
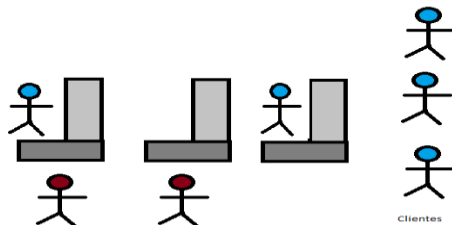
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		PRÁCTICA DE LABORATORIO	
CARRERA: COMPUTACION		ASIGNATURA: PROGRAMACION APLICADA	
NRO. PRÁCTICA:	2	TÍTULO PRÁCTICA: : Prueba Practica 2 Desarrollo e implementación de un sistema de simulación de acceso y atención bancaria	
OBJETIVO ALCANZADO: Reforzar los conocimientos adquiridos en clase sobre la programación en Hilos en un contexto real.			
ACTIVIDADES DESARROLLADAS			
1. Planteamiento y descripción del problema. 1. Enunciado: Realizar un sistema de simulación de acceso y atención a través de colas de un banco. Problema: Un banco necesita controlar el acceso a cuentas bancarias y para ello desea hacer un programa de prueba en Java que permita lanzar procesos que ingresen y retiren dinero a la vez y comprobar así si el resultado final es el esperado. Se parte de una cuenta con 100 euros y se pueden tener procesos que ingresen 100 euros, 50 o 20. También se pueden tener procesos que retiran 100, 50 o 20 euros. Se desean tener los siguientes procesos: <ul style="list-style-type: none"> • 40 procesos que ingresan 100 • 20 procesos que ingresan 50 • 60 que ingresen 20. De la misma manera se desean lo siguientes procesos que retiran cantidades. <ul style="list-style-type: none"> • 40 procesos que retiran 100 • 20 procesos que retiran 50 • 60 que retiran 20. Además en el banco, existen 3 cajeros que pueden atender y hay un cola inicial de 10 clientes para ser atendidos, el proceso de atención es de 20 – 15 segundos y los clientes llegan constantemente cada 30 - 50 segundos. Ningún cajero puede atender simultáneamente, adicionalmente el tiempo de moverme de la			

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

cola al estante del cajero es de 2 - 5 segundos, esto deberán ser generados aleatoriamente entre los 100 clientes que disponen una cuenta, estos pueden volver a ingresar el número de veces que sea necesario.

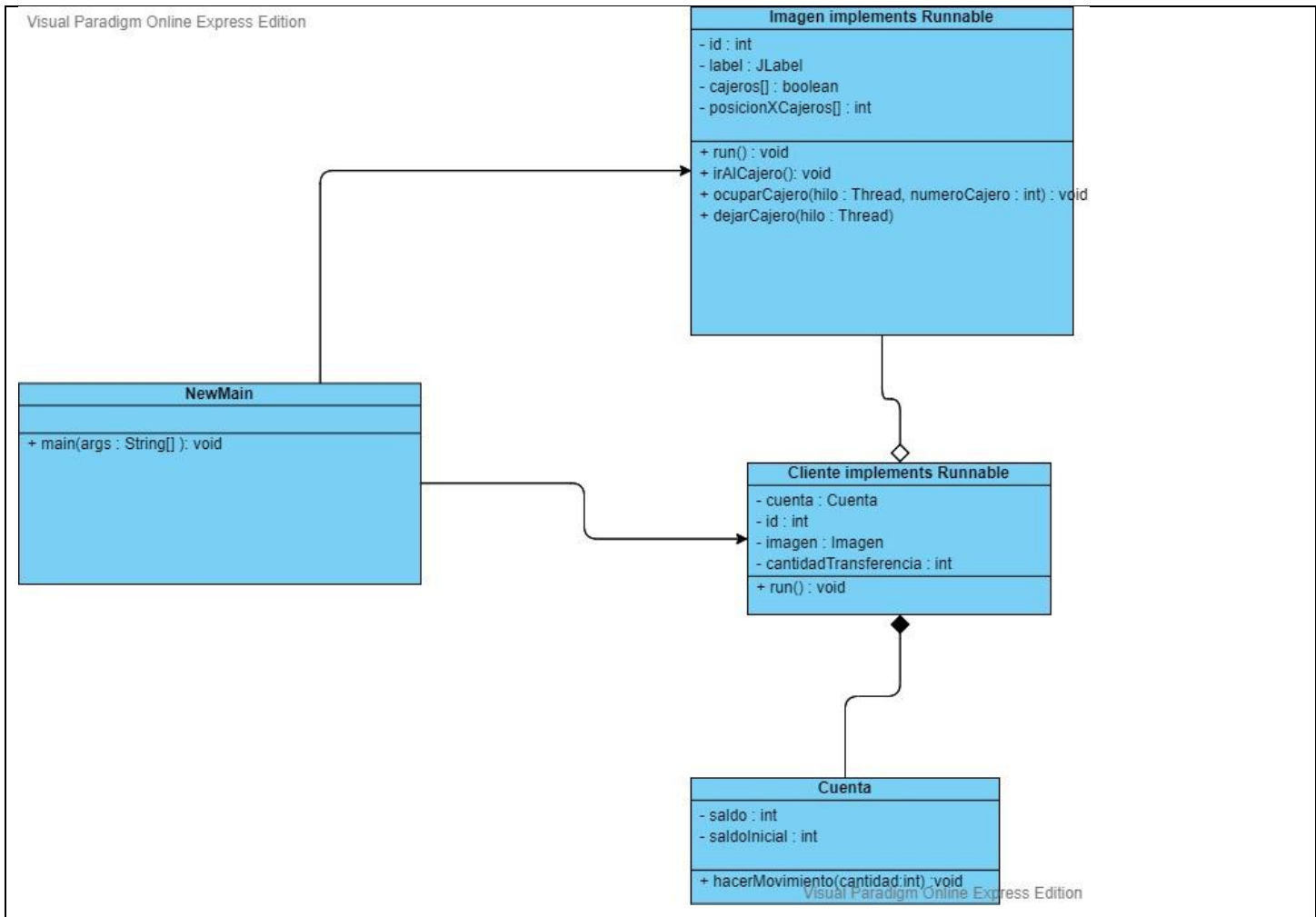


Se desea comprobar que tras la ejecución la cuenta tiene exactamente 100 euros, que era la cantidad de la que se disponía al principio. Realizar el programa Java que demuestra dicho hecho.

