

UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Rafael Kaustchr

Raíza Santana

Thiago Damasceno

## **LABORATÓRIO DE REDES**

Laboratório para a disciplina de Redes de Computadores

Professor Marcel Rocha

Julho de 2014

Nova Iguaçu – RJ

## **OBJETIVO**

Desenvolver, com os conhecimentos adquiridos na disciplina de Redes de Computadores, um jogo online simples com dois jogadores ativos e um jogador observador, utilizando os padrões TCP ou UDP e a API de sockets.

O jogo escolhido foi o “Jogo do NIM”, um famoso jogo de palitos tradicionalmente chinês que estimula a divisão, conversão para binário, e até mesmo equações de primeiro grau. A versão utilizada para o laboratório foi uma ideia de melhora do jogo proposta por alunos da USP.

O jogo foi implementado em Java swing, e conta com interface gráfica.

## REGRAS DO JOGO

Este jogo consiste em colocarmos sobre uma mesa três fileiras com quantidades diferentes de palitos. Este jogo é para dois participantes, sendo assim, perde o que retirar o último palito.

É necessário seguir as seguintes regras:

1. Cada jogador, em cada jogada, deverá escolher uma fileira para retirar os palitos, sem restrição de quantidade (no mínimo um e no máximo toda fileira).
2. Os jogadores alternam suas jogadas.

Exemplo:

Fileira 1: ||||| (9 palitos)

Fileira 2: ||||| (6 palitos)

Fileira 3: |||| (4 palitos)

- Estratégia para vencer o jogo:

No exemplo citado acima, converteremos as quantidades de palitos em cada fileira por sua representação em binário:

Fileira 1: 1 0 0 1 (9 em binário)

Fileira 2: 1 1 0 (6 em binário)

Fileira 3: + 1 0 0 (4 em binário)

---

1 2 1 1

Somando-se as colunas teremos um resultado com dígitos entre 0 e três, no caso, obtivemos “1 2 1 1”. Chamaremos de combinação segura, aquela que obtiver como resultado das somas das colunas apenas os dígitos “2” e “0”. Para vencer o jogo, basta o jogador transformar este resultado (1 2 1 1) numa combinação segura, retirando palitos. Observe que, como não se pode adicionar palitos, teremos que retirar palitos da fileira 1, de modo que tenhamos uma combinação segura.

XXX

1 1 0

+ 1 0 0

---

2 2 0

Logo, na fileira 1 devemos ter “0 1 0”, que representa 2 palitos (verifique que esta é a única solução possível). Para isso, basta retirarmos 7 palitos da fileira 1.

Após conseguir uma combinação segura, o próximo jogador não poderá fazer uma nova combinação segura. Não é difícil observar isso, pense que em binário, para diminuir um número somente podemos mudar de “0” para “1” e vice-versa, logo, pelo menos um “1” se tornaria “0”, e esta coluna, que antes tinha soma “2” passa a Ter soma “1” que não é um dígito de combinação segura.

Até agora conseguimos observar que se um jogador fizer uma combinação segura, poderá mantê-la, e por que então ele ganhará o jogo?

Adicionaremos algumas exceções de combinação segura: se a soma der 3, isto é, linha1: 1 palito, linha 2: 1 palito e linha 3: 1 palito será uma combinação segura, e a menor combinação segura será a de apenas um palito no total. Também, como exceção, se a soma das linhas der 2, não será uma combinação segura.

Vamos analisar a que ocorrerá: seja P uma combinação segura e I uma não segura, A o jogador que deixa na mesa uma combinação segura e B o outro jogador, teremos o seguinte:

P -> I -> P -> I ... como os palitos estão diminuindo, poderemos chegar as seguintes combinações finais que garantirão o desfecho do jogo:

- a) Se uma das linhas for eliminada pelo jogador B, como ele não consegue deixar uma combinação segura, significa que nas linhas restantes existe um número diferente de palitos, logo, basta o jogador A igualá-los, fazendo assim, uma nova configuração segura (salvo a única exceção já citada).
- b) Se o jogador A eliminar uma fila, significa que temos a configuração final do item anterior, ou seja, ficamos com 2 filas com a quantidade igual de palitos.

Analisando os casos a) e b), a sequência vai convergir para os seguintes resultados:

1. O jogador A compõe a menor configuração segura do tipo soma = “dois” e “zero”, que é deixar dois palitos em cada fileira, nesta condição, o jogador B executará mais uma jogada e permitirá ao jogador A compor a última e menor combinação segura que é a de apenas um palito na mesa, e ganhará o jogo.
2. O jogador B elimina uma fileira inteira, assim, restando palitos apenas numa fileira, basta o jogador A deixar somente um palito nesta, e vencerá.

c) Se nenhuma fileira for eliminada, a menor configuração segura do tipo soma = “0” ou “2”, será: as linhas com 1, 2 e 3 palitos, respectivamente, nesta situação, o jogador B, se retirar uma linha inteira, recorre no caso a) e perderá o jogo, se retirar um palito da linha que tem 3, deixará duas linhas com 2 palitos, levando o jogador A ao procedimento do item b) ; finalizando, se o jogador B retirar ou dois palitos da linha que tem três, ou um palito da linha que tem dois, permitirá ao jogador A realizar a configuração segura de soma=“3” (ou seja, um palito em cada linha), e vencerá o jogo em mais uma jogada.

## DESENVOLVIMENTO

O jogo foi criado utilizando Java *Swing*, que é uma biblioteca gráfica oficial inclusa em qualquer JRE ou JDK. As APIs de interface gráfica do Java favorecem, ao máximo, o lema de portabilidade da plataforma Java. O *look-and-feel* do *Swing* é único em todas as plataformas onde roda, seja ela Windows, Linux, ou qualquer outra. Isso implica que a aplicação terá exatamente a mesma interface (cores, tamanhos etc) em qualquer sistema operacional. A parte gráfica foi usada para tornar o jogo mais amigável e intuitivo, afim de facilitar a conexão ao servidor e as jogadas dos usuários.

Como o jogo foi feito em Java, foi utilizada sua API de *sockets*, através do pacote *java.net*. Foi escolhido o protocolo TCP para comunicar servidor e cliente, afim de garantir a entrega dos pacotes que transferirmos. Para aceitar mais de uma conexão o servidor possui *threads* e para cada conexão cria um *thread* diferente.

Dando início a construção do jogo, resolvemos dividi-lo em três partes, a fim de facilitar o entendimento das partes e a construção do todo:

- Servidor;
- Cliente jogador;
- Cliente observador.

### 1. Servidor.

O servidor é a parte principal do jogo, ele inicia o jogo, cria um cliente e fica aguardando conexão de mais um cliente para iniciar o jogo.

Durante a partida o servidor recebe as jogadas dos clientes e retira a quantidade de palitos indicada da fileira indicada e coordena os turnos de jogadas, além de continuar aguardando conexões de um ou mais clientes observadores, que podem entrar na partida a qualquer momento e obter atualizações do jogo.

A cada jogada o servidor verifica se a quantidade de palito nas três fileiras é maior que 1, enquanto isso, a partida continua.

