Universidad Carlos III

2023-24

Ejercicio 4 - Sistemas Distribuidos

Pablo García Rius 100428976

José Antonio Barrientos Andrés 100451290

Curso 2023-24

El primer paso para el desarrollo de la aplicación distribuida será identificar los agentes involucrados y los mensajes que estos itercambiarán.

En cuanto a los agentes, estos se pueden diferencian por:

- Transportista + dispositivo con aplicación embebida.
- Servidor que recibe las peticiones de los dispositivos.

De forma que la aplicación estará dividida en dos partes: un servidor que recibe peticiones, y una serie de clientes (transportistas) que envía peticiones a dicho servidor a través de una aplicación distribuida embebida en su dispositivo móvil.

Por tanto, los mensajes se pueden clasificar en:

- Enviados desde el transportista hacia el servidor:
    - Transportista inicia la jornada
    - Paquete/s entregado (id paquete, id transportista, firma cliente, dni cliente)
    - Transportista termina jornada (lista de paquetes no entregados)
- O enviados desde el servidor hacia el transportista:
    - Lista de paquetes a entregar
    - Mensajes en respuesta a paquete marcado como entregado en el sistema.

El protocolo de mensajes elegido será TCP sobre IP, ya que priorizaremos la fiabilidad en la entrega de los mensajes. Esto es crucial para evitar que los paquetes no se marquen correctamente en el sistema, lo que podría generar fallas de seguridad y responsabilidades que podrían recaer sobre el trabajador cuyo mensaje no llegó al servidor.

La función de cada agente en la aplicación cliente-servidor que vamos a definir será:
El cliente (transportista) es el elemento activo en la comunicación que inicia las transacciones, mientras que el servidor es el componente pasivo que espera a recibir las peticiones para responder en base a los datos que conoce y recibe.

El servidor ofrecerá tres operaciones:

- LOGIN <id trabajador>
- DELIVERED <Nº productos> [<id producto 1>, <id producto 2>, <id producto 3>, ...] <id transportista> <dni cliente> <firma cliente>
- LOGOUT <id transportista > <Nº productos> [<id producto i>, <id producto i+1>, ...]

Cada uno hace la función descrita en las tres funcionalidades necesarias para la aplicación. La comunicación entre dispositivo y servidor utilizando socket TCP será:

1. Servicio Login

Para comenzar la transacción, el cliente solicitará conectarse al socket del servidor.

Una vez conectado, enviará una cadena de caracteres con el código de la operación en formato cadena de caracteres = “LOGIN”.

El servidor procesará la cadena e internamente pasará el socket a un hilo nuevo.

Al ser una operación LOGIN, esperará a recibir el resto de los argumentos, que son:

- El id del trabajador en formato cadena de caracteres.

una vez recibido, realizará una serie de comprobaciones internas para asegurar la consistencia del sistema, como comprobar si dicho id ya tenía una sesión iniciada, o si el id no existiese en el almacén de datos. Si la operación resulta exitosamente, el Id del trabajador quedará como conectado en el sistema de almacenamiento de datos.

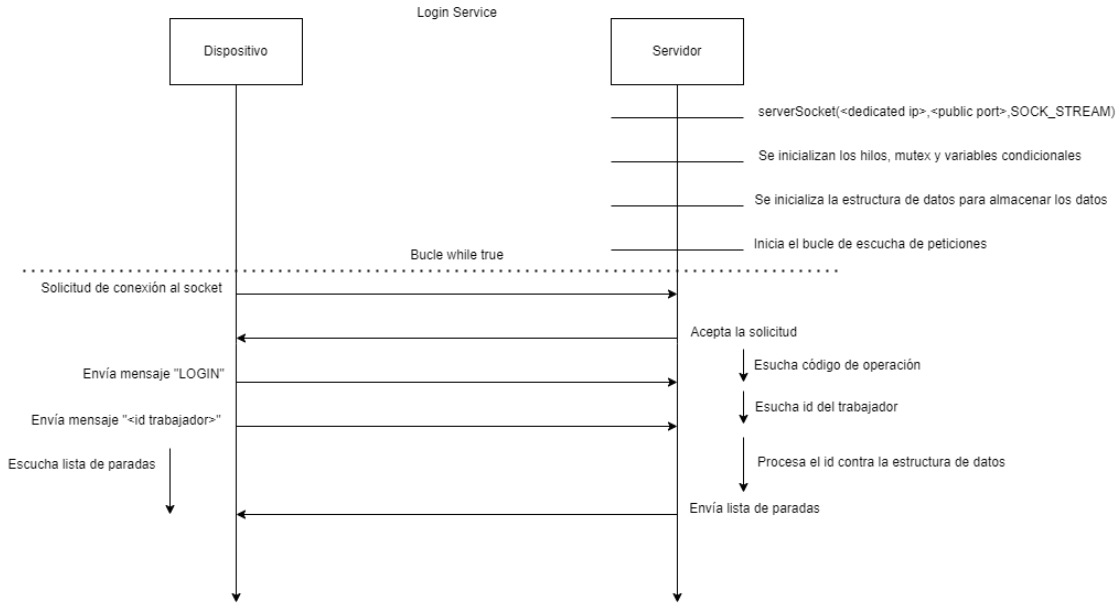
El servidor devolverá un byte codificando el resultado de la operación, este podrá ser:

0. La operación se realizó con éxito
1. El Id de repartidor es incorrecto - Porque no existe o porque ya está conectado → El repartidor deberá revisar si su perfil sufre algún problema.
2. Error de cualquier otro tipo – Comunicaciones, servidor caído, ...

