

Areas

A la ciudad la temenos que pensar como campos de memoria, solo asignar lo minimo para su correcto funcionamiento del programa

Concurrencia simula el uso de la memoria, por lo tanto debemos ser cuidadosos como la asignamos, siempre lo minimo posible, no compartirla si no es necesario y evitar que 2 o mas recursos utilicen el mismo lugar. Entonces:

Area Compartida (C): cuando todos los robots comparten el mismo area y ninguna puede ser privada porque las usan TODOS los demas.

***Area privada (P):** cuando el robot usa un area y nadie mas, si necesita estar en la esquina 2,2, solo declaro 2,2, es decir: "area1:AreaP(2,2,2,2)". Asi como si necesito las primeras 10 avenidas de la calle uno SOLO declaro eso, es decir:
area1:AreaP(1,1,10,1).

***Area Parcialmente compartida (PC):** cuando ingresan 2 o mas a un mismo lugar pero por lo menos 1 del resto no ingresa, por ejemplo si tengo 3 robots, 1 jefe (que se queda quieto lejos, es decir en su AreaP), 1 recolector de flores y 1 de papeles, los cuales ambos entran en un cuadrante en comun declaro asi por ej: area1:AreaP(10,1,10,1)
area2:AreaPC (1,1,6,10)
Despues le asigno area1 al jefe y area2 a los otros 2.

Linkardo con mas
data

Definir bien los limites de las areas

```
1 programa estructura;  
2 procesos  
3   proceso saras()  
4   comenzar  
5   fin  
6 areas  
7   areaBot1: AreaP();  
8   areaBot2: AreaPC();  
9   areaBot2: AreaC();  
10 robots  
11   robot sarasa  
12   comenzar  
13   fin  
14 variables  
15 fin
```

A mano

<https://github.com/NahuelArn>

```
. variables
.   bot1, bot2, bot3: recolector
.   jefe: coordinador
. comenzar  PRIMERO NOMBRE DEL ROBOT Y DESPUES COSA
.   AsignarArea(bot1, areaBot1)  {area default}
.   AsignarArea(bot2, areaBot2)
.   AsignarArea(bot3, areaBot3)
.   AsignarArea(jefe, areaJefe)
.   AsignarArea(bot1, areaRecoleccion)  {area default}
.   AsignarArea(bot2, areaRecoleccion)
.   AsignarArea(bot3, areaRecoleccion)
.   Iniciar(bot1, 1,1)
.   Iniciar(bot2, 2,1)
.   Iniciar(bot3, 3,1)
.   Iniciar(jefe, 9,9)
. fin
```

Colisionadores

<https://github.com/NahuelArn>

Si 2 robots o mas robots comparten una area y hay alguna posibilidad q puedan caer los 2 en la misma esquina, temenos que bloquear/liberar

```
{Concurrencia maximizada, DEPOSITAR DE A UNA}
proceso bloquearDepositarLiberar(E avIni: numero; E caIni: numero; ES cantF: numero)
comenzar
  repetir cantF
    BloquearEsquina(35,50) {lugar posible colision}
    Pos(35,50)
    depositarFlor
    Pos(avIni, caIni) {vuelvo safe}
    LiberarEsquina(35,50)
  cant:= 0
fin
```

```
{Variacion, solo depoista una y vuelve safe}
proceso bloquearDepositarLiberar(E avIni: numero; E caIni: numero; ES cantF: numero)
comenzar
  BloquearEsquina(35,50)
  Pos(35,50)
  depositarFlor
  cantF:= cantF-1
  verificarSigano(id,cantF)
  Pos(avIni, caIni)
  LiberarEsquina(35,50)
fin
```

La colisión no tiene que ver con el uso de areas, cuando existe esa posibilidad debo bloquear la esquina, hacer todo lo que tengo que hacer, irme a un lugar seguro y luego desbloquearla.
Con que hago eso?