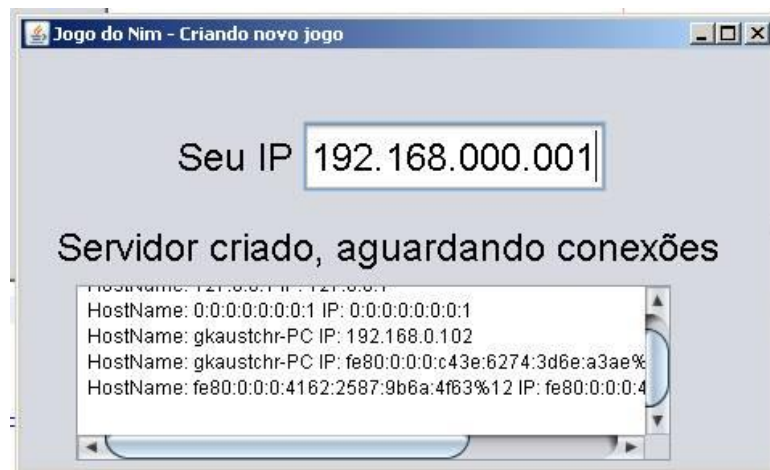


Figura 1 – Servidor criado, aguardando conexões.

## 2. Cliente ativo

O cliente se conecta no servidor e espera a partida ser iniciada, a cada jogada o servidor analisa as retiradas e atualiza a partida para os dois clientes ativos. Cada cliente ativo joga somente no seu turno e atualiza somente uma fileira.



Figura 2 – Cliente se conectando ao servidor.

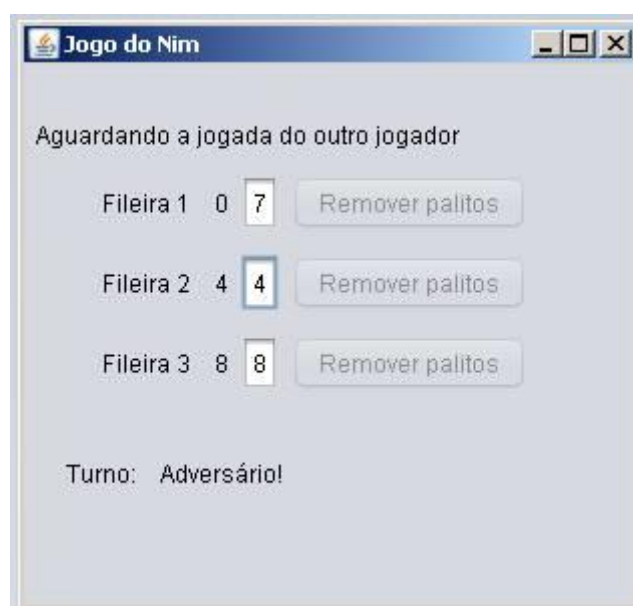


Figura 3 – Cliente bloqueado enquanto é o turno do adversário.



Figura 4 – Cliente desbloqueado para jogar na sua vez.

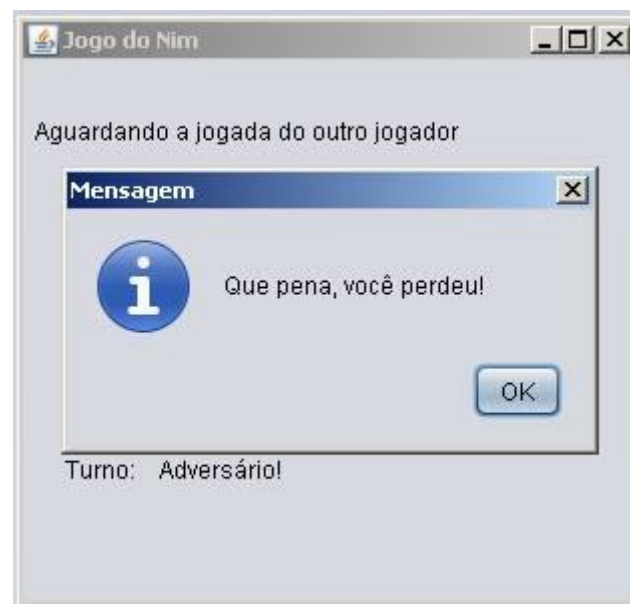
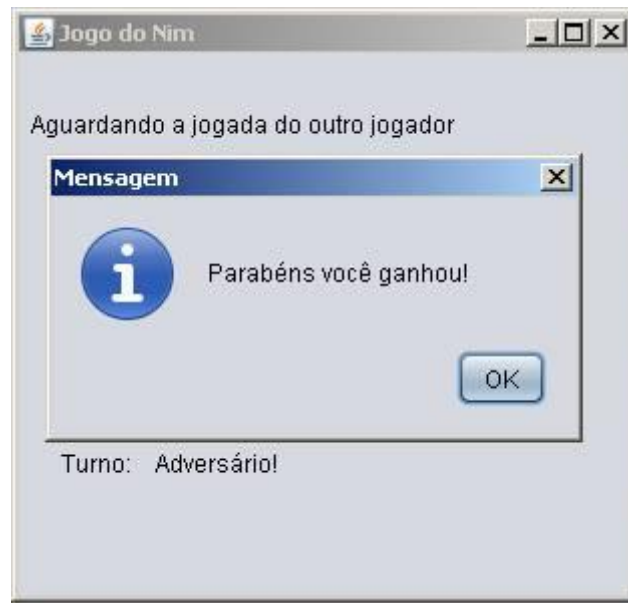


Figura 5 – Cliente perdedor.



**Figura 6 – Cliente vencedor.**

### **3. Cliente observador**

O cliente observador pode se conectar ao servidor a qualquer momento e passará a receber atualizações da partida.



