Programación II

MÓDULO CONCURRENTE

Memoria distribuida

Autores:
Alejandro Héctor Gonzalez
Silvana Lis Gallo

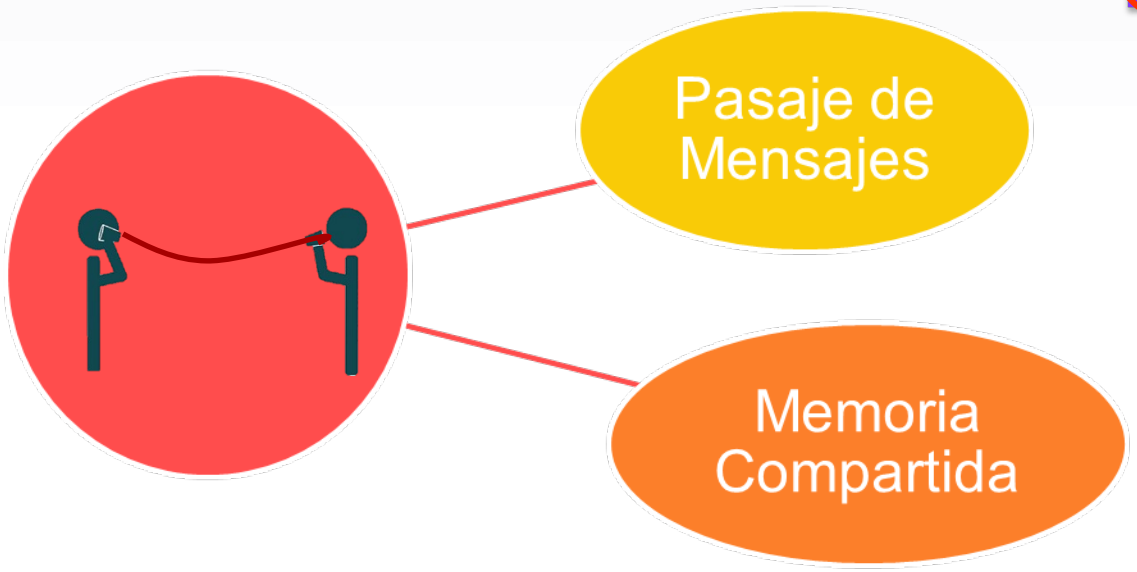
RESUMEN

En esta clase se trabaja el concepto de comunicación en la programación concurrente. En particular se explica el concepto de memoria distribuida a través de una implementación de pasaje de mensajes en RINFO

