

**Posibles soluciones a los ejercicios del primer recuperatorio del parcial práctico**

**Importante:** las soluciones que se muestran a continuación no son las únicas que se pueden considerar correctas para los ejercicios planteados.

1. Resolver con **SEMÁFOROS** el siguiente problema. En un restorán trabajan  $C$  cocineros y  $M$  mozos. De forma repetida, los cocineros preparan un plato y lo dejan listo en la bandeja de platos terminados, mientras que los mozos toman los platos de esta bandeja para repartirlos entre los comensales. Tanto los cocineros como los mozos trabajan de a un plato por vez. Modele el funcionamiento del restorán considerando que la bandeja de platos listos puede almacenar hasta  $P$  platos. No es necesario modelar a los comensales ni que los procesos terminen.

**NOTA:** es el problema de productores/consumidores con tamaño de buffer limitado visto en la página 14 de la teoría 4.

```
Platos bandeja[P];
int ocupado = 0, libre = 0;
sem vacio = P, lleno = 0;
sem mutexC = 1, mutexM = 1;

process Cocinero [i = 0..C-1] {
    while(true) {
        // Cocina el plato
        Plato plat = HacerPlato();
        // Espera hasta que haya espacio en la bandeja
        P(vacio);
        // Pone el plato en la bandeja
        P(mutexC);
        bandeja[libre] = plat;
        libre = (libre+1) mod P;
        V(mutexC);
        // Marca que hay un plato más en la bandeja
        V(lleno);
    }
}

process Mozo [i = 0..M-1] {
    Plato plat;
    while(true) {
        // Espera hasta que haya algún plato en la bandeja
        P(lleno);
        // Saca un plato de la bandeja
        P(mutexM);
        plat = buf[ocupado];
        ocupado = (ocupado+1) mod P;
        V(mutexM);
        // Marca que hay un lugar libre más en la bandeja
        V(vacio);
        repartir(plat)
    }
}
```

2. Resolver con MONITORES el siguiente problema. En una planta verificadora de vehículos existen 5 *estaciones de verificación*. Hay 75 *vehículos* que van para ser verificados, cada uno conoce el número de estación a la cual debe ir. Cada vehículo se dirige a la estación correspondiente y espera a que lo atiendan. Una vez que le entregan el comprobante de verificación, el vehículo se retira. Considere que en cada estación se atienden a los vehículos de acuerdo con el orden de llegada. **Nota:** maximizar la concurrencia.

```
Process Vehículo [id = 0..74] {
    Comprobante comp;
    int est = .....;
    Admin[est].EsperarAtencion(id, comp);
}

Process Estación [id = 0..4] {
    Comprobante comp;
    int idV;
    while (true) {
        Admin[id].siguiente(idV);
        // Verificar vehículo idV y generar comprobante
        comp = generarComprobante(idV);
        Admin[id].entregarComprobante(comp);
    }
}

Monitor Admin [id = 0..4] {
    cola autos;
    Comprobante comp;
    cond esperaV, esperaE, fin;

    Procedure EsperarAtencion (id: IN int; C: OUT Comprobante) {
        push (autos, id);
        signal (esperaE);
        // Espera a que esté su comprobante
        wait (esperaV);
        C := comp;
        // Avisa que ya tomo el comprobante y que se va.
        signal (fin);
    }

    Procedure siguiente (idV: OUT int) {
        if (empty(autos)) wait (esperaE);
        pop (autos, idV);
    }

    Procedure entregarComprobante (C: IN Comprobante) {
        comp = C;
        // Despierta al vehículo para que tome el comprobante
        signal (esperaV);
        // Espera a que lo haya tomado para atender al siguiente
        wait (fin);
    }
}
```