- 1-BloquearEsquina(avenida que quiero bloquear,calle que quiero bloquear)
- 2-LiberarEsquina(avenida que tenia bloqueada, calle que tenia bloqueada)

Quien salio primero / Quien Gano

Cuando la Carrera no se lleva por la misma calle o recurso, puede ser una alternativa bloquear el recurso, si se sabe de ante mano donde terminaria

```
programa prueba
areas
  areaRecoleccion: AreaPC(1,5,1,5)
  areaBots: AreaC(6,1,8,1) {cada Lugar deberia ser privado}
  areaCoordinador: AreaP(10,10,10,10)
robots
  robot recolector
  variables
    cantJuntar: numero
    caIni, avIni: numero
    id: numero
    sigo: boolean
    ACK: boolean
  comenzar
    RecibirMensaje(id, jefe)
    sigo:= V
    caIni:= PosCa
    avIni:= PosAv
    mientras(sigo)
      Random(cantJuntar, 1,5)
      BloquearEsquina(1,5)
      Pos(1,5)
      mientras (cantJuntar > 0 ) & (HayFlorEnLaEsquina)
        tomarFlor
        cantJuntar:= cantJuntar -1
      si(~HayFlorEnLaEsquina)
        sigo:= F
        EnviarMensaje(id, jefe)
        RecibirMensaje(ACK, jefe)
      Pos(avIni, caIni)
      LiberarEsquina(1,5)
  fin
```

Si quieres saber quien gano, cuando se entera el robot q gano, tenes q enviar un mensaje al jefe y dejar bloqueado el recurso hasta q te mande la confirmacion q recibio el mensaje el jefe

MANDO IDS, Y NO INTENTO MANEJARLO BLOQUEANDO RECURSOS
"PARA LA CATEDRA, SI NO NO COMPARTE EL MISMO FIN
DEJALO ASI Y SE ASUME Q NO PASA EL PROBLEMA DE UN SOLO NUCLEO"

<https://github.com/NahuelArn>

```
robot coordinador
variables
  id: numero
  PosQsalio: numero
comenzar
  EnviarMensaje(1, bot1)
  EnviarMensaje(2, bot2)
  EnviarMensaje(3, bot3)
  PosQsalio:= 0
  repetir 3
    RecibirMensaje(id, *)
    si(id = 1)
      EnviarMensaje(id, bot1)
    sino
      si(id = 2)
        EnviarMensaje(id, bot2)
      sino
        si(id = 3)
          EnviarMensaje(id, bot3)
    PosQsalio:= PosQsalio +1
    si(PosQsalio = 2)
      Informar('EsteRobotSalioSegundo',id)
  fin
variables
  bot1, bot2, bot3: recolector
  jefe: coordinador
comenzar
  AsignarArea(bot1, areaBots) {cada Lugar deberia ser privado}
  AsignarArea(bot2, areaBots)
  AsignarArea(bot3, areaBots)
  {recoleccion}
  AsignarArea(bot1, areaRecoleccion)
  AsignarArea(bot2, areaRecoleccion)
  AsignarArea(bot3, areaRecoleccion)
  AsignarArea(jefe, areaCoordinador)
  Iniciar(bot1, 6,1)
  Iniciar(bot2, 7,1)
  Iniciar(bot3, 8,1)
  Iniciar(jefe, 10,10)
fin
```

Quien salio primero / Quien Gano

Cuando los 2 accede a una esquina y quieren saber quien detecto q esa esquina estaba vacia

```
programa prueba
areas
  areaRecoleccion: AreaPC(1,5,1,5)
  areaBots: AreaC(6,1,8,1) {cada Lugar deberia ser privado}
  areaCoordinador: AreaP(10,10,10,10)
robots
  robot recolector
  variables
    cantJuntar: numero
    caIni, avIni: numero
    id: numero
    sigo: boolean
    ACK: boolean
  comenzar
    RecibirMensaje(id, jefe)
    sigo:= V
    caIni:= PosCa
    avIni:= PosAv
    mientras(sigo)
      Random(cantJuntar, 1,5)
      BloquearEsquina(1,5)
      Pos(1,5)
      mientras (cantJuntar > 0 ) & (HayFlorEnLaEsquina)
        tomarFlor
        cantJuntar:= cantJuntar -1
      si(~HayFlorEnLaEsquina)
        sigo:= F
        EnviarMensaje(id, jefe)
        RecibirMensaje(ACK, jefe)
      Pos(avIni, caIni)
      LiberarEsquina(1,5)
  fin
```

