

Colaboração
Arthur Rossi Cardoso
arthurrossi15@gmail.com
github.com/arthurrossi15

Servidor

Afazeres.cs

```
using System;
using System.Collections.Generic;
using System.Net.Sockets;
using System.Threading;

namespace ToDo_Server
{
    class Afazeres
    {
        private static int contador = 0;
        private static readonly List<Tarefa> tarefas = new List<Tarefa>();
        private static readonly object synchronized = new object();

        static void Main(string[] args)
        {
            Console.WriteLine("Servidor Afazeres");
            try
            {
                TcpListener servidor = new TcpListener(50000);
                servidor.Start();
                while (true)
                {
                    Socket socket = servidor.AcceptSocket();
                    Cliente c = new Cliente(socket);
                    ThreadStart ts = new ThreadStart(c.Run);
                    Thread t = new Thread(ts);
                    t.Start();
                }
            }
            catch (Exception e)
            {
                Console.WriteLine("Erro: " + e.Message);
            }
        }

        public static void Put(string tarefa)
        {
            lock (synchronized)
            {
                tarefas.Add(new Tarefa(++contador, tarefa));
            }
        }

        public static string Get(int numero)
        {
            lock (synchronized)
            {
                foreach (Tarefa tarefa in tarefas)
                {
                    if (tarefa.numero == numero)
                    {
                        string afazer = tarefa.tarefa;
                        tarefas.Remove(tarefa);
                        return afazer;
                    }
                }
            }
        }
    }
}
```

```

        return null;
    }
}

public static string List()
{
    lock (synchronized)
    {
        string afazeres = "";
        for (int i = 0; i < tarefas.Count; i++)
        {
            Tarefa tarefa = tarefas[i];
            afazeres += tarefa.numero + ". " + tarefa.tarefa;
            if (i < tarefas.Count - 1) afazeres += "\n";
        }
        return string.IsNullOrEmpty(afazeres) ? null : afazeres;
    }
}
}
}
}

```

Tarefa.cs

```

namespace ToDo_Server
{
    class Tarefa
    {
        public int numero { get; private set; }
        public string tarefa { get; private set; }

        public Tarefa(int numero, string tarefa)
        {
            this.numero = numero;
            this.tarefa = tarefa;
        }
    }
}

```

Cliente.cs

```

using System;
using System.IO;
using System.Net.Sockets;

namespace ToDo_Server
{
    class Cliente
    {
        private readonly Socket socket;

        public Cliente(Socket socket)
        {
            this.socket = socket;
        }

        public void Run()
        {
            try
            {
                NetworkStream socketStream = new NetworkStream(socket);
                BinaryWriter output = new BinaryWriter(socketStream);
                BinaryReader input = new BinaryReader(socketStream);

                string requisicao, resposta;
                do
                {
                    requisicao = input.ReadString();
                    Console.WriteLine(requisicao);
                    resposta = GetRespostaTDP(requisicao);
                    output.Write(resposta);
                }
            }
            catch { }
        }
    }
}

```

```

        } while (requisicao != "EXIT");
    }
    catch (Exception e)
    {
        Console.WriteLine("Erro: " + e.Message);
    }
}

private string GetRespostaTDP(string requisicao) // TDP - ToDo Protocol
{
    if (requisicao.StartsWith("PUT ") && requisicao.Length > 4)
    {
        string tarefa = requisicao.Substring(4);
        Afazeres.Put(tarefa);
        return "OK";
    }
    else if (requisicao.StartsWith("GET ") && requisicao.Length > 4)
    {
        try
        {
            int numero = int.Parse(requisicao.Substring(4));
            string tarefa = Afazeres.Get(numero);
            return tarefa == null ? "NOT-FOUND" : numero + " " + tarefa;
        }
        catch (Exception)
        {
            return "ERR";
        }
    }
    else if (requisicao == "LIST")
    {
        string afazeres = Afazeres.List();
        return afazeres == null ? "EMPTY" : afazeres;
    }
    else if (requisicao == "EXIT")
    {
        return "BYE";
    }
    else
    {
        return "ERR";
    }
}
}
}

```

Cliente

Afazer.cs

```

using System;
using System.IO;
using System.Net.Sockets;

namespace ToDo_Client
{
    class AFazer
    {
        static void Main(string[] args)
        {
            try
            {
                TcpClient client = new TcpClient();
                client.Connect("localhost", 50000);

                Console.WriteLine("Conectado ao servidor Afazeres...");

                NetworkStream socketStream = client.GetStream();
                BinaryWriter output = new BinaryWriter(socketStream);
            }
            catch (Exception e)
            {
                Console.WriteLine("Erro: " + e.Message);
            }
        }
    }
}

```

```
BinaryReader input = new BinaryReader(socketStream);
while (true)
{
    Console.Write("> ");
    string requisicao = Console.ReadLine();
    if (string.IsNullOrEmpty(requisicao.Trim())) continue;
    output.Write(requisicao);
    string resposta = input.ReadString();
    if (resposta != "OK") Console.WriteLine(resposta);
    if (requisicao == "EXIT")
    {
        client.Close();
        return;
    }
}
}
catch (Exception e)
{
    Console.WriteLine("Erro: " + e.Message);
}
}
}
```