



# Taller de programación Módulo Programación Concurrente

Clase 3

# Organización de la clase

1. Pasaje de mensajes
2. Pasaje de mensajes en CMRE
3. Ejercicios

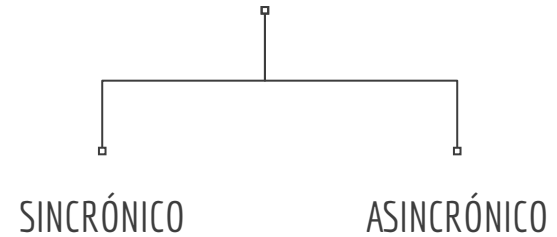
# Pasaje de mensajes



# Pasaje de mensajes



Enviar mensaje - Recibir mensaje



# Pasaje de mensajes

## PASAJE DE MENSAJES - ASINCRÓNICO

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



Instrucción 1

Instrucción 2

Sentencia de comunicación

Instrucción 3

...

# Pasaje de mensajes

## PASAJE DE MENSAJES - SINCRÓNICO

El proceso que envía/recibe el mensaje SI espera que se de la comunicación para continuar su ejecución.



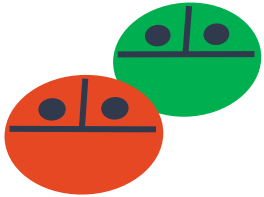
Instrucción 1  
Instrucción 2  
Sentencia de comunicación  
Instrucción 3  
...



# Pasaje de mensajes en CMRE

ENVÍO DE MENSAJES → ASINCRÓNICO

Un robot después de enviar un mensaje puede continuar su ejecución.



RECEPCIÓN DE MENSAJES → SINCRONICO

Un robot que está esperando recibir un mensaje NO puede continuar su ejecución hasta que otro robot no le haya mandado el mensaje.

# Pasaje de mensajes en CMRE

## Sintaxis - Enviar Mensaje

EnviarMensaje(número, variableRobot)

```
programa ejemploEnvio
areas
  area1: AreaPC(1,1,20,20)
robots
  robot tipo1
variables
  flores: numero
comenzar
  ...
fin
variables
  robot1: tipo1
  robot2: tipo1
  ...
```

robot1 necesita enviar un mensaje a robot2



EnviarMensaje(3, robot2)

EnviarMensaje(flores, robot2)



# Pasaje de mensajes en CMRE

## Sintaxis - Recibir Mensaje

RecibirMensaje(variable, variableRobot)

```
programa ejemploEnvio
areas
  area1: AreaPC(1,1,20,20)
robots
  robot tipo1
variables
  flores: numero
comenzar
  ...
fin
variables
  robot1: tipo1
  robot2: tipo1
  ...
```

robot1 necesita recibir un mensaje a robot2



RecibirMensaje(flores, robot2)

# Pasaje de mensajes en CMRE

## Sintaxis - Recibir Mensaje

RecibirMensaje(variable, variableRobot)

```
programa ejemploEnvio
areas
  area1: AreaPC(1,1,20,20)
robots
  robot tipo1
variables
  flores: numero
comenzar
  ...
fin
variables
  robot1: tipo1
  robot2: tipo1
  robot3: tipo1
  ...
```

robot1 necesita recibir un mensaje de cualquier robot



← - - - - Cualquier  
Robot

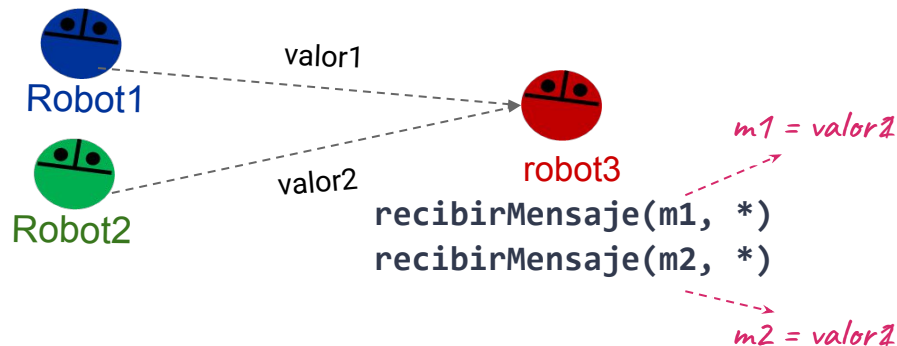
RecibirMensaje(flores, \*)

