



Universidad
Carlos III de Madrid



Universidad Carlos III

Sistemas Distribuidos

Curso 2024-25

EJERCICIO 4

06/05/2025

Inés Fuai Guillén Peña 100495752@alumnos.uc3m.es | g82

Claudia Fernández Menéndez 100495919@alumnos.uc3m.es | g82



Introducción

En este ejercicio se plantea el diseño de una aplicación distribuida basada en **sockets TCP/IP** para gestionar el acceso y salida de vehículos en un aparcamiento público. El sistema está compuesto por un ordenador central y varios ordenadores distribuidos, uno en cada puerta de entrada/salida. Cada uno de estos ordenadores controla distintos elementos físicos como barreras, botones, cámaras y pantallas, y debe comunicarse con el ordenador central para obtener información sobre el número de plazas disponibles y enviar datos relevantes sobre los vehículos que acceden.

El objetivo de esta aplicación es garantizar una coordinación eficiente entre las puertas y el servidor central, asegurando que sólo se permita la entrada cuando haya plazas libres y que toda la información del aparcamiento esté actualizada en tiempo real.

Planteamiento del problema

Para la implementación de la aplicación distribuida que gestione el sistema de ordenadores del aparcamiento usaremos los mencionados sockets TCP los cuales permitirán la comunicación entre los distintos dispositivos.

Ordenador central (servidor)

El ordenador central será el encargado de registrar y almacenar toda la actividad de entrada y salida del aparcamiento, así como mantener y distribuir de forma actualizada la información sobre el número de plazas libres y ocupadas. Por tanto, actúa como el servidor principal del sistema y se mantendrá permanentemente escuchando en un puerto específico, a la espera de conexiones entrantes provenientes de las otras puertas de acceso o salida.

Cuando recibe una conexión, el ordenador central crea un hilo específico para tratar esa operación (ya sea una entrada o una salida), lo que permite atender múltiples puertas de forma concurrente. Gracias a esto, el sistema puede escalar sin problemas y evita que una puerta bloquee a otra en su funcionamiento. Además de recibir conexiones, también debe comportarse como cliente en algunos casos, concretamente cuando necesita enviar el número actualizado de plazas libres a todas las puertas. Esto sucede cada vez que se registra una nueva entrada o salida, y se hace mediante sockets hacia cada una de las puertas que estén activas.

Cuando una puerta se activa por primera vez, cada puerta se conecta con el servidor y le envía su número de identificación y el puerto en el que estará escuchando actualizaciones. El servidor recoge además su dirección IP y guarda todos estos datos para poder comunicarse con ella en el futuro. En el momento en el que una puerta se desconecte o se cierre por algún motivo, se eliminará su información del registro para evitar intentos de conexión fallidos.



De este modo, el ordenador central será la clave para la coordinación del sistema. Asegura que todas las puertas tengan la información correcta en tiempo real, facilita una gestión eficiente del aparcamiento y permite una respuesta rápida ante cualquier evento.

Ordenadores de puerta (clientes)

Las puertas del aparcamiento se encargan de gestionar el acceso y la salida de los vehículos, funcionando de forma autónoma pero coordinada con el ordenador central.

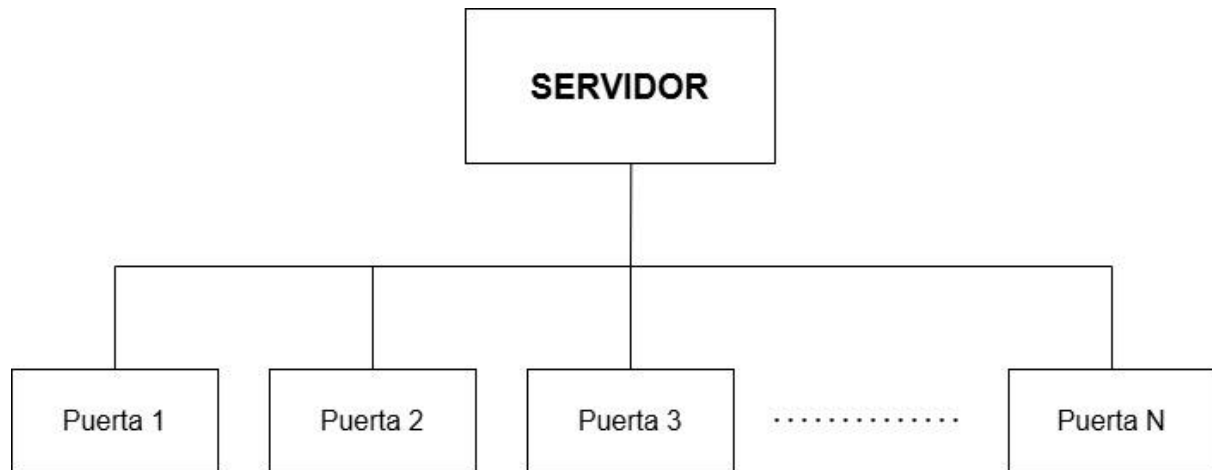
Cada puerta está equipada con varios componentes: una barrera para el acceso, botones de entrada y salida, una cámara que toma la fotografía de la matrícula del vehículo y una pantalla que muestra el número de plazas libres. Cuando una puerta se activa por primera vez, se establece una conexión con el servidor central; en esta primera conexión, la puerta envía su número de identificación, el puerto a través del cual estará escuchando las actualizaciones del servidor y su dirección IP, que será registrada por el servidor. Esto permite al servidor enviar posteriormente las actualizaciones de plazas libres a la puerta en tiempo real.

