# GRADO EN INGENIERÍA INFORMÁTICA
# CURSO 2022/2023-CONVOCATORIA ORDINARIA

54286212W- Reimundez Cecilia, Darío

# Análisis de alto nivel

La finalidad del programa es simular el comportamiento de un restaurante a través de cocineros, clientes y el restaurante con sus respectivas cocinas. Se quiere representar como los clientes y cocineros, utilizando hilos y ejecutándose de forma simultánea, se implementan mediante diferentes mecanismos de sincronización para acceder a los diferentes medios compartidos.

Como ya veníamos diciendo, podemos identificar dos tipos de actores principales: los cocineros y los clientes. Ambos desempañando papeles muy diferentes en el restaurante, cuyos comportamientos comentaremos más adelante.

Por un lado, los clientes deben de hacer cola para poder usar la máquina de pedidos, haciendo la cola incluso fuera del restaurante, en caso de que la cola interior de veinte personas este llena. Una vez ordenado su pedido, ira a la cocina correspondiente a la comanda que ha enviado y esperará hasta que un cocinero lo atienda y haga todos los productos que envió. Finalmente, el cliente pagará en el instante que el cocinero esté listo para cobrarle. Cada cliente tendrá su propio identificador y solo tendrá acceso al restaurante.

Por otro lado, los cocineros desde el inicio del programa se reparten entre las tres diferentes cocinas, estarán esperando a la llegada de una comanda en sus respectivas cocinas. El cocinero que tome la comanda, ira al primer puesto que este libre, pasará del estado disponible a cocinando y comenzará a cocinar cada producto. Cuando esté listo, avisa a los clientes para que comprueben si se trata de su pedido, si así es, el cocinero ya estará listo para cobrarle. El cocinero cansado cada diez pedidos se va a tomar un descanso de diez segundos. Cada cocinero, también tiene su identificador único, además del tipo de cocina al que pertenece, también tiene acceso a la cocina y restaurante.

Aunque los clientes hacen toda su vida en el restaurante, ya sea, esperando a la cola, haciendo el pedido, esperando la comanda, … cuando tienen que pagar, van al puesto donde se ha realizado la comida, comprueban el importe a pagar y todo tipo de información que puede ser relevante, por ejemplo, para el pago. La zona de la cocina es una zona exclusiva para los cocineros, excepto por los puestos, que como acabamos de mencionar, son utilizados por los clientes para obtener información de la comanda.

Además, el restaurante (fuera de la cocina), en su mayoría ocupado por los clientes, podemos encontrar a los cocineros descansando, puesto que esta zona se encuentra fuera de la cocina.

Al inicio habrá 8 cocineros ya listos para recibir pedidos, mientras que, los clientes van llegando al restaurante en intervalos de tiempo aleatorio a lo largo de la simulación, llegando a un total de 20000 clientes.

Muy importante para comprobar el comportamiento del sistema es la interfaz gráfica de usuario, que brinda la posibilidad de parar y reanudar la simulación. Incluso, tendremos a nuestra disposición de un módulo de acceso remoto usando RMI, que nos dará información complementaria en tiempo real. También nos permitirá parar y reanudar la simulación a nuestro antojo.

# Diseño general del sistema

El sistema del restaurante cuenta con múltiples mecanismos de sincronización y comunicación para garantizar la coherencia de los recursos compartidos entre los clientes y cocineros. Esto permite que los actores actúen de forma ordenada y eviten posibles condiciones de carrera.

A fin de controlar la llegada de los clientes, se ha utilizado un semáforo, que medirá el número de personas en la cola de dentro, de forma que, si ya hay 20 clientes esperando dentro, los clientes que vayan llegando se quedaran fuera. Cuando un cliente ya ha realizado la comanda, libera una posición en el semáforo para dejar pasar al resto.

La máquina de pedidos, donde los clientes realizan las comandas, estará controlada por un "Lock", que solo permitirá su uso de forma individual y controlada. Por supuesto, cada vez que alguien entra o sale de las colas, se representa en la interfaz gráfica a través de la clase ListaClientes, cuyos métodos están declarados como "synchronized", asegurando la consistencia de los datos.

Con el fin de controlar a los cocineros y que estos queden a la espera de nuevas comandas, se ha utilizado un "Lock" y un "Condition", así el camarero hará "await" y saldrá del bucle en caso de que haya alguna comanda disponible. Para que este mecanismo de sincronización funcione, debemos tener al otro lado a un cliente que, cuando haga un pedido, añada la comanda y avise mediante un "signal" al cocinero que mas tiempo lleve esperando este esperando.

Bastante similar es el mecanismo que apreciamos para que el cliente pueda esperar a que su pedido esté listo. En este caso, utilizamos el mecanismo del monitor en la propia cocina. Creando métodos muy breves implementados con 'synchronized' que mantendrán al cliente esperando, gracias al "wait". Cuando el cocinero añada su ID en la lista de preparados, lo avisará mediante "notifyAll". Aunque se despierten a todos los hilos, solo saldrá del bucle aquel al que pertenezca el pedido.

Para el intercambio entre la comida preparada por el cocinero y el dinero del cliente se ha empleado un “exchanger”, que supone un punto de encuentro entre los dos hilos para intercambiar los datos de forma segura. Así, cuando el cliente ya sabe que esta su pedido, toma el “exchanger” común y queda a la espera de que el cocinero este preparado para el cambio.

El sistema cuenta con un monitor llamado "Parada" que garantiza que la pausa y reanudación del sistema se realicen de manera sincronizada y sin conflictos. Este monitor involucra a los dos tipos de actores involucrados en la simulación, evitando condiciones de carrera y asegurando una ejecución coherente del sistema.

Recordando que tenemos la interfaz gráfica “PantallaCliente” que muestra en tiempo real algunos datos relevantes de la ejecución del sistema, es indispensable mencionar que se usa como herramienta de comunicación RMI, por sus siglas “Remote Method Invocation”, que conecta el servidor representado por el “Restaurante” y los clientes (encargados de decidir si ejecutar métodos remotos en el servidor) representados por las clases “PantallaCliente” y “RemoteCliente”.

Adicionalmente, se ha implementado una clase llamada “Log” que se encarga de capturar de manera instantánea los sucesos y operaciones que tienen lugar en el sistema. Con el propósito de salvaguardar la coherencia de los registros y prevenir disputas al escribir en el archivo de registro, se ha empleado la palabra clave "synchronized" en los métodos pertinentes de la clase “Log”.

En resumen, el uso de mecanismos de sincronización, como locks, conditions y monitores, garantiza que los hilos accedan a los recursos compartidos de forma exclusiva, evitando condiciones de carrera y conflictos en los datos. Esto asegura la integridad de la información y evita resultados inconsistentes. Además, los mecanismos de comunicación, como los exchangers o las colas sincronizadas, permiten la interacción y la coordinación entre los distintos hilos del sistema. Facilitan la transferencia de datos y la sincronización de actividades, lo que es esencial para el correcto funcionamiento y flujo de trabajo del sistema.

# Descripción de las clases principales

Cabe comentar que el programa cuenta con cuatro ejecutables (main). Dos de ellos son jFrame: PantallaCliente y PantallaR (del restaurante) ,que no se deben de ejecutar directamente. En ambas tenemos getJTF, para obtener una lista de los JTextField que contiene.

Por un lado PantallaCliente, es un cliente en el RMI, para que al pulsar alguno de los botones se ejecute también en el restaurante, que tiene los botones que paran y reanudan la simulación. Por otro lado, PantallaR cuenta con getJButton para devolver los JButton que servirá para que desde restaurante (servidor) se puedan pulsar los botones con doClick.

Proyecto hace visible la interfaz gráfica "PantallaR, recopila los datos ingresados en los JTextField y los envía a cada puesto de cocina correspondiente, así como a las cocinas y al restaurante. Además, se encarga de crear los puestos y las cocinas necesarios, así como de generar y ejecutar los cocineros en cada cocina. También se encarga de crear los clientes y ejecutarlos.

**Para garantizar una correcta ejecución es importante ejecutar primero Proyecto y después RemoteCliente.**

RemoteCliente se encarga de revisar continuamente algunos valores para representarlos en PantallaCliente.

## Clases:

1.- Cliente: Actor en la simulación, encargado de representar a los clientes en un restaurante. Se ha implementado mediante hilos y su función, que se puede encontrar en su método run(), espera la cola, usa la máquina, espera el pedido y paga.

