



Bachelor's degree in Computer Science and Engineering
Sistemas Distribuidos
2022-2023

Ejercicio Evaluable 4

“Diseño de una aplicación distribuida”

Luis Daniel Casais Mezquida - 100429021

Lucía María Moya Sans - 100383016

Profesor:
Félix García Carballeira

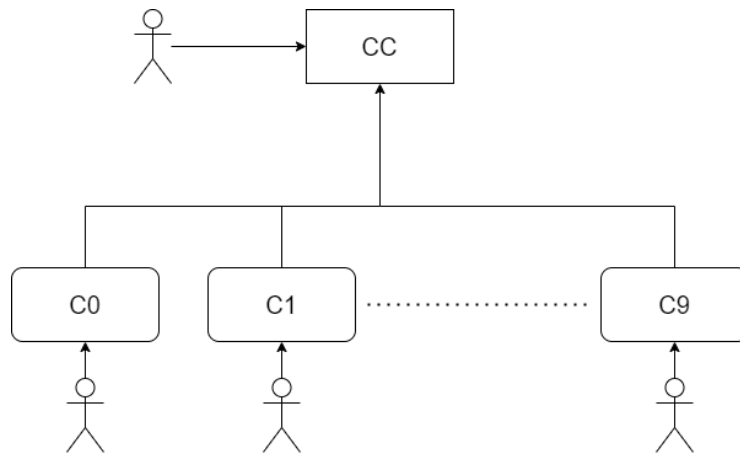
Solución propuesta

1. Identificación del cliente y el servidor

La solución constará de las siguientes partes:

- **Servidor:** Computador Central (CC)
- **Cliente n :** Computador del cajero n (Cn)

Por lo tanto, la arquitectura general de la solución es la siguiente:



2. Operaciones y secuencia de intercambio de mensajes

El sistema contará con cuatro operaciones, con sus consiguientes secuencias de intercambio de mensajes:

- **Apertura:** Cuando el cajero abre la caja.
 1. Cn → CC: Datos de la caja, hora actual
 2. CC → Cn:
 - OK en caso de que todo haya ido bien.
 - KO en caso de error (la caja ya está activa, la hora es incorrecta, etc.).
- **Venta:** Cuando se produce una venta en el cajero.
 1. Cn → CC: Datos de la caja, hora actual, total de la venta, artículos vendidos
 2. CC → Cn:
 - OK en caso de que todo haya ido bien.

- K0 en caso de error (el total no coincide con la suma de los precios de los artículos, la hora es incorrecta, etc.).
- **Cierre:** Cuando el cajero cierra la caja.
 1. Cn \rightarrow CC: Datos de la caja, hora actual, tiempo total, facturación total, artículos vendidos
 2. CC \rightarrow Cn:
 - OK en caso de que todo haya ido bien.
 - K0 en caso de error (la caja ya está cerrada, la hora es incorrecta, el tiempo total no coincide con el registro, etc.).
- **Monitorización:** Consulta de estado de una caja.
 1. CC \rightarrow Cn: Datos de la caja, hora actual
 2. Cn \rightarrow CC:
 - En caso de que todo haya ido bien:
 - (a) OK
 - (b) Datos de la caja, hora actual, tiempo total, estado, facturación total, artículos vendidos.
 - K0 en caso de error (hora incorrecta, identificador incorrecto, etc.).

Nótese que en todos los mensajes el código de operación va incluido.

3. Protocolo

- Se usarán sockets TCP, dado que verifica que los paquetes llegan al destino, y eso es importante en esta aplicación.
- Se usará una conexión por petición, ya que los intercambios de mensajes son cortos e infrecuentes, y permite ahorrar recursos.

4. Formato de los mensajes

Partiendo del esquema general del punto 2, se especifica la estructura de los mensajes:

- **Apertura:** Cuando el cajero abre la caja.
 1. Cn \rightarrow CC: {opcode (open), id, num-caja, hora}
 2. CC \rightarrow Cn:
 - OK en caso de que todo haya ido bien.

- K0 en caso de error (la caja ya está activa, la hora es incorrecta, etc.).
- **Venta:** Cuando se produce una venta en el cajero.
 1. Cn → CC: {opcode (sell), id, num-caja, hora, importe-total, num-artículos, id-artículo-0, precio-artículo-0, [...]}
 2. CC → Cn:
 - OK en caso de que todo haya ido bien.
 - K0 en caso de error (el total no coincide con la suma de los precios de los artículos, la hora es incorrecta, etc.).
- **Cierre:** Cuando el cajero cierra la caja.
 1. Cn → CC: {opcode (close), id, num-caja, hora, importe-total, tiempo-total, num-artículos}
 2. CC → Cn:
 - OK en caso de que todo haya ido bien.
 - K0 en caso de error (la caja ya está cerrada, la hora es incorrecta, el tiempo total no coincide con el registro, etc.).
- **Monitorización:** Consulta de estado de una caja.
 1. CC → Cn: {opcode (status), id, num-caja, hora}
 2. Cn → CC:
 - En caso de que todo haya ido bien:
 - (a) OK
 - (b) {id, num-caja, hora, estado (abierta|cerrada), importe-total, tiempo-total, num-artículos}.
 - K0 en caso de error (hora incorrecta, identificador incorrecto, etc.).

Los diferentes parámetros cuentan con el siguiente formato:

- **opcode:** Array de un caracter numérico que identifica la operación.
Dependiendo de la operación:
 - open: 0
 - sell: 1
 - close: 2
 - status: 3
- **id:** Array de nueve caracteres que identifica al empleado.

- **num-caja:** Array de dos caracteres que identifica la caja, e.g. C2.
- **hora:** Array de once caracteres con la hora en formato HH:MM:SS:ns.
- **importe-total:** Array de diez caracteres con el importe total de una compra o facturación, en formato XXXXXX,XX€
- **num-artículos:** Array de tres caracteres numéricos que indica el número de artículos vendidos.
- **id-artículo-n:** Array de seis caracteres que indica el id de un artículo n .
- **precio-artículo-n:** Array de nueve caracteres con el precio de un artículo n , en formato XXXXX,XX€.
- **tiempo-total:** Array de nueve caracteres que indica el número de minutos que la caja lleva abierta.
- **estado:** Array de un caracter numérico que indica el estado de la caja:
 - abierta: 0
 - cerrada: 1

5. Concurrencia

El Computador Central acepta peticiones de las cajas en un hilo secundario, y por cada petición lanza un hilo para atenderlo. Estos hilos acceden a una base de datos donde guardar y consultar la información, usando mecanismos de concurrencia para garantizar la exclusión y la coherencia en el acceso a los datos.

El hilo principal estará pendiente de atender al usuario del Computador Central para enviar las peticiones requeridas.

Cada Cajero lanza un hilo secundario para atender las peticiones del Computador Central y responderlas, mientras que el hilo principal se encarga de enviar las peticiones requeridas al Computador Central.

Se procurará el acceso a una base de datos donde guardar y consultar la información, usando mecanismos de concurrencia para garantizar la exclusión y la coherencia en el acceso a los datos.

6. Nombrado

Por motivos de seguridad, y asumiendo que en principio el número de cajas se mantendrá, se utilizará un direccionamiento estático.

Las IPs de los equipos se conocen, y pertenecen a la misma red local, por lo que habrá que

configurar manualmente los *hostname* de las máquinas.

Cada cajero sólo conocerá la IP del Computador Central, mientras que el CC conocerá las IPs de todas las cajas.