Si el código devuelto es 0, se procedería a utilizar algún tipo de algoritmo de optimización para obtener la mejor combinación de paquetes para el repartidor teniendo en cuenta su localización, las horas que dura su jornada y el destino y peso de los paquetes, la lista resultante se mandaría al dispositivo de la siguiente manera:

El servidor mandaría un mensaje codificando el número de paquetes asignados en formato cadena de caracteres, a continuación, enviará un mensaje conteniendo el Id de cada paquete a entregar por el repartidor.

Una vez terminada la operación y devuelto del código del resultado, se cerraría el socket de ambos lados y se cerraría el hilo encargado de la petición.



El diagrama representa una versión simplificada del paso de mensajes

En este diagrama se ha incluido el inicio de la ejecución del servidor, en esta parte el servidor crea un servidor socket en una dirección IP y puerto seleccionados, se inicializan tanto los hilos que se utilizará para atender a cada cliente como los mutex y variables condicionales para proteger las secciones críticas del código, por último se inicia la estructura de datos seleccionada para almacenar los datos sobre clientes, paquetes y trabajadores, esta podría ser, por ejemplo, árboles binarios de búsqueda, para optimizar la búsqueda de clientes/trabajadores/paquetes por una id única. A continuación, empieza el bucle de escucha de peticiones.

2. Servicio Delivered

En la funcionalidad de entrega de paquete, primero el cliente se conecta al socket del servidor y, una vez establecida la conexión, envía el código de operación como una cadena = “DELIVERED”. Una vez el servidor ha creado un hilo para procesar la petición, pasará a esperar a que el cliente envíe el resto de los argumentos, que son:

- El número de paquetes que se han entregado (debe ser >= 1)
- Un mensaje con cada Id de paquete entregado en formato cadena de caracteres
- La id del transportista en formato cadena de caracteres
- El DNI del cliente en formato cadena de caracteres
- La firma de cliente, esta se deberá transformar a formato cadena de caracteres para que el servidor pueda reconstruirla a formato imagen.

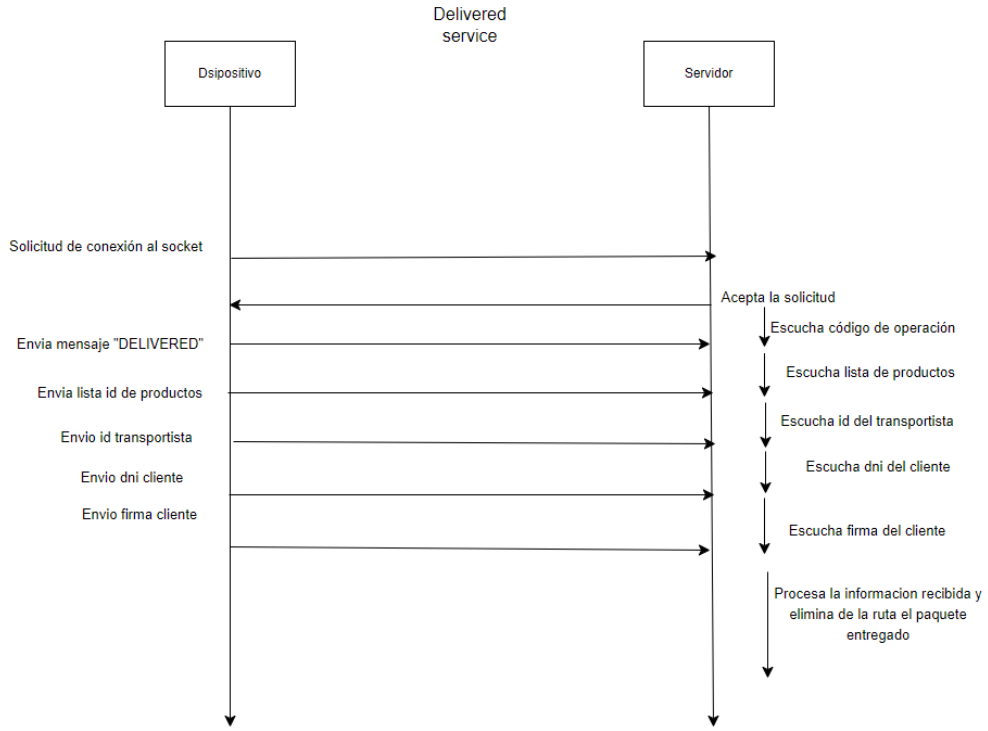
Una vez recibidos todos los argumentos, el servidor realizará una serie de operaciones de comprobación para asegurar la consistencia del sistema, como comprobar que el id del repartidor que ha mandado la operación existe y aparece

como conectado, que los datos del cliente son correctos y por cada paquete que este mande, se comprobará que ha sido entregado al cliente correcto.

