta == "-1")
            {
                Console.WriteLine("Jogada inválida, tente novamente.");
            }
            else
            {
                Console.WriteLine("Jogada registrada com sucesso!");
                // Recebe jogo da velha atualizado do Servidor
                bytesLidos = stream.Read(buffer, offset: 0, size: buffer.Length);
                resposta = Encoding.UTF8.GetString(buffer, index: 0, count: bytesLidos);
                break;
            }
        }
    }

    // Exibe o jogo da velha
    ExibirJogo(resposta);

    if (resposta.Length == 10)
    {
        // Verificar se houve vitória ou empate
        char resultadoJogo = resposta[9];
        if (resultadoJogo == '1' || resultadoJogo == '2')
        {
            // Exibe mensagem de vitória/derrota para ambos os jogadores
            if ((resultadoJogo == '1' && simboloJogador == "X") || (resultadoJogo == '2' && simboloJogador == "O"))
            {
                Console.WriteLine($"Parabens Pela Vitoria: {simboloJogador}");
            }
            else
            {
                // Mensagem de derrota
                Console.WriteLine("o oponente venceu te derrotou");
            }
            jogoEncerrado = true;
        }
```

```
                Console.WriteLine("Deu Velha! Empate");
                jogoEncerrado = true;
            }
        }

        // Al
        controle = !controle;
    }
```

Enquanto o jogo não estiver encerrado:

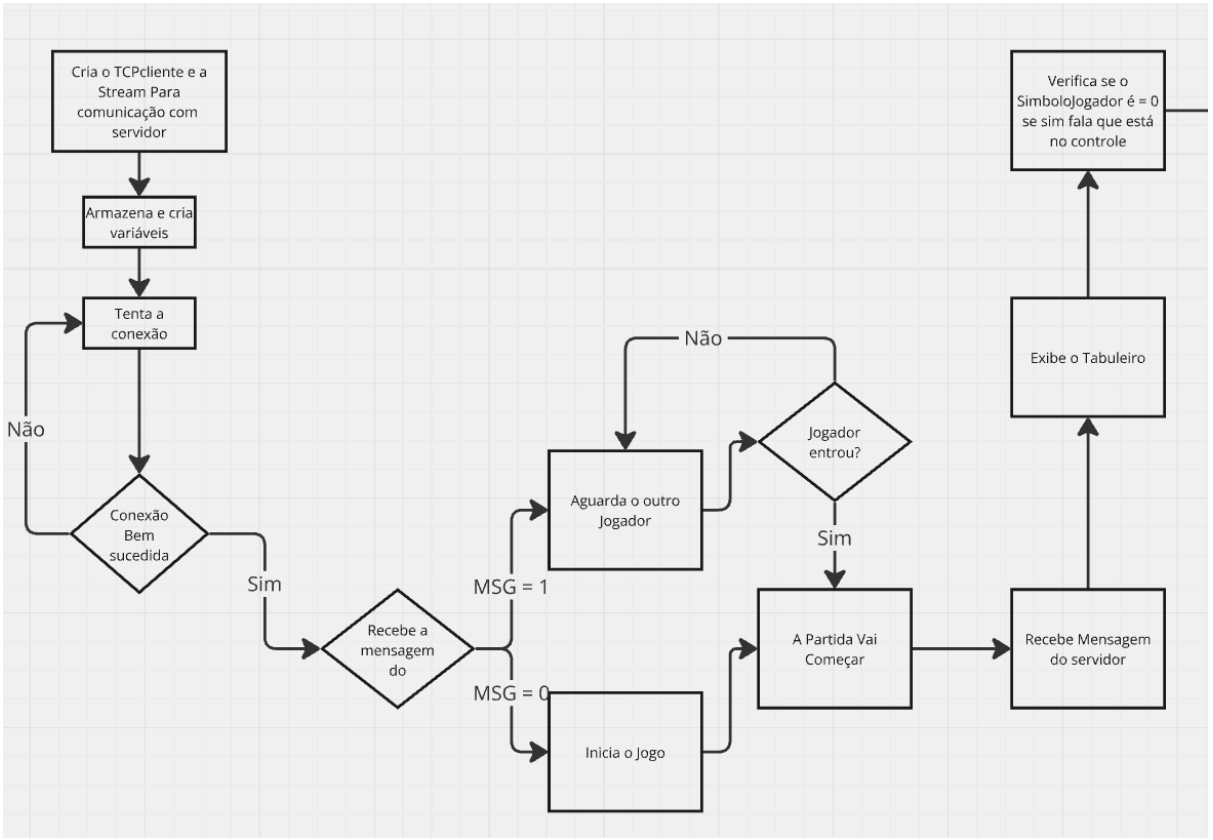
- Se o jogador tem o controle, aguarda a jogada do oponente.
- Recebe a mensagem do servidor e verifica se é uma solicitação de jogada.
- Solicita a jogada do usuário, envia para o servidor e confirma a jogada.
- Recebe o estado atualizado do jogo e exibe.
- Se o estado da resposta indica fim de jogo (resposta.Length == 10), verifica o resultado e exibe a mensagem apropriada.
- Alterna o controle do jogo para o outro jogador.

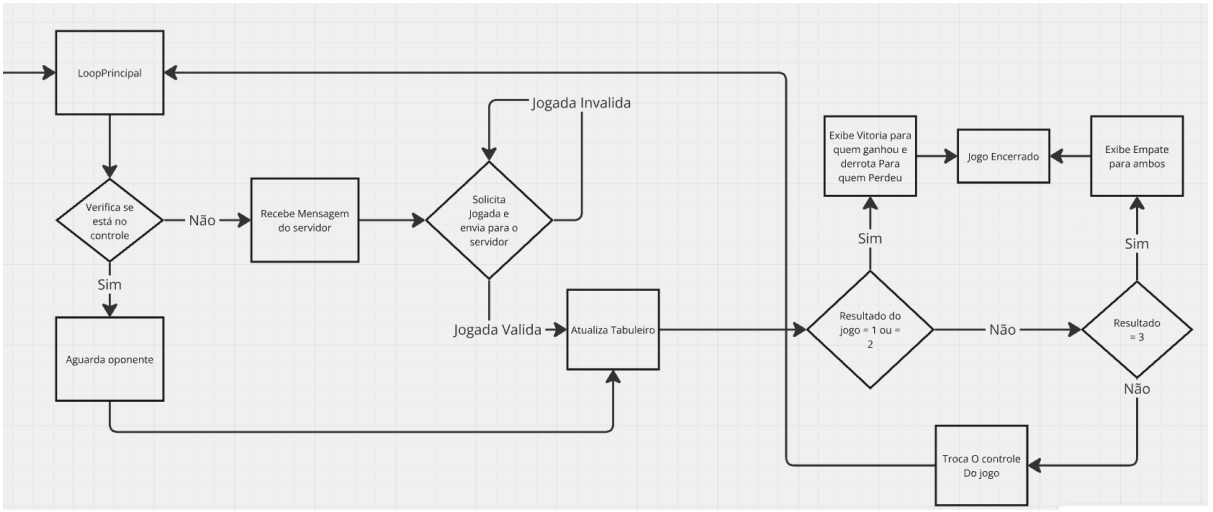
```
    catch (SocketException e)
    {
        Console.WriteLine("Erro de Conexão: {0}", e);
    }
    finally
    {
        stream?.Close();
        cliente?.Close();
    }

