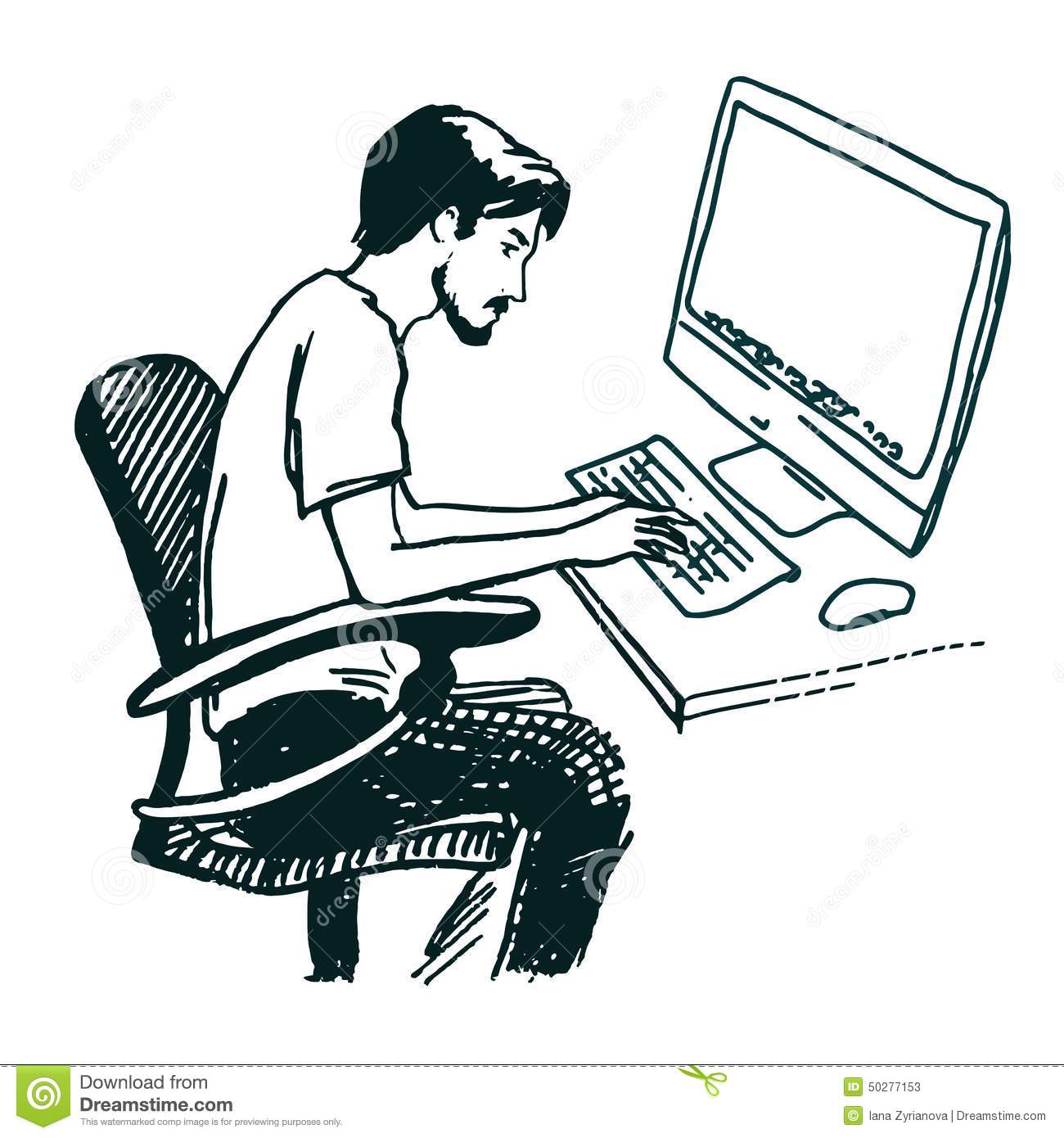
**MANUAL DE**

****

**PROGRAMADOR**

**“Barbero Dormilón”**

**Módulo “Barbero.java”**

**package peluquero;**

**import java.util.List;**

**import java.util.Random;**

**import java.util.logging.Level;**

**import java.util.logging.Logger;**

**import javax.swing.JLabel;**

**import javax.swing.JProgressBar;**

**public class Barbero implements Runnable {**

**boolean duerme;**

**Random r;**

**Monitor m;**

**JLabel stateBarBer;**

**JLabel chairStatus;**

**JProgressBar progress;**

**public Barbero(Monitor m, JLabel stateBarber, JLabel chairStatus, JProgressBar progress){**

**duerme = false;**

**r = new Random();**

**this.m = m;**

**this.stateBarBer = stateBarber;**

**this.chairStatus = chairStatus;**

**this.progress = progress;**

**}**

**public void dormir(){**

**try{**

**Log("Barbero se pone a dormir");**

**stateBarBer.setText("barbero se pone a dormir");**

**duerme = true;**

**m.masterChair = true;**

**progress.setValue(0);**

**chairStatus.setText("Barbero durmiendo en la silla");**

**synchronized(this){**

**wait();**

**}**

**Log("Desperto el barbero");**

**stateBarBer.setText("Desperto el barbero");**

**chairStatus.setText("desocupado");**

**duerme = false;**

**m.masterChair = false;**

**} catch (InterruptedException ex) {**

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

**}**

**}**

**@Override**

**public void run() {**

**while(true){**

**if(!m.masterChair && m.getCuantosClientes()==0 && !duerme){**

**dormir();**

**}**

**Log("Barbero Dormido?: "+duerme);**

**}**

**}**

**public void Log(String mensaje){**

**System.out.println(mensaje);**

**}**

**}**

**Módulo “Cliente.java”**

**package peluquero;**

**import javax.swing.DefaultListModel;**

**import javax.swing.JLabel;**

**import javax.swing.JList;**

**public class Cliente implements Runnable{**

**Monitor m;**

**Barbero barbero;**

