

Universidade Paulista

Ciência da Computação

Atividade Prática Supervisionada

FERRAMENTA DE COMUNICAÇÃO EM REDE

Aleksander Rocha - R.A.: C630IH-0

Daniel Jançanti - R.A.: C630IG-2

Ingrid Oliveira - R.A.: C51791-7

Rafaela Aranas - R.A.: C29CII-0

Sumário

1	Objetivo e motivação do trabalho	3
2	Introdução	4
3	Fundamentos de Comunicação de Dados em Rede	6
3.1	História das Tecnologias de Rede	6
3.2	Conceitos Sobre Redes	6
3.3	Classificação de Redes	8
3.4	Espaço de Redes	8
3.5	Modelos De Comunicação de Rede	11
3.5.1	Modelo OSI	11
3.5.2	Protocolo TCP	12
4	Plano de desenvolvimento da aplicação	13
5	Projeto	14
6	Código fonte	17
7	Bibliografia	38

1 Objetivo e motivação do trabalho

O objetivo desse trabalho consiste em estudar a comunicação de dados e implementar uma ferramenta para comunicação em rede. Após pesquisa bibliográfica sobre o assunto em questão o grupo deverá criar uma aplicação a qual permita que duas ou mais pessoas possam se comunicar em uma rede, utilizando o protocolo TCP/IP.

A ferramenta será utilizada no seguinte caso: "a empresa fictícia Ambiente que faz prevenção de acidentes ambientais resolveu desenvolver uma solução para baratear os custos de ligações telefônicas, essa consiste em um chat de comunicação em rede interna e também online para contato com os interessados".

A ferramenta será implementada na linguagem C#, e além da comunicação serão acrescentados recursos como login e cadastro de usuário, comunicação um a um e em broadcast onde um usuário envia mensagem para vários, *emoticons*, envio e recebimento de arquivos, etc.

2 Introdução

A comunicação dados é a teoria da ciência da computação que trata da comunicação entre computadores e dispositivos diferentes através de um meio de transmissão comum. A comunicação de dados necessita de uma infraestrutura de rede, composta de computadores ou hosts, roteadores, switches, hubs, cabeamento estruturado, protocolos de comunicação, etc. Pode ocorrer uma interligação entre redes e a Internet Pública, e essa infraestrutura segue especificações em normas de padronização, chamadas de RFC (sigla do original em inglês, Request for Comments) ?.

A comunicação de dados também ocorre através das redes de telecomunicações, as quais estão em constante aperfeiçoamento para suportar a transmissão de informações com a introdução de novas tecnologias, tanto do lado dos equipamentos da rede, quanto dos meios de transmissão (redes de transporte) e dos sistemas de operação para gerenciamento.

Uma rede pode ser composta de várias sub-redes, dependentes do tipo de serviço que é provido ao consumidor, como telefonia fixa e telefonia celular, e isso caracteriza uma rede de telecomunicações. Já a internet, na verdade trata-se de um sistema de redes de computadores interconectadas de proporções mundiais, atingindo muitos países e agregando cerca de mais de 300 milhões de computadores e mais de 400 milhões de usuários. Os computadores pessoais ou redes locais, se conectam a provedores de acesso, que se ligam a redes regionais que, por sua vez, se unem à redes nacionais e internacionais. A informação pode viajar através de todas essas redes até chegar ao seu destino. Aparelhos chamados “roteadores”, instalados em diversos pontos da rede, se encarregam de determinar qual a rota mais adequada.

A internet atual surgiu em meados dos anos 60, como uma ferramenta de comunicação militar alternativa, que resistisse ao conflito nuclear mundial, que não necessitasse de nenhum controle central. Para tanto, as mensagens passariam a ser divididos em “pacotes”, os quais seriam transmitidos com mais rapidez, flexibilidade e tolerância a erros, pois cada computador se torna um “nó” da rede. E sendo apenas um “nó” se este ficar impossibilitado de operar não interferiria no fluxo da rede.

No Brasil a internet teve seu desenvolvimento no meio acadêmico e científico, e instituições de pesquisa. Somente no ano de 1995 a internet deixou de ser privilégio das universidades e da iniciativa privada para se tornar de acesso público. Desde então o número de provedores que oferecem o serviço e número de usuários que

utilizam este recurso aumentam a cada ano.

Diante desse contexto a proposta deste trabalho consiste em estudar o funcionamento da rede de dados e implementar uma ferramenta de comunicação. Esse estudo será abordado no capítulo 3 o qual consiste do referencial teórico. Após, será abordado no capítulo 4 o desenvolvimento da ferramenta de comunicação, depois no capítulo 5 os resultados e discussões, isto é, como foram aplicados os testes e sobre o desempenho obtido pela ferramenta.

3 Fundamentos de Comunicação de Dados em Rede

Segundo GALLO (2003), uma rede é um conjunto de sistemas que possuem uma forma de comunicação entre si com o objetivo de compartilhar informações.

Como exemplo, podemos citar a rede de telefonia. Cada telefone desta rede possui ligação com qualquer outro telefone - desde que você saiba o seu número. Basta você discar o telefone de uma pessoa e com isso você estabelecerá uma conexão entre o seu telefone e o telefone dela. Os dois aparelhos irão mandar dados uns para os outros - no caso, a conversa entre você e a pessoa do outro lado da linha.

Outro exemplo é a televisão. Os programas de televisão também chegam à você por meio de uma rede. Mas esta possui características bem diferentes das redes de telefonia. Nela não se pode enviar informações para as emissoras de televisão. Somente elas transmitem informações, para você e para milhares de outras pessoas.

3.1 História das Tecnologias de Rede

Uma rede de computadores envolve a interconexão entre dois ou mais micros, o que permite a troca de dados entre essas unidades e otimiza os recursos de hardware e software. Deve ter regras básicas que garantam o envio seguro de informações. Para ser eficiente, ela precisa que os dados transitem de um computador para outro sem que sofram danos. Também é necessário que a rede seja capaz de determinar corretamente para onde as informações estão indo. Além disso, os computadores interligados tem que poder se identificar uns aos outros e deve existir um modo padronizado de nomear e identificar as partes que a compõem. Atualmente, existem milhões de máquina conectadas à Internet e ela tornou-se tão poderosa que é capaz de transmitir entre computadores todo o tipo de dados como imagens, sons, vídeos, textos escritos e até mesmo programas de computador. ?.