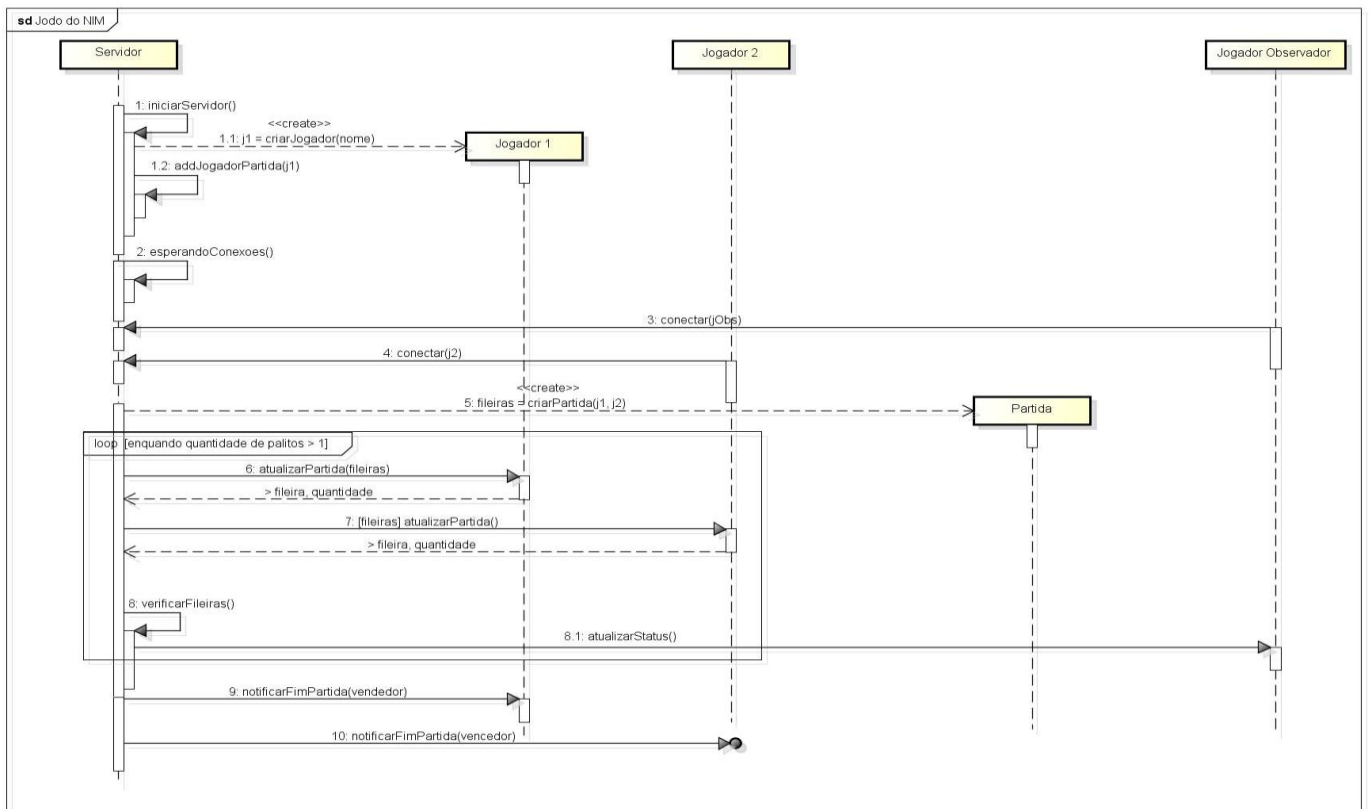
**Figura 7 – Cliente passivo.**



## FLUXO DO JOGO

1. O servidor é iniciado, cria um cliente e fica aguardando a conexão de mais um cliente;
2. Um cliente ativo se conecta ao servidor e esse inicia a partida com um número entre 2 e 10 de palitos em cada uma das três fileiras;
3. Enquanto a quantidade de palito nas três fileiras for maior que 1 o jogo continua, quando chegar a 1 é definido o vencedor da partida.

## DIAGRAMA DE SEQUÊNCIA



**CÓDIGO FONTE****1. Cliente**

```
import java.io.*;
import java.net.*;

public class Cliente {

    Socket conexao;

    BufferedReader entrada;

    DataOutputStream saida;

    boolean conectado = false;

    static boolean jogoIniciado = false;

    boolean meuTurno = false;

    String id = "0";

    String jogada = null;

    String tela = null;

    public Cliente(String host, int porta) throws
    UnknownHostException, IOException {

        conexao = new Socket(host, porta);

        entrada = new BufferedReader(new
        InputStreamReader(conexao.getInputStream())); //new
        BufferedReader(new InputStreamReader(System.in));

        saida = new
        DataOutputStream(conexao.getOutputStream());

        System.out.println("Cliente conectado!");

    }

    public void entrarNoJogo() throws IOException {

        String nome = "YOLO";

        while (true) {

            saida.writeBytes(nome + '\n');

            String in = entrada.readLine();

            if ( !in.equals("0") ) {
```

```

        conectado = true;
        id = in;
        break;
    } else {
        System.out.println("Cliente: entrarNoJogo:
Negado!");
    }
}

System.out.println("Cliente: Conectado ao server!");

}

    public void jogar() throws IOException,
    InterruptedException {

        while (true) {

            System.out.println("Cliente: testar se jogo
começou");

            saida.writeBytes(
Status.createStatus("JOGO_INICIADO").getValor() + '\n' );

            String in = entrada.readLine();

            System.out.println("Resposta SERVER: " + in);

            if
(in.equals(Status.createStatus("SIM").getValor())) {
                conectado = true;
                break;
            }

```

```
}

jogoIniciado = true;

while (true) {

    // Testar se é minha vez
    while (true) {

        saida.writeBytes(
Status.createStatus("MEU_TURNO").getValor() + '\n' );

        String in = entrada.readLine();

        if ( in.equals("NAO")) {

            Thread.sleep(3000);
            continue;

        } else {

            // Atualiza tela
            saida.writeBytes(
Status.createStatus("TELA").getValor() + '\n' );

            in = entrada.readLine();

            tela = in;

            meuTurno = true;
            break;
        }
    }
}
```

```

    }

    // Jogada
    while (true) {

        Thread.sleep(1000);

        if ( jogada == null ) {
            continue;
        }

        System.out.println("Cliente: Jogada feita: "
+ jogada + "\n" );

        saida.writeBytes(">" + jogada + "\n");
        jogada = null;
        meuTurno = false;

        String in = entrada.readLine();

        if ( in.equals("SIM") ) {
            meuTurno = false;
            break;
        }
    }

}

public void sairJogo () throws IOException {
    conexao.close();
}

```

```

public void jogada( String jogada ) {
    System.out.println("JOGADA: '" + jogada + "'");
    this.jogada = jogada;
}

public boolean meuTurno() throws IOException {
    saida.writeBytes(
Status.createStatus("MEU_TURNO").getValor() + '\n' );

    String in = entrada.readLine();

    return in.equals("SIM");
}

public String atualizarTela() throws IOException {
    saida.writeBytes(
Status.createStatus("TELA").getValor() + '\n' );

    String in = entrada.readLine();

    tela = in;

    return tela;
}

public int jogoAcabou() throws IOException {
    saida.writeBytes(
Status.createStatus("JOGO_ACABOU").getValor() + '\n' );

```

```

        String in = entrada.readLine();

        if ( in.equals("0") )
            return 0;

        if ( in.equals( String.valueOf(id) ) )
            return 1;
        else
            return 2;

    }

    public void fecharConexao() throws IOException {
        conexao.close();
    }
}

```

## 2. Servidor

```

import java.io.*;
import java.net.*;
import java.util.Random;

public class Servidor {

    ServerSocket socketServidor;

    int porta = 6789;

    Jogador jogador = null;

    static int numJogadores = 0;

    Jogador espectador;

    ServidorRequisicoes servidorRequisicoes;

    static String tela = "999"; // [0] - fileira 1 [1] - fileira
    2 [2] - fileira 3

```

```

int tipoJogador; // 1 = jog1, 2 = jog2, 3 = visitante

static int turno = 1;
static boolean jogoIniciado = false;
static int vitoria = 0;

public Servidor( int porta ) {
    this.porta = porta;

    jogador = new Jogador();
    espectador = null;
    numJogadores = 0;
    vitoria = 0;

    tela = "999";

    jogoIniciado = false;

    try {
        socketServidor = new ServerSocket(porta);
        System.out.println(" [ SERVIDOR CRIADO ]");
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

public void abrirConexao() throws IOException {

```



```

        // Cria palitos aleatoriamente

        Random r = new Random();

        tela = String.valueOf( r.nextInt(8) + 2 ) +
String.valueOf( r.nextInt(8) + 2 ) + String.valueOf(
r.nextInt(8) + 2 );

        while(true) {

            System.out.print("Servidor: Esperando alguem se
conectar... [" + String.valueOf(numJogadores) +"] \n");

            if ( numJogadores < 2 ) {

                jogador = new Jogador();

                jogador.socketConexao =
socketServidor.accept();

                servidorRequisicoes = new
ServidorRequisicoes(this, jogador);

                new Thread(servidorRequisicoes).start();

            } else {

                espectador.socketConexao =
socketServidor.accept();

                espectador = new Jogador();
                espectador.nome = "Espectador";

                System.out.println(" Conectou Espectador!");
            }
        }
    }
}

```

```

        tipoJogador = 3;

        // cria uma nova thread para tratar essa
conexão

        //new Thread(this).start();

        //System.out.println(" >> Conexão recusada:
Já existem dois jogadores conectados!");
    }

}

}

public void fecharConexao() throws IOException {
    numJogadores--;

    if ( numJogadores == 0 ) {
        socketServidor.close();
    }
}

}

```

### 3. Jogador

```

import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.net.Socket;

public class Jogador {
    String nome;
    Socket socketConexao;

```

```

        DataOutputStream entradaJogador;

        BufferedReader saidaJogador;

    }

