



Universidad de Guadalajara.

Centro Universitario de Ciencias Exactas e Ingenierías.

DIVISIÓN DE TECNOLOGÍAS PARA LA INTEGRACIÓN CIBER HUMANA.

DEPARTAMENTO DE CIENCIAS COMPUTACIONALES.

Practica6



**NOMBRE DE LOS ESTUDIANTES:**

Padilla Perez Jorge Daray

Juan Jesús Sámano Juárez

Ernesto Macias Flores

NOMBRE DE LA MATERIA: Seminario Sistemas Operativos

SECCION: D04

CICLO ESCOLAR: 2024-A

NOMBRE DEL PROFESOR: Julio Esteban

Esta actividad esta conformada por 3 códigos a continuación se mostrarán.

## Códigos:

### Main\_java

```
1  package practica6;
2
3  import java.awt.Image;
4  import java.awt.event.ActionEvent;
5  import java.awt.event.KeyEvent;
6  import java.awt.event.KeyListener;
7  import java.util.Random;
8  import javax.swing.ImageIcon;
9  import javax.swing.Timer;
10
11 public class Main extends javax.swing.JFrame implements KeyListener, teclas {
12     Random random = new Random();
13     private int randomProductor=0;
14     private int randomConsumidor=0;
15     private int producido=0;
16     private final Timer mTimer;
17     private int it=1;
18     private int copia=1;
19     private int itProductor=1;
20     private int itConsumidor=1;
21     ImageIcon imageIconMario= new ImageIcon("cliente_esperando.jpg");
22     Image imageMario = imageIconMario.getImage();
23     Image modMario = imageMario.getScaledInstance(100, 150, java.awt.Image.SCALE_SMOOTH);
24
25     ImageIcon imageIconMarioGigante= new ImageIcon("persona_gorda.jpg");
26     Image imageMarioGigante = imageIconMarioGigante.getImage();
27     Image modMarioGigante = imageMarioGigante.getScaledInstance(150, 225, java.awt.Image.SCALE_SMOOTH);
28
29     ImageIcon imageIconTinyMario= new ImageIcon("persona_saludable.jpeg");
30     Image imageTinyMario = imageIconTinyMario.getImage();
31     Image modTinyMario = imageTinyMario.getScaledInstance(150, 150, java.awt.Image.SCALE_SMOOTH);
```

importación de clases e imágenes

```

93 public Main() {
94     ImageIconBloque = new ImageIcon(modBloque);
95     ImageIconBloque2 = new ImageIcon(modBloque2);
96
97     ImageIconHongo = new ImageIcon(modHongo);
98     ImageIconFlorFuego = new ImageIcon(modFlorFuego);
99     ImageIconFlorHielo = new ImageIcon(modFlorHielo);
100     ImageIconHongoGigante = new ImageIcon(modHongoGigante);
101     ImageIconHongoAzul = new ImageIcon(modHongoAzul);
102     ImageIconHongoAbeja = new ImageIcon(modHongoAbeja);
103     ImageIconHuevoYoshi = new ImageIcon(modHuevoYoshi);
104     ImageIconMonedaMario = new ImageIcon(modMonedaMario);
105
106     ImageIconMario = new ImageIcon(modMario);
107     ImageIconTinyMario = new ImageIcon(modTinyMario);
108     ImageIconMarioGigante = new ImageIcon(modMarioGigante);
109     ImageIconMarioOro = new ImageIcon(modMarioOro);
110     ImageIconMarioFuego = new ImageIcon(modMarioFuego);
111     ImageIconMarioHielo = new ImageIcon(modMarioHielo);
112     ImageIconMarioAbeja = new ImageIcon(modMarioAbeja);
113     ImageIconMarioYoshi = new ImageIcon(modMarioYoshi);
114
115     initComponents();
116     mTimer = new Timer(2000, (ActionEvent e) -> {
117         iniciarCronometro();
118     });
119
120     addKeyListener(this);
121     setFocusable(true);
122 }