3.2 Conceitos Sobre Redes

Redes de Computadores formam uma tecnologia de rede única. Nenhuma outra tecnologia de rede é tão versátil e poderosa como ela. ?. Devido à isso, quando falamos sobre elas, podemos utilizar os seguintes termos:

Clientes são computadores que se conectam à um computador central para requisitar que este realize alguma tarefa na qual é especializado.

Confiabilidade Em todo o tipo de comunicação à distância, existe a possibilidade de ocorrer um erro na hora de se interpretar os dados. No caso das redes de computadores, isso é algo que pode ocorrer devido à vários motivos como interferência ou o enfraquecimento do sinal com a distância. Para se criar uma rede de computadores confiável, é preciso fazer com que os computadores sejam capazes de detectar erros na transmissão. Uma vez que isso ocorra, pode-se tentar corrigí-los ou então pedir para que os dados sejam retransmitidos.

Endereço para que possamos nos comunicar com outro elemento de uma rede, precisamos identificá-lo de alguma forma. Na rede telefônica, por exemplo, para falarmos com outra pessoa, precisamos discar o seu número de telefone - que é único para cada elemento da rede. O mesmo ocorre com a rede de computadores. Cada elemento possui um número único que é reconhecido como seu "Endereço". Quase todos os elementos de uma rede de computadores possuem um endereço. Chamamos o ato de distribuir Endereços para os elementos da rede de Endereçamento.

Meio É o ambiente físico usado para conectar membros de uma rede. Por exemplo, no caso dos telefones, o meio é o fio que forma toda a rede telefônica. Computadores podem usar os mais diversos meios, como cabos e ondas de rádio.

Nó Não são apenas computadores que podem ser ligados à uma rede de computadores. de fato, as primeiras redes de computadores foram criadas para controlar o caminho percorrido por ligações telefônicas. Existe uma gama muito grande de dispositivos que podem fazer parte deste tipo de rede como terminais, impressoras, repetidoras, pontes, chaves e roteadores. Por causa disso, costumamos chamar cada elemento conectado à uma Rede de Computadores de "Nó".

Protocolo Computadores só podem lidar com números binários. Eles só entende 0s e 1s. Por conta disso, é preciso criar algum tipo de alfabeto ou padrão para que possamos nos comunicar com apenas dois tipos de sinais. O nome das regras que os computadores seguem para se comunicar entre si chama-se "Protocolo".

Roteamento Rotear significa traçar uma rota. O roteamento é justamente a tarefa de traçar rotas entre os vários elementos de uma rede. Afinal, em uma rede com várias máquinas, é preciso estabelecer qual caminho os dados precisam seguir para que eles não terminem indo parar na máquina errada.

Segurança É comum que informações sigilosas sejam trocadas em uma rede. Por causa disso, existem muitas pessoas que podem tentar interceptar os dados. Para isso, pode-se utilizar várias estratégias para aumentar a segurança de uma rede como criptografar os dados, por exemplo.

Servidor Um Servidor é uma máquina que costuma ser freqüentemente acessada por outras para que ela realize algum tipo de tarefa.

3.3 Classificação de Redes

Quanto ao Tamanho das redes, ? diz que podemos classificá-las como:

PAN O alcance destas redes normalmente é o de alguns poucos metros.

LAN Qualquer rede cujo raio de alcance seja menor do que 10 Km se encaixa nesta categoria.

MAN Este nome é usado para redes maiores do que LANs e que normalmente ocupam a área de uma cidade inteira. Embora existam MANs que pertencem à uma única organização, o mais normal é que elas sejam formadas por redes interconectadas de vários indivíduos e organizações diferentes. Elas também podem ser usadas pela administração do município como serviços de utilidade pública.

WAN Qualquer rede cuja área é maior do que uma cidade se encaixa nesta categoria. Existem WANs que possuem uma área de alcance que cruzam até mesmo diferentes estados e países. Atualmente, a maior WAN existente é a Internet.

3.4 Espaço de Redes

Segundo ?, o espaço de uma rede mostra como as máquinas estão ligadas entre si. Dessa forma as redes podem ser classificadas em:

Redes Ponto-a-Ponto Em redes deste tipo, cada nó só pode se comunicar com nós adjacentes. É como em uma brincadeira de telefone sem fio no qual para que uma mensagem chegue até alguém, ela precisa passar por vários intermediários, já que só é possível falar com as pessoas que estejam ao seu lado.

Estrela Neste tipo de rede, existe um nó central (normalmente um hub ou switch) à partir do qual todas as máquinas estão conectadas. Para enviar uma men-

sagem à alguém, é preciso primeiro enviar para o nó central e só então o nó central passa a mensagem para o destinatário.



Figura 1: Rede Ponto-a-Ponto Estrela

Laço Neste tipo de rede, não existe um nó central. Ao invés disso, as máquinas então todas conectadas entre si e existem nós que estão conectados a mais de um outro nodo. Por não possuírem um nó central, não existe um único ponto cujo funcionamento mantém a rede inteira. Por isso, eles tendem a ser mais seguros. Entretanto, o roteamento neste tipo de rede tende a ser mais complexo. Existem também redes em laço que são totalmente conectadas. Nelas, cada nó está conectado à todos os demais. Por causa de sua complexidade e custo proibitivo, este tipo de laço só é usado em redes pequenas com poucos nós.



Figura 2: Rede Ponto-a-Ponto Laço

Árvore Neste tipo de topologia, existe um nó que é considerado a raiz. Ela possui ligada à ela outros nós que são considerados seus filhos e ele é o pai destes nós. Cada nó que é filho da raiz pode ter outros filhos e estes também podem ter seus filhos. Entretanto, cada nó, com exceção da raiz, deve

possuir um único pai. Normalmente, estas redes possuem como nós diversos Hubs ou Switchs. Nelas, os nós que não possuem filhos normalmente são os computadores e terminais de trabalho.



