Universidad de los Andes
Departamento de Ingeniería de Sistemas y Computación
ISIS 1311 – Tecnología e Infraestructura de Cómputo

Caso 3 - 20252

Concurrencia y Sincronización de Procesos

El propósito de este caso es aplicar los conceptos de concurrencia y sincronización de procesos vistos en clase, a través del diseño e implementación de un simulador de un sistema de mensajería distribuido.

1. Objetivos:

- Modelar interacciones concurrentes entre múltiples entidades (clientes emisores, filtros de spam, servidores de entrega) que cooperan y compiten por recursos compartidos.
- Implementar mecanismos de sincronización utilizando exclusivamente las primitivas básicas de Java (synchronized, wait, notify, notifyAll, yield, join, CyclicBarrier).
- Usar diferentes patrones de espera: espera pasiva y espera semi-activa.
- Analizar situaciones típicas de concurrencia, como condiciones de carrera, bloqueos mutuos y pérdidas de mensajes, proponiendo soluciones correctas mediante sincronización.

MUY IMPORTANTE: El entregable de este caso debe ser 100% de su autoría. No está permitido usar ayudas no autorizadas (incluyendo chatbots o tecnologías similares). El incumplimiento de lo anterior resultará en una calificación de cero (0.0) para el caso.

Además, tenga en cuenta que los casos están diseñados para apoyar su proceso de aprendizaje e identificar y resolver sus dudas sobre los temas estudiados antes del parcial.

Finalmente, recuerde que en el parcial habrá una pregunta sobre el caso.

2. El Caso: Centro de Mensajería en la Nube

Queremos simular un centro de mensajería en la nube que procesa correos electrónicos para clientes de una empresa de hosting. El sistema debe coordinar la recepción, filtrado y entrega de los mensajes, garantizando control de spam y disponibilidad de espacio.

A. Arquitectura general del sistema

El centro de mensajería en la nube está organizado como una cadena de procesamiento de correos electrónicos en la que participan tres tipos de actores: clientes emisores, filtro

de spam y servidores de entrega quienes siguen el comportamiento que la Figura 1 ilustra.

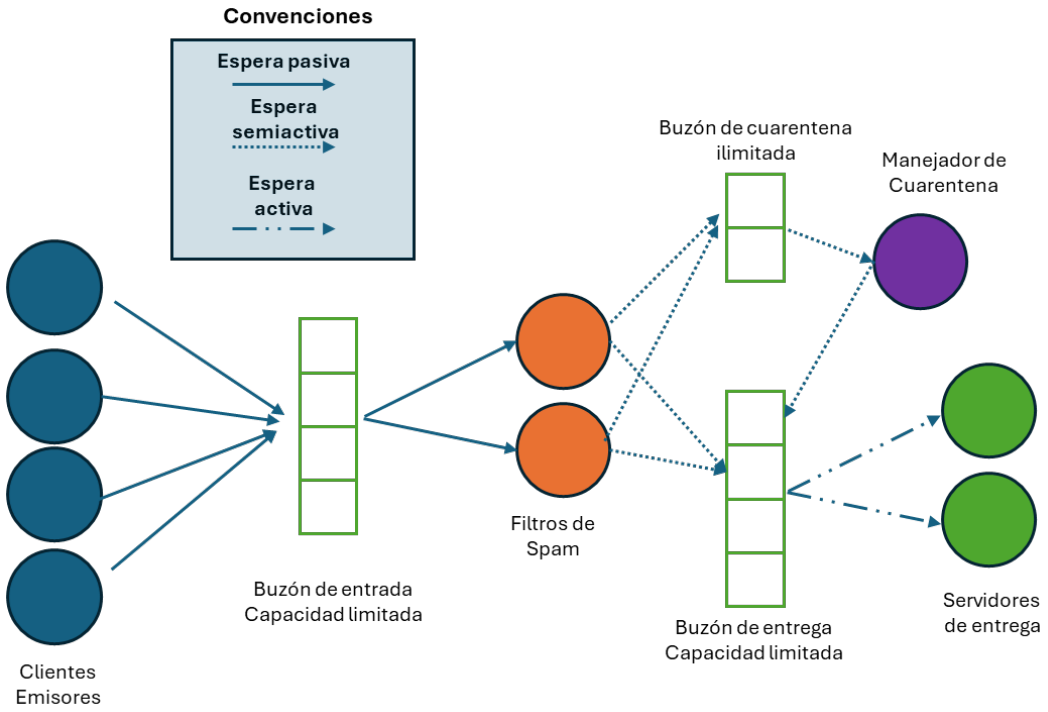

Figura 1. Arquitectura del Sistema de Mensajería

Descripción de los actores:

1. Clientes emisores
    - Son los responsables de generar mensajes de correo.
    - Cada cliente produce un número fijo de correos, delimitado por un mensaje de **inicio** al comenzar y un mensaje de **fin** al terminar.
    - Los clientes depositan sus mensajes en un **buzón de entrada compartido**.
2. Filtros de spam
    - Son los encargados de revisar los mensajes del buzón de entrada.
    - Si el mensaje es spam, lo envían al **buzón de cuarentena**.
    - Si es válido, lo envían al **buzón de entrega**.
    - Además, se encargan de enviar el mensaje de **fin** a los servidores de entrega.
3. Servidores de entrega
    - Son consumidores que leen mensajes del **buzón de entrega**.
    - Cada servidor inicia su operación al encontrar un mensaje de inicio.
    - Un servidor termina cuando encuentra un mensaje de fin.

4. Manejador de cuarentena
    - Es un consumidor que lee mensajes del **buzón de cuarentena**.
    - Inspecciona los mensajes en profundidad y descarta los maliciosos. Los mensajes que pasan la inspección se llevan al buzón de entrega.
    - Termina cuando recibe un mensaje de fin

B. Funcionamiento del Sistema

El centro de mensajería funciona de la siguiente manera:

1. Clientes emisores

- Cada **cliente emisor** (thread productor) se inicia con un número fijo de correos electrónicos que debe generar. Un número aleatorio entre 20 y 100.
- Al iniciar su ejecución, cada cliente envía un **mensaje de inicio**.
- Luego genera sus correos (cada uno con un identificador único y un flag de spam generado aleatoriamente). El identificador puede generarse con el identificador del cliente y un secuencial.
- Una vez termina de producir su cuota de mensajes, envía un **mensaje de fin**.

2. Buzón de entrada

- Los clientes depositan sus correos en un **buzón de entrada** con capacidad limitada.
- Si el buzón está lleno, los clientes deben esperar hasta que haya espacio (espera pasiva). Cuando un cliente logra enviar un correo debe despertar a uno de los filtros para que realice el análisis.

3. Filtros de spam

Los **filtros de spam** (threads analistas) consumen en espera pasiva, uno por uno, mensajes del buzón de entrada:

- Si un correo es marcado como spam, lo envían en espera semiactiva al **buzón de cuarentena** (ilimitado).
- Cuando un filtro de spam deposita un mensaje en el buzón de cuarentena le asocian un tiempo aleatorio (un número entre 10000 y 20000) que representa el tiempo que demorará la cuarentena.
- Si el mensaje es válido, lo envían al buzón de entrega en espera semiactiva.
- Los **mensajes de inicio y de fin** no se consideran spam:
    - Los mensajes de inicio permiten que los filtros de spam conozcan cuantos clientes están registrados en el sistema.

- Los mensajes de fin sirven para que los filtros sepan cuándo han finalizado todos los clientes y que por ende no se producirán más mensajes.
- Cuando todos los mensajes de correo han sido almacenados en el buzón de entrega, uno de los filtros deposita un mensaje de fin en el buzón de entrega. El mensaje de fin solo puede ser enviado al buzón de entrega cuando el buzón de entrada esté completamente vacío, el buzón de cuarentena está vacío y no se van a generar nuevos mensajes.
- Los filtros de spam finalizan cuando se han recibido tantos mensajes de FIN como número de clientes y se ha entregado un mensaje de FIN en el buzón de entrega y un mensaje de fin en el buzón de cuarentena.

