Repaso practico 1 - Sentencias Atómicas

Ejercicio 2

Se tiene un salon con cuatro puertas por donde entran alumnos a un examen. Cada puerta lleva la cuenta de los que entraron por ella y a su vez se lleva la cuenta del total de personas en el salón.

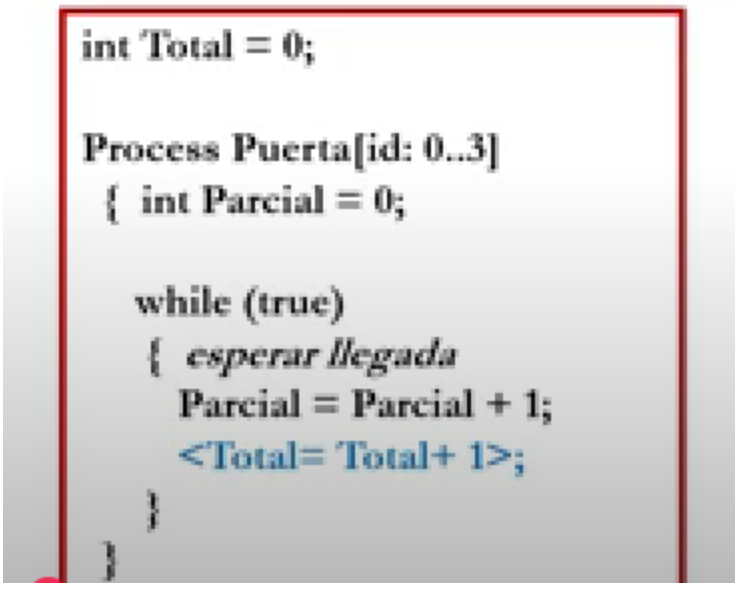
Mi solución :

```
int total = 0

process puerta[id: 1..4 ]

int entraron = 0
while (true){
    esperando que llegue alguien
    entraron++
    <total ++ >
}
```

Solución del profesor :



Ejercicio 3

Hay un docente que les debe tomar examen oral a 30 alumnos (de uno a la vez) de acuerdo al orden dado por el identificador del proceso.

Mi solución :

```
boolean me-fui = true
boolean estoy-libre = true

process profesor{

    for (int i = 0, i<30, i++){
        <estoy-libre = false>
        tomando examen oral
        <await me-fui>
        <estoy-libre = true>
    }

}

process alumno[id: 0..29]{

    if (estoy-libre){
        <me-fui = false>
        dando examen oral
        <me-fui = true>
    }
}
```

Solución del profesor :

```
Process Alumno [id: 0..29]
 { //Espera a que lo llamen
   //Rinde el examen
   //Espera a que termine el examen
 }


Process Docente
 { for i = 0..29
     { //Llama al alumno "i"
       //Toma el examen
       //Avisa a "i" que termino
     }
 }
```

```
int Actual = -1
bool Listo = False;

Process Alumno [id: 0..29]
 { <await (Actual == id)>;
   //Rinde el examen
   <await (Listo); Listo = false>;
 }

Process Docente
 { for i = 0..29
     { Actual = i
       //Toma el examen
       <Listo = true;>
       <await (not Listo)>
     }
 }
```

