

# Programación Concurrente

Proyecto 12: Matriculas Universitarias

# Indice

- Enunciado
- Problemas encontrados
- Mecanismos y demás
- Diagrama explicativo
- Código
- Bibliografía

# Enunciado

Una universidad tiene unos kioskos en los que hacer la matrícula de forma electrónica.

Cada kiosko tiene un proceso controlador que lee de un dispositivo de entrada los datos de matrícula, entre ellos el dni y el código de carrera, y los introduce en un buffer (común a todos los kioskos) de tamaño limitado. En otro lugar, unos procesos matriculadores leen los datos de matrícula del buffer, uno de los cuales es el código de carrera elegido y comprueban si hay plazas libres en dicha carrera. Si hay plazas en la carrera se disminuye en uno el número de plazas libres en esa carrera y se guarda el dni de la persona matriculada en la lista de matriculados de esa carrera.

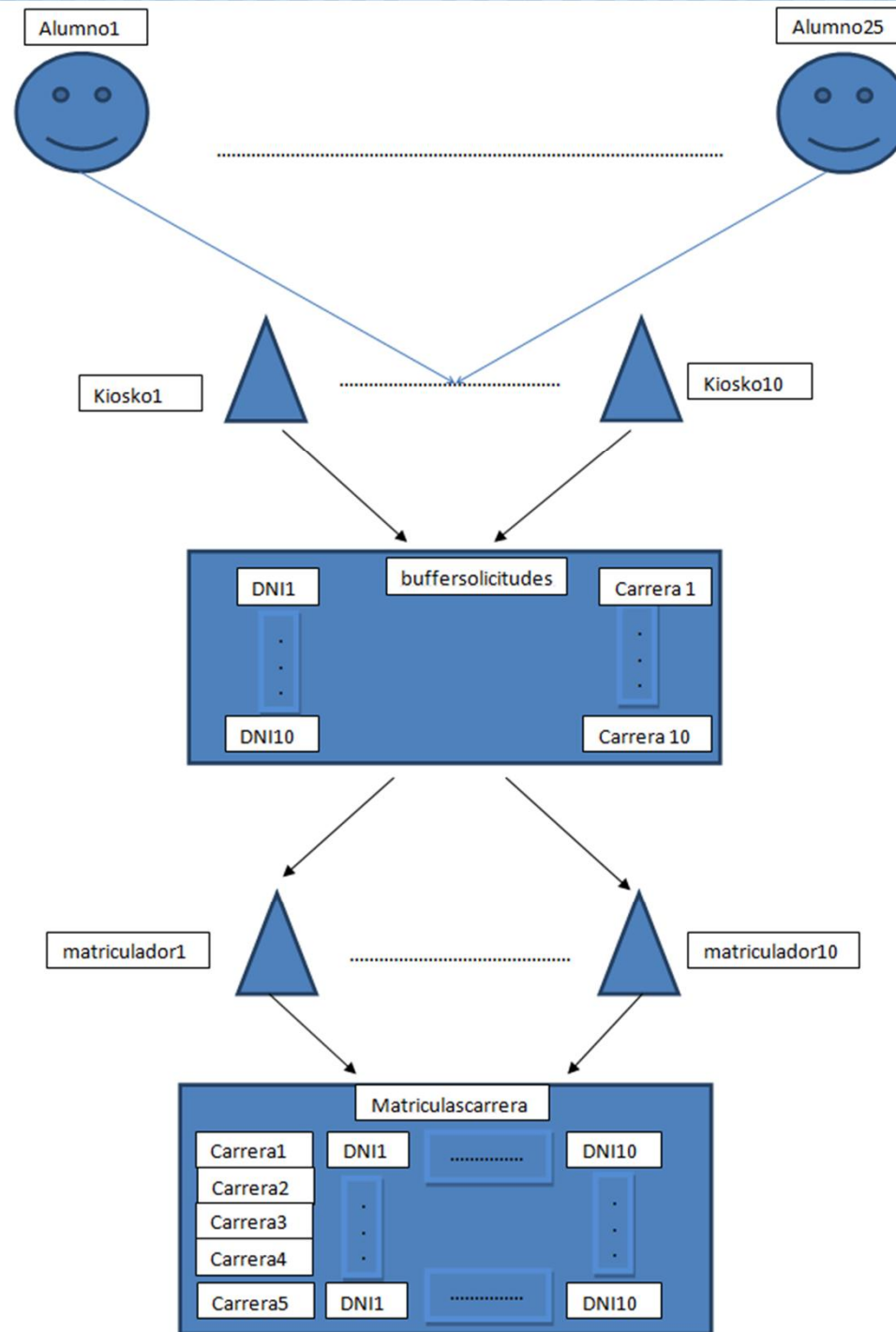
# Problemas encontrados en el enunciado

- Número de alumnos
- Número de procesos kiosko
- Número de procesos matriculadores
- Número de carreras y alumnos por carrera
- Número de solicitudes que podemos tener a la vez

# Mecanismos y demás

- Monitores
  - + Variables de condición
- Algoritmo Productor-Consumidor
- Variables globales
  - + Buffer para solicitudes
  - + Buffer para matriculados
  - + Número de alumnos