    Console.WriteLine("Conexão encerrada.");
}
```

Se ocorrer um erro de conexão (SocketException), exibe a mensagem de erro.
Fecha o NetworkStream e o TcpClient para garantir que os recursos sejam liberados.
Exibe uma mensagem indicando que a conexão foi encerrada.

```
2 usages
static void ExibirJogo(string tabuleiro)
{
    Console.WriteLine("");
    Console.WriteLine("Jogo da Velha Atual:");
    Console.WriteLine($"{tabuleiro[0]} | {tabuleiro[1]} | {tabuleiro[2]}");
    Console.WriteLine("--+---+--");
    Console.WriteLine($"{tabuleiro[3]} | {tabuleiro[4]} | {tabuleiro[5]}");
    Console.WriteLine("--+---+--");
    Console.WriteLine($"{tabuleiro[6]} | {tabuleiro[7]} | {tabuleiro[8]}");
    Console.WriteLine("");
}
```

Formata e exibe o tabuleiro do jogo da velha no console. Cada célula do tabuleiro é exibida com suas respectivas posições e linhas separadoras


Cria o TCPcliente e a Stream Para comunicação com servidor
Armazena e cria variáveis
Tenta a conexão
Não
Conexão Bem sucedida
Sim
Recebe a mensagem do
MSG = 1
MSG = 0
Aguarda o outro Jogador
Não
Jogador entrou?
Sim
Inicia o Jogo
A Partida Vai Começar
Recebe Mensagem do servidor
Exibe o Tabuleiro
Verifica se o SimboloJogador é = 0 se sim fala que está no controle


LoopPrincipal
Verifica se está no controle
Não
Recebe Mensagem do servidor
Solicita Jogada e envia para o servidor
Jogada Invalida
Sim
Aguarda oponente
Jogada Valida
Atualiza Tabuleiro
Resultado do jogo = 1 ou = 2
Sim
Não
Exibe Vitoria para quem ganhou e derrota Para quem Perdeu
Jogo Encerrado
Resultado = 3
Sim
Exibe Empate para ambos
Não
Troca O controle Do jogo