

DECISIONES DE PROGRAMACIÓN DISTRIBUIDA Y DESCRIPCIÓN DEL PROGRAMA.

En este programa, he implementado un sistema de gestión de pedidos para un restaurante utilizando programación concurrente para manejar múltiples clientes y cocineros de manera eficiente. Para coordinar los procesos de los clientes (que hacen pedidos) y los cocineros (que los preparan), he utilizado hilos, ya que permiten ejecutar tareas simultáneas y gestionar múltiples pedidos al mismo tiempo sin bloqueos innecesarios. Esto aborda la dificultad de coordinación entre múltiples productores (clientes) y consumidores (cocineros), permitiendo que los cocineros trabajen mientras los clientes siguen haciendo pedidos.

Para evitar condiciones de carrera y garantizar que no haya conflictos al acceder a recursos compartidos, he usado un semáforo que limita el número de cocineros que pueden acceder a la cola de pedidos simultáneamente. Esto asegura que los cocineros no accedan de manera desordenada a los pedidos y que el sistema funcione de forma segura.

Además, he utilizado una cola de prioridad para los pedidos, lo que permite que los cocineros atiendan primero los pedidos más urgentes o rápidos de preparar, optimizando el tiempo de espera y evitando retrasos innecesarios. Esto aborda la dificultad de optimización del tiempo de espera, ya que los pedidos con menor tiempo de preparación se procesan antes, evitando que se acumulen pedidos de larga espera.

También he implementado un ajuste dinámico del número de cocineros disponibles, permitiendo que se añadan más cocineros si hay una alta demanda de pedidos, sin exceder un límite máximo predefinido. Esto mejora la capacidad de respuesta del sistema, adaptándose a la carga de trabajo en tiempo real y evitando cuellos de botella que puedan generar retrasos. Estas decisiones de programación distribuidas aseguran que el sistema sea escalable y eficiente en la gestión de los recursos, manteniendo un flujo de trabajo fluido y minimizando el tiempo de espera para los clientes.