El Segundo que detecto que la esquina estaba vacia es el ganador, informar el id

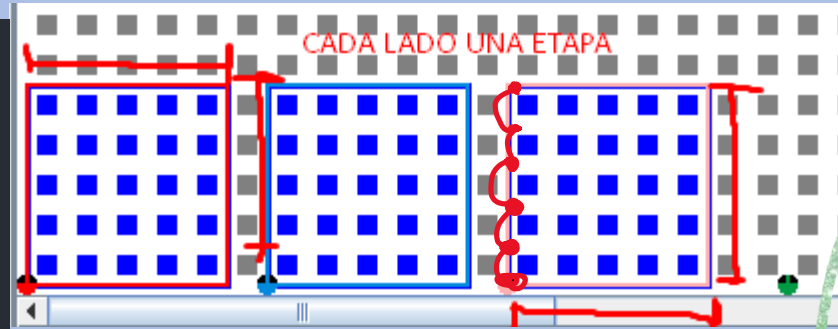
```
robot coordinador
variables
  id: numero
  PosQsalio: numero
comenzar
  EnviarMensaje(1, bot1)
  EnviarMensaje(2, bot2)
  EnviarMensaje(3, bot3)
  PosQsalio:= 0
  repetir 3
    RecibirMensaje(id, *)
    si(id = 1)
      EnviarMensaje(id, bot1)
    sino
      si(id = 2)
        EnviarMensaje(id, bot2)
      sino
        si(id = 3)
          EnviarMensaje(id, bot3)
    PosQsalio:= PosQsalio +1
    si(PosQsalio = 2)
      Informar('EsteRobotSalioSegundo',id)
  fin
variables
  bot1, bot2, bot3: recolector
  jefe: coordinador
comenzar
  AsignarArea(bot1, areaBots) {cada Lugar deberia ser privado}
  AsignarArea(bot2, areaBots)
  AsignarArea(bot3, areaBots)
  {recoleccion}
  AsignarArea(bot1, areaRecoleccion)
  AsignarArea(bot2, areaRecoleccion)
  AsignarArea(bot3, areaRecoleccion)
  AsignarArea(jefe, areaCoordinador)
  Iniciar(bot1, 6,1)
  Iniciar(bot2, 7,1)
  Iniciar(bot3, 8,1)
  Iniciar(jefe, 10,10)
fin
```

<https://github.com/NahuelArn>

Sincronizacion

Si es con jefe, puedo usar el * para la sincronizacion (la recepcion)

