Informe Laboratorio 2: Comunicación y Sincronización entre Procesos

# Opción B: Memoria Compartida

# Respuestas a las preguntas de análisis y discusión

1. ¿Cómo evitar que dos procesos cliente escriban al mismo tiempo en la misma posición de la cola?  
Para evitar que dos procesos cliente escriban al mismo tiempo en la misma posición de la cola, se implementa un semáforo de exclusión mutua (mutex) que asegura que solo un cliente pueda acceder a la memoria compartida y escribir su pedido en cualquier momento.

2. ¿Qué pasaría si la cocina intenta leer un pedido mientras el cliente aún lo está escribiendo?  
Si la cocina intenta leer un pedido mientras el cliente aún lo está escribiendo, podría leer datos incompletos o incorrectos. Para evitar esto, se utiliza un semáforo que sincroniza el acceso de la cocina a la memoria compartida, asegurando que los pedidos estén completos antes de ser procesados.

3. ¿Qué método de sincronización implementarían para que los procesos cliente esperen su confirmación?  
Se utiliza el semáforo de tipo 'P' (wait) y 'V' (signal) para sincronizar el acceso a la memoria compartida y controlar el flujo entre los procesos. El cliente escribe el pedido y luego espera la confirmación de la cocina antes de continuar.

4. ¿Cómo se comportaría el sistema si la cocina es más lenta que los clientes? ¿Qué riesgos hay?  
Si la cocina es más lenta que los clientes, puede haber un acaparamiento de los pedidos, lo que podría generar bloqueos o pérdida de pedidos. Los semáforos ayudan a regular el acceso, pero si la cocina es significativamente más lenta, se necesitarían optimizaciones para manejar el sobrepaso de pedidos.

5. ¿Qué diferencias prácticas ven entre usar FIFO y memoria compartida para este problema?  
La diferencia principal entre usar FIFO y memoria compartida es la forma de gestión de la comunicación. FIFO es más sencillo de implementar pero podría ser menos eficiente en términos de acceso concurrente. La memoria compartida permite una comunicación más directa y rápida entre los procesos, pero requiere un manejo cuidadoso de la sincronización para evitar problemas de concurrencia.

6. ¿Cómo podrían escalar este sistema para tener más de un proceso cocina trabajando en paralelo sin procesar el mismo pedido dos veces?  
Para escalar este sistema, se podría utilizar una estrategia de división de trabajo entre múltiples procesos cocina, garantizando que cada uno atienda un conjunto específico de pedidos. Además, se podría implementar un semáforo adicional para gestionar la cantidad de procesos cocina activos y evitar que más de uno procese el mismo pedido.