**DefaultListModel waitingList;**

**JLabel stateBarBer;**

**int posInList;**

**public Cliente(Monitor m, Barbero b, DefaultListModel waitingList, JLabel statusBarber, int numCliente){**

**this.waitingList = waitingList;**

**Log("Bienvenido, nuevo cliente");**

**barbero = b;**

**this.m = m;**

**this.stateBarBer = statusBarber;**

**posInList = numCliente;**

**m.clientesEspera++;**

**waitingList.addElement("Cliente Numero "+numCliente);**

**Log("Hay "+ m.getCuantosClientes() + " clientes esperando");**

**}**

**@Override**

**public void run() {**

**if(barbero.duerme){**

**despertar();**

**}**

**while(true){**

**if(Thread.currentThread().isAlive()){**

**if(!m.masterChair){**

**m.cortarPelo(this.posInList);**

**break;**

**}**

**}**

**}**

**}**

**public void despertar(){**

**Log("Despertando al barbero");**

**stateBarBer.setText("Despertando al barbero");**

**synchronized(barbero){**

**barbero.notify();**

**}**

**}**

**public void Log(String mensaje){**

**System.out.println(mensaje);**

**}**

**}**

**Módulo “Monitor.java”**

**package peluquero;**

**import java.util.ArrayList;**

**import java.util.List;**

**import java.util.Random;**

**import java.util.logging.Level;**

**import java.util.logging.Logger;**

**import javax.swing.DefaultListModel;**

**import javax.swing.JLabel;**

**import javax.swing.JProgressBar;**

**public class Monitor implements Runnable{**

**int nSillas;**

**boolean masterChair;**

**int clientesEspera;**

**JLabel stateBarBer;**

**DefaultListModel listModel;**

**JLabel chairStatus;**

**JProgressBar progress;**

**Monitor(int num, JLabel stateBarBer, DefaultListModel defaultList, JLabel chairStatus, JProgressBar progress){**

**nSillas = num;**

**masterChair = false;**

**clientesEspera = 0;**

**this.stateBarBer = stateBarBer;**

**listModel = defaultList;**

**this.chairStatus = chairStatus;**

**this.progress = progress;**

**}**