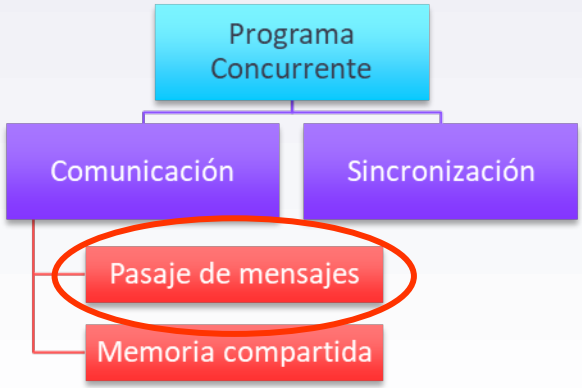
Palabras clave

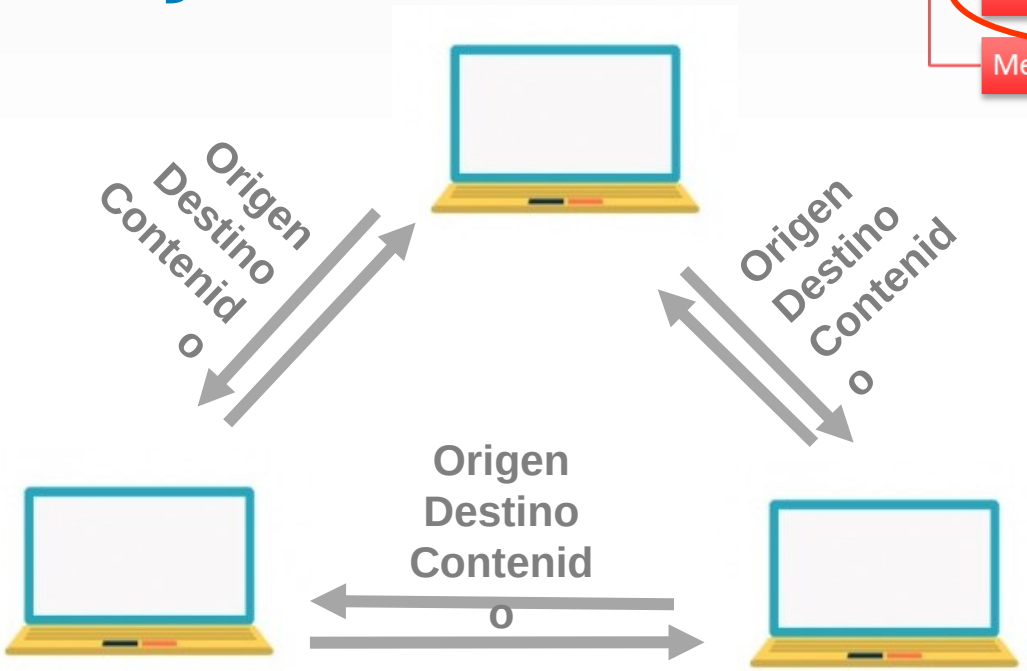
programación concurrente, memoria distribuida a pasaje de mensajes, enviar, recibir

Comunicación

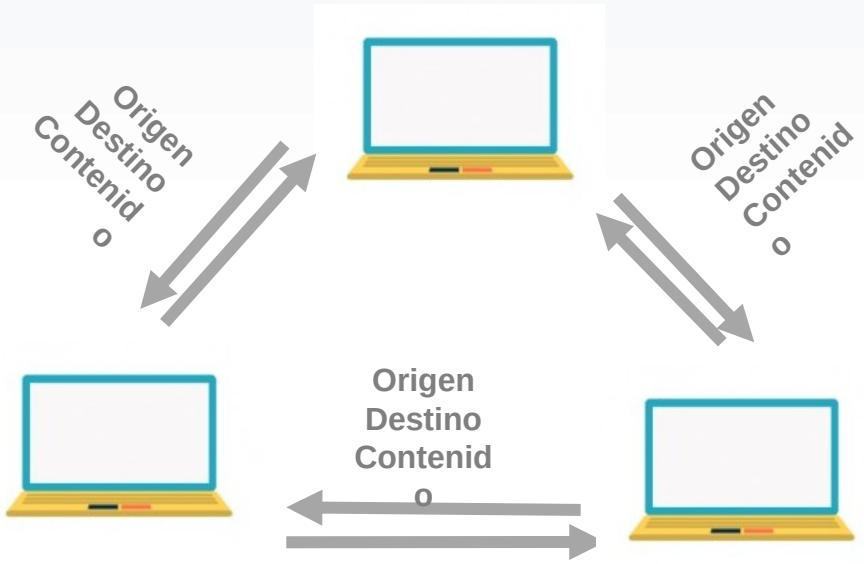


Comunicación: Pasaje de mensajes





Pasaje de mensajes



- Es necesario establecer **un canal** (lógico o físico) para transmitir información entre procesos.
- También el lenguaje debe proveer un **protocolo** adecuado.
- Para que la comunicación sea efectiva **los procesos deben "saber"** cuándo tienen mensajes para leer y cuando deben transmitir mensajes.