# Pasaje de mensajes en CMRE

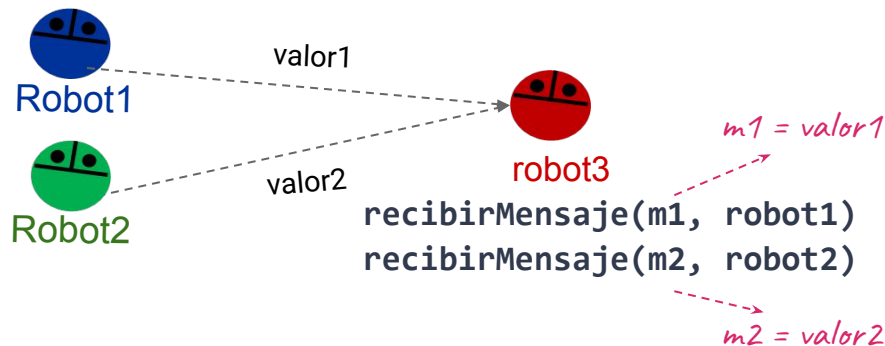
## Sintaxis - Recibir Mensaje

### Ejemplos

#### Ejemplo 1



#### Ejemplo 2



# Ejercicios

# Ejercicios

En Ideas descargar las versiones del Ejercicio3 llamadas:

- **Ejercicio3-1**
- **Ejercicio3-2**
- **Ejercicio3-3**

# Ejercicios



**Ejercicio 3-4:** Teniendo en cuenta el Ejercicio1-hb realice un programa donde 6 robots recorran cada uno un rectángulo (5x3) y devuelvan la cantidad de flores del perímetro. La solución debe tener un robot jefe que informe el total de flores juntadas entre los 6 robots. Los robots deben iniciar sus rectángulos en las esquinas (2,2), (6,2), (10,2), (14,2), (18,2) y (22,2) respectivamente. El robot jefe se posiciona en la esquina (1,1).



**Ejercicio 3-5:** Modifique el Ejercicio 3.4 de manera que el robot jefe le indique a los demás dónde deben empezar su rectángulo: robot1 (12,10), robot2 (16,10), robot3 (20,10), robot4 (24,10), robot5 (28,10) y robot6 (32,10). Los robots inician en las esquinas (2,2), (6,2), (10,2), (14,2), (18,2) y (22,2) respectivamente.

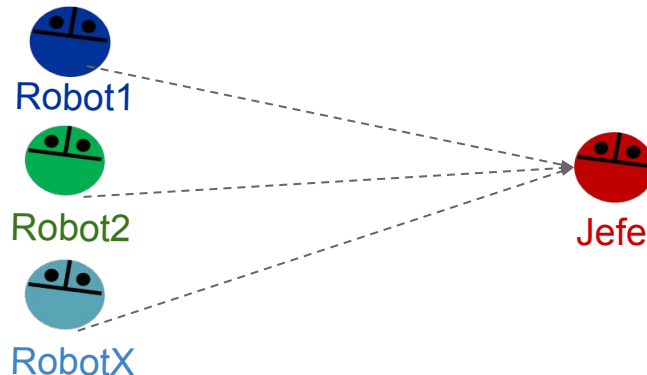
# Ejercicios



**Ejercicio 3-6:** Modifique el Ejercicio 3.5 para que el jefe, en lugar de informar la suma de las cantidades, informe el orden en que terminaron los robots

*¿Cómo puede saber el jefe quién fue terminando?*

*Los robots que recorren el perímetro deben saber quiénes son.*

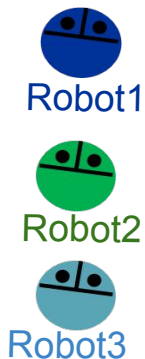


*El jefe debe recibir de cada robot que termina el identificador*

# Ejercicios

*¿Cómo puede saber el jefe quién fue terminando?*

*Los robots que recorren el perímetro deben saber quiénes son.*



*El jefe debe recibir de cada robot que termina el identificador*



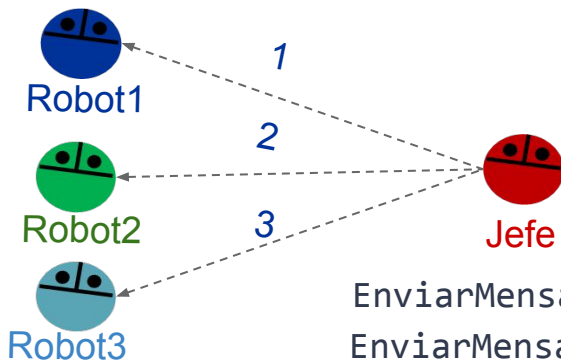
# Ejercicios

*¿Cómo puede saber el jefe quién fue terminando?*

*Los robots que recorren el perímetro deben saber quiénes son.*

*¿Cómo?*

*Cada robot debe recibir del jefe un identificador que le indique quién es*



```
EnviarMensaje(1, robot1)  
EnviarMensaje(2, robot2)  
EnviarMensaje(3, robot3)
```