Figura 3: Rede Ponto-a-Ponto Árvore

Redes de Difusão Neste tipo de rede, os nós compartilham um canal único de comunicação. Nele, os dados enviados por uma máquina são recebidos por todos os nós que compartilham um mesmo canal. É como em uma conversa normal. Quando você fala, várias pessoas ao redor ouvem o que você disse, mas somente a pessoa com quem você está falando responde.

Barramento Neste tipo de rede, todos os nós compartilham um mesmo canal. Se algum dos nós enviar uma mensagem pela rede, todos os demais irão ouvir. Deve-se tomar cuidado para que mais de um nó não tente falar ao mesmo tempo, pois se isso ocorrer, ninguém conseguirá entender a mensagem transmitida.



Figura 4: Rede de Difusão Barramento

Satélite Segundo ?, neste tipo de rede, existe um satélite capaz de transmitir dados para todos os nós em Terra que estejam na área de alcance e estejam equipados com antenas para captar o seu sinal. Se o satélite envia um sinal, todos os outros nós ouvem. Mas se um nó mandar uma mensagem para o satélite, somente o satélite será capaz de ouvir a mensagem.

3.5 Modelos De Comunicação de Rede

Redes de computadores são estruturas bastante complexas que para funcionarem, precisam que os seus diversos componentes funcionem de forma sincronizada e colaborativa. Entretanto, se cada fabricante de hardware e software para redes fizer com que seus produtos tratem a rede de forma diferente, torna-se impossível fazer com que os diversos produtos interajam entre si.

Para que redes de processadores tornarem-se possíveis, era preciso desenvolver uma arquitetura de redes - uma estrutura lógica e formal que especifica como os diversos componentes devem se comunicar entre si. Para isso, existem vários tipos de arquiteturas diferentes como o Modelo OSI, feito pela ISO, a SNA feita pela IBM, a DNA feita pela Digital Equipment Corporation e o Modelo TCP/IP.

3.5.1 Modelo OSI

Embora na prática o modelo OSI seja bem menos usado que o TCP/IP, ele costuma ser bastante estudado por motivos didáticos e é um ótimo ponto de partida para se compreender melhor como funcionam redes de computadores. Ele teve um impacto muito grande nas redes, inclusive no Modelo TCP/IP que é o mais usado. Ainda hoje, é comum que projetistas de rede usem o modelo OSI como um modelo teórico para auxiliar no desenvolvimento de redes e arquiteturas.

Uma das características mais úteis do Modelo OSI é a subdivisão da arquitetura de redes em diferentes camadas. Essa abstração é algo que facilita a compreensão do funcionamento das redes. Assim como na programação é mais fácil resolver um problema complexo dividindo-o em menores e criando funções específicas para cada um desses, também é mais fácil criar uma arquitetura de rede coerente separando a rede em diversas camadas menores que podem ser projetadas de forma independente das demais.

O Modelo OSI possui ao todo 7 camadas, são elas:

Camada Física é responsável por transferir os bits por meio de ligações. Ela cuida de questões como o tipo de cabo em uso e como é feita a conexão entre o cabo e a máquina.

Camada de Enlace de Dados é responsável por organizar os dados recebidos, colocando-os na ordem correta, detectando e talvez corrigindo eventuais erros de transmissões.

Camada de Rede cuida do estabelecimento de rotas e do chaveamento dos dados ao longo da rede.

Camada de Transporte é responsável por quebrar a mensagem em pacotes menores para que ela seja transmitida. Também é responsável por depois montar os diversos pacotes em uma única mensagem posteriormente.

Camada de Sessão cuida das regras de comunicação entre os nós que estão trocando mensagens. Ela verifica quando é possível ou não mandar dados e também sabe que tipo de comunicação os nós possuem (simplex, duplex, semi-duplex).

Camada de Apresentação é responsável por formatar e estruturar os dados de uma forma que eles possam ser entendidos por outra máquina. Ela cuida da criptografia se necessário.

Camada de Aplicação é responsável por cuidar das informações que chegam pela rede para cada programa de computador que está sendo usado no computador.

3.5.2 Protocolo TCP

O TCP (Transmission Control Protocol - Protocolo de Controle de Transmissão) é um dos protocolos, sob os quais assenta o núcleo da Internet nos dias de hoje. A versatilidade e robustez deste protocolo tornou-o adequado para redes globais, já que este verifica se os dados são enviados de forma correta, na sequência apropriada e sem erros, pela rede.

4 Plano de desenvolvimento da aplicação

Diante de um problema de comunicação frequente entre as pessoas: a questão da agilidade que as informações são transmitidas, pensamos numa solução onde uma única pessoa teria o poder de se comunicar com várias outras de maneira rápida, prática e intuitiva.

Nosso chat vai transmitir dados de um cliente para outro e também de um cliente para vários, tendo que evitar um método retrógrado que seria o uso do telefone. Embora o telefone ainda seja o meio de comunicação mais usado no mundo, ele ainda falha na questão de comunicar várias pessoas ao mesmo tempo.

Baseando-se nessas informações, criamos um meio de comunicação para solucionar essa falha.

Inicialmente a implementação do chat inclui: cadastro de novos usuários, login de usuário, comunicação um para um, comunicação um para muitos e envio de arquivos (documentos, planilhas, fotos, etc), usando o banco de dados mySQL como armazenamento e transmissão de dados por sockets pelo protocolo TCP/IP.

5 Projeto

Para a construção do nosso projeto optamos a linguagem C# pela facilidade e infinidade de recursos para esse tipo de aplicação. A construção do projeto foi feita na IDE Visual Studio com os recursos providos do software.

Para melhorar a usabilidade criamos algumas telas para facilitar a interação do usuário com o software.