```

#### 4. NovoJogo

```

import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;

// 127.0.0.1

public class NovoJogo extends Thread {
    Servidor servidor;

    public void criarServidor( ) {
        servidor = new Servidor(6789);
    }

    public void iniciar() throws IOException {
        servidor.abrirConexao();
    }

    public void run() {
        criarServidor();

        try {
            iniciar();
        } catch (IOException ex) {

            Logger.getLogger(NovoJogo.class.getName()).log(Level.SEVERE,
            null, ex);
        }
    }
}

```

```
    }
}
```

## 5. Jogo

```
import java.io.IOException;
import java.net.UnknownHostException;
import java.util.logging.Level;
import java.util.logging.Logger;

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author gkaustchr
 */

// 192.168.0.102
public class Jogo extends Thread{
    Cliente cliente;
    String ip_servidor;

    public Jogo( String ip) {
        this.ip_servidor = ip;
    }

    public void entrarNoJogo() throws IOException {
```

```

        cliente.entrarNoJogo();
    }

    public void criarCliente( ) throws UnknownHostException,
IOException {
        cliente = new Cliente(ip_servidor, 6789);
    }

    public void run() {
        try {
            criarCliente();
            System.out.println("Jogo: Cliente Criado");
        } catch (UnknownHostException ex) {
            Logger.getLogger(Jogo.class.getName()).log(Level.SEVERE, null,
ex);
        } catch (IOException ex) {
            Logger.getLogger(Jogo.class.getName()).log(Level.SEVERE, null,
ex);
        }

        try {
            System.out.println("Jogo: Cliente entrando no
jogo");
            entrarNoJogo();
        } catch (IOException ex) {
            Logger.getLogger(Jogo.class.getName()).log(Level.SEVERE, null,
ex);
        }

        while (true) {

```

```

        try {
            System.out.println("Jogo: Cliente jogar");
            cliente.jogar();
            // Durante o jogo
        } catch (IOException ex) {

Logger.getLogger(Jogo.class.getName()).log(Level.SEVERE, null,
ex);

            break;
        } catch (InterruptedException ex) {

Logger.getLogger(Jogo.class.getName()).log(Level.SEVERE, null,
ex);

        }
    }

}

public String atualizarTela() throws IOException {
    return cliente.atualizarTela();
}

public boolean meuTurno() throws IOException {
    return cliente.meuTurno();
}

public void jogada(String jogada) {
    cliente.jogada(jogada);
}

public int jogoAcabou() throws IOException {
    return cliente.jogoAcabou();
}

```

```

    }

    public void fecharConexao() throws IOException {
        cliente.fecharConexao();
    }

}

```

## 6. ServidorRequisicoes

```

import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.logging.Level;
import java.util.logging.Logger;

/*
 * To change this license header, choose License Headers in
 * Project Properties.
 *
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author aluno
 */
public class ServidorRequisicoes extends Thread {
    Jogador jogador;
    Servidor servidor;
    int indice;
}

```

```

static int cont = 0;

    public ServidorRequisicoes(Servidor servidor, Jogador
jogador) {

        this.servidor = servidor;

        this.jogador = jogador;

        cont++;

        indice = cont;
    }

    public void run() {

        try {

            jogador.saidaJogador = new BufferedReader(new
InputStreamReader(jogador.socketConexao.getInputStream()));

            jogador.entradaJogador = new
DataOutputStream(jogador.socketConexao.getOutputStream());

        } catch (IOException ex) {

            Logger.getLogger(ServidorRequisicoes.class.getName()).log(Level.
SEVERE, null, ex);

        }

        System.out.println("[ RUN SERVIDOR ]");

        try {

            // Espera nome do jogador

            String nome = jogador.saidaJogador.readLine();

```



```

jogador.nome = nome;

String msgSaida = String.valueOf( indice );

jogador.entradaJogador.writeBytes(msgSaida + '\n');

System.out.println("Servidor: Conectou Jogador
[NOME: " +jogador.nome+ "]);

if ( indice == 2 )
    servidor.jogoIniciado = true;

while ( true ) {
    //System.out.println("Servidor: Aguardando
Requisição");

    String msg = jogador.saidaJogador.readLine();

    if (
msg.equals(Status.createStatus("JOGO_INICIADO").getValor()) ) {

        //System.out.println("Thread["+ indice +"]:
Requisição de jogo iniciado");

        if ( servidor.jogoIniciado == true ) {

            msgSaida =
Status.createStatus("SIM").getValor();

        } else {

            msgSaida =
Status.createStatus("NAO").getValor();

        }

        //System.out.println("Thread["+ indice +"]:
Requisição respondida!");

```

```

jogador.entradaJogador.writeBytes(msgSaida +
'\n');

    } else if (
msg.equals(Status.createStatus("MEU_TURNO").getValor()) ) {

        //System.out.println("Thread["+ indice +"]:
Requisição meu turno");

        if ( servidor.turno == indice ) {

            msgSaida =
Status.createStatus("SIM").getValor();

        } else {

            msgSaida =
Status.createStatus("NAO").getValor();

        }

        //System.out.println("Thread["+ indice +"]:
Requisição respondida!");

        jogador.entradaJogador.writeBytes(msgSaida +
'\n');

    } else if (
msg.equals(Status.createStatus("TELA").getValor()) ) {

        //System.out.println("Thread["+ indice +"]:
Requisição Tela");

        msgSaida = servidor.tela;

        //System.out.println("Thread["+ indice +"]:
Requisição respondida!");

        jogador.entradaJogador.writeBytes(msgSaida +
'\n');

```

```

// Veio uma jogada ( ">FN" ) F - fileira | N
- retirados

} else if ( msg.charAt(0) == '>' ) {

    //System.out.println("Thread["+ indice +"]:
Requisição de jogada");

    // Altera a tela

    String temp = servidor.tela;

    servidor.tela = "";

    for (int i = 0; i < temp.length(); i++) {

        if ( Integer.parseInt( String.valueOf(
msg.charAt(1)) )-1 == i ) {

            servidor.tela += String.valueOf(
Integer.parseInt( String.valueOf(temp.charAt(i)) ) -
Integer.parseInt( String.valueOf( msg.charAt(2) ) ) );

        } else {

            servidor.tela += temp.charAt(i);

        }

    }

    // se teve mudança na jogada

    if ( ! temp.equals(servidor.tela) ) {

        // Testa se o Jogo Acabou

        if ( servidor.tela.equals("000") ) {

            if (indice == 1)

                servidor.vitoria = 2;

            else

                servidor.vitoria = 1;

        }

    }

}

```

```

        System.out.println(">>>>>>>   TELA NOVA: '"+
servidor.tela +"'");

```

```

        if ( servidor.turno == 1 )
            servidor.turno = 2;
        else
            servidor.turno = 1;

```

```

        msgSaida =
Status.createStatus("SIM").getValor();

```

```

        //System.out.println("Thread["+ indice +"]:
Requisição respondida!");
        jogador.entradaJogador.writeBytes(msgSaida +
'\n');

```

```

        } else if (
msg.equals(Status.createStatus("JOGO_ACABOU").getValor()) ) {

```

```

        //System.out.println("Thread["+ indice +"]:
Requisição Tela");

```

```

        msgSaida = String.valueOf( servidor.vitoria
);

```

```

        //System.out.println("Thread["+ indice +"]:
Requisição respondida!");
        jogador.entradaJogador.writeBytes(msgSaida +
'\n');

```

```

        // Veio uma jogada ( ">FN" ) F - fileira | N
- retirados

```

```

        } if (
msg.equals(Status.createStatus("FIM").getValor()) ) {

```

```

        msgSaida =
Status.createStatus("SIM").getValor();

        jogador.entradaJogador.writeBytes(msgSaida +
'\n');

        try {
            Thread.sleep(1500);
        } catch (InterruptedException ex) {

Logger.getLogger(ServidorRequisicoes.class.getName()).log(Level.
SEVERE, null, ex);

        }

        jogador.socketConexao.close();

        servidor.fecharConecao();

    }else {
        System.out.println("Thread["+ indice +"]:
Requisição X! <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<");
    }

}

} catch (IOException ex) {

Logger.getLogger(ServidorRequisicoes.class.getName()).log(Level.
SEVERE, null, ex);

}

}

