

1)

```
monitor sinOrden{
```

```
    int free = TOTAL // pagina total de la memoria.
```

```
    cond mem ;
```

```
    procedure request(int amount){
```

```
        while(free < amount) wait(mem);
```

```
        free -= amount;
```

```
    }
```

```
    procedure releaser(int amount){
```

```
        free += amount;
```

```
        signalAll(mem);
```

```
    }
```

```
}
```

```
monitor SJN {
```

```
    int free = TOTAL // pagina total de la memoria.
```

```
    cond mem ;
```

```
    procedure request(int amount){
```

```
        if((free < amount)) // si
```

```
no tengo memoria
```

```
        || (!empty(mem) && amount >= mirank(mem))) { //si no esta vacia la
cola y si he pedido mas que el que esta de primero en la cola ordenada
```

```

        // si he pedido lo mismo, va el que ha llegado primero
        wait(mem,amount);
    }else{
        free -= amount;
    }

    // si queda alguien mas en la cola.
    if(!empty(mem) && mirank(mem) <= free ){
        free -= mirank(mem);
        signal(mem);
    }

}

```

```

procedure realease(int amount){
    free += amount;
    if(!empty(mem) && mirank(mem) <= free ){
        free -= mirank(mem);
        signal(mem);
    }
}

```

3)

monitor FCFS{

```

    int free = total;

```

```
cond allocate, turn;
```

```
int waiting;
```

```
int allocating; // 0 y 1
```

```
procedure releaset(int amount){
```

```
    free += amount;
```

```
    signal(allocate);
```

```
}
```

```
procedure request(int amount){
```

```
    if(waiting > 0 || allocating > 0){
```

```
        waiting++;
```

```
        wait(turn);
```

```
        waiting--;
```

```
    }
```

```
    while(free < amount){
```

```
        allocating++;
```

```
        wait(allocate);
```

```
        allocating--;
```

```
    }
```

```
    free -= amount;
```

```
    signal(turn);
```

```
}
```

```
}
```

4)

```
monitor FCFS{
```

```
    int free = total;
```

```
    cond mem;
```

```
    int mem[total] = ([total]0);
```

```
    procedure realease(int amount, int ini){
```

```
        free += amount;
```

```
        for(int i = ini; i < amount + ini; i++){
```

```
            map[i] = 0;
```

```
        }
```

```
        signalAll(mem);
```

```
    }
```

```
    procedure request(int amount){
```

```
        while(true){ // bucle hasta que coga hueco
```

```
            int ini = -1;
```

```
            if(free >= amount){
```

```
                for(int i = 0; i < total - amount; i++){
```

```
                    if (hayhueco(i,amount)){
```

```
                        ini = i;
```

```
                        break;
```

```
}
```

```
}
```

```
}
```

```
if(ini == -1) wait(mem);
```

```
else break;
```

```
}
```

```
free -= amount;
```

```
}
```

```
}
```