ENVIAR y RECIBIR

Comunicación: Pasaje de mensajes - Ejemplos

```
MPI_Send (buff, 128, MPI_CHAR,1 , 0,
MPI_COMM_WORLD);
```

Nombre Operación

Mensaje

Tipo Mensaje

Destino

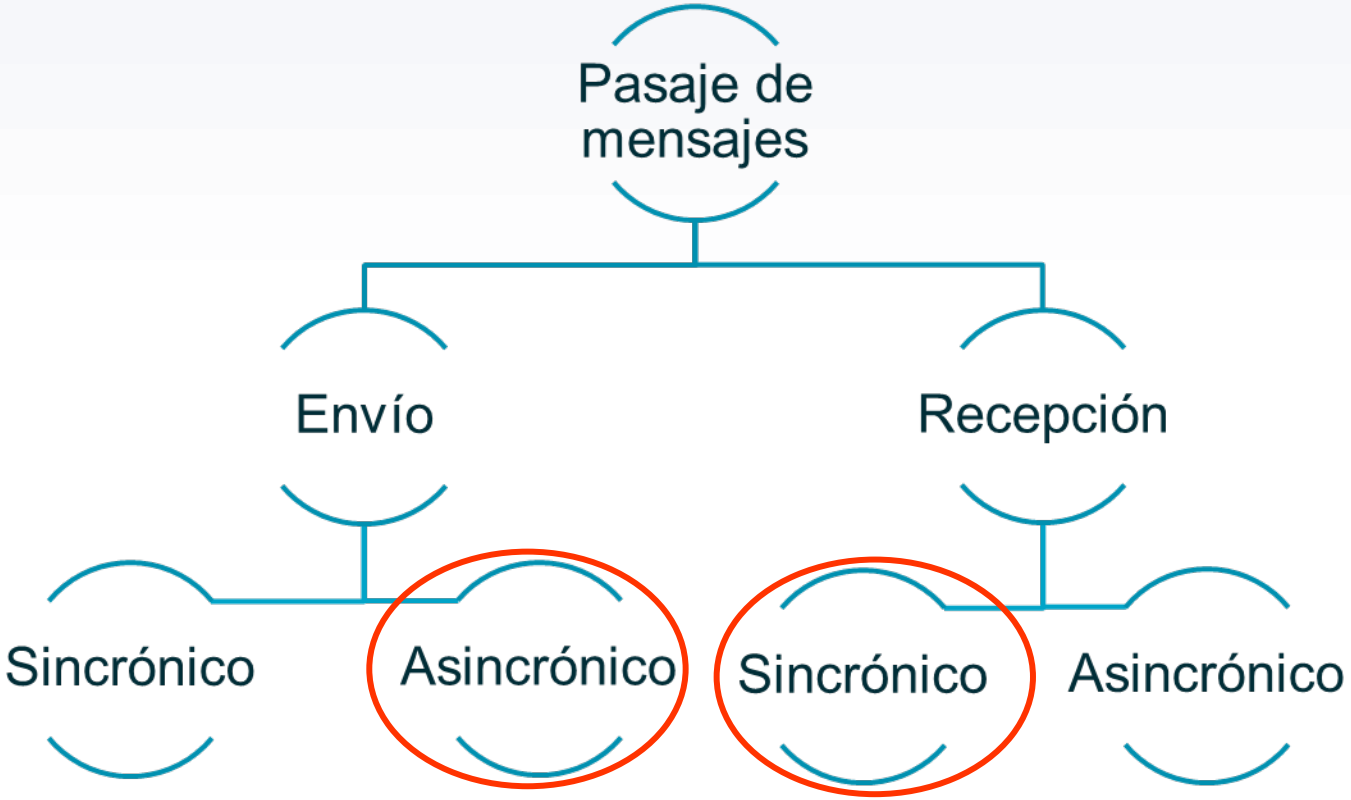
Origen

```
EnviarMensaje(dato, robot);
```