Calificación:

- Diagrama de Clase 10%
- MVC: 10%
- Técnicas de Programación aplicadas (Java 8, Reflexión y Programación Genérica): 10%
- Hilos 30%
- Sincronización 10%
- Interfaz Gráfica de simulación 20%
- Informe: 10%

2. Diagramas de Clases.



3. Descripción de la solución y pasos seguidos.

Para la solución de este problema se planteó la siguiente solución:

- Se creó la clase Cuenta donde se guardan los datos de una cuenta bancaria: saldo, saldoInicial. Ambos parámetros son de tipo int. Son de ese tipo ya que las cuentas deben comenzar en cero y además todas las transferencias que sean necesarias serán de: 100, 50, 20; esto nos facilita la manipulación de datos.
- A continuación, se creó la clase Cliente que alberga una cuenta bancaria, una imagen (se explicará más adelante sobre esta clase), una cantidad de transferencia que nos sirve para saber de cuanto es la transferencia que la persona quiere hacer, y cuenta con un id que es un identificador del cliente, este va desde el cero hasta el 99. Además, esta clase implementa la interfaz Runnable, por lo que se tuvo que sobrescribir el método run(). En este método solamente iniciamos el hilo y esperamos que acabe. Eso significa que se hace la transferencia y el label que contiene la imagen se mueve hacia el cajero y también deja el cajero.
- Se creó la clase Imagen. Esta clase es como un controlador del Label para poder mover el componente por la ventana. Esta clase tiene como atributos: int id, el cual es el identificador; JLabel label el cual es el componente que tiene la imagen y la cual la vamos a mover por la ventana, boolean[] cajeros que nos sirve para saber que cajeros están ocupados y cuáles no, int[] posicionXCajeros para saber cuál es la posición de los cajeros en la ventana y así poder dirigir el label hacia ellos.

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

- Por último, se creo una clase principal llamada NewMain. Aquí creamos la ventana manualmente, es decir se le añadieron las cosas creándolas en el método main de la clase. Esta clase es la encargada de crear los clientes, imágenes, labels y cuentas; también muestra en otra ventana los estados de cuenta. Para eso cerramos la ventana de la simulación y creamos otra ventana con JList y un JScrollBar para poder observar todos los estados de cuenta los clientes creados.

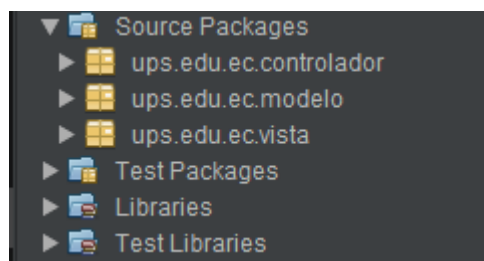
Java 8 en adelante:


Nuevo switch-case:

```
switch (this.id) {
    case 9 -> {
        switch (numeroCajero) {
            case 0 -> {
                int movimiento = 55 / tiempoLlegarCajero;
                for (int i = 0; i < tiempoLlegarCajero; i++) {
                    label.setLocation(label.getLocation().x + movimiento, label.getLocation().y);
                    //System.out.println(i);
                    Thread.sleep(308);
                }
            }

            case 1 -> {
                int movimiento2 = 130 / tiempoLlegarCajero;
                for (int i = 0; i < tiempoLlegarCajero; i++) {
                    label.setLocation(label.getLocation().x + movimiento2, label.getLocation().y);
                    //System.out.println(i);
                    Thread.sleep(300);
                }
            }

            case 2 -> {
                int movimiento3 = 204 / tiempoLlegarCajero;
                for (int i = 0; i < tiempoLlegarCajero; i++) {
                    label.setLocation(label.getLocation().x + movimiento3, label.getLocation().y);
                    //System.out.println(i);
                    Thread.sleep(290);
                }
            }
        }
    }
}
```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

CLASE CLIENTE:

Public class Cliente implements Runnable {

Imagen imagen;

private int id;

private Cuenta cuenta;

private int cantidadTransferencia;

private JTextField texto;

//private JFrame frame;

public Cliente(Cuenta cuenta, int cantidadTransferencia) {

 this.cuenta = cuenta;

 this.cantidadTransferencia = cantidadTransferencia;

}

public Cliente(int id, Cuenta cuenta, int cantidadTransferencia) {

 this.id = id;

 this.cuenta = cuenta;

 this.cantidadTransferencia = cantidadTransferencia;

}

public Cliente(Imagen imagen, int id, Cuenta cuenta, int cantidadTransferencia) {

 this.imagen = imagen;

 this.id = id;

 this.cuenta = cuenta;

 this.cantidadTransferencia = cantidadTransferencia;

 texto = new JTextField("Informacion");

}

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

/*
public JFrame getFrame() {
    return frame;
}

public void setFrame(JFrame frame) {
    this.frame = frame;
}
*/
public JTextField getTexto() {
    return texto;
}

public void setTexto(JTextField texto) {
    this.texto = texto;
}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public Cuenta getCuenta() {
    return cuenta;
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

public void setCuenta(Cuenta cuenta) {
    this.cuenta = cuenta;
}

public int getCantidadTransferencia() {
    return cantidadTransferencia;
}

public void setCantidadTransferencia(int cantidadTransferencia) {
    this.cantidadTransferencia = cantidadTransferencia;
}

public Imagen getImagen() {
    return imagen;
}


public void setImagen(Imagen imagen) {
    this.imagen = imagen;
}

@Override
public String toString() {
    return "Cliente: " + "id: " + id + " cuenta: " + cuenta;
}

@Override
public void run() {

    Thread hilo = new Thread(imagen);

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

hilo.start();

try {
    cuenta.hacerMovimiento(cantidadTransferencia);
    hilo.join();