2.- Cocinero: Junto con cliente es otro actor involucrado en la simulación y representa a los cocineros. Se ha implementado con hilos y sus funciones básicas son: esperar comanda, cocinar, cobrar y cada diez pedidos completados descansar.

3.- Comanda: Clase para representar la comanda en la simulación. Como atributos tiene tipo (tipo de cocina), nProd (productos para hacer en la comanda) y cliente (id del cliente que ha hecho la comanda). Tiene los métodos para poder realizar “get” los valores mencionados.

4.- JTF: Clase para gestionar de forma sincronizada los valores que se guarden en los JTextField y que no almacenen listas de clientes ni cocineros. Por ello, sus métodos (get y set) son declarados como synchronized y solo tiene como atributo el JTextField.

5.- ListaCocineros: Clase para gestionar de forma sincronizada las listas de los cocineros que se guarden en los JTextField. Por ello, sus métodos (meter y sacar) son declarados como synchronized, teniendo como atributos el JTextField y ArrayList<Cocinero>.

6.- ListaClientes: Tiene la misma función e implementación que ListaCocineros pero para clientes y este como atributo tambien cuanta con un contador, que será útil para el RMI.

7.- Log: Clase encargada de crear y escribir el archivo evolucionRestaurante.txt. Tiene el método para escribir el fichero y para obtener la instancia del Log (get).

8.- Parada: El constructor actúa como un controlador en el sistema. Su función es supervisar el estado de pausa o reanudación del sistema. Si se activa el botón de pausa en la interfaz gráfica, la clase "Parada" detiene todas los clientes y cocineros y suspende la simulación. Por otro lado, al presionar el botón de reanudar, la clase "Parada" permite que los actores continúen su funcionamiento normal y el restaurante vuelva a sus actividades.

9.- Puesto: Clase que va a ser muy útil para poder almacenar la información de un puesto en concreto, almacenando en sus atributos los JTF que contienen los JTextField para actualizar el cocinero, pedido y pendiente. Además, guarda el coste (para que pueda tomarlo el cliente y saber cuánto tiene que pagar), el exchanger (para el pago), un ocupado (booleano que índice si está ocupado) y cliente (cliente que tiene asociado el pedido del puesto). De esto vienen acompañados los get y set de algunos atributos y el método limpiar, que deja a 0 o “” todos los valores.

10.- InterfazComun: Es la interfaz remota que define los métodos que los clientes pueden invocar en el servidor remoto. Todos los métodos deben de poder lanzar la excepción “RemoteException”.

11.- Cocina: Es la zona donde se producen todas las acciones de los cocineros.Cuenta con muchos atributos y métodos, al igual que restaurante, por ser la clase usada por los cocineros para realizar sus acciones:

**Atributos:**

- pedido/hayComanda: Lock/Condition que permiten establecer secciones críticas controladas por el lock y permiten a los hilos esperar y ser notificados de si hay nuevas comandas.
- comandas: ArrayList de comanda que almacena las comandas por cocinar por los clientes, así , cuando un cocinero va a tomar una nueva para cocinar, elige y borra de la lista la posición cero, es decir, la que más tiempo lleva guardada.
- puesto: ArrayList de puesto, muy útil para generalizar los puestos, ya que según la cocina, habrá más o menos cocineros, y por ende, más o menos puestos.
- pf/ impf: Representa los pedidos e importe finales respectivamente dentro de la propia cocina.
- listas: Array de String que contiene los id de los clientes cuyos pedidos han sido terminados.
- parada: Clase Parada para poder controlar la pausa y continuación de la simulación.

**Métodos:**

- Cocina(…): Constructor de la clase.
- esperarNueva(): Hace que el cocinero que lo ejecuta, quede esperando hasta que haya alguna comanda disponible. Para ello usa el Lock/Condition.

- cocinar(Comanda comanda, String id): Toma el primer puesto libre, cocina todos los productos de la comanda y llama al método avisar. Actualiza los valores JTF del puesto.
- avisar(Puesto puesto): (Método privado) Añade en listas el id del pedido finalizado y realiza notifyAll() para avisar a los clientes.
- cobrar(Puesto puesto) : Toma el exchanger que también usara el cliente e intercambian los valores de comida e importe. Limpia los valores de JTF del puesto.
- añadirNueva(Comanda comanda): Método llamado desde el restaurante por un cliente, añade en las lista de comandas la nueva comanda. Una vez más, se hace el aviso usando el mecanismo de Lock/Condition.
- estaLista(String id): Llama al método contiene(id), busca el puesto con su id y devuelve el puesto.
- contiene(): (Método privado) Comprueba si el id del cliente esta en listas. Si lo esta, borra el id, sino queda esperando por el wait() hasta que sea avisado por un cocinero.

12.- Restaurante: Es la zona donde se producen todas las acciones de Cliente. Por ello es, junto a la cocina, el núcleo del programa. Cuenta con muchos atributos y métodos que le dan vida, y por ello, se procede a explicar en detalle cada uno:

**Atributos:**

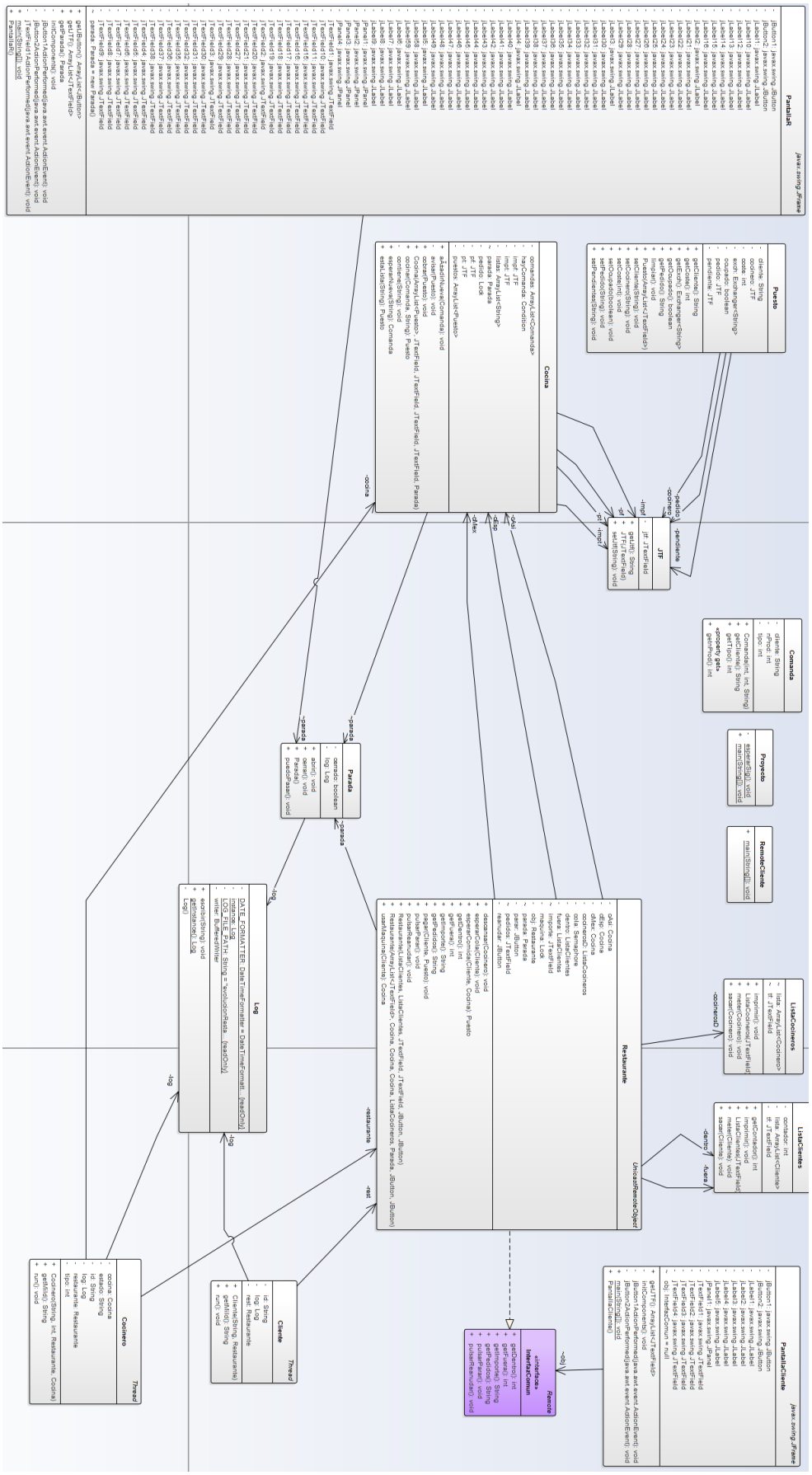
- fuera: Se trata de una ListaClientes que se encargara de guardar y actualizar todos los nuevos clientes que tengan que esperar la cola fuera del restaurante.
- dentro: Se trata de una ListaClientes que se encargará de guardar y actualizar todos los clientes que tengan que esperar la cola dentro del restaurante.
- cola: Es un semáforo cuya finalidad es poder controlar los clientes que hay en la cola de espera de dentro. Si dentro esta lleno, los clientes que lleguen quedarán esperando y se mantendrán fuera.
- maquina: Representa la máquina de pedidos y se trata de un Lock, que, de forma simultánea, solo permite a un cliente usarlo.
- cEsp: Representa la cocina española y nos facilita la comunicación con esa cocina, pudiendo enviar y recibir información de las comandas.
- cMex: Representa la cocina mexicana y es muy parecida a cEsp.
- cAsi: Representa la cocina asiática y es muy parecida a cEsp.
- cocinerosD: Es una ListaCocineros que guardara y actualizara a los cocineros que vayan y dejen de descansar.
- parada: Almacena la instancia de Parada y se usa después de cada sleep, para saber si se puede continuar o no con la simulación.
- obj: Instancia de la clase Restaurante que representa el objeto remoto que sirve para establecer la comunicación con el cliente.
- pedidos: Guarda el JTextField que marca los pedidos en conjunto de todas las cocinas y, en esta clase, se usa para poder enviar información a RMI.
- importe: Guarda el JTextField que marca el importe recaudado en conjunto de todas las cocinas y, en esta clase, se usa para poder enviar información a RMI.
- parar: Guarda el JButton para parar la simulación y su finalidad también está relacionada con el RMI.