```

```
}
```

## 7. Status

```
/**
 *
 * @author aluno
 */
public class Status {

    public static final Status SIM = new Status("SIM");
    public static final Status NAO = new Status("NAO");
    public static final Status CONECTAR = new
Status("CONECTAR");
    public static final Status JOGO_INICIADO = new
Status("JOGO_INICIADO");
    public static final Status MEU_TURNNO = new
Status("MEU_TURNNO");
    public static final Status TELA = new Status("TELA");
    public static final Status JOGO_ACABOU = new
Status("JOGO_ACABOU");
    public static final Status FIM = new Status("FIM");

    private String tipo;

    private Status(String tipo) {
        this.tipo = tipo;
    }

    public static Status createStatus(String tipo) {
        if("SIM".equals(tipo))
            return Status.SIM;
        else if("NAO".equals(tipo))
```

```

        return Status.NAO;
    else if("CONECTAR".equals(tipo))
        return Status.CONECTAR;
    else if("JOGO_INICIADO".equals(tipo))
        return Status.JOGO_INICIADO;
    else if("MEU_TURNO".equals(tipo))
        return Status.MEU_TURNO;
    else if("TELA".equals(tipo))
        return Status.TELA;
    else if("JOGO_ACABOU".equals(tipo))
        return Status.JOGO_ACABOU;
    else if("FIM".equals(tipo))
        return Status.FIM;

    return null;
}

public boolean check(String tipo) {
    if(this.tipo.equals(tipo))
        return true;

    return false;
}

public String getValor() {
    return tipo;
}

}

```

## 8. Reminder

```
/**
```

```

*
* @author gkaustchr
*/
import java.util.Timer;
import java.util.TimerTask;

/**
 * Simple demo that uses java.util.Timer to schedule a task
 * to execute once 5 seconds have passed.
 */

public class Reminder {
    Timer timer;

    public Reminder(int seconds) {
        timer = new Timer();
        timer.schedule(new RemindTask(), seconds*1000);
    }

    class RemindTask extends TimerTask {
        public void run() {

            System.out.println("Time's up!");
            timer.cancel(); //Terminate the timer thread

        }
    }

    public void cancelTimer() {
        timer.cancel();
    }
}

```



```

    }

    public static void main(String args[]) {
        new Reminder(5);
        System.out.println("Task scheduled.");
    }
}

```

## 9. frmCriarJogo

```

import java.net.InetAddress;
import java.net.NetworkInterface;
import java.net.SocketException;
import java.util.Enumeration;
import java.util.Timer;
import java.util.TimerTask;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;

/*
 * frmCriarJogo.java
 *
 * Created on 22/07/2014, 10:04:57
 */
/**
 *
 * @author gkaustchr
 */
public class frmCriarJogo extends javax.swing.JFrame {
    Timer timer;
    NovoJogo novoJogo;
}

```

```

Jogo jogo;

JFrame formJogo;

/** Creates new form frmCriarJogo */
public frmCriarJogo() {
    initComponents();
}

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this
method is
 * always regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN: initComponents
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    txtIp = new javax.swing.JTextField();
    jLabel2 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    txaEndereços = new javax.swing.JTextArea();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_
CLOSE);

    setTitle("Jogo do Nim - Criando novo jogo");
    addWindowListener(new java.awt.event.WindowAdapter() {
        public void windowOpened(java.awt.event.WindowEvent
evt) {

            formWindowOpened(evt);

```

```

        }

    });

    jLabel1.setFont(new java.awt.Font("Dialog", 0, 24)); //
NOI18N
    jLabel1.setText("Seu IP");

    txtIp.setFont(new java.awt.Font("Dialog", 0, 24)); //
NOI18N
    txtIp.setText("192.168.000.001");

    jLabel2.setFont(new java.awt.Font("Dialog", 0, 24)); //
NOI18N
    jLabel2.setText("Servidor criado, aguardando conexões");

    txtaEndereços.setColumns(20);
    txtaEndereços.setRows(5);
    jScrollPane1.setViewportView(txtaEndereços);

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

.addContainerGap(97, Short.MAX_VALUE)

.addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(txtIp,
javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(104, 104, 104))

        .addGroup(layout.createSequentialGroup())

        .addGap(22, 22, 22)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING)

        .addGroup(layout.createSequentialGroup())

        .addGap(10, 10, 10)

        .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 374,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addComponent(jLabel2))

        .addContainerGap(26, Short.MAX_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)

        .addGroup(layout.createSequentialGroup())

        .addGap(44, 44, 44)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.BASELINE)

        .addComponent(jLabel1)

        .addComponent(txtIp,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(18, 18, 18)

        .addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED)

        .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addContainerGap(23, Short.MAX_VALUE))

    );

    pack();

} // </editor-fold> // GEN-END: initComponents

private void formWindowOpened(java.awt.event.WindowEvent evt)
{ // GEN-FIRST: event_formWindowOpened

    /*
    HostName: 127.0.0.1 IP: 127.0.0.1
    HostName: 0:0:0:0:0:0:0:1 IP: 0:0:0:0:0:0:0:1
    HostName: gkaustchr-PC IP: 192.168.0.102
    HostName: gkaustchr-PC IP: fe80:0:0:0:c43e:6274:3d6e:a3ae%11
    HostName: fe80:0:0:0:4162:2587:9b6a:4f63%12 IP:
    fe80:0:0:0:4162:2587:9b6a:4f63%12
    */

    Enumeration e = null;

    try {
        e = NetworkInterface.getNetworkInterfaces();
    } catch (SocketException ex) {

        Logger.getLogger(frmCriarJogo.class.getName()).log(Level.SEVERE,
            null, ex);
    }

    while (e.hasMoreElements()) {
        NetworkInterface i = (NetworkInterface)
            e.nextElement();

        Enumeration ds = i.getInetAddresses();

        while (ds.hasMoreElements()) {
            InetAddress myself = (InetAddress) ds.nextElement();

            System.out.println("HostName: " +
                myself.getHostName() + " IP: " + myself.getHostAddress());
        }
    }
}

```

```

        txaEndereços.setText( txaEndereços.getText() +
"HostName: " + myself.getHostName() + " IP: " +
myself.getHostAddress()+ "\n");

    }

}

    novoJogo = new NovoJogo();
    novoJogo.start();

    //Thread t_novoJogo = novoJogo;
    //t_novoJogo.start();

    try {
        Thread.currentThread().sleep(2000);
    } catch (InterruptedException ex) {

Logger.getLogger(frmCriarJogo.class.getName()).log(Level.SEVERE,
null, ex);

    }

    jogo = new Jogo("127.0.0.1");
    jogo.start();
    //Thread t_jogo = jogo;
    //t_jogo.start();

    reminder(3);

} //GEN-LAST:event_formWindowOpened

public void fechar( ) {
    this.dispose();
}

```

```

}

public void reminder(int seconds) {
    timer = new Timer();
    timer.schedule(new RemindTask(), seconds*1000);
}

class RemindTask extends TimerTask {
    public void run() {

        //System.out.println("Time's up!");
        //timer.cancel(); //Terminate the timer thread

        System.out.println("<TIMER>");
        /*
        if ( novoJogo.servidor.jogoIniciado == true ) {
            System.out.println("<TIMER> Ok :)");

            new frmJogo(jogo).setVisible(true);

            timer.cancel();
            // this.dispose();
        }
        */
        if ( jogo.cliente.jogoIniciado == true ) {
            formJogo = new frmJogo(jogo);
            formJogo.setVisible(true);

            timer.cancel();

            while (true) {