ConUnJefe



```
proceso sincronizarme
variables
  ok: boolean
comenzar
  EnviarMensaje(ok, jefe)
  RecibirMensaje(ok, jefe)
fin

robot cuadrado
variables
  cantF: numero
  id,avIni,caIni: numero
  ok: boolean
comenzar
  RecibirMensaje(id, jefe)
  repetir 4
    repetir 5
      juntarFlores(cantF)
      mover
      juntarFlores(cantF)
      derecha
      sincronizarme
  EnviarMensaje(id, jefe)
  EnviarMensaje(cantF, jefe)
  RecibirMensaje(ok, jefe) {si gane no hago naty}
  si(~ok)
    avIni:= PosAv
    caIni:= PosCa
    mientras (cantF > 0)
      bloquearDepositarLiberar(cantF,avIni,caIni)
  fin
```

```
robot coordinador
variables
  ok: boolean
  id: numero
  max,idMax,cantF: numero
comenzar
  asignarIds
  max:= -99
  idMax:= 0
  repetir 4
    repetir 3
      RecibirMensaje(ok, *)
      sincronizacionExitosa
    repetir 3
      RecibirMensaje(id, *)
      si(id = 1)
        RecibirMensaje(cantF, bot1)
      sino
        si(id = 2)
          RecibirMensaje(cantF, bot2)
        sino
          RecibirMensaje(cantF, bot3)
      sacarMax(cantF,id,max, idMax)
  si(idMax <> 0)
    Informar('ELmAXBotfuEee',idMax)
  sino
    Informar('naoMax',0)
  avisarResultados(idMax)
fin
```

```
proceso sincronizacionExitosa
variables
  ok: boolean
comenzar
  EnviarMensaje(ok, bot1)
  EnviarMensaje(ok, bot2)
  EnviarMensaje(ok, bot3)
fin
```

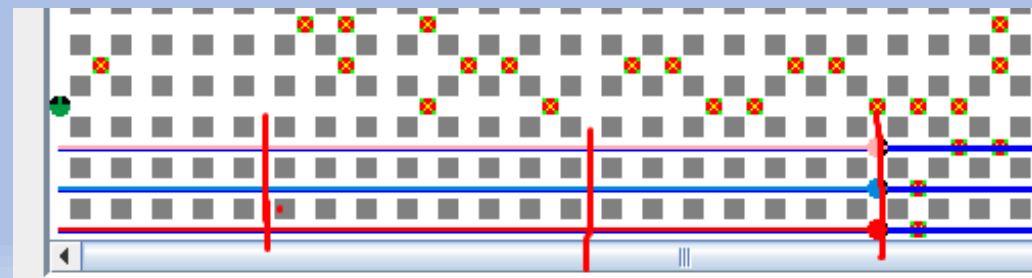
Cuando es entre trabajadores,
mando y espero en el orden que
envie el mensaje o
puede pasar un desfase

Sincronizacion

entreTrabajadores

```
robot recolector
variables
  contPasos,cantF: numero
  avIni,caIni,id: numero
  ok: boolean
comenzar
  derecha
  RecibirMensaje(id, jefe)
  contPasos:= 0
  repetir 99
    juntarFlores(cantF)
    mover
    contPasos:= contPasos+1
    si(contPasos = 10)
      avIni:= PosAv
      caIni:= PosCa
      bloquearDepositarLiberar(cantF,avIni,caIni)
      sincronizar(id)
      contPasos:= 0
    {el paso 10, lo cuenta bien}
  EnviarMensaje(ok, jefe)
```

```
proceso sincronizar(E id: numero)
variables
  ok: boolean
comenzar
  si(id = 1)
    EnviarMensaje(ok, bot2)
    EnviarMensaje(ok, bot3)
    RecibirMensaje(ok, bot2)
    RecibirMensaje(ok, bot3)
  sino
    si(id = 2)
      EnviarMensaje(ok, bot1)
      EnviarMensaje(ok, bot3)
      RecibirMensaje(ok, bot1)
      RecibirMensaje(ok, bot3)
    sino {soy el 3}
      EnviarMensaje(ok, bot1)
      EnviarMensaje(ok, bot2)
      RecibirMensaje(ok, bot1)
      RecibirMensaje(ok, bot2)
  fin
```



