



Proyecto
Parking

Integrantes:

Samuel Moreno Vincent

Santiago Rangel Colón

Datos del proyecto:

Integrantes:

Samuel Moreno Vincent (Coordinador)

Santiago Rangel Colón

Sesiones realizadas:

17/05/2013

18/05/2013

24/05/2013

27/05/2013

Todo el trabajo se hizo en conjunto

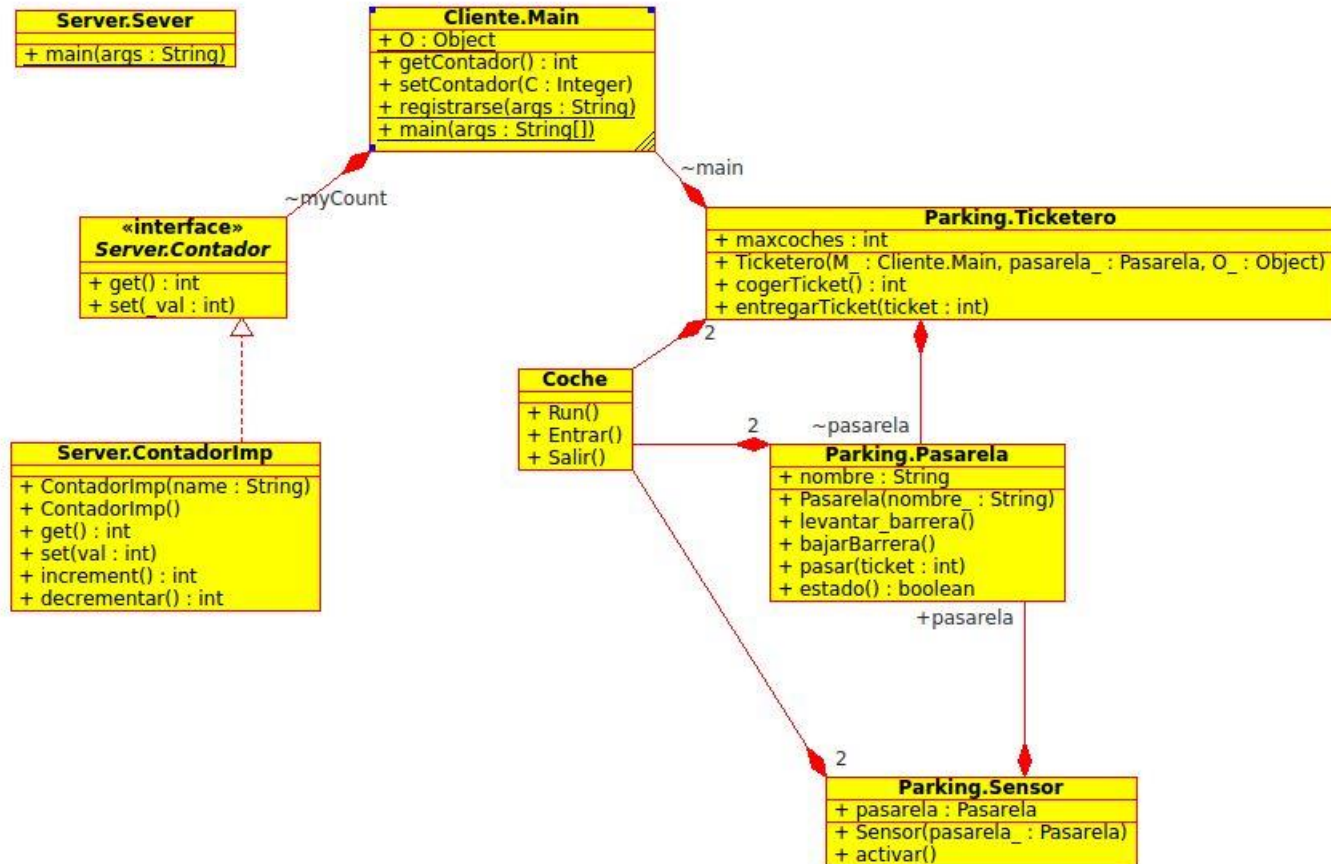
Problemas encontrados:

No funcionaba RMI en nuestro eclipse de Windows

Solución:

Instalar un eclipse mas actualizado en una maquina virtual con ubuntu.

Diagrama de clases



Ejecución del coche:

```
public void entrar() {
    PEntrada.pasar(ticket);
    SEntrada.activar();
}

public void salir() {
    PSalida.pasar(ticket);
    SSalida.activar();
}

public void run() {
    while(true) {
        ticket = TEntrada.cogerTicket();
        entrar();
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        TSalida.entregarTicket(ticket);
        salir();
    }
}
```

Intenta sacar el ticket en el ticketero de entrada. Lo sacará o no dependiendo de las condiciones de entrada.

Condición de entrada:

```
synchronized public int cogerTicket() {
```

```
    synchronized (O) {  
        while (main.getContador() >= maxcoches) {  
            try {  
                O.wait();  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
        int contador = main.getContador();  
        contador++;  
        main.setContador(contador);  
    }
```

Comprobación de si esta lleno

```
    ticket++;  
    System.out.println("Ticket numero " + ticket + " expendido.");  
    synchronized (O) {  
        System.out.println("Coches dentro: " + main.getContador());  
    }
```

Entrega de ticket

```
    while (pasarela.estado()) {  
        try {  
            wait();  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
    }
```

Comprobación de pasarela

```
    pasarela.levantar_barrera();  
    synchronized (O) {  
        O.notifyAll();  
    }
```

Se levanta la pasarela

```
    this.notifyAll();  
    return ticket;  
}
```


Liberación del Wait

Entrega del ticket al coche

Liberación del wait

Ejecución del coche:

```
public void entrar() {
    PEntrada.pasar(ticket);
    SEntrada.activar();
}
public void salir() {
    PSalida.pasar(ticket);
    SSalida.activar();
}
public void run() {
    while(true) {
        ticket = TEntrada.cogerTicket();
        entrar();
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        TSalida.entregarTicket(ticket);
        salir();
    }
}
```



Pasa por la pasarela si la pasarela esta subida
Si no esta subida, espera a que lo este.

```
synchronized public void pasar(int ticket) {
    while(estado != 2) {
        try {
            wait();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
    System.out.println("Un coche " + ticket + " ha p
}
```


Ejecución del coche:

```
public void entrar() {
    PEntrada.pasar(ticket);
    SEntrada.activar();
}
public void salir() {
    PSalida.pasar(ticket);
    SSalida.activar();
}
public void run() {
    while(true) {
        ticket = TEntrada.cogerTicket();
        entrar();
        try {
            Thread.sleep (1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        TSalida.entregarTicket(ticket);
        salir();
    }
}
```

Entrega de ticket en el
ticketero de salida.

Condición de salida:

```
synchronized public void entregarTicket(int ticket) {  
    synchronized (O) {  
        int contador = main.getContador();  
        contador--;  
        main.setContador(contador);  
    }  
    System.out.println("Entregado ticket numero " + ticket);  
    synchronized (O) {  
        System.out.println("Coches dentro: " + main.getContador());  
    }  
    pasarela.levantar_barrera();  
    synchronized (O) {  
        O.notifyAll();  
    }  
    this.notifyAll();  
}
```

Decremento del contador.

Se levanta la pasarela

Liberación de los wait

Ejecución del coche:

```
public void entrar() {  
    PEntrada.pasar(ticket);  
    SEntrada.activar();  
}  
public void salir() {  
    PSalida.pasar(ticket);  
    SSalida.activar();  
}  
public void run() {  
    while(true) {  
        ticket = TEntrada.cogerTicket();  
        entrar();  
        try {  
            Thread.sleep(1000);  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
        TSalida.entregarTicket(ticket);  
        salir();  
    }  
}
```

El coche sale del parking

Ampliación: Tiempo de reacción de la pasarela.

```
synchronized public void levantar_barrera() {  
    while (estado != 0){  
        try {  
            wait();  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
    }  
    estado = 1;  
    try {  
        Thread.sleep (5); //Tiempo que tarda la barrera en subir  
    } catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
    estado = 2;  
    System.out.println("La barrera " + nombre + " se ha subido.");  
    this.notifyAll();  
}
```

→ Tiempo de reacción de subida de la barra.

Estados de la pasarela:

0: Barrera bajada

1: Barrera subiendo/bajando

3: Barrera subida

Ampliación: Tiempo de reacción de la pasarela.


Cambios:

```
synchronized public boolean estado() {  
    while(estado == 1) {  
        try {  
            wait();  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
    }  
    if(estado == 0) {  
        return false;  
    }  
    else {  
        return true;  
    }  
}
```

Al preguntar por el estado de la barrera en al condición de entrega del ticket o de movimiento de la pasarela, el hilo quedara en espera a que termine de moverse si se esta moviendo

Ampliación: Tiempo de reacción de la pasarela.

```
synchronized public void levantar_barrera() {  
    while (estado != 0){  
        try {  
            wait();  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
    }  
    estado = 1;  
    try {  
        Thread.sleep (5);  
    } catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
    estado = 2;  
    System.out.println("La barrera " + nombre + " se ha subido.");  
    this.notifyAll();  
}
```



Al terminar de moverse liberara todos los hilos que estuvieran esperando a que lo haga.