```

```

        if ( !formJogo.isVisible() )
            fechar();

        try {
            Thread.sleep(1000);
        } catch (InterruptedException ex) {

Logger.getLogger(frmCriarJogo.class.getName()).log(Level.SEVERE,
null, ex);

        }
    }

    // this.dispose();
} else {
    reminder(3);
}

}

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and
feel setting code (optional) ">

    /* If Nimbus (introduced in Java SE 6) is not available,
stay with the default look and feel.

    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/p
laf.html
    */

    try {

```



```

        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(frmCriarJogo.class.getName())
.log(java.util.logging.Level.SEVERE, null, ex);

        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(frmCriarJogo.class.getName())
.log(java.util.logging.Level.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(frmCriarJogo.class.getName())
.log(java.util.logging.Level.SEVERE, null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex)
{

java.util.logging.Logger.getLogger(frmCriarJogo.class.getName())
.log(java.util.logging.Level.SEVERE, null, ex);

        }
    }
}

//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {
        new frmCriarJogo().setVisible(true);
    }
});
}

```

```

        // Variables declaration - do not modify//GEN-BEGIN:variables
        private javax.swing.JLabel jLabel1;
        private javax.swing.JLabel jLabel2;
        private javax.swing.JScrollPane jScrollPane1;
        private javax.swing.JTextArea txtaEndereços;
        private javax.swing.JTextField txtIp;
        // End of variables declaration//GEN-END:variables
    }

```

## 10. frmEntrarJogo

```

import java.util.Timer;
import java.util.TimerTask;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;

/*
 * To change this license header, choose License Headers in
 * Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author aluno
 */
public class frmEntrarJogo extends javax.swing.JFrame {

    Timer timer;

    Jogo jogo;

    JFrame formJogo;

```

```

/**
 * Creates new form frmEntrarJogo
 */
public frmEntrarJogo() {
    initComponents();
}

/**
 * This method is called from within the constructor to
 initialize the form.
 * WARNING: Do NOT modify this code. The content of this
 method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN: initComponents
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    txtIp = new javax.swing.JTextField();
    jButton1 = new javax.swing.JButton();
    lblStatus = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_
CLOSE);

    setTitle("Jogo do Nim - Entrar em um jogo");
    addWindowListener(new java.awt.event.WindowAdapter() {
        public void windowOpened(java.awt.event.WindowEvent
evt) {
            formWindowOpened(evt);
        }
    }

```

```

    });

    jLabel1.setFont(new java.awt.Font("Ubuntu", 0, 24)); //
NOI18N
    jLabel1.setText("IP do Servidor");

    txtIp.setFont(new java.awt.Font("Ubuntu", 0, 24)); //
NOI18N
    txtIp.setText("192.168.26.129");

    jButton1.setFont(new java.awt.Font("Ubuntu", 0, 24)); //
NOI18N
    jButton1.setText("Conectar");
    jButton1.addActionListener(new
java.awt.event.ActionListener() {
        public void
actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });

    lblStatus.setText("Status");

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)

        .addGroup(layout.createSequentialGroup()

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING)

                .addGroup(layout.createSequentialGroup()

                    .addGap(148, 148, 148)

```

```

        .addComponent(jButton1))

        .addGroup(layout.createSequentialGroup())

        .addGap(27, 27, 27)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING)

        .addComponent(lblStatus)

        .addGroup(layout.createSequentialGroup())

        .addComponent(jLabel1)

        .addGap(2, 2, 2)

        .addComponent(txtIp,
javax.swing.GroupLayout.PREFERRED_SIZE, 204,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addContainerGap(36, Short.MAX_VALUE))

    );

    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)

        .addGroup(layout.createSequentialGroup())

        .addGap(35, 35, 35)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.BASELINE)

        .addComponent(jLabel1)

        .addComponent(txtIp,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED)

        .addComponent(jButton1)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED)

        .addComponent(lblStatus)

```

```

        .addContainerGap(152, Short.MAX_VALUE))

    );

    pack();

} // </editor-fold> // GEN-END: initComponents

    private void formWindowOpened(java.awt.event.WindowEvent
evt) { // GEN-FIRST: event_formWindowOpened

        // Windows Opened

    } // GEN-LAST: event_formWindowOpened

    private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST: event_jButton1ActionPerformed

        // Botão Conectar

        lblStatus.setText("Status: Conectando ao Servidor ...");

        jogo = new Jogo( txtIp.getText() );
        jogo.start();

        // Thread t_jogo = jogo;
        // t_jogo.start();

        reminder(3);

    } // GEN-LAST: event_jButton1ActionPerformed

    public void fechar( ) {

        this.dispose();

    }

    public void reminder(int seconds) {

        timer = new Timer();

```

```

        timer.schedule(new RemindTask(), seconds*1000);
    }

    class RemindTask extends TimerTask {
        public void run() {
            //System.out.println("Time's up!");
            //timer.cancel(); //Terminate the timer thread

            System.out.println(">> " + jogo.cliente.conectado);
            System.out.println(">> " +
jogo.cliente.jogoIniciado);

            if ( jogo.cliente.conectado == true ) {
                lblStatus.setText("Status: Conectado ao
Servidor!");
            }

            if ( jogo.cliente.jogoIniciado == true ) {
                formJogo = new frmJogo(jogo);
                formJogo.setVisible(true);

                timer.cancel();

                while (true) {
                    if ( !formJogo.isVisible() )
                        fechar();

                    try {
                        Thread.sleep(1000);
                    } catch (InterruptedException ex) {

Logger.getLogger(frmCriarJogo.class.getName()).log(Level.SEVERE,
null, ex);

```

```

        }
    }

    // this.dispose();
} else {
    reminder(3);
}

}

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and
    feel setting code (optional) ">

    /* If Nimbus (introduced in Java SE 6) is not available,
    stay with the default look and feel.

    * For details see
    http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/p
    laf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
        javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {

                javax.swing.UIManager.setLookAndFeel(info.getClassName());

                break;
            }
        }
    } catch (ClassNotFoundException ex) {

```



```

java.util.logging.Logger.getLogger(frmEntrarJogo.class.getName())
).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(frmEntrarJogo.class.getName())
).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(frmEntrarJogo.class.getName())
).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex)
{

java.util.logging.Logger.getLogger(frmEntrarJogo.class.getName())
).log(java.util.logging.Level.SEVERE, null, ex);

        }

//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new frmEntrarJogo().setVisible(true);
    }
});
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel lblStatus;
private javax.swing.JTextField txtIp;
// End of variables declaration//GEN-END:variables
}

```

## 11. frmJogo

```

import java.io.IOException;
import java.util.Timer;
import java.util.TimerTask;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

/*
 * To change this license header, choose License Headers in
 * Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author aluno
 */
public class frmJogo extends javax.swing.JFrame {

    Jogo jogo;

    Timer timer;

    int vencedor;

    /**
     * Creates new form frmJogo
     */
    public frmJogo( ) {
        initComponents();
    }
}

