

**Posibles soluciones a los ejercicios del parcial práctico MC del 10-10-23**

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

1. Resolver con SEMÁFOROS los problemas siguientes:

- a) En una estación de trenes, asisten P personas que deben realizar una carga de su tarjeta SUBE en la terminal disponible. La terminal es utilizada en forma exclusiva por cada persona de acuerdo con el orden de llegada. Implemente una solución utilizando sólo emplee procesos Persona. Nota: la función UsarTerminal() le permite cargar la SUBE en la terminal disponible.

```
Sem s=1;
Sem esperar[P]={ [P]0}
Boolean libre = true;
Queue cola;

Process Personas[id=1..P] {
    P(s)
    if (libre==false) {
        Push(cola,id);
        V(s)
        P(esperar[id]);
    else
        libre=false;
        V(s);
    end
    UsarTerminal();
    P(s)
    if (not empty(cola)) then
        V(esperar[pop(cola)]);
    else
        libre=true;
        V(s);
    }
}
```

- b) Resuelva el mismo problema anterior pero ahora considerando que hay T terminales disponibles. Las personas realizan una única fila y la carga la realizan en la primera terminal que se libera. Recuerde que sólo debe emplear procesos Persona. Nota: la función UsarTerminal(t) le permite cargar la SUBE en la terminal t.

```
sem s=1, sCola=1;
sem esperar[P]={ [P]0 }
int libres = T;
Queue personas = new Queue();
Queue terminales = new Queue(); // comienza con valores de 1 a T ya cargados

Process Personas[id=1..P] {
    P(s)
    if (libres==0) {
        push(personas,id);
        V(s)
        P(esperar[id]);
    else
        libres--;
        V(s);
    end
    P(sCola);
    t = pop(terminales);
    V(sCola);
    UsarTerminal(t);
    P(sCola);
    push(terminales,t);
    V(sCola);
    P(s)
    if (not empty(personas)) then
        V(esperar[pop(personas)]);
    else
        libres++;
    V(s)
}
```

2. Resolver con MONITORES el siguiente problema: En una elección estudiantil, se utiliza una máquina para voto electrónico. Existen N Personas que votan y una Autoridad de Mesa que les da acceso a la máquina de acuerdo con el orden de llegada, aunque ancianos y embarazadas tienen prioridad sobre el resto. La máquina de voto sólo puede ser usada por una persona a la vez. Nota: la función Votar() permite usar la máquina.

```
Monitor Mesa {
    cond colas[N], autoridad, fin;
    Queue(int,int) esperando;

    procedure llegada (int id, int prioridad) {
        push(esperando, (prioridad, id)); // inserta ordenado por prioridad
        signal(authoridad);
        wait(colas[id]);
    }

    procedure salida () {
        signal(fin);
    }

    procedure siguiente () {
        if (empty(esperando)) wait (autoridad);
        int id = pop(esperando);
        signal(colas[id]);
        wait (fin);
    }
}

Process Persona [i: 1..N]{
    int prioridad = obtenerPrioridad();
    // solicitar acceso
    Mesa.llegada(i,prioridad);
    // usar máquina
    Votar();
    // liberar
    Mesa.salida();
}

Process Autoridad_de_Mesa {
    int prioridad = obtenerPrioridad();
    While (true) {
        // dar acceso para votar a la siguiente persona
        Mesa.siguiente();
    }
}
```