El servidor devolverá un byte codificando el resultado de la operación, este podrá ser:

0. La operación se ha realizado con éxito.
1. El Id de repartidor es incorrecto - Porque no existe o porque no está conectado → El repartidor deberá revisar si su perfil sufre algún problema.
2. La identificación del cliente ha fallado - Porque el DNI no tiene un formato correcto o la firma no se ha recibido correctamente → Es posible que el cliente deba revisar alguno de estos datos para poder formalizar la entrega.
3. Paquete incorrecto – Alguno de los paquetes que se están intentando marcar como entregados no corresponde con el cliente al que se está intentando entregar → Se debe recuperar dicho paquete y entregar al cliente correcto.
4. Error de cualquier otro tipo – Comunicaciones, servidor caído, …

Una vez terminada la operación y devuelto del código del resultado, se cerraría el socket de ambos lados y se cerraría el hilo encargado de la petición.



El diagrama representa una versión simplificada del paso de mensajes

3. Servicio Logout

En la funcionalidad Log out, primero el cliente se conecta al socket del servidor y, una vez establecida la conexión, envía el código de operación como una cadena = “LOGOUT”. Una vez el servidor ha creado un hilo para procesar la petición, pasará a esperar a que el cliente envíe el resto de los argumentos, que son:

- Id del trabajador

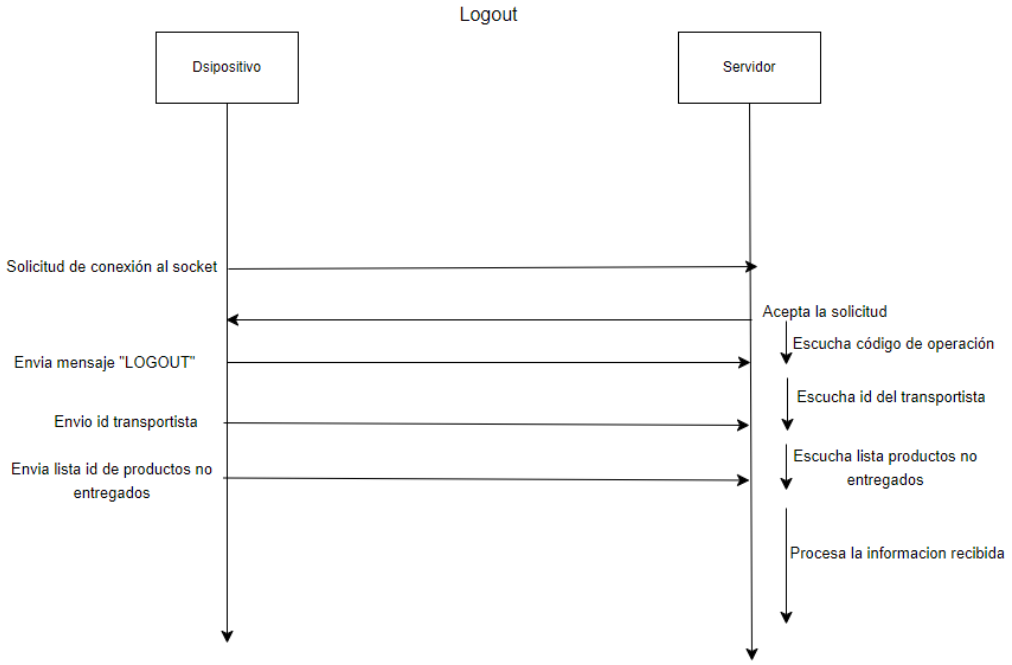
- Número de paquetes no entregados
- Un mensaje con cada Id de paquete no entregado en formato cadena de caracteres

Una vez recibidos todos los argumentos, el servidor realizará una serie de operaciones de comprobación para asegurar la consistencia del sistema, como comprobar que el id del repartidor que ha mandado la operación existe y aparece como conectado.

El servidor devolverá un byte codificando el resultado de la operación, este podrá ser:

0. La operación se ha realizado con éxito.
1. El Id de repartidor es incorrecto - Porque no existe o porque no está conectado → El repartidor deberá revisar si su perfil sufre algún problema.
2. Error de cualquier otro tipo – Comunicaciones, servidor caído, …

Una vez terminada la operación y devuelto del código del resultado, se cerraría el socket de ambos lados y se cerraría el hilo encargado de la petición.



El diagrama representa una versión simplificada del paso de mensajes