Si la puerta se desconecta, el servidor eliminará automáticamente su información para evitar futuros errores de comunicación. Cada puerta mantiene un hilo de escucha constante en el puerto asignado. Este hilo estará esperando a recibir mensajes del servidor con actualizaciones sobre el número de plazas libres. Cuando el servidor registre una entrada o salida de vehículo, este enviará un mensaje a todas las puertas activas, las cuales actualizarán la pantalla con el número de plazas disponibles. Este sistema asegura que la información se mantenga sincronizada en todas las puertas en todo momento.

Respecto a la gestión de accesos, cuando un conductor pulsa el botón de entrada, la puerta verifica si hay plazas disponibles. Si las hay, la cámara toma una foto de la matrícula del vehículo y la envía al servidor para su procesamiento. Luego, la puerta espera una respuesta del servidor para saber si se debe levantar la barrera y permitir el acceso. De manera similar, en el caso de la salida, la puerta informa al servidor de la solicitud de salida y espera la confirmación antes de permitir que el vehículo salga.

En resumen, cada puerta del aparcamiento actúa como un cliente dentro del sistema distribuido, siendo responsable tanto de enviar datos al servidor como de recibir actualizaciones sobre el estado del aparcamiento. Este modelo permite que las puertas gestionen el acceso de forma eficiente y coordinada, con un flujo de información en tiempo real entre todos los módulos del sistema.

Esquema



En este esquema podemos ver como las puertas se conectan individualmente con el servidor sin conectarse entre ellas. Esto permite que todas las interacciones se realicen exclusivamente entre las puertas y el servidor, facilitando la gestión centralizada. Se observa como el servidor es el centro de todo y que toda la información que se transmita pasará automáticamente por él, por lo que se convierte en el núcleo.

Protocolo

En el sistema distribuido diseñado para la gestión del aparcamiento, se contemplan cuatro tipos de operaciones principales: activación de puerta, solicitud de entrada, solicitud de salida y actualización del número de plazas disponibles. Las tres primeras son comunicaciones que inician las puertas con el ordenador central mientras que la cuarta operación es iniciada por el servidor, que se encarga de notificar a todas las puertas el nuevo estado de ocupación del aparcamiento. Las conexiones se realizan bajo demanda, usando sockets.

Puertas

Activación de la puerta (cliente)

Cuando se inicia, la puerta establece una conexión con el servidor, creando un socket que le permitirá recibir mensajes de actualización, como por ejemplo cambios en el número de plazas disponibles. Este socket se almacena como una variable global con el fin de poder cerrarlo correctamente cuando la puerta se desactive. A continuación, se lanza un hilo asociado a dicho socket que permanece en escucha continua mientras la puerta está activa. Durante este proceso, se envía una cadena que indica la operación “ACTIVAR”, así como el número de puerto en el que la puerta estará escuchando las actualizaciones y su identificador único. La IP de la puerta se recoge desde el lado del servidor. Posteriormente, se espera la respuesta del servidor para confirmar si la activación se ha realizado correctamente y, finalmente, la conexión entre la puerta y el servidor se cierra.