Iterar N veces desde el recolector

<https://github.com/NahuelArn>

```
robot fiscalizador
variables
  id: numero
  botRandom: numero
  avRandom, cantFtotal, caRandom: numero
  ok: boolean
comenzar
  cantFtotal:= 0
  repetir 8
    Random(botRandom, 1, 3)
    Random(avRandom, 40, 60)
    Random(caRandom, 40, 60)
    si(botRandom = 1)
      EnviarMensaje(avRandom, bot1)
      EnviarMensaje(caRandom, bot1)
    sino
      si(botRandom = 2)
        EnviarMensaje(avRandom, bot2)
        EnviarMensaje(caRandom, bot2)
      sino
        si(botRandom = 3)
          EnviarMensaje(avRandom, bot3)
          EnviarMensaje(caRandom, bot3)
  EnviarMensaje(0, bot1)
  EnviarMensaje(0, bot2)
  EnviarMensaje(0, bot3)
  repetir 3
    RecibirMensaje(ok, *)
  Pos(10,10) {no tengo peligro de colision, porq los 3 robots
  juntarFlores(cantFtotal)
  Pos(4,1)
  Informar('totalFjuntadas',cantFtotal)
fin
variables
  bot1, bot2, bot3: recolector
  jefe: fiscalizador
comenzar
```

```
robot recolector
variables
  cantF, id: numero
  sigo: boolean
  avRecibida, caRecibida: numero
  avIni, caIni: numero
  ok: boolean
comenzar
  cantF:= 0
  avIni:= PosAv
  caIni:= PosCa
  RecibirMensaje(avRecibida, jefe)
  mientras(avRecibida <> 0)
    RecibirMensaje(caRecibida, jefe)
    bloquearJuntarLiberar(avRecibida,caRecibida,cantF,avIni,caIni)
    RecibirMensaje(avRecibida, jefe)
  bloquearDepositarLiberar(avIni,caIni,cantF)
  EnviarMensaje(ok, jefe)
fin
```


Iterar N veces desde el jefe

<https://github.com/NahuelArn>

```
robot servidor
variables
  id: numero
  cont3: numero
  cantPrandom: numero
comenzar
  asignarIds
  cont3:= 0
  mientras(cont3 < 3)
    Random(cantPrandom, 1,5)
    RecibirMensaje(id, *)
    si(id = 1)
      bloquearDepositarLiberar(cantPrandom)
      EnviarMensaje(cantPrandom, bot1)
    sino
      si(id = 2)
        bloquearDepositarLiberar(cantPrandom)
        EnviarMensaje(cantPrandom, bot2)
      sino
        si(id = 3)
          bloquearDepositarLiberar(cantPrandom)
          EnviarMensaje(cantPrandom, bot3)
        sino
          cont3:= cont3+1
  fin
```

```
robot cliente
variables
  id,cantPaRecolectar: numero
  avIni, caIni: numero
comenzar
  RecibirMensaje(id, jefe)
  avIni:= PosAv
  caIni:= PosCa
  repetir 4
    EnviarMensaje(id, jefe)
    RecibirMensaje(cantPaRecolectar, jefe)
    repetir cantPaRecolectar
      movermeRecolectarLiberar(avIni,caIni)
      depositarPapeles(cantPaRecolectar) {no especifica si tambien lo deposito 1 a 1}
    EnviarMensaje(0, jefe) {lo atrapa el recibir id, si es 0 uno ya termino}
  fin
```

sacarUnMax

Con cantidades ya listas

```

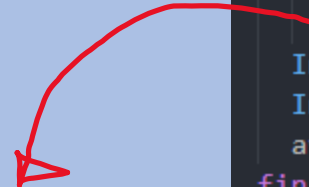
proceso sacarMax(ES idMax: numero; E bot1Cant: numero; E bot2Cant: numero;
E bot3Cant: numero; E bot4Cant: numero; ES max: numero)
comenzar
  max:= bot1Cant
  idMax:= 1
  si(bot2Cant > max)
    max:= bot2Cant
    idMax:= 2
  si(bot3Cant > max)
    max:= bot3Cant
    idMax:= 3
  si(bot4Cant > max)
    max:= bot4Cant
    idMax:= 4
fin