Mensaje

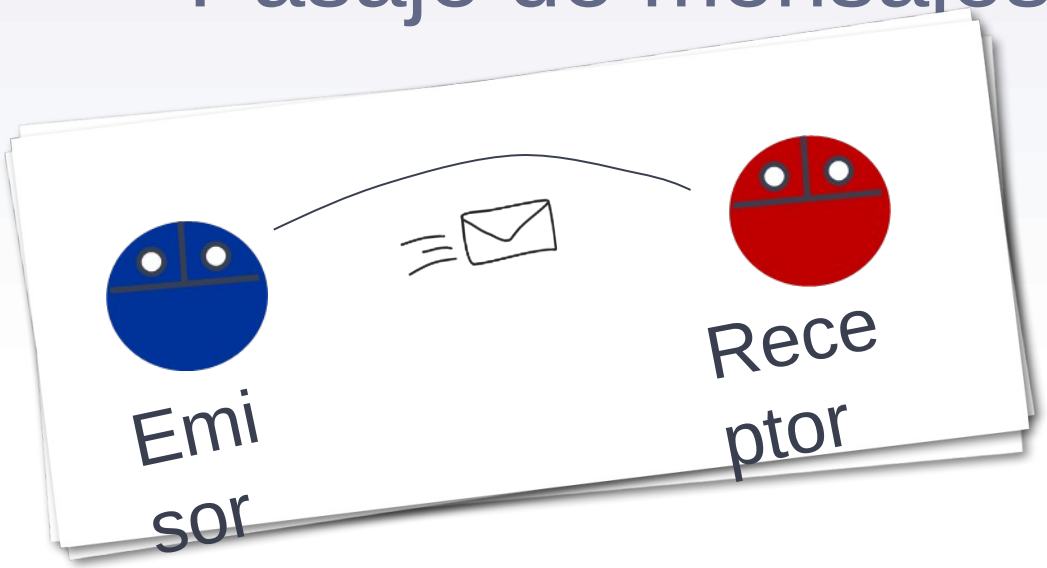
Destino

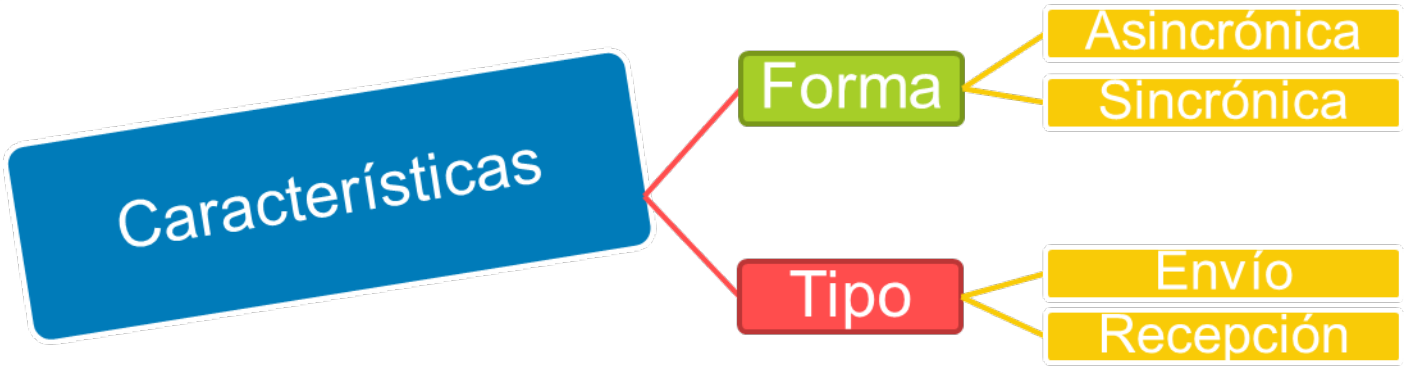
Comunicación: Pasaje de mensajes - Ejemplos



