

# MEMORIA PEAJES

Grado en Ingeniería de Computadores

Curso 2021/2022-Convocatoria Extraordinaria

03221211s – Bachiller Vela, David

## Contenido

1-Análisis de alto nivel.....	3
2-Diseño general del sistema y de las herramientas de sincronización utilizadas .....	4
3) Las clases principales que intervienen con su descripción (atributos y métodos) .....	5
4-DIAGRAMA DE CLASES .....	13
5)FUENTE CÓDIGO ANEXO .....	14
Paquete Clases Aux .....	14
ListasThreads.java .....	14
Paquete Concurrencia .....	19
Cabina.java .....	19
CabinaPeaje.java .....	21
CabinaPeajeManual.java .....	41
CabinaPeajeTarjeta.java .....	51
Paso.java.....	56
Paquete Distribuida.....	58
Gestor.java .....	58
InterfazDistribuida.java .....	63
InterfazPeaje.java.....	91
Paquete Hilos .....	92
Ambulancia.java .....	92
Camion.java .....	95
Coche.java .....	98
CrearCamiones.java .....	101
CrearCoches.java .....	103
Empleado.java .....	105
Paquete Interfaz.....	108
Main.java.....	108
Paquete Log.....	130
FuncionesLog.java .....	130

## 1-Análisis de alto nivel

El proyecto consistirá en la implementación de una cabina de peaje de una carretera, formada por empleados que se crearan como hilos e coches y camiones que a su vez también se crearan como hilos y irán pasando por las distintas cabinas de peaje y serán atendidos por los empleados de métodos concurrentes y sincronizados, los cuales, irán realizando diferentes actividades y por otra parte habrá un nuevo módulo de acceso remoto implementado mediante RMI o Sockets.

El resultado del proyecto dispondrá de la siguiente funcionalidad:

De dos diferentes interfaces gráficas:

En la primera de ellas se mostrara el desarrollo de las actividades del las 10 cabinas de peaje 6 de ellas serán para los coches y de esas seis, tres serán de pago manual y otras tres de pago con tarjeta, en las de pago manual habrá un empleado encargado de atender y cobrar a los coches; A su vez habrá otras 4 cabinas para camiones dos de pago manual y dos de pago con tarjeta, también se implementara un botón para reanudar y parar el programa donde las funciones clave estarán implementadas en la clase Paso.java , también habrá un botón ambulancia lo que generara cada vez que se pulse una ambulancia que se colocara al principio de la cola cuando llegue y accederá una vez que haya una cabina libre por las cabinas de los coches.

En segundo lugar tenemos la primera actividad que es la Soga y en la que se vera la cola de la soga formada por los diferentes niños , el monitor correspondiente a la soga y los dos diferentes equipos que realizaran la actividad.

En la segunda interfaz tenemos un control de la cabina de peaje mediante RMI en el que podemos ir actualizando los jTextField cada segundo y por otra parte podemos abrir y cerrar las cabinas cuatro, cinco y seis de coches, correspondientes a cabinas automáticas de coches y las cabinas cuatro y cinco de camiones, correspondientes a cabinas automáticas, se creara una clase llamada gestor con los métodos para poder llevar acabo esta clase y su correspondiente interface, luego en la clase main que es por asi decirlo la que hace de servidor, se crea el objeto rmi y se establece la conexión con la interfaz distribuida, en la que podremos usar los métodos del gestor.

La parte del servidor cuyo código base será el implementado en la parte 1, correspondiente con las actividades y la primera de las dos interfaces que será necesario para dar soporte al módulo de programación distribuida y habrá una parte cliente que permite hacer las consultas previamente mencionadas

El proyecto no incluye uso de semáforos ni de pool de hilos

## 2-Diseño general del sistema y de las herramientas de sincronización utilizadas

Hemos dividido en trabajo en 6 paquetes:

El primero se llama ClasesAux, en el estarán clases sin mucha relación al tema de la parte de concurrencia o a la parte de distribuida, en el tengo las clase ListaThread, en la cual estarán los diferentes arraylist de coches, camiones, empleados, y el objetivo de esta clase será poder imprimir el contenido de los arraylist en los jTextField de las interfaces

El segundo se llama Concurrencia, este paquete es el esqueleto del proyecto aquí se gestionara las entradas y salidas de los coches , las entradas a las diferentes cabinas el pago de los empleados, las colas de espera, habrá una clase llamada cabina de peaje donde se creen las 10 cabinas y será como la clase principal ya que en el constructor estarán todos los jTextField y todas las cabinas y si quiero acceder a cualquier método de las cabinas o a cualquier jTextField de las interfaces simplemente tendre que hacer un get de esta clase y ya podre acceder a sus respectivos objetos.

El tercero se llama Distribuida, aquí se gestionara todo el tema de RMI, y la creación de los método para actualizar la interfaz cada segundo y para poder cerrar y abrir las cabinas de forma remota, tendrá las clases Gestor.java y InterfazPeaje.interface , donde definire los métodos que usare en la interfaz de la parte de distribuida.

El cuarto se llama Hilos, en este paquete se crearan los coches, camiones y ambulancias, con sus respectivos ids, y todos ellos pertenecientes a la cabnina de peaje y tendrán su método run donde se gestionara la entrada al peaje , posteriormente la entrada a las cabinas, el pago si fuera necesario, y la salida de las cabinas.

El quinto paquete se llama Log, donde implementamos los métodos para crear un fichero.log donde se va a ver todo el Debug del código el cual nos va a ser fundamental para poder ver el desarrollo del programa y poder corregir ciertos errores y poder optimizar mas el programa.

El sexto y se llama Interfaz y es el main de la parte de concurrente, aquí se encuentra la primera de las interfaces, de la cual he hablado y desarrollado con mayor medida en la parte uno de la memoria , y aquí se lanzaran los diferentes hilos , se creara la cabina de peaje a la que le pasare los JtextFields de la interfaz, y también por otra parte se gestionara el botón de detener y reanudar el programa y la creación de ambulancias

### 3) Las clases principales que intervienen con su descripción (atributos y métodos)

#### Paquete Hilos

##### Coche.java:

En esta clase principalmente se creara el hilo con su id y perteneciente a la cabina de peaje y se gestionara el run del hilo tantos las entradas salidas, pagos

Los atributos de esta clase son:

```
private String id;  
private CabinaPeaje cabinapeaje;  
private CabinaPeajeManual cab = null;  
private CabinaPeajeTarjeta cpt = null;
```

Los métodos de esta clase son:

```
El constructor, que tiene como argumentos(int id, CabinaPeaje cb,Paso paso)  
setCpt(CabinaPeajeTarjeta cpt) {, establece la cabina automática  
setCab(CabinaPeajeManual cab) { establece la cabina manual  
Paso getPaso() { devuelve la clase paso  
CabinaPeajeTarjeta getCpt() { obtiene la cabina de peaje automática  
CabinaPeajeManual getCab() { obtiene la cabina de peaje manual  
public String getIdCoche() { obtiene el id del coche
```

##### Camion.java:

En esta clase principalmente se creara el hilo con su id y perteneciente a la cabina de peaje y se gestionara el run del hilo tantos las entradas salidas, pagos

Los atributos de esta clase son:

```
private String id;  
private CabinaPeaje cabinapeaje;  
private CabinaPeajeManual cab = null;  
private CabinaPeajeTarjeta cpt = null;
```

Los métodos de esta clase son:

```
El constructor, que tiene como argumentos(int id, CabinaPeaje cb,Paso paso)  
setCpt(CabinaPeajeTarjeta cpt) {, establece la cabina automática  
setCab(CabinaPeajeManual cab) { establece la cabina manual  
Paso getPaso() { devuelve la clase paso
```

CabinaPeajeTarjeta getCpt() { obtiene la cabina de peaje automática

CabinaPeajeManual getCab() { obtiene la cabina de peaje manual

public String getIdCoche() { obtiene el id del coche

### Ambulancia.java:

En esta clase principalmente se creara el hilo con su id y perteneciente a la cabina de peaje y se gestionara el run del hilo tanto las entradas salidas, pagos

Los atributos de esta clase son:

private String id;

private CabinaPeaje cabinapeaje;

private CabinaPeajeManual cab = null;

private CabinaPeajeTarjeta cpt = null;

Los métodos de esta clase son:

El constructor, que tiene como argumentos(int id, CabinaPeaje cb,Paso paso)

setCpt(CabinaPeajeTarjeta cpt) {, establece la cabina automática

setCab(CabinaPeajeManual cab) { establece la cabina manual

Paso getPaso() { devuelve la clase paso

CabinaPeajeTarjeta getCpt() { obtiene la cabina de peaje automática

CabinaPeajeManual getCab() { obtiene la cabina de peaje manual

public String getIdCoche() { obtiene el id del coche

## Paquete Concurrencia

### Cabina.java:

Los atributos de esta clase son:

private String nombreCabina = "";

private String tipo = "";

private boolean disponible = true;

Los métodos de esta clase son:

El constructor, que tiene como argumentos(String nombre, String tipo)

getNombreCabina () {, obtiene el nombre de la cabina

getNombreCabina (String nombreCabina) { establece el nombre a la cabina

String getTipo () {

setTipo () { obtiene la cabina de peaje automática

boolean isDisponible() { devuelve el valor booleano de disponible

boolean setDisponible (boolean disponible) { establece el valor booleano de disponible

### CabinaPeajeManual.java:

En esta clase se gestionaran las 6 cabinas de peaje manual, habrá métodos para la entrada y salida de los hilos, y también la gestión del pago del empleado, su descanso y su vuelta al trabajo.

Los atributos de esta clase son:

```
private JTextField vehiculo;  
private JTextField empleado;  
private ListaThreadsEmpleado empleados;  
private ListaThreadsCoche coches;  
private ListasThreadsCamion camiones;  
private ListasThreadAmbulancia ambulancia;  
private FuncionesLog fg;  
private int i;  
private boolean accesoCoche = true;  
private boolean accesoCamion = true;  
private boolean cocheEsperando = false;  
private boolean camionEsperando = false;  
private boolean ambulanciaEsperando = false;  
private boolean empleadoEsperando = false;
```

Los métodos de esta clase son:

El constructor, que tiene como argumentos(String nombre, JTextField vehiculo, JTextField empleado, FuncionesLog fg, int i)

entradaEmpleadosCabinas(Empleado e), gestiona la entrada del empleado a la cabina

salidaEmpleadoDescanso(Empleado e), gestiona la salida del empleado una vez que lleve seis vehículos atendidos

vueleveEmpleado(Empleado e), gestiona la vuelta del empleado una vez haya descansado 5 minutos

entradaCabinasCoches(Coche coche), gestiona la entrada de las cabinas de los coches

pagoAmbulancia(Empleado e), gestiona el empleado el pago a las ambulancias, todo con conditions para que el programa no se desincronice

realizarPagoCoches(Empleado e) gestiona el empleado el pago a los coches, todo con conditions para que el programa no se desincronice

realizarPagoCamiones(Empleado e) gestiona el empleado el pago a los camiones, todo con conditions para que el programa no se desincronice salidaCoche(Coche coche)

cocheCabinaManual(Coche coche), gestiona la entrada y salida del coche una vez que se haya hecho el pago

entradaCabinasCamiones(Camion camion), gestiona la entrada a las cabinas de los camiones

salidaCamion(Camion camion), gestiona la salida de los camiones

camionCabinaManual(Camion camion), gestiona la entrada y salida del camion una vez que se haya hecho el pago

entradaCabinasAmbulancia(Ambulancia a)

salidaAmbulancia(Ambulancia am)

ambulanciaCabinaManual(Ambulancia am)

### CabinaPeajeTarjeta.java:

En esta clase se gestionan las 4 cabinas automáticas, y encontramos los métodos de entrada y salida de los hilos los cuales incluyen el pago que durará unos 5 segundos

Los atributos de esta clase son:

```
private FuncionesLog fg;
```

```
private boolean accesoCoche = true;
```

```
private boolean accesoCamion = true;
```

```
private boolean accesoAmbulancia = true;
```

```
private int i;
```

Los métodos de esta clase son:

El constructor, que tiene como argumentos(String nombre, JTextField vehiculo, FuncionesLog fg, int i)

entradaCabinasCamiones(Camion camion), gestiona la entrada a las cabinas de los camiones

pagoCabinasCamiones(Camion camion), gestiona el pago de los camiones

salidaCabinasCamiones(Camion camion), gestiona la salida de los camiones de las cabinas

entradaCabinaCoches(Coche cc), gestiona la entrada, pago y salida de los coches a las cabinas

entradaCabinaAmbulancia(Ambulancia a), gestiona la entrada, pago y salida de las ambulancias



boolean isAbierta(), devuelve si la cabina esta abierto, el cual será básico a la hora de asociar cabinas a los coches

setAbierta(boolean abierta), establece si la cabina esta abierto, el cual será básico a la hora de asociar cabinas a los coches

### CabinaPeaje.java:

Es la clase principal aquí se crean las 10 cabinas , 6 de coches y 4 de camiones , también se asocian a esas cabinas los JTextFields , se crean todos los getters de estos para en la parte de distribuida y poder actualizar los JTextFields, también se crean getters y setters de la clase paso, se gestiona la entrada y salida de los hilos al peaje, no a las cabinas, se usara condition como mecanismo de sincronización.

Los atributos de esta clase son:

```
private JTextField colaEntradaPeaje, EmpleadoCabinaCamiones1M,
EmpleadoCabinaCamiones2M, EmpleadoCabinaCoches1M,
EmpleadoCabinaCoches2M, EmpleadoCabinaCoches3M, camionCabina1M, camionCabina2M,
camionCabina3T1, camionCabina4T, cocheCabina1M1, cocheCabina2M, cocheCabina3M
cocheCabina4T, cocheCabina5T1, cocheCabina6T;

private String pago;

private CabinaPeajeManual cabinaCoches1, cabinaCoches2, cabinaCoches3, cabinaCamiones1,
cabinaCamiones2;

private CabinaPeajeTarjeta cabinaCoches4, cabinaCoches5, cabinaCoches6, cabinaCamiones3,
cabinaCamiones4;

private Paso paso;

private boolean cabinaCocheDisponible = true, cabinaCamionDisponible = true;
```

Los métodos de esta clase son:

El constructor, que tiene como argumentos(JTextField colaEntrada, FuncionesLog fg, JTextField cocheCabina1M, JTextField EmpleadoCabinaCoches1M, JTextField cocheCabina2M, JTextField EmpleadoCabinaCoches2M, JTextField cocheCabina3M, JTextField EmpleadoCabinaCoches3M, JTextField camionCabina1M, JTextField EmpleadoCabinaCamiones1M, JTextField camionCabina2M, JTextField EmpleadoCabinaCamiones2M, JTextField cocheCabina4T, JTextField cocheCabina5T1, JTextField cocheCabina6T, JTextField camionCabina3T1, JTextField camionCabina4T, Paso paso)

entradaColaPeajeCoche(Coche c), se gestiona la entrada del hilo se comprueba si hay cabinas disponibles y si las hay se le asocia a ese coche una cabina y entra a ella, en el caso de que no haya cabinas disponibles se bloquea el hilo hasta que en la salida se libere

entradaColaPeajeCamion(Camion c), se gestiona la entrada del hilo se comprueba si hay cabinas disponibles y si las hay se le asocia a ese coche una cabina y entra a ella, en el caso de que no haya cabinas disponibles se bloquea el hilo hasta que en la salida se libere

entradaColaPeajeAmbulancia(Ambulancia a), se gestiona la entrada del hilo se comprueba si hay cabinas disponibles y si las hay se le asocia a ese coche una cabina y entra a ella, en el caso de que no haya cabinas disponibles se bloquea el hilo hasta que en la salida se libere

salidaPeajeAmbulancia(Ambulancia a), se gestiona la salida del hilo, se comprueba por que cabina ha salido y se libera al hilo que estaba bloqueado.