- reanudar: Guarda el JButton para reanudar la simulación y su finalidad también esta relacionada con el RMI.

**Métodos:**

- Restaurante(…): Es el constructor que se emplea para crear el restaurante.
- Restaurante(ArrayList<JTextField> jtf, Cocina cEsp, Cocina cMex, Cocina cAsi, ListaCocineros cocinerosD, Parada parada, JButton parar, JButton reanudar): Constructor que se utilizara para crear el objeto remoto. Contiene solo los atributos necesarios para la implementación del RMI.
- esperarCola(): Se encarga de utilizar fuera, dentro, y máquina para asegurarse de controlar el aforo de la cola dentro y que solo termine la función cuando el cliente puede usar la máquina.
- usarMaquina(Cliente c): Crea la comanda y la envia a una cocina aleatoria. Como es de esperar, cuando termina lo saca de la cola de espera y el método devuelve la cocina en la que esta su comanda.
- esperarComida(Cliente c, Cocina cocina): Se encarga de encontrar el puesto donde está preparándose su comanda (cuando lo este), termina cuando ya ha encontrado su comanda lista y devuelve el puesto en el que esta su comida.
- pagar(Cliente c, Puesto puesto): Toma el importe y el exchanger del puesto y se realiza el intercambio con el cocinero.
- descansar(Cocinero c): Actualiza la lista de cocinerosD descansando y controla el tiempo de descanso.
- getFuera()/getDentro(): (RMI) Devuelve el contador de fuera y dentro respectivamente.
- getPedidos()/getImporte(): (RMI) Devuelve texto almacenado en el JTextField de pedidos e importe total respectivamente.
- pulsarParar()/pulsarReanudar(): (RMI) Se encarga de hacer “doClick” en los botones de parar y reanudar respectivamente.

# Diagrama de clases



# Código del proyecto

**Clase Cliente:**

```
package com.mycompany.proyecto;

public class Cliente extends Thread{

    private String id;
    private Restaurante rest;
    private Log log;

    public Cliente(String id, Restaurante rest)
    {
        this.id=id;
        this.rest=rest;
        this.log=Log.getInstance();
    }

    public void run()
    {
        try{
            log.escribir("Cliente "+id+" esta ejecutandose");
            log.escribir("Cliente "+id+" esta esperando la cola");
            rest.esperarCola(this);
            log.escribir("Cliente "+id+" esta usando maquina de pedidos");
            Cocina cocina=rest.usarMaquina(this);
            log.escribir("Cliente "+id+" esta esperando la comdia");
            Puesto puesto=rest.esperarComida(this,cocina);
            log.escribir("Cliente "+id+" esta pagando");
            rest.pagar(this,puesto);
            log.escribir("Cliente "+id+" ha terminado su ejecucion");
```

```
        }catch(Exception e){}
    }

    public String getMiId()
    {
        return id;
    }

}
```

**Clase Cocina:**

```
package com.mycompany.proyecto;

import static java.lang.Thread.sleep;
import java.rmi.Naming;
import java.rmi.RemoteException;
import java.util.ArrayList;
import java.util.concurrent.Exchanger;
import java.util.concurrent.locks.Condition;
import java.util.concurrent.locks.Lock;
import java.util.concurrent.locks.ReentrantLock;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JTextField;

public class Cocina {

    private Lock pedido;
    private Condition hayComanda;
    private ArrayList<Comanda> comandas;

    //Puestos
```

```
    private ArrayList<Puesto> puestos;

    //Pedidos e importe totales de esta cocina
    private JTF pt;
    private JTF impt;

    //Pedidos e importe de todas las cocinas
    private JTF pf;
    private JTF impf;

    //Lista con los id de los clientes cuyas comandas ya se han terminado
    private ArrayList<String> listas;

    //Monitor para controlar las paradas
    Parada parada;

    public Cocina(ArrayList<Puesto> puestos, JTextField pt, JTextField impt, JTextField
pf, JTextField impf, Parada parada)
    {
        pedido=new ReentrantLock();
        hayComanda=pedido.newCondition();
        comandas=new ArrayList<>();

        this.puestos=puestos;

        this.pt=new JTF(pt);
        this.impt=new JTF(impt);

        this.pf=new JTF(pf);
        this.impf=new JTF(impf);

        listas=new ArrayList<>();
```

```
    this.parada=parada;
}

public Comanda esperarNueva(String id)
{
    Comanda nueva = null;
    try{
        pedido.lock();
        while(comandas.isEmpty())
        {
            hayComanda.await();
        }
        nueva=comandas.remove(0);
    }catch(Exception e){
    }finally{
        pedido.unlock();
    }
    return nueva;
}

public Puesto cocinar(Comanda comanda, String id)
{
    //Locizamos un puesto que no se este usando
    Puesto puesto=null;
    int i=0;
    boolean encontrado=false;
    while (!encontrado)
    {
        puesto=puestos.get(i);
        if (!puestos.get(i).getOcupado())
```

```
    {
        encontrado=true;
    }else
    {
        i=(i+1)%puestos.size();
    }
}
//Ocupamos el puesto
puesto.setOcupado(true);
//Guardamos el numero de productos pendientes
int nProd=comanda.getnProd();
//Damos valores a los jtf
puesto.setCocinero(id);
puesto.setPedido(comanda.getCliente());
puesto.setPendientes(Integer.toString(nProd));
puesto.setCliente(comanda.getCliente());

//Mientras hay productos cocinamos
int importe=0;
while (nProd>0)
{
    try {
        sleep(1000+(int)(1000*Math.random()));
        parada.puedoPasar();
        nProd--;
        puesto.setPendientes(Integer.toString(nProd));
        importe+=4+(int)(5*Math.random());
    } catch (InterruptedException ex) {}
}
//Indicamos el coste correspondiente a la comanda
puesto.setPendientes("PAGANDO "+importe+"€");
```

```
    puesto.setCoste(importe);

    //Avisamos al cliente de que ya tenemos su pedido
    avisar(puesto);

    return puesto;
}

private synchronized void avisar(Puesto puesto)
{
    listas.add(puesto.getPedido());
    notifyAll();
}

public void cobrar(Puesto puesto)
{
    Exchanger<String> e=puesto.getExch();
    String cambio="<Comida>";
    try {
        sleep(1000);
        parada.puedoPasar();
        cambio=e.exchange("<Comida>");
    } catch (InterruptedException ex) {}
    //Actualizamos los pedidos totales de la cocina
    int antes=Integer.parseInt(pt.getJtf());
    pt.setJtf(Integer.toString(antes+1));

    //Actualizamos importe total de la cocina
    antes=Integer.parseInt(impt.getJtf());
    impt.setJtf(Integer.toString(antes+Integer.parseInt(cambio)));
```

```
    //Actualizamos los pedidos totales de todas las cocinas
    antes=Integer.parseInt(pf.getJtf());
    int pedidos=antes+1;
    pf.setJtf(Integer.toString(antes+1));

    //Actualizamos importe total de todas las cocinas
    antes=Integer.parseInt(impf.getJtf());
    int importe=antes+Integer.parseInt(cambio);
    impf.setJtf(Integer.toString(importe));

    //Limpiamos y dejamos libre el puesto
    puesto.limpiar();
}

public void añadirNueva(Comanda comanda)
{
    comandas.add(comanda);
    try
    {
        pedido.lock();
        hayComanda.signal();
    }catch(Exception e){
    }finally{
        pedido.unlock();
    }
}

public Puesto estaLista(String id)
{
    contiene(id);
    boolean encontrado=false;
```

```
        int i=0;
        while (!encontrado)
        {
            if (puestos.get(i).getCliente()==id)
            {
                encontrado=true;
            }else
            {
                i=(i+1)%puestos.size();
            }
        }
        return puestos.get(i);
    }

    private synchronized void contiene(String id)
    {
        while (!listas.contains(id))
        {
            try {
                wait();
            } catch (InterruptedException ex) {}
        }
        listas.remove(id);
    }
}
```

**Clase Cocinero:**

```
ackage com.mycompany.proyecto;

public class Cocinero extends Thread{

    private String id;
```

```
private int tipo;
private Restaurante restaurante;
private Cocina cocina;
private String estado;
private Log log;

public Cocinero(String id, int tipo, Restaurante restaurante, Cocina cocina)
{
   this.id=id;
   this.tipo=tipo;
   this.restaurante=restaurante;
   this.cocina=cocina;
   estado="DISPONIBLE";
   this.log=Log.getInstance();
}

public void run()
{
   log.escribir("Cocinero "+id+" esta ejecutandose");
   while(true)
   {
      for (int i=0;i<10;i++)
      {
         log.escribir("Cocinero "+id+" esta DISPONIBLE");
         log.escribir("Cocinero "+id+" esta esperando nueva comanda");
         Comanda nueva=cocina.esperarNueva(this.id);
         estado="COCINANDO";
         log.escribir("Cocinero "+id+" esta COCINANDO");
         Puesto puesto=cocina.cocinar(nueva,this.id);
         log.escribir("Cocinero "+id+" cobra el pedido");
         cocina.cobrar(puesto);
```

```
            estado="DISPONIBLE";
        }
        estado="DESCANSANDO";
        log.escribir("Cocinero "+id+" esta DESCANSANDO");
        restaurante.descansar(this);
      }
   }

   public String getMiId()
   {
      return id;
   }

}
```

**Clase Comanda:**

```
package com.mycompany.proyecto;

public class Comanda {

   private int tipo;
   private int nProd;
   private String cliente;



   public Comanda (int tipo, int nProd, String cliente)
   {
      this.tipo=tipo;
      this.nProd=nProd;
      this.cliente=cliente;
   }
```

```
    public int getTipo()
    {
        return tipo;
    }

    public int getnProd()
    {
        return nProd;
    }

    public String getCliente()
    {
        return cliente;
    }
}
```

**Clase InterfazComun:**

```
package com.mycompany.proyecto;

import java.rmi.Remote;
import java.rmi.RemoteException;

public interface InterfazComun extends Remote{

    public int getFuera() throws RemoteException;
    public int getDentro() throws RemoteException;
    public String getPedidos() throws RemoteException;
    public String getImporte() throws RemoteException;
    public void pulsarParar() throws RemoteException;
    public void pulsarReanudar() throws RemoteException;
}
```

**Clase JTF:**

```
package com.mycompany.proyecto;

import javax.swing.JTextField;

public class JTF {

    private JTextField jtf;

    public JTF(JTextField jtf)
    {
        this.jtf=jtf;
    }

    public synchronized void setJtf(String t)
    {
        jtf.setText(t);
    }

    public synchronized String getJtf()
    {
        return jtf.getText();
    }
}
```

**Clase ListaClientes:**

```
package com.mycompany.proyecto;

import java.util.ArrayList;
import javax.swing.JTextField;

public class ListaClientes {
```

```
private ArrayList<Cliente> lista;
private JTextField tf;
private int contador;

public ListaClientes(JTextField tf)
{
   lista=new ArrayList<>();
   this.tf=tf;
}

public synchronized void meter(Cliente h)
{
   lista.add(h);
   contador++;
   imprimir();
}

public synchronized void sacar(Cliente h)
{
   lista.remove(h);
   contador--;
   imprimir();
}

public int getContador()
{
   return this.contador;
}

public void imprimir()
{
```

```
        String contenido="";
        for(int i=0; i<lista.size(); i++)
        {
          contenido=contenido+lista.get(i).getMiId()+" ";
        }
        tf.setText(contenido);
    }
}
```

**Clase ListaCocineros:**

```
package com.mycompany.proyecto;

import java.util.ArrayList;
import javax.swing.JTextField;

public class ListaCocineros {

    ArrayList<Cocinero> lista;
    JTextField tf;

    public ListaCocineros(JTextField tf)
    {
        lista=new ArrayList<>();
        this.tf=tf;
    }

    public synchronized void meter(Cocinero c)
    {
        lista.add(c);
        imprimir();
    }
```

```
    public synchronized void sacar(Cocinero c)
    {
        lista.remove(c);
        imprimir();
    }

    public void imprimir()
    {
        String contenido="";
        for(int i=0; i<lista.size(); i++)
        {
          contenido=contenido+lista.get(i).getMiId()+" ";
        }
        tf.setText(contenido);
    }
}
```

**Clase Log:**

```
package com.mycompany.proyecto;

import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;

public class Log {

    private static final String LOG_FILE_PATH = "evolucionRestaurante.txt";
    private static final DateTimeFormatter DATE_FORMATTER =
DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");
    private static Log instance;
    private BufferedWriter writer;
```

```
    private Log() {
        try {
            writer = new BufferedWriter(new FileWriter(LOG_FILE_PATH, true));
        } catch (IOException e) {}
    }

    public synchronized void escribir(String event) {
        try {
            //Le añadimos la fecha y hora al evento
            LocalDateTime currentTime = LocalDateTime.now();
            String formattedTime = currentTime.format(DATE_FORMATTER);
            String logMessage = formattedTime + " - " + event;
            //Escribimos en el archivo log
            writer.write(logMessage);
            writer.newLine();
            writer.flush();
        } catch (IOException e) {
        }
    }

    public static synchronized Log getInstance() {
        if (instance == null) {
            instance = new Log();
        }
        return instance;
    }
}
```

**Clase PantallaCliente:**

```
package com.mycompany.proyecto;
```

```
import java.rmi.Naming;
import java.rmi.RemoteException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JTextField;



public class PantallaCliente extends javax.swing.JFrame {

    //RMI cliente
    InterfazComun obj= null;

    public PantallaCliente() {
        initComponents();
        boolean encontrado=false;

        //Conectado al servidor
        try {
            obj=(InterfazComun) Naming.lookup("//localhost/ObjetoC");
            encontrado=true;
        }catch (Exception e){}

    }

    public ArrayList<JTextField> getJTF()
    {
        ArrayList<JTextField> textFields = new ArrayList<>(Arrays.asList(jTextField1,
jTextField2, jTextField3, jTextField4));
        return textFields;
    }
```

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

    jLabel1 = new javax.swing.JLabel();
    jPanel1 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jTextField1 = new javax.swing.JTextField();
    jTextField2 = new javax.swing.JTextField();
    jTextField3 = new javax.swing.JTextField();
    jTextField4 = new javax.swing.JTextField();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
    jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel1.setText("RESTAURANTE (INFORMACIÓN REMOTA)");

    jPanel1.setBackground(new java.awt.Color(204, 204, 204));
    jPanel1.setForeground(new java.awt.Color(204, 204, 204));

    jLabel2.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
    jLabel2.setText("Clientes esperando fuera");

    jLabel3.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
```

```
        jLabel3.setText("Clientes esperando dentro");

        jLabel4.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
        jLabel4.setText("Total de pedidos completados");

        jLabel5.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
        jLabel5.setText("Cantidad recaudada");

        jTextField1.setText("0");

        jTextField2.setText("0");

        jTextField3.setText("0");

        jTextField4.setText("0");

        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(133, 133, 133)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                    .addComponent(jLabel2)
                    .addComponent(jLabel3)
                    .addComponent(jLabel4)
                    .addComponent(jLabel5))
                .addGap(73, 73, 73)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jTextField1, javax.swing.GroupLayout.DEFAULT_SIZE, 100, Short.MAX_VALUE)
                .addComponent(jTextField2)
                .addComponent(jTextField3)
                .addComponent(jTextField4))
            .addGap(122, 122, 122))
        );
        jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(33, 33, 33)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel2)
                .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel3)
                .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel4)
```

```
                .addComponent(jTextField3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(16, 16, 16)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                .addComponent(jLabel5)
                .addComponent(jTextField4,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap(47, Short.MAX_VALUE))
        );


        jButton1.setText("Parar");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });


        jButton2.setText("Reanudar");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });


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

```
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(33, 33, 33)
                        .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(258, 258, 258)
                        .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
                        .addComponent(jButton2,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addGap(0, 43, Short.MAX_VALUE)))
            .addContainerGap())
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(44, 44, 44)
```

```
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(34, 34, 34)
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELIN
E)
                .addComponent(jButton1)
                .addComponent(jButton2))
            .addContainerGap(16, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            // TODO add your handling code here:
            obj.pulsarParar();
        } catch (RemoteException ex) {}
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            // TODO add your handling code here:
            obj.pulsarReanudar();
        } catch (RemoteException ex) {}
    }

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

```
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new PantallaCliente().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JTextField jTextField3;
    private javax.swing.JTextField jTextField4;
    // End of variables declaration
}
```

**Clase PantallaR:**

```
package com.mycompany.proyecto;

import java.util.ArrayList;
import java.util.Arrays;
import javax.swing.JButton;
import javax.swing.JTextField;

public class PantallaR extends javax.swing.JFrame {
```

```
    //Para poder usar los botones de pausa y reanudar
    Parada parada=new Parada();

    public PantallaR() {
        initComponents();
    }

    public Parada getParada()
    {
        return parada;
    }

    public ArrayList<JTextField> getJTF()
    {
        return new ArrayList<>(Arrays.asList(jTextField1, jTextField2, jTextField3,
jTextField4, jTextField5, jTextField6, jTextField7, jTextField8, jTextField9,
jTextField10, jTextField11, jTextField15, jTextField16, jTextField17, jTextField18,
jTextField19, jTextField20, jTextField21, jTextField22, jTextField23, jTextField24,
jTextField25, jTextField26, jTextField27, jTextField28, jTextField29, jTextField30,
jTextField31,jTextField32, jTextField33, jTextField34, jTextField35, jTextField36,
jTextField37, jTextField38));
    }

    public ArrayList<JButton> getJButton()
    {
        return new ArrayList<>(Arrays.asList(jButton1,jButton2));
    }

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

```
jLabel1 = new javax.swing.JLabel();
jLabel2 = new javax.swing.JLabel();
jTextField1 = new javax.swing.JTextField();
jTextField2 = new javax.swing.JTextField();
jLabel3 = new javax.swing.JLabel();
jPanel1 = new javax.swing.JPanel();
jLabel5 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
jLabel7 = new javax.swing.JLabel();
jLabel8 = new javax.swing.JLabel();
jTextField3 = new javax.swing.JTextField();
jTextField4 = new javax.swing.JTextField();
jTextField5 = new javax.swing.JTextField();
jLabel9 = new javax.swing.JLabel();
jLabel10 = new javax.swing.JLabel();
jLabel11 = new javax.swing.JLabel();
jLabel12 = new javax.swing.JLabel();
jTextField6 = new javax.swing.JTextField();
jTextField7 = new javax.swing.JTextField();
jTextField8 = new javax.swing.JTextField();
jLabel13 = new javax.swing.JLabel();
jLabel14 = new javax.swing.JLabel();
jLabel15 = new javax.swing.JLabel();
jLabel16 = new javax.swing.JLabel();
jTextField9 = new javax.swing.JTextField();
jTextField10 = new javax.swing.JTextField();
jTextField11 = new javax.swing.JTextField();
jLabel22 = new javax.swing.JLabel();
jTextField15 = new javax.swing.JTextField();
jLabel23 = new javax.swing.JLabel();
```

```
jTextField16 = new javax.swing.JTextField();
jPanel2 = new javax.swing.JPanel();
jLabel21 = new javax.swing.JLabel();
jLabel27 = new javax.swing.JLabel();
jLabel25 = new javax.swing.JLabel();
jTextField18 = new javax.swing.JTextField();
jLabel28 = new javax.swing.JLabel();
jLabel29 = new javax.swing.JLabel();
jTextField20 = new javax.swing.JTextField();
jLabel30 = new javax.swing.JLabel();
jTextField21 = new javax.swing.JTextField();
jTextField22 = new javax.swing.JTextField();
jLabel32 = new javax.swing.JLabel();
jTextField23 = new javax.swing.JTextField();
jLabel33 = new javax.swing.JLabel();
jTextField24 = new javax.swing.JTextField();
jLabel31 = new javax.swing.JLabel();
jTextField19 = new javax.swing.JTextField();
jLabel26 = new javax.swing.JLabel();
jLabel24 = new javax.swing.JLabel();
jTextField17 = new javax.swing.JTextField();
jPanel3 = new javax.swing.JPanel();
jLabel34 = new javax.swing.JLabel();
jLabel35 = new javax.swing.JLabel();
jLabel36 = new javax.swing.JLabel();
jLabel37 = new javax.swing.JLabel();
jLabel38 = new javax.swing.JLabel();
jTextField25 = new javax.swing.JTextField();
jTextField26 = new javax.swing.JTextField();
jTextField27 = new javax.swing.JTextField();
jLabel39 = new javax.swing.JLabel();
```

```
jLabel40 = new javax.swing.JLabel();
jLabel41 = new javax.swing.JLabel();
jLabel42 = new javax.swing.JLabel();
jTextField28 = new javax.swing.JTextField();
jTextField29 = new javax.swing.JTextField();
jTextField30 = new javax.swing.JTextField();
jLabel43 = new javax.swing.JLabel();
jLabel44 = new javax.swing.JLabel();
jLabel45 = new javax.swing.JLabel();
jLabel46 = new javax.swing.JLabel();
jTextField31 = new javax.swing.JTextField();
jTextField32 = new javax.swing.JTextField();
jTextField33 = new javax.swing.JTextField();
jLabel47 = new javax.swing.JLabel();
jTextField34 = new javax.swing.JTextField();
jLabel48 = new javax.swing.JLabel();
jTextField35 = new javax.swing.JTextField();
jPanel4 = new javax.swing.JPanel();
jLabel49 = new javax.swing.JLabel();
jTextField36 = new javax.swing.JTextField();
jTextField37 = new javax.swing.JTextField();
jTextField38 = new javax.swing.JTextField();
jLabel58 = new javax.swing.JLabel();
jLabel59 = new javax.swing.JLabel();
jButton1 = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel1.setText("Clientes esperando fuera");
```

```
jLabel2.setText("Clientes esperando dentro");

jTextField1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField1ActionPerformed(evt);
    }
});

jLabel3.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N
jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel3.setText("Restaurante");

jPanel1.setBackground(new java.awt.Color(204, 204, 204));

jLabel5.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel5.setText("COCINA ESPAÑOLA");

jLabel4.setText("Puesto 1");

jLabel6.setText("Cocinero");

jLabel7.setText("Pedido");

jLabel8.setText("Pendiente");

jLabel9.setText("Puesto 2");

jLabel10.setText("Cocinero");

jLabel11.setText("Pedido");
```

```
        jLabel12.setText("Pendiente");

        jLabel13.setText("Puesto 3");

        jLabel14.setText("Cocinero");

        jLabel15.setText("Pedido");

        jLabel16.setText("Pendiente");

        jLabel22.setText("Pedidos completados");

        jTextField15.setText("0");

        jLabel23.setText("Importe total");

        jTextField16.setText("0");

        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addContainerGap())
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(16, 16, 16)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                .addComponent(jLabel6)
                .addComponent(jLabel7)
                .addComponent(jLabel8))
            .addGap(24, 24, 24)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                .addComponent(jTextField5,
javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING, false)
                    .addComponent(jLabel4)
                    .addComponent(jTextField3)
                    .addComponent(jTextField4,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(47, 47, 47)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                .addComponent(jLabel10)
                .addComponent(jLabel11)
                .addComponent(jLabel12))
            .addGap(24, 24, 24)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                .addComponent(jTextField8,
javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING, false)
```

```
                    .addComponent(jLabel9)

                    .addComponent(jTextField6)

                    .addComponent(jTextField7,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)))


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


.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)

                .addComponent(jLabel14)

                .addComponent(jLabel15)

                .addComponent(jLabel16))

            .addGap(24, 24, 24)


.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)

                .addComponent(jTextField11,
javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)


.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING, false)

                    .addComponent(jLabel13)

                    .addComponent(jTextField9)

                    .addComponent(jTextField10,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)))

            .addGap(24, 24, 24))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()

            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

            .addComponent(jLabel22)

            .addGap(18, 18, 18)
```

```
            .addComponent(jTextField15,
javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(39, 39, 39)
            .addComponent(jLabel23)
            .addGap(18, 18, 18)
            .addComponent(jTextField16,
javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(92, 92, 92))
        );
        jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(14, 14, 14)
            .addComponent(jLabel5)
            .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.T
RAILING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(jLabel4)

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

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                            .addComponent(jTextField3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel6))
                        .addGap(18, 18, 18)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jLabel7)
                    .addComponent(jTextField4,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jLabel8)
                    .addComponent(jTextField5,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addComponent(jLabel9)

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

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jTextField6,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel10))
                .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jLabel11)
                    .addComponent(jTextField7,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jLabel12)
                    .addComponent(jTextField8,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))))
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addComponent(jLabel13)

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

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jTextField9,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel14))
                .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jLabel15)
                    .addComponent(jTextField10,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jLabel16)
                    .addComponent(jTextField11,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))))
            .addGap(29, 29, 29)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                .addComponent(jLabel22)
                .addComponent(jTextField15,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel23)
                .addComponent(jTextField16,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap(39, Short.MAX_VALUE))
        );

        jPanel2.setBackground(new java.awt.Color(204, 204, 204));

        jLabel21.setBackground(new java.awt.Color(204, 204, 204));
        jLabel21.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
        jLabel21.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel21.setText("COCINA MEXICANA");

        jLabel27.setText("Puesto 1");

        jLabel25.setText("Pedido");

        jLabel28.setText("Puesto 2");

        jLabel29.setText("Cocinero");

        jLabel30.setText("Pedido");

        jLabel32.setText("Pedidos completados");
```

```
        jTextField23.setText("0");

        jLabel33.setText("Importe total");

        jTextField24.setText("0");

        jLabel31.setText("Pendiente");

        jLabel26.setText("Pendiente");

        jLabel24.setText("Cocinero");

        javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(jLabel21, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                    .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                            .addGroup(jPanel2Layout.createSequentialGroup()
                                .addGap(17, 17, 17)
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                    .addGroup(jPanel2Layout.createSequentialGroup()
                        .addComponent(jLabel32)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jTextField23,
javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(jPanel2Layout.createSequentialGroup()
                        .addComponent(jLabel26)

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

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                        .addComponent(jTextField19,
javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jTextField18,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)))))
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(16, 16, 16)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                        .addComponent(jLabel25)
                        .addComponent(jLabel24))
                    .addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                    .addGroup(jPanel2Layout.createSequentialGroup()
                        .addGap(6, 6, 6)
```

.addComponent(jTextField17,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addComponent(jLabel27))))

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

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(6, 6, 6)

.addComponent(jLabel33)

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

.addComponent(jTextField24,
javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel29)

.addComponent(jLabel30)

.addComponent(jLabel31))

.addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jTextField22,
javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jLabel28)

.addComponent(jTextField21,
javax.swing.GroupLayout.DEFAULT_SIZE, 90, Short.MAX_VALUE)

```
                        .addComponent(jTextField20)))))
                .addGap(0, 34, Short.MAX_VALUE)))
            .addContainerGap())
        );
        jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(14, 14, 14)
                .addComponent(jLabel21)
                .addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                    .addGroup(jPanel2Layout.createSequentialGroup()
                        .addComponent(jLabel27)

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

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                            .addComponent(jLabel24)
                            .addComponent(jTextField17,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                            .addComponent(jTextField18,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel25))
                        .addGap(18, 18, 18)
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jTextField19,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel26)))
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addComponent(jLabel28)

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

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jLabel29)
                    .addComponent(jTextField20,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jTextField21,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel30))
                .addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jTextField22,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel31))))
            .addGap(26, 26, 26)
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                .addComponent(jLabel32)
                .addComponent(jTextField23,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel33)
                .addComponent(jTextField24,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );

        jPanel3.setBackground(new java.awt.Color(204, 204, 204));

        jLabel34.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
        jLabel34.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel34.setText("COCINA ASIÁTICA");

        jLabel35.setText("Puesto 1");

        jLabel36.setText("Cocinero");

        jLabel37.setText("Pedido");

        jLabel38.setText("Pendiente");

        jLabel39.setText("Puesto 2");

        jLabel40.setText("Cocinero");
```

```
        jLabel41.setText("Pedido");

        jLabel42.setText("Pendiente");

        jLabel43.setText("Puesto 3");

        jLabel44.setText("Cocinero");

        jLabel45.setText("Pedido");

        jLabel46.setText("Pendiente");

        jLabel47.setText("Pedidos completados");

        jTextField34.setText("0");

        jLabel48.setText("Importe total");

        jTextField35.setText("0");

        javax.swing.GroupLayout jPanel3Layout = new
javax.swing.GroupLayout(jPanel3);
        jPanel3.setLayout(jPanel3Layout);
        jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel34, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addContainerGap())
```

```
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addGap(16, 16, 16)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                .addComponent(jLabel36)
                .addComponent(jLabel37)
                .addComponent(jLabel38))
            .addGap(24, 24, 24)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                .addComponent(jTextField27,
javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING, false)
                    .addComponent(jLabel35)
                    .addComponent(jTextField25)
                    .addComponent(jTextField26,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(47, 47, 47)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                .addComponent(jLabel40)
                .addComponent(jLabel41)
                .addComponent(jLabel42))
            .addGap(24, 24, 24)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                .addComponent(jTextField30,
javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING, false)
                    .addComponent(jLabel39)
                    .addComponent(jTextField28)
                    .addComponent(jTextField29,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 53,
Short.MAX_VALUE)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                .addComponent(jLabel44)
                .addComponent(jLabel45)
                .addComponent(jLabel46))
            .addGap(24, 24, 24)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                .addComponent(jTextField33,
javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING, false)
                    .addComponent(jLabel43)
                    .addComponent(jTextField31)
                    .addComponent(jTextField32,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(24, 24, 24))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addComponent(jLabel47)
```

```
            .addGap(18, 18, 18)
            .addComponent(jTextField34,
javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(39, 39, 39)
            .addComponent(jLabel48)
            .addGap(18, 18, 18)
            .addComponent(jTextField35,
javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(92, 92, 92))
        );
        jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()
            .addGap(14, 14, 14)
            .addComponent(jLabel34)
            .addGap(18, 18, 18)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.T
RAILING)
                    .addGroup(jPanel3Layout.createSequentialGroup()
                        .addComponent(jLabel35)

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

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                            .addComponent(jTextField25,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel36))
```

```
                .addGap(18, 18, 18)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jLabel37)
                    .addComponent(jTextField26,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jLabel38)
                    .addComponent(jTextField27,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addGroup(jPanel3Layout.createSequentialGroup()
                    .addComponent(jLabel39)

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

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jTextField28,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel40))
                .addGap(18, 18, 18)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jLabel41)
                    .addComponent(jTextField29,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
                .addGap(18, 18, 18)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jLabel42)
                    .addComponent(jTextField30,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))))
                .addGroup(jPanel3Layout.createSequentialGroup()
                    .addComponent(jLabel43)

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

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jTextField31,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel44))
                .addGap(18, 18, 18)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jLabel45)
                    .addComponent(jTextField32,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jLabel46)
                    .addComponent(jTextField33,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))))
```

```
            .addGap(29, 29, 29)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                .addComponent(jLabel47)
                .addComponent(jTextField34,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel48)
                .addComponent(jTextField35,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap(18, Short.MAX_VALUE))
        );

        jPanel4.setBackground(new java.awt.Color(204, 204, 204));

        jLabel49.setBackground(new java.awt.Color(204, 204, 204));
        jLabel49.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
        jLabel49.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel49.setText("SALA DESCANSO");

        jTextField37.setText("0");

        jTextField38.setText("0");

        jLabel58.setText("Pedidos completados");

        jLabel59.setText("Importe total");

        javax.swing.GroupLayout jPanel4Layout = new
javax.swing.GroupLayout(jPanel4);
```

```
        jPanel4.setLayout(jPanel4Layout);
        jPanel4Layout.setHorizontalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel4Layout.createSequentialGroup()

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.T
RAILING)
                    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel4Layout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(jLabel49, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                    .addGroup(jPanel4Layout.createSequentialGroup()
                        .addGap(0, 14, Short.MAX_VALUE)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                            .addComponent(jTextField36,
javax.swing.GroupLayout.PREFERRED_SIZE, 377,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGroup(jPanel4Layout.createSequentialGroup()

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                                    .addComponent(jLabel58)
                                    .addComponent(jLabel59))
                                .addGap(38, 38, 38)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING, false)
                                    .addComponent(jTextField37,
javax.swing.GroupLayout.DEFAULT_SIZE, 75, Short.MAX_VALUE)
                                    .addComponent(jTextField38))))
                        .addGap(14, 14, 14)))
                .addContainerGap())
```

```
        );
        jPanel4Layout.setVerticalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel4Layout.createSequentialGroup()
                .addGap(14, 14, 14)
                .addComponent(jLabel49)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jTextField36,
javax.swing.GroupLayout.PREFERRED_SIZE, 62,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jTextField37,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel58))
                .addGap(12, 12, 12)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jTextField38,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel59))
                .addContainerGap(25, Short.MAX_VALUE))
        );


        jButton1.setText("Parar");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
            jButton1ActionPerformed(evt);
        }
    });

    jButton2.setText("Reanudar");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton2ActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addGap(65, 65, 65))
                .addGroup(layout.createSequentialGroup()
                    .addGap(12, 12, 12)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
                        .addComponent(jLabel2)
                        .addGroup(layout.createSequentialGroup()
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILIN
G, false)
                    .addComponent(jPanel3,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jPanel1,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
                            .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jPanel4,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(135, 135, 135)
                        .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
                        .addComponent(jButton2,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE))))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILIN
G, false)
```

```
                        .addComponent(jTextField2,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 860, Short.MAX_VALUE)
                        .addComponent(jTextField1,
javax.swing.GroupLayout.Alignment.LEADING))
                    .addComponent(jLabel1))
                .addContainerGap()))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(24, 24, 24)
                .addComponent(jLabel3)
                .addGap(18, 18, 18)
                .addComponent(jLabel1)
                .addGap(18, 18, 18)
                .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(26, 26, 26)
                .addComponent(jLabel2)
                .addGap(18, 18, 18)
                .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
, false)
                    .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addGap(18, 18, 18)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
                .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELIN
E)
                        .addComponent(jButton1)
                        .addComponent(jButton2))))
            .addContainerGap(29, Short.MAX_VALUE))
        );


        pack();
    }// </editor-fold>


    private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }


    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        parada.abrir();
    }


    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        parada.cerrar();
    }
```

```
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new PantallaR().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel22;
private javax.swing.JLabel jLabel23;
private javax.swing.JLabel jLabel24;
private javax.swing.JLabel jLabel25;
private javax.swing.JLabel jLabel26;
private javax.swing.JLabel jLabel27;
private javax.swing.JLabel jLabel28;
private javax.swing.JLabel jLabel29;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel30;
```

```
private javax.swing.JLabel jLabel31;
private javax.swing.JLabel jLabel32;
private javax.swing.JLabel jLabel33;
private javax.swing.JLabel jLabel34;
private javax.swing.JLabel jLabel35;
private javax.swing.JLabel jLabel36;
private javax.swing.JLabel jLabel37;
private javax.swing.JLabel jLabel38;
private javax.swing.JLabel jLabel39;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel40;
private javax.swing.JLabel jLabel41;
private javax.swing.JLabel jLabel42;
private javax.swing.JLabel jLabel43;
private javax.swing.JLabel jLabel44;
private javax.swing.JLabel jLabel45;
private javax.swing.JLabel jLabel46;
private javax.swing.JLabel jLabel47;
private javax.swing.JLabel jLabel48;
private javax.swing.JLabel jLabel49;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel58;
private javax.swing.JLabel jLabel59;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
```

```
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField10;
private javax.swing.JTextField jTextField11;
private javax.swing.JTextField jTextField15;
private javax.swing.JTextField jTextField16;
private javax.swing.JTextField jTextField17;
private javax.swing.JTextField jTextField18;
private javax.swing.JTextField jTextField19;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField20;
private javax.swing.JTextField jTextField21;
private javax.swing.JTextField jTextField22;
private javax.swing.JTextField jTextField23;
private javax.swing.JTextField jTextField24;
private javax.swing.JTextField jTextField25;
private javax.swing.JTextField jTextField26;
private javax.swing.JTextField jTextField27;
private javax.swing.JTextField jTextField28;
private javax.swing.JTextField jTextField29;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField30;
private javax.swing.JTextField jTextField31;
private javax.swing.JTextField jTextField32;
private javax.swing.JTextField jTextField33;
private javax.swing.JTextField jTextField34;
private javax.swing.JTextField jTextField35;
private javax.swing.JTextField jTextField36;
private javax.swing.JTextField jTextField37;
private javax.swing.JTextField jTextField38;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
```

```
    private javax.swing.JTextField jTextField6;
    private javax.swing.JTextField jTextField7;
    private javax.swing.JTextField jTextField8;
    private javax.swing.JTextField jTextField9;
    // End of variables declaration
}
```

**Clase Parada:**

```
package com.mycompany.proyecto;

public class Parada {

    private boolean cerrado;
    private Log log;

    public Parada()
    {
        cerrado=false;
        log=Log.getInstance();
    }

    public synchronized void puedoPasar()
    {
        while(cerrado){
            try
            {
                wait();
            }catch(InterruptedException e){}
        }
    }

    public synchronized void cerrar()
```

```
    {
        //Si estaba en marcha y se para lo representamos en el log
        if (!cerrado){log.escribir("Se ha parado la simulacion");}
        cerrado=true;

    }

    public synchronized void abrir()
    {
        //Si estaba parada y se reanuda lo representamos en el log
        if (cerrado){log.escribir("Se ha reanudado la simulacion");}
        cerrado=false;
        notifyAll();
    }
}
```

**Clase Proyecto:**

```
package com.mycompany.proyecto;

import static java.lang.Thread.sleep;
import java.rmi.RemoteException;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JButton;
import javax.swing.JTextField;

public class Proyecto {

    public static void main(String[] args) {
        //Iniciamos jFrame
        PantallaR pr=new PantallaR();
```

```
        pr.setVisible(true);
        Parada parada=pr.getParada();

        //Tomamos todos los jFrame
        ArrayList<JTextField> datos=pr.getJTF();

        //Tomamos los botones
        ArrayList<JButton> botones=pr.getJButton();

        //Creamos listas para gestionar los cocineros en descanso
        ListaCocineros cocinerosD=new ListaCocineros(datos.get(32));



        //Creamos cada puesto de la cocina española
        ArrayList<Puesto> puestos=new ArrayList<>();
        for (int i=0; i<3; i++)
        {
           ArrayList<JTextField> campos=new ArrayList<>();
           for (int j=0; j<3; j++)
           {
              campos.add(datos.get(i*3+j+2));
           }
           Puesto p=new Puesto(campos);
           puestos.add(p);
        }
        //Creamos la cocina española
        Cocina cEsp=new Cocina(puestos, datos.get(11), datos.get(12), datos.get(33),
datos.get(34), parada);

        //Creamos cada puesto de la cocina asiatica
        puestos=new ArrayList<>();
        for (int i=0; i<2; i++)
```

```
        {
            ArrayList<JTextField> campos=new ArrayList<>();
            for (int j=0; j<3; j++)
            {
                campos.add(datos.get(i*3+j+13));
            }
            Puesto p=new Puesto(campos);
            puestos.add(p);
        }
        //Creamos la cocina asiatica
        Cocina cMex=new Cocina(puestos, datos.get(19), datos.get(20), datos.get(33),
datos.get(34), parada);



        //Creamos cada puesto de la cocina mexicana
        puestos=new ArrayList<>();
        for (int i=0; i<3; i++)
        {
            ArrayList<JTextField> campos=new ArrayList<>();
            for (int j=0; j<3; j++)
            {
                campos.add(datos.get(i*3+j+21));
            }
            Puesto p=new Puesto(campos);
            puestos.add(p);
        }
        //Creamos la cocina mexicana
        Cocina cAsi=new Cocina(puestos, datos.get(30), datos.get(31), datos.get(33),
datos.get(34), parada);


        //Creamos el restaurante
        Restaurante r=null;
```

```
        try {
            r = new Restaurante(datos, cEsp, cMex, cAsi, cocinerosD, parada,
botones.get(0), botones.get(1));
        } catch (RemoteException ex) {
            System.out.println("No se ha podido registrar correctamente");
        }

        //Creamos a los cocineros
        for (int i=1; i<4; i++)
        {
            Cocinero c=new Cocinero("CO"+i, 0, r, cEsp);
            c.start();
        }

        for (int i=4; i<6; i++)
        {
            Cocinero c=new Cocinero("CO"+i, 1, r, cMex);
            c.start();
        }

        for (int i=6; i<9; i++)
        {
            Cocinero c=new Cocinero("CO"+i, 2, r, cAsi);
            c.start();
        }

        //Creamos los clientes
        for (int i=0;i<20000;i++)
        {
            try {
                //Esperamos de 0,5 a 6 segundos para generar el siguiente cliente
                esperarSig();
```

```
                parada.puedoPasar();
            } catch (InterruptedException ex) {}
            Cliente c=new Cliente("CL"+i, r);
            c.start();
        }
    }

    private static void esperarSig() throws InterruptedException
    {
        sleep(500+(int)(500*Math.random()));
    }
}
```

**Clase Puesto:**

```
package com.mycompany.proyecto;

import java.util.ArrayList;
import java.util.concurrent.Exchanger;
import javax.swing.JTextField;

public class Puesto {

    private JTF cocinero;
    private JTF pedido;
    private JTF pendiente;
    private int coste;
    private Exchanger<String> exch;
    private boolean ocupado;
    private String cliente;

    public Puesto(ArrayList<JTextField> campos)
    {
```

```
    this.cocinero=new JTF(campos.get(0));
    this.pedido=new JTF(campos.get(1));
    this.pendiente=new JTF(campos.get(2));
    this.coste=0;
    this.exch=new Exchanger();
    ocupado=false;
    cliente="";
}

public void setOcupado(boolean oc)
{
    ocupado=oc;
}

public boolean getOcupado()
{
    return ocupado;
}

public void setCocinero(String cocinero)
{
    this.cocinero.setJtf(cocinero);
}

public void setPedido(String pedido)
{
    this.pedido.setJtf(pedido);
}

public String getPedido()
{
```

```
    return pedido.getJtf();
}

public void setPendientes(String pendiente)
{
    this.pendiente.setJtf(pendiente);
}

public int getCoste()
{
    return coste;
}

public void setCoste(int coste)
{
    this.coste=coste;
}

public Exchanger<String> getExch()
{
    return exch;
}

public String getCliente()
{
    return cliente;
}

public void setCliente(String cli)
{
    this.cliente=cli;
```

```
    }

    public void limpiar()
    {
        cocinero.setJtf("");
        pedido.setJtf("");
        pendiente.setJtf("");
        coste=0;
        ocupado=false;
        cliente="";
    }
}
```

**Clase RemoteCliente:**

**Clase Restaurante:**

```
package com.mycompany.proyecto;

import static java.lang.Thread.sleep;
import java.rmi.Naming;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemoteObject;
import java.util.ArrayList;
import java.util.concurrent.Exchanger;
import java.util.concurrent.Semaphore;
import java.util.concurrent.locks.Lock;
import java.util.concurrent.locks.ReentrantLock;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JButton;
```

```
import javax.swing.JTextField;

public class Restaurante extends UnicastRemoteObject implements InterfazComun{

    //Listas para actualizar los clientes
    private ListaClientes fuera;
    private ListaClientes dentro;

    //Controlar maquina de pedidos
    private Semaphore cola;
    private Lock maquina;

    //Cocinas
    private Cocina cEsp;
    private Cocina cAsi;
    private Cocina cMex;

    //Lista de cocineros descansando
    private ListaCocineros cocinerosD;

    //Monitor para controlar la parada
    Parada parada;

    //RMI
    private Restaurante obj;

    //Pedidos e importe total
    JTextField pedidos;
    JTextField importe;

    //Boton de parar y reanudar
```

```
    private JButton parar;
    private JButton reanudar;

    public Restaurante(ListaClientes fuera, ListaClientes dentro, JTextField pedidos,
JTextField importe, JButton parar, JButton reanudar) throws RemoteException
    {
        this.fuera=fuera;
        this.dentro=dentro;
        this.pedidos=pedidos;
        this.importe=importe;
        this.parar=parar;
        this.reanudar=reanudar;
    }

    public Restaurante(ArrayList<JTextField> jtf, Cocina cEsp, Cocina cMex, Cocina
cAsi, ListaCocineros cocinerosD, Parada parada, JButton parar, JButton reanudar)
throws RemoteException
    {
        fuera=new ListaClientes(jtf.get(0));
        dentro=new ListaClientes(jtf.get(1));

        cola=new Semaphore(20);
        maquina=new ReentrantLock();

        this.cEsp=cEsp;
        this.cMex=cMex;
        this.cAsi=cAsi;

        this.cocinerosD=cocinerosD;

        this.parada=parada;
```

```
    this.pedidos=jtf.get(33);
    this.importe=jtf.get(34);

    this.parar=parar;
    this.reanudar=reanudar;

    try{
        obj = new Restaurante(fuera,dentro,pedidos,importe,parar,reanudar);
        Registry registry = LocateRegistry.createRegistry(1099);
        Naming.rebind("//localhost/ObjetoC", obj);

    }catch(Exception e){
        System.out.println("doy error");
    }
}

public void esperarCola(Cliente c) throws InterruptedException
{
    fuera.meter(c);

    cola.acquire();
    //Estamos dentro si no nos quedamos en la cola de afuera
    fuera.sacar(c);
    dentro.meter(c);
    //Cola (dentro) para usar la maquina
    maquina.lock();
}

public Cocina usarMaquina(Cliente c) throws InterruptedException
{
    //Hacemos la comanda con la maquina de pedidos
```

```
int tipo=(int) (3*Math.random());
int nProd=5+(int)(6*Math.random());
Comanda comanda=new Comanda(tipo,nProd,c.getMiId());
sleep(2000+(int)(1000*Math.random()));
parada.puedoPasar();
//Una vez usada la maquina, ya esta lista la comanda
//Tomamos la cocina correspondiente
Cocina cocina=null;
switch(tipo)
{
   case 0:
      cocina=cEsp;
      break;
   case 1:
      cocina=cMex;
      break;
   case 2:
      cocina=cAsi;
      break;
   default:
      cocina=cEsp;
      break;
}

cocina.añadirNueva(comanda);
//Salimos de la cola
maquina.unlock();
dentro.sacar(c);
cola.release();

return cocina;
```

```
}

public Puesto esperarComida(Cliente c, Cocina cocina) throws InterruptedException
{
   //Tomamos el puesto que corresponde con nuestra comanda
   Puesto puesto=cocina.estaLista(c.getMiId());
   return puesto;
}

public void pagar(Cliente c, Puesto puesto)
{
   String cambio=Integer.toString(puesto.getCoste());
   Exchanger<String> e=puesto.getExch();
   try {
      cambio=e.exchange(cambio);
   } catch (InterruptedException ex) {}
}

public void descansar(Cocinero c)
{
   //Cocinero descansa 10s
   try {
      cocinerosD.meter(c);
      sleep(10000);
      parada.puedoPasar();
   } catch (InterruptedException ex) {
   } finally {cocinerosD.sacar(c);}
}


//Metodos remotos
```

```
    public int getFuera() throws RemoteException {
        return fuera.getContador();
    }

    public int getDentro() throws RemoteException {
        return dentro.getContador();
    }

    public String getPedidos() throws RemoteException {
        return pedidos.getText();
    }

    public String getImporte() throws RemoteException {
        return importe.getText();
    }

    public void pulsarParar() throws RemoteException {
        parar.doClick();
    }
    public void pulsarReanudar() throws RemoteException {
        reanudar.doClick();
    }
}
```