    /*
    int numeroCompo = frame.getComponentCount();

    JTextField txtInfo = (JTextField) frame.getComponent(numeroCompo);

    txtInfo.setText("Cliente id: " + this.id + " saldo: " + this.cuenta.getSaldo());

    frame.add(txtInfo);
    frame.repaint();
    frame.getContentPane().validate();*/
    hilo.interrupt();
} catch (Exception ex) {
    Logger.getLogger(Cliente.class.getName()).log(Level.SEVERE, null, ex);
}

}

public JTextField mostrarInformacion(JTextField textoInfo) {
    textoInfo.setText("Cliente id: " + this.id + " saldo: " + this.cuenta.getSaldo());

    return textoInfo;
}
}

```


	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

CLASE CUENTA:

/*

* 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.

*/

package ups.edu.ec.modelo;

/**

*

* @author User

*/

public class Cuenta {

private int saldo;

private int saldoInicial;

public Cuenta(int saldo) {

 this.saldoInicial = saldo;

 this.saldo = saldo;

}

public synchronized void hacerMovimiento(int cantidad) {

 this.saldo = this.saldo + cantidad;

}

public boolean esSimulacionCorrecta() {

 return this.saldo == this.saldoInicial;

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

}

public int getSaldo() {
    return this.saldo;
}

public int getSaldolnicial() {
    return saldolnicial;
}

public void setSaldolnicial(int saldolnicial) {
    this.saldolnicial = saldolnicial;
}

@Override
public String toString() {
    return "Cuenta{" + "saldo=" + saldo + ", saldolnicial=" + saldolnicial + "}";
}

}

```

CLASE IMAGEN:

```
Public class Imagen implements Runnable {
```

```

    int id;


    JLabel label;

    boolean[] cajeros = new boolean[3];

    int[] posicionXCajeros = new int[3];

    Cuenta cuenta;

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
//private JFrame frame;

public Imagen(int id, JLabel label, boolean[] cajeros, int[] posicionXCajeros) {
    this.id = id;
    this.label = label;
    this.cajeros = cajeros;
    this.posicionXCajeros = posicionXCajeros;
}

/*
public JFrame getFrame() {
    return frame;
}

public void setFrame(JFrame frame) {
    this.frame = frame;
}*/

public Cuenta getCuenta() {
    return cuenta;
}

public void setCuenta(Cuenta cuenta) {
    this.cuenta = cuenta;
}

@Override
public synchronized void run() {
```

```
irAlCajero();

//this.run();

/*int ubicacionXCajero = 0;

int numeroCajero = 0;

Random random = new Random();

boolean ocupados = false;

int ubicacionX = label.getLocation().x;

int ubicacionY = label.getLocation().y;

try {

    for (int i = 0; i < 3; i++) {
        if (!cajeros[i]) {
            numeroCajero = i;
            ocupados = false;
            cajeros[i] = true;
            break;
        } else {
            ocupados = true;
        }
    }

    if (ocupados) {
        this.wait();
    }

    for (int i = 0; i < 2; i++) {
        label.setLocation(label.getLocation().x, label.getLocation().y + 60);
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

//label.setLocation((label.getLocationOnScreen().x+10), label.getLocationOnScreen().y);

Thread.sleep(100);

//System.out.println("5555");
}

int tiempoLlegarCajero = (int) (Math.random() * (3 - 1 + 1) + 1);
//System.out.println(tiempoLlegarCajero);

//int movimiento = posicionXCajeros[numeroCajero] / tiempoLlegarCajero;
//System.out.println(movimiento);
switch (id) {
    case 9 -> {
        switch (numeroCajero) {
            case 0 -> {
                int movimiento = 55 / tiempoLlegarCajero;
                for (int i = 0; i < tiempoLlegarCajero; i++) {
                    label.setLocation(label.getLocation().x + movimiento, label.getLocation().y);
                    //System.out.println(i);
                    Thread.sleep(500);
                }
            }
        }

        case 1 -> {
            int movimiento2 = 130 / tiempoLlegarCajero;
            for (int i = 0; i < tiempoLlegarCajero; i++) {
                label.setLocation(label.getLocation().x + movimiento2, label.getLocation().y);
                //System.out.println(i);
                Thread.sleep(500);
            }
        }
    }
}