# Diagrama explicativo



# Código

```
program proyecto;
```

```
const n=10; (* seran 25 alumnos pero con un limite en el  
buffer de 10*)
```

```
var
```

```
alumnos:integer;
```

```
buffersolicitudes: array[1..n,1..2] of integer;
```

```
matriculascarrera: array[1..5,1..n] of integer;
```

## **monitor solicitudes;**

export Insertar, Extraer;

var

tam, frente, cola,dni,carrera: integer;

nolleno, novacio: condition;

**procedure Insertar(var m:integer);**

begin

if(alumnos<>0) then

begin

if tam=n then delay (nolleno); (\*bloquearse si no hay espacio\*)

buffersolicitudes[cola][1]:= dni;

repeat

carrera:=random(5);(\*5carreras\*)

until (carrera<>0);(\*el valor 0 no se tiene en cuenta como codigo de carrera\*)

buffersolicitudes[cola][2]:= carrera;

alumnos:=alumnos-1;

writeln('meto dni',dni,'carrera',carrera);

cola:=cola+1 ;

if(cola=11) then cola:=1;

tam:=tam+1;

dni:=dni+1;

resume (novacio); (\*desbloquear al consumidor\*)

end;

m:=alumnos;

end;



```
procedure Extraer(var dniaux,carreraaux,m:integer);  
begin  
  if((alumnos=0) and (tam=1)) then m:=0  
  else  
    begin  
      m:=1;  
      if tam=1 then delay(novacio); (*bloquearse si no hay elementos*)  
      dniaux:=buffersolicitudes[frente][1];  
      carreraaux:=buffersolicitudes[frente][2];  
      writeln('saco dni',dniaux,'carrera',carreraaux);  
      frente:=frente+1;  
      if(frente=11) then frente:=1;  
      tam:=tam-1;  
      resume(nolleno); (*desbloquear a un productor*)  
    end;  
  end;  
begin  
  alumnos:=25;  
  frente:=1;  
  cola:=1;  
  tam:=1;  
  dni:=1;  
end; (*fin monitor******)
```

### **monitor acceso;**

export matricular;

var alum1,alum2,alum3,alum4,alum5: integer;

### **procedure matricular(dni,carrera:integer);**

var dentro:boolean;

begin

dentro:=false;

while(dentro=false) do

begin

if (carrera=1)then

begin

if(alum1<>11)then

begin

dentro:=true;

matriculascarrera[carrera][alum1]:=dni;

alum1:=alum1+1;

end;

end

begin

alum1:=1;

alum2:=1;

alum3:=1;

alum4:=1;

alum5:=1;

end;

```

else
begin
  if (carrera=2)then
  begin
    if(alum2<>11)then
    begin
      dentro:=true;
      matriculascarrera[carrera][alum2]:=dni;
      alum2:=alum2+1;
    end;
  end
  else
  begin
    if (carrera=3)then
    begin
      if(alum3<>11)then
      begin
        dentro:=true;
        matriculascarrera[carrera][alum3]:=dni;
        alum3:=alum3+1;
      end;
    end
    else
    begin
      if (carrera=4)then
      begin
        if(alum4<>11)then
        begin
          dentro:=true;
          matriculascarrera[carrera][alum4]:=dni;
          alum4:=alum4+1;
        end;
      end;
    end;
  end;
end;

end
else
begin
  if (carrera=5)then
  begin
    if(alum5<>11)then
    begin
      dentro:=true;
      matriculascarrera[carrera][alum5]:=dni;
      alum5:=alum5+1;
    end;
  end
  end;
end;
end;
end;
if(dentro=false)then
begin
  writeln('El alumno con DNI',dni,'no se puede matricular en la
carrera'
,carrera);
repeat
  carrera:=random(5);(*5carreras*)
until (carrera<>0);(*el valor 0 no se tiene en cuenta como codigo*)
end
else
begin
  write('El alumno con DNI:',dni);
  writeln('se ha matriculado en la carrera:',carrera);
end;
end;(*fin while*) end;

```

**process type kiosko;**

```
var m:integer;  
begin  
    repeat  
        solicitudes.insertar(m);  
    until(m=0);  
end;
```

**process type matriculador;**

```
var dniaux,carreraaux,m:integer;  
begin  
    repeat  
        solicitudes.Extraer(dniaux,carreraaux,m);  
        if(m<>0) then acceso.matricular(dniaux,carreraaux);  
    until(m=0);  
end;
```

```
var  
kiosko1,kiosko2,kiosko3,kiosko4,kiosko5,  
kiosko6,kiosko7,kiosko8,kiosko9,kiosko10:kiosko;  
matric1,matric2,matric3,matric4,matric5,matric6,  
matric7,matric8,matric9,matric10:matriculador;  
begin  
    alumnos:=25;  
cobegin  
    kiosko1;kiosko2;kiosko3;kiosko4;kiosko5;kiosko6;  
    kiosko7;kiosko8;kiosko9;kiosko10;  
    matric1;matric2;matric3;matric4;matric5;matric6;matric7;  
    matric8;matric9;matric10;  
coend;  
end.
```

# Bibliografía

- Problema del productor-consumidor de las diapositivas de monitores.

# **Realizado por:**

David Sánchez García

Cristina Abad Cortés

GRACIAS POR VUESTRA  
ATENCIÓN

=)