
	Computación	Docente: Diego Quisi Peralta
	Programación Aplicada	Período Lectivo: Septiembre 2020 – Febrero 2021

		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES	
CARRERA: COMPUTACIÓN/INGENIERÍA DE SISTEMAS		ASIGNATURA: PROGRAMACIÓN APLICADA	
NRO. PROYECTO:	1.1	TÍTULO PROYECTO: Prueba Practica 2 Desarrollo e implementación de un sistema de simulación de acceso y atención bancaria	
OBJETIVO: Reforzar los conocimientos adquiridos en clase sobre la programación en Hilos en un contexto real.			
INSTRUCCIONES:		1. Revisar el contenido teórico y práctico del tema	
		2. Profundizar los conocimientos revisando los libros guías, los enlaces contenidos en los objetos de aprendizaje Java y la documentación disponible en fuentes académicas en línea.	
		3. Deberá desarrollar un sistema informático para la simulación y una interfaz gráfica.	
		4. Deberá generar un informe de la práctica en formato PDF y en conjunto con el código se debe subir al GitHub personal y AVAC.	
		5. Fecha de entrega: El sistema debe ser subido al git hasta 17 de enero del 2021 – 23:55.	
ACTIVIDADES POR DESARROLLAR			

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.

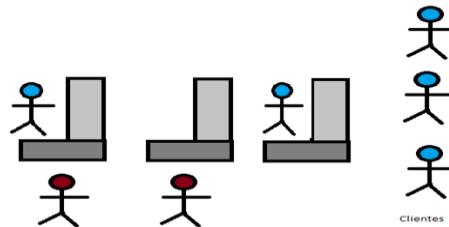
Se parte de una cuenta con 100 euros y se pueden tener procesos que ingresen 100 euros, 50 o 20. También se pueden tener procesos que retiran 100, 50 o 20 euros. Se desean tener los siguientes procesos:

- 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.

- 40 procesos que retiran 100
- 20 procesos que retiran 50
- 60 que retiran 20.

Ademas 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 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 numero 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%

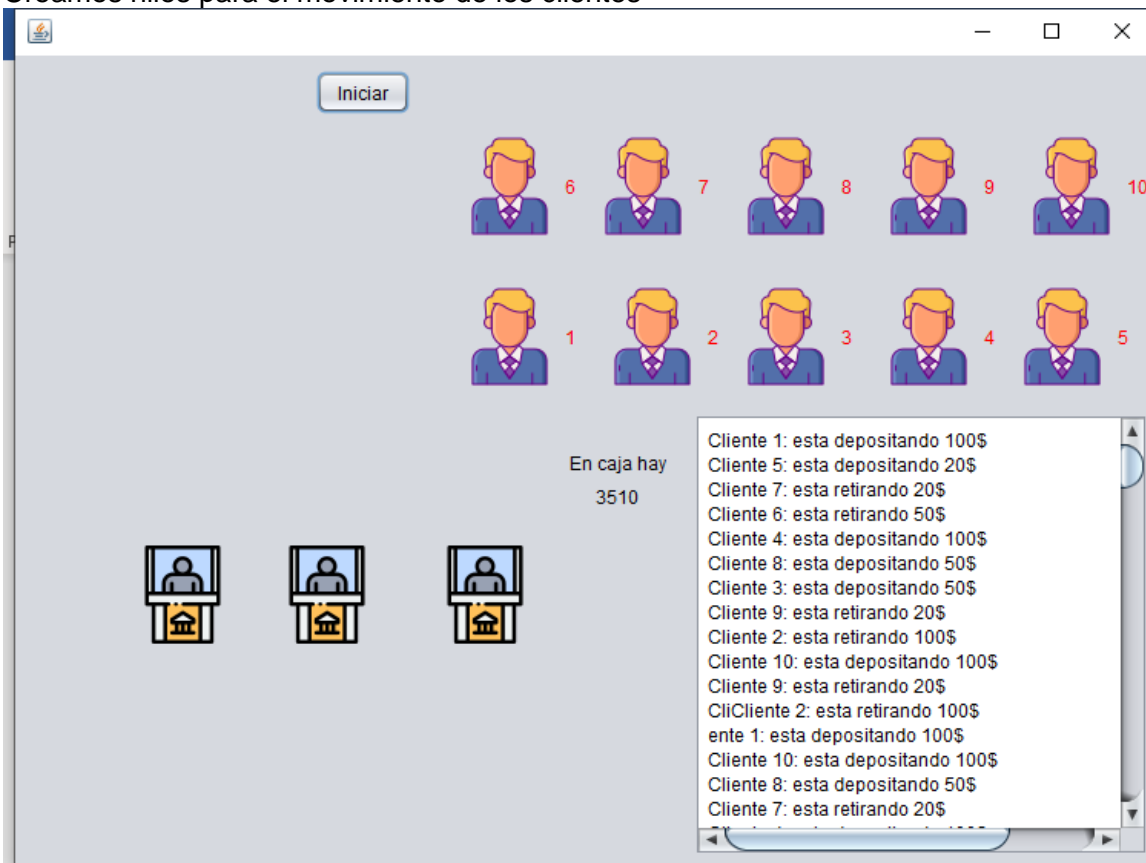
- Sincronización 10%
- Interfaz Gráfica de simulación 20%
- Informe: 10%