**public synchronized void cortarPelo(int posInList){**

**try {**

**masterChair = true;**

**clientesEspera--;**

**listModel.removeElement("Cliente Numero "+posInList);**

**chairStatus.setText("ocupado por cliente "+posInList);**

**Log("Barbero cortando el pelo...");**

**stateBarBer.setText("Barbero cortando el pelo...");**

**Random r = new Random();**

**int i = 0;**

**int max = r.nextInt(5000);**

**progress.setMaximum(max);**

**while(i<max){**

**Thread.sleep(1);**

**progress.setValue(i);**

**i++;**

**}**

**Log("Servido");**

**Log("Clientes esperando: "+clientesEspera);**

**masterChair = false;**

**chairStatus.setText("desocupado");**

**} catch (InterruptedException ex) {**

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

**}**

**}**

**public int getCuantosClientes(){**

**return clientesEspera;**

**}**

**@Override**

**public void run() {**

**while(true){**

**//para las sillas...**

**}**

**}**

**public void Log(String mensaje){**

**System.out.println(mensaje);**

**}**

**}**

**Módulo “Principal.java”**

**package peluquero;**

**public class Principal {**

**public static void main(String args[]){**

**Ventana v = new Ventana();**

**v.setTitle("Integrantes");**

**v.setVisible(true);**

**v.setLocationRelativeTo(null);**

**}**

**}**

**Módulo “Integrantes.java”**

**package peluquero;**

**import javax.swing.ImageIcon;**

**public class Integrantes extends javax.swing.JFrame {**

**public Integrantes() {**

**initComponents();**

**setIconImage(new ImageIcon(getClass().getResource("/ico/u.png")).getImage());**

**}**

**// Variables declaration - do not modify**

**private javax.swing.JLabel jLabel15;**

**private javax.swing.JLabel jLabel16;**

**private javax.swing.JLabel jLabel17;**

**private javax.swing.JLabel jLabel19;**

**private javax.swing.JLabel jLabel20;**

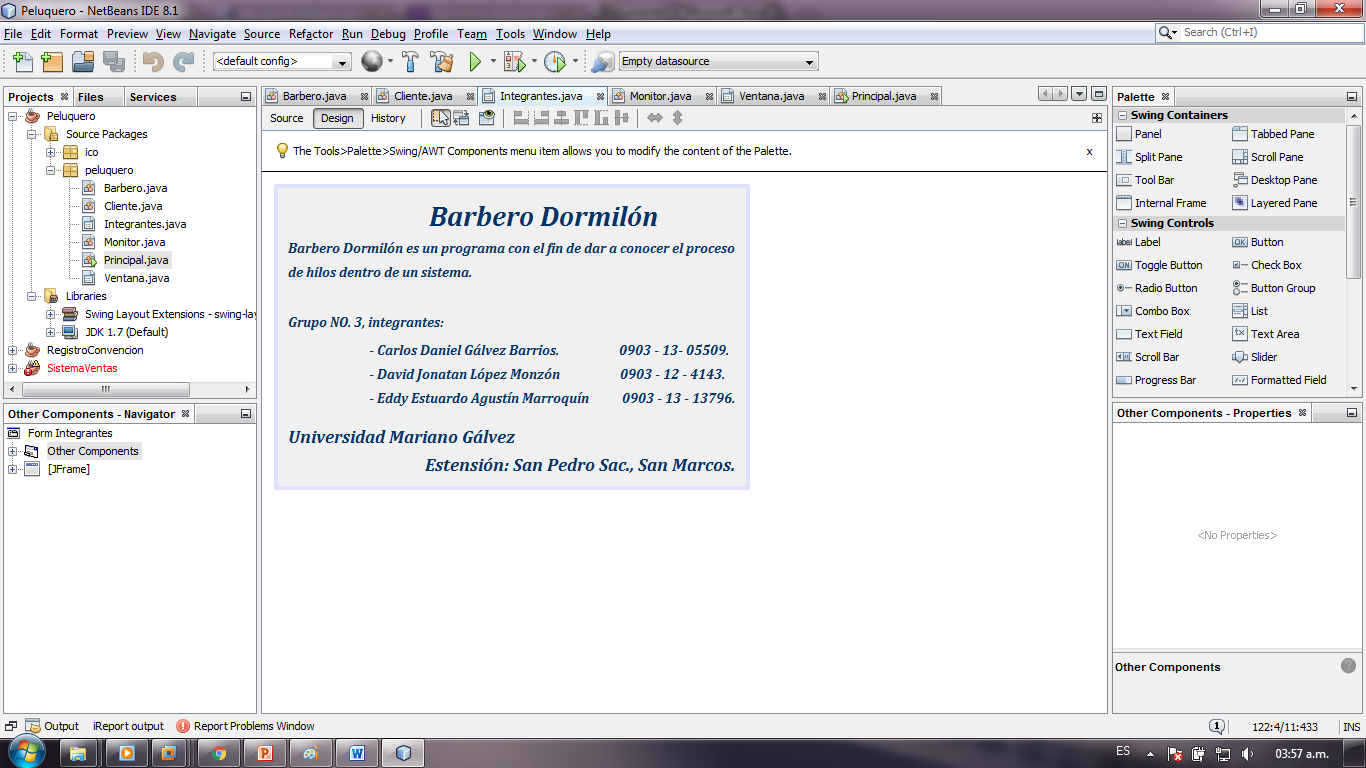
**private javax.swing.JLabel jLabel21;**