```

```
}

case 2 -> {

    int movimiento3 = 204 / tiempoLlegarCajero;

    for (int i = 0; i < tiempoLlegarCajero; i++) {

        label.setLocation(label.getLocation().x + movimiento3, label.getLocation().y);

        //System.out.println(i);

        Thread.sleep(500);

    }

}

}

}

case 8 -> {

    switch (numeroCajero) {

        case 0 -> {

            int movimiento = 95 / tiempoLlegarCajero;

            for (int i = 0; i < tiempoLlegarCajero; i++) {

                label.setLocation(label.getLocation().x + movimiento, label.getLocation().y);

                System.out.println(i);

                Thread.sleep(500);

            }

        }

    }

case 1 -> {

    int movimiento2 = 170 / tiempoLlegarCajero;

    for (int i = 0; i < tiempoLlegarCajero; i++) {

        label.setLocation(label.getLocation().x + movimiento2, label.getLocation().y);

        System.out.println(i);

    }

}
```

```
        Thread.sleep(500);
    }
}

case 2 -> {
    int movimiento3 = 244 / tiempoLlegarCajero;
    for (int i = 0; i < tiempoLlegarCajero; i++) {
        label.setLocation(label.getLocation().x + movimiento3, label.getLocation().y);
        System.out.println(i);
        Thread.sleep(500);
    }
}

}
}

default -> {
    switch (numeroCajero) {
        case 0 -> {
            int movimiento = 115 / tiempoLlegarCajero;
            for (int i = 0; i < tiempoLlegarCajero; i++) {
                label.setLocation(label.getLocation().x + movimiento, label.getLocation().y);
                System.out.println(i);
                Thread.sleep(500);
            }
        }

        case 1 -> {
            int movimiento2 = 190 / tiempoLlegarCajero;
            for (int i = 0; i < tiempoLlegarCajero; i++) {
```

```
        label.setLocation(label.getLocation().x + movimiento2, label.getLocation().y);
        System.out.println(i);
        Thread.sleep(500);
    }
}

case 2 -> {
    int movimiento3 = 264 / tiempoLlegarCajero;
    for (int i = 0; i < tiempoLlegarCajero; i++) {
        label.setLocation(label.getLocation().x + movimiento3, label.getLocation().y);
        System.out.println(i);
        Thread.sleep(500);
    }
}

}
}

for (int i = 0; i < 7; i++) {
    label.setLocation(label.getLocation().x, label.getLocation().y - 10);
    //label.setLocation((label.getLocationOnScreen().x+10), label.getLocationOnScreen().y);
    Thread.sleep(100);
}

this.ocuparCajero(new Thread(this), numeroCajero);

this.dejarCajero(new Thread(this));
```



```
int tiempoLlegarCajero2 = (int) (Math.random() * (50 - 30 + 1) + 30);

Thread.sleep(tiempoLlegarCajero2);

label.setLocation(ubicacionX, ubicacionY);

} catch (InterruptedException e) {
    e.printStackTrace();
}*/
}

public synchronized void irAlCajero() {

    int ubicacionXCajero = 0;
    int numeroCajero = 0;
    Random random = new Random();
    boolean ocupados = false;
    int ubicacionX = label.getLocation().x;
    int ubicacionY = label.getLocation().y;

    try {

        for (int i = 0; i < 3; i++) {
            if (!cajeros[i]) {
                numeroCajero = i;
                ocupados = false;
                cajeros[i] = true;
                break;
            } else {
                ocupados = true;
            }
        }
    }
}
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

    }

}

if (ocupados) {
    this.wait();
}

for (int i = 0; i < 5; i++) {
    label.setLocation(label.getLocation().x, label.getLocation().y + 22);
    //label.setLocation((label.getLocationOnScreen().x+10), label.getLocationOnScreen().y);
    Thread.sleep(10);
    //System.out.println("5555");
}

int tiempoLlegarCajero = (int) (Math.random() * (9 - 7 + 1) + 7);
//System.out.println(tiempoLlegarCajero);

/*int movimiento = posicionXCajeros[numeroCajero] / tiempoLlegarCajero;
System.out.println(movimiento);*/
switch (this.id) {
    case 9 -> {
        switch (numeroCajero) {
            case 0 -> {
                int movimiento = 55 / tiempoLlegarCajero;
                for (int i = 0; i < tiempoLlegarCajero; i++) {
                    label.setLocation(label.getLocation().x + movimiento, label.getLocation().y);
                    //System.out.println(i);
                    Thread.sleep(308);
                }
            }
        }
    }
}

```

```
}

case 1 -> {

    int movimiento2 = 130 / tiempoLlegarCajero;

    for (int i = 0; i < tiempoLlegarCajero; i++) {

        label.setLocation(label.getLocation().x + movimiento2, label.getLocation().y);

        //System.out.println(i);

        Thread.sleep(300);

    }

}

case 2 -> {

    int movimiento3 = 204 / tiempoLlegarCajero;

    for (int i = 0; i < tiempoLlegarCajero; i++) {

        label.setLocation(label.getLocation().x + movimiento3, label.getLocation().y);

        //System.out.println(i);

        Thread.sleep(290);

    }

}

}

}

case 8 -> {

    switch (numeroCajero) {

        case 0 -> {

            int movimiento = 95 / tiempoLlegarCajero;

            for (int i = 0; i < tiempoLlegarCajero; i++) {

                label.setLocation(label.getLocation().x + movimiento, label.getLocation().y);

                System.out.println(i);

            }

        }

    }

}
```

```
        Thread.sleep(310);
    }
}

case 1 -> {
    int movimiento2 = 170 / tiempoLlegarCajero;
    for (int i = 0; i < tiempoLlegarCajero; i++) {
        label.setLocation(label.getLocation().x + movimiento2, label.getLocation().y);
        System.out.println(i);
        Thread.sleep(300);
    }
}

case 2 -> {
    int movimiento3 = 244 / tiempoLlegarCajero;
    for (int i = 0; i < tiempoLlegarCajero; i++) {
        label.setLocation(label.getLocation().x + movimiento3, label.getLocation().y);
        System.out.println(i);
        Thread.sleep(290);
    }
}

}
}

default -> {
    switch (numeroCajero) {
        case 0 -> {
            int movimiento = 115 / tiempoLlegarCajero;
            for (int i = 0; i < tiempoLlegarCajero; i++) {
```

```
        label.setLocation(label.getLocation().x + movimiento, label.getLocation().y);

        System.out.println(i);

        Thread.sleep(310);
    }
}

case 1 -> {

    int movimiento2 = 190 / tiempoLlegarCajero;

    for (int i = 0; i < tiempoLlegarCajero; i++) {

        label.setLocation(label.getLocation().x + movimiento2, label.getLocation().y);

        System.out.println(i);

        Thread.sleep(300);

    }

}

case 2 -> {

    int movimiento3 = 264 / tiempoLlegarCajero;

    for (int i = 0; i < tiempoLlegarCajero; i++) {

        label.setLocation(label.getLocation().x + movimiento3, label.getLocation().y);

        System.out.println(i);

        Thread.sleep(290);

    }

}

}

}

}

/**
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

* para que suba al cajero

*/

```
for (int i = 0; i < 7; i++) {
```

```
    label.setLocation(label.getLocation().x, label.getLocation().y - 10);
```

```
    //label.setLocation((label.getLocationOnScreen().x+10), label.getLocationOnScreen().y);
```

```
    Thread.sleep(100);
```

```
}
```

```
this.ocuparCajero(new Thread(this), numeroCajero);
```

```
this.dejarCajero(new Thread(this));
```

```
notifyAll();
```

```
/* int tiempoLlegarCajero2 = (int) (Math.random() * (50 - 30 + 1) + 30);
```

```
Thread.sleep(tiempoLlegarCajero2);*/
```

```
label.setLocation(ubicacionX, ubicacionY);
```

```
} catch (InterruptedException e) {
```

```
    e.printStackTrace();
```


```
}
```

```
}
```

```
public void ocuparCajero(Thread hilo, int numeroCajero) {
```

```
    int tiempoCajero = (int) (Math.random() * (20 - 15 + 1) + 15);
```

```
    try {
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

Thread.sleep(tiempoCajero * 100);
cajeros[numeroCajero] = false;
/*
int numeroCompo = frame.getComponentCount();