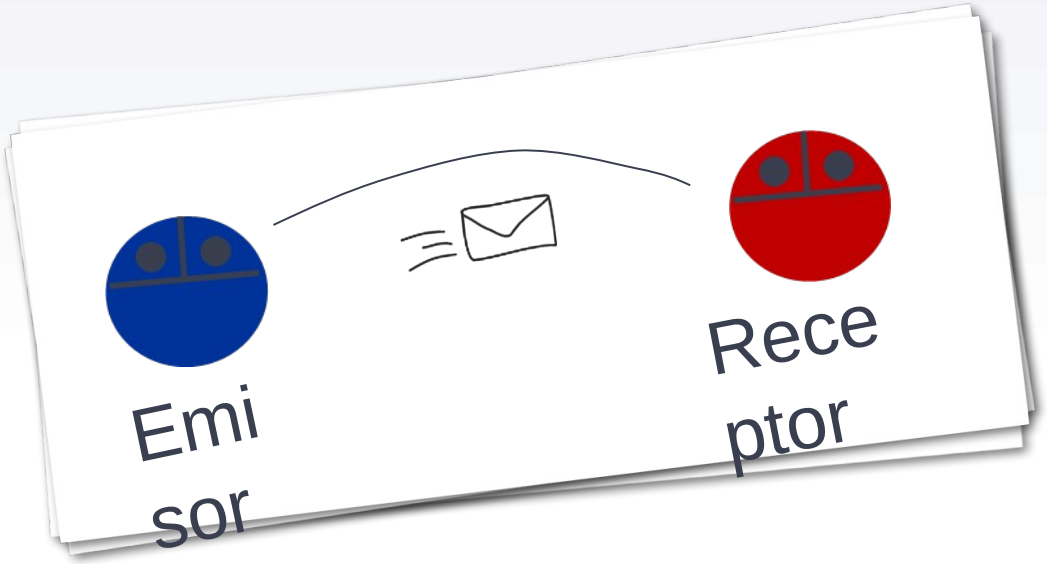
En RInfo

COMUNICACIÓN
Pasaje de mensajes





COMUNICACIÓN
ASINCRONICA



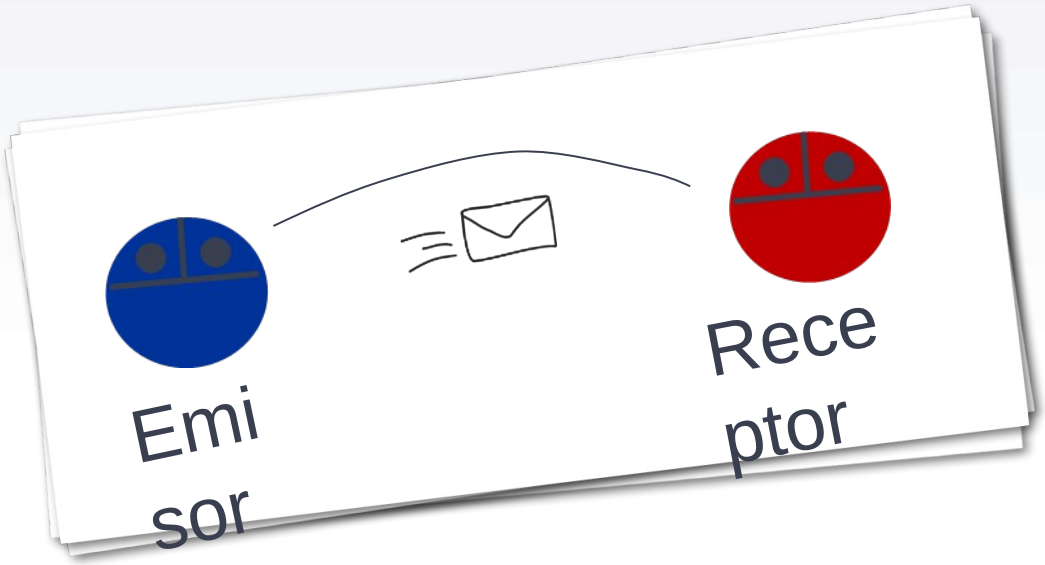
Asincrónica

El proceso que envía/recibe el mensaje **NO** espera que se de la comunicación para continuar

```
Instrucción 1
Instrucción 2

Sentencia de
comunicación
Instrucción 3
```

