

Ejercicio 1. Sistema de Verificación de ITV para Vehículos

Una estación de ITV desea implementar un sistema automatizado para ayudar a los propietarios de vehículos a verificar el estado de la inspección técnica (ITV) de sus coches. Para ello, se desarrollará una aplicación cliente-servidor que permita a los usuarios enviar los datos de sus vehículos y recibir información sobre su ITV.

Requisitos:

- **Atributos clase Coche:** Matrícula, marca, modelo, añoMatriculacion y fechaUltimaltv (Opcional).
- **Cliente:** Cada cliente cuando se conecta pide al usuario los datos del coche y los envía al servidor. Después muestra la respuesta del servidor por pantalla.
- **Servidor:** Debe estar preparado para aceptar múltiples clientes simultáneamente. Recibe los datos del vehículo, realiza una comprobación basada en el año de matriculación y el año de la última ITV para determinar si el vehículo necesita pasar la ITV y cuando, después envía la respuesta al cliente.
 - Si el coche tiene menos de 4 años no tiene que pasar la ITV.
 - Si el coche tiene entre 4 y 10 años tiene que pasar la ITV cada 2 años.
 - Si el coche tiene más de 10 años tiene que pasar la ITV cada año.

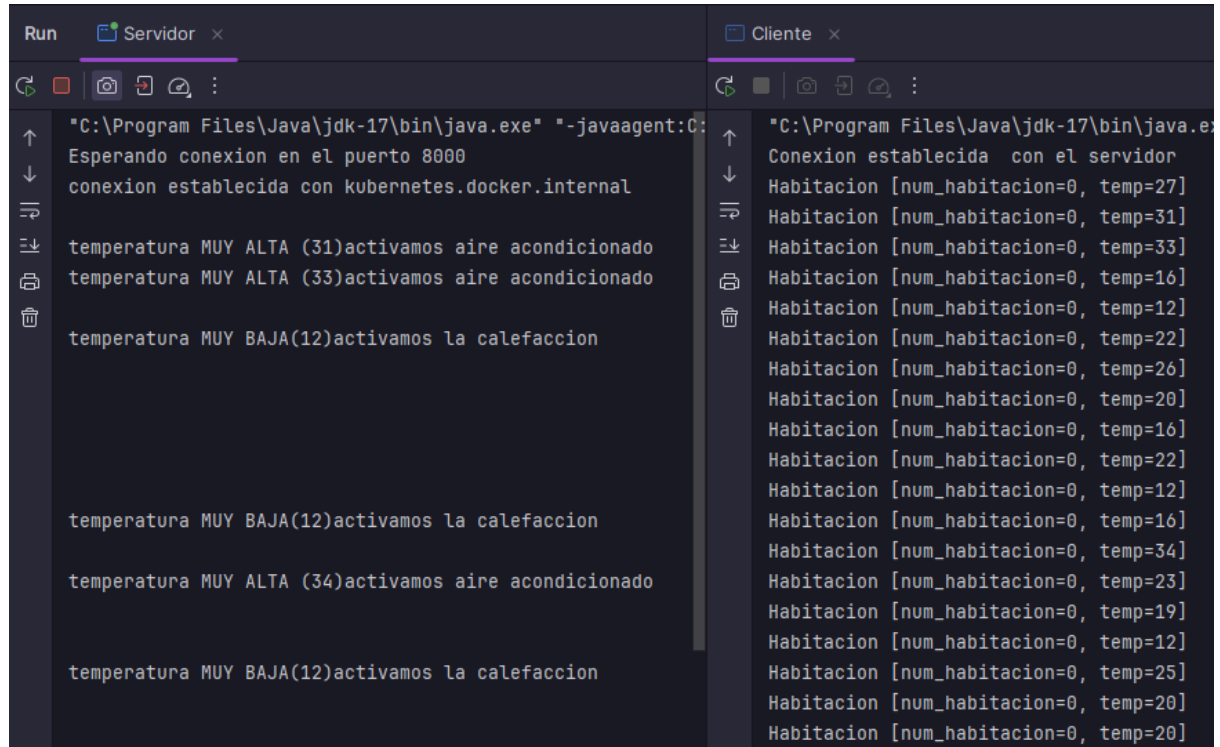
Genera los mensajes de respuesta del servidor como creas conveniente teniendo en cuenta todas las posibilidades.

Ejercicio 2. Sistema de Control de Temperatura en Habitaciones

Un edificio requiere implementar una aplicación para monitorear y controlar la temperatura de sus habitaciones de manera automática. El sistema consiste en un conjunto de clientes, cada uno representando una habitación, que envían información sobre la temperatura actual a un servidor central. El servidor recibe estos datos, evalúa las condiciones de temperatura y, dependiendo de esta, toma acciones para ajustar el ambiente de la habitación.

Requisitos:

- **Atributos clase Habitación:** NumHabitación y temperatura.
- **Cliente:** Cada cliente simula una habitación que genera y envía datos de temperatura al servidor cada *10 segundos*. Los valores de temperatura se generan de forma aleatoria entre 10°C y 35°C.
- **Servidor:** Debe estar preparado para aceptar múltiples clientes simultáneamente. Analiza la temperatura recibida de cada habitación y, según el valor, envía un mensaje de vuelta al cliente con la acción a tomar:
 - Activar el aire acondicionado si la temperatura es mayor a 30°
 - Activar la calefacción si la temperatura es menor a 15°
 - Desactivar sistemas si la temperatura es igual a 24°.



```
Run  Servidor x
"\"C:\Program Files\Java\jdk-17\bin\java.exe\" \"-javaagent:C:\Program Files\Java\jdk-17\bin\javaagent.jar\" -Dkubernetes.docker.internal=172.17.0.1 -jar C:\Users\user\IdeaProjects\temperatura-control\server.jar\"
Esperando conexión en el puerto 8000
conexión establecida con kubernetes.docker.internal

temperatura MUY ALTA (31)activamos aire acondicionado
temperatura MUY ALTA (33)activamos aire acondicionado

temperatura MUY BAJA(12)activamos la calefaccion

temperatura MUY BAJA(12)activamos la calefaccion

temperatura MUY ALTA (34)activamos aire acondicionado

temperatura MUY BAJA(12)activamos la calefaccion

Cliente x
"\"C:\Program Files\Java\jdk-17\bin\java.exe\" \"-javaagent:C:\Program Files\Java\jdk-17\bin\javaagent.jar\" -Dkubernetes.docker.internal=172.17.0.1 -jar C:\Users\user\IdeaProjects\temperatura-control\client.jar\"
Conexión establecida con el servidor
Habitacion [num_habitacion=0, temp=27]
Habitacion [num_habitacion=0, temp=31]
Habitacion [num_habitacion=0, temp=33]
Habitacion [num_habitacion=0, temp=16]
Habitacion [num_habitacion=0, temp=12]
Habitacion [num_habitacion=0, temp=22]
Habitacion [num_habitacion=0, temp=26]
Habitacion [num_habitacion=0, temp=20]
Habitacion [num_habitacion=0, temp=16]
Habitacion [num_habitacion=0, temp=22]
Habitacion [num_habitacion=0, temp=12]
Habitacion [num_habitacion=0, temp=16]
Habitacion [num_habitacion=0, temp=34]
Habitacion [num_habitacion=0, temp=23]
Habitacion [num_habitacion=0, temp=19]
Habitacion [num_habitacion=0, temp=12]
Habitacion [num_habitacion=0, temp=25]
Habitacion [num_habitacion=0, temp=20]
Habitacion [num_habitacion=0, temp=20]
```