# Ejercicios

*¿Cómo puede saber el jefe quién envió el mensaje?*

*El jefe debe recibir de cada robot que termina el identificador*



Robot2

```
...  
enviarMensaje(quienSoy, Jefe)  
...
```



Jefe

```
...  
recibirMensaje(quiensos, *)  
Informar('Llegó ', quiensos)
```

# Ejercicios



**Ejercicio 3-7:** Modifique el Ejercicio 3-6 para que el robot jefe informe qué robot juntó más flores.

*¿Cómo puede calcular el jefe quién juntó más flores?*

*Los robots que recorren el perímetro deben saber quiénes son.*

*El jefe debe recibir de cada robot el identificador y la cantidad juntada*

# Ejercicios

*¿Cómo puede calcular el jefe quién juntó más flores?*

Opción 1



Robot1



Robot2



RobotX

...

enviarMensaje(**quienSoy**, Jefe)

enviarMensaje(valor, Jefe)



Jefe

...

Repetir 6

recibirMensaje(**quienSos**, \*)

recibirMensaje(valor, \*)

ActualizarMaximo(quienSos, valor, rMax, max)

Informar('El robot máximo ', **rMax**)

Informar('La cant. máxima ', **max**)

*El jefe debe recibir  
de cada robot el  
identificador y la  
cantidad juntada*

# Ejercicios

## Opción 1 (erronea)



Robot1

*QuienSoy = 1*

```
enviarMensaje(quienSoy, Jefe)
```

```
enviarMensaje(valor, Jefe)
```



Robot2

*QuienSoy = 2*

```
enviarMensaje(quienSoy, Jefe)
```

```
enviarMensaje(valor, Jefe)
```

*¿Por qué es errónea?*



Jefe

Repetir 6

```
recibirMensaje(quienSos, *) quienSos = 1
```

```
recibirMensaje(valor, *) valor = 2
```

```
ActualizarMaximo(quienSos, valor, rMax, max)
```

# Ejercicios

*¿Cómo puede calcular el jefe quién juntó más flores?*

Opción correcta



Robot1



Robot2



RobotX

...

```
enviarMensaje(quienSoy, Jefe)
```

```
enviarMensaje(valor, Jefe)
```



Jefe

...

Repetir 6

```
recibirMensaje(quienSos, *)
```

```
si ( quienSos = 1)
```

```
    recibirMensaje(valor, robot1)
```

*El jefe debe recibir  
de cada robot el  
identificador y la  
cantidad juntada*

# Ejercicios

## Opción 2



Robot1

*QuienSoy = 1*

enviarMensaje(*quienSoy*, Jefe)

enviarMensaje(valor, Jefe)

*valor = 15*



Robot2

*QuienSoy = 2*

enviarMensaje(*quienSoy*, Jefe)

enviarMensaje(valor, Jefe)



Jefe

Repetir 6

recibirMensaje(*quienSos*, \*) *quienSoS = 1*

si(*quienSos* = 1)

recibirMensaje(valor, robot1) *valor = 15*

# Ejercicios



**Ejercicio 3-7:** Modifique el Ejercicio 3-6 para que el robot jefe informe qué robot juntó más flores.



Enviar por la mensajería el Ejercicio 3-7



# Ejercicios



**Ejercicio 3-8:** Teniendo en cuenta el Ejercicio 3-6 realice un programa donde 6 robots decidan de qué alto es el rectángulo a recorrer utilizando una función random.

Los robots arrancan sus rectángulos en la esquina (2,2), (6,2), (10,2), (14,2), (18,2) y (22,2) respectivamente. El robot jefe se posiciona en la esquina (1,1).

Random (num, inferior, superior)

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

# Ejercicios



**Ejercicio 3-9:** Implemente el siguiente juego: Existen 4 robots. Uno de los robots será el fiscalizador, y los otros tres robots, serán los Jugadores que realizarán una tarea en sus áreas privadas delimitadas por las esquinas (2,2) (7,7); (8,2) (13,7); (14,2) (19,7) respectivamente.

Cada robot jugador debe tratar de juntar la mayor cantidad de flores posible, para esto tiene 3 intentos. En cada intento se posicionará en una esquina determinada al azar (dentro de su área), juntará todas las flores de esa esquina y volverá a su esquina original. El robot fiscalizador determinará cuántas flores juntaron entre los 3 robots. Los robots se posicionarán inicialmente en (2,2), (8,2) y (14,2) y el robot fiscalizador en la esquina (1,1).

*¿Qué ocurre si los jugadores se encuentran en un área compartida?*