

Examen.Paso de Mensajes.pdf



Anónimo



Sistemas Concurrentes y Distribuidos



3º Doble Grado en Ingeniería Informática y Matemáticas



**Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada**

Máster

Online en Ciberseguridad

Nº1 en España según El Mundo



**Hasta el 46%
de beca**



Mejor Máster
según el
Ranking de
ELMUNDO

Para ser el mejor hay que aprender
de los mejores.

IMEF

Smart Education

Deloitte

Infórmate



Paso de Mensajes

Asignatura:
Sistemas Concurrentes y Distribuidos

T-FORMACIÓN, Centro de Estudios

Ejercicio 1

Una tienda de licores es frecuentada por varios clientes, que periódicamente la visitan para comprar botellas de licor. El tendero dispone de poco espacio para almacenar botellas (sólo le caben 30 botellas de licor). Además sólo puede atender a los clientes de uno en uno (que sólo pueden comprar una botella cada vez). Para poder contentar a la mayor cantidad de clientes, no permite vender dos botellas seguidas al mismo cliente, pero cuando quedan menos de 5 botellas se ignora la norma y puede comprar cualquiera. Cuando se vacía el almacén, y sólo en ese momento, el propio tendero rellena el almacén y vuelve a impedir comprar dos veces seguidas al mismo cliente (otra vez hasta que queden menos de 5 botellas).

Implementar los procesos Cliente(i) y Tendero usando paso de mensajes síncrono de forma que no haya interbloqueo y se cumplan las características del problema. Suponed que el almacén está lleno inicialmente, que el índice en el vector de procesos es su nombre y que hay N clientes: Cliente(0), Cliente(1), ... Cliente(N-1).

Ejercicio 2

Supongamos que tenemos tres procesos (llamados proc0, proc1 y proc2) de forma que cada uno pase un dato (el valor de una variable local) al siguiente para que el siguiente lo imprima usando las operaciones de paso de mensaje. Esto se ha programado usando el siguiente esquema de mensajes síncrono:

```
proceso proc0
begin
  x := . . . . . ;
  s_send( x, proc1 );
  receive( y, proc2 );
  imprime( y );
end
```

```
proceso proc1
begin
  x := . . . . . ;
  s_send( x, proc2 );
  receive( y, proc0 );
  imprime( y );
end
```

```
proceso proc2
begin
  x := . . . . . ;
  s_send( x, proc0 );
  receive( y, proc1 );
  imprime( y );
end
```

Contesta a las siguientes cuestiones:

- Este programa produce interbloqueo. Describe brevemente a que se debe esto.
- Si el paso de mensajes es asíncrono seguro, ¿se podría producir un interbloqueo?
- Describe brevemente los cambios que harías en los procesos para cumplir la ausencia de interbloqueo manteniendo un paso de mensajes síncrono.