```

```

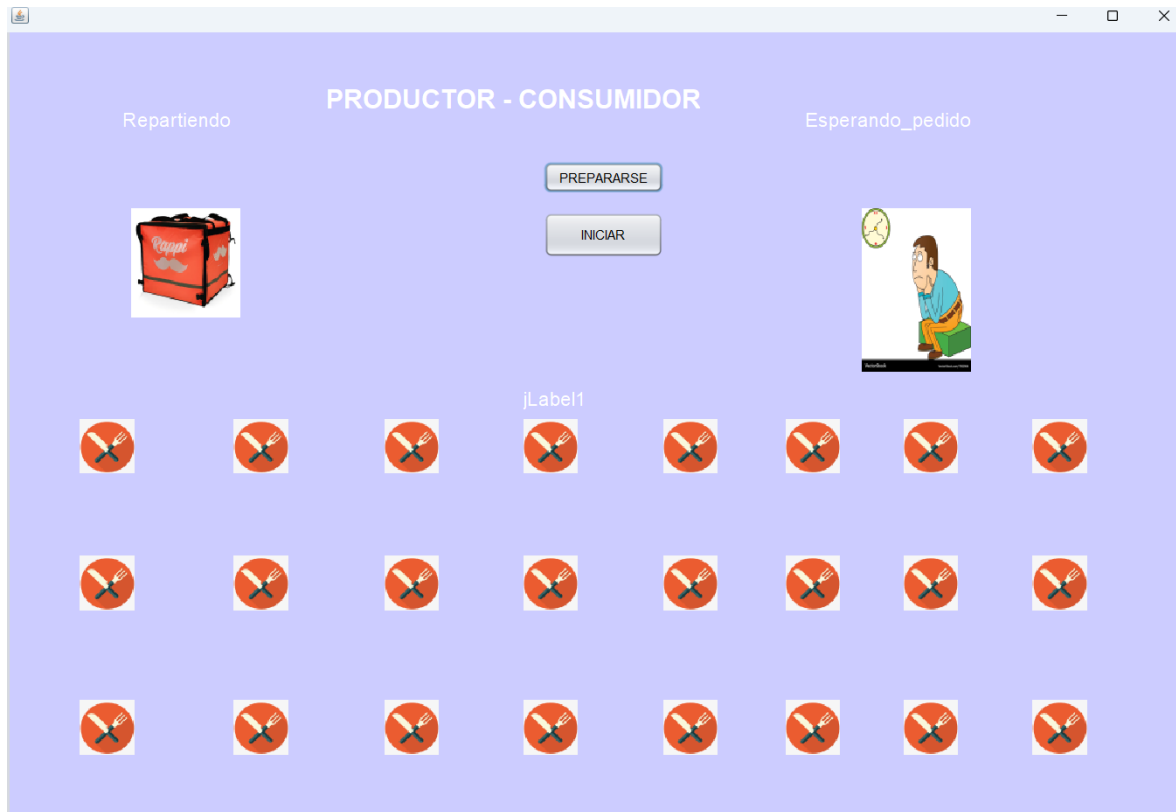
489 private void iniciarCronometro() {
490     lblTiempo.setText(String.valueOf(it));
491
492     //////////CONSUMIR//////////
493     if (randomProductor==0&&randomConsumidor==0&&it!=1){//CUANTO SE VA A CONSUMIR
494         lblEstado1.setText("Esperando_pedido_realizado");
495         lblEstado2.setText("CONSUMIENDO...");
496         randomConsumidor = random.nextInt((producido - 2) + 1) + 2;//numero aleatorio entre 1
497         System.out.println("A consumir: "+randomConsumidor);
498     }
499     if(randomConsumidor!=0&&randomProductor==0&&it!=1){//CONSUMIR CADA SEGUNDO
500         int aux2=0;
501         if(itConsumidor>24)
502             aux2=itConsumidor-24;
503         if(itConsumidor>48)
504             aux2=itConsumidor-48;
505         if(itConsumidor>72)
506             aux2=itConsumidor-72;
507         if(itConsumidor<=24)
508             aux2=itConsumidor;
509
510         switch (aux2) {
511             case 1:lbl12.setIcon(imageIconMarioFuego);
512                 lblBloque1.setIcon(imageIconBloque2);
513                 break;
514             case 2:lbl12.setIcon(imageIconMarioHielo);
515                 lblBloque2.setIcon(imageIconBloque2);
516                 break;
517             case 3:lbl12.setIcon(imageIconMarioOro);
518                 lblBloque3.setIcon(imageIconBloque2);