COMUNICACIÓN
SINCRONICA



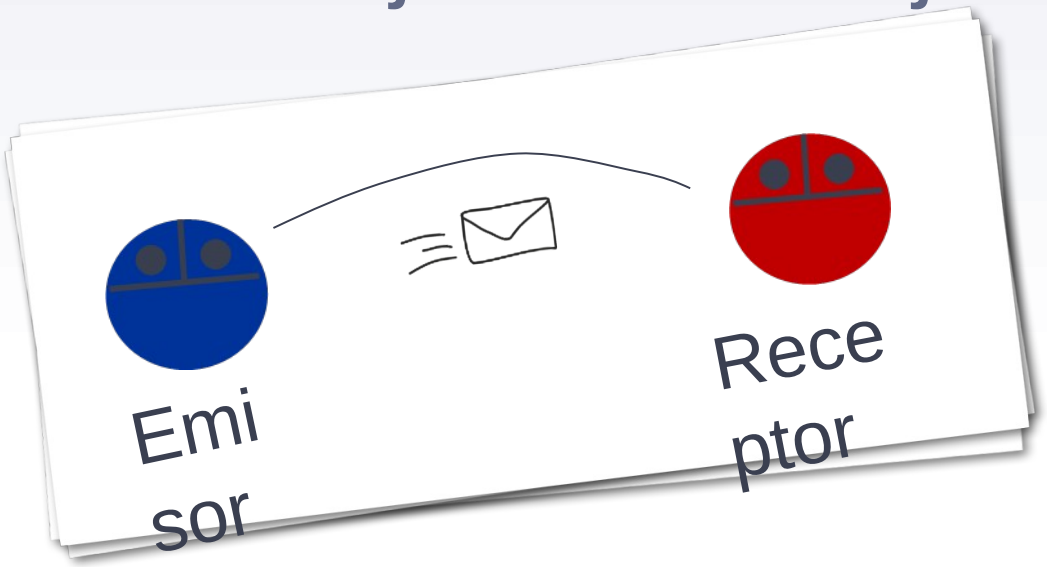
Sincrónica

El proceso que envía/recibe el mensaje espera que se de la comunicación para continuar

```
Instrucción 1
Instrucción 2
Sentencia de comunicación
Instrucción 3
```

Se da la comunicación

COMUNICACIÓN
Pasaje de mensajes



Puede haber varias combinaciones de sincronización

En el entorno RINFO el envío de un mensaje es no bloqueante (**asincrónico**) y la recepción es bloqueante (sincrónico).

ENVIO ASINCRONICO

Para enviar un mensaje en RInfo

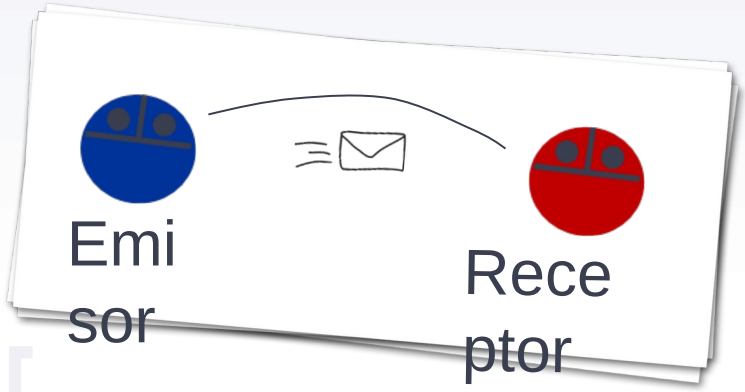

```
EnviarMensaje(número,variableRobot)
```

```
EnviarMensaje(3,robot1)

EnviarMensaje(flores,robot1)
```

EJERCICIOS DE RECEPCION SINCRONICA

Para recibir un mensaje en RIFNO



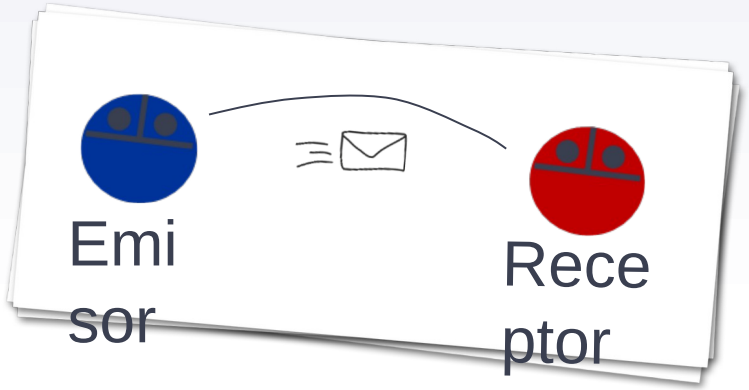

```
RecibirMensaje(variable,variableRobot)
```

```
RecibirMensaje(num,robot1)
```