JTextField txtInfo = (JTextField) frame.getComponent(numeroCompo);

txtInfo.setText(t);*/

} catch (InterruptedException ex) {
    Logger.getLogger(Imagen.class.getName()).log(Level.SEVERE, null, ex);
}

System.out.println((int) (Math.random() * (20 - 15 + 1) + 15));

/*
try {

    int tiempoCajero = (int) (Math.random() * (20 - 15 + 1) + 15);

    Thread.sleep(tiempoCajero * 1000);

    cajeros[numeroCajero] = false;
} catch (InterruptedException ex) {
    Logger.getLogger(Imagen.class.getName()).log(Level.SEVERE, null, ex);
}*/
}

public synchronized void dejarCajero(Thread hilo) {

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

for (int i = 0; i < 13; i++) {
    notifyAll();
    label.setLocation(label.getLocation().x, label.getLocation().y + 30);
    try {
        //label.setLocation((label.getLocationOnScreen().x+10), label.getLocationOnScreen().y);
        Thread.sleep(100);
    } catch (InterruptedException ex) {
        Logger.getLogger(Imagen.class.getName()).log(Level.SEVERE, null, ex);
    }
}

}

}

/*
public void moverPersonas(JLabel[] labelsFila){

    for (int i=0;i<)
    }*/
}

CLASE NEWMAIN:

/**
 *
 * @author User
 */
public class NewMain {

    /**
     * @param args the command line arguments

```


	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

*/

public static void main(String[] args) throws InterruptedException {

    // TODO code application logic here

    JFrame frame = new JFrame();

    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //application will be closed when you close frame

    frame.setSize(1300, 600);

    frame.setLayout(new GridLayout(1, 2, 4, 5));

    //JPanel panelMayor = new JPanel(new GridLayout(1, 2, 5, 10));

    JPanel panelPersonas = new JPanel();

    JLabel[] labels = new JLabel[13];

    ImageIcon imagenPersona = new ImageIcon("contornoPersona.jpg");

    for (int i = 0; i < 10; i++) {

        JLabel l2 = new JLabel(imagenPersona, JLabel.CENTER);

        panelPersonas.add(l2);

        labels[i] = l2;

    }

    boolean[] cajeros = new boolean[3];

    int[] posicionXCajeros = new int[3];

    //JPanel panelCajeros = new JPanel(new GridLayout(1, 3, 5, 10));

    ImageIcon imagenCajero = new ImageIcon("cajero.jpg");


    for (int i = 0; i < 3; i++) {

        JLabel l2 = new JLabel(imagenCajero, JLabel.CENTER);

        panelPersonas.add(l2);

        posicionXCajeros[i] = l2.getLocation().x;

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

labels[i + 10] = l2;
cajeros[i] = false;
}


/*int contador = 0;
JLabel[] labelsSalidos = new JLabel[3];
for (int i = 0; i < 3; i++) {
    JLabel l2 = new JLabel(imagenPersona, JLabel.CENTER);
    l2.setLocation(600 + contador, 55);
    labelsSalidos[i] = l2;
    contador += 50;
    panelPersonas.add(l2);
}*/
//panelMayor.add(panelPersonas);
//panelMayor.add(panelCajeros);
frame.add(panelPersonas);
/*
JList<String> jList = new JList<>();
frame.add(jList);*/
JTextField txtInfo = new JTextField("Informacion");
frame.add(txtInfo);

frame.getContentPane().validate();
frame.repaint();

frame.setVisible(true);

//System.out.println(labels[9].getText());
Imagen im1 = new Imagen(9, labels[9], cajeros, posicionXCajeros);

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

Imagen i2 = new Imagen(8, labels[8], cajeros, posicionXCajeros);
Imagen i3 = new Imagen(7, labels[7], cajeros, posicionXCajeros);

Imagen[] imagenes = new Imagen[3];
imagenes[0] = im1;
imagenes[1] = i2;
imagenes[2] = i3;

/*    im1.setFrame(frame);
i2.setFrame(frame);
i3.setFrame(frame);*/

/* Thread t = new Thread(i1);
    Thread t2 = new Thread(i2);
    Thread t3 = new Thread(i3);
    t.start();

    t2.start();

    t3.start();*/
/**
 * intento de unir con la clase test
 *
 */
final int NUM_OPS_CON_100 = 40;
final int NUM_OPS_CON_50 = 20;
final int NUM_OPS_CON_20 = 60;

Thread[] hilosIngresan100 = new Thread[NUM_OPS_CON_100];

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
Thread[] hilosRetiran100 = new Thread[NUM_OPS_CON_100];
Thread[] hilosIngresan50 = new Thread[NUM_OPS_CON_50];
Thread[] hilosRetiran50 = new Thread[NUM_OPS_CON_50];
Thread[] hilosIngresan20 = new Thread[NUM_OPS_CON_20];
Thread[] hilosIngresan20SegundaVez = new Thread[20];
```


```
Cliente[] listadoClientes = new Cliente[100];
int contadorClientes = 0;
int numeroOperaciones100 = 0;
int numeroOperaciones50 = 0;
int numeroOperaciones20 = 0;
int numeroOperaciones20SegundaVez = 0;
int numeroRepetirAccion = 60;
int numeroEscoger = 1;
```

```
//creacion 100 clientes
for (int i = 0; i < 100; i++) {
    Cliente cl = new Cliente(i, new Cuenta(100), 0);
    //cl.setFrame(frame);
    listadoClientes[i] = cl;
}
```

```
int numeroImagenPersona = 2;
```

```
String[] informacionTransacciones = new String[120];
```

```
int contadorBorrarPersonaFila = 0;
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

for (int i = 0; i < 120; i++) {

    if (i == NUM_OPS_CON_100) {
        numeroEscoger++;
    } else if (i == (NUM_OPS_CON_100 + NUM_OPS_CON_50)) {
        numeroEscoger++;
    }
    /* if (numeroEscoger >3) {
        numeroEscoger = 1;
    }*/

    if (numerolMagenPersona == -1) {
        numerolMagenPersona = 2;
    }

    if (numeroEscoger == 1) {
        //numeroEscoger++;
        Cliente ingresa = listadoClientes[contadorClientes];
        ingresa.setlMagen(imagenes[numerolMagenPersona]);

        ingresa.setCantidadTransferencia(100);
        hilosIngresan100[numeroOperaciones100] = new Thread(ingresa);
        hilosIngresan100[numeroOperaciones100].start();

        hilosIngresan100[numeroOperaciones100].join(4000);

        numerolMagenPersona--;

        numeroOperaciones100++;
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

informacionTransacciones[i] = ingresa.toString();

//DefaultListModel listModel = (DefaultListModel) jList.getModel();
//jList.setListData(informacionTransacciones);
txtInfo.setText("Cliente id: " + ingresa.getId() + " saldo: " + (ingresa.getCuenta().getSaldo()));
System.out.println("Cliente id: " + ingresa.getId() + " saldo: " + (ingresa.getCuenta().getSaldo()));
//hilosIngresan100[numeroOperaciones100].join(7500);
//frame.add(ingresa.mostrarInformacion(txtInfo));

frame.add(txtInfo);

} else if (numeroEscoger == 2) {
    //numeroEscoger++;
    Cliente ingresa = listadoClientes[contadorClientes];
    ingresa.setImagen(imagenes[numeroImagenPersona]);
    numeroImagenPersona--;
    ingresa.setCantidadTransferencia(50);
    hilosIngresan50[numeroOperaciones50] = new Thread(ingresa);
    hilosIngresan50[numeroOperaciones50].start();
    hilosIngresan50[numeroOperaciones50].join(4000);
    numeroOperaciones50++;

    txtInfo.setText("Cliente id: " + ingresa.getId() + " saldo: " + (ingresa.getCuenta().getSaldo()));

    frame.add(txtInfo);

    System.out.println("Cliente id: " + ingresa.getId() + " saldo: " + (ingresa.getCuenta().getSaldo()));

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

} else if (numeroEscoger == 3) {
    //numeroEscoger++;

    if (i < 100) {
        //System.out.println(i + " " + contadorClientes);

        Cliente ingresa = listadoClientes[contadorClientes];
        ingresa.setImagen(imagenes[numerolImagenPersona]);
        numerolImagenPersona--;
        ingresa.setCantidadTransferencia(20);
        hilosIngresan20[numeroOperaciones20] = new Thread(ingresa);
        hilosIngresan20[numeroOperaciones20].start();
        hilosIngresan20[numeroOperaciones20].join(4000);
        numeroOperaciones20++;

        txtInfo.setText("Cliente id: " + ingresa.getId() + " saldo: " + (ingresa.getCuenta().getSaldo()));

        frame.add(txtInfo);

        System.out.println("Cliente id: " + ingresa.getId() + " saldo: " + (ingresa.getCuenta().getSaldo()));

    } else {
        Cliente ingresa = listadoClientes[numeroRepetirAccion];
        ingresa.setImagen(imagenes[numerolImagenPersona]);
        numerolImagenPersona--;
        numeroRepetirAccion++;
        ingresa.setCantidadTransferencia(20);
        hilosIngresan20SegundaVez[numeroOperaciones20SegundaVez] = new Thread(ingresa);
        hilosIngresan20SegundaVez[numeroOperaciones20SegundaVez].start();
    }
}