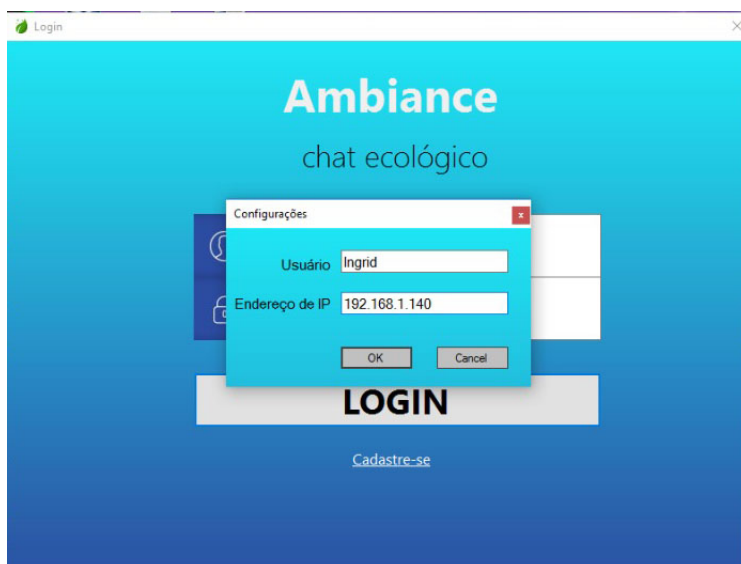


Figura 5: Tela de configurações

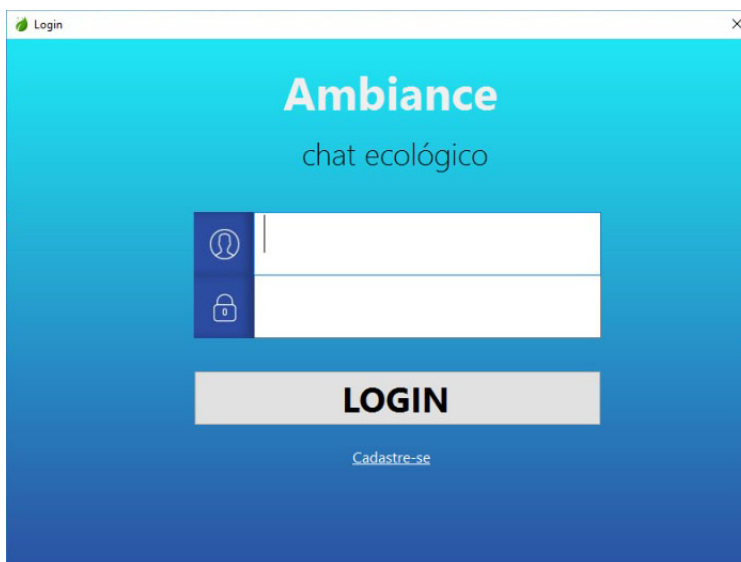


Figura 6: Tela de login

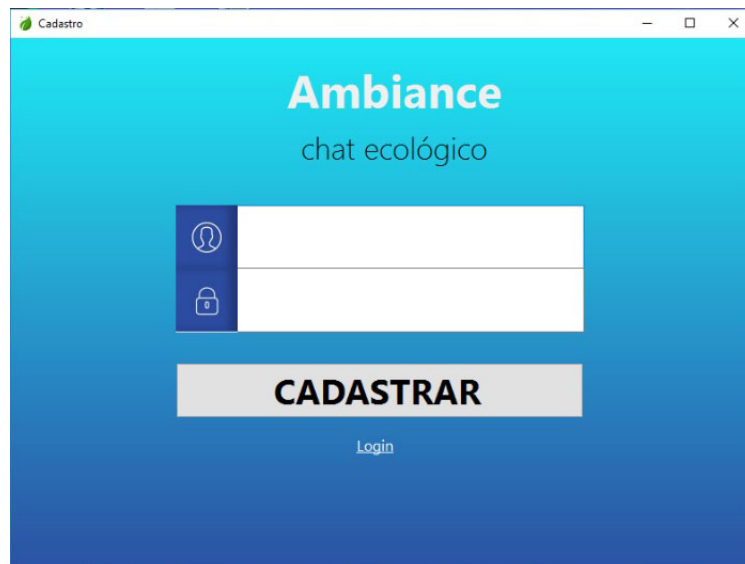


Figura 7: Tela de cadastro

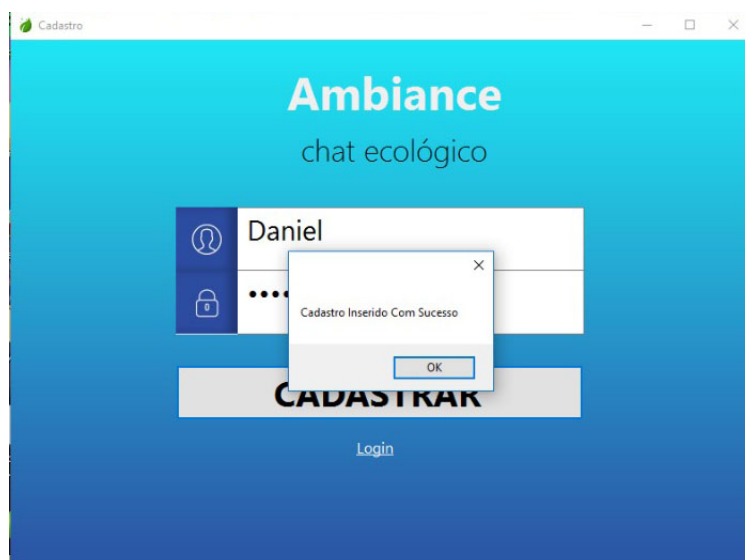


Figura 8: Tela de cadastro inserido com sucesso

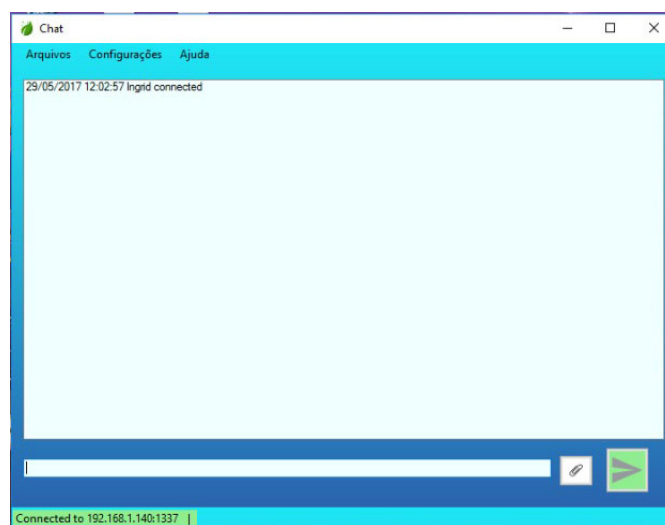


Figura 9: Aplicação em funcionamento

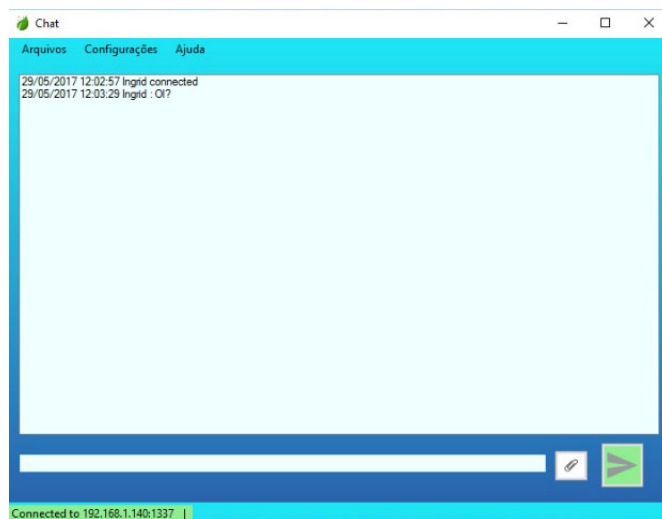


Figura 10: Mensagem enviada

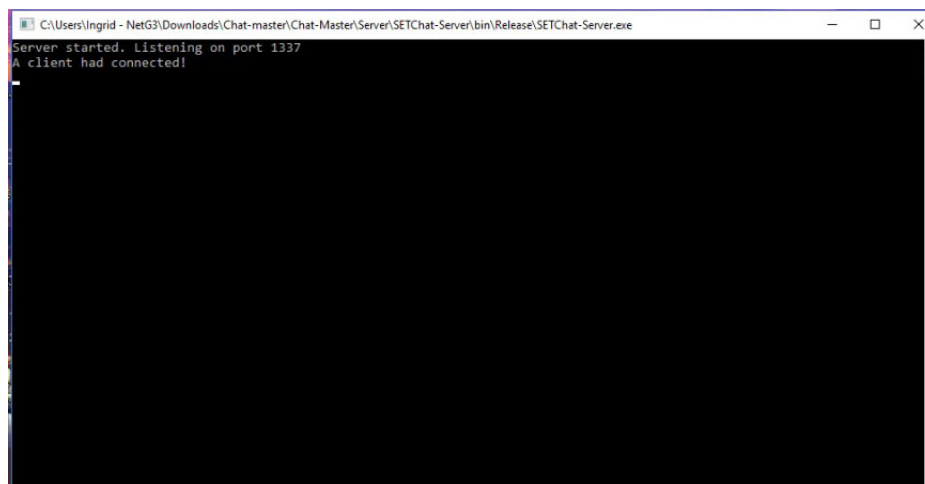


Figura 11: Log do servidor quando cliente se conecta

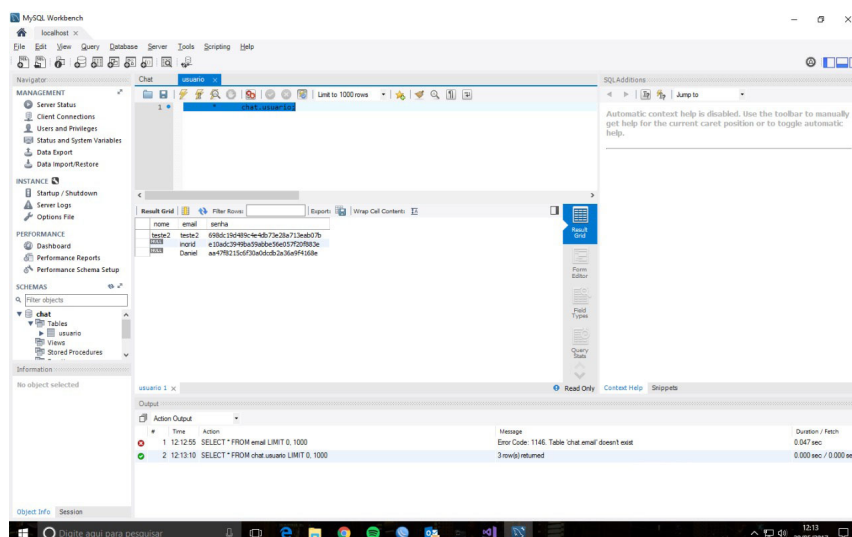


Figura 12: Tabela de senha do banco de dados criptografada com MD5

6 Código fonte

SERVIDOR

Servidor-classe-program.cs

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Net;
using System.Net.Sockets;

namespace SETChat_Server
{
    class Program
    {
        public static List<string> msg_Queue = new List<string>();
        public static List<Thread> cli_Conns = new List<Thread>();
        public static List<TCPComms> cli_List = new List<TCPComms>();

        private const int tcp_Port = 1337;

        static void Main(string[] args)
        {
            Boolean done = false;
            Mutex mutex = new Mutex();

            IPEndPoint ipHostInfo = Dns.Resolve(Dns.GetHostName());
            IPAddress ipAddress = ipHostInfo.AddressList[0];
            IPEndPoint localEndPoint = new IPEndPoint(ipAddress, tcp_Port);

            Socket listener = new Socket(AddressFamily.InterNetwork,
                SocketType.Stream, ProtocolType.Tcp);

            try
            {
                listener.Bind(localEndPoint);
                listener.Listen(10);

                Console.WriteLine("Server started. Listening on port {0}, IP {1}", tcp_Port, ipAddress);

                QueryReader reader = new QueryReader(msg_Queue, cli_List, mutex);
                Thread t = new Thread(reader.Run);
                t.Start();
            }
            catch { }
        }
    }
}
```

