Cenário Concorrência

O restaurante Sim, que precisa processar muitos pedidos de forma rápida. A ideia é escalar o sistema usando vários cozinheiros simultâneos (threads ou tarefas concorrentes) para preparar os pedidos.

Eles tentaram algo simples: cada cozinheiro retira pedidos de uma fila global e marca como pronto. Mas… algo deu errado.

Problemas observados:

- Alguns pedidos desaparecem
- Cozinheiros preparam o mesmo pedido duas vezes
- A lista de pedidos prontos vem incompleta ou fora de ordem

Tarefa

Cada grupo deve:

1. Simular esse cenário em sua linguagem de programação
2. Reproduzir o problema de concorrência.
3. Discutir por que o erro ocorreu (race condition).
4. Corrigir a implementação **com controle de concorrência, ou equivalente da linguagem.**

Exemplo básico (em Python):

```
pedido = fila_de_pedidos[0]

fila_de_pedidos.remove(pedido)
```

Pode ser interrompido entre essas duas linhas, fazendo com que várias threads leiam o mesmo valor antes de removê-lo.

Etapas da entrega

- **Código com erro reproduzido e comentários explicando o problema.**
- **Código corrigido com o controle de concorrência.**
- **Relatório breve:**
    - **Qual mecanismo foi usado para proteger a região crítica?**
    - **Como você validou que agora funciona corretamente?**

Dicas:

- Um bom teste é imprimir o tamanho da fila no final — se sobrar pedido, houve problema.
- Use `print()`s para rastrear quem pegou qual pedido.
- Pode usar `sleep()` para simular o preparo do pedido e forçar o conflito.