```

```
hilosIngresan20SegundaVez[numeroOperaciones20SegundaVez].join(4000);
numeroOperaciones20SegundaVez++;

txtInfo.setText("Cliente id: " + ingresa.getId() + " saldo: " + (ingresa.getCuenta().getSaldo()));

frame.add(txtInfo);


System.out.println("Cliente id: " + ingresa.getId() + " saldo: " + (ingresa.getCuenta().getSaldo()));

    }
}
/*
if (contadorBorrarPersonaFila < 7) {
    JLabel l = labels[i];
    l.setText("");
    labels[i] = l;
    frame.repaint();
}
*/
contadorClientes++;

System.out.println("i: " + i + " " + numeroEscoger);

}

numeroEscoger = 1;
contadorClientes = 0;
numeroOperaciones100 = 0;
numeroOperaciones50 = 0;
numeroOperaciones20 = 0;
```


	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

numeroOperaciones20SegundaVez = 0;

numeroRepetirAccion = 60;

numeroImagenPersona = 2;

for (int i = 0; i < 120; i++) {

    if (i == NUM_OPS_CON_100) {
        numeroEscoger++;
    } else if (i == (NUM_OPS_CON_100 + NUM_OPS_CON_50)) {
        numeroEscoger++;
    }

    if (numeroImagenPersona == -1) {
        numeroImagenPersona = 2;
    }

    if (numeroEscoger == 1 && i < NUM_OPS_CON_100) {

        Cliente retira = listadoClientes[contadorClientes];
        retira.setImagen(imagenes[numeroImagenPersona]);
        numeroImagenPersona--;
        retira.setCantidadTransferencia(-100);
        hilosRetiran100[numeroOperaciones100] = new Thread(retira);
        hilosRetiran100[numeroOperaciones100].start();
        hilosRetiran100[numeroOperaciones100].join(4000);

        numeroOperaciones100++;

        txtInfo.setText("Cliente id: " + retira.getId() + " saldo: " + (retira.getCuenta().getSaldo()));

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
frame.add(txtInfo);
```

```
System.out.println("Cliente id: " + retira.getId() + " saldo: " + (retira.getCuenta().getSaldo()));
```