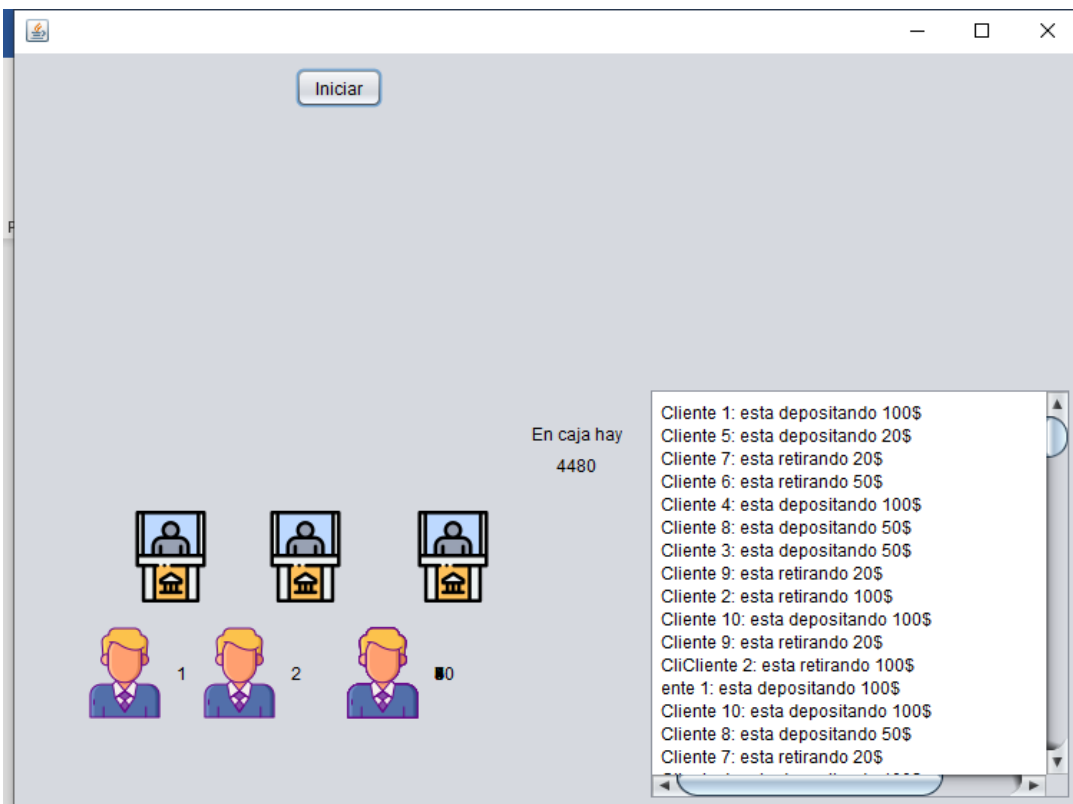
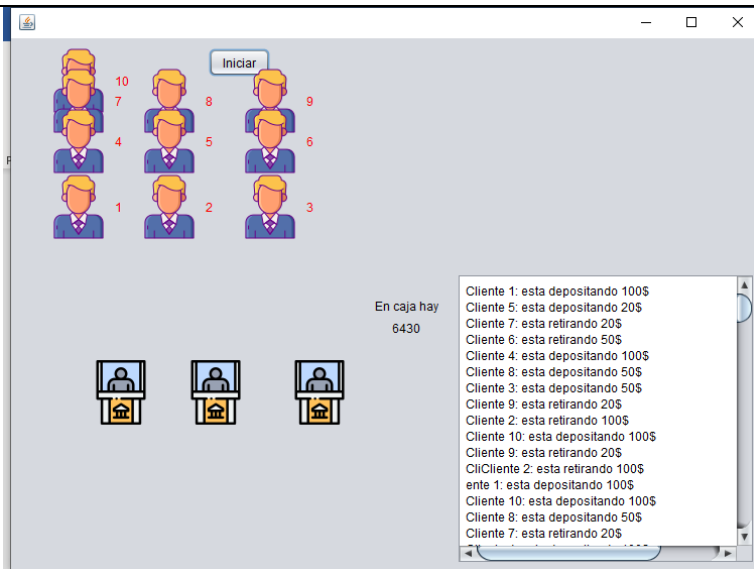
2. Informe de Actividades:

- Patrón de diseño aplicado

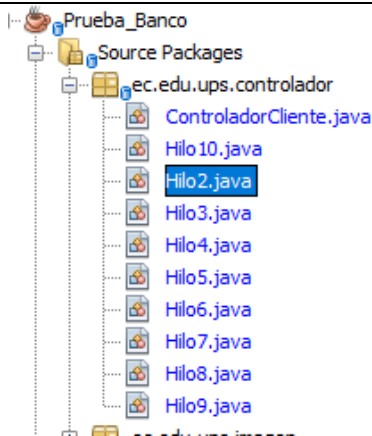
MVC

- Descripción de la solución y pasos seguidos.
Creamos hilos para el movimiento de los clientes





Para hacer el programa
Creamos hilos



Creamos el modelo con sus atributos

```

10  import javax.swing.JLabel;
11  import javax.swing.JTextArea;
12
13  /**
14   *
15   * @author Andres
16   */
17  public class Cliente implements Runnable {
18      private int id;
19      private int resultado;
20      private Thread t;
21      private JLabel cliente;
22      private JLabel derecha;
23      private JLabel izquierda;
24      private JLabel resultado1;
25      private String process;
26      private JTextArea area;
27
28      public Cliente(int id, JLabel derecha, JLabel izquierda, JLabel Cliente, JLabel resultado1, JTextArea area) {
29          this.id = id;
30          this.cliente = Cliente;
31          this.derecha = derecha;

```

Código

public class Cliente implements Runnable {

private int id;

private int resultado;

private Thread t;

private JLabel cliente;

private JLabel derecha;

private JLabel izquierda;

private JLabel resultado1;

private String process;

private JTextArea area;

public Cliente(int id, JLabel derecha, JLabel izquierda, JLabel Cliente, JLabel resultado1, JTextArea area) {

```
this.id = id;

this.cliente = Cliente;

this.derecha = derecha;

this.izquierda = izquierda;

this.resultado1 = resultado1;

this.area = area;

t= new Thread(this);

t.start();

}
```