salidaPeajeCoche(Coche c), se gestiona la salida del hilo, se comprueba por que cabina ha salido y se libera al hilo que estaba bloqueado

salidaPeajeCamiones(Camion c), se gestiona la salida del hilo, se comprueba por que cabina ha salido y se libera al hilo que estaba bloqueado

CabinaPeajeManual cabinaManualCocheDisponible(), te devuelve la cabina manual de coches libres en ese momento si las hay

CabinaPeajeTarjeta cabinaTarjetaCocheDisponible(), te devuelve la cabina de tarjeta de coches libres en ese momento si las hay

CabinaPeajeManual cabinaManualCamionDisponible(), te devuelve la cabina manual de camiones libres en ese momento si las hay

CabinaPeajeTarjeta cabinaTarjetaCamionDisponible(), te devuelve la cabina de tarjeta de camiones libres en ese momento si las hay

cabinaDisponible(Coche c), te devuelve un booleano comprobando si hay cabinas libres o no

cabinaDisponibleCamion(Camion c), te devuelve un booleano comprobando si hay cabinas libres o no

abreCabinas(String cabina), método que se usara en la parte distribuida para abrir las cabinas

getCabinaCoches1(), getCabinaCoches2(), getCabinaCoches3(), getCabinaCoches4(), getCabinaCoches5(), getCabinaCoches6().

getCabinaCamiones1(), getCabinaCamiones2(), getCabinaCamiones3(), getCabinaCamiones4()

isCabinaCocheDisponible()

isCabinaCamionDisponible()

## Paquete Distribuida

### InterfazPeaje.java

Los métodos de esta clase son:

void cierraCabinaCoche4() throws RemoteException;

void abreCabinaCoche4() throws RemoteException;

void cierraCabinaCoche5() throws RemoteException;

void abreCabinaCoche5() throws RemoteException;

void cierraCabinaCoche6() throws RemoteException;

void abreCabinaCoche6() throws RemoteException;

```
void cierraCabinaCamion3() throws RemoteException;  
void abreCabinaCamion3() throws RemoteException;  
void cierraCabinaCamion4() throws RemoteException;  
void abreCabinaCamion4() throws RemoteException;  
String devuelveContenidoJTextFields(String cuadroDeTexto) throws RemoteException;
```

### InterfazDistribuida.java

Los atributos de esta clase son:

```
private boolean botonCierraCabinaCoches4 = false;  
private boolean botonCierraCabinaCoches5 = false;  
private boolean botonCierraCabinaCoches6 = false;  
private boolean botonCierraCabinaCamiones3 = false;  
private boolean botonCierraCabinaCamiones4 = false;
```

Los métodos de esta clase son:

```
initComponents();  
botonCerrarCabina5CocheActionPerformed(java.awt.event.ActionEvent evt);  
botonCerrarCabina4CocheActionPerformed(java.awt.event.ActionEvent evt);  
actualizarPrograma();  
botonCerrarCabina6CocheActionPerformed(java.awt.event.ActionEvent evt);  
botonCerrarCabina3CamionActionPerformed(java.awt.event.ActionEvent evt);  
botonCerrarCabina4CamionActionPerformed(java.awt.event.ActionEvent evt);  
setBotonCierraCabinaCoches4(boolean botonCierraCabinaCoches4) {}  
isBotonCierraCabinaCoches5(){}  
setBotonCierraCabinaCoches5(boolean botonCierraCabinaCoches5){}  
isBotonCierraCabinaCoches6(){}  
setBotonCierraCabinaCoches6(boolean botonCierraCabinaCoches6){}  
isBotonCierraCabinaCamiones3(){}  
setBotonCierraCabinaCamiones3(boolean botonCierraCabinaCamiones3){}  
isBotonCierraCabinaCamiones4(){}  

```

### PaqueteAux:

### ListasThreads.java:

Los atributos de esta clase son:

```
ArrayList<Coche> coche;
```

```
ArrayList<Empleado> empleado;
```

```
ArrayList<Ambulancia> ambulancia;
```

```
ArrayList<Camion> camion;
```

```
TextField tf;
```

Los métodos de esta clase son:

El constructor, que tiene como argumentos(TextField tf);

```
meter(Camion n)
```

```
sacar(Camion n)
```

```
limpiar(Camion c)
```

```
imprimir()
```

```
meter(Empleado n)
```

```
sacar(Empleado n)
```

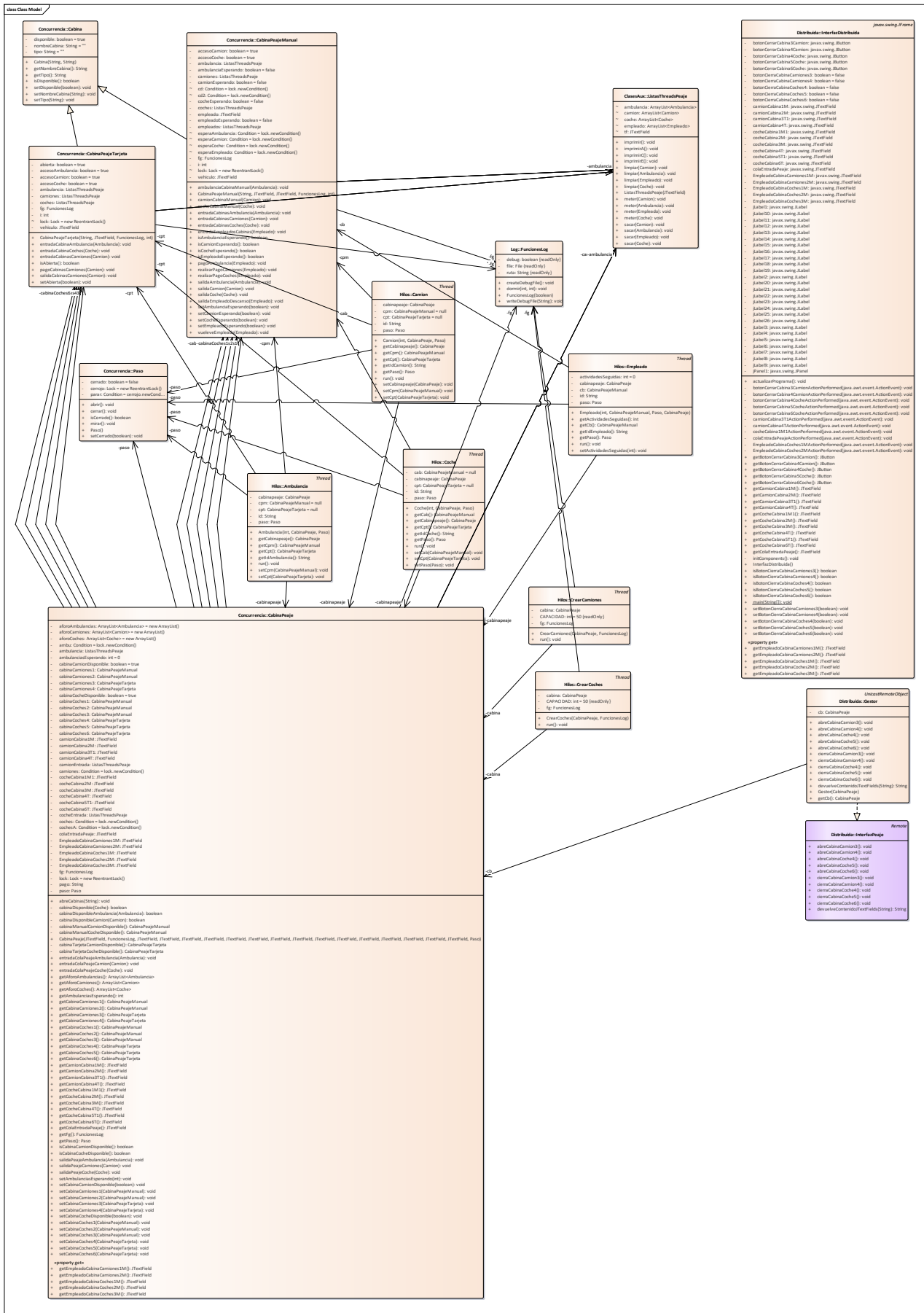
```
limpiar(Empleado c)
```

```
meter(Ambulancia n)
```

```
sacar(Ambulancia n)
```

```
limpiar(Ambulancia c)
```

#### 4-DIAGRAMA DE CLASES



## 5)FUENTE CÓDIGO ANEXO

Paquete Clases Aux

ListasThreads.java

```
/*
```

```
 * To change this license header, choose License Headers in Project Properties.
```

```
 * To change this template file, choose Tools | Templates
```

```
 * and open the template in the editor.
```

```
*/
```

```
package ClasesAux;
```

```
import Hilos.Ambulancia;
```

```
import Hilos.Camion;
```

```
import Hilos.Coche;
```

```
import Hilos.Empleado;
```

```
import java.util.ArrayList;
```

```
import javax.swing.JTextField;
```

```
/**
```

```
 *
```

```
 * @author david
```

```
*/
```

```
public class ListasThreadsPeaje {
```

```
    ArrayList<Coche> coche;
```

```
    ArrayList<Camion> camion;
```

```
    ArrayList<Ambulancia> ambulancia;
```

```
    ArrayList<Empleado> empleado;
```

```
    JTextField tf;
```

```
    public ListasThreadsPeaje(JTextField tf) {
```

```
        this.tf = tf;
```

```
        ambulancia=new ArrayList<Ambulancia>();
```

```
camion=new ArrayList<Camion>();  
empleado = new ArrayList<Empleado>() ;  
coche=new ArrayList<Coche>();  
}
```

```
public synchronized void meter(Camion n)  
{  
    camion.add(n);  
    imprimirC();  
}
```

```
public synchronized void sacar(Camion n)  
{  
    camion.remove(n);  
    imprimirC();  
}
```

```
public void imprimirC()  
{  
    String contenido="";  
    for(int i=0; i<camion.size(); i++)  
    {  
        contenido=contenido+camion.get(i).getIdCamion()+" ";  
    }  
    tf.setText(contenido);  
}
```

```
public void limpiar(Camion c){
```

```
for(int i = 0; i<camion.size(); i++){  
    sacar(c);  
    imprimirC();  
}  
}
```

```
public synchronized void meter(Ambulancia n)  
{  
    ambulancia.add(n);  
    imprimirA();  
}
```

```
public synchronized void sacar(Ambulancia n)  
{  
    ambulancia.remove(n);  
    imprimirA();  
}
```

```
public void imprimirA()  
{  
    String contenido="";  
    for(int i=0; i<ambulancia.size(); i++)  
    {  
        contenido=contenido+ambulancia.get(i).getIdAmbulancia()+" ";  
    }  
    tf.setText(contenido);  
}
```



```
public void limpiar(Ambulancia c){  
    for(int i = 0; i<ambulancia.size(); i++){  
        sacar(c);  
        imprimirA();  
    }  
}
```

```
public synchronized void meter(Empleado n)  
{  
    empleado.add(n);  
    imprimirE();  
}
```

```
public synchronized void sacar(Empleado n)  
{  
    empleado.remove(n);  
    imprimirE();  
}
```

```
public void imprimirE()  
{  
    String contenido="";  
    for(int i=0; i<empleado.size(); i++){  
        {  
            contenido=contenido+empleado.get(i).getIdEmpleado()+" ";  
        }  
    }  
    tf.setText(contenido);  
}
```

```
public void limpiar(Empleado c){  
    for(int i = 0; i<empleado.size(); i++){  
        sacar(c);  
        imprimirE();  
    }  
}
```

```
public synchronized void meter(Coche n)  
{  
    coche.add(n);  
    imprimir();  
}
```

```
public synchronized void sacar(Coche n)  
{  
    coche.remove(n);  
    imprimir();  
}
```

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

```
}
```

```
public void limpiar(Coche c){  
    for(int i = 0; i<coche.size(); i++){  
        sacar(c);  
        imprimir();  
    }  
}
```

```
}
```

## Paquete Concurrency

Cabina.java

```
/*
```

```
 * To change this license header, choose License Headers in Project Properties.
```

```
 * To change this template file, choose Tools | Templates
```

```
 * and open the template in the editor.
```

```
*/
```

```
package Concurrency;
```

```
/**
```

```
 *
```

```
 * @author david
```

```
 */
```

```
public class Cabina {
```

```
    private String nombreCabina = "";
```

```
    private String tipo = "";
```

```
    private boolean disponible = true;
```

```
    public Cabina(String nombre, String tipo) {
```

```
        this.nombreCabina = nombre;
```

```
    this.tipo = tipo;  
}
```

```
public String getNombreCabina() {  
    return nombreCabina;  
}
```

```
public void setNombreCabina(String nombreCabina) {  
    this.nombreCabina = nombreCabina;  
}
```

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

```
public void setTipo(String tipo) {  
    this.tipo = tipo;  
}
```

```
public boolean isDisponible() {  
    return disponible;  
}
```

```
public void setDisponible(boolean disponible) {  
    this.disponible = disponible;  
}
```

```
}
```