```

```
705 // Variables declaration - do not modify
706 private javax.swing.JButton btnIniciar;
707 private javax.swing.JButton btnPrepararse;
708 private javax.swing.JPanel jPanel1;
709 private javax.swing.JLabel lbl1;
710 private javax.swing.JLabel lbl2;
711 private javax.swing.JLabel lblBloque1;
712 private javax.swing.JLabel lblBloque10;
713 private javax.swing.JLabel lblBloque11;
714 private javax.swing.JLabel lblBloque12;
715 private javax.swing.JLabel lblBloque13;
716 private javax.swing.JLabel lblBloque14;
717 private javax.swing.JLabel lblBloque15;
718 private javax.swing.JLabel lblBloque16;
719 private javax.swing.JLabel lblBloque17;
720 private javax.swing.JLabel lblBloque18;
721 private javax.swing.JLabel lblBloque19;
722 private javax.swing.JLabel lblBloque2;
723 private javax.swing.JLabel lblBloque20;
724 private javax.swing.JLabel lblBloque21;
725 private javax.swing.JLabel lblBloque22;
726 private javax.swing.JLabel lblBloque23;
727 private javax.swing.JLabel lblBloque24;
728 private javax.swing.JLabel lblBloque3;
729 private javax.swing.JLabel lblBloque4;
730 private javax.swing.JLabel lblBloque5;
731 private javax.swing.JLabel lblBloque6;
732 private javax.swing.JLabel lblBloque7;
733 private javax.swing.JLabel lblBloque8;
734 private javax.swing.JLabel lblBloque9;
735 private javax.swing.JLabel lblEstado1;
```

## Capturas de ejecución:

Lo primero que se muestra al usuario es esta página, la cual mostrará el botón de iniciar para ejecutar el programa.





Luego tiene 2 estados en el que el repartidor espera pedidos mientras que el cliente consume.



Y el que el rappi esta repartiendo la comida mientras el cliente espera.

Referencias:

## Referencias

Disco Duro de Roer Ejemplo modelo Productor Consumidor en Java

<https://www.discoduroderoer.es/ejemplo-modelo-productor-consumidor-en-java/>

## **Conclusiones:**

### **Juan Jesús Sámano Juárez:**

Al realizar este proyecto fue un poco complicado el manejo de frames o también conocidos como marcos, después de haber solucionado este paso fue la elaboración de icono, los cuales en esta actividad son (gallina, zorro, semillas y bote) esto fue relativamente fácil de este punto la siguiente tarea era asignarles un evento de movimiento y arrastre para que el usuario pueda manejarlos y por último agregar las condicionales del juego.

### **Padilla Perez Jorge Daray:**

Para concluir con esta actividad interesante ya que es raro tener que simular que una computadora actual tarde tanto en hacer ciertos procesos y más cuando es algo tan sencillo, el tema de tener varias ventanas fue todo un reto que el compañero Ernesto soluciono, la comunicación en equipo fue importante para la realización de esta actividad, aprendí juntos a mis compañeros y espero que podamos seguir así de comunicados.

### **Ernesto Macias Flores:**

Durante la ejecución de este proyecto, nos encontramos con algunos desafíos al gestionar los marcos o frames. Una vez resueltos, nos enfocamos en el diseño de los íconos necesarios para la actividad: una gallina, un zorro, semillas y un bote. Este proceso resultó relativamente sencillo. A continuación, nos centramos en dotar a cada ícono con eventos de movimiento y capacidad de arrastre, brindando así al usuario el control necesario. Finalmente, completamos el proyecto integrando las condiciones del juego para una experiencia de usuario completa.