```
} else if (numeroEscoger == 2) {
```

```
    Cliente retira = listadoClientes[contadorClientes];
```

```
    retira.setImagen(imagenes[numeroImagenPersona]);
```

```
    numeroImagenPersona--;
```

```
    retira.setCantidadTransferencia(-50);
```

```
    hilosRetiran50[numeroOperaciones50] = new Thread(retira);
```

```
    hilosRetiran50[numeroOperaciones50].start();
```

```
    hilosRetiran50[numeroOperaciones50].join(4000);
```

```
    numeroOperaciones50++;
```

```
    txtInfo.setText("Cliente id: " + retira.getId() + " saldo: " + (retira.getCuenta().getSaldo()));
```

```
    frame.add(txtInfo);
```

```
    System.out.println("Cliente id: " + retira.getId() + " saldo: " + (retira.getCuenta().getSaldo()));
```

```
} else if (numeroEscoger == 3) {
```

```
    if (i < 100) {
```

```
        Cliente retira = listadoClientes[contadorClientes];
```

```
        retira.setImagen(imagenes[numeroImagenPersona]);
```

```
        numeroImagenPersona--;
```

```
        retira.setCantidadTransferencia(-20);
```

```
        hilosIngresan20[numeroOperaciones20] = new Thread(retira);
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

hilosIngresan20[numeroOperaciones20].start();
hilosIngresan20[numeroOperaciones20].join(4000);
numeroOperaciones20++;

txtInfo.setText("Cliente id: " + retira.getId() + " saldo: " + (retira.getCuenta().getSaldo()));

frame.add(txtInfo);

System.out.println("Cliente id: " + retira.getId() + " saldo: " + (retira.getCuenta().getSaldo()));

} else {
    Cliente retira = listadoClientes[numeroRepetirAccion];
    retira.setImagen(imagenes[numeroImagenPersona]);
    numeroImagenPersona--;
    numeroRepetirAccion++;
    retira.setCantidadTransferencia(-20);
    hilosIngresan20SegundaVez[numeroOperaciones20SegundaVez] = new Thread(retira);
    hilosIngresan20SegundaVez[numeroOperaciones20SegundaVez].start();
    hilosIngresan20SegundaVez[numeroOperaciones20SegundaVez].join(4000);
    numeroOperaciones20SegundaVez++;


    txtInfo.setText("Cliente id: " + retira.getId() + " saldo: " + (retira.getCuenta().getSaldo()));

    frame.add(txtInfo);

    System.out.println("Cliente id: " + retira.getId() + " saldo: " + (retira.getCuenta().getSaldo()));

}
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

    contadorClientes++;
}

frame.setVisible(false);
frame.dispose();


JFrame frameCuentas = new JFrame("Cuentas");
frameCuentas.setSize(800, 600);
frameCuentas.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //application will be closed when you close
frame

String[] saldoCuentas = new String[100];
int posicion = 0;
for (Cliente cl : listadoClientes) {
    String texto = "Cliente id: " + cl.getId() + " saldo: " + cl.getCuenta().getSaldo();
    saldoCuentas[posicion] = texto;
    posicion++;
}

JList<String> listCuentas = new JList<>(saldoCuentas);
JScrollPane scrollPane = new JScrollPane(listCuentas);
/*scrollPane.setViewportView(listCuentas);
listCuentas.setLayoutOrientation(JList.VERTICAL);*/

/* JPanel panelSaldoCuentas = new JPanel();
panelSaldoCuentas.add(listCuentas);
panelSaldoCuentas.add(scrollPane);
frameCuentas.add(panelSaldoCuentas);*/

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
//frameCuentas.add(scrollPane);

Container contentPane = frameCuentas.getContentPane();
contentPane.add(scrollPane, BorderLayout.CENTER);

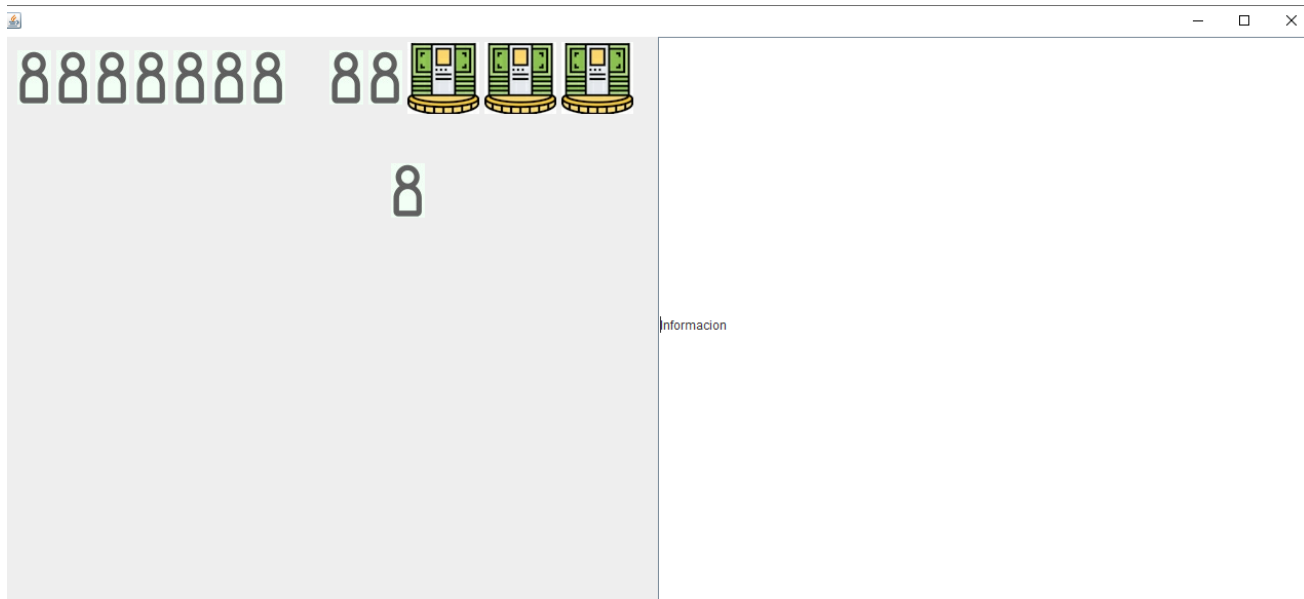
frameCuentas.getContentPane().validate();
frameCuentas.repaint();
frameCuentas.setVisible(true);