CabinaPeaje.java

```
/*
```

```
 * To change this license header, choose License Headers in Project Properties.
```

```
 * To change this template file, choose Tools | Templates
```

```
 * and open the template in the editor.
```

```
*/
```

```
package Concurrencia;
```

```
import ClasesAux.ListasThreadsPeaje;
```

```
import Hilos.*;
```

```
import Log.*;
```

```
import java.util.ArrayList;
```

```
import java.util.Random;
```

```
import java.util.concurrent.locks.Condition;
```

```
import java.util.concurrent.locks.Lock;
```

```
import java.util.concurrent.locks.ReentrantLock;
```

```
import java.util.logging.Level;
```

```
import java.util.logging.Logger;
```

```
import javax.swing.JTextField;
```

```
/**
```

```
 *
```

```
 * @author david
```

```
*/
```

```
public class CabinaPeaje {
```

```
    //Elementos de la interfaz
```

```
    private JTextField colaEntradaPeaje;
```

```
    private JTextField EmpleadoCabinaCamiones1M;
```

```
private JTextField EmpleadoCabinaCamiones2M;  
private JTextField EmpleadoCabinaCoches1M;  
private JTextField EmpleadoCabinaCoches2M;  
private JTextField EmpleadoCabinaCoches3M;  
private JTextField camionCabina1M;  
private JTextField camionCabina2M;  
private JTextField camionCabina3T1;  
private JTextField camionCabina4T;  
private JTextField cocheCabina1M1;  
private JTextField cocheCabina2M;  
private JTextField cocheCabina3M;  
private JTextField cocheCabina4T;  
private JTextField cocheCabina5T1;  
private JTextField cocheCabina6T;
```

```
private String pago;
```

```
//Clases concurrentes
```

```
private CabinaPeajeManual cabinaCoches1, cabinaCoches2, cabinaCoches3,  
cabinaCamiones1, cabinaCamiones2;
```

```
private CabinaPeajeTarjeta cabinaCoches4, cabinaCoches5, cabinaCoches6,  
cabinaCamiones3, cabinaCamiones4;
```

```
private Paso paso;
```

```
private boolean cabinaCocheDisponible = true, cabinaCamionDisponible = true;
```

```
//Clases log
```

```
private FuncionesLog fg;
```

```
//
```

```
private Lock lock = new ReentrantLock();
```

```
private Condition coches = lock.newCondition();
```

```
private Condition cochesA = lock.newCondition();
```

```
private Condition camiones = lock.newCondition();
```

```
private Condition ambu = lock.newCondition();

private ArrayList<Coche> aforoCoches = new ArrayList();

private ArrayList<Camion> aforoCamiones = new ArrayList();

private ArrayList<Ambulancia> aforoAmbulancias = new ArrayList();

private ListasThreadsPeaje cocheEntrada,camionEntrada, ambulancia;

private int ambulanciasEsperando = 0;
```

```
//Crear constructor con sus metodos
```

```
public CabinaPeaje(JTextField colaEntrada, FuncionesLog fg, JTextField cocheCabina1M,
JTextField EmpleadoCabinaCoches1M, JTextField cocheCabina2M, JTextField
EmpleadoCabinaCoches2M, JTextField cocheCabina3M, JTextField EmpleadoCabinaCoches3M,
JTextField camionCabina1M, JTextField EmpleadoCabinaCamiones1M, JTextField
camionCabina2M, JTextField EmpleadoCabinaCamiones2M, JTextField cocheCabina4T,
JTextField cocheCabina5T1, JTextField cocheCabina6T, JTextField camionCabina3T1, JTextField
camionCabina4T, Paso paso) {
```

```
    this.colaEntradaPeaje = colaEntrada;

    cocheEntrada = new ListasThreadsPeaje(colaEntrada);

    camionEntrada = new ListasThreadsPeaje(colaEntrada);

    ambulancia = new ListasThreadsPeaje(colaEntrada);

    this.fg = fg;

    this.paso = paso;

    this.EmpleadoCabinaCamiones1M = EmpleadoCabinaCamiones1M;
    this.EmpleadoCabinaCamiones2M = EmpleadoCabinaCamiones2M;
    this.EmpleadoCabinaCoches1M = EmpleadoCabinaCoches1M;
    this.EmpleadoCabinaCoches2M = EmpleadoCabinaCoches2M;
    this.EmpleadoCabinaCoches3M = EmpleadoCabinaCoches3M;
    this.cocheCabina1M1 = cocheCabina1M;
    this.cocheCabina2M = cocheCabina2M;
    this.cocheCabina3M = cocheCabina3M;
    this.cocheCabina4T = cocheCabina4T;
    this.cocheCabina5T1 = cocheCabina5T1;
    this.cocheCabina6T = cocheCabina6T;
    this.camionCabina1M = camionCabina1M;
```

```

this.camionCabina2M = camionCabina2M;

this.camionCabina3T1 = camionCabina3T1;

this.camionCabina4T = camionCabina4T;

this.cabinaCoches1 = new CabinaPeajeManual("CabinaCoches1", cocheCabina1M,
EmpleadoCabinaCoches1M, fg, 1);

this.cabinaCoches2 = new CabinaPeajeManual("CabinaCoches2", cocheCabina2M,
EmpleadoCabinaCoches2M, fg, 2);

this.cabinaCoches3 = new CabinaPeajeManual("CabinaCoches3", cocheCabina3M,
EmpleadoCabinaCoches3M, fg, 3);

this.cabinaCamiones1 = new CabinaPeajeManual("CabinaCamiones1", camionCabina1M,
EmpleadoCabinaCamiones1M, fg, 4);

this.cabinaCamiones2 = new CabinaPeajeManual("CabinaCamiones2", camionCabina2M,
EmpleadoCabinaCamiones2M, fg, 5);

this.cabinaCoches4 = new CabinaPeajeTarjeta("CabinaCoches4", cocheCabina4T, fg, 1);

this.cabinaCoches5 = new CabinaPeajeTarjeta("CabinaCoches5", cocheCabina5T1, fg, 2);

this.cabinaCoches6 = new CabinaPeajeTarjeta("CabinaCoches6", cocheCabina6T, fg, 3);

this.cabinaCamiones3 = new CabinaPeajeTarjeta("CabinaCamiones3", camionCabina3T1,
fg, 4);

this.cabinaCamiones4 = new CabinaPeajeTarjeta("CabinaCamiones4", camionCabina4T, fg,
5);

}

```

```

public void entradaColaPeajeCoche(Coche c) {

    Random r = new Random();

    lock.lock();

    try {

        fg.writeDebugFile("El " + c.getIdCoche() + " llega al Peaje \n");

        aforoCoches.add(c);

        if(ambulanciasEsperando > 0){

            try {

                cochesA.await();

            } catch (InterruptedException ex) {

                Logger.getLogger(CabinaPeaje.class.getName()).log(Level.SEVERE, null, ex);

            }

        }

    }

}

```



```

    }
}

if (!cabinaDisponible(c)) {
    cocheEntrada.meter(c);
    try {
        fg.writeDebugFile("El " + c.getIdCoche() + " esta en la cola de espera, al no haber
cabinas disponibles \n");
        coches.await();

    } catch (InterruptedException e) {
        System.out.println(e.toString());
    }
    cocheEntrada.sacar(c);
}
c.setCab(cabinaManualCocheDisponible());

if (c.getCab() == null) {
    c.setCpt(cabinaTarjetaCocheDisponible());

}

} finally {
    lock.unlock();
}
}

public void entradaColaPeajeCamion(Camion c) {
    Random r = new Random();
    lock.lock();
    try {

```

```

fg.writeDebugFile("El " + c.getIdCamion() + " llega al Peaje \n");

aforoCamiones.add(c);

if (!cabinaDisponibleCamion(c)) {
    camionEntrada.meter(c);

    try {
        fg.writeDebugFile("El " + c.getIdCamion() + " esta en la cola de espera, al no haber
cabinas disponibles \n");

        camiones.await();

    } catch (InterruptedException e) {
        System.out.println(e.toString());
    }

    camionEntrada.sacar(c);
}

c.setCpm(cabinaManualCamionDisponible());

if (c.getCpm() == null) {
    c.setCpt(cabinaTarjetaCamionDisponible());

}

camionEntrada.sacar(c);
} finally {
    lock.unlock();
}
}

public void entradaColaPeajeAmbulancia(Ambulancia a) {
    lock.lock();

    try {

```

```

ambulanciasEsperando++;

fg.writeDebugFile("La " + a.getIdAmbulancia()+ " llega al Peaje \n");

aforoAmbulancias.add(a);

if (!cabinaDisponibleAmbulancia(a)) {
    ambulancia.meter(a);
    try {
        fg.writeDebugFile("La " + a.getIdAmbulancia()+ " esta en la cola de espera, al no
haber cabinas disponibles \n");
        ambu.await();

    } catch (InterruptedException e) {
        System.out.println(e.toString());
    }
    ambulancia.sacar(a);
}

a.setCpm(cabinaManualCocheDisponible());

if (a.getCpm() == null) {
    a.setCpt(cabinaTarjetaCocheDisponible());

}

ambulanciasEsperando--;

cochesA.signalAll();

} finally {
    lock.unlock();
}
}

public void salidaPeajeAmbulancia(Ambulancia a) {

```

```

lock.lock();

try {
    if (a.getCpm() == null) {
        a.getCpt().setDisponible(true);
        if (a.getCpt().isAbierta()) {
            ambu.signal();
            aforoAmbulancias.remove(a);
        }
    } else {
        a.getCpm().setDisponible(true);
        ambu.signal();
        aforoAmbulancias.remove(a);
    }

    fg.writeDebugFile("La " + a.getIdAmbulancia() + " ha abandonado el peaje\n");
} finally {
    lock.unlock();
}
}

```

```

public void salidaPeajeCoche(Coche c) {
    lock.lock();
    try {
        if (c.getCab() == null) {
            c.getCpt().setDisponible(true);
            if (c.getCpt().isAbierta()) {
                aforoCoches.remove(c);
                if (getAmbulanciasEsperando() > 0) {
                    ambu.signal();
                } else {
                    coches.signal();
                }
            }
        }
    }
}

```

```

    }
} else {
    c.getCab().setDisponible(true);
    aforoCoches.remove(c);
    if (getAmbulanciasEsperando() > 0) {
        ambu.signal();
    } else {
        coches.signal();
    }
}

fg.writeDebugFile("El " + c.getIdCoche() + " ha abandonado el peaje\n");
} finally {
    lock.unlock();
}
}

```

```

public void salidaPeajeCamiones(Camion c) {
    lock.lock();
    try {
        if (c.getCpm() == null) {
            c.getCpt().setDisponible(true);
            if (c.getCpt().isAbierta()) {
                camiones.signal();
                aforoCamiones.remove(c);
            }
        } else {
            c.getCpm().setDisponible(true);
            camiones.signal();
            aforoCamiones.remove(c);
        }
        fg.writeDebugFile("El " + c.getIdCamion() + " ha abandonado el peaje\n");
    }
}

```

```
    } finally {  
        lock.unlock();  
    }  
}
```

```
private CabinaPeajeManual cabinaManualCocheDisponible() {  
    CabinaPeajeManual cabinaDisponible = null;  
    if (cabinaCoches1.isDisponible()) {  
        cabinaCoches1.setDisponible(false);  
        cabinaDisponible = cabinaCoches1;  
    } else if (cabinaCoches2.isDisponible()) {  
        cabinaCoches2.setDisponible(false);  
        cabinaDisponible = cabinaCoches2;  
    } else if (cabinaCoches3.isDisponible()) {  
        cabinaCoches3.setDisponible(false);  
        cabinaDisponible = cabinaCoches3;  
    }  
    return cabinaDisponible;  
}
```

```
private CabinaPeajeTarjeta cabinaTarjetaCocheDisponible() {  
    CabinaPeajeTarjeta cabinaDisponible = null;  
    if (cabinaCoches4.isDisponible()) {  
        cabinaCoches4.setDisponible(false);  
        if (cabinaCoches4.isAbierta()) {  
            cabinaDisponible = cabinaCoches4;  
        }  
    } else if (cabinaCoches5.isDisponible()) {  
        cabinaCoches5.setDisponible(false);  
        if (cabinaCoches5.isAbierta()) {  
            cabinaDisponible = cabinaCoches5;  
        }  
    }  
}
```

```

    }
} else if (cabinaCoches6.isDisponible()) {
    cabinaCoches6.setDisponible(false);
    if (cabinaCoches6.isAbierta()) {
        cabinaDisponible = cabinaCoches6;
    }
}
return cabinaDisponible;
}

```

```

private CabinaPeajeManual cabinaManualCamionDisponible() {
    CabinaPeajeManual cabinaDisponible = null;
    if (cabinaCamiones1.isDisponible()) {
        cabinaCamiones1.setDisponible(false);
        cabinaDisponible = cabinaCamiones1;
    } else if (cabinaCamiones2.isDisponible()) {
        cabinaCamiones2.setDisponible(false);
        cabinaDisponible = cabinaCamiones2;
    }
    return cabinaDisponible;
}

```

//Método que asigna una cabina automática a un camion

```

private CabinaPeajeTarjeta cabinaTarjetaCamionDisponible() {
    CabinaPeajeTarjeta cabinaDisponible = null;
    if (cabinaCamiones3.isDisponible()) {
        cabinaCamiones3.setDisponible(false);
        if (cabinaCamiones3.isAbierta()) {
            cabinaDisponible = cabinaCamiones3;
        }
    } else if (cabinaCamiones4.isDisponible()) {

```

```

        cabinaCamiones4.setDisponible(false);

        if (cabinaCamiones4.isAbierta()) {
            cabinaDisponible = cabinaCamiones4;
        }
    }

    return cabinaDisponible;
}

```

```

private boolean cabinaDisponible(Coche c) {

```

```

    if ((cabinaCoches1.isDisponible() || cabinaCoches2.isDisponible()) ||
cabinaCoches3.isDisponible()) {

```

```

        return true;

```

```

    } else {

```

```

        return ((cabinaCoches4.isAbierta() && cabinaCoches4.isDisponible()) ||
(cabinaCoches5.isAbierta() && cabinaCoches5.isDisponible()) || (cabinaCoches6.isAbierta() &&
cabinaCoches6.isDisponible()));

```

```

    }

```

```

}

```

```

private boolean cabinaDisponibleAmbulancia(Ambulancia a) {

```

```

    if ((cabinaCoches1.isDisponible() || cabinaCoches2.isDisponible()) ||
cabinaCoches3.isDisponible()) {

```

```

        return true;

```

```

    } else {

```

```

        return ((cabinaCoches4.isAbierta() && cabinaCoches4.isDisponible()) ||
(cabinaCoches5.isAbierta() && cabinaCoches5.isDisponible()) || (cabinaCoches6.isAbierta() &&
cabinaCoches6.isDisponible()));