```

```
public frmJogo( Jogo jogo ) {  
    initComponents();  
  
    this.jogo = jogo;  
  
    vencedor = 0;  
  
    iniciarJogo();  
  
}  
  
public void iniciarJogo() {  
  
    reminder(3);  
  
    lblMensagens.setText("Jogo iniciado");  
    desabilitarBotoes();  
  
}  
  
public void desabilitarBotoes() {  
    btnRemover1.setEnabled(false);  
    btnRemover2.setEnabled(false);  
    btnRemover3.setEnabled(false);  
}  
  
public void habilitarBotoes() {  
  
    txtFileira1.setText( lblQtdFileira1.getText() );  
    txtFileira2.setText( lblQtdFileira2.getText() );  
}
```

```

txtFileira3.setText( lblQtdFileira3.getText() );

btnRemover1.setEnabled(true);
btnRemover2.setEnabled(true);
btnRemover3.setEnabled(true);

if ( lblQtdFileira1.getText().equals("0") )
    btnRemover1.setEnabled(false);
if ( lblQtdFileira2.getText().equals("0") )
    btnRemover2.setEnabled(false);
if ( lblQtdFileira3.getText().equals("0") )
    btnRemover3.setEnabled(false);
}

public void fechar() {
    this.dispose();
}

public void reminder(int seconds) {
    timer = new Timer();
    timer.schedule(new RemindTask(), seconds*1000);
}

class RemindTask extends TimerTask {
    public void run() {

        //System.out.println("Time's up!");
        //timer.cancel(); //Terminate the timer thread

        System.out.println("JOGO: Testando se é meu turno");
    }
}

```

```

        atualizarTela();

    try {
        if ( jogo.meuTurno() == true ) {

            atualizarTela();

            vencedor = jogoAcabou();
            if ( vencedor != 0 ) {
                timer.cancel();
                // Eu venci
                if ( vencedor == 1 ) {
                    JOptionPane.showMessageDialog(null,
"Parabéns você ganhou!");

                    // Passar o turno
                    jogo.jogada("10");

                    // Eu perdi
                } else {
                    JOptionPane.showMessageDialog(null,
"Que pena, você perdeu!");
                }

                try {
                    Thread.sleep(1500);
                } catch (InterruptedException ex) {

                    Logger.getLogger(frmJogo.class.getName()).log(Level.SEVERE,
null, ex);
                }
            }
        }
    }

```

```

        jogo.fecharConexao();

        fechar();
    }

    habilitarBotoes();
    lblMensagens.setText("Faça sua jogada");
    lblTurno.setText("Sua Vez!");

    timer.cancel();

    } else {

        desabilitarBotoes();
        lblMensagens.setText("Aguardando a jogada do
outro jogador");
        lblTurno.setText("Adversário!");

        reminder(3);

    }
    } catch (IOException ex) {

        Logger.getLogger(frmJogo.class.getName()).log(Level.SEVERE,
        null, ex);

    }

}

}

```

```

public void atualizarTela() {
    String atualizacao = "999";

    try {
        atualizacao = jogo.atualizarTela();
    } catch (IOException ex) {

Logger.getLogger(frmJogo.class.getName()).log(Level.SEVERE,
null, ex);

    }

    System.out.println("TELA ATUALIZADA: '"+ atualizacao
+ "'");

    lblQtdFileira1.setText( String.valueOf(
atualizacao.charAt(0) ) );

    lblQtdFileira2.setText( String.valueOf(
atualizacao.charAt(1) ) );

    lblQtdFileira3.setText( String.valueOf(
atualizacao.charAt(2) ) );

}

public void cancelTimer() {
    timer.cancel();
}

public int jogoAcabou() throws IOException {
    return jogo.jogoAcabou();
}

/**

```

```

    * This method is called from within the constructor to
    initialize the form.

    * WARNING: Do NOT modify this code. The content of this
    method is always

    * regenerated by the Form Editor.

    */

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
    Code">//GEN-BEGIN: initComponents

    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        lblQtdFileira1 = new javax.swing.JLabel();
        lblQtdFileira2 = new javax.swing.JLabel();
        lblQtdFileira3 = new javax.swing.JLabel();
        btnRemover1 = new javax.swing.JButton();
        btnRemover2 = new javax.swing.JButton();
        btnRemover3 = new javax.swing.JButton();
        jLabel4 = new javax.swing.JLabel();
        lblTurno = new javax.swing.JLabel();
        lblMensagens = new javax.swing.JLabel();
        txtFileira1 = new javax.swing.JTextField();
        txtFileira2 = new javax.swing.JTextField();
        txtFileira3 = new javax.swing.JTextField();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_
        CLOSE);

        setTitle("Jogo do Nim");

        jLabel1.setText("Fileira 1");

```



```

jLabel2.setText("Fileira 2");

jLabel3.setText("Fileira 3");

lblQtdFileira1.setText("9");

lblQtdFileira2.setText("9");

lblQtdFileira3.setText("9");

btnRemover1.setText("Remover palitos");

btnRemover1.addActionListener(new
java.awt.event.ActionListener() {

    public void
actionPerformed(java.awt.event.ActionEvent evt) {

        btnRemover1ActionPerformed(evt);

    }

});

btnRemover2.setText("Remover palitos");

btnRemover2.addActionListener(new
java.awt.event.ActionListener() {

    public void
actionPerformed(java.awt.event.ActionEvent evt) {

        btnRemover2ActionPerformed(evt);

    }

});

btnRemover3.setText("Remover palitos");

btnRemover3.addActionListener(new
java.awt.event.ActionListener() {

    public void
actionPerformed(java.awt.event.ActionEvent evt) {

```



```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING, false)
```

```
.addGroup(layout.createSequentialGroup())
```

```
.addComponent(jLabel3)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED)
```

```
.addComponent(lblQtdFileira3)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
.addComponent(txtFileira3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
```

```
.addComponent(jLabel2)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED)
```

```
.addComponent(lblQtdFileira2)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
.addComponent(txtFileira2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
```

```
.addComponent(jLabel1)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED)
```

```
.addComponent(lblQtdFileira1)
```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(txtFileiral,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING)

        .addComponent(btnRemover3)

        .addComponent(btnRemover2)

        .addComponent(btnRemover1)))

.addGroup(layout.createSequentialGroup()

        .addGap(21, 21, 21)

        .addComponent(jLabel4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED)

        .addComponent(lblTurno))

.addGroup(layout.createSequentialGroup()

        .addContainerGap()

        .addComponent(lblMensagens)))

.addContainerGap(54, Short.MAX_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)

        .addGroup(layout.createSequentialGroup()

                .addGap(27, 27, 27)