4. Manejador de Cuarentena

El manejador de cuarentena consume en espera semiactiva, uno por uno, los mensajes del buzón de cuarentena:

- El manejador revisa todos los mensajes en el buzón de cuarentena y disminuye el contador de tiempo de cada mensaje (resta 1).
- Cuando el contador de tiempo de un mensaje llega a 0, el manejador retira el mensaje del buzón de cuarentena y lo pasa al buzón de entrega (en espera semiactiva).
- Adicionalmente, vamos a simular que algunos mensajes son descartados porque son maliciosos. Para eso, el manejador genera un número aleatorio (entre 1 y 21), cada vez que revisa un mensaje. Si el número es múltiplo de 7, el mensaje se descarta.
- El manejador corre cada segundo.
- Termina cuando recibe un mensaje de FIN.

5. Buzón de Entrega

- Debe entregar un mensaje de fin a los servidores cuando ya no hay ni se producirán nuevos mensajes. Observe que el mensaje de fin que se entrega a los servidores no tiene que ser uno de los mensajes generados por los clientes.
- Después de recibir el mensaje de FIN y constatar que el buzón de entrega una copia del mismo mensaje a todos los servidores para que ellos puedan terminar.

6. Servidores de entrega

- Hay un número fijo **conocido de servidores de entrega (threads** consumidores)
- Cada servidor debe:
    - Procesar los correos válidos que vaya tomando del buzón de entrega. La lectura de mensajes se hace uno a uno del buzón de entrega en espera activa. Para simular la lectura, cuando reciben un mensaje lo procesan

durante un tiempo aleatorio y luego quedan a la espera del siguiente mensaje.

- Terminar su ejecución cuando recibe un **mensaje de fin**.

C. Terminación del sistema

- Los clientes terminan su ejecución después de enviar su mensaje de fin.
- Los filtros terminan cuando el buzón de entrada está vacío, todos los mensajes han sido entregados, no hay mensajes en cuarentena y se ha entregado el mensaje de fin al buzón de entrega.
- Los servidores terminan al recibir un mensaje de fin.
- Todos los buzones deben quedar vacíos.

D. Parámetros configurables

El programa recibe un archivo texto que define los siguientes parámetros:

- Número de clientes emisores.
- Número de mensajes que debe generar cada cliente.
- Número de filtros de spam.
- Número de servidores de entrega.
- Capacidad máxima del buzón de entrada.
- Capacidad del buzón de entrega.

3. Informe

- En un archivo .zip entregue el código fuente del programa, y un informe explicando el diseño (diagrama de clase) y funcionamiento del programa. Incluya una sección de validación (qué pruebas hizo para verificar los resultados).
- En el informe explique, para cada pareja de objetos que interactúan, cómo se realiza la sincronización, así como el funcionamiento global del sistema.
- El nombre del archivo zip debe ser: caso3_login1_login2.zip

4. Condiciones de entrega

- El trabajo se debe realizar en grupos de 2 estudiantes. No debe haber consultas entre grupos. Recuerde que debe conformar un grupo diferente a los casos anteriores y debe trabajar en grupo.
- El grupo responde solidariamente por el contenido de todo el trabajo, y lo elabora conjuntamente (no es trabajo en grupo repartirse puntos o trabajos diferentes).

- Tenga en cuenta las reglas establecidas en el programa del curso sobre la obligatoriedad del trabajo en grupo. Si no cumple con las reglas habrá una penalización de 0,5/5.
- En el parcial se incluirá una pregunta sobre el caso.
- El proyecto debe ser entregado por Bloque Neón por uno solo de los integrantes del grupo. **Al comienzo del informe, deben estar los nombres y carnés de los integrantes del grupo.** Si un integrante no aparece en el documento entregado, el grupo podrá informarlo posteriormente, sin embargo, habrá una penalización: la calificación asignada será distribuida (dividida de forma equitativa) entre los integrantes del grupo.
- **Se debe entregar por Bloque Neón a más tardar el 29 de octubre a las 11:59 p.m.**
- **La fecha de entrega no será movida.**
- **Política de entrega tarde.** Para las entregas tarde, se aplicará la siguiente política: por cada 30 minutos o fracción de retraso, con respecto a la hora de entrega establecida en Bloque Neón, habrá una penalización de 0,5/5.
- En Bloque Neón, **es necesario inscribirse** en alguno de los grupos disponibles para poder acceder a la actividad de entrega del caso. Para ello, debe ingresar a la sección Grupos → Ver grupos disponibles → Caso 3. Es muy importante asegurarse de que ambos integrantes estén registrados en el mismo grupo; de lo contrario, quien no se inscriba no recibirá la información de la calificación.