```

```

    }

```



```
}
```

```
private boolean cabinaDisponibleCamion(Camion c) {
```

```
    if (cabinaCamiones1.isDisponible() || cabinaCamiones2.isDisponible()) {
```

```
        return true;
```

```
    } else {
```

```
        return ((cabinaCamiones3.isAbierta() && cabinaCamiones3.isDisponible()) ||  
(cabinaCamiones4.isAbierta() && cabinaCamiones4.isDisponible()));
```

```
    }
```

```
}
```

```
public void abreCabinas(String cabina) {
```

```
    //Dependiendo del string, despertaremos a los coches de una cabina u otra
```

```
    lock.lock();
```

```
    try {
```

```
        switch (cabina) {
```

```
            case "CabinaCoches4":
```

```
                cabinaCoches4.setDisponible(true);
```

```
                if (cabinaCoches4.isAbierta()) {
```

```
                    if (getAmbulanciasEsperando() > 0) {
```

```
                        ambu.signal();
```

```
                    } else {
```

```
                        coches.signal();
```

```
                    }
```

```
                } break;
```

```
            case "CabinaCoches5":
```

```
cabinaCoches5.setDisponible(true);
if (cabinaCoches5.isAbierta()) {
    if (getAmbulanciasEsperando() > 0) {
        ambu.signal();
    } else {
        coches.signal();
    }
} break;
case "CabinaCoches6":
    cabinaCoches6.setDisponible(true);
    if (cabinaCoches6.isAbierta()) {
        if (getAmbulanciasEsperando() > 0) {
            ambu.signal();
        } else {
            coches.signal();
        }
    } break;
case "CabinaCamiones3":
    cabinaCamiones3.setAbierta(true);
    if (cabinaCamiones3.isDisponible()) {
        camiones.signal();
    } break;

case "CabinaCamiones4":
    cabinaCamiones4.setAbierta(true);
    if (cabinaCamiones4.isDisponible()) {
        camiones.signal();
    } break;
default:
```

```
        break;
    }
} finally {
    lock.unlock();
}
}
```

```
public CabinaPeajeManual getCabinaCoches1() {
    return cabinaCoches1;
}
```

```
public void setCabinaCoches1(CabinaPeajeManual cabinaCoches1) {
    this.cabinaCoches1 = cabinaCoches1;
}
```

```
public CabinaPeajeManual getCabinaCoches2() {
    return cabinaCoches2;
}
```

```
public void setCabinaCoches2(CabinaPeajeManual cabinaCoches2) {
    this.cabinaCoches2 = cabinaCoches2;
}
```

```
public CabinaPeajeManual getCabinaCoches3() {
    return cabinaCoches3;
}
```

```
public void setCabinaCoches3(CabinaPeajeManual cabinaCoches3) {
    this.cabinaCoches3 = cabinaCoches3;
}
```

```
public CabinaPeajeManual getCabinaCamiones1() {  
    return cabinaCamiones1;  
}
```

```
public void setCabinaCamiones1(CabinaPeajeManual cabinaCamiones1) {  
    this.cabinaCamiones1 = cabinaCamiones1;  
}
```

```
public CabinaPeajeManual getCabinaCamiones2() {  
    return cabinaCamiones2;  
}
```

```
public void setCabinaCamiones2(CabinaPeajeManual cabinaCamiones2) {  
    this.cabinaCamiones2 = cabinaCamiones2;  
}
```

```
public CabinaPeajeTarjeta getCabinaCoches4() {  
    return cabinaCoches4;  
}
```

```
public void setCabinaCoches4(CabinaPeajeTarjeta cabinaCoches4) {  
    this.cabinaCoches4 = cabinaCoches4;  
}
```

```
public CabinaPeajeTarjeta getCabinaCoches5() {  
    return cabinaCoches5;  
}
```

```
public void setCabinaCoches5(CabinaPeajeTarjeta cabinaCoches5) {  
    this.cabinaCoches5 = cabinaCoches5;  
}
```

```
}
```

```
public CabinaPeajeTarjeta getCabinaCoches6() {  
    return cabinaCoches6;  
}
```

```
public void setCabinaCoches6(CabinaPeajeTarjeta cabinaCoches6) {  
    this.cabinaCoches6 = cabinaCoches6;  
}
```

```
public CabinaPeajeTarjeta getCabinaCamiones3() {  
    return cabinaCamiones3;  
}
```

```
public void setCabinaCamiones3(CabinaPeajeTarjeta cabinaCamiones3) {  
    this.cabinaCamiones3 = cabinaCamiones3;  
}
```

```
public CabinaPeajeTarjeta getCabinaCamiones4() {  
    return cabinaCamiones4;  
}
```

```
public void setCabinaCamiones4(CabinaPeajeTarjeta cabinaCamiones4) {  
    this.cabinaCamiones4 = cabinaCamiones4;  
}
```

```
public Paso getPaso() {  
    return paso;  
}
```

```
public boolean isCabinaCocheDisponibile() {
```

```
    return cabinaCocheDisponible;
}
```

```
public void setCabinaCocheDisponible(boolean cabinaCocheDisponible) {
    this.cabinaCocheDisponible = cabinaCocheDisponible;
}
```

```
public boolean isCabinaCamionDisponible() {
    return cabinaCamionDisponible;
}
```

```
public void setCabinaCamionDisponible(boolean cabinaCamionDisponible) {
    this.cabinaCamionDisponible = cabinaCamionDisponible;
}
```

```
public int getAmbulanciasEsperando() {
    return ambulanciasEsperando;
}
```

```
public void setAmbulanciasEsperando(int ambulanciasEsperando) {
    this.ambulanciasEsperando = ambulanciasEsperando;
}
```

```
public JTextField getColaEntradaPeaje() {
    return colaEntradaPeaje;
}
```

```
public JTextField getEmpleadoCabinaCamiones2M() {
    return EmpleadoCabinaCamiones2M;
}
```

```
public JTextField getEmpleadoCabinaCamiones1M() {  
    return EmpleadoCabinaCamiones1M;  
}
```

```
public JTextField getEmpleadoCabinaCoches1M() {  
    return EmpleadoCabinaCoches1M;  
}
```

```
public JTextField getEmpleadoCabinaCoches2M() {  
    return EmpleadoCabinaCoches2M;  
}
```

```
public JTextField getEmpleadoCabinaCoches3M() {  
    return EmpleadoCabinaCoches3M;  
}
```

```
public JTextField getCamionCabina1M() {  
    return camionCabina1M;  
}
```

```
public JTextField getCamionCabina2M() {  
    return camionCabina2M;  
}
```

```
public JTextField getCamionCabina3T1() {  
    return camionCabina3T1;  
}
```

```
public JTextField getCamionCabina4T() {  
    return camionCabina4T;  
}
```

```
public JTextField getCocheCabina1M1() {  
    return cocheCabina1M1;  
}
```

```
public JTextField getCocheCabina2M() {  
    return cocheCabina2M;  
}
```

```
public JTextField getCocheCabina3M() {  
    return cocheCabina3M;  
}
```

```
public JTextField getCocheCabina4T() {  
    return cocheCabina4T;  
}
```

```
public JTextField getCocheCabina5T1() {  
    return cocheCabina5T1;  
}
```

```
public JTextField getCocheCabina6T() {  
    return cocheCabina6T;  
}
```

```
public FuncionesLog getFg() {  
    return fg;  
}
```

```
public ArrayList<Coche> getAforoCoches() {  
    return aforoCoches;  
}
```



```
}
```

```
public ArrayList<Camion> getAforoCamiones() {  
    return aforoCamiones;  
}
```

```
public ArrayList<Ambulancia> getAforoAmbulancias() {  
    return aforoAmbulancias;  
}
```

```
}
```

CabinaPeajeManual.java

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package Concurrencia;
```

```
import ClasesAux.ListasThreadsPeaje;
```

```
import Hilos.Ambulancia;
```

```
import Hilos.Camion;
```

```
import Hilos.Coche;
```

```
import Hilos.Empleado;
```

```
import Log.FuncionesLog;
```

```
import java.util.ArrayList;
```

```
import java.util.Random;
```

```
import java.util.concurrent.*;
```

```
import java.util.concurrent.locks.Condition;
```

```

import java.util.concurrent.locks.Lock;

import java.util.concurrent.locks.ReentrantLock;

import java.util.logging.Level;

import java.util.logging.Logger;

import static javax.management.Query.or;

import javax.swing.JTextField;

/**
 *
 * @author david
 */

public class CabinaPeajeManual extends Cabina{

    private JTextField vehiculo;

    private JTextField empleado;

    private ListasThreadsPeaje empleados, coches, camiones,ambulancia;

    private FuncionesLog fg;

    private int i;

    private boolean accesoCoche = true;

    private boolean accesoCamion = true;

    private boolean cocheEsperando = false;

    private boolean camionEsperando = false;

    private boolean ambulanciaEsperando = false;

    private boolean empleadoEsperando = false;


    //Concurrencia

    Lock lock = new ReentrantLock();

    Condition esperaEmpleado = lock.newCondition();

    Condition esperaCoche = lock.newCondition();

    Condition esperaCamion = lock.newCondition();

```

```
Condition esperaAmbulancia = lock.newCondition();
```

```
Condition cd = lock.newCondition();
```

```
Condition cd2 = lock.newCondition();
```

```
public CabinaPeajeManual(String nombre, JTextField vehiculo, JTextField empleado,  
FuncionesLog fg, int i) {
```

```
    super(nombre, "Manual");
```

```
    this.fg = fg;
```

```
    this.vehiculo = vehiculo;
```

```
    coches = new ListasThreadsPeaje(vehiculo);
```

```
    camiones = new ListasThreadsPeaje(vehiculo);
```

```
    ambulancia = new ListasThreadsPeaje(vehiculo);
```

```
    this.empleado = empleado;
```

```
    empleados = new ListasThreadsPeaje(empleado);
```

```
    this.i = i;
```

```
}
```

```
public void entradaEmpleadosCabinas(Empleado e){
```

```
    lock.lock();
```

```
    try{
```

```
        empleados.meter(e);
```

```
        fg.writeDebugFile("El "+e.getIdEmpleado()+" accede a la cabina de peaje "+i+"\n");
```

```
        if(!isCocheEsperando() || !isCamionEsperando() || !isAmbulanciaEsperando()){
```

```
            setEmpleadoEsperando(true);
```

```
            try{
```

```
                esperaEmpleado.await();
```

```
            }catch(InterruptedException ar){
```

```
                System.out.println(ar.toString());
```

```
            }
```

```

    }

    }finally{
        lock.unlock();
    }

    }

    public void salidaEmpleadoDescanso(Empleado e) throws InterruptedException{

        lock.lock();
        try{
            empleados.sacar(e);
            fg.writeDebugFile("El "+e.getIdEmpleado()+" comienza su descanso en la cabina "+i+"\n");
            Thread.sleep(5000);
            fg.writeDebugFile("El "+e.getIdEmpleado()+" finaliza su descanso en la cabina "+i+"\n");
            setEmpleadoEsperando(false);
        }finally{
            lock.unlock();
        }
    }

    public void vueleveEmpleado(Empleado e){
        lock.lock();
        try{
            if(!isCocheEsperando() ){
                setEmpleadoEsperando(true);
                try{
                    esperaEmpleado.await();
                }catch(InterruptedException a){
                    System.out.println(a.toString());
                }
            }
        }
    }

```

```
    }  
    }finally{  
        lock.unlock();  
    }  
}
```

```
public void entradaCabinasCoches(Coche coche){
```

```
    lock.lock();
```

```
    try{
```

```
        coches.meter(coche);
```

```
        fg.writeDebugFile("El "+coche.getIdCoche()+" accede a la cabina de peaje "+i+"\n");
```

```
        accesoCoche = false;
```

```
        if(isEmpleadoEsperando()){
```

```
            esperaEmpleado.signal();
```

```
        }else{
```

```
            setCocheEsperando(true);
```

```
        }
```

```
        try{
```

```
            esperaCoche.await();
```

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

```
            System.out.println(e.toString());
```

```
        }
```

```
    }finally{
```

```
        lock.unlock();
```

```

    }

}

public void entradaCabinasAmbulancia(Ambulancia a){
    lock.lock();

    try{

        ambulancia.meter(a);

        fg.writeDebugFile("La "+a.getIdAmbulancia()+" accede a la cabina de peaje "+i+"\n");

        if(isEmpleadoEsperando()){
            esperaEmpleado.signal();
        }else{
            setAmbulanciaEsperando(true);
        }

        try{
            esperaAmbulancia.await();
        }catch(InterruptedException e){
            System.out.println(e.toString());
        }

    }

    finally{
        lock.unlock();
    }

}

public void pagoAmbulancia(Empleado e) throws InterruptedException{
    lock.lock();

    try{

        fg.writeDebugFile("El empleado "+e.getIdEmpleado()+" esta cobrando a la ambulancia
en la cabina "+i+"\n");

        Thread.sleep((int) (Math.random() * 8000 + 6000));
    }
}

```

```

        setAmbulanciaEsperando(false);

        esperaAmbulancia.signal();

        e.setActividadesSeguidas(e.getActividadesSeguidas()+1);
    }finally{
        lock.unlock();
    }
}

public void realizarPagoCoches(Empleado e) throws InterruptedException{

    lock.lock();

    try{

        fg.writeDebugFile("El empleado "+e.getIdEmpleado()+" esta cobrando al coche en la
cabina "+i+"\n");

        Thread.sleep((int) (Math.random() * 8000 + 6000));

        setCocheEsperando(false);

        esperaCoche.signal();

        e.setActividadesSeguidas(e.getActividadesSeguidas()+1);
    }finally{
        lock.unlock();
    }

}

public void realizarPagoCamiones(Empleado e) throws InterruptedException{

    lock.lock();

    try{

        fg.writeDebugFile("El empleado "+e.getIdEmpleado()+" esta cobrando al camion en la
cabina "+i+"\n");

        Thread.sleep((int) (Math.random() * 8000 + 6000));

        setCamionEsperando(false);

        esperaCamion.signal();

        e.setActividadesSeguidas(e.getActividadesSeguidas()+1);
    }
}

```

```

    }finally{

    lock.unlock();

}

}

public void salidaCoche(Coche coche){

    coches.sacar(coche);

    fg.writeDebugFile("El "+coche.getIdCoche()+" sale de la cabina de peaje de vuelta a la
carretera \n");

    accesoCoche = true;

}

public void cocheCabinaManual(Coche coche){

    lock.lock();

    try{

        entradaCabinasCoches(coche);

        salidaCoche(coche);

    }finally{

        lock.unlock();

    }

}

}

public void entradaCabinasCamiones(Camion camion){

    lock.lock();

    try{

        camiones.meter(camion);

        fg.writeDebugFile("El "+camion.getIdCamion()+" accede a la cabina de peaje "+i+"\n");

        accesoCamion = false;

        if(isEmpleadoEsperando()){

            esperaEmpleado.signal();

        }else{

            setCamionEsperando(true);

        }

    }

}

}

```



```

    }

    try{
        esperaCamion.await();
    }catch(InterruptedException e){
        System.out.println(e.toString());
    }

}finally{
    lock.unlock();
}

}

public void salidaCamion(Camion camion){
    camiones.sacar(camion);

    fg.writeDebugFile("El "+camion.getIdCamion()+" sale de la cabina de peaje de vuelta a la
carretera \n");

    accesoCamion = true;
}

public void camionCabinaManual(Camion camion){
    lock.lock();

    try{
        entradaCabinasCamiones(camion);

        salidaCamion(camion);
    }finally{
        lock.unlock();
    }
}

}

public void salidaAmbulancia(Ambulancia am){
    ambulancia.sacar(am);
}

```

```
        fg.writeDebugFile("La "+am.getIdAmbulancia()+" sale de la cabina de peaje de vuelta a la  
carretera \n");
```

```
    }
```

```
    public void ambulanciaCabinaManual(Ambulancia am){
```

```
        lock.lock();
```

```
        try{
```

```
            entradaCabinasAmbulancia(am);
```

```
            salidaAmbulancia(am);
```

```
        }finally{
```

```
            lock.unlock();
```

```
    }
```

```
}
```

```
public boolean isCocheEsperando() {
```

```
    return cocheEsperando;
```

```
}
```

```
public void setCocheEsperando(boolean cocheEsperando) {
```

```
    this.cocheEsperando = cocheEsperando;
```

```
}
```

```
public boolean isCamionEsperando() {
```

```
    return camionEsperando;
```

```
}
```

```
public void setCamionEsperando(boolean camionEsperando) {
```

```
    this.camionEsperando = camionEsperando;
```

```
}
```

```
public boolean isEmpleadoEsperando() {  
    return empleadoEsperando;  
}
```

```
public void setEmpleadoEsperando(boolean empleadoEsperando) {  
    this.empleadoEsperando = empleadoEsperando;  
}
```

```
public boolean isAmbulanciaEsperando() {  
    return ambulanciaEsperando;  
}
```

```
public void setAmbulanciaEsperando(boolean ambulanciaEsperando) {  
    this.ambulanciaEsperando = ambulanciaEsperando;  
}  
}
```

CabinaPeajeTarjeta.java

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

```
package Concurrencia;
```

```
import ClasesAux.ListasThreadsPeaje;
```

```
import Hilos.Ambulancia;
```

```
import Hilos.Camion;
```

```
import Hilos.Coche;
```

```
import Hilos.Empleado;
```

```
import Log.FuncionesLog;

import java.util.Random;

import java.util.concurrent.locks.Lock;

import java.util.concurrent.locks.ReentrantLock;

import java.util.concurrent.locks.ReentrantReadWriteLock;

import javax.swing.JTextField;

/**
 *
 * @author david
 */
public class CabinaPeajeTarjeta extends Cabina {

    private boolean abierta = true;

    private JTextField vehiculo;

    private ListasThreadsPeaje coches, camiones, ambulancia;

    private FuncionesLog fg;

    private boolean accesoCoche = true;

    private boolean accesoCamion = true;

    private boolean accesoAmbulancia = true;

    private int i;

    Lock lock = new ReentrantLock();

    public CabinaPeajeTarjeta(String nombre, JTextField vehiculo, FuncionesLog fg, int i) {

        super(nombre, "Tarjeta");

        this.vehiculo = vehiculo;

        this.fg = fg;

        this.i = i;

        coches = new ListasThreadsPeaje(vehiculo);

        camiones = new ListasThreadsPeaje(vehiculo);

        ambulancia = new ListasThreadsPeaje(vehiculo);
```

```
}
```

```
public void entradaCabinasCamiones(Camion camion){
```

```
    Random r = new Random();
```

```
    lock.lock();
```

```
    try{
```

```
        camiones.meter(camion);
```

```
        fg.writeDebugFile("El "+camion.getIdCamion()+" accede a la cabina de peaje "+i+"\n");
```

```
    }finally{
```

```
        lock.unlock();
```

```
    }
```

```
}
```

```
public void pagoCabinasCamiones(Camion camion) throws InterruptedException{
```

```
    Random r = new Random();
```

```
    lock.lock();
```

```
    try{
```

```
        fg.writeDebugFile("El "+camion.getIdCamion()+" esta pagando en la cabina de peaje "+i+"\n");
```

```
        Thread.sleep((int)(Math.random() * 8000 + 6000));
```

```
    }finally{
```

```
        lock.unlock();
```

```
    }
```

```
}
```

```
public void salidaCabinasCamiones(Camion camion){  
    Random r = new Random();  
    lock.lock();  
    try{  
        camiones.sacar(camion);  
        fg.writeDebugFile("El "+camion.getIdCamion()+" sale de la cabina de peaje "+i+"\n");  
  
    }finally{  
        lock.unlock();  
    }  
}
```

```
public void entradaCabinaCoche(Coche cc){  
    Random r = new Random();  
    lock.lock();  
    try{  
        accesoCoche = false;  
        coches.meter(cc);  
        fg.writeDebugFile("El "+cc.getIdCoche()+" accede a la cabina de peaje "+i+"\n");  
        Thread.sleep(6000+r.nextInt(8000));  
        coches.sacar(cc);  
        fg.writeDebugFile("El "+cc.getIdCoche()+" sale de la cabina de peaje de vuelta a la  
carretera\n");  
        accesoCoche = true;  
  
    }catch(InterruptedException e){  
        System.out.println(e.toString());  
    }finally{  

```

```
        lock.unlock();  
    }  
  
}
```

```
public void entradaCabinaAmbulancia(Ambulancia a){  
    Random r = new Random();  
    lock.lock();  
    try{  
        accesoAmbulancia = false;  
        ambulancia.meter(a);  
        fg.writeDebugFile("La "+a.getIdAmbulancia()+" accede a la cabina de peaje "+i+"\n");  
  
        Thread.sleep(6000+r.nextInt(8000));  
        ambulancia.sacar(a);  
        fg.writeDebugFile("La "+a.getIdAmbulancia()+" sale de la cabina de peaje de vuelta a la  
carretera\n");  
        accesoAmbulancia = true;  
  
    }catch(InterruptedException e){  
        System.out.println(e.toString());  
    }finally{  
        lock.unlock();  
    }  
  
}
```

```
public boolean isAbierta() {
```

```
        return abierta;
    }

    public void setAbierta(boolean abierta) {
        this.abierta = abierta;
    }
}
```

Paso.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Concurrencia;

import Hilos.Coche;
import java.util.concurrent.locks.Condition;
import java.util.concurrent.locks.Lock;
import java.util.concurrent.locks.ReentrantLock;

/**
 *
 * @author david
 */
public class Paso {
    private boolean cerrado = false;
```



```
private Lock cerrojo = new ReentrantLock();
private Condition parar = cerrojo.newCondition();

//Métodos de la clase paso

//Método constructor
public Paso(){

}

//Método que mira si nos tenemos que detener o no
public void mirar() {
    try {
        cerrojo.lock();
        while(isCerrado()) {
            try {
                parar.await();
            } catch (InterruptedException ie){ }
        }
    }
    finally {
        cerrojo.unlock();
    }
}

//Método para reanudar ejecucion de hilos
public void abrir() {
    try{
        cerrojo.lock();
        setCerrado(false);
        parar.signalAll();
    }
    finally {
```

```

        cerrojo.unlock();
    }
}

//Métodos para reanuar ejecucion de hilos
public void cerrar() {
    try{
        cerrojo.lock();
        setCerrado(true);
    }
    finally
    {
        cerrojo.unlock();
    }
}

```

```

public boolean isCerrado() {
    return cerrado;
}

```

```

public void setCerrado(boolean cerrado) {
    this.cerrado = cerrado;
}
}

```

## Paquete Distribuida

Gestor.java

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Distribuida;

```

import Concurrencia.CabinaPeaje;

import java.rmi.RemoteException;

import java.rmi.server.UnicastRemoteObject;

/**
 *
 * @author david
 */
public class Gestor extends UnicastRemoteObject implements InterfazPeaje {
    private CabinaPeaje cb;

    public Gestor(CabinaPeaje cb) throws RemoteException {
        this.cb = cb;
    }

    @Override
    public void cierraCabinaCoche4(){
        cb.getCabinaCoches4().setAbierta(false);
    }

    @Override
    public void abreCabinaCoche4(){

        cb.abreCabinas("CabinaCoches4");
    }

    @Override
    public void cierraCabinaCoche5(){
        cb.getCabinaCoches5().setAbierta(false);
    }
}

```

@Override

```
public void abreCabinaCoche5(){
```

```
    cb.abreCabinas("CabinaCoches5");  
}
```

@Override

```
public void cierraCabinaCoche6(){
```

```
    cb.getCabinaCoches6().setAbierta(false);  
}
```

@Override

```
public void abreCabinaCoche6(){
```

```
    cb.abreCabinas("CabinaCoches6");  
}
```

@Override

```
public void cierraCabinaCamion3(){
```

```
    cb.getCabinaCamiones3().setAbierta(false);  
}
```

@Override

```
public void abreCabinaCamion3(){
```

```
    cb.abreCabinas("CabinaCamiones3");  
}
```

```
public void cierraCabinaCamion4(){
```

```
    cb.getCabinaCamiones4().setAbierta(false);
```

```
}
```

```
@Override
```

```
public void abreCabinaCamion4(){
```

```
    cb.abreCabinas("CabinaCamiones4");
```

```
}
```

```
public String devuelveContenidoJTextFields(String contenidoC){
```

```
    String contenido = "";
```

```
    switch (contenidoC) {
```

```
        case "colaEntradaPeaje":
```

```
            contenido = cb.getColaEntradaPeaje().getText();
```

```
            break;
```

```
        case "cocheCabina1M1":
```

```
            contenido = cb.getCocheCabina1M1().getText();
```

```
            break;
```

```
        case "cocheCabina2M":
```

```
            contenido = cb.getCocheCabina2M().getText();
```

```
            break;
```

```
        case "cocheCabina3M":
```

```
            contenido = cb.getCocheCabina3M().getText();
```

```
            break;
```

```
        case "cocheCabina4T":
```

```
            contenido = cb.getCocheCabina4T().getText();
```

```
            break;
```

```
        case "cocheCabina5T1":
```

```
            contenido = cb.getCocheCabina5T1().getText();
```

```
            break;
```

```
        case "cocheCabina6T":
```

```
            contenido = cb.getCocheCabina6T().getText();
```

```
            break;
```

```
case "camionCabina1M":
    contenido = cb.getCamionCabina1M().getText();
    break;
case "camionCabina2M":
    contenido = cb.getCamionCabina2M().getText();
    break;
case "camionCabina3T1":
    contenido = cb.getCamionCabina3T1().getText();
    break;
case "camionCabina4T":
    contenido = cb.getCamionCabina4T().getText();
    break;
case "EmpleadoCabinaCoche1M":
    contenido = cb.getEmpleadoCabinaCoche1M().getText();
    break;
case "EmpleadoCabinaCoche2M":
    contenido = cb.getEmpleadoCabinaCoche2M().getText();
    break;
case "EmpleadoCabinaCoche3M":
    contenido = cb.getEmpleadoCabinaCoche3M().getText();
    break;
case "EmpleadoCabinaCamiones1M":
    contenido = cb.getEmpleadoCabinaCamiones1M().getText();
    break;
case "EmpleadoCabinaCamiones2M":
    contenido = cb.getEmpleadoCabinaCamiones2M().getText();
    break;
default:
    break;
}
return contenido;
```

```
}
```

```
public CabinaPeaje getCb() {  
    return cb;  
}
```

```
}
```

InterfazDistribuida.java

```
/*
```

```
 * To change this license header, choose License Headers in Project Properties.
```

```
 * To change this template file, choose Tools | Templates
```

```
 * and open the template in the editor.
```

```
*/
```

```
package Distribuida;
```

```
import java.rmi.Naming;
```

```
import javax.swing.JButton;
```

```
import javax.swing.JTextField;
```

```
/**
```

```
 *
```

```
 * @author david
```

```
*/
```

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

```
    /**
```

```
     * Creates new form InterfazDistribuida
```

```
     */
```

```
    private boolean botonCierraCabinaCoche4 = false;
```

```
    private boolean botonCierraCabinaCoche5 = false;
```

```
private boolean botonCierraCabinaCoches6 = false;
private boolean botonCierraCabinaCamiones3 = false;
private boolean botonCierraCabinaCamiones4 = false;
```

```
public InterfazDistribuida() {
    initComponents();
}
}
```

```
/**
 * 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">
private void initComponents() {
```

```
    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    colaEntradaPeaje = new javax.swing.JTextField();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();
    cocheCabina1M1 = new javax.swing.JTextField();
    EmpleadoCabinaCoches1M = new javax.swing.JTextField();
```



```
jLabel7 = new javax.swing.JLabel();
jLabel8 = new javax.swing.JLabel();
cocheCabina2M = new javax.swing.JTextField();
EmpleadoCabinaCoche2M = new javax.swing.JTextField();
jLabel9 = new javax.swing.JLabel();
jLabel10 = new javax.swing.JLabel();
cocheCabina3M = new javax.swing.JTextField();
EmpleadoCabinaCoche3M = new javax.swing.JTextField();
jLabel11 = new javax.swing.JLabel();
jLabel12 = new javax.swing.JLabel();
jLabel13 = new javax.swing.JLabel();
jLabel14 = new javax.swing.JLabel();
jLabel15 = new javax.swing.JLabel();
jLabel16 = new javax.swing.JLabel();
cocheCabina4T = new javax.swing.JTextField();
cocheCabina5T1 = new javax.swing.JTextField();
cocheCabina6T = new javax.swing.JTextField();
botonCerrarCabina5Coche = new javax.swing.JButton();
botonCerrarCabina6Coche = new javax.swing.JButton();
botonCerrarCabina4Coche = new javax.swing.JButton();
jLabel17 = new javax.swing.JLabel();
jLabel18 = new javax.swing.JLabel();
jLabel19 = new javax.swing.JLabel();
jLabel20 = new javax.swing.JLabel();
jLabel21 = new javax.swing.JLabel();
jLabel22 = new javax.swing.JLabel();
camionCabina1M = new javax.swing.JTextField();
EmpleadoCabinaCamiones1M = new javax.swing.JTextField();
camionCabina2M = new javax.swing.JTextField();
EmpleadoCabinaCamiones2M = new javax.swing.JTextField();
jLabel23 = new javax.swing.JLabel();
```

```
jLabel24 = new javax.swing.JLabel();
jLabel25 = new javax.swing.JLabel();
jLabel26 = new javax.swing.JLabel();
camionCabina4T = new javax.swing.JTextField();
camionCabina3T1 = new javax.swing.JTextField();
botonCerrarCabina4Camion = new javax.swing.JButton();
botonCerrarCabina3Camion = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel1.setText("Vehículos esperando para entrar al peaje");

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

jLabel2.setText("Cabina Coches1(Manual)");

jLabel3.setText("Cabina Coches2(Manual)");

jLabel4.setText("Cabina Coches3(Manual)");

jLabel5.setText("Vehículo:");

jLabel6.setText("Empleado:");

cocheCabina1M1.addActionListener(new java.awt.event.ActionListener() {
```

```
public void actionPerformed(java.awt.event.ActionEvent evt) {  
    cocheCabina1M1ActionPerformed(evt);  
}  
});
```

```
EmpleadoCabinaCoches1M.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        EmpleadoCabinaCoches1MActionPerformed(evt);  
    }  
});
```

```
jLabel7.setText("Vehículo:");
```

```
jLabel8.setText("Empleado:");
```

```
EmpleadoCabinaCoches2M.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        EmpleadoCabinaCoches2MActionPerformed(evt);  
    }  
});
```

```
jLabel9.setText("Vehículo:");
```

```
jLabel10.setText("Empleado:");
```

```
jLabel11.setText("Cabina Coches4(Tarjeta)");
```

```
jLabel12.setText("Cabina Coches5(Tarjeta)");
```

```
jLabel13.setText("Cabina Coches6(Tarjeta)");
```

```
jLabel14.setText("Vehículo:");
```

```
jLabel15.setText("Vehículo:");
```

```
jLabel16.setText("Vehículo:");
```

```
botonCerrarCabina5Coche.setText("Cerrar");
```

```
botonCerrarCabina5Coche.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        botonCerrarCabina5CocheActionPerformed(evt);  
    }  
});
```

```
botonCerrarCabina6Coche.setText("Cerrar");
```

```
botonCerrarCabina6Coche.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        botonCerrarCabina6CocheActionPerformed(evt);  
    }  
});
```

```
botonCerrarCabina4Coche.setText("Cerrar");
```

```
botonCerrarCabina4Coche.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        botonCerrarCabina4CocheActionPerformed(evt);  
    }  
});
```

```
jLabel17.setText("Cabina Camiones1(Manual)");
```

```
jLabel18.setText("Cabina Camiones2(Manual)");
```

```
jLabel19.setText("Vehículo:");
```

```
jLabel20.setText("Empleado:");
```

```
jLabel21.setText("Vehículo:");
```

```
jLabel22.setText("Empleado:");
```

```
jLabel23.setText("Cabina Camiones4(Tarjeta)");
```

```
jLabel24.setText("Cabina Camiones3(Tarjeta)");
```

```
jLabel25.setText("Vehículo:");
```

```
jLabel26.setText("Vehículo:");
```

```
camionCabina4T.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        camionCabina4TActionPerformed(evt);  
    }  
});
```

```
camionCabina3T1.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        camionCabina3T1ActionPerformed(evt);  
    }  
});
```

```
botonCerrarCabina4Camion.setText(" Cerrar");
```

```
botonCerrarCabina4Camion.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        botonCerrarCabina4CamionActionPerformed(evt);
    }
});
```

```
botonCerrarCabina3Camion.setText("Cerrar");
botonCerrarCabina3Camion.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        botonCerrarCabina3CamionActionPerformed(evt);
    }
});
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .add(layout.createSequentialGroup()
                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .add(layout.createSequentialGroup()
                                    .addGap(40, 40, 40)
                                )
                            )
                        )
                    )
                )
            )
        )
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .add(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .add(layout.createSequentialGroup()
                        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 446,
                            javax.swing.GroupLayout.PREFERRED_SIZE)
                    )
                )
            )
        )
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
            .add(layout.createSequentialGroup()
                .addComponent(colaEntradaPeaje,
                    javax.swing.GroupLayout.PREFERRED_SIZE, 776, javax.swing.GroupLayout.PREFERRED_SIZE)
            )
        )
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .add(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .add(layout.createSequentialGroup()
                        .addGap(332, 332, 332)
                    )
                    .add(layout.createSequentialGroup()
                        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
                            javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    )
                )
            )
        )
    );
```

```

        .addGroup(layout.createSequentialGroup())
        .addGap(32, 32, 32)
        .addComponent(jLabel2)
        .addGap(498, 498, 498)
        .addComponent(jLabel4))))))
    .addGroup(layout.createSequentialGroup())
    .addContainerGap()

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(jLabel5)
        .addComponent(jLabel6)
        .addComponent(jLabel14))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

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

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addGroup(layout.createSequentialGroup()
            .addComponent(cocheCabina1M1,
javax.swing.GroupLayout.PREFERRED_SIZE, 111, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(130, 130, 130)
            .addComponent(jLabel7))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(EmpleadoCabinaCoches1M,
javax.swing.GroupLayout.PREFERRED_SIZE, 111, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addComponent(cocheCabina4T)

```

```

        .addGap(112, 112, 112)))

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

        .addComponent(jLabel8)

        .addComponent(jLabel15))))

        .addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED_SIZE, 121,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(layout.createSequentialGroup())

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

        .addComponent(jLabel19)

        .addComponent(botonCerrarCabina4Coche,
javax.swing.GroupLayout.PREFERRED_SIZE, 90, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel22)

        .addComponent(jLabel25))

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

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

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

        .addComponent(camionCabina1M)

        .addComponent(EmpleadoCabinaCamiones1M,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

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

        .addComponent(camionCabina3T1,
javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jLabel24, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

        .addComponent(botonCerrarCabina3Camion,
javax.swing.GroupLayout.PREFERRED_SIZE, 73, javax.swing.GroupLayout.PREFERRED_SIZE))))

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

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

```



```

        .addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED_SIZE, 121,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(layout.createSequentialGroup())

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

        .addGroup(layout.createSequentialGroup())

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

        .addComponent(cocheCabina2M,
javax.swing.GroupLayout.PREFERRED_SIZE, 125, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(EmpleadoCabinaCoche2M,
javax.swing.GroupLayout.PREFERRED_SIZE, 125, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel3))

        .addGap(140, 140, 140))

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

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

        .addComponent(jLabel26)

        .addComponent(jLabel20)

        .addComponent(jLabel21)

        .addComponent(cocheCabina5T1))

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

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

        .addComponent(jLabel23)

        .addComponent(EmpleadoCabinaCamiones2M,
javax.swing.GroupLayout.PREFERRED_SIZE, 135, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(camionCabina4T,
javax.swing.GroupLayout.PREFERRED_SIZE, 132, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(botonCerrarCabina4Camion,
javax.swing.GroupLayout.PREFERRED_SIZE, 83, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(5, 5, 5)))

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

```

```

        .addComponent(jLabel16)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(jLabel9)
        .addComponent(jLabel10))))
    .addComponent(botonCerrarCabina5Coche,
        javax.swing.GroupLayout.PREFERRED_SIZE, 90, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(cocheCabina3M)
        .addComponent(EmpleadoCabinaCoches3M)
        .addGroup(layout.createSequentialGroup()

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

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(jLabel13, javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(cocheCabina6T))
        .addComponent(botonCerrarCabina6Coche,
            javax.swing.GroupLayout.PREFERRED_SIZE, 90, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addGroup(layout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE))))
        .addGap(109, 109, 109))
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addGroup(layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(camionCabina2M, javax.swing.GroupLayout.PREFERRED_SIZE,
                        133, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
                        layout.createSequentialGroup()
                            .addGap(170, 170, 170)
                            .addComponent(jLabel17)

```

```

        .addGap(191, 191, 191)

        .addComponent(jLabel18)))

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

layout.setVerticalGroup(

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

    .addGroup(layout.createSequentialGroup())

        .addGap(24, 24, 24)

        .addComponent(jLabel1)

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

        .addComponent(coleEntradaPeaje, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(39, 39, 39)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel2)

            .addComponent(jLabel3)

            .addComponent(jLabel4))

        .addGap(23, 23, 23)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel5)

            .addComponent(cocheCabina1M1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(jLabel7)

            .addComponent(cocheCabina2M, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(jLabel9)

            .addComponent(cocheCabina3M, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

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

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel6)

```

```

        .addComponent(EmpleadoCabinaCoche1M,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel8)

        .addComponent(EmpleadoCabinaCoche2M,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel10)

        .addComponent(EmpleadoCabinaCoche3M,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(38, 38, 38)

        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

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

        .addGroup(layout.createSequentialGroup()

        .addGap(18, 18, 18)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel11)

        .addComponent(jLabel12)))

        .addGroup(layout.createSequentialGroup()

        .addGap(27, 27, 27)

        .addComponent(jLabel13)))

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

        .addGroup(layout.createSequentialGroup()

        .addGap(12, 12, 12)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel14)

        .addComponent(jLabel16)

        .addComponent(cocheCabina4T, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(cocheCabina6T, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))

```

```

        .addGroup(layout.createSequentialGroup())
        .addGap(18, 18, 18)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel15)
        .addComponent(cocheCabina5T1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(18, 18, 18)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(botonCerrarCabina5Coche)
        .addComponent(botonCerrarCabina6Coche)
        .addComponent(botonCerrarCabina4Coche))
        .addGap(39, 39, 39)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel17)
        .addComponent(jLabel18))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel19)
        .addComponent(jLabel21)
        .addComponent(camionCabina1M, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(camionCabina2M, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel20)
        .addComponent(jLabel22)
        .addComponent(EmpleadoCabinaCamiones1M,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(EmpleadoCabinaCamiones2M,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

```

```

        .addGap(26, 26, 26)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel23)
            .addComponent(jLabel24))

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

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel25)
            .addComponent(jLabel26)

            .addComponent(camionCabina4T, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(camionCabina3T1, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

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

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(botonCerrarCabina4Camion)
            .addComponent(botonCerrarCabina3Camion))

        .addContainerGap(48, Short.MAX_VALUE)

    );

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

```

```

private void cocheCabina1M1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void EmpleadoCabinaCoches1MActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void botonCerrarCabina5CocheActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

try{
    InterfazPeaje obj = (InterfazPeaje) Naming.lookup("//localhost/ObjetoCabinaPeaje");
    if(!isBotonCierraCabinaCoches5()){
        obj.cierraCabinaCoche5();
        getBotonCerrarCabina5Coche().setText("Abrir");
        setBotonCierraCabinaCoches5(true);
    }
    else{
        obj.abreCabinaCoche5();

        getBotonCerrarCabina5Coche().setText("Cerrar");
        //Cambiamos el booleano
        setBotonCierraCabinaCoches5(false);
    }
}catch(Exception e){
    e.printStackTrace();
}
}

```

```

private void botonCerrarCabina4CocheActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        InterfazPeaje obj = (InterfazPeaje) Naming.lookup("//localhost/ObjetoCabinaPeaje");
        if(!isBotonCierraCabinaCoches4()){
            obj.cierraCabinaCoche4();
            getBotonCerrarCabina4Coche().setText("Abrir");
            setBotonCierraCabinaCoches4(true);
        }
        else{
            obj.abreCabinaCoche4();

```

```

        getBotonCerrarCabina4Coche().setText("Cerrar");

        //Cambiamos el booleano

        setBotonCierraCabinaCoches4(false);

    }

} catch (Exception e) {

    e.printStackTrace();

}

}

public void actualizarPrograma(){

    try{

        InterfazPeaje obj = (InterfazPeaje) Naming.lookup("//localhost/ObjetoCabinaPeaje");

        getColaEntradaPeaje().setText(obj.devuelveContenidoJTextFields("colaEntradaPeaje"));

        getCocheCabina1M1().setText(obj.devuelveContenidoJTextFields("cocheCabina1M1"));

        getCocheCabina2M().setText(obj.devuelveContenidoJTextFields("cocheCabina2M"));

        getCocheCabina3M().setText(obj.devuelveContenidoJTextFields("cocheCabina3M"));

        getCocheCabina4T().setText(obj.devuelveContenidoJTextFields("cocheCabina4T"));

        getCocheCabina5T1().setText(obj.devuelveContenidoJTextFields("cocheCabina5T1"));

        getCocheCabina6T().setText(obj.devuelveContenidoJTextFields("cocheCabina6T"));

        getCamionCabina1M().setText(obj.devuelveContenidoJTextFields("camionCabina1M"));

        getCamionCabina2M().setText(obj.devuelveContenidoJTextFields("camionCabina2M"));

        getCamionCabina3T1().setText(obj.devuelveContenidoJTextFields("camionCabina3T1"));

        getCamionCabina4T().setText(obj.devuelveContenidoJTextFields("camionCabina4T"));

        getEmpleadoCabinaCoches1M().setText(obj.devuelveContenidoJTextFields("EmpleadoCabinaCoches1M"));

        getEmpleadoCabinaCoches2M().setText(obj.devuelveContenidoJTextFields("EmpleadoCabinaCoches2M"));

        getEmpleadoCabinaCoches3M().setText(obj.devuelveContenidoJTextFields("EmpleadoCabinaCoches3M"));

        getEmpleadoCabinaCamiones1M().setText(obj.devuelveContenidoJTextFields("EmpleadoCabinaCamiones1M"));
    }
}

```



```
getEmpleadoCabinaCamiones2M().setText(obj.devuelveContenidoJTextFields("EmpleadoCabin  
aCamiones2M"));
```

```
    }catch(Exception e){  
        e.printStackTrace();  
    }  
}
```

```
private void camionCabina4TActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

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

```
private void camionCabina3T1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

```
private void EmpleadoCabinaCoches2MActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

```
private void botonCerrarCabina6CocheActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{  
        InterfazPeaje obj = (InterfazPeaje) Naming.lookup("//127.0.0.1/ObjetoCabinaPeaje");  
        if(!isBotonCierraCabinaCoches6()){  
            obj.cierraCabinaCoche6();  
            getBotonCerrarCabina6Coche().setText("Abrir");  
            setBotonCierraCabinaCoches6(true);  
        }  
    }
```

```

else{
    obj.abreCabinaCoche6();

    getBotonCerrarCabina6Coche().setText("Cerrar");
    //Cambiamos el booleano
    setBotonCierraCabinaCoches6(false);
}
}catch(Exception e){
    e.printStackTrace();
}
}

private void botonCerrarCabina4CamionActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        InterfazPeaje obj = (InterfazPeaje) Naming.lookup("//localhost/ObjetoCabinaPeaje");
        if(!isBotonCierraCabinaCamiones4()){
            obj.cierraCabinaCamion4();
            getBotonCerrarCabina4Camion().setText("Abrir");
            setBotonCierraCabinaCamiones4(true);
        }
        else{
            obj.abreCabinaCamion4();

            getBotonCerrarCabina4Camion().setText("Cerrar");
            //Cambiamos el booleano
            setBotonCierraCabinaCamiones4(false);
        }
    }catch(Exception e){
        e.printStackTrace();
    }
}

```

```
}
```

```
private void botonCerrarCabina3CamionActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    try{
```

```
        InterfazPeaje obj = (InterfazPeaje) Naming.lookup("//localhost/ObjetoCabinaPeaje");
```

```
        if(!isBotonCierraCabinaCamiones3()){
```

```
            obj.cierraCabinaCamion3();
```

```
            getBotonCerrarCabina3Camion().setText("Abrir");
```

```
            setBotonCierraCabinaCamiones3(true);
```

```
        }
```

```
    else{
```

```
        obj.abreCabinaCamion3();
```

```
        getBotonCerrarCabina3Camion().setText("Cerrar");
```

```
        //Cambiamos el booleano
```

```
        setBotonCierraCabinaCamiones3(false);
```

```
    }
```

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

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

```
    }
```

```
}
```

```
/**
```

```
 * @param args the command line arguments
```

```
 */
```

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

```
    InterfazDistribuida id = new InterfazDistribuida();
```

```
    /* 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(InterfazDistribuida.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

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

    } catch (IllegalAccessException ex) {

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

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

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

    }
}

//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {

```

```

        id.setVisible(true);
    }

});

while(true){
    id.actualizarPrograma();
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

}

// Variables declaration - do not modify
private javax.swing.JTextField EmpleadoCabinaCamiones1M;
private javax.swing.JTextField EmpleadoCabinaCamiones2M;
private javax.swing.JTextField EmpleadoCabinaCoches1M;
private javax.swing.JTextField EmpleadoCabinaCoches2M;
private javax.swing.JTextField EmpleadoCabinaCoches3M;
private javax.swing.JButton botonCerrarCabina3Camion;
private javax.swing.JButton botonCerrarCabina4Camion;
private javax.swing.JButton botonCerrarCabina4Coche;
private javax.swing.JButton botonCerrarCabina5Coche;
private javax.swing.JButton botonCerrarCabina6Coche;
private javax.swing.JTextField camionCabina1M;
private javax.swing.JTextField camionCabina2M;
private javax.swing.JTextField camionCabina3T1;
private javax.swing.JTextField camionCabina4T;
private javax.swing.JTextField cocheCabina1M1;

```

```
private javax.swing.JTextField cocheCabina2M;
private javax.swing.JTextField cocheCabina3M;
private javax.swing.JTextField cocheCabina4T;
private javax.swing.JTextField cocheCabina5T1;
private javax.swing.JTextField cocheCabina6T;
private javax.swing.JTextField colaEntradaPeaje;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel jLabel19;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel20;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel22;
private javax.swing.JLabel jLabel23;
private javax.swing.JLabel jLabel24;
private javax.swing.JLabel jLabel25;
private javax.swing.JLabel jLabel26;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
```

```
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
// End of variables declaration

public JTextField getEmpleadoCabinaCamiones1M() {
    return EmpleadoCabinaCamiones1M;
}

public JTextField getEmpleadoCabinaCamiones2M() {
    return EmpleadoCabinaCamiones2M;
}

public JTextField getEmpleadoCabinaCoches1M() {
    return EmpleadoCabinaCoches1M;
}

public JTextField getEmpleadoCabinaCoches2M() {
    return EmpleadoCabinaCoches2M;
}

public JTextField getEmpleadoCabinaCoches3M() {
    return EmpleadoCabinaCoches3M;
}

public JButton getBotonCerrarCabina3Camion() {
    return botonCerrarCabina3Camion;
}

public JButton getBotonCerrarCabina4Camion() {
    return botonCerrarCabina4Camion;
}
```

```
public JButton getBotonCerrarCabina4Coche() {  
    return botonCerrarCabina4Coche;  
}
```

```
public JButton getBotonCerrarCabina5Coche() {  
    return botonCerrarCabina5Coche;  
}
```

```
public JButton getBotonCerrarCabina6Coche() {  
    return botonCerrarCabina6Coche;  
}
```

```
public JTextField getCamionCabina2M() {  
    return camionCabina2M;  
}
```

```
public JTextField getCamionCabina3T1() {  
    return camionCabina3T1;  
}
```

```
public JTextField getCamionCabina4T() {  
    return camionCabina4T;  
}
```

```
public JTextField getCocheCabina1M1() {  
    return cocheCabina1M1;  
}
```

```
public JTextField getCocheCabina2M() {  
    return cocheCabina2M;  
}
```



```
}
```

```
public JTextField getCocheCabina3M() {  
    return cocheCabina3M;  
}
```

```
public JTextField getCocheCabina4T() {  
    return cocheCabina4T;  
}
```

```
public JTextField getCocheCabina5T1() {  
    return cocheCabina5T1;  
}
```

```
public JTextField getCocheCabina6T() {  
    return cocheCabina6T;  
}
```

```
public JTextField getColaEntradaPeaje() {  
    return colaEntradaPeaje;  
}
```

```
public JTextField getCamionCabina1M() {  
    return camionCabina1M;  
}
```

```
public boolean isBotonCierraCabinaCoches4() {  
    return botonCierraCabinaCoches4;  
}
```

```
public void setBotonCierraCabinaCoches4(boolean botonCierraCabinaCoches4) {
```

```
    this.botonCierraCabinaCoches4 = botonCierraCabinaCoches4;  
}
```

```
public boolean isBotonCierraCabinaCoches5() {  
    return botonCierraCabinaCoches5;  
}
```

```
public void setBotonCierraCabinaCoches5(boolean botonCierraCabinaCoches5) {  
    this.botonCierraCabinaCoches5 = botonCierraCabinaCoches5;  
}
```

```
public boolean isBotonCierraCabinaCoches6() {  
    return botonCierraCabinaCoches6;  
}
```

```
public void setBotonCierraCabinaCoches6(boolean botonCierraCabinaCoches6) {  
    this.botonCierraCabinaCoches6 = botonCierraCabinaCoches6;  
}
```

```
public boolean isBotonCierraCabinaCamiones3() {  
    return botonCierraCabinaCamiones3;  
}
```

```
public void setBotonCierraCabinaCamiones3(boolean botonCierraCabinaCamiones3) {  
    this.botonCierraCabinaCamiones3 = botonCierraCabinaCamiones3;  
}
```

```
public boolean isBotonCierraCabinaCamiones4() {  
    return botonCierraCabinaCamiones4;  
}
```

```
public void setBotonCierraCabinaCamiones4(boolean botonCierraCabinaCamiones4) {  
    this.botonCierraCabinaCamiones4 = botonCierraCabinaCamiones4;  
}
```

```
}
```

InterfazPeaje.java

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package Distribuida;
```

```
import java.rmi.Remote;
```

```
import java.rmi.RemoteException;
```

```
/**
```

```
*
```

```
* @author david
```

```
*/
```

```
public interface InterfazPeaje extends Remote{
```

```
    void cierraCabinaCoche4() throws RemoteException;
```

```
    void abreCabinaCoche4() throws RemoteException;
```

```
    void cierraCabinaCoche5() throws RemoteException;
```

```
    void abreCabinaCoche5() throws RemoteException;
```

```
    void cierraCabinaCoche6() throws RemoteException;
```

```
    void abreCabinaCoche6() throws RemoteException;
```

```
    void cierraCabinaCamion3() throws RemoteException;
```

```
    void abreCabinaCamion3() throws RemoteException;
```

```
void cierraCabinaCamion4() throws RemoteException;

void abreCabinaCamion4() throws RemoteException;

String devuelveContenidoJTextFields(String cuadroDeTexto) throws RemoteException;

}
```

## Paquete Hilos

### Ambulancia.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Hilos;

import Concurrencia.CabinaPeaje;
import Concurrencia.CabinaPeajeManual;
import Concurrencia.CabinaPeajeTarjeta;
import Concurrencia.Paso;
import java.util.Random;

/**
 *
 * @author david
 */
public class Ambulancia extends Thread {

    private String id;

    private CabinaPeaje cabinapeaje;

    private CabinaPeajeManual cpm = null;

    private CabinaPeajeTarjeta cpt = null;

    private Paso paso;

    public Ambulancia(int id, CabinaPeaje cb, Paso paso){
```

```
this.id = "Ambulancia"+id;

this.cabinapeaje = cb;

this.paso = paso;
}
```

@Override

```
public void run() {

    cabinapeaje.getPaso().mirar();
    cabinapeaje.entradaColaPeajeAmbulancia(this);


    if(this.getCpm()!= null){

        //Comprobamos que ha entrado a una cabina manual
        cabinapeaje.getPaso().mirar();

        getCpm().ambulanciaCabinaManual(this);
    }

    else{

        //Comprobamos que ha entrado a una cabina automática
        cabinapeaje.getPaso().mirar();

        getCpt().entradaCabinaAmbulancia(this);


    }

    //Una vez que hemos salido de la cabina, nos saldremos del peaje
    cabinapeaje.getPaso().mirar();
    cabinapeaje.salidaPeajeAmbulancia(this);

}
```

```
public String getIdAmbulancia() {  
    return id;  
}
```

```
public CabinaPeaje getCabinapeaje() {  
    return cabinapeaje;  
}
```

```
public CabinaPeajeManual getCpm() {  
    return cpm;  
}
```

```
public void setCpm(CabinaPeajeManual cpm) {  
    this.cpm = cpm;  
}
```

```
public CabinaPeajeTarjeta getCpt() {  
    return cpt;  
}
```

```
public void setCpt(CabinaPeajeTarjeta cpt) {  
    this.cpt = cpt;  
}
```

```
}
```

Camion.java

```
/*
```

```
 * To change this license header, choose License Headers in Project Properties.
```

```
 * To change this template file, choose Tools | Templates
```

```
 * and open the template in the editor.
```

```
*/
```

```
package Hilos;
```

```
import Concurrencia.CabinaPeaje;
```

```
import Concurrencia.CabinaPeajeManual;
```

```
import Concurrencia.CabinaPeajeTarjeta;
```

```
import Concurrencia.Paso;
```

```
import java.util.logging.Level;
```

```
import java.util.logging.Logger;
```

```
/**
```

```
 *
```

```
 * @author david
```

```
*/
```

```
public class Camion extends Thread{
```

```
    private String id;
```

```
    private CabinaPeaje cabinapeaje;
```

```
    private CabinaPeajeManual cpm = null;
```

```
    private CabinaPeajeTarjeta cpt = null;
```

```
    private Paso paso;
```

```
    public Camion(int id, CabinaPeaje cb, Paso paso){
```

```
        this.id = "Camion"+id;
```

```

this.cabinapeaje = cb;

this.paso = paso;

}

@Override

public void run(){

    cabinapeaje.getPaso().mirar();

    cabinapeaje.entradaColaPeajeCamion(this);


    if(this.getCpm()!= null){

        //Comprobamos que ha entrado a una cabina manual

        cabinapeaje.getPaso().mirar();

        getCpm().camionCabinaManual(this);

    }

    else{

        //Comprobamos que ha entrado a una cabina automática

        cabinapeaje.getPaso().mirar();

        getCpt().entradaCabinasCamiones(this);


        try {

            cabinapeaje.getPaso().mirar();

            getCpt().pagoCabinasCamiones(this);

        } catch (InterruptedException ex) {

            Logger.getLogger(Camion.class.getName()).log(Level.SEVERE, null, ex);

        }

        cabinapeaje.getPaso().mirar();

        getCpt().salidaCabinasCamiones(this);

```



```
}
```

```
    paso.mirar();  
    cabinapeaje.salidaPeajeCamiones(this);  
}
```

```
public String getIdCamion() {  
    return id;  
}
```

```
public CabinaPeaje getCabinapeaje() {  
    return cabinapeaje;  
}
```

```
public CabinaPeajeTarjeta getCpt() {  
    return cpt;  
}
```

```
public CabinaPeajeManual getCpm() {  
    return cpm;  
}
```

```
public Paso getPaso() {  
    return paso;  
}
```

```
public void setCabinapeaje(CabinaPeaje cabinapeaje) {  
    this.cabinapeaje = cabinapeaje;  
}
```

```
}
```

```
public void setCpt(CabinaPeajeTarjeta cpt) {  
    this.cpt = cpt;  
}
```

```
public void setCpm(CabinaPeajeManual cpm) {  
    this.cpm = cpm;  
}
```

```
}
```

Coche.java

```
/*
```

```
 * To change this license header, choose License Headers in Project Properties.
```

```
 * To change this template file, choose Tools | Templates
```

```
 * and open the template in the editor.
```

```
*/
```

```
package Hilos;
```

```
import Concurrencia.CabinaPeaje;
```

```
import Concurrencia.CabinaPeajeManual;
```

```
import Concurrencia.CabinaPeajeTarjeta;
```

```
import Concurrencia.Paso;
```

```
import java.util.logging.Level;
```

```
import java.util.logging.Logger;
```

```
/**
```

```
 *
```

```
* @author david
```

```
*/
```

```
public class Coche extends Thread{
```

```
    private String id;
```

```
    private CabinaPeaje cabinapeaje;
```

```
    private CabinaPeajeManual cab = null;
```

```
    private CabinaPeajeTarjeta cpt = null;
```

```
    private Paso paso;
```

```
    public Coche(int id, CabinaPeaje cb,Paso paso){
```

```
        this.id = "Coche"+id;
```

```
        this.cabinapeaje = cb;
```

```
        this.paso = paso;
```

```
    }
```

```
@Override
```

```
    public void run(){
```

```
        cabinapeaje.getPaso().mirar();
```

```
        cabinapeaje.entradaColaPeajeCoche(this);
```

```
        if(this.getCab() != null){
```

```
            //Comprobamos que ha entrado a una cabina manual
```

```
            cabinapeaje.getPaso().mirar();
```

```
            getCab().cocheCabinaManual(this);
```

```
        }
```

```
        else{
```

```
            //Comprobamos que ha entrado a una cabina automática
```

```
        cabinapeaje.getPaso().mirar();  
        getCpt().entradaCabinaCoches(this);  
  
    }  
    //Una vez que hemos salido de la cabina, nos saldremos del peaje  
    cabinapeaje.getPaso().mirar();  
    cabinapeaje.salidaPeajeCoche(this);  
  
}
```

```
public CabinaPeajeManual getCab() {  
    return cab;  
}
```

```
public void setCpt(CabinaPeajeTarjeta cpt) {  
    this.cpt = cpt;  
}
```

```
public void setCab(CabinaPeajeManual cab) {  
    this.cab = cab;  
}
```

```
public Paso getPaso() {  
    return paso;  
}
```

```
public void setPaso(Paso paso) {  
    this.paso = paso;  
}
```

```
}
```

```
public CabinaPeajeTarjeta getCpt() {  
    return cpt;  
}
```

```
public String getIdCoche() {  
    return id;  
}
```

```
public CabinaPeaje getCabinapeaje() {  
    return cabinapeaje;  
}
```

```
}
```

```
CrearCamiones.java  
/*
```

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

```
package Hilos;
```

```
import Concurrencia.CabinaPeaje;
```

```
import Concurrencia.CabinaPeajeManual;
```

```
import Concurrencia.Paso;
```

```
import Log.FuncionesLog;
```

```
import java.util.Random;
```

```
/**
```

```
 *
```

```
 * @author david
```

```
 */
```

```
public class CrearCamiones extends Thread{
```

```
    private final int CAPACIDAD = 50;
```

```
    private CabinaPeaje cabina;
```

```
    private FuncionesLog fg;
```

```
    public CrearCamiones(CabinaPeaje cb,FuncionesLog fg){
```

```
        this.cabina = cb;
```

```
        this.fg = fg;
```

```
}
```

```

@Override

public void run(){

    Random r = new Random();

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

        try{

            Camion camion = new Camion(i, cabina,cabina.getPaso());

            cabina.getPaso().mirar();

            camion.start();

            Thread.sleep(1000+r.nextInt(3000));

        }catch(InterruptedException ie){

            ie.toString();

        }

    }

}

```

```

}

```

CrearCoches.java

```

/*

```

```

    * To change this license header, choose License Headers in Project Properties.

```

```

    * To change this template file, choose Tools | Templates

```

```

    * and open the template in the editor.

```

```

*/

```

```

package Hilos;

```

```

import Concurrencia.CabinaPeaje;

```

```

import Concurrencia.CabinaPeajeManual;

```

```

import Concurrencia.Paso;

```

```

import Log.*;

import java.util.Random;

/**
 *
 * @author david
 */
public class CrearCoches extends Thread{

    private final int CAPACIDAD = 50;

    private CabinaPeaje cabina;

    private FuncionesLog fg;


    public CrearCoches(CabinaPeaje cb,FuncionesLog fg){

        this.cabina = cb;

        this.fg = fg;

    }

    @Override
    public void run(){

        Random r = new Random();

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

            try{

                Coche coche = new Coche(i, cabina,cabina.getPaso());

                cabina.getPaso().mirar();

                coche.start();

                Thread.sleep(1000+r.nextInt(3000));

            }catch(InterruptedException ie){

```



}

$$/^{*}$$
 $\ast/$ 

```
import java.util.logging.Logger;
```

\*/

```
private int actividadesSeguidas = 0;
```

```
private Paso paso;

private CabinaPeaje cabinapeaje;
```

```
public Empleado(int id, CabinaPeajeManual cb, Paso paso, CabinaPeaje cabinapeaje){

    this.id = "Empleado"+id;

    this.cb = cb;

    this.paso = paso;

    this.cabinapeaje = cabinapeaje;

}
```

@Override

```
public void run(){

    cabinapeaje.getPaso().mirar();

    cb.entradaEmpleadosCabinas(this);

    while(true){

        try {

            cabinapeaje.getPaso().mirar();

            cb.realizarPagoCoche(this);

            cabinapeaje.getPaso().mirar();

            cb.realizarPagoCamiones(this);

            cabinapeaje.getPaso().mirar();

            cb.pagoAmbulancia(this);

            //cb.pagoAmbulancia(this);

        } catch (InterruptedException ex) {

            Logger.getLogger(Empleado.class.getName()).log(Level.SEVERE, null, ex);

        }

        cabinapeaje.getPaso().mirar();

        if(getActividadesSeguidas() == 6){

            try {

                cabinapeaje.getPaso().mirar();

                cb.salidaEmpleadoDescanso(this);

            }

        }

    }

}
```

```
        } catch (InterruptedException ex) {
            Logger.getLogger(Empleado.class.getName()).log(Level.SEVERE, null, ex);
        }
        setActividadesSeguidas(0);
        cabinapeaje.getPaso().mirar();
        cb.entradaEmpleadosCabinas(this);
    }else{
        cabinapeaje.getPaso().mirar();
        cb.vueleveEmpleado(this);
    }
}

}

public String getIdEmpleado() {
    return id;
}

public CabinaPeajeManual getCb() {
    return cb;
}

public int getActividadesSeguidas() {
    return actividadesSeguidas;
}

public void setActividadesSeguidas(int actividadesSeguidas) {
    this.actividadesSeguidas = actividadesSeguidas;
}
```

```
public Paso getPaso() {  
    return paso;  
}
```

```
}
```

## Paquete Interfaz

Main.java

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package Interfaz;
```

```
import Concurrencia.CabinaPeaje;
```

```
import Concurrencia.CabinaPeajeManual;
```

```
import Concurrencia.Paso;
```

```
import Distribuida.Gestor;
```

```
import Hilos.Ambulancia;
```

```
import Hilos.CrearCamiones;
```

```
import Hilos.CrearCoches;
```

```
import Hilos.Empleado;
```

```
import Log.FuncionesLog;
```

```
import java.rmi.Naming;
```

```
import java.rmi.registry.LocateRegistry;
```

```
import java.rmi.registry.Registry;
```

```
import javax.swing.JButton;
```

```

/**
 *
 * @author david
 */
public class main extends javax.swing.JFrame {

    private boolean botonPulsado = false;

    private CabinaPeaje cb;

    private final FuncionesLog fg;

    private final boolean debug = true;

    private Paso paso;

    public main() {

        initComponents();

        fg = new FuncionesLog(debug);

        paso = new Paso();

        cb = new CabinaPeaje(colaEntradaPeaje, fg, cocheCabina1M1, EmpleadoCabinaCoches1M,
cocheCabina2M, EmpleadoCabinaCoches2M, cocheCabina3M, EmpleadoCabinaCoches3M,
camionCabina1M, EmpleadoCabinaCamiones1M, camionCabina2M,
EmpleadoCabinaCamiones2M, cocheCabina4T, cocheCabina5T1, cocheCabina6T,
camionCabina3T1, camionCabina4T, paso);

        Empleado e1 = new Empleado(1, cb.getCabinaCoches1(), cb.getPaso(), cb);

        Empleado e2 = new Empleado(2, cb.getCabinaCoches2(), cb.getPaso(), cb);

        Empleado e3 = new Empleado(3, cb.getCabinaCoches3(), cb.getPaso(), cb);

        Empleado e4 = new Empleado(4, cb.getCabinaCamiones1(), cb.getPaso(), cb);

        Empleado e5 = new Empleado(5, cb.getCabinaCamiones2(), cb.getPaso(), cb);

        e1.start();

        e2.start();

        e3.start();

        e4.start();

        e5.start();

        CrearCoches co = new CrearCoches(cb, fg);

```

```

co.start();

CrearCamiones cc = new CrearCamiones(cb, fg);

cc.start();


//Inicializamos la conexión del RMI
try{
    Gestor obj = new Gestor(cb);

    Registry registry = LocateRegistry.createRegistry(1099);

    Naming.rebind("//localhost/ObjetoCabinaPeaje",obj);

    System.out.println("El Objeto Peaje ha quedado registrado");
}
catch(Exception e) {
    System.out.println("Error: " + e.getMessage());
    e.printStackTrace();
}

}


/**
 * 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">
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();

```

```
colaEntradaPeaje = new javax.swing.JTextField();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

camionCabina1M = new javax.swing.JTextField();

EmpleadoCabinaCoches1M = new javax.swing.JTextField();

jLabel7 = new javax.swing.JLabel();

jLabel8 = new javax.swing.JLabel();

EmpleadoCabinaCoches2M = new javax.swing.JTextField();

cocheCabina3M = new javax.swing.JTextField();

jLabel9 = new javax.swing.JLabel();

jLabel10 = new javax.swing.JLabel();

EmpleadoCabinaCoches3M = new javax.swing.JTextField();

cocheCabina2M = new javax.swing.JTextField();

jLabel11 = new javax.swing.JLabel();

jLabel12 = new javax.swing.JLabel();

jLabel13 = new javax.swing.JLabel();

jLabel14 = new javax.swing.JLabel();

cocheCabina4T = new javax.swing.JTextField();

jLabel15 = new javax.swing.JLabel();

cocheCabina6T = new javax.swing.JTextField();

jLabel16 = new javax.swing.JLabel();

cocheCabina5T1 = new javax.swing.JTextField();

botonParar = new javax.swing.JButton();

jLabel17 = new javax.swing.JLabel();

jLabel18 = new javax.swing.JLabel();

jLabel19 = new javax.swing.JLabel();

jLabel20 = new javax.swing.JLabel();

jLabel21 = new javax.swing.JLabel();
```

```
jLabel22 = new javax.swing.JLabel();
cocheCabina1M1 = new javax.swing.JTextField();
camionCabina2M = new javax.swing.JTextField();
EmpleadoCabinaCamiones1M = new javax.swing.JTextField();
EmpleadoCabinaCamiones2M = new javax.swing.JTextField();
jLabel23 = new javax.swing.JLabel();
jLabel24 = new javax.swing.JLabel();
jLabel25 = new javax.swing.JLabel();
jLabel26 = new javax.swing.JLabel();
camionCabina4T = new javax.swing.JTextField();
camionCabina3T1 = new javax.swing.JTextField();
botonAmbulancia = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel1.setText("Vehículos esperando entrar al peaje ");

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

jLabel2.setText("Cabina Coches 1 (Manual) ");

jLabel3.setText("Cabina Coches 2 (Manual) ");

jLabel4.setText("Cabina Coches 3 (Manual) ");
```



```
jLabel5.setText("Vehículo:");
```

```
jLabel6.setText("Empleado:");
```

```
camionCabina1M.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        camionCabina1MActionPerformed(evt);  
    }  
});
```

```
jLabel7.setText("Vehículo:");
```

```
jLabel8.setText("Empleado:");
```

```
cocheCabina3M.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        cocheCabina3MActionPerformed(evt);  
    }  
});
```

```
jLabel9.setText("Vehículo:");
```

```
jLabel10.setText("Empleado:");
```

```
jLabel11.setText("Cabina Coches 4 (Tarjeta ");
```

```
jLabel12.setText("Cabina Coches 5 (Tarjeta ");
```

```
jLabel13.setText("Cabina Coches 6 (Tarjeta ");
```

```
jLabel14.setText("Vehículo:");
```

```
jLabel15.setText("Vehículo:");
```

```
cocheCabina6T.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        cocheCabina6TActionPerformed(evt);  
    }  
});
```

```
jLabel16.setText("Vehículo:");
```

```
cocheCabina5T1.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        cocheCabina5T1ActionPerformed(evt);  
    }  
});
```

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

```
jLabel17.setText("Cabina Camiones 2 (Manual) ");
```

```
jLabel18.setText("Cabina Camiones 1 (Manual) ");
```

```
jLabel19.setText("Vehículo:");
```

```
jLabel20.setText("Vehículo:");
```

```
jLabel21.setText("Empleado:");
```

```
jLabel22.setText("Empleado:");
```

```
camionCabina2M.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        camionCabina2MActionPerformed(evt);  
    }  
});
```

```
EmpleadoCabinaCamiones1M.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        EmpleadoCabinaCamiones1MActionPerformed(evt);  
    }  
});
```

```
EmpleadoCabinaCamiones2M.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        EmpleadoCabinaCamiones2MActionPerformed(evt);  
    }  
});
```

```
jLabel23.setText("Cabina Camiones 3 (Tarjeta) ");
```

```
jLabel24.setText("Cabina Camiones 4 (Tarjeta) ");
```

```
jLabel25.setText("Vehículo:");
```

```
jLabel26.setText("Vehículo:");
```



```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 187,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(coleEntradaPeaje, javax.swing.GroupLayout.PREFERRED_SIZE,
1020, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGroup(layout.createSequentialGroup())

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(jLabel5)
    .addComponent(jLabel6)
    .addComponent(jLabel14))

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

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

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup())
    .addComponent(jLabel11)
    .addGap(225, 225, 225))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())
    .addComponent(cocheCabina1M1)
    .addGap(164, 164, 164)
    .addComponent(jLabel7))
    .addComponent(jLabel10,
javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(jLabel15,
javax.swing.GroupLayout.Alignment.TRAILING))

```

```

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

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addComponent(jLabel12)
    .addGroup(layout.createSequentialGroup()
        .addComponent(EmpleadoCabinaCoches2M,
javax.swing.GroupLayout.PREFERRED_SIZE, 136, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(jLabel8))
    .addGroup(layout.createSequentialGroup()
        .addComponent(cocheCabina2M,
javax.swing.GroupLayout.PREFERRED_SIZE, 136, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(169, 169, 169)
        .addComponent(jLabel9))
    .addGroup(layout.createSequentialGroup()
        .addComponent(cocheCabina5T1,
javax.swing.GroupLayout.PREFERRED_SIZE, 136, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(jLabel16))))
.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addComponent(jLabel2)
    .addComponent(EmpleadoCabinaCoches1M)
    .addComponent(cocheCabina4T,
javax.swing.GroupLayout.DEFAULT_SIZE, 137, Short.MAX_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(jLabel3)
    .addGap(213, 213, 213)))

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(16, 16, 16)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel4)
            .addComponent(cocheCabina3M,
javax.swing.GroupLayout.PREFERRED_SIZE, 136, javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGroup(layout.createSequentialGroup()
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel13)
            .addComponent(EmpleadoCabinaCoches3M,
javax.swing.GroupLayout.PREFERRED_SIZE, 136, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(cocheCabina6T,
javax.swing.GroupLayout.PREFERRED_SIZE, 136,
javax.swing.GroupLayout.PREFERRED_SIZE))))))
            .addGroup(layout.createSequentialGroup()
                .addGap(246, 246, 246)
                .addComponent(jLabel18)
                .addGap(228, 228, 228)
                .addComponent(jLabel17))
            .addGroup(layout.createSequentialGroup()
                .addGap(176, 176, 176)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(jLabel19)
            .addComponent(jLabel21)
            .addComponent(jLabel25))
            .addGap(18, 18, 18)

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

```

```

        .addComponent(jLabel23)

        .addGroup(layout.createSequentialGroup()

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

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
            .addComponent(EmpleadoCabinaCamiones1M,
javax.swing.GroupLayout.DEFAULT_SIZE, 137, Short.MAX_VALUE)
            .addComponent(camionCabina1M))
            .addComponent(camionCabina3T1,
javax.swing.GroupLayout.PREFERRED_SIZE, 137, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel20, javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(jLabel22, javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(jLabel26,
javax.swing.GroupLayout.Alignment.TRAILING))))
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel24)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(camionCabina2M, javax.swing.GroupLayout.DEFAULT_SIZE,
137, Short.MAX_VALUE)
            .addComponent(EmpleadoCabinaCamiones2M))
            .addComponent(camionCabina4T, javax.swing.GroupLayout.PREFERRED_SIZE,
137, javax.swing.GroupLayout.PREFERRED_SIZE))))
        .addContainerGap(66, Short.MAX_VALUE)
        .addGroup(layout.createSequentialGroup()

.addGap(416, 416, 416)

.addComponent(botonParar, javax.swing.GroupLayout.PREFERRED_SIZE, 87,
javax.swing.GroupLayout.PREFERRED_SIZE)

```



```

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

        .addComponent(botonAmbulancia, javax.swing.GroupLayout.PREFERRED_SIZE, 97,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(91, 91, 91))

    );

    layout.setVerticalGroup(

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

        .addGroup(layout.createSequentialGroup())

        .addGap(25, 25, 25)

        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

        .addComponent(colaEntradaPeaje, javax.swing.GroupLayout.PREFERRED_SIZE, 24,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(18, 18, 18)

        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(32, 32, 32)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel2)

            .addComponent(jLabel3)

            .addComponent(jLabel4))

        .addGap(18, 18, 18)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel5)

            .addComponent(jLabel7)

            .addComponent(cocheCabina3M, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(jLabel9)

            .addComponent(cocheCabina2M, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(cocheCabina1M1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

```

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

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel6)

    .addComponent(EmpleadoCabinaCoche1M,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addComponent(jLabel8)

    .addComponent(EmpleadoCabinaCoche2M,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addComponent(jLabel10)

    .addComponent(EmpleadoCabinaCoche3M,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(44, 44, 44)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel11)

    .addComponent(jLabel12)

    .addComponent(jLabel13))

.addGap(15, 15, 15)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel14)

    .addComponent(cocheCabina4T, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addComponent(jLabel15)

    .addComponent(cocheCabina6T, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addComponent(jLabel16)

    .addComponent(cocheCabina5T1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 39,
Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel18)

    .addComponent(jLabel17))
```

```
.addGap(8, 8, 8)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel19)

    .addComponent(jLabel20)

    .addComponent(camionCabina1M, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addComponent(camionCabina2M, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel21)

    .addComponent(jLabel22)

    .addComponent(EmpleadoCabinaCamiones1M,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addComponent(EmpleadoCabinaCamiones2M,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(29, 29, 29)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel23)

    .addComponent(jLabel24))

.addGap(14, 14, 14)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel25)

    .addComponent(jLabel26)

    .addComponent(camionCabina4T, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addComponent(camionCabina3T1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(97, 97, 97)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(botonParar)

    .addComponent(botonAmbulancia))
```

```

        .addContainerGap()

    );

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

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

private void cocheCabina6TActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void cocheCabina5T1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void botonPararActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if(!isBotonPulsado()) //Si no se ha pulsado
    {
        setBotonPulsado(true); //Lo cambiamos a pulsado
        getBotonParar().setText("Reanudar"); //Cambiamos el texto
        paso.cerrar();//Cerramos el paso
    }
    else //Si ya se había pulsado
    {
        setBotonPulsado(false); //Lo cambiamos
        getBotonParar().setText("Parar"); //Cambiamos el texto
        paso.abrir();//Abrimos el paso
    }
}

```

```
}
```

```
}
```

```
private void cocheCabina3MActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

```
private void camionCabina1MActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

```
private void camionCabina2MActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

```
private void EmpleadoCabinaCamiones1MActionPerformed(java.awt.event.ActionEvent evt)  
{  
    // TODO add your handling code here:  
}
```

```
private void EmpleadoCabinaCamiones2MActionPerformed(java.awt.event.ActionEvent evt)  
{  
    // TODO add your handling code here:  
}
```

```
private void camionCabina4TActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

```
private void camionCabina3T1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:
```

```
}
```

```
private void botonAmbulanciaActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    Ambulancia a = new Ambulancia(1, cb,paso);
```

```
    a.start();
```

```
}
```

```
/**
```

```
 * @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(main.class.getName()).log(java.util.logging.Level.SEVERE,  
            null, ex);
```

```
        } catch (InstantiationException ex) {
```

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

```
    } catch (IllegalAccessException ex) {
```

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

```
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
```

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

```
    }
```

```
//</editor-fold>
```

```
/* Create and display the form */
```

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

```
    public void run() {
```

```
        new main().setVisible(true);
```

```
    }
```

```
});
```

```
}
```

```
public boolean isBotonPulsado() {
```

```
    return botonPulsado;
```

```
}
```

```
public void setBotonPulsado(boolean botonPulsado) {
```

```
    this.botonPulsado = botonPulsado;
```

```
}
```

```
public JButton getBotonParar() {
```

```
        return botonParar;
    }
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JTextField EmpleadoCabinaCamiones1M;
private javax.swing.JTextField EmpleadoCabinaCamiones2M;
private javax.swing.JTextField EmpleadoCabinaCoches1M;
private javax.swing.JTextField EmpleadoCabinaCoches2M;
private javax.swing.JTextField EmpleadoCabinaCoches3M;
private javax.swing.JButton botonAmbulancia;
private javax.swing.JButton botonParar;
private javax.swing.JTextField camionCabina1M;
private javax.swing.JTextField camionCabina2M;
private javax.swing.JTextField camionCabina3T1;
private javax.swing.JTextField camionCabina4T;
private javax.swing.JTextField cocheCabina1M1;
private javax.swing.JTextField cocheCabina2M;
private javax.swing.JTextField cocheCabina3M;
private javax.swing.JTextField cocheCabina4T;
private javax.swing.JTextField cocheCabina5T1;
private javax.swing.JTextField cocheCabina6T;
private javax.swing.JTextField colaEntradaPeaje;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
```



```
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel jLabel19;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel20;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel22;
private javax.swing.JLabel jLabel23;
private javax.swing.JLabel jLabel24;
private javax.swing.JLabel jLabel25;
private javax.swing.JLabel jLabel26;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
// End of variables declaration
```

```
public Paso getPaso() {
    return paso;
}
```

```
public void setPaso(Paso paso) {
    this.paso = paso;
}
}
```

## Paquete Log

FuncionesLog.java

```
/*  
  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
  
package Log;  
  
  
  
import java.io.BufferedWriter;  
import java.io.File;  
import java.io.FileWriter;  
import java.io.IOException;  
import static java.lang.Thread.sleep;  
import java.text.DateFormat;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import javax.swing.JTextArea;  
import javax.swing.JTextField;  
  
  
/**  
 *  
 * @author david  
 */  
  
public class FuncionesLog {  
  
    private final String ruta;  
    private final File file;  
    private final boolean debug;  
  
  
    /* Creamos esta clase para las funciones comunes que se usan en todas las clases  
    * Y para implementar un sistema de logs de ejecucion
```

```
*/
```

```
public FuncionesLog(boolean debug) {
```

```
    //Variable que determina si va a dejar log o no
```

```
    this.debug = debug;
```

```
    //Declaramos la ruta donde queremos que se genere el historial del log o el nombre del  
    archivo para que se guarde en la carpeta del proyecto
```

```
    ruta = "debug.log";
```

```
    this.file = new File(ruta);
```

```
    if(debug) {
```

```
        //Funcion para crear el log
```

```
        createDebugFile();
```

```
    }
```

```
}
```

```
//Metodo para realizar un sleep segun los tiempos que se le pasen
```

```
public void dormir(int min, int max) {
```

```
    try {
```

```
        sleep( min + (int)( ( max - min ) * Math.random() ) );
```

```
    } catch( InterruptedException ex ) {
```

```
        System.out.println( "ERROR: " + ex );
```

```
    }
```

```
} // Cierre del metodo
```

```
//Metodo para crear el archivo de debug donde se van a ir escribiendo los logs
```

```
public void createDebugFile() {
```

```
    try {
```

```
        String contenido = "PECL2 LABORATORIO - CAMPAMENTO\n\n";
```

```

if( file.exists() ) {
    //Si existe lo borramos y creamos uno nuevo al inicio de la ejecucion del programa
    if( file.delete() ){
        file.createNewFile();
    }
} else {
    file.createNewFile();
}

//Mostramos la ruta absoluta del fichero
System.out.println("Fichero creado en la ruta: " + file.getAbsolutePath());

FileWriter fw = new FileWriter(file);
try (BufferedWriter bw = new BufferedWriter(fw)) {
    bw.write(contenido);
}
} catch(IOException ex) {
    System.out.println("ERROR: " + ex);
}
} // Cierre del método

```

```

//Metodo para escribir en el log en caso de que la variable debug este activada
public synchronized void writeDebugFile(String contenido) {
    if(debug) {
        Date date = new Date();
        DateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
        try {
            FileWriter fw = new FileWriter(file, true);
            try (BufferedWriter bw = new BufferedWriter(fw)) {
                bw.write( dateFormat.format(date) + " => " + contenido );
            }
        }
    }
}

```

```
    } catch(IOException ex) {  
        System.out.println("ERROR: " + ex);  
    }  
}  
} // Cierre  
}
```