```

```

robot coordinador
variables
  id, totalF, idMax, maxF: numero
  ok: boolean
comenzar
  asignarIds
  maxF:= -99
  repetir 4
    repetir 3
      RecibirMensaje(ok, *)
      EnviarMensaje(ok, bot1)
      EnviarMensaje(ok, bot2)
      EnviarMensaje(ok, bot3)
    repetir 3
      RecibirMensaje(id, *)
      si(id = 1)
        RecibirMensaje(totalF, bot1)
      sino
        si(id = 2)
          RecibirMensaje(totalF, bot2)
        sino
          si(id = 3)
            RecibirMensaje(totalF, bot3)
      sacarMax(id, totalF, idMax, maxF)
      Informar('GANOelBot',idMax)
      Informar('conUnAcANTIDAD',maxF)
      avisarAlosQnoganaron(idMax)
  fin

```

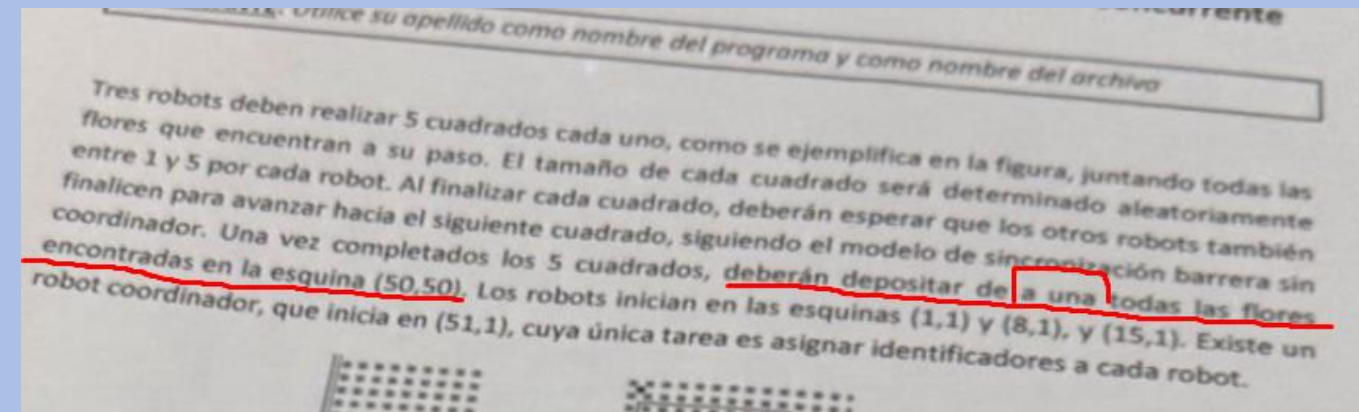


```

proceso sacarMax(E id: numero;E totalF: numero;ES idMax: numero;ES maxF: numero)
comenzar
  si(totalF > maxF)
    maxF:= totalF
    idMax:= id
fin

```

Depositar Maximizando la concurrencia



```
proceso bloquearDepositarLiberar(ES cantF: numero)
variables
  avIni, caIni: numero
comenzar
  avIni:= PosAv
  caIni:= PosCa
  repetir cantF
    BloquearEsquina(50,50)
    Pos(50,50)
    depositarFlor
    Pos(avIni, caIni)
    LiberarEsquina(50,50)
  cantF:= 0
fin
```

Figuras

```
procesos
proceso hacerLado
comenzar
  repetir 2
    repetir 4
      mover
      derecha
  fin
proceso hacerLado2
comenzar
  repetir 7
    mover
  derecha
  repetir 4
    mover
  derecha
fin
proceso girar(E cant: numero)
comenzar
  repetir cant
    derecha
fin
proceso hacerEscalon
comenzar
  girar(1)
  mover
  girar(3)
  mover
fin
```

```
robots
robot jeje
comenzar
  {cuadrado}
  repetir 2
    hacerLado
  {rectangulo}
  Pos(8,1)
  repetir 2
    hacerLado2
  {triangulo}
  Pos(14,1)
  repetir 5
    hacerEscalon
  derecha
  mover
  repetir 5
    hacerEscalon
  girar(2)
  repetir (25-14)
    mover
fin
```