}

}
```


4. Comprobación de las cuentas bancarias e interfaz gráfica.
































Inicio de la simulación:



Como podemos ver hay la parte donde están las personas y los cajeros, y la segunda parte donde se despliega la información del cliente que está haciendo la transacción y cuanto es su saldo.

Fotos simulación:

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Comprobación cuentas bancarias:

Cuentas	
Cliente id: 0 saldo: 100	
Cliente id: 1 saldo: 100	
Cliente id: 2 saldo: 100	
Cliente id: 3 saldo: 100	
Cliente id: 4 saldo: 100	
Cliente id: 5 saldo: 100	
Cliente id: 6 saldo: 100	
Cliente id: 7 saldo: 100	
Cliente id: 8 saldo: 100	
Cliente id: 9 saldo: 100	
Cliente id: 10 saldo: 100	
Cliente id: 11 saldo: 100	
Cliente id: 12 saldo: 100	
Cliente id: 13 saldo: 100	
Cliente id: 14 saldo: 100	
Cliente id: 15 saldo: 100	
Cliente id: 16 saldo: 100	
Cliente id: 17 saldo: 100	
Cliente id: 18 saldo: 100	
Cliente id: 19 saldo: 100	
Cliente id: 20 saldo: 100	
Cliente id: 21 saldo: 100	
Cliente id: 22 saldo: 100	
Cliente id: 23 saldo: 100	
Cliente id: 24 saldo: 100	
Cliente id: 25 saldo: 100	
Cliente id: 26 saldo: 100	
Cliente id: 27 saldo: 100	
Cliente id: 28 saldo: 100	
Cliente id: 29 saldo: 100	
Cliente id: 30 saldo: 100	
Cliente id: 31 saldo: 100	
Cliente id: 32 saldo: 100	
Cliente id: 33 saldo: 100	
Cliente id: 34 saldo: 100	
Cliente id: 35 saldo: 100	
Cliente id: 36 saldo: 100	
Cliente id: 37 saldo: 100	

Cuentas	
Cliente id: 37 saldo: 100	
Cliente id: 38 saldo: 100	
Cliente id: 39 saldo: 100	
Cliente id: 40 saldo: 100	
Cliente id: 41 saldo: 100	
Cliente id: 42 saldo: 100	
Cliente id: 43 saldo: 100	
Cliente id: 44 saldo: 100	
Cliente id: 45 saldo: 100	
Cliente id: 46 saldo: 100	
Cliente id: 47 saldo: 100	
Cliente id: 48 saldo: 100	
Cliente id: 49 saldo: 100	
Cliente id: 50 saldo: 100	
Cliente id: 51 saldo: 100	
Cliente id: 52 saldo: 100	
Cliente id: 53 saldo: 100	
Cliente id: 54 saldo: 100	
Cliente id: 55 saldo: 100	
Cliente id: 56 saldo: 100	
Cliente id: 57 saldo: 100	
Cliente id: 58 saldo: 100	
Cliente id: 59 saldo: 100	
Cliente id: 60 saldo: 100	
Cliente id: 61 saldo: 100	
Cliente id: 62 saldo: 100	
Cliente id: 63 saldo: 100	
Cliente id: 64 saldo: 100	
Cliente id: 65 saldo: 100	
Cliente id: 66 saldo: 100	
Cliente id: 67 saldo: 100	
Cliente id: 68 saldo: 100	
Cliente id: 69 saldo: 100	
Cliente id: 70 saldo: 100	
Cliente id: 71 saldo: 100	
Cliente id: 72 saldo: 100	
Cliente id: 73 saldo: 100	
Cliente id: 74 saldo: 100	

Cuentas

Cuenta id: 62 saldo: 100
Cuenta id: 63 saldo: 100
Cuenta id: 64 saldo: 100
Cuenta id: 65 saldo: 100
Cuenta id: 66 saldo: 100
Cuenta id: 67 saldo: 100
Cuenta id: 68 saldo: 100
Cuenta id: 69 saldo: 100
Cuenta id: 70 saldo: 100
Cuenta id: 71 saldo: 100
Cuenta id: 72 saldo: 100
Cuenta id: 73 saldo: 100
Cuenta id: 74 saldo: 100
Cuenta id: 75 saldo: 100
Cuenta id: 76 saldo: 100
Cuenta id: 77 saldo: 100
Cuenta id: 78 saldo: 100
Cuenta id: 79 saldo: 100
Cuenta id: 80 saldo: 100
Cuenta id: 81 saldo: 100
Cuenta id: 82 saldo: 100
Cuenta id: 83 saldo: 100
Cuenta id: 84 saldo: 100
Cuenta id: 85 saldo: 100
Cuenta id: 86 saldo: 100
Cuenta id: 87 saldo: 100
Cuenta id: 88 saldo: 100
Cuenta id: 89 saldo: 100
Cuenta id: 90 saldo: 100
Cuenta id: 91 saldo: 100
Cuenta id: 92 saldo: 100
Cuenta id: 93 saldo: 100
Cuenta id: 94 saldo: 100
Cuenta id: 95 saldo: 100
Cuenta id: 96 saldo: 100
Cuenta id: 97 saldo: 100
Cuenta id: 98 saldo: 100
Cuenta id: 99 saldo: 100

RESULTADO(S) OBTENIDO(S):

- Profundizar en el tema del uso de Threads y la interfaz Runnable.
- Agrandar el conocimiento sobre como funcionan los hilos en java y como implementarlos.
- Resolver problemas de hilos con interfaces gráficas acorde a lo solicitado.

CONCLUSIONES:

En conclusión, esta prueba me ha servido mucho para tener un mayor conocimiento sobre los hilos en Java, ya que durante este tipo de ejercicios se aprende más que solamente leyendo información. Además, para resolver este ejercicio fue necesario pensar mucho con respecto a la resolución de la prueba práctica.

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

RECOMENDACIONES:

No existe ninguna recomendación.

Nombre de estudiante: Adolfo Sebastián Jara Gavilanes.

Firma de estudiante:

