

Práctica 4

Ejercicio Evaluable 4: Diseño de una aplicación distribuida



Miguel Jimeno Casas - 100495932 - 100495932@alumnos.uc3m.es

Alberto Menchén Montero - 100495692 - 100495692@alumnos.uc3m.es

Grupo 82 - Ingeniería Informática

ÍNDICE

1 Identificación de cliente-servidor	3
2 Elección del protocolo de transporte	4
3 Características del servidor	4
4 Política de mensajes	4
Tipos de mensajes definidos:	4
ENTRADA:	5
FOTO MATRICULA:	5
SALIDA:	5
ESTADO	5
Respuesta de barrera:	6
5 Intercambio de mensajes (diagrama)	6
6 Direccionamiento	7
IP y Puertos:	7
Máscara de red:	7
Elección de puertos:	7
7 Seguridad	7

Introducción:

Se quiere diseñar una aplicación distribuida para controlar el acceso a un aparcamiento público. En cada una de las N puertas de acceso existe un pequeño computador que controla:

- Una barrera de acceso. La barrera se puede subir o bajar.
- Un botón de entrada, que utilizan los conductores cuando quieren acceder al aparcamiento.
- Una cámara que permite tomar una fotografía de la matrícula del vehículo.
- Una pantalla, que muestra el número de plazas libres que hay en cada momento.
- Un botón de salida, que utilizan los conductores cuando quieren acceder al aparcamiento.

El sistema está compuesto por un computador central que gestiona las plazas libres del aparcamiento. Cada vez que llega un nuevo vehículo al aparcamiento o sale de él, este computador tiene que notificar a los computadores de todas las puertas con el número actualizado de plazas libres, para que las muestren en su pantalla. Cada vez que entra un vehículo en el aparcamiento se toma una fotografía de la matrícula del vehículo y se envía al computador central para su procesamiento. El computador de cada puerta puede abrir la barrera para dejar acceso a un vehículo siempre que haya plazas en el interior.

Solución:

1 Identificación de cliente-servidor

Durante la solución, tendremos que referirnos constantemente a estos conceptos, por ello, se deja claro la definición de cada uno de ellos.

- **Servidor:** Computador Central (CC) encargado de gestionar las plazas disponibles del aparcamiento y de recibir las matrículas capturadas por las cámaras.
- **Cliente:** Cada uno de los N Computadores de Puerta (P_n), ubicados en las entradas/salidas del aparcamiento.

Cada cliente se comunica con el servidor central para:

- Notificar una solicitud de entrada o salida de un vehículo.
- Enviar la imagen de la matrícula capturada.
- Recibir la confirmación para subir o no la barrera.

- Recibir el número actualizado de plazas libres para mostrar en pantalla.

2 Elección del protocolo de transporte

Se utilizarán **sockets TCP** y **conexión por petición** por las siguientes razones:

- Se requiere garantizar la entrega y el orden de los mensajes (ej. control de plazas libres).
- Es importante que no se pierda ninguna matrícula o instrucción de barrera.
- Los intercambios son controlados y puntuales (no es streaming, donde usaríamos UDP).

3 Características del servidor

Concurrencia: El servidor será **concurrente**. Por cada cliente que se conecta (es decir, cada computador de puerta), el servidor gestionará un hilo (o proceso) que atienda su solicitud sin bloquear al resto. Para evitar errores en el recuento de plazas libres, que es un **recurso compartido**, se emplea un mecanismo de **exclusión mutua** (**mutex**) que protege las operaciones de lectura y escritura sobre esta variable. Así, aunque múltiples puertas notifiquen entradas o salidas al mismo tiempo, el servidor garantiza la consistencia del estado..

Servidor sin estado: El servidor será **sin estado**, ya que no necesita recordar información previa del mismo cliente para responder adecuadamente. Todas las decisiones se toman con la información enviada en cada petición.

Conexión por petición: Se usará una conexión por petición, ya que los intercambios de mensajes son cortos e infrecuentes, y permite ahorrar recursos. Se solicita la conexión por petición al interactuar un cliente con la barrera de entrada al parking.

4 Política de mensajes

Tipos de mensajes definidos:

Tipo	Código	Dirección	Contenido principal
ENTRADA	00	Cliente → Servidor	Solicitud de entrada

FOTO MATRICULA	01	Cliente → Servidor	Imagen de matrícula
SALIDA	02	Cliente → Servidor	Notificación de salida
ESTADO	03	Servidor → Cliente	Número actualizado de plazas
RESPUESTA	04	Servidor → Cliente	Confirmación (OK/NO) para subir barrera

Todos los mensajes finalizarán con el carácter especial \0.

ENTRADA:

Cuando un vehículo solicita entrar al aparcamiento.

1. Pn → CC: {opcode (00), id_puerta, hora, sentido (entrada)}
2. CC → Pn:
 - OK en caso de que la solicitud se realice de manera correcta.
 - NO en caso contrario.

FOTO MATRICULA:

Envío de la imagen capturada de la matrícula.

1. Pn → CC: {opcode (01), id_puerta, hora, imagen_matricula}
2. CC → Pn:
 - OK si la imagen fue recibida y procesada correctamente.
 - NO en caso de error en la imagen o fallo de transmisión

SALIDA:

Cuando un vehículo solicita salir del aparcamiento.

1. Pn → CC: {opcode (02), id_puerta, hora, sentido (salida)}
2. CC → Pn:
 - OK en caso de que se reconozca la matrícula y se permita la salida.
 - NO si la matrícula no se encuentra registrada como dentro del aparcamiento.

ESTADO

Envío del número actualizado de plazas libres.

1. CC → Pn: {opcode (03), id_puerta, hora, plazas_libres}

2. Pn → CC:
 - OK
 - NO (id_puerta incorrecto, hora inválida...)

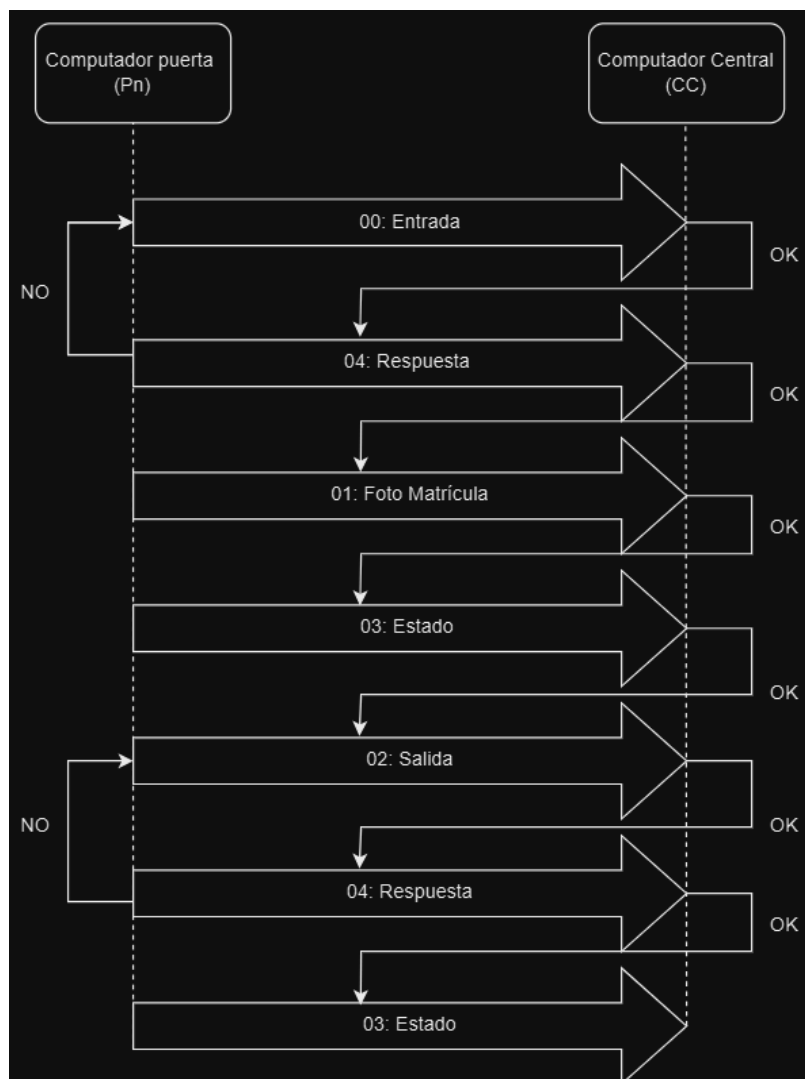
Respuesta de barrera:

Confirmación al cliente de si debe abrir la barrera.

1. CC → Pn: {opcode (04), id_puerta, hora, resultado (OK|NO)}
2. Pn → CC:
 - ACK si la instrucción fue recibida y ejecutada.
 - NO si hubo fallo en la apertura de la barrera.

5 Intercambio de mensajes (diagrama)

Aquí debes añadir un diagrama tipo secuencia con el flujo:



No se han puesto los casos de error en ESTADO y FOTO MATRÍCULA por considerar los casos generales y no entrar en implementaciones de código. En ella se contemplan todos los casos de error correspondientes a cada uno de los códigos de operación.

6 Direccionamiento

IP y Puertos:

- Los computadores de puerta conocen la IP y puerto del servidor central (ej: 192.168.1.100:5000).
- El servidor puede mantener una lista con las IPs de las puertas para enviarles actualizaciones.

Máscara de red:

Se asume una red local con máscara 255.255.255.0.

Elección de puertos:

Se utilizarán puertos altos (por ejemplo, desde el 5000 en adelante) por ser libres y sin privilegios.

7 Seguridad

Aunque no se implementará, se consideran los siguientes aspectos:

- En una aplicación real, el envío de imágenes y datos sensibles (como matrículas) requeriría:
 - **Cifrado TLS/SSL** para proteger la transmisión.
 - **Autenticación** de clientes (puertas) para evitar accesos no autorizados.
 - **Control de acceso** para validar quién puede conectarse.