

```

// Declaración de variables

entero numMesas = 20

arreglo pedidos[numMesas]

real totalVentas = 0

entero tiempoInicial = 0


// Asignar pedidos aleatorios a cada mesa

for 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

    #pragma omp for reduction(+:totalVentas)

    for i = 0 hasta numMesas-1:

        atenderMesa(i)

        totalVentas = totalVentas + pedidos[i]


    // Sección 2: Cocinero y cajero en paralelo

    #pragma omp sections

    {

        #pragma omp section

        cocinarPedidos(pedidos)

        #pragma omp section

        cobrarPedidos(pedidos)
    }
}

```

```
}  
}
```

```
// Mostrar resultados
```

```
imprimir("Total recaudado: ", totalVentas)
```