**private javax.swing.JLabel jLabel7;**

**private javax.swing.JLabel jLabel8;**

**private javax.swing.JLabel jLabel9;**

**INTERFAZ GRAFICA**



**Módulo “Ventana.java”**

**package peluquero;**

**import java.util.ArrayList;**

**import java.util.List;**

**import java.util.Random;**

**import java.util.logging.Level;**

**import java.util.logging.Logger;**

**import javax.swing.DefaultListModel;**

**import javax.swing.ImageIcon;**

**import javax.swing.JFormattedTextField;**

**import javax.swing.JOptionPane;**

**import javax.swing.JTextField;**

**import javax.swing.event.DocumentEvent;**

**import javax.swing.event.DocumentListener;**

**public class Ventana extends javax.swing.JFrame {**

**public Ventana() {**

**initComponents();**

**setTextFieldChanged(nChairsTextBox);**

**startButton.setEnabled(false);**

**stopButton.setEnabled(false);**

**this.setTitle("PROYECTO SISTEMAS OPERATIVOS");**

**this.setLocationRelativeTo(null);**

**setIconImage(new ImageIcon(getClass().getResource("/ico/u.png")).getImage());**

**}**

**private void startButtonActionPerformed(java.awt.event.ActionEvent evt) {**

**try{**

**Integer.parseInt(nChairsTextBox.getText());**

**}catch(NumberFormatException e){**

**JOptionPane.showMessageDialog(this, "Invalid number");**

**return;**

**}**

**final int numeroSillas = Integer.parseInt(nChairsTextBox.getText());**

**final Monitor m = new Monitor(numeroSillas, stateBarber, model, chairStatus, progressCut);**

**final Barbero barbero = new Barbero(m, stateBarber, chairStatus, progressCut);**

**List<Cliente> clientes = new ArrayList<Cliente>();**

**new Thread(barbero).start();**

**new Thread(m).start();**

**startButton.setEnabled(false);**

**stopButton.setEnabled(true);**

**new Thread(new Runnable() {**

**@Override**

**public void run() {**

**try {**

**int numCliente=0;**

**Random r = new Random();**

**while(true){**

**Thread.sleep(r.nextInt(10000));**

**if(m.clientesEspera<numeroSillas){**

**Cliente nuevo = new Cliente(m, barbero, model, stateBarber, ++numCliente);**

**Thread t = new Thread(nuevo);**

**t.start();**

**}else{**

**System.out.println("Llego un cliente y se fue");**

**}**

**}**

**} catch (InterruptedException ex) {**

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

**}**

**}**

**}).start();**

**}**

**private void stopButtonActionPerformed(java.awt.event.ActionEvent evt) {**

**System.exit(0);**

**}**

**private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {**

**System.exit(0);**

**}**

**private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {**

**Integrantes i = new Integrantes();**

**i.setTitle("Integrantes");**

**i.setVisible(true);**

**i.setLocationRelativeTo(null);**

**}**

**private void setTextFieldChanged(JTextField txt){**

**final JTextField t = txt;**

**DocumentListener documentListener = new DocumentListener() {**

**@Override**

**public void insertUpdate(DocumentEvent de) {**

**changeEnabled(de);**

**}**

**@Override**

**public void removeUpdate(DocumentEvent de) {**

**changeEnabled(de);**

**}**

**@Override**

**public void changedUpdate(DocumentEvent de) {**

**changeEnabled(de);**

**}**

**};**

**txt.getDocument().addDocumentListener(documentListener);**

**}**

**private void changeEnabled(DocumentEvent event){**

**DocumentEvent.EventType type = event.getType();**

**if (type.equals(DocumentEvent.EventType.CHANGE)){**

**}**

**else if (type.equals(DocumentEvent.EventType.INSERT)){**

**if(nChairsTextBox.getText().length()>0){**

**startButton.setEnabled(true);**

**}**

**}**

**else if (type.equals(DocumentEvent.EventType.REMOVE)){**

**if(nChairsTextBox.getText().length()==0){**

**startButton.setEnabled(false);**

**}**

**}**

**}**

**// Variables declaration - do not modify**

**private javax.swing.JLabel chairStatus;**

**private javax.swing.JPanel container;**

**private javax.swing.JLabel inChairLabel;**

**private javax.swing.JMenu jMenu1;**

**private javax.swing.JMenuBar jMenuBar1;**

**private javax.swing.JMenuItem jMenuItem1;**

**private javax.swing.JMenuItem jMenuItem2;**

**private javax.swing.JPanel jPanel1;**

**private javax.swing.JScrollPane jScrollPane1;**

**private javax.swing.JLabel nChairsLabel;**

**private javax.swing.JTextField nChairsTextBox;**

**private javax.swing.JProgressBar progressCut;**

**private javax.swing.JButton startButton;**

**private javax.swing.JLabel stateBarber;**

**private javax.swing.JLabel stateLabel;**

**private javax.swing.JButton stopButton;**

**private javax.swing.JLabel waitingLabel;**

**private javax.swing.JList waitingList;**

**private DefaultListModel model = new DefaultListModel();**

**// End of variables declaration**

**}**

**INTERFAZ GRAFICA**