@Override

```
public void run() {

    for (int i = 0; i < 4; i++) { // se controla el numero de veces que van a comer

        synchronized (this.izquierda) {

            synchronized (this.derecha) {

                ingresa();

            }

        }

    }

}

private void ingresa(){

    izquierda.setText("Depositando");

    izquierda.setForeground(Color.red);

    resultado = Integer.parseInt(resultado1.getText());

    resultado += 100;

    resultado1.setText(String.valueOf(resultado));

    process = "Cliente " + (id + 1) + ": " + "Esta depositando\n";

    area.append(process);

}
```

```
try {
    Thread.sleep(10);
} catch (InterruptedException e) {

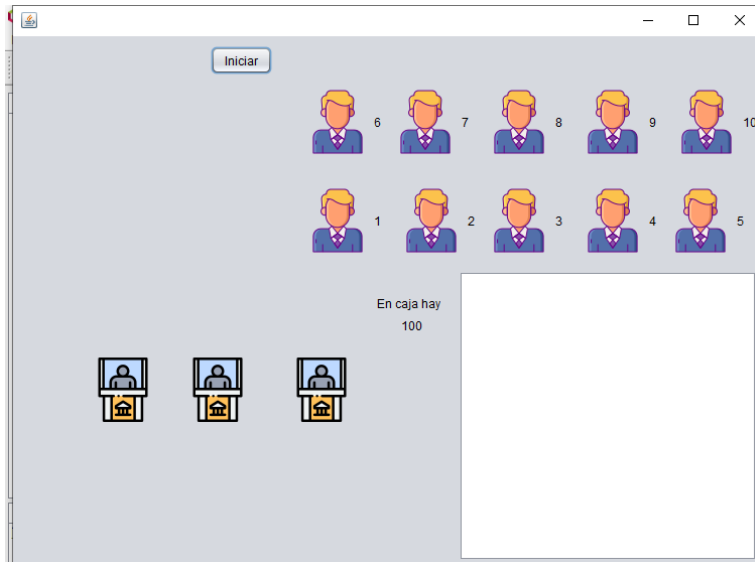
}

izquierda.setText("Sale");
izquierda.setForeground(Color.black);

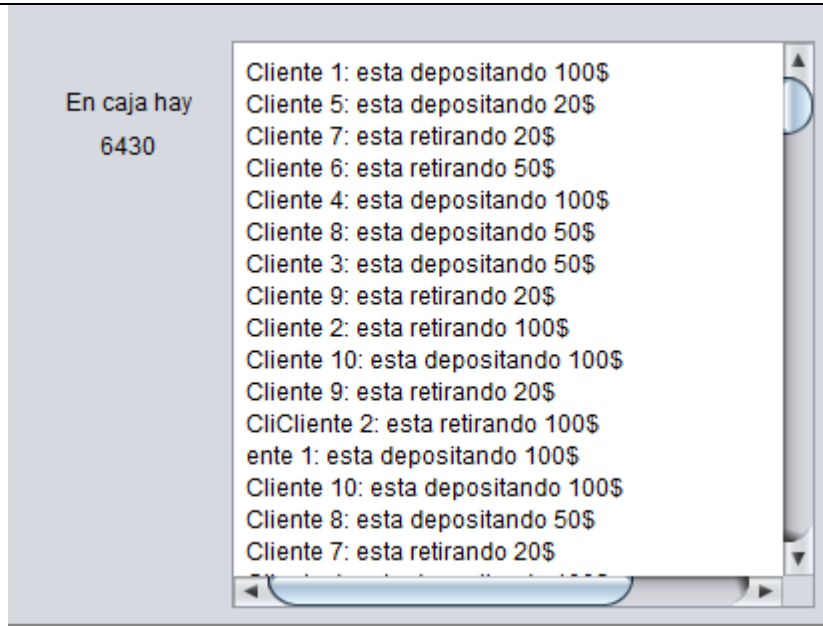
process = "Cliente" + (id + 1) + ": " + "Esta saliendo\n";
area.append(process);
}

}
```

Creemos la interfaz



- Comprobación de las cuentas bancarias e interfaz grafica.



- Conclusiones y recomendaciones.

En conclusión, aprendí a usar java y aplicar el patrón de diseño de una mejor manera

- Resultados.

Interpreta de forma correcta los algoritmos de programación y su aplicabilidad.

Identifica correctamente qué herramientas de programación se pueden aplicar.

RESULTADO(S) OBTENIDO(S):

- Interpreta de forma correcta los algoritmos de programación y su aplicabilidad.
- Identifica correctamente qué herramientas de programación se pueden aplicar.

CONCLUSIONES:

- Los estudiantes identifican las principales estructuras para la creación de sistemas informáticos.
- Los estudiantes implementan soluciones gráficas en sistemas.
- Los estudiantes están en la capacidad de implementar hilos.


RECOMENDACIONES:

- Revisar la información proporcionada por el docente previo a la práctica.
- Haber asistido a las sesiones de clase.
- **Consultar con el docente las dudas que puedan surgir al momento de realizar la prueba.**

BIBLIOGRAFIA:

[1]: <https://www.ups.edu.ec/evento?calendarBookingId=98892>

Nombre del Estudiante: Andres Alvarado

 UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR	Computación	Docente: Diego Quisi Peralta
	Programacion Aplicada	Período Lectivo: Septiembre 2020 – Febrero 2021

Firma:

