Cookie Factory

Un dibujo de una cara feliz

Descripción generada automáticamente con confianza baja

**Curso 2024/2025 - Convocatoria Ordinaria**

**Grado en Ingeniería en Sistemas de Información (GISI)**

**Asignatura: Paradigmas de programación**

**Práctica Final Laboratorio**

**Tutores: Eva García López**

**David de Fitero Domínguez**

**Antonio García Cabot**

**Sergio Caro Álvaro**

**David Sánchez Sánchez**

**02751903E**

Universidad de Alcalá

Madrid, Spain

d.sanchezsanchez@edu.uah.es





Índice

[1.Análisis aplicación 3](#_Toc184748625)

[2.Diseño general del sistema y sincronización 4](#_Toc184748626)

[3.Clases principales 5](#_Toc184748627)

[4.Diagrama de clases 6](#_Toc184748628)

[5.Anexos 7](#_Toc184748629)

[5.1 Código fuente del programa 7](#_Toc184748630)

[5.1.1 NombreDelArchivo 7](#_Toc184748631)

[5.1.2 NombreDelArchivo 7](#_Toc184748632)

[5.1.3 NombreDelArchivo 7](#_Toc184748633)

[5.1.4 NombreDelArchivo 7](#_Toc184748634)

[5.1.5 NombreDelArchivo 7](#_Toc184748635)

[5.1.6 NombreDelArchivo 7](#_Toc184748636)

[5.1.7 NombreDelArchivo 7](#_Toc184748637)

[5.1.8 NombreDelArchivo 7](#_Toc184748638)

[5.1.9 NombreDelArchivo 7](#_Toc184748639)

[5.1.10 NombreDelArchivo 7](#_Toc184748640)

[5.1.11 NombreDelArchivo 7](#_Toc184748641)

[5.1.12 NombreDelArchivo 7](#_Toc184748642)

[5.1.13 NombreDelArchivo 7](#_Toc184748643)

# 1.Análisis aplicación

En el siguiente apartado se tratará de manera general los puntos más claves de la aplicación presentada a rasgos generales, sobre todo empleado a la finalidad de la aplicación y al desarrollo de las características principales de la interfaz

# 2.Diseño general del sistema y sincronización

En este apartado hablaremos de diseño de la aplicación, es decir de que forman se interconectan las distintas clases, y de que manera se ha conseguido generar la conexión cliente-servidor.

# 3.Clases principales

En este punto se describirán las clases principales del proyecto, enfocándonos en sus **atributos** y sus **métodos principales**.

# 4.Diagrama de clases

A continuación, se muestra un diagrama de clases de la aplicación desarrollado en **Enterprise Architect** con sus respectivas **relaciones**.

# 5.Anexos (Código fuente del programa)

Se mostrará de manera anexada (Los paquetes y dentro sus archivos correspondientes) cada uno de los códigos fuentes del programa en función de cada archivo \*.java para que se pueda seguir de manera adecuada.

## 5.1 Server

### 5.1.1 InicializacionServer.java

*/\*\**

*\* version 1.0*

*\* contacto --> https://www.linkedin.com/in/david-sanchez-1366b1253/*

*\* @author David Sánchez*

*\*/*

*//Paquete al que pertenece la clase*

package Server;

import ServerInterface.PanelSCM;

import java.util.List;

import java.util.ArrayList;

import java.net.MalformedURLException;

import java.rmi.RemoteException;

import java.rmi.Naming;

import java.rmi.registry.LocateRegistry;

*//Importes*

public class InicializacionServer

{

    public static void main(String[] args)

    {

*//Siempre agruparemos en listas los reposteros,horno y empaquetadores.*

        List<Repostero> listaReposteros = new ArrayList<>();

        List<Horno> listaHornos = new ArrayList<>();

        List<Empaquetador> listaEmpaquetadores = new ArrayList<>();

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*\*    INICIALIZACIÓN DE CLASES    \**

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*//Creamos la cafeteria*

        Cafeteria cafeteria = new Cafeteria();

*//Creamos el almacen*

        Almacen almacen = new Almacen();

*//Creamos los empaquetadores*

        for (int i = 1; i <= 3; i++)

        {

*//Creamos el empaquetador*

            Empaquetador empaquetador = new Empaquetador(i,almacen);

*//Lo añadimos a la lista para posteriormente referenciarlo en los hornos y al RMI*

            listaEmpaquetadores.add(empaquetador);

        }

*//Creamos los hornos*

        for (int i = 1; i <= 3; i++)

        {

*//Creamo el horno*

            Horno horno = new Horno(i,listaEmpaquetadores.get(i-1));

*//Inicializamos el hilo*

            horno.start();

*//Lo añadimos a la lista para posteriormente referenciarlo en los reposteros y al RMI*

            listaHornos.add(horno);

        }

*//Creamos los reposteros*

        for (int i = 1; i <= 5; i++)

        {

*//Creamos al repostero*

            Repostero repostero = new Repostero(i, listaHornos,cafeteria);

*//Inicializamos el hilo*

            repostero.start();

*//Lo añadimos a la lista para referenciarlo al RMI*

            listaReposteros.add(repostero);

        }

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*\*   INICIALIZACIÓN DE INTERFAZ   \**

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*//Inicializamos la interfaz del servidor*

        PanelSCM interfazServer = new PanelSCM(listaReposteros, listaHornos, listaEmpaquetadores, cafeteria,almacen);

        interfazServer.setVisible(true);

        Thread hiloInterfazServidor = new Thread(interfazServer);

        hiloInterfazServidor.start();

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*\*   INICIALIZACIÓN DE SERVIDOR   \**

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

        try

        {

            MetodosParaCliente objetoTraspasable = new MetodosParaCliente(listaReposteros, listaHornos, listaEmpaquetadores, cafeteria,almacen);

            LocateRegistry.createRegistry(1099);

*//Le pasamos los objetos al "Server"*

            Naming.rebind("//localhost/objetoTraspasable", objetoTraspasable);

        }

        catch (RemoteException error)

        {

            System.out.println("Se ha producido un error mientras se levantaba el servidor en el Server --> " + error);

        }

        catch (MalformedURLException error)

        {

            System.out.println("Se ha producido un error mientras se subia delsde el Servidor el Objeto Traspasable --> " + error);

        }

    }

}

### 5.1.2 Repostero.java

*/\*\**

*\* version 1.0*

*\* contacto --> https://www.linkedin.com/in/david-sanchez-1366b1253/*

*\* @author David Sánchez*

*\*/*

*//Paquete al que pertenece la clase*

package Server;

*//Importes*

import Logger.Logger;

import java.io.Serializable;

import java.util.List;

import java.util.Random;

import java.util.concurrent.locks.ReentrantLock;

import Logger.Logger;

*//CLASE*

public class Repostero extends Thread implements Serializable

{

*//Atributos de la clase*

    private int identificador;

*//Con este jTextField podremos mostrar en la interfaz diferentes textos sobre los reposteros*

*//En esta ArrayList almacenaremos los horno de manera ordenada*

    private List<Horno> listaHornos;

*//Este stributo se utilizara para mandar la accion que esta realizando el repostero a la interfaz a la interfaz*

    private String accion;

*//Atributos de producción*

    private int totalGalletas = 0;                  *//Cuando se crea el repostero aun no ha creado ninguna galleta*

    private int totalGalletasDesperdiciadas = 0;    *//Cuando se crea el repostero aun no ha desperdiciado ninguna galleta*

    private int tandaGalletas;

    private int tandaGalletasDesperdiciadas;

    private boolean pausaCafe = false;              *//Por defecto al crearse el repostero primero debera de hacer una tanda de galletas antes de descansar*

*//Para la cafeteria*

    private Cafeteria cafeteria;

*//Usamos estos atributos para generar la pausa del hilo*

    private boolean paradaManual = false;       *//Por defecto la parada manual se pondra a true despues de crear el hilo*

    private ReentrantLock lockHorno = new ReentrantLock();

*//Para saber si hay hornoDisponibles*

    private boolean hornosDisponinles = true;       *//Por defecto siempre estaran disponibles hasta que se marque que estan todos llenos*

*//Usaremos este objeto para la aletoridad*

    private Random aleatorio = new Random();

*//Usaremos este objeto para los logs del sistema*

    private Logger log = Logger.getInstance();

    private int contadorLog = 0;

*//Contructor*

    public Repostero(int \_identificador, List<Horno> \_listaHornos,Cafeteria \_cafeteria)

    {

        this.identificador = \_identificador;

        this.listaHornos = \_listaHornos;

        this.cafeteria = \_cafeteria;

    }

*//Contructor vacio [NO HAREMOS USO DE ESTE]*

*//Contructores alternativos o sobrecargados [NO HAREMOS USO DE ESTOS]*

*//Getters*

    public int getTotalGalletas(){return totalGalletas;}

    public int getTotalGalletasDesperdiciadas(){return totalGalletasDesperdiciadas;}

    public int getIdentificador(){return identificador;}

    public boolean getParadaManual(){return paradaManual;}

    public String getAccion(){return accion;}

*//Setters*

    public void setParadaManual(boolean \_paradaManual){this.paradaManual = \_paradaManual;}

    public void setAccion(String \_accion){this.accion = \_accion;}

*//To String (Solo si es necesario)*

    public String toStringIdentificador()

    {

        return String.valueOf(identificador);

    }

    @Override

    public String toString()

    {

        return "Repostero ["+identificador+"]";

    }

*//Metodos de la clase*

    @Override

    public void run()

    {

*//Los reposteros nunca dejaran de trabajar*

        while (true)

        {

            try{

*//Realizaran esta secuencia entre 3 y 5 veces*

                for (int i=0; i <= (3 + aleatorio.nextInt(3)); i++)

                {

*//Comprobamos si hay paradaManual*

                    comprobamosParadaManual();

*//Producción de la tanda de galletas*

                    accion = "PRODUCIENDO";

                    log.log("Repostero["+identificador+"] -->"+accion);

                    Thread.sleep(2000 + aleatorio.nextInt(2001));

                    tandaGalletas = 37 + aleatorio.nextInt(9);

                    totalGalletas += tandaGalletas;

                    log.log("Repostero["+identificador+"] =========> Ha producido "+tandaGalletas+" galletas");

*//Comprobamos si hay parada manual*

                    comprobamosParadaManual();

*//Depositamos las galletas en cualquier horno disponible*

                    accion = "DEPOSITANDO";

                    log.log("Repostero["+identificador+"] -->"+accion);

                    tandaGalletasDesperdiciadas = depositarGalletas(tandaGalletas);

*//Si se han desperdiciado galletas se registrara en el log*

                    if(tandaGalletasDesperdiciadas > 0)

                    {

                        log.log("Repostero["+identificador+"] =========> Ha desperdiciado "+tandaGalletasDesperdiciadas+" galletas");

                    }

                    totalGalletasDesperdiciadas += tandaGalletasDesperdiciadas;

                }

*//Comprobamos si hay parada manual*

                comprobamosParadaManual();

*//Parada para el cafe*

                cafeteria.tomarCafe(this);              *//Le pasamos el propio hilo a la cafeteria*

*//Comprobamos si hay parada manual*

                comprobamosParadaManual();

*//Descanso hasta que vuelvan a generar más tandas de galletas*

                accion = "DESCANSANDO";

                log.log("Repostero["+identificador+"] -->"+accion);

                Thread.sleep(3000 + aleatorio.nextInt(3001));

            }

            catch (InterruptedException error)

            {

                System.out.println("Durante la ejecucion del Repostero["+identificador+"] --> " + error);

            }

        }

    }

*//El metodo implica synchronized para evitar que dos reposteros metan a la vez galletas en el mismo horno*

    public int depositarGalletas(int \_tandaGalletas)

    {

*//Aqui depositaremos las galletas que se desperdicien*

        int \_tandaGalletasDesperdiciadas = 0;   *//Por defecto ninguna*

*//Para indicar que ya hemos depositado las galletas en el horno*

        boolean depositoTerminado = false;

        try

        {

            lockHorno.lock();

*//Antes de elegir el horno en al que vamos a depositar las galletas*

            while (!depositoTerminado)

            {

                for(int indexHornos = 0; indexHornos < listaHornos.size(); indexHornos++)

                {

                    if(listaHornos.get(indexHornos).getCantidadGalletas() < listaHornos.get(indexHornos).getCapacidadMaxima() && !listaHornos.get(indexHornos).getEstaHorneando() && !listaHornos.get(indexHornos).getEstaEmpaquetando())

                    {

                        \_tandaGalletasDesperdiciadas = listaHornos.get(indexHornos).añadirGalletas(\_tandaGalletas);

*//Registramos cuantas galletas se han depositado en el horno*

                        int galletasDepositadasHorno = \_tandaGalletas - \_tandaGalletasDesperdiciadas;

                        log.log("Repostero["+identificador+"] =========> Ha depositado "+galletasDepositadasHorno+" galletas en el Horno["+listaHornos.get(indexHornos).getIdentificador()+"]");

                        depositoTerminado = true;

                        break;  *//Break del bucle for*

                    }

                    else

                    {

                        try

                        {

                            Thread.sleep(100);

                        }

                        catch(InterruptedException error)

                        {

                            System.out.println("Durante una espera de seguridad para liberar memoria a la hora de encontrar un horno que no este lleno se a interrumpido la ejecucion de codigo generando el siguiente error --> " + error);

                        }

                    }

                }

            }

        }

        catch (Error error)

        {

            System.out.println("Se ha producido un error en el repostero ["+identificador+"] mientras e le asignaba un horno al que transportar las galletas producidas --> ) + error");

        }

        finally

        {

*//Siempre desbloqueamos el lock de los hornos*

            lockHorno.unlock();

        }

        return \_tandaGalletasDesperdiciadas;

    }

    public void comprobamosParadaManual()

    {

        try

        {

            while(paradaManual)

            {

*//Espera activa*

                accion = "BLOQUEADO";

*//Solo mostramos una vez en el log que esta bloqueado*

                if(contadorLog == 0)

                {

                    log.log("Repostero["+identificador+"] -->"+accion);

                    contadorLog++;

                }

                Thread.sleep(1000);

            }

*//Si pasa por aqui significa que puede escribir el log de que esta bloqueado, entonces*

            contadorLog = 0;

        }

        catch(InterruptedException error)

        {

            System.out.println("Se ha producido un error mientras se hacia la espera activa para reanudar el repostero["+identificador+"] --> " + error);

        }

    }

}

### 5.1.3 Horno.java

*/\*\**

*\* version 1.0*

*\* contacto --> https://www.linkedin.com/in/david-sanchez-1366b1253/*

*\* @author David Sánchez*

*\*/*

*//Paquete al que pertenece la clase*

package Server;

*//Importes*

import Logger.Logger;

import java.io.Serializable;

*//CLASE*

public class Horno extends Thread implements Serializable

{

*//Contantes de la clase (Bloqueamos la variable con la sentencia final)*

    private final int capacidadMaxima = 200;

*//Atributos de la clase*

    private Empaquetador empaquetador;          *//Su empaquetador asociado*

    private int identificador;

    private int cantidadGalletas = 0;           *//Por defecto cuando se cree el horno siempre estara vacio*

    private int totalGalletasHorneadas = 0;     *//Por defecto cuando se cree el horno no llevara ninguna galleta horneada*

    private boolean estaHorneando = false;      *//Por defecto cuando se cree el horno no estara en proceso de horneado hasta que no se llene*

    private boolean estaEmpaquetando = false;

    private String accion;

*//Para los logs*

    private Logger log = Logger.getInstance();

*//Contructor*

    public Horno(int \_identificador, Empaquetador \_empaquetador)

    {

        this.identificador = \_identificador;

        this.empaquetador = \_empaquetador;

    }

*//Contructor vacio*

*//Contructores alternativos o sobrecargados*

*//Getters*

    public int getIdentificador(){return identificador;}

    public int getCantidadGalletas(){return cantidadGalletas;}

    public int getCapacidadMaxima(){return capacidadMaxima;}

    public int getTotalGalletasHorneadas(){return totalGalletasHorneadas;}

    public boolean getEstaHorneando(){return estaHorneando;}

    public boolean getEstaEmpaquetando(){return estaEmpaquetando;}

    public String getAccion(){return accion;}

    public int contadorLog = 0;

*//Setters*

    public void setCantidadGalletas(int \_cantidadGalletas){this.cantidadGalletas = \_cantidadGalletas;}

    public void setAccion(String \_accion){this.accion = \_accion;}

*//To String (Solo si es necesario)*

    public String toStringIdentificador()

    {

        return "Horno ["+identificador+"]";

    }

*//Metodos de la clase*

    public int añadirGalletas(int galletas)

    {

*//Una vez que entran aqui significa que hay huecos disponibles*

        int galletasDesperdiciadas = 0;

        if((cantidadGalletas+galletas) <= capacidadMaxima)

        {

            cantidadGalletas += galletas;

        }

        else

        {

            galletasDesperdiciadas = (cantidadGalletas+galletas)-capacidadMaxima;

            cantidadGalletas = capacidadMaxima;

        }

        return galletasDesperdiciadas;

    }

    @Override

    public void run()

    {

        try

        {

*//Un horno nunca va a parar de trabajar*

            while(true)

            {

*//Solo lo escribimos una vez en el log cada vez que se empieza a llenar*

                if(contadorLog == 0)

                {

*//Si esta aqui significa que esta en estado de llenado*

                    accion = "LLENANDOSE";

                    log.log("Horno["+identificador+"] -->"+accion);

                    contadorLog++;

                }

                if(cantidadGalletas == capacidadMaxima && !estaHorneando)

                {

*//Comienza el horneado*

                    accion = "HORNEANDO";

                    log.log("Horno["+identificador+"] -->"+accion);

                    estaHorneando = true;

                    Thread.sleep(8000);             *//Tardan en hornear la tanda 8000 milisegundos (8 segundos)*

*//Anotamos que las galletas ya estan horneadas*

                    totalGalletasHorneadas += capacidadMaxima;

*//Termina el horneado*

                    estaHorneando = false;

*//Empieza empaquetado, es decir, se empieza a vaciar*

                    accion = "VACIANDOSE";

                    log.log("Horno["+identificador+"] -->"+accion);

                    estaEmpaquetando = true;

*//Avisamos al empaquetador para que vacie el horno*

                    empaquetador.vaciarHorno(this);

*//Ahora esperamos a que se vacie el horno*

*//Termina empaquetado*

                    estaEmpaquetando = false;

*//Cuando termina todo el proceso significa que ya esta vacio por lo que cambiamos el contador del log*

                    contadorLog = 0;

                }

                else

                {

*//Pequeña espera hasta que vuelva a comprobar si ya esta lleno*

                    Thread.sleep(500);

                }

            }

        }

        catch(InterruptedException error)

        {

            System.out.println("Se ha producido un error mientras se ejecutaba el Horno["+identificador+"] --> " + error);

        }

    }

    public int sacarHorneadoGalletas(int numeroGalletasSacar)

    {

        cantidadGalletas -= numeroGalletasSacar;

        return numeroGalletasSacar;

    }

}

### 5.1.4 Empaquetador.java

*/\*\**

*\* version 1.0*

*\* contacto --> https://www.linkedin.com/in/david-sanchez-1366b1253/*

*\* @author David Sánchez*

*\*/*

*//Paquete al que pertenece la clase*

package Server;

*//Importes*

import java.io.Serializable;

import java.time.Duration;

import java.util.Random;

import Logger.Logger;

*//CLASE*

public class Empaquetador extends Thread implements Serializable

{

*//Constantes de la clase*

    private final int cantidadRecogidaGalletas = 20;

    private final int cantidadEmpaquetadoGalletas = 100;

*//Atributos de la clase*

    private int identificador;

    private int galletasRecogidas = 0;

    private int tandasGalletasRecogidas = 0;

    private Horno horno;                        *//Los empaquetadores tienen un horno asociado a el*

    private Almacen almacen;

    private String accion = "ESPERANDO";        *//Por defecto cuando se cree el objeto estara esperando*

*//Con esta semilla generaremos la aleatoriedad*

    private Random aleatorio = new Random();

*//Para los logs*

    private Logger log = Logger.getInstance();

*//Contructor*

    public Empaquetador(int \_identificador, Almacen \_almacen)

    {

        this.identificador = \_identificador;

        this.almacen = \_almacen;

    }

*//Contructor vacio*

*//Contructores alternativos o sobrecargados*

*//Getters*

    public int getIdentificador(){return identificador;}

    public String getAccion(){return accion;}

    public int getTandasGalletasRecogidas(){return tandasGalletasRecogidas;}

*//Setters*

*//To String (Solo si es necesario)*

    public String toStringIdentificador()

    {

        return "Empaquetador ["+identificador+"]";

    }

*//Metodos de la clase*

    public void vaciarHorno(Horno horno)

    {

        boolean hornoVaciado = false;

        while(!hornoVaciado)

        {

            try

            {

*//Antes de empezar a empaquetar no habra recogido ninguna tanda aun*

                tandasGalletasRecogidas = 0;

                accion = "EMPAQUETANDO";

                log.log("Empaquetador["+identificador+"] -->"+accion);

*//Realizamos las tandas de 5 en 5 para que las empaquetemos siempre de 100 en 100*

                for(tandasGalletasRecogidas = 1; tandasGalletasRecogidas < 6; tandasGalletasRecogidas++)

                {

*//Tanda de recogida galletas*

                    galletasRecogidas += horno.sacarHorneadoGalletas(cantidadRecogidaGalletas);

                    log.log("Empaquetador["+identificador+"] =========> Ha sacado "+cantidadRecogidaGalletas+" galletas del Horno["+horno.getIdentificador()+"]");

                    Thread.sleep(500 + aleatorio.nextInt(1000));

                }

*//Empaquetamos las galletas para llevarlas al almacen*

                accion = "TRANSPORTANDO";

                log.log("Empaquetador["+identificador+"] -->"+accion);

*//Cuando termina de transportar, las tandas recogidas vuelve a ser 0*

                tandasGalletasRecogidas = 0;

                Thread.sleep(2000 + aleatorio.nextInt(4000));

                almacen.añadirGalletas(cantidadEmpaquetadoGalletas);

                log.log("Empaquetador["+identificador+"] =========> Ha transportado "+cantidadEmpaquetadoGalletas+" galletas  al almacen");

*//Comprobamos si ha terminado de vaciar el horno*

                if(galletasRecogidas == horno.getCapacidadMaxima())

                {

                    hornoVaciado = true;

                }

            }

            catch(InterruptedException error)

            {

                System.out.println("Se ha generado un error mientras el empaquetador["+identificador+"] estaba vaciando el Horno["+horno.getIdentificador()+"] --> " + error);

            }

        }

*//Como ya ha terminado su vaciado de horno, restauramos las galletas recogidas a 0*

        galletasRecogidas = 0;

        accion = "ESPERANDO";

        log.log("Empaquetador["+identificador+"] -->"+accion);

    }

}

### 5.1.5 Almacen.java

*/\*\**

*\* version 1.0*

*\* contacto --> https://www.linkedin.com/in/david-sanchez-1366b1253/*

*\* @author David Sánchez*

*\*/*

*//Paquete al que pertenece la clase*

package Server;

*//Importes*

import java.io.Serializable;

import java.util.concurrent.locks.ReentrantLock;

public class Almacen implements Serializable

{

*//Constantes de la clase*

    private final int capacidadMaxima = 1000;

    private final int cantidadGalletasConsumidas = 100;

*//Atributos de la clase*

    private int galletasTotal = 0;

    private int galletasTotalConsumidas = 0;

    private int galletasTotalAlmacenadas = 0;

*//Contructor vacio*

    public Almacen(){};

*//Contructores alternativos o sobrecargados*

*//Getters*

    public int getCapacidadMaxima(){return capacidadMaxima;}

    public int getGalletasTotal(){return galletasTotal;}

    public int getCantidadGalletasConsumidas() {return cantidadGalletasConsumidas;}

    public int getGalletasTotalConsumidas(){return galletasTotalConsumidas;}

    public int getGalletasTotalAlmacenadas(){return galletasTotalAlmacenadas;}

*//Setters*

*//To String (Solo si es necesario)*

*//Metodos de la clase*

    public synchronized void añadirGalletas(int galletas)

    {

*//Revisamos si el almacen esta lleno*

        while(galletasTotal == capacidadMaxima)

        {

*//Espera activa*

            try

            {

                Thread.sleep(1000);

            }

            catch(InterruptedException error)

            {

                System.out.println("Se ha producido un error mientras se realizaba una espera activa para poder añadir galletas al almacen --> " + error);

            }

        }

        galletasTotal += galletas;

*//Log de almacenamiento total*

        galletasTotalAlmacenadas += galletas;

    }

    public void consumirGalletas(int cantidadGalletasConsumir)

    {

        int galletasRestantes = galletasTotal - cantidadGalletasConsumir;

*//Comprobamos que si es inferior a 0, se establezca en 0*

        if (galletasRestantes < 0)

        {

            galletasTotal = 0;

        }

        else

        {

            galletasTotal = galletasRestantes;

            galletasTotalConsumidas += cantidadGalletasConsumir;

        }

    }

}

### 5.1.6 Cafeteria.java

*/\*\**

*\* version 1.0*

*\* contacto --> https://www.linkedin.com/in/david-sanchez-1366b1253/*

*\* @author David Sánchez*

*\*/*

*//Paquete al que pertenece la clase*

package Server;

*//Importes*

import java.util.List;

import java.util.ArrayList;

import java.util.concurrent.locks.ReentrantLock;

import java.util.concurrent.Semaphore;

import java.io.Serializable;

import Logger.Logger;

*//CLASE*

public class Cafeteria implements Serializable

{

*//Atributos*

    private List<Repostero> colaCafeteria = new ArrayList<>();      *//La cola en la creacion del objeto estara vacia por defecto*

    private ReentrantLock lockColaCafeteria = new ReentrantLock();

    private Semaphore semaforoClienteAtendido = new Semaphore(1, true);     *//Generamos que se atienda solo y solo a un Repostero y que se les atienda en funcion del orden de llegada*

    private String reposteroAtendido;

    private String colaReposterosCafeteria;

*//Para lo logs*

    private Logger log = Logger.getInstance();

*//Contructor vacio*

    public Cafeteria(){}

*//Contructores alternativos o sobrecargados*

*//Getters*

    public String getReposteroAtendido(){return reposteroAtendido;}

    public String getColaReposterosCafeteria(){return colaReposterosCafeteria;}

*//Setters*

*//ToString*

    public String toStringColaCafeteria()

    {

        String cadena = "";

        for (int indice = 0; indice < colaCafeteria.size(); indice++)

        {

            cadena += colaCafeteria.get(indice).toString() + " --> ";

        }

        return cadena;

    }

*//Metodos de la clase*

    public void tomarCafe(Repostero repostero)

    {

*//Sacamos el identificador del repostero para usarlo a la hora de cambiar el texto en la interfaz*

        int identificadorRepostero = repostero.getIdentificador();

*//Marcamos como que el repostero acaba de entrar en la cafeteria*

        repostero.setAccion("CAFETERÍA");

        log.log("Repostero["+repostero.getIdentificador()+"] -->"+repostero.getAccion());

*//Primero nos aseguramos de que la cola sea escrita correctamente*

        try

        {

            lockColaCafeteria.lock();

            colaCafeteria.add(repostero);

            colaReposterosCafeteria = toStringColaCafeteria();

        }

        catch(Error error)

        {

            System.out.println("Se ha producido un error mientras el repostero["+repostero.toStringIdentificador()+"] estaba añadiendose a la cola de la cafeteria --> " + error);

        }

        finally

        {

*//Nos aseguramos de liberar el cerrojo siempre*

            lockColaCafeteria.unlock();

        }

*//Una vez nos hemos añadido a la cola de la cafeteria adquirimos nuestro Ticket*

*//Pidiendo aqui el Ticket nos aseguramos de que no se cuele nadie y el orden se mantenga segun lo imprimimos en el JTextField de la cola de la cafeteria*

        try

        {

            semaforoClienteAtendido.acquire();

*//Cuando entro aqui significa que estoy siendo atendido, entonces me quito de la cola y actualizo el texto*

            try

            {

                lockColaCafeteria.lock();

                colaCafeteria.remove(repostero);

                colaReposterosCafeteria = toStringColaCafeteria();

            }

            catch(Error error)

            {

                System.out.println("Se ha producido un error mientras el repostero["+repostero.toStringIdentificador()+"] estaba quitandose de la cola de la cafeteria --> " + error);

            }

            finally

            {

                lockColaCafeteria.unlock();

            }

*//Estando aqui ya se que puedo marcarme como que estoy siendo atendido*

            reposteroAtendido = repostero.toString();

*//Me preparo el cafe*

            Thread.sleep(2000);

        }

        catch(InterruptedException error)

        {

            System.out.println("Se ha producido un error mientras el repostero["+repostero.toStringIdentificador()+"] estaba cogiendo un ticket para la cafeteria y preparandose el cafe --> " + error);

        }

        finally

        {

*//Si ocurriese un error muy inesperado nos aseguramos de unlockear el candado*

            if(lockColaCafeteria.isLocked()){lockColaCafeteria.unlock();}

*//Una vez tengo mi cafe preparado salgo de la cafeteria*

            reposteroAtendido = "";

*//Nos aseguramos de confirmar que hemos sido atentidos siempre*

            semaforoClienteAtendido.release();

        }

    }

}

### 5.1.7 MetodosParaCliente.java

*/\*\**

*\* version 1.0*

*\* contacto --> https://www.linkedin.com/in/david-sanchez-1366b1253/*

*\* @author David Sánchez*

*\*/*

*//Paquete al que pertenece la clase*

package Server;

*//Importes*

import SharedZone.InterfazImplementacionRMI;

import java.rmi.server.UnicastRemoteObject;

import java.rmi.RemoteException;

import java.util.ArrayList;

import java.util.List;

*//CLASE*

public class MetodosParaCliente extends UnicastRemoteObject implements InterfazImplementacionRMI

{

    private List<Repostero> listaReposteros = new ArrayList<>();

    private List<Horno> listaHornos = new ArrayList<>();

    private List<Empaquetador> listaEmpaquetadores = new ArrayList<>();

    private Cafeteria cafeteria;

    private Almacen almacen;

*//Contructor*

    public MetodosParaCliente (List<Repostero> \_listaReposteros, List<Horno> \_listaHornos, List<Empaquetador> \_listaEmpaquetadores, Cafeteria \_cafeteria, Almacen \_almacen) throws RemoteException

    {

        this.listaReposteros = \_listaReposteros;

        this.listaHornos = \_listaHornos;

        this.listaEmpaquetadores = \_listaEmpaquetadores;

        this.cafeteria = \_cafeteria;

        this.almacen = \_almacen;

    }

*//GETTERS (ESENCIALES)*

    public int getTotalGalletasProducidas(int indiceRepostero){return listaReposteros.get(indiceRepostero).getTotalGalletas();}

    public int getTotalGalletasDesperdiciadas(int indiceRepostero){return listaReposteros.get(indiceRepostero).getTotalGalletasDesperdiciadas();}

    public List<Repostero> getListaReposteros(){return listaReposteros;}

    public List<Horno> getListaHornos(){return listaHornos;}

    public List<Empaquetador> getListaEmpaquetadores(){return listaEmpaquetadores;}

    public Almacen getAlmacen(){return almacen;}

    public void setParadaManualDesdeCliente(boolean \_paradaManual, int indexRepostero)

    {

        getListaReposteros().get(indexRepostero).setParadaManual(\_paradaManual);

    }

}

## 5.2 ServerInterface

### 5.2.1 PanelSCM.java

/\*\*

\* version 1.0

\* contacto --> https://www.linkedin.com/in/david-sanchez-1366b1253/

\* @author David Sánchez

\*/

//Paquete al que pertenece la clase

package ServerInterface;

//Importes

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.List;

import Server.Repostero;

import Server.Empaquetador;

import Server.Cafeteria;

import Server.Horno;

import Server.Almacen;

//CLASE

public class PanelSCM extends javax.swing.JFrame implements Runnable{

//Atributos

//Guardamos las posiciones del mouse en la pantalla

private int xMouse;

private int yMouse;

private List<Repostero> listaReposteros;

private List<Horno> listaHornos;

private List<Empaquetador> listaEmpaquetadores;

private Cafeteria cafeteria;

private Almacen almacen;

//Hilos para las barras de progres

Thread hiloBarraProgresoHorno1 = null;

Thread hiloBarraProgresoHorno2 = null;

Thread hiloBarraProgresoHorno3 = null;

public PanelSCM(List<Repostero> \_listaReposteros, List<Horno> \_listaHorno, List<Empaquetador> \_listaEmpaquetadores, Cafeteria \_cafeteria, Almacen \_almacen) {

initComponents();

Image icono = Toolkit.getDefaultToolkit().getImage(PanelSCM.class.getResource("/Fototeca/iconoAplicacion.png"));

setIconImage(icono);

//Centramos la ventana en la pantalla del usuario que ejecute la aplicacion

setLocationRelativeTo(null);

//Asignamos los atributos

this.listaReposteros = \_listaReposteros;

this.listaHornos = \_listaHorno;

this.listaEmpaquetadores = \_listaEmpaquetadores;

this.cafeteria = \_cafeteria;

this.almacen = \_almacen;

//Para que cuando vuelva de minimizarse se coloque bien

this.addWindowStateListener(e -> {

if (e.getNewState() == JFrame.NORMAL) {

setLocationRelativeTo(null); // Centra la ventana en la pantalla

}

});

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

backgroung = new javax.swing.JPanel();

separadorHeader = new javax.swing.JSeparator();

separadorMenu = new javax.swing.JSeparator();

header = new javax.swing.JPanel();

labelCerrar = new javax.swing.JLabel();

iconoCookienFactory = new javax.swing.JLabel();

panelSCM = new javax.swing.JLabel();

iconoMinimizado = new javax.swing.JLabel();

lableCafeteria = new javax.swing.JLabel();

respostero1 = new javax.swing.JLabel();

respostero2 = new javax.swing.JLabel();

respostero3 = new javax.swing.JLabel();

respostero4 = new javax.swing.JLabel();

respostero5 = new javax.swing.JLabel();

horno1 = new javax.swing.JLabel();

horno2 = new javax.swing.JLabel();

horno3 = new javax.swing.JLabel();

empaquetador1 = new javax.swing.JLabel();

empaquetador2 = new javax.swing.JLabel();

empaquetador3 = new javax.swing.JLabel();

menuDerecha = new javax.swing.JPanel();

almacenJlabel = new javax.swing.JLabel();

logoMedioGrande = new javax.swing.JLabel();

textoAlmacen = new javax.swing.JTextField();

botonComer = new javax.swing.JButton();

textoRepostero1 = new javax.swing.JTextField();

textoRepostero2 = new javax.swing.JTextField();

textoRepostero3 = new javax.swing.JTextField();

textoRepostero4 = new javax.swing.JTextField();

textoRepostero5 = new javax.swing.JTextField();

textoCafeteria = new javax.swing.JTextField();

textoColaCafeteria = new javax.swing.JTextField();

imagenCafeteria = new javax.swing.JLabel();

imagenCartelCafeteria = new javax.swing.JLabel();

labelNumeroDeGalletasHornos = new javax.swing.JLabel();

labelHorneandoHornos = new javax.swing.JLabel();

numeroDeGalletasHorno1 = new javax.swing.JTextField();

numeroDeGalletasHorno2 = new javax.swing.JTextField();

numeroDeGalletasHorno3 = new javax.swing.JTextField();

jSeparator1 = new javax.swing.JSeparator();

accionEmpaquetador2 = new javax.swing.JTextField();

accionEmpaquetador3 = new javax.swing.JTextField();

accionEmpaquetador1 = new javax.swing.JTextField();

paquete1\_1 = new javax.swing.JTextField();

paquete1\_2 = new javax.swing.JTextField();

paquete1\_3 = new javax.swing.JTextField();

paquete1\_4 = new javax.swing.JTextField();

paquete1\_5 = new javax.swing.JTextField();

paquete2\_1 = new javax.swing.JTextField();

paquete2\_2 = new javax.swing.JTextField();

paquete2\_3 = new javax.swing.JTextField();

paquete2\_4 = new javax.swing.JTextField();

paquete2\_5 = new javax.swing.JTextField();

paquete3\_1 = new javax.swing.JTextField();

paquete3\_2 = new javax.swing.JTextField();

paquete3\_3 = new javax.swing.JTextField();

paquete3\_4 = new javax.swing.JTextField();

paquete3\_5 = new javax.swing.JTextField();

labelEstadoHornos = new javax.swing.JLabel();

estadoHorno3 = new javax.swing.JTextField();

estadoHorno1 = new javax.swing.JTextField();

estadoHorno2 = new javax.swing.JTextField();

barraProgresoHorno1 = new javax.swing.JProgressBar();

barraProgresoHorno2 = new javax.swing.JProgressBar();

barraProgresoHorno3 = new javax.swing.JProgressBar();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setLocationByPlatform(true);

setUndecorated(true);

setResizable(false);

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

backgroung.setForeground(new java.awt.Color(0, 0, 0));

backgroung.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

separadorHeader.setBorder(null);

separadorHeader.setName(""); // NOI18N

backgroung.add(separadorHeader, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 58, 1390, 10));

separadorMenu.setOrientation(javax.swing.SwingConstants.VERTICAL);

separadorMenu.setBorder(null);

separadorMenu.setName(""); // NOI18N

backgroung.add(separadorMenu, new org.netbeans.lib.awtextra.AbsoluteConstraints(1118, 60, 20, 630));

header.setBackground(new java.awt.Color(64, 166, 38));

header.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {

public void mouseDragged(java.awt.event.MouseEvent evt) {

headerMouseDragged(evt);

}

});

header.addMouseListener(new java.awt.event.MouseAdapter() {

public void mousePressed(java.awt.event.MouseEvent evt) {

headerMousePressed(evt);

}

});

labelCerrar.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/IconoCerrar.png"))); // NOI18N

labelCerrar.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

labelCerrar.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

labelCerrarMouseClicked(evt);

}

public void mouseEntered(java.awt.event.MouseEvent evt) {

labelCerrarMouseEntered(evt);

}

public void mouseExited(java.awt.event.MouseEvent evt) {

labelCerrarMouseExited(evt);

}

});

iconoCookienFactory.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/logoCookeiFactorySinFondo.png"))); // NOI18N

panelSCM.setFont(new java.awt.Font("Arial Black", 0, 36)); // NOI18N

panelSCM.setForeground(new java.awt.Color(0, 0, 0));

panelSCM.setText("Panel SCM");

iconoMinimizado.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/iconoMinimizado.PNG"))); // NOI18N

iconoMinimizado.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

iconoMinimizado.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

iconoMinimizadoMouseClicked(evt);

}

public void mouseEntered(java.awt.event.MouseEvent evt) {

iconoMinimizadoMouseEntered(evt);

}

public void mouseExited(java.awt.event.MouseEvent evt) {

iconoMinimizadoMouseExited(evt);

}

});

javax.swing.GroupLayout headerLayout = new javax.swing.GroupLayout(header);

header.setLayout(headerLayout);

headerLayout.setHorizontalGroup(

headerLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(headerLayout.createSequentialGroup()

.addComponent(iconoCookienFactory)

.addGap(455, 455, 455)

.addComponent(panelSCM, javax.swing.GroupLayout.PREFERRED\_SIZE, 266, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 495, Short.MAX\_VALUE)

.addComponent(iconoMinimizado, javax.swing.GroupLayout.PREFERRED\_SIZE, 47, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(labelCerrar)

.addGap(411, 411, 411))

);

headerLayout.setVerticalGroup(

headerLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(headerLayout.createSequentialGroup()

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

.addComponent(iconoCookienFactory, javax.swing.GroupLayout.PREFERRED\_SIZE, 0, Short.MAX\_VALUE)

.addGroup(headerLayout.createSequentialGroup()

.addComponent(labelCerrar, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, Short.MAX\_VALUE))

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

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(panelSCM)))

.addContainerGap())

.addGroup(headerLayout.createSequentialGroup()

.addGap(18, 18, 18)

.addComponent(iconoMinimizado, javax.swing.GroupLayout.PREFERRED\_SIZE, 16, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

backgroung.add(header, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1800, 60));

lableCafeteria.setBackground(new java.awt.Color(0, 0, 0));

lableCafeteria.setFont(new java.awt.Font("Arial Black", 0, 18)); // NOI18N

lableCafeteria.setForeground(new java.awt.Color(0, 0, 0));

lableCafeteria.setText("Cafeteria");

backgroung.add(lableCafeteria, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 70, 110, 30));

respostero1.setFont(new java.awt.Font("Arial Black", 0, 18)); // NOI18N

respostero1.setForeground(new java.awt.Color(0, 0, 0));

respostero1.setText("Repostero 1");

backgroung.add(respostero1, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 150, 130, 40));

respostero2.setFont(new java.awt.Font("Arial Black", 0, 18)); // NOI18N

respostero2.setForeground(new java.awt.Color(0, 0, 0));

respostero2.setText("Repostero 2");

backgroung.add(respostero2, new org.netbeans.lib.awtextra.AbsoluteConstraints(290, 150, 130, 40));

respostero3.setFont(new java.awt.Font("Arial Black", 0, 18)); // NOI18N

respostero3.setForeground(new java.awt.Color(0, 0, 0));

respostero3.setText("Repostero 3");

backgroung.add(respostero3, new org.netbeans.lib.awtextra.AbsoluteConstraints(500, 150, 130, 40));

respostero4.setFont(new java.awt.Font("Arial Black", 0, 18)); // NOI18N

respostero4.setForeground(new java.awt.Color(0, 0, 0));

respostero4.setText("Repostero 4");

backgroung.add(respostero4, new org.netbeans.lib.awtextra.AbsoluteConstraints(710, 150, 130, 40));

respostero5.setFont(new java.awt.Font("Arial Black", 0, 18)); // NOI18N

respostero5.setForeground(new java.awt.Color(0, 0, 0));

respostero5.setText("Repostero 5");

backgroung.add(respostero5, new org.netbeans.lib.awtextra.AbsoluteConstraints(920, 150, 130, 40));

horno1.setFont(new java.awt.Font("Arial Black", 0, 18)); // NOI18N

horno1.setForeground(new java.awt.Color(0, 0, 0));

horno1.setText("Horno 1");

backgroung.add(horno1, new org.netbeans.lib.awtextra.AbsoluteConstraints(310, 270, 80, 40));

horno2.setFont(new java.awt.Font("Arial Black", 0, 18)); // NOI18N

horno2.setForeground(new java.awt.Color(0, 0, 0));

horno2.setText("Horno 2");

backgroung.add(horno2, new org.netbeans.lib.awtextra.AbsoluteConstraints(520, 270, 80, 40));

horno3.setFont(new java.awt.Font("Arial Black", 0, 18)); // NOI18N

horno3.setForeground(new java.awt.Color(0, 0, 0));

horno3.setText("Horno 3");

backgroung.add(horno3, new org.netbeans.lib.awtextra.AbsoluteConstraints(730, 270, 80, 40));

empaquetador1.setFont(new java.awt.Font("Arial Black", 0, 18)); // NOI18N

empaquetador1.setForeground(new java.awt.Color(0, 0, 0));

empaquetador1.setText("Empaquetador 1");

backgroung.add(empaquetador1, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 500, 170, 40));

empaquetador2.setFont(new java.awt.Font("Arial Black", 0, 18)); // NOI18N

empaquetador2.setForeground(new java.awt.Color(0, 0, 0));

empaquetador2.setText("Empaquetador 2");

backgroung.add(empaquetador2, new org.netbeans.lib.awtextra.AbsoluteConstraints(480, 500, 170, 40));

empaquetador3.setFont(new java.awt.Font("Arial Black", 0, 18)); // NOI18N

empaquetador3.setForeground(new java.awt.Color(0, 0, 0));

empaquetador3.setText("Empaquetador 3");

backgroung.add(empaquetador3, new org.netbeans.lib.awtextra.AbsoluteConstraints(880, 500, 170, 40));

menuDerecha.setBackground(new java.awt.Color(64, 166, 38));

almacenJlabel.setFont(new java.awt.Font("Arial Black", 0, 18)); // NOI18N

almacenJlabel.setForeground(new java.awt.Color(0, 0, 0));

almacenJlabel.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

almacenJlabel.setText("Almacen");

almacenJlabel.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

logoMedioGrande.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/fotoTamañoMedioCookeiFactorySinFondo.png"))); // NOI18N

textoAlmacen.setEditable(false);

textoAlmacen.setBackground(new java.awt.Color(51, 204, 0));

textoAlmacen.setFont(new java.awt.Font("sansserif", 0, 36)); // NOI18N

textoAlmacen.setForeground(new java.awt.Color(0, 0, 0));

textoAlmacen.setHorizontalAlignment(javax.swing.JTextField.CENTER);

textoAlmacen.setFocusable(false);

textoAlmacen.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

textoAlmacenActionPerformed(evt);

}

});

botonComer.setBackground(new java.awt.Color(64, 166, 38));

botonComer.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/botonComerSinFondo.PNG"))); // NOI18N

botonComer.setBorder(null);

botonComer.setBorderPainted(false);

botonComer.setContentAreaFilled(false);

botonComer.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

botonComer.setFocusPainted(false);

botonComer.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

botonComerMouseClicked(evt);

}

public void mouseEntered(java.awt.event.MouseEvent evt) {

botonComerMouseEntered(evt);

}

public void mouseExited(java.awt.event.MouseEvent evt) {

botonComerMouseExited(evt);

}

public void mousePressed(java.awt.event.MouseEvent evt) {

botonComerMousePressed(evt);

}

});

botonComer.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

botonComerActionPerformed(evt);

}

});

javax.swing.GroupLayout menuDerechaLayout = new javax.swing.GroupLayout(menuDerecha);

menuDerecha.setLayout(menuDerechaLayout);

menuDerechaLayout.setHorizontalGroup(

menuDerechaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

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

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(menuDerechaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addComponent(almacenJlabel, javax.swing.GroupLayout.DEFAULT\_SIZE, 128, Short.MAX\_VALUE)

.addComponent(textoAlmacen, javax.swing.GroupLayout.Alignment.LEADING))

.addGap(81, 81, 81))

.addGroup(menuDerechaLayout.createSequentialGroup()

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

.addGroup(menuDerechaLayout.createSequentialGroup()

.addGap(27, 27, 27)

.addComponent(logoMedioGrande, javax.swing.GroupLayout.PREFERRED\_SIZE, 225, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(menuDerechaLayout.createSequentialGroup()

.addGap(72, 72, 72)

.addComponent(botonComer, javax.swing.GroupLayout.PREFERRED\_SIZE, 109, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

menuDerechaLayout.setVerticalGroup(

menuDerechaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(menuDerechaLayout.createSequentialGroup()

.addContainerGap(154, Short.MAX\_VALUE)

.addComponent(almacenJlabel, javax.swing.GroupLayout.PREFERRED\_SIZE, 40, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(textoAlmacen, javax.swing.GroupLayout.PREFERRED\_SIZE, 64, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(botonComer, javax.swing.GroupLayout.PREFERRED\_SIZE, 101, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(46, 46, 46)

.addComponent(logoMedioGrande, javax.swing.GroupLayout.PREFERRED\_SIZE, 180, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(21, 21, 21))

);

backgroung.add(menuDerecha, new org.netbeans.lib.awtextra.AbsoluteConstraints(1120, 60, 270, 630));

textoRepostero1.setEditable(false);

textoRepostero1.setBackground(new java.awt.Color(238, 212, 130));

textoRepostero1.setForeground(new java.awt.Color(0, 0, 0));

textoRepostero1.setHorizontalAlignment(javax.swing.JTextField.CENTER);

textoRepostero1.setFocusable(false);

textoRepostero1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

textoRepostero1ActionPerformed(evt);

}

});

backgroung.add(textoRepostero1, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 190, 120, -1));

textoRepostero2.setEditable(false);

textoRepostero2.setBackground(new java.awt.Color(238, 212, 130));

textoRepostero2.setForeground(new java.awt.Color(0, 0, 0));

textoRepostero2.setHorizontalAlignment(javax.swing.JTextField.CENTER);

textoRepostero2.setFocusable(false);

textoRepostero2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

textoRepostero2ActionPerformed(evt);

}

});

backgroung.add(textoRepostero2, new org.netbeans.lib.awtextra.AbsoluteConstraints(290, 190, 120, -1));

textoRepostero3.setEditable(false);

textoRepostero3.setBackground(new java.awt.Color(238, 212, 130));

textoRepostero3.setForeground(new java.awt.Color(0, 0, 0));

textoRepostero3.setHorizontalAlignment(javax.swing.JTextField.CENTER);

textoRepostero3.setFocusable(false);

textoRepostero3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

textoRepostero3ActionPerformed(evt);

}

});

backgroung.add(textoRepostero3, new org.netbeans.lib.awtextra.AbsoluteConstraints(500, 190, 120, -1));

textoRepostero4.setEditable(false);

textoRepostero4.setBackground(new java.awt.Color(238, 212, 130));

textoRepostero4.setForeground(new java.awt.Color(0, 0, 0));

textoRepostero4.setHorizontalAlignment(javax.swing.JTextField.CENTER);

textoRepostero4.setFocusable(false);

textoRepostero4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

textoRepostero4ActionPerformed(evt);

}

});

backgroung.add(textoRepostero4, new org.netbeans.lib.awtextra.AbsoluteConstraints(710, 190, 120, -1));

textoRepostero5.setEditable(false);

textoRepostero5.setBackground(new java.awt.Color(238, 212, 130));

textoRepostero5.setForeground(new java.awt.Color(0, 0, 0));

textoRepostero5.setHorizontalAlignment(javax.swing.JTextField.CENTER);

textoRepostero5.setFocusable(false);

textoRepostero5.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

textoRepostero5ActionPerformed(evt);

}

});

backgroung.add(textoRepostero5, new org.netbeans.lib.awtextra.AbsoluteConstraints(920, 190, 120, -1));

textoCafeteria.setEditable(false);

textoCafeteria.setBackground(new java.awt.Color(238, 212, 130));

textoCafeteria.setForeground(new java.awt.Color(0, 0, 0));

textoCafeteria.setHorizontalAlignment(javax.swing.JTextField.CENTER);

textoCafeteria.setFocusable(false);

textoCafeteria.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

textoCafeteriaActionPerformed(evt);

}

});

backgroung.add(textoCafeteria, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 100, 120, -1));

textoColaCafeteria.setEditable(false);

textoColaCafeteria.setBackground(new java.awt.Color(238, 212, 130));

textoColaCafeteria.setForeground(new java.awt.Color(0, 0, 0));

textoColaCafeteria.setHorizontalAlignment(javax.swing.JTextField.LEFT);

textoColaCafeteria.setFocusable(false);

textoColaCafeteria.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

textoColaCafeteriaActionPerformed(evt);

}

});

backgroung.add(textoColaCafeteria, new org.netbeans.lib.awtextra.AbsoluteConstraints(290, 100, 750, -1));

imagenCafeteria.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/cafeteria.PNG"))); // NOI18N

backgroung.add(imagenCafeteria, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 80, -1, 50));

imagenCartelCafeteria.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/cartelCafeteria.PNG"))); // NOI18N

backgroung.add(imagenCartelCafeteria, new org.netbeans.lib.awtextra.AbsoluteConstraints(220, 80, 60, 50));

labelNumeroDeGalletasHornos.setFont(new java.awt.Font("Arial Black", 0, 18)); // NOI18N

labelNumeroDeGalletasHornos.setForeground(new java.awt.Color(0, 0, 0));

labelNumeroDeGalletasHornos.setText("Numero Galletas");

backgroung.add(labelNumeroDeGalletasHornos, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 350, 170, 40));

labelHorneandoHornos.setFont(new java.awt.Font("Arial Black", 0, 18)); // NOI18N

labelHorneandoHornos.setForeground(new java.awt.Color(0, 0, 0));

labelHorneandoHornos.setText("Horneando");

backgroung.add(labelHorneandoHornos, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 420, 170, 40));

numeroDeGalletasHorno1.setEditable(false);

numeroDeGalletasHorno1.setBackground(new java.awt.Color(238, 212, 130));

numeroDeGalletasHorno1.setForeground(new java.awt.Color(0, 0, 0));

numeroDeGalletasHorno1.setHorizontalAlignment(javax.swing.JTextField.CENTER);

numeroDeGalletasHorno1.setFocusable(false);

numeroDeGalletasHorno1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

numeroDeGalletasHorno1ActionPerformed(evt);

}

});

backgroung.add(numeroDeGalletasHorno1, new org.netbeans.lib.awtextra.AbsoluteConstraints(290, 360, 120, -1));

numeroDeGalletasHorno2.setEditable(false);

numeroDeGalletasHorno2.setBackground(new java.awt.Color(238, 212, 130));

numeroDeGalletasHorno2.setForeground(new java.awt.Color(0, 0, 0));

numeroDeGalletasHorno2.setHorizontalAlignment(javax.swing.JTextField.CENTER);

numeroDeGalletasHorno2.setFocusable(false);

numeroDeGalletasHorno2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

numeroDeGalletasHorno2ActionPerformed(evt);

}

});

backgroung.add(numeroDeGalletasHorno2, new org.netbeans.lib.awtextra.AbsoluteConstraints(500, 360, 120, -1));

numeroDeGalletasHorno3.setEditable(false);

numeroDeGalletasHorno3.setBackground(new java.awt.Color(238, 212, 130));

numeroDeGalletasHorno3.setForeground(new java.awt.Color(0, 0, 0));

numeroDeGalletasHorno3.setHorizontalAlignment(javax.swing.JTextField.CENTER);

numeroDeGalletasHorno3.setFocusable(false);

numeroDeGalletasHorno3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

numeroDeGalletasHorno3ActionPerformed(evt);

}

});

backgroung.add(numeroDeGalletasHorno3, new org.netbeans.lib.awtextra.AbsoluteConstraints(710, 360, 120, -1));

backgroung.add(jSeparator1, new org.netbeans.lib.awtextra.AbsoluteConstraints(290, 400, 540, 10));

accionEmpaquetador2.setEditable(false);

accionEmpaquetador2.setBackground(new java.awt.Color(238, 212, 130));

accionEmpaquetador2.setForeground(new java.awt.Color(255, 255, 255));

accionEmpaquetador2.setHorizontalAlignment(javax.swing.JTextField.CENTER);

accionEmpaquetador2.setFocusable(false);

accionEmpaquetador2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

accionEmpaquetador2ActionPerformed(evt);

}

});

backgroung.add(accionEmpaquetador2, new org.netbeans.lib.awtextra.AbsoluteConstraints(480, 550, 190, -1));

accionEmpaquetador3.setEditable(false);

accionEmpaquetador3.setBackground(new java.awt.Color(238, 212, 130));

accionEmpaquetador3.setForeground(new java.awt.Color(255, 255, 255));

accionEmpaquetador3.setHorizontalAlignment(javax.swing.JTextField.CENTER);

accionEmpaquetador3.setFocusable(false);

accionEmpaquetador3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

accionEmpaquetador3ActionPerformed(evt);

}

});

backgroung.add(accionEmpaquetador3, new org.netbeans.lib.awtextra.AbsoluteConstraints(880, 550, 190, -1));

accionEmpaquetador1.setEditable(false);

accionEmpaquetador1.setBackground(new java.awt.Color(238, 212, 130));

accionEmpaquetador1.setForeground(new java.awt.Color(255, 255, 255));

accionEmpaquetador1.setHorizontalAlignment(javax.swing.JTextField.CENTER);

accionEmpaquetador1.setFocusable(false);

accionEmpaquetador1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

accionEmpaquetador1ActionPerformed(evt);

}

});

backgroung.add(accionEmpaquetador1, new org.netbeans.lib.awtextra.AbsoluteConstraints(60, 550, 190, -1));

paquete1\_1.setEditable(false);

paquete1\_1.setBackground(new java.awt.Color(255, 255, 255));

paquete1\_1.setForeground(new java.awt.Color(255, 255, 255));

paquete1\_1.setHorizontalAlignment(javax.swing.JTextField.CENTER);

paquete1\_1.setFocusable(false);

paquete1\_1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

paquete1\_1ActionPerformed(evt);

}

});

backgroung.add(paquete1\_1, new org.netbeans.lib.awtextra.AbsoluteConstraints(60, 590, 30, -1));

paquete1\_2.setEditable(false);

paquete1\_2.setBackground(new java.awt.Color(255, 255, 255));

paquete1\_2.setForeground(new java.awt.Color(255, 255, 255));

paquete1\_2.setHorizontalAlignment(javax.swing.JTextField.CENTER);

paquete1\_2.setFocusable(false);

paquete1\_2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

paquete1\_2ActionPerformed(evt);

}

});

backgroung.add(paquete1\_2, new org.netbeans.lib.awtextra.AbsoluteConstraints(100, 590, 30, -1));

paquete1\_3.setEditable(false);

paquete1\_3.setBackground(new java.awt.Color(255, 255, 255));

paquete1\_3.setForeground(new java.awt.Color(255, 255, 255));

paquete1\_3.setHorizontalAlignment(javax.swing.JTextField.CENTER);

paquete1\_3.setFocusable(false);

paquete1\_3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

paquete1\_3ActionPerformed(evt);

}

});

backgroung.add(paquete1\_3, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 590, 30, -1));

paquete1\_4.setEditable(false);

paquete1\_4.setBackground(new java.awt.Color(255, 255, 255));

paquete1\_4.setForeground(new java.awt.Color(255, 255, 255));

paquete1\_4.setHorizontalAlignment(javax.swing.JTextField.CENTER);

paquete1\_4.setFocusable(false);

paquete1\_4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

paquete1\_4ActionPerformed(evt);

}

});

backgroung.add(paquete1\_4, new org.netbeans.lib.awtextra.AbsoluteConstraints(180, 590, 30, -1));

paquete1\_5.setEditable(false);

paquete1\_5.setBackground(new java.awt.Color(255, 255, 255));

paquete1\_5.setForeground(new java.awt.Color(255, 255, 255));

paquete1\_5.setHorizontalAlignment(javax.swing.JTextField.CENTER);

paquete1\_5.setFocusable(false);

paquete1\_5.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

paquete1\_5ActionPerformed(evt);

}

});

backgroung.add(paquete1\_5, new org.netbeans.lib.awtextra.AbsoluteConstraints(220, 590, 30, -1));

paquete2\_1.setEditable(false);

paquete2\_1.setBackground(new java.awt.Color(255, 255, 255));

paquete2\_1.setForeground(new java.awt.Color(255, 255, 255));

paquete2\_1.setHorizontalAlignment(javax.swing.JTextField.CENTER);

paquete2\_1.setFocusable(false);

paquete2\_1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

paquete2\_1ActionPerformed(evt);

}

});

backgroung.add(paquete2\_1, new org.netbeans.lib.awtextra.AbsoluteConstraints(480, 590, 30, -1));

paquete2\_2.setEditable(false);

paquete2\_2.setBackground(new java.awt.Color(255, 255, 255));

paquete2\_2.setForeground(new java.awt.Color(255, 255, 255));

paquete2\_2.setHorizontalAlignment(javax.swing.JTextField.CENTER);

paquete2\_2.setFocusable(false);

paquete2\_2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

paquete2\_2ActionPerformed(evt);

}

});

backgroung.add(paquete2\_2, new org.netbeans.lib.awtextra.AbsoluteConstraints(520, 590, 30, -1));

paquete2\_3.setEditable(false);

paquete2\_3.setBackground(new java.awt.Color(255, 255, 255));

paquete2\_3.setForeground(new java.awt.Color(255, 255, 255));

paquete2\_3.setHorizontalAlignment(javax.swing.JTextField.CENTER);

paquete2\_3.setFocusable(false);

paquete2\_3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

paquete2\_3ActionPerformed(evt);

}

});

backgroung.add(paquete2\_3, new org.netbeans.lib.awtextra.AbsoluteConstraints(560, 590, 30, -1));

paquete2\_4.setEditable(false);

paquete2\_4.setBackground(new java.awt.Color(255, 255, 255));

paquete2\_4.setForeground(new java.awt.Color(255, 255, 255));

paquete2\_4.setHorizontalAlignment(javax.swing.JTextField.CENTER);

paquete2\_4.setFocusable(false);

paquete2\_4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

paquete2\_4ActionPerformed(evt);

}

});

backgroung.add(paquete2\_4, new org.netbeans.lib.awtextra.AbsoluteConstraints(600, 590, 30, -1));

paquete2\_5.setEditable(false);

paquete2\_5.setBackground(new java.awt.Color(255, 255, 255));

paquete2\_5.setForeground(new java.awt.Color(255, 255, 255));

paquete2\_5.setHorizontalAlignment(javax.swing.JTextField.CENTER);

paquete2\_5.setFocusable(false);

paquete2\_5.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

paquete2\_5ActionPerformed(evt);

}

});

backgroung.add(paquete2\_5, new org.netbeans.lib.awtextra.AbsoluteConstraints(640, 590, 30, -1));

paquete3\_1.setEditable(false);

paquete3\_1.setBackground(new java.awt.Color(255, 255, 255));

paquete3\_1.setForeground(new java.awt.Color(255, 255, 255));

paquete3\_1.setHorizontalAlignment(javax.swing.JTextField.CENTER);

paquete3\_1.setFocusable(false);

paquete3\_1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

paquete3\_1ActionPerformed(evt);

}

});

backgroung.add(paquete3\_1, new org.netbeans.lib.awtextra.AbsoluteConstraints(880, 590, 30, -1));

paquete3\_2.setEditable(false);

paquete3\_2.setBackground(new java.awt.Color(255, 255, 255));

paquete3\_2.setForeground(new java.awt.Color(255, 255, 255));

paquete3\_2.setHorizontalAlignment(javax.swing.JTextField.CENTER);

paquete3\_2.setFocusable(false);

paquete3\_2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

paquete3\_2ActionPerformed(evt);

}

});

backgroung.add(paquete3\_2, new org.netbeans.lib.awtextra.AbsoluteConstraints(920, 590, 30, -1));

paquete3\_3.setEditable(false);

paquete3\_3.setBackground(new java.awt.Color(255, 255, 255));

paquete3\_3.setForeground(new java.awt.Color(255, 255, 255));

paquete3\_3.setHorizontalAlignment(javax.swing.JTextField.CENTER);

paquete3\_3.setFocusable(false);

paquete3\_3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

paquete3\_3ActionPerformed(evt);

}

});

backgroung.add(paquete3\_3, new org.netbeans.lib.awtextra.AbsoluteConstraints(960, 590, 30, -1));

paquete3\_4.setEditable(false);

paquete3\_4.setBackground(new java.awt.Color(255, 255, 255));

paquete3\_4.setForeground(new java.awt.Color(255, 255, 255));

paquete3\_4.setHorizontalAlignment(javax.swing.JTextField.CENTER);

paquete3\_4.setFocusable(false);

paquete3\_4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

paquete3\_4ActionPerformed(evt);

}

});

backgroung.add(paquete3\_4, new org.netbeans.lib.awtextra.AbsoluteConstraints(1000, 590, 30, -1));

paquete3\_5.setEditable(false);

paquete3\_5.setBackground(new java.awt.Color(255, 255, 255));

paquete3\_5.setForeground(new java.awt.Color(255, 255, 255));

paquete3\_5.setHorizontalAlignment(javax.swing.JTextField.CENTER);

paquete3\_5.setFocusable(false);

paquete3\_5.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

paquete3\_5ActionPerformed(evt);

}

});

backgroung.add(paquete3\_5, new org.netbeans.lib.awtextra.AbsoluteConstraints(1040, 590, 30, -1));

labelEstadoHornos.setFont(new java.awt.Font("Arial Black", 0, 18)); // NOI18N

labelEstadoHornos.setForeground(new java.awt.Color(0, 0, 0));

labelEstadoHornos.setText("Estado hornos");

backgroung.add(labelEstadoHornos, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 310, 170, 40));

estadoHorno3.setEditable(false);

estadoHorno3.setBackground(new java.awt.Color(238, 212, 130));

estadoHorno3.setForeground(new java.awt.Color(0, 0, 0));

estadoHorno3.setHorizontalAlignment(javax.swing.JTextField.CENTER);

estadoHorno3.setFocusable(false);

estadoHorno3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

estadoHorno3ActionPerformed(evt);

}

});

backgroung.add(estadoHorno3, new org.netbeans.lib.awtextra.AbsoluteConstraints(710, 320, 120, -1));

estadoHorno1.setEditable(false);

estadoHorno1.setBackground(new java.awt.Color(238, 212, 130));

estadoHorno1.setForeground(new java.awt.Color(0, 0, 0));

estadoHorno1.setHorizontalAlignment(javax.swing.JTextField.CENTER);

estadoHorno1.setFocusable(false);

estadoHorno1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

estadoHorno1ActionPerformed(evt);

}

});

backgroung.add(estadoHorno1, new org.netbeans.lib.awtextra.AbsoluteConstraints(290, 320, 120, -1));

estadoHorno2.setEditable(false);

estadoHorno2.setBackground(new java.awt.Color(238, 212, 130));

estadoHorno2.setForeground(new java.awt.Color(0, 0, 0));

estadoHorno2.setHorizontalAlignment(javax.swing.JTextField.CENTER);

estadoHorno2.setFocusable(false);

estadoHorno2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

estadoHorno2ActionPerformed(evt);

}

});

backgroung.add(estadoHorno2, new org.netbeans.lib.awtextra.AbsoluteConstraints(500, 320, 120, -1));

barraProgresoHorno1.setBackground(new java.awt.Color(238, 212, 130));

barraProgresoHorno1.setForeground(new java.awt.Color(68, 41, 26));

backgroung.add(barraProgresoHorno1, new org.netbeans.lib.awtextra.AbsoluteConstraints(290, 430, 120, -1));

barraProgresoHorno2.setBackground(new java.awt.Color(238, 212, 130));

barraProgresoHorno2.setForeground(new java.awt.Color(68, 41, 26));

backgroung.add(barraProgresoHorno2, new org.netbeans.lib.awtextra.AbsoluteConstraints(500, 430, 120, -1));

barraProgresoHorno3.setBackground(new java.awt.Color(238, 212, 130));

barraProgresoHorno3.setForeground(new java.awt.Color(68, 41, 26));

backgroung.add(barraProgresoHorno3, new org.netbeans.lib.awtextra.AbsoluteConstraints(710, 430, 120, -1));

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(backgroung, javax.swing.GroupLayout.PREFERRED\_SIZE, 1387, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(backgroung, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void headerMousePressed(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_headerMousePressed

xMouse = evt.getX();

yMouse = evt.getY();

}//GEN-LAST:event\_headerMousePressed

private void headerMouseDragged(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_headerMouseDragged

int x = evt.getXOnScreen();

int y = evt.getYOnScreen();

this.setLocation(x - xMouse, y - yMouse);

}//GEN-LAST:event\_headerMouseDragged

private void labelCerrarMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_labelCerrarMouseClicked

System.exit(0);

}//GEN-LAST:event\_labelCerrarMouseClicked

private void labelCerrarMouseEntered(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_labelCerrarMouseEntered

labelCerrar.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/IconoCerrar2.png")));

}//GEN-LAST:event\_labelCerrarMouseEntered

private void labelCerrarMouseExited(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_labelCerrarMouseExited

labelCerrar.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/IconoCerrar.png")));

}//GEN-LAST:event\_labelCerrarMouseExited

private void textoRepostero1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_textoRepostero1ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_textoRepostero1ActionPerformed

private void textoRepostero2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_textoRepostero2ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_textoRepostero2ActionPerformed

private void textoRepostero3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_textoRepostero3ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_textoRepostero3ActionPerformed

private void textoRepostero4ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_textoRepostero4ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_textoRepostero4ActionPerformed

private void textoRepostero5ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_textoRepostero5ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_textoRepostero5ActionPerformed

private void textoColaCafeteriaActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_textoColaCafeteriaActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_textoColaCafeteriaActionPerformed

private void textoCafeteriaActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_textoCafeteriaActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_textoCafeteriaActionPerformed

private void iconoMinimizadoMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_iconoMinimizadoMouseClicked

Point startLocation = this.getLocation();

Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

int taskbarHeight = Toolkit.getDefaultToolkit().getScreenInsets(this.getGraphicsConfiguration()).bottom;

Point endLocation = new Point(

(int) screenSize.getWidth() / 2,

(int) screenSize.getHeight() - taskbarHeight

);

Timer timer = new Timer(10, null);

timer.addActionListener(new ActionListener() {

double progress = 0;

final double step = 0.05;

@Override

public void actionPerformed(ActionEvent evt) {

progress += step;

if (progress >= 1.0) {

timer.stop();

setExtendedState(JFrame.ICONIFIED);

} else {

int x = (int) (startLocation.x + progress \* (endLocation.x - startLocation.x));

int y = (int) (startLocation.y + progress \* (endLocation.y - startLocation.y));

setLocation(x, y);

}

}

});

timer.start();

}//GEN-LAST:event\_iconoMinimizadoMouseClicked

private void iconoMinimizadoMouseEntered(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_iconoMinimizadoMouseEntered

iconoMinimizado.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/iconoMinimizado2.png")));

}//GEN-LAST:event\_iconoMinimizadoMouseEntered

private void iconoMinimizadoMouseExited(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_iconoMinimizadoMouseExited

iconoMinimizado.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/iconoMinimizado.png")));

}//GEN-LAST:event\_iconoMinimizadoMouseExited

private void numeroDeGalletasHorno1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_numeroDeGalletasHorno1ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_numeroDeGalletasHorno1ActionPerformed

private void numeroDeGalletasHorno2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_numeroDeGalletasHorno2ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_numeroDeGalletasHorno2ActionPerformed

private void numeroDeGalletasHorno3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_numeroDeGalletasHorno3ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_numeroDeGalletasHorno3ActionPerformed

private void accionEmpaquetador2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_accionEmpaquetador2ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_accionEmpaquetador2ActionPerformed

private void accionEmpaquetador3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_accionEmpaquetador3ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_accionEmpaquetador3ActionPerformed

private void accionEmpaquetador1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_accionEmpaquetador1ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_accionEmpaquetador1ActionPerformed

private void textoAlmacenActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_textoAlmacenActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_textoAlmacenActionPerformed

private void botonComerActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_botonComerActionPerformed

botonComer.setBackground(new java.awt.Color(64, 166, 38));

almacen.consumirGalletas(almacen.getCantidadGalletasConsumidas());

}//GEN-LAST:event\_botonComerActionPerformed

private void botonComerMouseEntered(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_botonComerMouseEntered

botonComer.setBackground(new java.awt.Color(64, 166, 38));

}//GEN-LAST:event\_botonComerMouseEntered

private void botonComerMouseExited(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_botonComerMouseExited

botonComer.setBackground(new java.awt.Color(64, 166, 38));

}//GEN-LAST:event\_botonComerMouseExited

private void botonComerMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_botonComerMouseClicked

botonComer.setBackground(new java.awt.Color(64, 166, 38));

}//GEN-LAST:event\_botonComerMouseClicked

private void botonComerMousePressed(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_botonComerMousePressed

botonComer.setBackground(new java.awt.Color(64, 166, 38));

}//GEN-LAST:event\_botonComerMousePressed

private void paquete1\_1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_paquete1\_1ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_paquete1\_1ActionPerformed

private void paquete1\_2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_paquete1\_2ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_paquete1\_2ActionPerformed

private void paquete1\_3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_paquete1\_3ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_paquete1\_3ActionPerformed

private void paquete1\_4ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_paquete1\_4ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_paquete1\_4ActionPerformed

private void paquete1\_5ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_paquete1\_5ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_paquete1\_5ActionPerformed

private void paquete2\_1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_paquete2\_1ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_paquete2\_1ActionPerformed

private void paquete2\_2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_paquete2\_2ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_paquete2\_2ActionPerformed

private void paquete2\_3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_paquete2\_3ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_paquete2\_3ActionPerformed

private void paquete2\_4ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_paquete2\_4ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_paquete2\_4ActionPerformed

private void paquete2\_5ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_paquete2\_5ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_paquete2\_5ActionPerformed

private void paquete3\_1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_paquete3\_1ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_paquete3\_1ActionPerformed

private void paquete3\_2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_paquete3\_2ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_paquete3\_2ActionPerformed

private void paquete3\_3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_paquete3\_3ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_paquete3\_3ActionPerformed

private void paquete3\_4ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_paquete3\_4ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_paquete3\_4ActionPerformed

private void paquete3\_5ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_paquete3\_5ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_paquete3\_5ActionPerformed

private void estadoHorno3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_estadoHorno3ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_estadoHorno3ActionPerformed

private void estadoHorno1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_estadoHorno1ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_estadoHorno1ActionPerformed

private void estadoHorno2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_estadoHorno2ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_estadoHorno2ActionPerformed

//METODOS CREADOS A MANO

public JTextField getTextosReposteros(int identificadorTexto)

{

switch (identificadorTexto)

{

case 0:

return textoRepostero1;

case 1:

return textoRepostero2;

case 2:

return textoRepostero3;

case 3:

return textoRepostero4;

case 4:

return textoRepostero5;

}

//Si no entra en ningun case, no devolvemos anda porque no han seleccionado el indice bien

return null;

}

public JTextField getTextoCafeteria(){return textoCafeteria;}

public JTextField getTextoColaCafeteria(){return textoColaCafeteria;}

public void ponerColorTextoReposteros(int identificador, String accion)

{

if (accion.equals("PRODUCIENDO")) {getTextosReposteros(identificador).setForeground(Color.black);}

else if (accion.equals("DEPOSITANDO")) {getTextosReposteros(identificador).setForeground(Color.magenta);}

else if (accion.equals("DESCANSANDO")) {getTextosReposteros(identificador).setForeground(Color.white);}

else if (accion.equals("BLOQUEADO")) {getTextosReposteros(identificador).setForeground(Color.red);}

else if (accion.equals("CAFETERÍA")) {getTextosReposteros(identificador).setForeground(Color.blue);}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(PanelSCM.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(PanelSCM.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(PanelSCM.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(PanelSCM.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

}

});

}

public JTextField getTextoGalletasEnHorno(int identificadorTexto)

{

switch (identificadorTexto)

{

case 0:

return numeroDeGalletasHorno1;

case 1:

return numeroDeGalletasHorno2;

case 2:

return numeroDeGalletasHorno3;

}

//Si no entra en ningun case, no devolvemos anda porque no han seleccionado el indice bien

return null;

}

public JTextField getTextoAccionEmpaquetadores(int identificadorTexto)

{

switch (identificadorTexto)

{

case 0:

return accionEmpaquetador1;

case 1:

return accionEmpaquetador2;

case 2:

return accionEmpaquetador3;

}

//Si no entra en ningun case, no devolvemos anda porque no han seleccionado el indice bien

return null;

}

public void ponerColorTextoAccionEmpaquetadores(int identificador, String accion)

{

if (accion.equals("EMPAQUETANDO")) {getTextoAccionEmpaquetadores(identificador).setForeground(Color.black);}

else if (accion.equals("TRANSPORTANDO")) {getTextoAccionEmpaquetadores(identificador).setForeground(Color.magenta);}

else if (accion.equals("ESPERANDO")) {getTextoAccionEmpaquetadores(identificador).setForeground(Color.white);}

}

public void checkColorPaquetes(Empaquetador empaquetador)

{

if(empaquetador.getIdentificador() == 1)

{

//Almacenamos en una variable temporal las tandas que lleva

int tandasRecogidas = empaquetador.getTandasGalletasRecogidas();

if(tandasRecogidas == 0)

{

paquete1\_1.setBackground(Color.white);

paquete1\_2.setBackground(Color.white);

paquete1\_3.setBackground(Color.white);

paquete1\_4.setBackground(Color.white);

paquete1\_5.setBackground(Color.white);

}

else if( tandasRecogidas >= 1)

{

paquete1\_1.setBackground(Color.black);

if(tandasRecogidas >= 2)

{

paquete1\_2.setBackground(Color.black);

if(tandasRecogidas >=3)

{

paquete1\_3.setBackground(Color.black);

if(tandasRecogidas >=4)

{

paquete1\_4.setBackground(Color.black);

if(tandasRecogidas >=5)

{

paquete1\_5.setBackground(Color.black);

}

}

}

}

}

}

else if(empaquetador.getIdentificador() == 2)

{

//Almacenamos en una variable temporal las tandas que lleva

int tandasRecogidas = empaquetador.getTandasGalletasRecogidas();

if(tandasRecogidas == 0)

{

paquete2\_1.setBackground(Color.white);

paquete2\_2.setBackground(Color.white);

paquete2\_3.setBackground(Color.white);

paquete2\_4.setBackground(Color.white);

paquete2\_5.setBackground(Color.white);

}

else if( tandasRecogidas >= 1)

{

paquete2\_1.setBackground(Color.black);

if(tandasRecogidas >= 2)

{

paquete2\_2.setBackground(Color.black);

if(tandasRecogidas >=3)

{

paquete2\_3.setBackground(Color.black);

if(tandasRecogidas >=4)

{

paquete2\_4.setBackground(Color.black);

if(tandasRecogidas >=5)

{

paquete2\_5.setBackground(Color.black);

}

}

}

}

}

}

else if(empaquetador.getIdentificador() == 3)

{

//Almacenamos en una variable temporal las tandas que lleva

int tandasRecogidas = empaquetador.getTandasGalletasRecogidas();

if(tandasRecogidas == 0)

{

paquete3\_1.setBackground(Color.white);

paquete3\_2.setBackground(Color.white);

paquete3\_3.setBackground(Color.white);

paquete3\_4.setBackground(Color.white);

paquete3\_5.setBackground(Color.white);

}

else if( tandasRecogidas >= 1)

{

paquete3\_1.setBackground(Color.black);

if(tandasRecogidas >= 2)

{

paquete3\_2.setBackground(Color.black);

if(tandasRecogidas >=3)

{

paquete3\_3.setBackground(Color.black);

if(tandasRecogidas >=4)

{

paquete3\_4.setBackground(Color.black);

if(tandasRecogidas >=5)

{

paquete3\_5.setBackground(Color.black);

}

}

}

}

}

}

}

public void ponerColorTextoHornos(int identificador, String accion)

{

if (accion.equals("LLENANDOSE")) {getTextoAccionHornos(identificador).setForeground(Color.black);}

else if (accion.equals("HORNEANDO")) {getTextoAccionHornos(identificador).setForeground(Color.magenta);}

else if (accion.equals("VACIANDOSE")) {getTextoAccionHornos(identificador).setForeground(Color.white);}

}

public JTextField getTextoAccionHornos(int identificadorTexto)

{

switch (identificadorTexto)

{

case 0:

return estadoHorno1;

case 1:

return estadoHorno2;

case 2:

return estadoHorno3;

}

//Si no entra en ningun case, no devolvemos anda porque no han seleccionado el indice bien

return null;

}

public void checkHornoHorneando(int indexHorno, JProgressBar progressBar)

{

//Comprobamos si el horno esta horneando

if(listaHornos.get(indexHorno).getEstaHorneando() && getHilosProgressBarHornos(indexHorno) == null)

{

//Hilo base para las progressBar

//Crear el hilo solo si no está en ejecución

Thread animacionThread = new Thread(new Runnable()

{

@Override

public void run()

{

try

{

//Tiempo total de la animación (8 segundos)

int tiempoTotal = 8000;

int unidadesPorAvance = 1; //Aumentar la barra en 1 unidad por vez (a mas unidades menos fluido)

int totalAvances = 100 / unidadesPorAvance; //Número de avances (100 avances)

//Configurar la barra para que empiece en 0

progressBar.setValue(0);

//Ciclo para actualizar la barra de progreso

for (int i = 0; i <= 100; i += unidadesPorAvance) {

progressBar.setValue(i); //Actualiza el valor de la barra

Thread.sleep(tiempoTotal / totalAvances); //Espera para cada avance (400 ms) (en este caso)

}

}

catch (InterruptedException error) {

System.out.println("Se ha producido un error mientras se elabora el hilo para el llenado de la progressBar --> " + error);

}

finally

{

// Al terminar, liberamos el hilo para que se pueda crear uno nuevo la próxima vez

if (indexHorno == 0)

{

hiloBarraProgresoHorno1 = null;

}

if (indexHorno == 1)

{

hiloBarraProgresoHorno2 = null;

}

if (indexHorno == 2)

{

hiloBarraProgresoHorno3 = null;

}

}

}

});

//Iniciar el hilo de animación

if (indexHorno == 0) {

hiloBarraProgresoHorno1 = animacionThread;

}

else if (indexHorno == 1) {

hiloBarraProgresoHorno2 = animacionThread;

}

else if (indexHorno == 2) {

hiloBarraProgresoHorno3 = animacionThread;

}

//Iniciamos el hilo

animacionThread.start();

}

else if (!listaHornos.get(indexHorno).getEstaHorneando() && !listaHornos.get(indexHorno).getEstaEmpaquetando()&& listaHornos.get(indexHorno).getAccion().equals("LLENANDOSE"))

{

//Se vacia si esta en proceso de llenado

progressBar.setValue(0);

}

}

public JProgressBar getProgressBarHornos(int identificadorHorno)

{

switch (identificadorHorno)

{

case 0:

return barraProgresoHorno1;

case 1:

return barraProgresoHorno2;

case 2:

return barraProgresoHorno3;

}

//Si no entra en ningun case, no devolvemos anda porque no han seleccionado el indice bien

return null;

}

public Thread getHilosProgressBarHornos(int identificadorHilo)

{

switch (identificadorHilo)

{

case 0:

return hiloBarraProgresoHorno1;

case 1:

return hiloBarraProgresoHorno2;

case 2:

return hiloBarraProgresoHorno3;

}

//Si no entra en ningun case, no devolvemos anda porque no han seleccionado el indice bien

return null;

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JTextField accionEmpaquetador1;

private javax.swing.JTextField accionEmpaquetador2;

private javax.swing.JTextField accionEmpaquetador3;

private javax.swing.JLabel almacenJlabel;

private javax.swing.JPanel backgroung;

private javax.swing.JProgressBar barraProgresoHorno1;

private javax.swing.JProgressBar barraProgresoHorno2;

private javax.swing.JProgressBar barraProgresoHorno3;

private javax.swing.JButton botonComer;

private javax.swing.JLabel empaquetador1;

private javax.swing.JLabel empaquetador2;

private javax.swing.JLabel empaquetador3;

private javax.swing.JTextField estadoHorno1;

private javax.swing.JTextField estadoHorno2;

private javax.swing.JTextField estadoHorno3;

private javax.swing.JPanel header;

private javax.swing.JLabel horno1;

private javax.swing.JLabel horno2;

private javax.swing.JLabel horno3;

private javax.swing.JLabel iconoCookienFactory;

private javax.swing.JLabel iconoMinimizado;

private javax.swing.JLabel imagenCafeteria;

private javax.swing.JLabel imagenCartelCafeteria;

private javax.swing.JSeparator jSeparator1;

private javax.swing.JLabel labelCerrar;

private javax.swing.JLabel labelEstadoHornos;

private javax.swing.JLabel labelHorneandoHornos;

private javax.swing.JLabel labelNumeroDeGalletasHornos;

private javax.swing.JLabel lableCafeteria;

private javax.swing.JLabel logoMedioGrande;

private javax.swing.JPanel menuDerecha;

private javax.swing.JTextField numeroDeGalletasHorno1;

private javax.swing.JTextField numeroDeGalletasHorno2;

private javax.swing.JTextField numeroDeGalletasHorno3;

private javax.swing.JLabel panelSCM;

private javax.swing.JTextField paquete1\_1;

private javax.swing.JTextField paquete1\_2;

private javax.swing.JTextField paquete1\_3;

private javax.swing.JTextField paquete1\_4;

private javax.swing.JTextField paquete1\_5;

private javax.swing.JTextField paquete2\_1;

private javax.swing.JTextField paquete2\_2;

private javax.swing.JTextField paquete2\_3;

private javax.swing.JTextField paquete2\_4;

private javax.swing.JTextField paquete2\_5;

private javax.swing.JTextField paquete3\_1;

private javax.swing.JTextField paquete3\_2;

private javax.swing.JTextField paquete3\_3;

private javax.swing.JTextField paquete3\_4;

private javax.swing.JTextField paquete3\_5;

private javax.swing.JLabel respostero1;

private javax.swing.JLabel respostero2;

private javax.swing.JLabel respostero3;

private javax.swing.JLabel respostero4;

private javax.swing.JLabel respostero5;

private javax.swing.JSeparator separadorHeader;

private javax.swing.JSeparator separadorMenu;

private javax.swing.JTextField textoAlmacen;

private javax.swing.JTextField textoCafeteria;

private javax.swing.JTextField textoColaCafeteria;

private javax.swing.JTextField textoRepostero1;

private javax.swing.JTextField textoRepostero2;

private javax.swing.JTextField textoRepostero3;

private javax.swing.JTextField textoRepostero4;

private javax.swing.JTextField textoRepostero5;

// End of variables declaration//GEN-END:variables

@Override

public void run()

{

//No queremos que el hilo pare nunca de ejecutarse

while(true)

{

try

{

//Checkeo de estado para los reposteros

for (int identificador = 0; identificador < listaReposteros.size(); identificador++)

{

String accion = listaReposteros.get(identificador).getAccion();

//Primero cambio el color del texto

ponerColorTextoReposteros(identificador, accion);

//Luego cambio el texto

getTextosReposteros(identificador).setText(accion);

}

//Checkeo la cafeteria

textoCafeteria.setText(cafeteria.getReposteroAtendido());

textoColaCafeteria.setText(cafeteria.getColaReposterosCafeteria());

//Checkeo hornos

for (int identificador = 0; identificador < listaHornos.size(); identificador++)

{

//Actualizamos las galletas que hay en cada horno

int numeroGalletas = listaHornos.get(identificador).getCantidadGalletas();

getTextoGalletasEnHorno(identificador).setText(String.valueOf(numeroGalletas));

//Actualizamos el estado de los hornos

String accion = listaHornos.get(identificador).getAccion();

//Primero cambio el color del texto

ponerColorTextoHornos(identificador, accion);

//Luego cambio el texto

getTextoAccionHornos(identificador).setText(accion);

//Actualizamos los progressBar

checkHornoHorneando(identificador, getProgressBarHornos(identificador));

}

//Checkeo empaquetadores

for (int identificador = 0; identificador < listaEmpaquetadores.size(); identificador++)

{

//Recogemos la accion que estan realizando los empaquetadores

String accion = listaEmpaquetadores.get(identificador).getAccion();

//Escribimos la accion que esten realizando

getTextoAccionEmpaquetadores(identificador).setText(accion);

//Cambiamos el color del texto

ponerColorTextoAccionEmpaquetadores(identificador, accion);

//Checkeo el color de los paquete

checkColorPaquetes(listaEmpaquetadores.get(identificador));

}

//Checkeo almacen

textoAlmacen.setText(String.valueOf(almacen.getGalletasTotal()));

//Generamos un breve retardo para no colapsar memoria con comprobaciones

Thread.sleep(100);

}

catch (InterruptedException error)

{

System.out.println("Se ha producido un error mientras se ejecutaba el hilo de la interfaz de PanelSCM --> " + error);

}

}

}

}

## 5.3 SharedZone

### 5.3.1 InterfazImplementacionRMI.java

*/\*\**

*\* version 1.0*

*\* contacto --> https://www.linkedin.com/in/david-sanchez-1366b1253/*

*\* @author David Sánchez*

*\*/*

*//Paquete al que pertenece la clase*

package SharedZone;

*//Importes*

import Server.Almacen;

import java.rmi.Remote;

import java.rmi.RemoteException;

import Server.Repostero;

import Server.Horno;

import Server.Empaquetador;

import java.util.List;

*//CLASE*

public interface InterfazImplementacionRMI extends Remote

{

*//TODOS LOS METODOS AÑADIDOS TENDRAN UN --> throws RemoteExecption*

    public int getTotalGalletasProducidas(int indiceRepostero) throws RemoteException;

    public int getTotalGalletasDesperdiciadas(int indiceRepostero) throws RemoteException;

    public List<Repostero> getListaReposteros() throws RemoteException;

    public List<Horno> getListaHornos() throws RemoteException;

    public List<Empaquetador> getListaEmpaquetadores() throws RemoteException;

    public Almacen getAlmacen()throws RemoteException;

    public void setParadaManualDesdeCliente(boolean \_paradaManual, int indexRepostero) throws RemoteException;

}

## 5.4 Client

### 5.4.1 InicializacionCliente.java

*/\*\**

*\* version 1.0*

*\* contacto --> https://www.linkedin.com/in/david-sanchez-1366b1253/*

*\* @author David Sánchez*

*\*/*

*//Paquete al que pertenece la clase*

package Client;

*//Importes*

import ClientInterface.PanelCliente;

import SharedZone.InterfazImplementacionRMI;

import java.net.MalformedURLException;

import java.rmi.RemoteException;

import java.rmi.Naming;

import java.rmi.NotBoundException;

*//CLASE*

public class InicializacionCliente

{

    public static void main(String[] args)

    {

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*\*   INICIALIZACIÓN DE CLIENTE    \**

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

        try

        {

            InterfazImplementacionRMI metodosServer = (InterfazImplementacionRMI) Naming.lookup("//localhost/objetoTraspasable");

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*\*   INICIALIZACIÓN DE INTERFAZ   \**

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*//Inicializamos la interfaz del servidor*

            PanelCliente interfazCliente = new PanelCliente(metodosServer);

            interfazCliente.setVisible(true);

            Thread hiloInterfazCliente = new Thread(interfazCliente);

            hiloInterfazCliente.start();

        }

        catch (MalformedURLException error)

        {

            System.out.println("Se ha producido un error de MalformedURLException cuando se importaba un objeto del Servidor en el cliente --> " + error );

        }

        catch (NotBoundException error)

        {

            System.out.println("Se ha producido un error de NotBoundException cuando se importaba un objeto del Servidor en el cliente --> " + error );

        }

        catch (RemoteException error)

        {

            System.out.println("Se ha producido un error de RemoteException cuando se importaba un objeto del Servidor en el cliente --> " + error );

        }

    }

}

## 5.5 ClientInterface

### 5.5.1 PanelCliente.java

*/\*\**

*\* version 1.0*

*\* contacto --> https://www.linkedin.com/in/david-sanchez-1366b1253/*

*\* @author David Sánchez*

*\*/*

*//Paquete al que pertenece la clase*

package ClientInterface;

*//Importes*

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.List;

import SharedZone.InterfazImplementacionRMI;

import java.rmi.RemoteException;

import Server.Repostero;

import Server.Horno;

import Server.Empaquetador;

import Server.Almacen;

*//CLASE*

public class PanelCliente extends javax.swing.JFrame implements Runnable{

*//Atributos*

*//Aqui guardaremos la info de todos los reposteros, hornos y empaquetadores*

    InterfazImplementacionRMI metodosServer;

*//Guardamos las posiciones del mouse en la pantalla*

    int xMouse;

    int yMouse;

    public PanelCliente(InterfazImplementacionRMI \_metodosServer) {

*//Gestiono el argumento*

        this.metodosServer = \_metodosServer;

        initComponents();

        Image icono = Toolkit.getDefaultToolkit().getImage(PanelCliente.class.getResource("/Fototeca/iconoAplicacionCliente.png"));

        setIconImage(icono);

*//Centramos la ventana en la pantalla del usuario que ejecute la aplicacion*

        setLocationRelativeTo(null);

*//Para que cuando vuelva de minimizarse se coloque bien*

        this.addWindowStateListener(e -> {

            if (e.getNewState() == JFrame.NORMAL) {

                setLocationRelativeTo(null); *// Centra la ventana en la pantalla*

            }

        });

    }

*/\*\**

*\* This method is called from within the constructor to initialize the form.*

*\* WARNING: Do NOT modify this code. The content of this method is always*

*\* regenerated by the Form Editor.*

*\*/*

    @SuppressWarnings("unchecked")

*// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents*

    private void initComponents() {

        backgroung = new javax.swing.JPanel();

        separadorHeader = new javax.swing.JSeparator();

        separadorMenu = new javax.swing.JSeparator();

        header = new javax.swing.JPanel();

        labelCerrar = new javax.swing.JLabel();

        iconoCookienFactory = new javax.swing.JLabel();

        panelCliente = new javax.swing.JLabel();

        iconoMinimizado = new javax.swing.JLabel();

        respostero1 = new javax.swing.JLabel();

        respostero2 = new javax.swing.JLabel();

        respostero3 = new javax.swing.JLabel();

        respostero4 = new javax.swing.JLabel();

        respostero5 = new javax.swing.JLabel();

        horno1 = new javax.swing.JLabel();

        horno2 = new javax.swing.JLabel();

        horno3 = new javax.swing.JLabel();

        menuDerecha = new javax.swing.JPanel();

        logoMedioGrande = new javax.swing.JLabel();

        textoAlmacen = new javax.swing.JLabel();

        textoGalletasAlmacenadas = new javax.swing.JLabel();

        textoGalletasDesperdiciadas1 = new javax.swing.JLabel();

        textoAlmacenGalletasConsumidas = new javax.swing.JTextField();

        textoAlmacenGalletasAlmacenadas = new javax.swing.JTextField();

        textoRepostero1Galletas = new javax.swing.JTextField();

        textoRepostero2Galletas = new javax.swing.JTextField();

        textoRepostero3Galletas = new javax.swing.JTextField();

        textoRepostero4Galletas = new javax.swing.JTextField();

        textoRepostero5Galletas = new javax.swing.JTextField();

        textoRepostero1GalletasDesperdiciadas = new javax.swing.JTextField();

        textoRepostero2GalletasDesperdiciadas = new javax.swing.JTextField();

        textoRepostero3GalletasDesperdiciadas = new javax.swing.JTextField();

        textoRepostero4GalletasDesperdiciadas = new javax.swing.JTextField();

        textoRepostero5GalletasDesperdiciadas = new javax.swing.JTextField();

        textoGalletasDesperdiciadas = new javax.swing.JLabel();

        textoGalletasProducidas = new javax.swing.JLabel();

        separadorColumnaReposteros = new javax.swing.JSeparator();

        textoGalletasHorneadas = new javax.swing.JLabel();

        textoHorneando = new javax.swing.JLabel();

        textoHorno1GalletasHorneadas = new javax.swing.JTextField();

        textoHorno2GalletasHorneadas = new javax.swing.JTextField();

        textoHorno3GalletasHorneadas = new javax.swing.JTextField();

        separadorColumnaHornos = new javax.swing.JSeparator();

        panelControlHorneado1 = new javax.swing.JTextField();

        panelControlHorneado2 = new javax.swing.JTextField();

        panelControlHorneado3 = new javax.swing.JTextField();

        botonPararRepostero1 = new javax.swing.JButton();

        botonPararRepostero2 = new javax.swing.JButton();

        botonPararRepostero3 = new javax.swing.JButton();

        botonPararRepostero4 = new javax.swing.JButton();

        botonPararRepostero5 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

        setLocationByPlatform(true);

        setUndecorated(true);

        setResizable(false);

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

        backgroung.setForeground(new java.awt.Color(0, 0, 0));

        backgroung.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        separadorHeader.setBorder(null);

        separadorHeader.setName(""); *// NOI18N*

        backgroung.add(separadorHeader, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 58, 1390, 10));

        separadorMenu.setOrientation(javax.swing.SwingConstants.VERTICAL);

        separadorMenu.setBorder(null);

        separadorMenu.setName(""); *// NOI18N*

        backgroung.add(separadorMenu, new org.netbeans.lib.awtextra.AbsoluteConstraints(1118, 60, 20, 630));

        header.setBackground(new java.awt.Color(64, 166, 38));

        header.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {

            public void mouseDragged(java.awt.event.MouseEvent evt) {

                headerMouseDragged(evt);

            }

        });

        header.addMouseListener(new java.awt.event.MouseAdapter() {

            public void mousePressed(java.awt.event.MouseEvent evt) {

                headerMousePressed(evt);

            }

        });

        labelCerrar.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/IconoCerrar.png"))); *// NOI18N*

        labelCerrar.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

        labelCerrar.addMouseListener(new java.awt.event.MouseAdapter() {

            public void mouseClicked(java.awt.event.MouseEvent evt) {

                labelCerrarMouseClicked(evt);

            }

            public void mouseEntered(java.awt.event.MouseEvent evt) {

                labelCerrarMouseEntered(evt);

            }

            public void mouseExited(java.awt.event.MouseEvent evt) {

                labelCerrarMouseExited(evt);

            }

        });

        iconoCookienFactory.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/logoCookeiFactorySinFondo.png"))); *// NOI18N*

        panelCliente.setFont(new java.awt.Font("Arial Black", 0, 36)); *// NOI18N*

        panelCliente.setForeground(new java.awt.Color(0, 0, 0));

        panelCliente.setText("Panel Cliente");

        iconoMinimizado.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/iconoMinimizado.PNG"))); *// NOI18N*

        iconoMinimizado.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

        iconoMinimizado.addMouseListener(new java.awt.event.MouseAdapter() {

            public void mouseClicked(java.awt.event.MouseEvent evt) {

                iconoMinimizadoMouseClicked(evt);

            }

            public void mouseEntered(java.awt.event.MouseEvent evt) {

                iconoMinimizadoMouseEntered(evt);

            }

            public void mouseExited(java.awt.event.MouseEvent evt) {

                iconoMinimizadoMouseExited(evt);

            }

        });

        javax.swing.GroupLayout headerLayout = new javax.swing.GroupLayout(header);

        header.setLayout(headerLayout);

        headerLayout.setHorizontalGroup(

            headerLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(headerLayout.createSequentialGroup()

                .addComponent(iconoCookienFactory)

                .addGap(441, 441, 441)

                .addComponent(panelCliente, javax.swing.GroupLayout.PREFERRED\_SIZE, 266, javax.swing.GroupLayout.PREFERRED\_SIZE)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 509, Short.MAX\_VALUE)

                .addComponent(iconoMinimizado, javax.swing.GroupLayout.PREFERRED\_SIZE, 47, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

                .addComponent(labelCerrar)

                .addGap(411, 411, 411))

        );

        headerLayout.setVerticalGroup(

            headerLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

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

                .addGroup(headerLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

                    .addGroup(headerLayout.createSequentialGroup()

                        .addGap(0, 0, Short.MAX\_VALUE)

                        .addComponent(panelCliente))

                    .addComponent(iconoCookienFactory, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED\_SIZE, 0, Short.MAX\_VALUE)

                    .addGroup(javax.swing.GroupLayout.Alignment.LEADING, headerLayout.createSequentialGroup()

                        .addComponent(labelCerrar, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE)

                        .addGap(0, 4, Short.MAX\_VALUE)))

                .addContainerGap())

            .addGroup(headerLayout.createSequentialGroup()

                .addGap(15, 15, 15)

                .addComponent(iconoMinimizado, javax.swing.GroupLayout.PREFERRED\_SIZE, 16, javax.swing.GroupLayout.PREFERRED\_SIZE)

                .addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

        );

        backgroung.add(header, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1800, 60));

        respostero1.setFont(new java.awt.Font("Arial Black", 0, 18)); *// NOI18N*

        respostero1.setForeground(new java.awt.Color(0, 0, 0));

        respostero1.setText("Repostero 1");

        backgroung.add(respostero1, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 140, 130, 40));

        respostero2.setFont(new java.awt.Font("Arial Black", 0, 18)); *// NOI18N*

        respostero2.setForeground(new java.awt.Color(0, 0, 0));

        respostero2.setText("Repostero 2");

        backgroung.add(respostero2, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 210, 130, 40));

        respostero3.setFont(new java.awt.Font("Arial Black", 0, 18)); *// NOI18N*

        respostero3.setForeground(new java.awt.Color(0, 0, 0));

        respostero3.setText("Repostero 3");

        backgroung.add(respostero3, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 280, 130, 40));

        respostero4.setFont(new java.awt.Font("Arial Black", 0, 18)); *// NOI18N*

        respostero4.setForeground(new java.awt.Color(0, 0, 0));

        respostero4.setText("Repostero 4");

        backgroung.add(respostero4, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 340, 130, 40));

        respostero5.setFont(new java.awt.Font("Arial Black", 0, 18)); *// NOI18N*

        respostero5.setForeground(new java.awt.Color(0, 0, 0));

        respostero5.setText("Repostero 5");

        backgroung.add(respostero5, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 400, 130, 40));

        horno1.setFont(new java.awt.Font("Arial Black", 0, 18)); *// NOI18N*

        horno1.setForeground(new java.awt.Color(0, 0, 0));

        horno1.setText("Horno 1");

        backgroung.add(horno1, new org.netbeans.lib.awtextra.AbsoluteConstraints(170, 540, 80, 40));

        horno2.setFont(new java.awt.Font("Arial Black", 0, 18)); *// NOI18N*

        horno2.setForeground(new java.awt.Color(0, 0, 0));

        horno2.setText("Horno 2");

        backgroung.add(horno2, new org.netbeans.lib.awtextra.AbsoluteConstraints(170, 590, 80, 40));

        horno3.setFont(new java.awt.Font("Arial Black", 0, 18)); *// NOI18N*

        horno3.setForeground(new java.awt.Color(0, 0, 0));

        horno3.setText("Horno 3");

        backgroung.add(horno3, new org.netbeans.lib.awtextra.AbsoluteConstraints(170, 640, 80, 40));

        menuDerecha.setBackground(new java.awt.Color(64, 166, 38));

        logoMedioGrande.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/fotoTamañoMedioCookeiFactorySinFondo.png"))); *// NOI18N*

        textoAlmacen.setFont(new java.awt.Font("Arial Black", 0, 36)); *// NOI18N*

        textoAlmacen.setForeground(new java.awt.Color(0, 0, 0));

        textoAlmacen.setText("Almacen");

        textoGalletasAlmacenadas.setFont(new java.awt.Font("Arial Black", 0, 18)); *// NOI18N*

        textoGalletasAlmacenadas.setForeground(new java.awt.Color(0, 0, 0));

        textoGalletasAlmacenadas.setText("Galletas Almacenadas");

        textoGalletasDesperdiciadas1.setFont(new java.awt.Font("Arial Black", 0, 18)); *// NOI18N*

        textoGalletasDesperdiciadas1.setForeground(new java.awt.Color(0, 0, 0));

        textoGalletasDesperdiciadas1.setText("Galletas Consumidas");

        textoAlmacenGalletasConsumidas.setEditable(false);

        textoAlmacenGalletasConsumidas.setBackground(new java.awt.Color(238, 212, 130));

        textoAlmacenGalletasConsumidas.setForeground(new java.awt.Color(0, 0, 0));

        textoAlmacenGalletasConsumidas.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        textoAlmacenGalletasConsumidas.setFocusable(false);

        textoAlmacenGalletasConsumidas.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                textoAlmacenGalletasConsumidasActionPerformed(evt);

            }

        });

        textoAlmacenGalletasAlmacenadas.setEditable(false);

        textoAlmacenGalletasAlmacenadas.setBackground(new java.awt.Color(238, 212, 130));

        textoAlmacenGalletasAlmacenadas.setForeground(new java.awt.Color(0, 0, 0));

        textoAlmacenGalletasAlmacenadas.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        textoAlmacenGalletasAlmacenadas.setFocusable(false);

        textoAlmacenGalletasAlmacenadas.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                textoAlmacenGalletasAlmacenadasActionPerformed(evt);

            }

        });

        javax.swing.GroupLayout menuDerechaLayout = new javax.swing.GroupLayout(menuDerecha);

        menuDerecha.setLayout(menuDerechaLayout);

        menuDerechaLayout.setHorizontalGroup(

            menuDerechaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(menuDerechaLayout.createSequentialGroup()

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

                    .addGroup(menuDerechaLayout.createSequentialGroup()

                        .addGap(51, 51, 51)

                        .addComponent(textoAlmacen))

                    .addGroup(menuDerechaLayout.createSequentialGroup()

                        .addGap(27, 27, 27)

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

                            .addComponent(textoGalletasAlmacenadas)

                            .addComponent(logoMedioGrande, javax.swing.GroupLayout.PREFERRED\_SIZE, 225, javax.swing.GroupLayout.PREFERRED\_SIZE)))

                    .addGroup(menuDerechaLayout.createSequentialGroup()

                        .addGap(41, 41, 41)

                        .addComponent(textoAlmacenGalletasAlmacenadas, javax.swing.GroupLayout.PREFERRED\_SIZE, 200, javax.swing.GroupLayout.PREFERRED\_SIZE))

                    .addGroup(menuDerechaLayout.createSequentialGroup()

                        .addGap(30, 30, 30)

                        .addGroup(menuDerechaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

                            .addComponent(textoGalletasDesperdiciadas1)

                            .addComponent(textoAlmacenGalletasConsumidas, javax.swing.GroupLayout.PREFERRED\_SIZE, 200, javax.swing.GroupLayout.PREFERRED\_SIZE))))

                .addContainerGap(18, Short.MAX\_VALUE))

        );

        menuDerechaLayout.setVerticalGroup(

            menuDerechaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(menuDerechaLayout.createSequentialGroup()

                .addGap(69, 69, 69)

                .addComponent(textoAlmacen)

                .addGap(28, 28, 28)

                .addComponent(textoGalletasAlmacenadas, javax.swing.GroupLayout.PREFERRED\_SIZE, 40, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

                .addComponent(textoAlmacenGalletasAlmacenadas, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

                .addGap(49, 49, 49)

                .addComponent(textoGalletasDesperdiciadas1, javax.swing.GroupLayout.PREFERRED\_SIZE, 40, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

                .addComponent(textoAlmacenGalletasConsumidas, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 71, Short.MAX\_VALUE)

                .addComponent(logoMedioGrande, javax.swing.GroupLayout.PREFERRED\_SIZE, 180, javax.swing.GroupLayout.PREFERRED\_SIZE)

                .addGap(21, 21, 21))

        );

        backgroung.add(menuDerecha, new org.netbeans.lib.awtextra.AbsoluteConstraints(1120, 60, 270, 630));

        textoRepostero1Galletas.setEditable(false);

        textoRepostero1Galletas.setBackground(new java.awt.Color(238, 212, 130));

        textoRepostero1Galletas.setForeground(new java.awt.Color(0, 0, 0));

        textoRepostero1Galletas.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        textoRepostero1Galletas.setFocusable(false);

        textoRepostero1Galletas.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                textoRepostero1GalletasActionPerformed(evt);

            }

        });

        backgroung.add(textoRepostero1Galletas, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 150, 200, -1));

        textoRepostero2Galletas.setEditable(false);

        textoRepostero2Galletas.setBackground(new java.awt.Color(238, 212, 130));

        textoRepostero2Galletas.setForeground(new java.awt.Color(0, 0, 0));

        textoRepostero2Galletas.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        textoRepostero2Galletas.setFocusable(false);

        textoRepostero2Galletas.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                textoRepostero2GalletasActionPerformed(evt);

            }

        });

        backgroung.add(textoRepostero2Galletas, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 220, 200, -1));

        textoRepostero3Galletas.setEditable(false);

        textoRepostero3Galletas.setBackground(new java.awt.Color(238, 212, 130));

        textoRepostero3Galletas.setForeground(new java.awt.Color(0, 0, 0));

        textoRepostero3Galletas.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        textoRepostero3Galletas.setFocusable(false);

        textoRepostero3Galletas.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                textoRepostero3GalletasActionPerformed(evt);

            }

        });

        backgroung.add(textoRepostero3Galletas, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 290, 200, -1));

        textoRepostero4Galletas.setEditable(false);

        textoRepostero4Galletas.setBackground(new java.awt.Color(238, 212, 130));

        textoRepostero4Galletas.setForeground(new java.awt.Color(0, 0, 0));

        textoRepostero4Galletas.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        textoRepostero4Galletas.setFocusable(false);

        textoRepostero4Galletas.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                textoRepostero4GalletasActionPerformed(evt);

            }

        });

        backgroung.add(textoRepostero4Galletas, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 350, 200, -1));

        textoRepostero5Galletas.setEditable(false);

        textoRepostero5Galletas.setBackground(new java.awt.Color(238, 212, 130));

        textoRepostero5Galletas.setForeground(new java.awt.Color(0, 0, 0));

        textoRepostero5Galletas.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        textoRepostero5Galletas.setFocusable(false);

        textoRepostero5Galletas.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                textoRepostero5GalletasActionPerformed(evt);

            }

        });

        backgroung.add(textoRepostero5Galletas, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 410, 200, -1));

        textoRepostero1GalletasDesperdiciadas.setEditable(false);

        textoRepostero1GalletasDesperdiciadas.setBackground(new java.awt.Color(238, 212, 130));

        textoRepostero1GalletasDesperdiciadas.setForeground(new java.awt.Color(255, 0, 0));

        textoRepostero1GalletasDesperdiciadas.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        textoRepostero1GalletasDesperdiciadas.setFocusable(false);

        textoRepostero1GalletasDesperdiciadas.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                textoRepostero1GalletasDesperdiciadasActionPerformed(evt);

            }

        });

        backgroung.add(textoRepostero1GalletasDesperdiciadas, new org.netbeans.lib.awtextra.AbsoluteConstraints(670, 150, 200, -1));

        textoRepostero2GalletasDesperdiciadas.setEditable(false);

        textoRepostero2GalletasDesperdiciadas.setBackground(new java.awt.Color(238, 212, 130));

        textoRepostero2GalletasDesperdiciadas.setForeground(new java.awt.Color(255, 0, 0));

        textoRepostero2GalletasDesperdiciadas.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        textoRepostero2GalletasDesperdiciadas.setFocusable(false);

        textoRepostero2GalletasDesperdiciadas.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                textoRepostero2GalletasDesperdiciadasActionPerformed(evt);

            }

        });

        backgroung.add(textoRepostero2GalletasDesperdiciadas, new org.netbeans.lib.awtextra.AbsoluteConstraints(670, 220, 200, -1));

        textoRepostero3GalletasDesperdiciadas.setEditable(false);

        textoRepostero3GalletasDesperdiciadas.setBackground(new java.awt.Color(238, 212, 130));

        textoRepostero3GalletasDesperdiciadas.setForeground(new java.awt.Color(255, 0, 0));

        textoRepostero3GalletasDesperdiciadas.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        textoRepostero3GalletasDesperdiciadas.setFocusable(false);

        textoRepostero3GalletasDesperdiciadas.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                textoRepostero3GalletasDesperdiciadasActionPerformed(evt);

            }

        });

        backgroung.add(textoRepostero3GalletasDesperdiciadas, new org.netbeans.lib.awtextra.AbsoluteConstraints(670, 290, 200, -1));

        textoRepostero4GalletasDesperdiciadas.setEditable(false);

        textoRepostero4GalletasDesperdiciadas.setBackground(new java.awt.Color(238, 212, 130));

        textoRepostero4GalletasDesperdiciadas.setForeground(new java.awt.Color(255, 0, 0));

        textoRepostero4GalletasDesperdiciadas.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        textoRepostero4GalletasDesperdiciadas.setFocusable(false);

        textoRepostero4GalletasDesperdiciadas.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                textoRepostero4GalletasDesperdiciadasActionPerformed(evt);

            }

        });

        backgroung.add(textoRepostero4GalletasDesperdiciadas, new org.netbeans.lib.awtextra.AbsoluteConstraints(670, 350, 200, -1));

        textoRepostero5GalletasDesperdiciadas.setEditable(false);

        textoRepostero5GalletasDesperdiciadas.setBackground(new java.awt.Color(238, 212, 130));

        textoRepostero5GalletasDesperdiciadas.setForeground(new java.awt.Color(255, 0, 0));

        textoRepostero5GalletasDesperdiciadas.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        textoRepostero5GalletasDesperdiciadas.setFocusable(false);

        textoRepostero5GalletasDesperdiciadas.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                textoRepostero5GalletasDesperdiciadasActionPerformed(evt);

            }

        });

        backgroung.add(textoRepostero5GalletasDesperdiciadas, new org.netbeans.lib.awtextra.AbsoluteConstraints(670, 410, 200, -1));

        textoGalletasDesperdiciadas.setFont(new java.awt.Font("Arial Black", 0, 18)); *// NOI18N*

        textoGalletasDesperdiciadas.setForeground(new java.awt.Color(0, 0, 0));

        textoGalletasDesperdiciadas.setText("Galletas Desperdiciadas");

        backgroung.add(textoGalletasDesperdiciadas, new org.netbeans.lib.awtextra.AbsoluteConstraints(670, 90, 250, 40));

        textoGalletasProducidas.setFont(new java.awt.Font("Arial Black", 0, 18)); *// NOI18N*

        textoGalletasProducidas.setForeground(new java.awt.Color(0, 0, 0));

        textoGalletasProducidas.setText("Galletas Producidas");

        backgroung.add(textoGalletasProducidas, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 90, 220, 40));

        separadorColumnaReposteros.setOrientation(javax.swing.SwingConstants.VERTICAL);

        backgroung.add(separadorColumnaReposteros, new org.netbeans.lib.awtextra.AbsoluteConstraints(600, 150, 10, 290));

        textoGalletasHorneadas.setFont(new java.awt.Font("Arial Black", 0, 18)); *// NOI18N*

        textoGalletasHorneadas.setForeground(new java.awt.Color(0, 0, 0));

        textoGalletasHorneadas.setText("Galletas Horneadas");

        backgroung.add(textoGalletasHorneadas, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 490, 250, 40));

        textoHorneando.setFont(new java.awt.Font("Arial Black", 0, 18)); *// NOI18N*

        textoHorneando.setForeground(new java.awt.Color(0, 0, 0));

        textoHorneando.setText("Horneando");

        backgroung.add(textoHorneando, new org.netbeans.lib.awtextra.AbsoluteConstraints(670, 490, 140, 40));

        textoHorno1GalletasHorneadas.setEditable(false);

        textoHorno1GalletasHorneadas.setBackground(new java.awt.Color(238, 212, 130));

        textoHorno1GalletasHorneadas.setForeground(new java.awt.Color(0, 0, 0));

        textoHorno1GalletasHorneadas.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        textoHorno1GalletasHorneadas.setFocusable(false);

        textoHorno1GalletasHorneadas.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                textoHorno1GalletasHorneadasActionPerformed(evt);

            }

        });

        backgroung.add(textoHorno1GalletasHorneadas, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 550, 200, -1));

        textoHorno2GalletasHorneadas.setEditable(false);

        textoHorno2GalletasHorneadas.setBackground(new java.awt.Color(238, 212, 130));

        textoHorno2GalletasHorneadas.setForeground(new java.awt.Color(0, 0, 0));

        textoHorno2GalletasHorneadas.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        textoHorno2GalletasHorneadas.setFocusable(false);

        textoHorno2GalletasHorneadas.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                textoHorno2GalletasHorneadasActionPerformed(evt);

            }

        });

        backgroung.add(textoHorno2GalletasHorneadas, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 600, 200, -1));

        textoHorno3GalletasHorneadas.setEditable(false);

        textoHorno3GalletasHorneadas.setBackground(new java.awt.Color(238, 212, 130));

        textoHorno3GalletasHorneadas.setForeground(new java.awt.Color(0, 0, 0));

        textoHorno3GalletasHorneadas.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        textoHorno3GalletasHorneadas.setFocusable(false);

        textoHorno3GalletasHorneadas.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                textoHorno3GalletasHorneadasActionPerformed(evt);

            }

        });

        backgroung.add(textoHorno3GalletasHorneadas, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 650, 200, -1));

        separadorColumnaHornos.setOrientation(javax.swing.SwingConstants.VERTICAL);

        backgroung.add(separadorColumnaHornos, new org.netbeans.lib.awtextra.AbsoluteConstraints(600, 550, 10, 130));

        panelControlHorneado1.setEditable(false);

        panelControlHorneado1.setBackground(new java.awt.Color(255, 255, 255));

        panelControlHorneado1.setForeground(new java.awt.Color(0, 0, 0));

        panelControlHorneado1.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        panelControlHorneado1.setFocusable(false);

        panelControlHorneado1.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                panelControlHorneado1ActionPerformed(evt);

            }

        });

        backgroung.add(panelControlHorneado1, new org.netbeans.lib.awtextra.AbsoluteConstraints(670, 550, 200, -1));

        panelControlHorneado2.setEditable(false);

        panelControlHorneado2.setBackground(new java.awt.Color(255, 255, 255));

        panelControlHorneado2.setForeground(new java.awt.Color(0, 0, 0));

        panelControlHorneado2.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        panelControlHorneado2.setFocusable(false);

        panelControlHorneado2.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                panelControlHorneado2ActionPerformed(evt);

            }

        });

        backgroung.add(panelControlHorneado2, new org.netbeans.lib.awtextra.AbsoluteConstraints(670, 600, 200, -1));

        panelControlHorneado3.setEditable(false);

        panelControlHorneado3.setBackground(new java.awt.Color(255, 255, 255));

        panelControlHorneado3.setForeground(new java.awt.Color(0, 0, 0));

        panelControlHorneado3.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        panelControlHorneado3.setFocusable(false);

        panelControlHorneado3.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                panelControlHorneado3ActionPerformed(evt);

            }

        });

        backgroung.add(panelControlHorneado3, new org.netbeans.lib.awtextra.AbsoluteConstraints(670, 650, 200, -1));

        botonPararRepostero1.setBackground(new java.awt.Color(255, 0, 0));

        botonPararRepostero1.setFont(new java.awt.Font("Arial Black", 0, 12)); *// NOI18N*

        botonPararRepostero1.setForeground(new java.awt.Color(0, 0, 0));

        botonPararRepostero1.setText("PARAR");

        botonPararRepostero1.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

        botonPararRepostero1.setFocusPainted(false);

        botonPararRepostero1.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                botonPararRepostero1ActionPerformed(evt);

            }

        });

        backgroung.add(botonPararRepostero1, new org.netbeans.lib.awtextra.AbsoluteConstraints(960, 150, 130, 20));

        botonPararRepostero2.setBackground(new java.awt.Color(255, 0, 0));

        botonPararRepostero2.setFont(new java.awt.Font("Arial Black", 0, 12)); *// NOI18N*

        botonPararRepostero2.setForeground(new java.awt.Color(0, 0, 0));

        botonPararRepostero2.setText("PARAR");

        botonPararRepostero2.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

        botonPararRepostero2.setFocusPainted(false);

        botonPararRepostero2.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                botonPararRepostero2ActionPerformed(evt);

            }

        });

        backgroung.add(botonPararRepostero2, new org.netbeans.lib.awtextra.AbsoluteConstraints(960, 220, 130, 20));

        botonPararRepostero3.setBackground(new java.awt.Color(255, 0, 0));

        botonPararRepostero3.setFont(new java.awt.Font("Arial Black", 0, 12)); *// NOI18N*

        botonPararRepostero3.setForeground(new java.awt.Color(0, 0, 0));

        botonPararRepostero3.setText("PARAR");

        botonPararRepostero3.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

        botonPararRepostero3.setFocusPainted(false);

        botonPararRepostero3.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                botonPararRepostero3ActionPerformed(evt);

            }

        });

        backgroung.add(botonPararRepostero3, new org.netbeans.lib.awtextra.AbsoluteConstraints(960, 290, 130, 20));

        botonPararRepostero4.setBackground(new java.awt.Color(255, 0, 0));

        botonPararRepostero4.setFont(new java.awt.Font("Arial Black", 0, 12)); *// NOI18N*

        botonPararRepostero4.setForeground(new java.awt.Color(0, 0, 0));

        botonPararRepostero4.setText("PARAR");

        botonPararRepostero4.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

        botonPararRepostero4.setFocusPainted(false);

        botonPararRepostero4.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                botonPararRepostero4ActionPerformed(evt);

            }

        });

        backgroung.add(botonPararRepostero4, new org.netbeans.lib.awtextra.AbsoluteConstraints(960, 350, 130, 20));

        botonPararRepostero5.setBackground(new java.awt.Color(255, 0, 0));

        botonPararRepostero5.setFont(new java.awt.Font("Arial Black", 0, 12)); *// NOI18N*

        botonPararRepostero5.setForeground(new java.awt.Color(0, 0, 0));

        botonPararRepostero5.setText("PARAR");

        botonPararRepostero5.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

        botonPararRepostero5.setFocusPainted(false);

        botonPararRepostero5.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                botonPararRepostero5ActionPerformed(evt);

            }

        });

        backgroung.add(botonPararRepostero5, new org.netbeans.lib.awtextra.AbsoluteConstraints(960, 410, 130, 20));

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

        getContentPane().setLayout(layout);

        layout.setHorizontalGroup(

            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(backgroung, javax.swing.GroupLayout.PREFERRED\_SIZE, 1387, Short.MAX\_VALUE)

        );

        layout.setVerticalGroup(

            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(backgroung, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

        );

        pack();

    }*// </editor-fold>//GEN-END:initComponents*

    private void headerMousePressed(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_headerMousePressed*

        xMouse = evt.getX();

        yMouse = evt.getY();

    }*//GEN-LAST:event\_headerMousePressed*

    private void headerMouseDragged(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_headerMouseDragged*

        int x = evt.getXOnScreen();

        int y = evt.getYOnScreen();

        this.setLocation(x - xMouse, y - yMouse);

    }*//GEN-LAST:event\_headerMouseDragged*

    private void labelCerrarMouseClicked(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_labelCerrarMouseClicked*

        System.exit(0);

    }*//GEN-LAST:event\_labelCerrarMouseClicked*

    private void labelCerrarMouseEntered(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_labelCerrarMouseEntered*

        labelCerrar.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/IconoCerrar2.png")));

    }*//GEN-LAST:event\_labelCerrarMouseEntered*

    private void labelCerrarMouseExited(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_labelCerrarMouseExited*

        labelCerrar.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/IconoCerrar.png")));

    }*//GEN-LAST:event\_labelCerrarMouseExited*

    private void textoRepostero1GalletasActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_textoRepostero1GalletasActionPerformed*

*// TODO add your handling code here:*

    }*//GEN-LAST:event\_textoRepostero1GalletasActionPerformed*

    private void textoRepostero2GalletasActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_textoRepostero2GalletasActionPerformed*

*// TODO add your handling code here:*

    }*//GEN-LAST:event\_textoRepostero2GalletasActionPerformed*

    private void textoRepostero3GalletasActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_textoRepostero3GalletasActionPerformed*

*// TODO add your handling code here:*

    }*//GEN-LAST:event\_textoRepostero3GalletasActionPerformed*

    private void textoRepostero4GalletasActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_textoRepostero4GalletasActionPerformed*

*// TODO add your handling code here:*

    }*//GEN-LAST:event\_textoRepostero4GalletasActionPerformed*

    private void textoRepostero5GalletasActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_textoRepostero5GalletasActionPerformed*

*// TODO add your handling code here:*

    }*//GEN-LAST:event\_textoRepostero5GalletasActionPerformed*

    private void textoRepostero1GalletasDesperdiciadasActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_textoRepostero1GalletasDesperdiciadasActionPerformed*

*// TODO add your handling code here:*

    }*//GEN-LAST:event\_textoRepostero1GalletasDesperdiciadasActionPerformed*

    private void textoRepostero2GalletasDesperdiciadasActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_textoRepostero2GalletasDesperdiciadasActionPerformed*

*// TODO add your handling code here:*

    }*//GEN-LAST:event\_textoRepostero2GalletasDesperdiciadasActionPerformed*

    private void textoRepostero3GalletasDesperdiciadasActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_textoRepostero3GalletasDesperdiciadasActionPerformed*

*// TODO add your handling code here:*

    }*//GEN-LAST:event\_textoRepostero3GalletasDesperdiciadasActionPerformed*

    private void textoRepostero4GalletasDesperdiciadasActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_textoRepostero4GalletasDesperdiciadasActionPerformed*

*// TODO add your handling code here:*

    }*//GEN-LAST:event\_textoRepostero4GalletasDesperdiciadasActionPerformed*

    private void textoRepostero5GalletasDesperdiciadasActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_textoRepostero5GalletasDesperdiciadasActionPerformed*

*// TODO add your handling code here:*

    }*//GEN-LAST:event\_textoRepostero5GalletasDesperdiciadasActionPerformed*

    private void textoAlmacenGalletasAlmacenadasActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_textoAlmacenGalletasAlmacenadasActionPerformed*

*// TODO add your handling code here:*

    }*//GEN-LAST:event\_textoAlmacenGalletasAlmacenadasActionPerformed*

    private void textoHorno1GalletasHorneadasActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_textoHorno1GalletasHorneadasActionPerformed*

*// TODO add your handling code here:*

    }*//GEN-LAST:event\_textoHorno1GalletasHorneadasActionPerformed*

    private void textoHorno2GalletasHorneadasActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_textoHorno2GalletasHorneadasActionPerformed*

*// TODO add your handling code here:*

    }*//GEN-LAST:event\_textoHorno2GalletasHorneadasActionPerformed*

    private void textoAlmacenGalletasConsumidasActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_textoAlmacenGalletasConsumidasActionPerformed*

*// TODO add your handling code here:*

    }*//GEN-LAST:event\_textoAlmacenGalletasConsumidasActionPerformed*

    private void textoHorno3GalletasHorneadasActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_textoHorno3GalletasHorneadasActionPerformed*

*// TODO add your handling code here:*

    }*//GEN-LAST:event\_textoHorno3GalletasHorneadasActionPerformed*

    private void iconoMinimizadoMouseClicked(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_iconoMinimizadoMouseClicked*

        Point startLocation = this.getLocation();

        Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

        int taskbarHeight = Toolkit.getDefaultToolkit().getScreenInsets(this.getGraphicsConfiguration()).bottom;

        Point endLocation = new Point(

            (int) screenSize.getWidth() / 2,

            (int) screenSize.getHeight() - taskbarHeight

        );

        Timer timer = new Timer(10, null);

        timer.addActionListener(new ActionListener() {

            double progress = 0;

            final double step = 0.05;

            @Override

            public void actionPerformed(ActionEvent evt) {

                progress += step;

                if (progress >= 1.0) {

                    timer.stop();

                    setExtendedState(JFrame.ICONIFIED);

                } else {

                    int x = (int) (startLocation.x + progress \* (endLocation.x - startLocation.x));

                    int y = (int) (startLocation.y + progress \* (endLocation.y - startLocation.y));

                    setLocation(x, y);

                }

            }

        });

        timer.start();

    }*//GEN-LAST:event\_iconoMinimizadoMouseClicked*

    private void iconoMinimizadoMouseEntered(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_iconoMinimizadoMouseEntered*

        iconoMinimizado.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/iconoMinimizado2.png")));

    }*//GEN-LAST:event\_iconoMinimizadoMouseEntered*

    private void iconoMinimizadoMouseExited(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_iconoMinimizadoMouseExited*

        iconoMinimizado.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fototeca/iconoMinimizado.png")));

    }*//GEN-LAST:event\_iconoMinimizadoMouseExited*

    private void panelControlHorneado1ActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_panelControlHorneado1ActionPerformed*

*// TODO add your handling code here:*

    }*//GEN-LAST:event\_panelControlHorneado1ActionPerformed*

    private void panelControlHorneado2ActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_panelControlHorneado2ActionPerformed*

*// TODO add your handling code here:*

    }*//GEN-LAST:event\_panelControlHorneado2ActionPerformed*

    private void panelControlHorneado3ActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_panelControlHorneado3ActionPerformed*

*// TODO add your handling code here:*

    }*//GEN-LAST:event\_panelControlHorneado3ActionPerformed*

    private void botonPararRepostero2ActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_botonPararRepostero2ActionPerformed*

*//Hay que generar las dos casuisticas en funcion del texto que haya en el boton*

        if (botonPararRepostero2.getText().equals("PARAR"))

        {

*//Debemos de para la ejecucion del repostero*

            try

            {

                metodosServer.setParadaManualDesdeCliente(true, 1);

            }

            catch(RemoteException error)

            {

                System.out.println("Se ha producido el siguiente error mientras se intentaba parar el repostero 2");

            }

*//Una vez cambiado el estado del repostero actualizamos la apariencia del boton*

            botonPararRepostero2.setText("REANUDAR");

            botonPararRepostero2.setBackground(Color.green);

        }

        else

        {

*//Debemos de reanudar la ejecucion del repostero*

            try

            {

                metodosServer.setParadaManualDesdeCliente(false, 1);

            }

            catch(RemoteException error)

            {

                System.out.println("Se ha producido el siguiente error mientras se intentaba parar el repostero 2");

            }

*//Una vez cambiado el estado del repostero actualizamos la apariencia del boton*

            botonPararRepostero2.setText("PARAR");

            botonPararRepostero2.setBackground(Color.red);

        }

    }*//GEN-LAST:event\_botonPararRepostero2ActionPerformed*

    private void botonPararRepostero3ActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_botonPararRepostero3ActionPerformed*

*//Hay que generar las dos casuisticas en funcion del texto que haya en el boton*

        if (botonPararRepostero3.getText().equals("PARAR"))

        {

*//Debemos de para la ejecucion del repostero*

            try

            {

                metodosServer.setParadaManualDesdeCliente(true, 2);

            }

            catch(RemoteException error)

            {

                System.out.println("Se ha producido el siguiente error mientras se intentaba parar el repostero 3");

            }

*//Una vez cambiado el estado del repostero actualizamos la apariencia del boton*

            botonPararRepostero3.setText("REANUDAR");

            botonPararRepostero3.setBackground(Color.green);

        }

        else

        {

*//Debemos de reanudar la ejecucion del repostero*

            try

            {

                metodosServer.setParadaManualDesdeCliente(false, 2);

            }

            catch(RemoteException error)

            {

                System.out.println("Se ha producido el siguiente error mientras se intentaba parar el repostero 3");

            }

*//Una vez cambiado el estado del repostero actualizamos la apariencia del boton*

            botonPararRepostero3.setText("PARAR");

            botonPararRepostero3.setBackground(Color.red);

        }

    }*//GEN-LAST:event\_botonPararRepostero3ActionPerformed*

    private void botonPararRepostero4ActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_botonPararRepostero4ActionPerformed*

*//Hay que generar las dos casuisticas en funcion del texto que haya en el boton*

        if (botonPararRepostero4.getText().equals("PARAR"))

        {

*//Debemos de para la ejecucion del repostero*

            try

            {

                metodosServer.setParadaManualDesdeCliente(true, 3);

            }

            catch(RemoteException error)

            {

                System.out.println("Se ha producido el siguiente error mientras se intentaba parar el repostero 4");

            }

*//Una vez cambiado el estado del repostero actualizamos la apariencia del boton*

            botonPararRepostero4.setText("REANUDAR");

            botonPararRepostero4.setBackground(Color.green);

        }

        else

        {

*//Debemos de reanudar la ejecucion del repostero*

            try

            {

                metodosServer.setParadaManualDesdeCliente(false, 3);

            }

            catch(RemoteException error)

            {

                System.out.println("Se ha producido el siguiente error mientras se intentaba parar el repostero 4");

            }

*//Una vez cambiado el estado del repostero actualizamos la apariencia del boton*

            botonPararRepostero4.setText("PARAR");

            botonPararRepostero4.setBackground(Color.red);

        }

    }*//GEN-LAST:event\_botonPararRepostero4ActionPerformed*

    private void botonPararRepostero5ActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_botonPararRepostero5ActionPerformed*

*//Hay que generar las dos casuisticas en funcion del texto que haya en el boton*

        if (botonPararRepostero5.getText().equals("PARAR"))

        {

*//Debemos de para la ejecucion del repostero*

            try

            {

                metodosServer.setParadaManualDesdeCliente(true, 4);

            }

            catch(RemoteException error)

            {

                System.out.println("Se ha producido el siguiente error mientras se intentaba parar el repostero 5");

            }

*//Una vez cambiado el estado del repostero actualizamos la apariencia del boton*

            botonPararRepostero5.setText("REANUDAR");

            botonPararRepostero5.setBackground(Color.green);

        }

        else

        {

*//Debemos de reanudar la ejecucion del repostero*

            try

            {

                metodosServer.setParadaManualDesdeCliente(false, 4);

            }

            catch(RemoteException error)

            {

                System.out.println("Se ha producido el siguiente error mientras se intentaba parar el repostero 5");

            }

*//Una vez cambiado el estado del repostero actualizamos la apariencia del boton*

            botonPararRepostero5.setText("PARAR");

            botonPararRepostero5.setBackground(Color.red);

        }

    }*//GEN-LAST:event\_botonPararRepostero5ActionPerformed*

    private void botonPararRepostero1ActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_botonPararRepostero1ActionPerformed*

*//Hay que generar las dos casuisticas en funcion del texto que haya en el boton*

        if (botonPararRepostero1.getText().equals("PARAR"))

        {

*//Debemos de para la ejecucion del repostero*

            try

            {

                metodosServer.setParadaManualDesdeCliente(true, 0);

            }

            catch(RemoteException error)

            {

                System.out.println("Se ha producido el siguiente error mientras se intentaba parar el repostero 1");

            }

*//Una vez cambiado el estado del repostero actualizamos la apariencia del boton*

            botonPararRepostero1.setText("REANUDAR");

            botonPararRepostero1.setBackground(Color.green);

        }

        else

        {

*//Debemos de reanudar la ejecucion del repostero*

            try

            {

                metodosServer.setParadaManualDesdeCliente(false, 0);

            }

            catch(RemoteException error)

            {

                System.out.println("Se ha producido el siguiente error mientras se intentaba parar el repostero 1");

            }

*//Una vez cambiado el estado del repostero actualizamos la apariencia del boton*

            botonPararRepostero1.setText("PARAR");

            botonPararRepostero1.setBackground(Color.red);

        }

    }*//GEN-LAST:event\_botonPararRepostero1ActionPerformed*

*//METODOS CREADOS A MANO*

    public JTextField getTextosReposterosGalletas(int identificadorTexto)

    {

            switch (identificadorTexto)

            {

                case 0:

                    return textoRepostero1Galletas;

                case 1:

                    return textoRepostero2Galletas;

                case 2:

                    return textoRepostero3Galletas;

                case 3:

                    return textoRepostero4Galletas;

                case 4:

                    return textoRepostero5Galletas;

            }

*//Si no entra en ningun case, no devolvemos anda porque no han seleccionado el indice bien*

            return null;

    }

    public JTextField getTextosReposterosGalletasDesperdiciadas(int identificadorTexto)

    {

            switch (identificadorTexto)

            {

                case 0:

                    return textoRepostero1GalletasDesperdiciadas;

                case 1:

                    return textoRepostero2GalletasDesperdiciadas;

                case 2:

                    return textoRepostero3GalletasDesperdiciadas;

                case 3:

                    return textoRepostero4GalletasDesperdiciadas;

                case 4:

                    return textoRepostero5GalletasDesperdiciadas;

            }

*//Si no entra en ningun case, no devolvemos anda porque no han seleccionado el indice bien*

            return null;

    }

    public JTextField getTextosTotalGalletasHorneadas(int identificadorTexto)

    {

            switch (identificadorTexto)

            {

                case 0:

                    return textoHorno1GalletasHorneadas;

                case 1:

                    return textoHorno2GalletasHorneadas;

                case 2:

                    return textoHorno3GalletasHorneadas;

            }

*//Si no entra en ningun case, no devolvemos anda porque no han seleccionado el indice bien*

            return null;

    }

    public JTextField getPanelControlHorneado(int identificadorTexto)

    {

            switch (identificadorTexto)

            {

                case 0:

                    return panelControlHorneado1;

                case 1:

                    return panelControlHorneado2;

                case 2:

                    return panelControlHorneado3;

            }

*//Si no entra en ningun case, no devolvemos anda porque no han seleccionado el indice bien*

            return null;

    }

*/\*\**

*\* @param args the command line arguments*

*\*/*

    public static void main(String args[]) {

*/\* Set the Nimbus look and feel \*/*

*//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">*

*/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.*

*\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html*

*\*/*

        try {

            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

                if ("Nimbus".equals(info.getName())) {

                    javax.swing.UIManager.setLookAndFeel(info.getClassName());

                    break;

                }

            }

        } catch (ClassNotFoundException ex) {

            java.util.logging.Logger.getLogger(PanelCliente.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (InstantiationException ex) {

            java.util.logging.Logger.getLogger(PanelCliente.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {

            java.util.logging.Logger.getLogger(PanelCliente.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

            java.util.logging.Logger.getLogger(PanelCliente.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        }

*//</editor-fold>*

*//</editor-fold>*

*//</editor-fold>*

*//</editor-fold>*

*//</editor-fold>*

*//</editor-fold>*

*//</editor-fold>*

*//</editor-fold>*

*/\* Create and display the form \*/*

        java.awt.EventQueue.invokeLater(new Runnable() {

            public void run() {

            }

        });

    }

*// Variables declaration - do not modify//GEN-BEGIN:variables*

    private javax.swing.JPanel backgroung;

    private javax.swing.JButton botonPararRepostero1;

    private javax.swing.JButton botonPararRepostero2;

    private javax.swing.JButton botonPararRepostero3;

    private javax.swing.JButton botonPararRepostero4;

    private javax.swing.JButton botonPararRepostero5;

    private javax.swing.JPanel header;

    private javax.swing.JLabel horno1;

    private javax.swing.JLabel horno2;

    private javax.swing.JLabel horno3;

    private javax.swing.JLabel iconoCookienFactory;

    private javax.swing.JLabel iconoMinimizado;

    private javax.swing.JLabel labelCerrar;

    private javax.swing.JLabel logoMedioGrande;

    private javax.swing.JPanel menuDerecha;

    private javax.swing.JLabel panelCliente;

    private javax.swing.JTextField panelControlHorneado1;

    private javax.swing.JTextField panelControlHorneado2;

    private javax.swing.JTextField panelControlHorneado3;

    private javax.swing.JLabel respostero1;

    private javax.swing.JLabel respostero2;

    private javax.swing.JLabel respostero3;

    private javax.swing.JLabel respostero4;

    private javax.swing.JLabel respostero5;

    private javax.swing.JSeparator separadorColumnaHornos;

    private javax.swing.JSeparator separadorColumnaReposteros;

    private javax.swing.JSeparator separadorHeader;

    private javax.swing.JSeparator separadorMenu;

    private javax.swing.JLabel textoAlmacen;

    private javax.swing.JTextField textoAlmacenGalletasAlmacenadas;

    private javax.swing.JTextField textoAlmacenGalletasConsumidas;

    private javax.swing.JLabel textoGalletasAlmacenadas;

    private javax.swing.JLabel textoGalletasDesperdiciadas;

    private javax.swing.JLabel textoGalletasDesperdiciadas1;

    private javax.swing.JLabel textoGalletasHorneadas;

    private javax.swing.JLabel textoGalletasProducidas;

    private javax.swing.JLabel textoHorneando;

    private javax.swing.JTextField textoHorno1GalletasHorneadas;

    private javax.swing.JTextField textoHorno2GalletasHorneadas;

    private javax.swing.JTextField textoHorno3GalletasHorneadas;

    private javax.swing.JTextField textoRepostero1Galletas;

    private javax.swing.JTextField textoRepostero1GalletasDesperdiciadas;

    private javax.swing.JTextField textoRepostero2Galletas;

    private javax.swing.JTextField textoRepostero2GalletasDesperdiciadas;

    private javax.swing.JTextField textoRepostero3Galletas;

    private javax.swing.JTextField textoRepostero3GalletasDesperdiciadas;

    private javax.swing.JTextField textoRepostero4Galletas;

    private javax.swing.JTextField textoRepostero4GalletasDesperdiciadas;

    private javax.swing.JTextField textoRepostero5Galletas;

    private javax.swing.JTextField textoRepostero5GalletasDesperdiciadas;

*// End of variables declaration//GEN-END:variables*

    @Override

    public void run() {

*//la interfaz del cliente estara constantemente refrescando la informacion*

                while(true)

                {

                    try

                    {

*//REPOSTEROS*

*//Checkeamos las galletas totales producidas por cada repostero y lo imprimimos en el JTextField*

*//Checkeamos las galletas totales desperdiciadas por cada repostero y lo imprimimos en el JTextField*

                        List<Repostero> listaReposteros = metodosServer.getListaReposteros();

                        for (int indice = 0; indice < listaReposteros.size(); indice++ )

                        {

                            int galletasTotales = listaReposteros.get(indice).getTotalGalletas();

                            int galletasTotalesDesperdiciadas = listaReposteros.get(indice).getTotalGalletasDesperdiciadas();

                            getTextosReposterosGalletas(indice).setText(String.valueOf(galletasTotales));

                            getTextosReposterosGalletasDesperdiciadas(indice).setText(String.valueOf(galletasTotalesDesperdiciadas));

                        }

*//HORNOS*

                        List<Horno> listaHornos = metodosServer.getListaHornos();

                        for (int indice = 0; indice < listaHornos.size(); indice++)

                        {

*//Checkeamos el total galletas horneadas*

                            int totalGalletasHorneadas = listaHornos.get(indice).getTotalGalletasHorneadas();

                            getTextosTotalGalletasHorneadas(indice).setText(String.valueOf(totalGalletasHorneadas));

*//Check si esta horneando (SI o NO)*

                            if(listaHornos.get(indice).getEstaHorneando())

                            {

                                getPanelControlHorneado(indice).setBackground(Color.red);

                            }

                            else

                            {

                                getPanelControlHorneado(indice).setBackground(Color.white);

                            }

                        }

*//ALMACEN*

                        textoAlmacenGalletasAlmacenadas.setText(String.valueOf(metodosServer.getAlmacen().getGalletasTotalAlmacenadas()));

                        textoAlmacenGalletasConsumidas.setText(String.valueOf(metodosServer.getAlmacen().getGalletasTotalConsumidas()));

*//Finalizamos el checkeo para el cliente y esperamos un segundo hasta el siguiente checkeo*

                        Thread.sleep(1000);

                    }

                    catch (InterruptedException error)

                    {

                        System.out.println("Se ha producido un error mientras la interfaz del cliente estaba actualizando los datos de la fabrica --> " + error);

                    }

                    catch (RemoteException error)

                    {

                        System.out.println("Se ha producido un error mientras se usaba algun hilo (Reposteros,Hornos,Empaquetadores) del servidor --> " + error);

                    }

                }

    }

}

## 5.6 Logger

### 5.6.1 Logger.java

*/\*\**

*\* version 1.0*

*\* contacto --> https://www.linkedin.com/in/david-sanchez-1366b1253/*

*\* @author David Sánchez*

*\*/*

*//Paquete al que pertenece la clase*

package Logger;

*//Importes*

import java.io.BufferedWriter;

import java.io.FileWriter;

import java.io.IOException;

import java.io.Serializable;

import java.time.LocalDateTime;

import java.time.format.DateTimeFormatter;

public class Logger implements Serializable

{

    private static final String NOMBRE\_ARCHIVO = "evolucionGalletas.txt";

    private static Logger instance;

    private Logger()

    {

*//Crear el archivo si no existe (opcional, para asegurar su existencia al inicio)*

        try (BufferedWriter salida = new BufferedWriter(new FileWriter(NOMBRE\_ARCHIVO, true)))

        {

            salida.write("=== Inicio de Logs: " + getCurrentTimestamp() + " ===\n");

        }

        catch (IOException error)

        {

            System.out.println("Error al inicializar el log --> " + error);

        }

    }

*//Método para obtener la instancia única*

    public static synchronized Logger getInstance()

    {

        if (instance == null)

        {

            instance = new Logger();

        }

        return instance;

    }

*//Método para registrar un mensaje en el log*

    public synchronized void log(String mensaje)

    {

        String mensajeLog = "[" + getCurrentTimestamp() + "] " + mensaje;

*//Mostrar el mensaje en consola*

        System.out.println(mensajeLog);

*//Guardar el mensaje en el archivo*

        try (BufferedWriter salida = new BufferedWriter(new FileWriter(NOMBRE\_ARCHIVO, true)))

        {

            salida.write(mensajeLog + "\n");

        }

        catch (IOException error)

        {

            System.out.println("Error al escribir en el log -->" + error);

        }

    }

*//Método para obtener la marca de tiempo actual*

    private String getCurrentTimestamp()

    {

        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");

        return LocalDateTime.now().format(formatter);

    }

}