```

        while (done == false)
        {
            Socket handler = listener.Accept();
            Console.WriteLine("A client had connected!");
            mutex.WaitOne();
            cli_List.Add(new TCPComms(handler, msg_Queue, mutex));
            cli_Conns.Add(new Thread(cli_List[cli_List.Count - 1].Run));
            cli_Conns[cli_Conns.Count - 1].Start();
            mutex.ReleaseMutex();
        }
    } catch (Exception ex)
    {
    }
}
}
}

```

Servidor-Classe-QueryReader.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;

namespace SETChat_Server
{
    class QueryReader
    {
        List<String> messageQuery;
        List<TCPComms> listenerList;
        Boolean done;
        Mutex mutex;

        /// <summary>
        /// Method : constructor for object
        /// </summary>
        /// <param name="query">Message list from the users</param>
        /// <param name="listeners">List of clients</param>
        public QueryReader(List<String> query, List<TCPComms> listeners, Mutex mut)
        {
            messageQuery = query;
            listenerList = listeners;
            mutex = mut;
            done = false;
        }

        public void Run()
        {
            while (done == false)

```

```

        {
            mutex.WaitOne();
            for (int i = 0; i < listenerList.Count; i++)
            {
                if (listenerList[i].Connected == false)
                {
                    listenerList.RemoveAt(i);
                }
            }

            if (messageQuery.Count > 0)
            {
                foreach (TCPComms listener in listenerList)
                {
                    listener.send_Msg(messageQuery[0]);
                }
                messageQuery.RemoveAt(0);
                Thread.Sleep(10);
            }
            mutex.ReleaseMutex();
        }
    }
}

```

Servidor-Classe-TCP.cs

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Net.Sockets;
using System.Net;
using System.IO;
using System.IO.Ports;

namespace SETChat_Server
{
    class TCPComms
    {
        Socket handler;
        List<String> msgQuery;
        Mutex mutex;
        public Boolean Connected { get { return handler.Connected; } }

        /// <summary>
        /// Constructor for the object
        /// </summary>
    }
}

```

```

/// <param name="socket"></param>
/// <param name="query"></param>
/// <param name="mut"></param>
public TCPComms(Socket socket, List<String> query, Mutex mut)
{
    handler = socket;
    msgQuery = query;
    mutex = mut;
}

private String Rec_Msg()
{
    byte[] buf = new byte[1024];

    int bytesRecv = handler.Receive(buf);

    String str = Encoding.ASCII.GetString(buf, 0, bytesRecv);

    return str;
}

/// <summary>
/// Method : send_Msg
/// Takes a string, appends datestamp to it, and sends it to handler
/// </summary>
/// <param name="data"></param>
public void send_Msg(String data)
{
    byte[] msg = Encoding.ASCII.GetBytes(DateTime.Now + " " + data);

    handler.Send(msg);
}

public void Run()
{
    String message;

    while( handler.Connected )
    {
        try
        {
            message = "";
            if ((message = Rec_Msg()).Equals("") == false)
            {

```

Servidor-Transferencia-Program.cs

Usuario.cs

21

UsuarioControler.cs

```
using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ChatEco
{
    public class UsuarioControler
    {
        public List<Usuario> ListarUsuarios()
        {
            string Configuracao = "server=localhost; user id=root; password=chat2017; database=chat";
            MySqlConnection Conexao = new MySqlConnection(Configuracao);
            try
            {
                List<Usuario> IstUsuario = new List<Usuario>();
                string sql = $"SELECT * FROM usuario";
                MySqlCommand Comando = new MySqlCommand();
                Comando.Connection = Conexao;
                Comando.CommandText = sql;
                Conexao.Open();
                MySqlDataReader dtreader = Comando.ExecuteReader();

                while (dtreader.Read())
                {
                    IstUsuario.Add(new Usuario {
                        Nome = dtreader["nome"].ToString(),

                    });
                }
                Conexao.Close();

                return IstUsuario;
            }
            catch (Exception e)
            {
                return null;
            }
        }
    }
}
```

CLIENTE

cliente-cadastro.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
```

```

using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using System.Security.Cryptography;

namespace ChatEco
{
    public partial class Cadastro : Form
    {
        Login log;
        public Cadastro(Login login)
        {
            InitializeComponent();

            log = login;
        }

        private void linkLabel2_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
        {
            log.Show();

            Close();
        }

        private void Cadastro_Load(object sender, EventArgs e)
        {
        }

        private void btnLogin_Click(object sender, EventArgs e)
        {
            log.Show();
        }

        private void btn_Login_Click(object sender, EventArgs e)
        {
            try
            {
                if ( (!txtBox_Usu.Text.Equals("")) & (!txt_senha1.Text.Equals("")) )
                {
                    string Configuracao = "server=localhost; user id=root; password=chat2017; database=chat";
                    MySqlConnection Conexao = new MySqlConnection(Configuracao);

                    Conexao.Open();
                    MySqlCommand INSERT = new MySqlCommand("INSERT INTO usuario (email, senha) VALUES (@user, " +
                    string hash = "";
                    using (MD5 md5Hash = MD5.Create())
                    {

```

```

        hash = MD5Util.GetMd5Hash(md5Hash, txt_senha1.Text);

    }
    INSERT.Parameters.AddWithValue("@pass", hash);
    INSERT.Parameters.AddWithValue("@user", txtBox_Usu.Text);

    INSERT.ExecuteNonQuery();
    Conexao.Close();

    MessageBox.Show("Cadastro Inserido Com Sucesso");
}
else
{
    MessageBox.Show("Preencha todo formulario");
}
}

catch (Exception ex)
{
    MessageBox.Show(ex.Message.ToString());
    MessageBox.Show("MySQL Nao Conectado!", "ERRO DE CONEXAO");
}

}

private void txt_senha1_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == (char)13)
    {
    }
}
}
}

```

cliente-chat.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading;
using System.Windows.Forms;
using System.Net;
using System.Net.Sockets;
using System.Threading.Tasks;

```



```

using A6;

namespace ChatEco
{
    public partial class MainForm : Form
    {
        private int port = 01337;

        String username;
        String IPAddress;
        Boolean connected;

        Socket sender;

        Listener listener;
        Thread t;