Es necesario hacerlo con <>

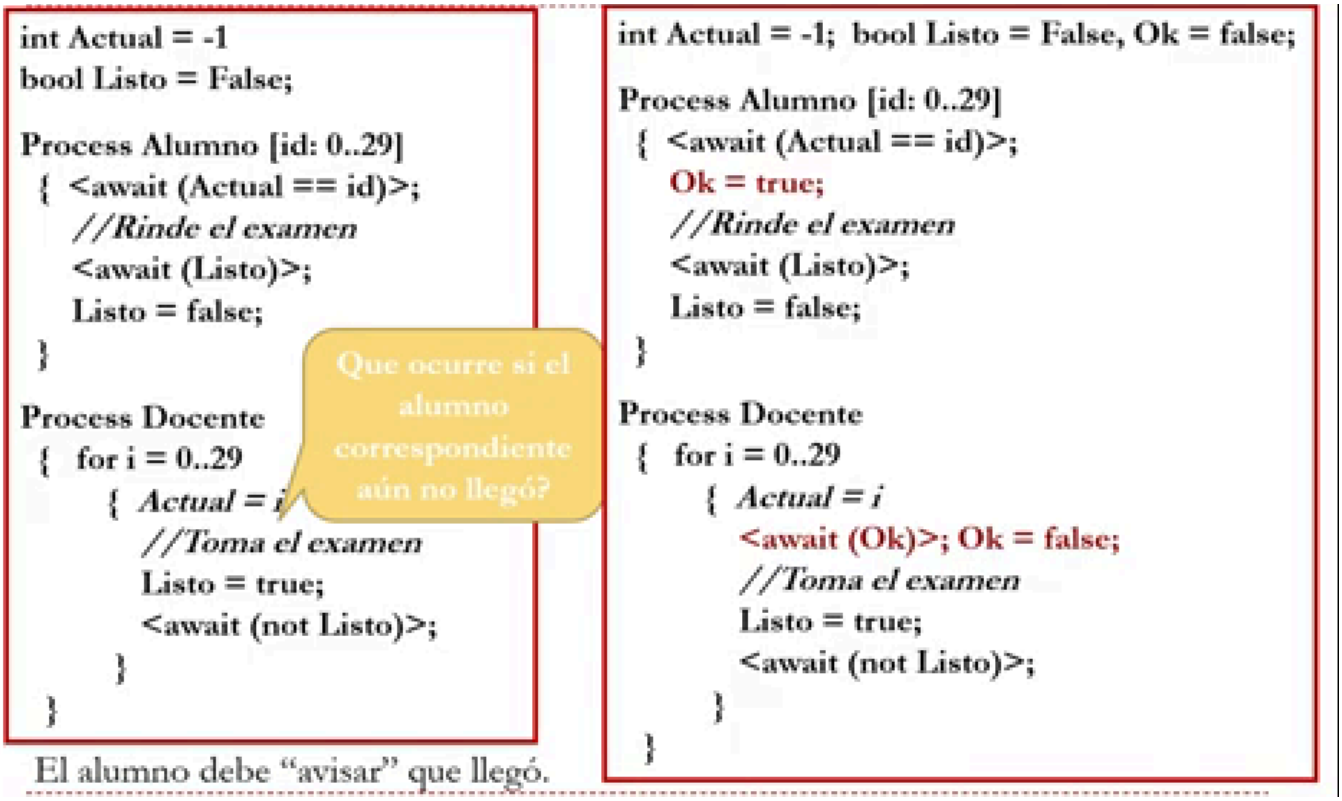
Es necesario hacerlo con <>

No es necesario el <> por las mismas razones que Actual = i

```
int Actual = -1
bool Listo = False;

Process Alumno [id: 0..29]
 { <await (Actual == id)>;
   //Rinde el examen
   <await (Listo)>;
   Listo = false;
 }

Process Docente
 { for i = 0..29
     { Actual = i
       //Toma el examen
       Listo = true;
       <await (not Listo)>;
     }
 }
```



Ejercicio 4

Un cajero automático debe ser usado por N personas de uno a la vez y según el orden de llegada del mismo. En caso de que llegue una persona anciana, la deben dejar ubicarse al principio de la cola.

Mi solución :

```
colaEspecial c;
boolean libre = true

process persona[id:0..N-1]{

    int edad = inicializa la edad de la persona
    c.agregar(c,edad,id)
    <await not(libre) >
    libre = false
    usa el cajero automatico
    <libre = true>
}

-> En este caso el cajero es el recurso, lo que debemos validar y manejar es
el uso de la cola y cuando esta libre el cajero
```

Solución del profesor :

-> Debemos tener en cuanta si el cajero esta libre y si la cola esta o no vacía.
-> La solución esta dada por el orden de llegada pero con prioridades :

```
colaEspecial C;
int Siguiente = -1;

Process Persona [id: 0..N-1]
  { int edad = ….;

    <if (Siguiente = -1) Siguiente = id
      else Agregar(C, edad, id)>;
    <await (Siguiente == id)>;
    //Usa el cajero
    <if (empty(C)) Siguiente = -1
      else Siguiente = Sacar(C)>;
  }
```