Solicitud de entrada (cliente)

Cuando un vehículo desea acceder al aparcamiento y pulsa el botón de entrada, la puerta correspondiente establece una conexión con el servidor. En primer lugar, se envía una cadena que indica la operación solicitada, en este caso “ENTRADA”. A continuación, se transmite el identificador de la puerta que está gestionando la solicitud. Simultáneamente, la cámara de la puerta captura la matrícula del vehículo y la envía al servidor en formato de texto. Una vez enviados todos los datos necesarios, la puerta queda a la espera de la respuesta del servidor. Si esta respuesta es positiva, es decir, si hay plazas disponibles, la barrera se eleva permitiendo el acceso del vehículo. Por último, se cierra la conexión entre la puerta y el servidor.

Solicitud de salida (cliente)

Del mismo modo que en la entrada, cuando un vehículo desea abandonar el aparcamiento y se pulsa el botón de salida, la puerta establece una nueva conexión con el servidor. Se envía una cadena que indica la operación “SALIDA” junto con el identificador de la puerta correspondiente. En este caso, también se transmite la matrícula del vehículo que desea salir. Tras el envío de esta información, se espera una respuesta por parte del servidor. Si la salida es válida, se permite el paso del vehículo y la barrera baja automáticamente una vez que ha salido. Finalmente, se cierra la conexión.



Monitorización del estado (servidor)

El hilo que se inicia durante la activación de la puerta permanece en escucha continua mientras la puerta esté operativa. Cada vez que se produce una variación en el número de plazas disponibles, ya sea por una entrada o una salida, el servidor envía un mensaje a la puerta con la nueva cantidad de plazas libres. La puerta recibe esta información y actualiza la pantalla informativa que muestra al usuario. En el momento en que se desactiva la puerta, el socket que se utilizaba para estas comunicaciones se cierra y el hilo correspondiente se interrumpe.

Ordenador principal

El servidor central funciona como el núcleo del sistema y está preparado en todo momento para aceptar nuevas conexiones que le llegan desde las distintas puertas. Cada vez que se establece una conexión, se lanza un hilo independiente que se encarga de gestionar la operación solicitada por esa puerta. Para evitar problemas de concurrencia, como que dos procesos intenten modificar al mismo tiempo el número de plazas o el registro de puertas, se implementa un sistema de exclusión mutua en las secciones críticas del código. Esto asegura que los datos del aparcamiento se mantengan siempre consistentes.

Activación de puerta (servidor)

Cuando una puerta quiere activarse, envía al servidor un mensaje con la operación "ACTIVAR". En ese mensaje también se incluye el puerto en el que la puerta escuchará actualizaciones y su identificador. La dirección IP no hace falta enviarla, ya que el servidor la detecta automáticamente al recibir la conexión. Toda esta información se guarda internamente, ya sea en un fichero o en una estructura en memoria. Una vez procesado todo, el servidor responde con un byte que indica el resultado: 0 si la activación ha ido bien, 1 si la puerta ya estaba activa o 2 si ha ocurrido algún otro problema.

Solicitud de entrada (servidor)

Cuando el servidor recibe una operación de tipo "ENTRADA", lo primero que hace es extraer el identificador de la puerta que la ha enviado y la matrícula del vehículo. Después, comprueba si todavía quedan plazas libres en el aparcamiento. Si es así, resta una unidad al contador de plazas y guarda un registro con la matrícula del coche y la hora a la que ha entrado. Como respuesta, envía un byte al cliente: 0 si se permite la entrada o 1 si el aparcamiento ya está lleno.

Solicitud de salida (servidor)

En el caso de una operación de salida, el proceso es parecido. El servidor recibe el identificador de la puerta y la matrícula del vehículo que quiere salir. Luego, actualiza el contador de plazas sumando una unidad, ya que se libera una plaza, y registra la salida del coche. Para finalizar, envía un byte de confirmación al cliente, siendo 0 si todo ha ido bien.



Envío de actualizaciones (cliente)

Cada vez que entra o sale un coche del aparcamiento, el servidor actualiza el número de plazas libres y manda esa información a todas las puertas que estén activas. Para hacerlo, utiliza las direcciones IP y los puertos que ha guardado anteriormente cuando se activaron las puertas. Si alguna puerta no responde (por ejemplo, si se ha desconectado sin avisar), el servidor la elimina automáticamente de la lista de puertas activas para evitar errores futuros.

Conclusión

En conclusión, para el diseño de esta aplicación distribuida, hemos optado por utilizar **sockets TCP**, una tecnología que permite establecer conexiones fiables y eficientes entre los diferentes componentes del sistema. Al definir de manera clara el protocolo de comunicación y al esquematizar detalladamente el funcionamiento de la aplicación, conseguimos un diseño completo que facilita tanto la implementación como el desarrollo del código.