        List<String> messageQuery = new List<String>();
        List<String> fileQuery = new List<String>();

        public MainForm()
        {
            InitializeComponent();
            username = "noname";
            IPAddress = "127.0.0.1";

            connected = false;
            connectionLabel.Text = "Disconnected";

            sendButton.Enabled = false;

            getSettings();
            connect();
        }

        private void informationMenuItem_Click(object sender, EventArgs e)
        {
            getSettings();
        }

        private void getSettings()
        {
            Settings settingWindow = new Settings(username, IPAddress);
            settingWindow.ShowDialog();

            username = settingWindow.GetName();
            IPAddress = settingWindow.GetAddress();
        }

        private void connect()

```

```

{
    try
    {
        System.IO.File.AppendAllText("log.txt", "");

        sender = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

        sender.Connect(IPAddress, port);
        sendMsg(username + " connected");

        listener = new Listener(sender, messageQuery);
        t = new Thread(new ThreadStart(listener.Run));
        t.Start();

        MainTimer.Start();

        connected = true;

        sendButton.Enabled = true;
        sendButton.BackColor = Color.LightGreen;

        connectionLabel.Text = "Connected to " + sender.RemoteEndPoint.ToString();
        errorLabel.Text = "";
    }
    catch (SocketException se)
    {
        errorLabel.Text = "Socket Error";
        using (System.IO.StreamWriter dataFile = new System.IO.StreamWriter("log.txt", true))
        {
            dataFile.WriteLine(DateTime.Now + " | " + se.ToString() + '\n');
        }
    }
    catch (Exception e)
    {
        errorLabel.Text = "Unexpected Error";
        using (System.IO.StreamWriter dataFile = new System.IO.StreamWriter("log.txt", true))
        {
            dataFile.WriteLine(DateTime.Now + " | " + e.ToString() + '\n');
        }
    }
}

private void disconnect()
{
    try
    {

        sendMsg(username + " disconnected");
    }
}

```

```

sender.Shutdown(SocketShutdown.Both);
sender.Close();

MainTimer.Stop();

connected = false;
sendButton.Enabled = false;
sendButton.BackColor = Color.LightGray;
connectionLabel.Text = "Disconnected";
errorLabel.Text = "";
}
catch (SocketException se)
{
    errorLabel.Text = "Socket Error";
    using (System.IO.StreamWriter dataFile = new System.IO.StreamWriter("log.txt", true))
    {
        dataFile.WriteLine(DateTime.Now + " | " + se.ToString() + '\n');
    }
}
catch (Exception e)
{
    errorLabel.Text = "Unexpected Error";
    using (System.IO.StreamWriter dataFile = new System.IO.StreamWriter("log.txt", true))
    {
        dataFile.WriteLine(DateTime.Now + " | " + e.ToString() + '\n');
    }
}
}

private void sendMsg(String data)
{
    byte[] msg = Encoding.ASCII.GetBytes(data);

    int bytesSent = sender.Send(msg);
}

private void connectMenuItem_Click(object sender, EventArgs e)
{
    if (connected == false)
    {
        connect();
    }
}

private void disconnectMenuItem_Click(object sender, EventArgs e)
{
    if (connected == true)
    {

```

```

        disconnect();
    }
}

private void MainForm_FormClosing(object sender, FormClosingEventArgs e)
{
    if (connected == true)
    {
        disconnect();
    }
}

private void sendButton_Click(object sender, EventArgs e)
{
    sendMsg(username + " : " + inputBox.Text + "\n");
    inputBox.Text = "";
}

private void MainTimer_Tick(object sender, EventArgs e)
{
    if (messageQuery.Count > 0)
    {
        if (messageQuery[0].Contains("-/File$"))
        {
            string[] splited = messageQuery[0].Split('$');
            string x = splited[1].Replace("\n", "");
            ClientWebDownload(x);
            messageList.Items.Add($"Voce Recebeu o arquivo {splited[1]} em C:/Upload");

        }
        else
        {
            messageList.Items.Add(messageQuery[0] + '\n');
        }

        messageQuery.RemoveAt(0);
    }

    if (connected == true)
    {
        if (listener.Connected == false)
        {
            disconnect();
        }
    }
}

```

```

        if (fileQuery.Count > 0)
        {
            sendMsg("/File$ " + fileQuery[0] + "\n");
            fileQuery.RemoveAt(0);
        }
    }

private void ClientWebDownload(string fileName)
{
    try
    {
        string remoteUri = "http://" + IPAddress + "/chat/";
        string myStringWebResource = remoteUri + fileName;
        WebClient myWebClient = new WebClient();
        myWebClient.DownloadFile(myStringWebResource, @"C:/Upload/" + fileName);
    }
    catch (Exception e)
    {
        MessageBox.Show(e.Message);
        throw;
    }
}

private void exitMenuItem_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void aboutMenuItem_Click(object sender, EventArgs e)
{
    AboutBox about = new AboutBox();
    about.ShowDialog();
}

private void btnAnexo_Click(object sender, EventArgs e)
{
    OpenFileDialog openFileDialog1 = new OpenFileDialog();
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        EnviarArquivo.EnviarAnexo(openFileDialog1.FileName, IPAddress);
        fileQuery.Add(EnviarArquivo.mensagemCliente);
    }
}

private void sendButton_Click_1(object sender, EventArgs e)
{

```

```

        sendMsg(username + " : " + inputBox.Text + "\n");
        inputBox.Text = "";
    }
}

```

cliente-configuracoes.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ChatEco
{
    public partial class Settings : Form
    {
        String username;
        String IP_address;

        public Settings()
        {
            InitializeComponent();
        }

        public Settings(String name, String address)
        {
            InitializeComponent();
            inputName.Text = username = name;
            inputAddress.Text = IP_address = address;
        }

        public String GetName()
        {
            return username;
        }

        public String GetAddress()
        {
            return IP_address;
        }

        private void buttonOK_Click(object sender, EventArgs e)
        {
            username = inputName.Text;

```

```

        IP_address = inputAddress.Text;
        this.Close();
    }

    private void buttonCancel_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void inputName_TextChanged(object sender, EventArgs e)
    {
    }

    private void Settings_Load(object sender, EventArgs e)
    {
    }
}