Analice la solución presentada en el ***Ejercicio3-1.***

Analice la solución presentada en el ***Ejercicio3-2***. *(Observar la espera del robot 2 para informar)*

RECEPCION SINCRONICA




```
RecibirMensaje(variable,variableRob
ot)
```

```
RecibirMensaje(num,robot1
)
RecibirMensaje(num,*)
```

Analice la solución presentada en el ***Ejercicio3-3***. *(Observar como se modifica la espera del robot 2)*

- Ejercicios a realizar y consultar con los ayudantes

Ejercicios

Ejercicio 3-4: Teniendo en cuenta el Ejercicio **2-2c** donde 4 robots recorren cada uno un rectángulo y devuelve la cantidad de flores del perímetro, implemente una solución donde se agrega un robot jefe que informa el total de flores juntadas entre los 4 robots

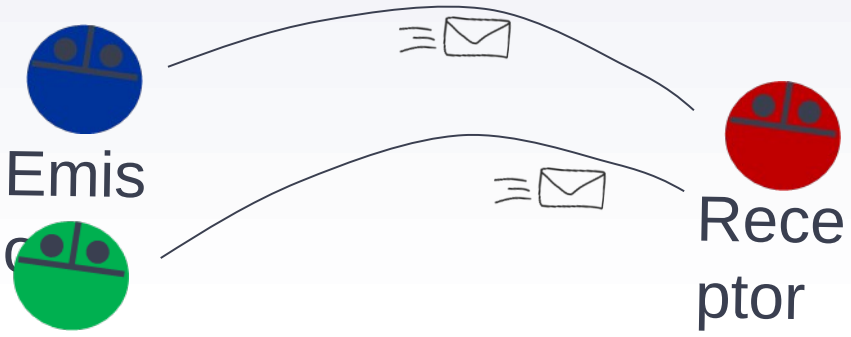
Ejercicio 3-5: Modifique el ejercicio 3-4 de manera que el **robot jefe** le indique a los demás dónde deben empezar su rectángulo. El robot1: (12,10), robot2 (16,10), robot3 (20,10) y robot4 (24,10). Inicialmente se encuentran en (2,2), (6,2), (10,2) y (14,2) como en el ejercicio 2.2c.

RECEPCION DE CUALQUIER ROBOT

¿Cómo sabe el receptor qué emisor envió el mensaje?

¿Cómo se quien envió?

¿Cómo sabe el receptor qué emisor envió el mensaje?



¿Cómo sabe un proceso quién es?

Emisor2

```
enviarMensaje(quienSoy,Receptor)
enviarMensaje(valor,Receptor)
```

Receptor

```
recibirMensaje(quienSos, *)
Si quienSos = 2
  recibirMensaje(valor, Emisor2)
```

Ejercicios

Ejercicio 3-6: Modifique el ejercicio 3-5 de manera que el jefe informe qué robot juntó más flores.

COMUNICACIÓN
Pasaje de mensajes - Ejercicios

Ejercicio 3-7: Modifique el ejercicio 3-6 de manera que cada robot realice un rectángulo de un alto variable. Para ello utilice el procedimiento Random.

```
Random (num, inferior, superior)
```

En la variable **num** queda almacenado un valor entre **inferior** y **superior**

Ejercicios

Ejercicio 3-8: : Implemente el siguiente juego. Existen 3 áreas privadas para cada uno de 3 robots (jugadores). Cada área se encuentra delimitada por las esquinas (2,2) (7,7); (8,2) (13,7); (14,2) (19,7) respectivamente. Además existe un robot fiscalizador.

El juego consiste en que cada robot jugador debe tratar de juntar la mayor cantidad de flores posible, para esto tiene tres intentos. En cada intento se posiciona en una esquina determinada al azar (dentro de su área) y junta todas las flores de esa esquina y vuelve a su esquina original. El robot fiscalizador determinará cuántas flores juntaron entre los 3 robots. Los robots se posicionan
(14,2) y el ro

¿Qué ocurre si en lugar de estar en 3 áreas privadas los robots deben juntar las flores de un área compartida?