                .addComponent(lblMensagens)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED)

```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.BASELINE)
```

```
    .addComponent(jLabel1)

    .addComponent(lblQtdFileira1)

    .addComponent(btnRemover1)

    .addComponent(txtFileira1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.BASELINE)
```

```
    .addComponent(jLabel2)

    .addComponent(lblQtdFileira2)

    .addComponent(btnRemover2)

    .addComponent(txtFileira2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.BASELINE)
```

```
    .addComponent(btnRemover3)

    .addComponent(jLabel3)

    .addComponent(lblQtdFileira3)

    .addComponent(txtFileira3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED, 33, Short.MAX_VALUE)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jLabel14)
```

```
    .addComponent(lblTurno))
```

```
    .addGap(58, 58, 58))
```

```
);
```

```
pack();
```

```
}// </editor-fold>//GEN-END:initComponents
```

```
private void
```

```
btnRemover1ActionPerformed(java.awt.event.ActionEvent evt)
```

```
{//GEN-FIRST:event_btnRemover1ActionPerformed
```

```
    // Remover Fileira 1
```

```
    if ( Integer.parseInt( txtFileira1.getText() ) >
Integer.parseInt( lblQtdFileira1.getText() ) ) {
```

```
        JOptionPane.showMessageDialog(this, "O valor
retirado não pode exceder o valor da fileira");
```

```
        return;
```

```
    }
```

```
    if ( Integer.parseInt( txtFileira1.getText() ) < 1 ) {
```

```
        JOptionPane.showMessageDialog(this, "Deve-se remover
pelo menos um elemento da fileira");
```

```
        return;
```

```
    }
```

```
desabilitarBotoes();
```

```
jogo.jogada("1" + txtFileira1.getText());
```

```
reminder(3);
```

```
}//GEN-LAST:event_btnRemover1ActionPerformed
```

```

        private void
        btnRemover2ActionPerformed(java.awt.event.ActionEvent evt)
        { //GEN-FIRST:event_btnRemover2ActionPerformed

            // Remover Fileira 2

            if ( Integer.parseInt( txtFileira2.getText() ) >
                Integer.parseInt( lblQtdFileira2.getText() ) ) {

                JOptionPane.showMessageDialog(this, "O valor
                retirado não pode exceder o valor da fileira");

                return;

            }

            if ( Integer.parseInt( txtFileira2.getText() ) < 1 ) {

                JOptionPane.showMessageDialog(this, "Deve-se remover
                pelo menos um elemento da fileira");

                return;

            }

            desabilitarBotoes();

            jogo.jogada("2" + txtFileira2.getText());

            reminder(3);

        } //GEN-LAST:event_btnRemover2ActionPerformed

        private void
        btnRemover3ActionPerformed(java.awt.event.ActionEvent evt)
        { //GEN-FIRST:event_btnRemover3ActionPerformed

            // Remover Fileira 3

            if ( Integer.parseInt( txtFileira3.getText() ) >
                Integer.parseInt( lblQtdFileira3.getText() ) ) {

                JOptionPane.showMessageDialog(this, "O valor
                retirado não pode exceder o valor da fileira");

                return;

            }

```

```

        if ( Integer.parseInt( txtFileira3.getText() ) < 1 ) {
            JOptionPane.showMessageDialog(this, "Deve-se remover
pelo menos um elemento da fileira");

            return;
        }

        desabilitarBotoes();

        jogo.jogada("3" + txtFileira3.getText());

        reminder(3);

    } //GEN-LAST:event_btnRemover3ActionPerformed

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and
feel setting code (optional) ">

        /* If Nimbus (introduced in Java SE 6) is not available,
        stay with the default look and feel.

        * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/p
laf.html
        */

        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
                javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {

                    javax.swing.UIManager.setLookAndFeel(info.getClassName());

                    break;
                }
            }
        }
    }

```



```

        }

    }

    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(frmJogo.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);

        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(frmJogo.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(frmJogo.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex)
{

java.util.logging.Logger.getLogger(frmJogo.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);

        }

//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        new frmJogo().setVisible(true);

    }

});

}

// Variables declaration - do not modify//GEN-
BEGIN:variables

private javax.swing.JButton btnRemover1;
private javax.swing.JButton btnRemover2;
private javax.swing.JButton btnRemover3;
private javax.swing.JLabel jLabel1;

```

```

private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel lblMensagens;
private javax.swing.JLabel lblQtdFileira1;
private javax.swing.JLabel lblQtdFileira2;
private javax.swing.JLabel lblQtdFileira3;
private javax.swing.JLabel lblTurno;
private javax.swing.JTextField txtFileira1;
private javax.swing.JTextField txtFileira2;
private javax.swing.JTextField txtFileira3;
// End of variables declaration//GEN-END:variables
}

```

## 12. frmPrincipal

```

/*
 * frmPrincipal.java
 *
 * Created on 22/07/2014, 10:01:24
 */
/**
 *
 * @author gkaustchr
 */
public class frmPrincipal extends javax.swing.JFrame {

    /** Creates new form frmPrincipal */
    public frmPrincipal() {
        initComponents();
    }

    /** This method is called from within the constructor to

```

```

    * initialize the form.

    * WARNING: Do NOT modify this code. The content of this
method is

    * always regenerated by the Form Editor.

    */

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN: initComponents
    private void initComponents() {

        jMenuBar1 = new javax.swing.JMenuBar();
        jMenu1 = new javax.swing.JMenu();
        jMenuItem1 = new javax.swing.JMenuItem();
        jMenu2 = new javax.swing.JMenu();
        jMenuItem2 = new javax.swing.JMenuItem();
        jMenuItem3 = new javax.swing.JMenuItem();
        jMenu3 = new javax.swing.JMenu();
        jMenuItem4 = new javax.swing.JMenuItem();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLO
SE);

        setTitle("Jogo do Nim");

        jMenu1.setText("Menu");

        jMenuItem1.setText("Sair");
        jMenuItem1.addActionListener(new
java.awt.event.ActionListener() {

            public void
actionPerformed(java.awt.event.ActionEvent evt) {

                jMenuItem1ActionPerformed(evt);

            }

        });
    }

```

```

jMenu1.add(jMenuItem1);

jMenuBar1.add(jMenu1);

jMenu2.setText("Jogo");

jMenuItem2.setText("Criar Jogo");
jMenuItem2.addActionListener(new
java.awt.event.ActionListener() {

    public void
actionPerformed(java.awt.event.ActionEvent evt) {

        jMenuItem2ActionPerformed(evt);

    }

});
jMenu2.add(jMenuItem2);

jMenuItem3.setText("Entrar em um Jogo");
jMenuItem3.addActionListener(new
java.awt.event.ActionListener() {

    public void
actionPerformed(java.awt.event.ActionEvent evt) {

        jMenuItem3ActionPerformed(evt);

    }

});
jMenu2.add(jMenuItem3);

jMenuBar1.add(jMenu2);

jMenu3.setText("Ajuda");

jMenuItem4.setText("Como Jogar");
jMenu3.add(jMenuItem4);

```

```

jMenuBar1.add(jMenu3);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 400, Short.MAX_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 279, Short.MAX_VALUE)

);

pack();

} // </editor-fold> // GEN-END: initComponents

private void
 jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST: event_jMenuItem1ActionPerformed

// Menu Sair

System.exit(0);

} // GEN-LAST: event_jMenuItem1ActionPerformed

private void
 jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST: event_jMenuItem2ActionPerformed

// Menu Novo Jogo

new frmCriarJogo().setVisible(true);

```

```

} //GEN-LAST:event_jMenuItem2ActionPerformed

    private void
jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jMenuItem3ActionPerformed

    // Menu Entrar em um jogo

    new frmEntrarJogo().setVisible(true);

} //GEN-LAST:event_jMenuItem3ActionPerformed

/**
 * @param args the command line arguments
 */

public static void main(String args[]) {

    /* Set the Nimbus look and feel */

    //<editor-fold defaultstate="collapsed" desc=" Look and
feel setting code (optional) ">

    /* If Nimbus (introduced in Java SE 6) is not available,
stay with the default look and feel.

    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/p
laf.html
    */

    try {

        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {

            if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

                break;

            }

        }

    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(frmPrincipal.class.getName())
.log(java.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

```

```

java.util.logging.Logger.getLogger(frmPrincipal.class.getName())
.log(java.util.logging.Level.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(frmPrincipal.class.getName())
.log(java.util.logging.Level.SEVERE, null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex)
{

java.util.logging.Logger.getLogger(frmPrincipal.class.getName())
.log(java.util.logging.Level.SEVERE, null, ex);

        }

//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {
        new frmPrincipal().setVisible(true);
    }

});

}

// Variables declaration - do not modify//GEN-
BEGIN:variables

private javax.swing.JMenu jMenuItem1;
private javax.swing.JMenu jMenuItem2;
private javax.swing.JMenu jMenuItem3;
private javax.swing.JMenuBar jMenuItemBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JMenuItem jMenuItem3;
private javax.swing.JMenuItem jMenuItem4;

// End of variables declaration//GEN-END:variables

}

```

