

Pseudocodigo:

```
// Declaración de variables

entero numMesas = 20

arreglo real
pedidos[numMesas]

real totalVentas = 0

entero tiempoInicial = 12 //
Hora de apertura del
restaurante


// Asignar pedidos aleatorios
a cada mesa

para i = 0 hasta numMesas -
1:
    pedidos[i] =
precioAleatorio()


// Inicio del paralelismo
#pragma omp parallel
shared(pedidos, totalVentas)
firstprivate(tiempoInicial)
{
    // -----
    -----
}
```

```
// Sección 1: Meseros
atienden las mesas

// Usamos parallel for con
reduction

// - shared(pedidos): todos
los hilos ven el mismo
arreglo

// -
firstprivate(tiempoInicial):
cada hilo tiene su copia de la
hora inicial

// -
reduction(+:totalVentas):
cada hilo suma al total de
forma segura

// -----
-----

#pragma omp for
reduction(+:totalVentas)
    para i = 0 hasta numMesas
- 1:
        atenderMesa(i,
pedidos[i])
        totalVentas = totalVentas
+ pedidos[i]
```

```
// -----
-----

// Sección 2: Otras tareas
en paralelo

// Usamos sections para
ejecutar diferentes funciones
al mismo tiempo

// -----
-----

#pragma omp sections
{
    #pragma omp section

cocinarPedidos(numMesas)

    #pragma omp section

cobrarPedidos(numMesas)

    #pragma omp section

recogerPlatos(numMesas) //
Tarea extra para mostrar más
paralelismo
}
}
```

```
// Mostrar resultados finales  
imprimir("Total recaudado: ",  
totalVentas)
```