process alumnos[i:1..50]{

int g, nota; Texto tarea;

fila.formarFila(i);

alumno[i].obtenerGrupo(g);

tarea:= hacerTarea();

grupo[g].entregar(i,tarea);

alumno[i].esperarNota(nota);

}

process profesor(){

int g, id;

for (int j:=1;j<50; j++){

fila.proximoAlumno(id);

g := numeroGrupo();

alumno[id].entregarNumeroGrupo(g);

}

for (int j:=25; j > 1; j--){

profesor.proximaTarea(tarea,id1,id2);

alumno[id1].calificar(j);

alumno[id2].calificar(j);

}

}

monitor grupo[i:1..25]{

Texto tarea;

int alumno[2]; indice := 1;

bool listo := false;

procedure entregar(int id: in ; texto t in){

if (indice < 3){ alumno[indice] := id;

tarea := tarea + t;

indice++;

}

if (indice == 3){

profesor.entregar(alumno[1],alumno[2],tarea);

}

}

}

monitor profesor(){

texto tarea;

int alumno1,alumno2,grupo; cant:= 0;

queue c;

cond prof;

procedure entregar(int : a1,a2 in; texto t){

c.push(a1,a2,t);

cant++;

signal (prof);

}

procedure proximaTarea(texto : tarea out; int : al1,al2 out){

if (cant==0) wait(prof);

c.pop(tarea,al1,al2);

cant--;

}

}

monitor alumno[i:1..50]{

int ngr, nota; bool listo:= false; hayNota:= false;

cond al, prof, not;

procedure obtenerGrupo(int g out){

if (not listo) wait (al);

g := ngr;

signal (prof);

}

procedre entregarNumeroGrupo(int n in){

ngr := n;

listo := true;

signal (al);

wait (prof);

}

procedure esperarNota(int n out){

if (not hayNota) wait (not);

n:= nota;

}

procedure calificar(int n in){

nota := n;

hayNota := true;

signal(not);

}

}

monitor fila(){

queue c; int cant:= 0;

cond fila, profe;

procedure formaFila(int id in){

if (cant < 50){

c.push(id);

cant++;

wait(fila);

}else

signal(profe);

}

procedure proximoAlumno(int id out){

if(cant > 0){

id:= c.pop();

cant--;

signal (fila);

}else

wait(profe);

}

}

}