```

cliente-enviar-arquivo.cs

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Net.Sockets;
using System.Text;
using System.Threading.Tasks;

namespace A6
{
    public static class EnviarArquivo
    {
        public static string mensagemCliente = "em espera";
        public static void EnviaArquivo(string nomeArquivo, string strEnderecoIP)
        {
            try
            {
                IPEndPoint ipEnd_cliente = new IPEndPoint(IPAddress.Parse(strEnderecoIP), 1557);
                Socket clientSock_cliente = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

                string caminhoArquivo = "";
                nomeArquivo = nomeArquivo.Replace("\\", "/");

                while (nomeArquivo.IndexOf("/") > -1)
                {
                    caminhoArquivo += nomeArquivo.Substring(0, nomeArquivo.IndexOf("/") + 1);
                    nomeArquivo = nomeArquivo.Substring(nomeArquivo.IndexOf("/") + 1);
                }
            }
        }
    }
}

```

```

byte[] nomeArquivoByte = Encoding.UTF8.GetBytes(nomeArquivo);
if (nomeArquivoByte.Length > 5000 * 1024)
{
    mensagemCliente = "O tamanho do arquivo e maior que 5Mb, tente um arquivo menor.";
    return;
}

string caminhoCompleto = caminhoArquivo + nomeArquivo;
byte[] fileData = File.ReadAllBytes(caminhoCompleto);
byte[] clientData = new byte[4 + nomeArquivoByte.Length + fileData.Length];
byte[] nomeArquivoLen = BitConverter.GetBytes(nomeArquivoByte.Length);
nomeArquivoLen.CopyTo(clientData, 0);
nomeArquivoByte.CopyTo(clientData, 4);
fileData.CopyTo(clientData, 4 + nomeArquivoByte.Length);
clientSock_cliente.Connect(ipEnd_cliente);
clientSock_cliente.Send(clientData, 0, clientData.Length, 0);
clientSock_cliente.Close();
mensagemCliente = nomeArquivo;
}

catch (Exception ex)
{
    mensagemCliente = ex.Message + " " + "\nFalha, pois o Servidor nao esta atendendo....";
}

}

}

```

cliente-listeners.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Net;
using System.Net.Sockets;

namespace ChatEco
{
    class Listener
    {
        private List<String> query;
        private Socket handler;

        public Boolean Connected { get { return handler.Connected; } }

        public Listener(Socket socket, List<String> stringList)

```



```

    {
        handler = socket;
        query = stringList;
    }

    public void Run()
    {
        while (handler.Connected)
        {
            try
            {
                byte[] bytes = new Byte[1024];

                int bytesRecv = handler.Receive(bytes);

                query.Add(Encoding.ASCII.GetString(bytes, 0, bytesRecv));
            }
            catch (SocketException)
            {
                using (System.IO.StreamWriter dataFile = new System.IO.StreamWriter("log.txt", true))
                {
                    dataFile.WriteLine(DateTime.Now + " | Disconnected\n");
                }
            }
        }
    }
}

```

cliente-login.cs

```

using System;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using System.Security.Cryptography;

namespace ChatEco
{
    public partial class Login : Form
    {
        public Login()
        {
            InitializeComponent();
        }

        private void Usuario_TextChanged(object sender, EventArgs e)
        {

```

```

}

private void Label1_Click(object sender, EventArgs e)
{

}

private void Fundo_Paint(object sender, PaintEventArgs e)
{

}

private void PictureBox1_Click(object sender, EventArgs e)
{

}

private void LinkLabel_Cadas_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    Cadastro objeto = new Cadastro(this);
    objeto.Show();

    this.Hide();

}

private void txt_SenhaL_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == (char)13)
    {

    }

}

private void Btn_Login_Click(object sender, EventArgs e)
{
    string Configuracao = "server=localhost; user id=root; password=chat2017; database=chat";
    MySqlConnection Conexao = new MySqlConnection(Configuracao);
    try
    {

        txt_User.Enabled = true;
        txt_SenhaL.Enabled = true;
        btn_Login.Enabled = true;
        string senhaquevemdobanco = "";
        string senhaDigitada = "";
        string sql = $"SELECT * FROM usuario where email = '{txt_User.Text}'";
        MySqlCommand Comando = new MySqlCommand();
        Comando.Connection = Conexao;
        Comando.CommandText = sql;
    }
}

```

```

        Conexao.Open();
        MySqlDataReader dtreader = Comando.ExecuteReader();
        while (dtreader.Read())
        {
            senhaquevemdobanco = dtreader["senha"].ToString();
        }
        using (MD5 md5Hash = MD5.Create())
        {
            senhaDigitada = MD5Util.GetMd5Hash(md5Hash, txt_SenhaL.Text);
        }
        if (senhaquevemdobanco.Equals(senhaDigitada))
        {
            using (var dlg = new MainForm())
            {
                dlg.StartPosition = FormStartPosition.Manual;
                dlg.Location = this.Location;
                dlg.FormClosing += (s, ea) => this.Show();    // <=== Here
                this.Hide();
                dlg.ShowDialog();
            }
        }

        else
        {
            MessageBox.Show("Erro de Login");
        }

        Conexao.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString());
        MessageBox.Show("MySQL Nao Conectado!", "ERRO DE CONEXAO");
    }
}
}

```

cliente-main-chat.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ChatEco
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.

```

```

    /// </summary>
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new Login());
    }
}

```

cliente-md5.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;

namespace ChatEco
{
    public static class MD5Util
    {
        public static string GetMd5Hash(MD5 md5Hash, string input)
        {
            // Convert the input string to a byte array and compute the hash.
            byte[] data = md5Hash.ComputeHash(Encoding.UTF8.GetBytes(input));

            // Create a new StringBuilder to collect the bytes
            // and create a string.
            StringBuilder sBuilder = new StringBuilder();

            // Loop through each byte of the hashed data
            // and format each one as a hexadecimal string.
            for (int i = 0; i < data.Length; i++)
            {
                sBuilder.Append(data[i].ToString("x2"));
            }

            // Return the hexadecimal string.
            return sBuilder.ToString();
        }

        // Verify a hash against a string.
        public static bool VerifyMd5Hash(MD5 md5Hash, string input, string hash)
        {
            // Hash the input.
            string hashOfInput = GetMd5Hash(md5Hash, input);

            // Create a StringComparer and compare the hashes.
            StringComparer comparer = StringComparer.OrdinalIgnoreCase;

```

```
        if (0 == comparer.Compare(hashOfInput, hash))
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}
```

7 Bibliografia