

Exemple

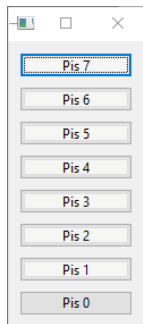
Sistemes distribuïts

Problema: Botoneres

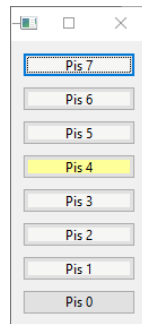
Una botonera és un artefacte gràfic que simula una botonera d'ascensor. En la implementació que us subministrem, aquesta simulació es fa usant la llibreria de widgets wxWidgets i el seu vincle a Erlang. Des del punt de vista d'Erlang una botonera s'abstreu com un procés: cada missatge enviat al procés provoca una acció a la botonera. Tota aquesta infraestructura està continguda en el mòdul bots (de BOTonera Simple). El mòdul bots exporta una funció:

`bots:nou(N,Pid)` Inicialitza el sistema de widgets gràfics. Crea una nova botonera amb N botons, que corresponen a N plantes. El procés associat a aquesta nova botonera informarà dels esdeveniments capturats per la botonera al procés Pid. Retorna el pid del procés associat a la botonera.

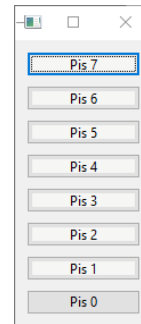
`B=bots:nou(8,self()).`



`B!{light_on,4}.`



`B!{light_off,4}.`



- El protocol amb que es pot parlar al procés de la botonera és el que defineixen els següents missatges:

{light_on, N}	Il·lumina el botó corresponent al pis N.
{light_off, N}	Apaga la il·luminació del botó corresponent al pis N.
close	Tanca la botonera.

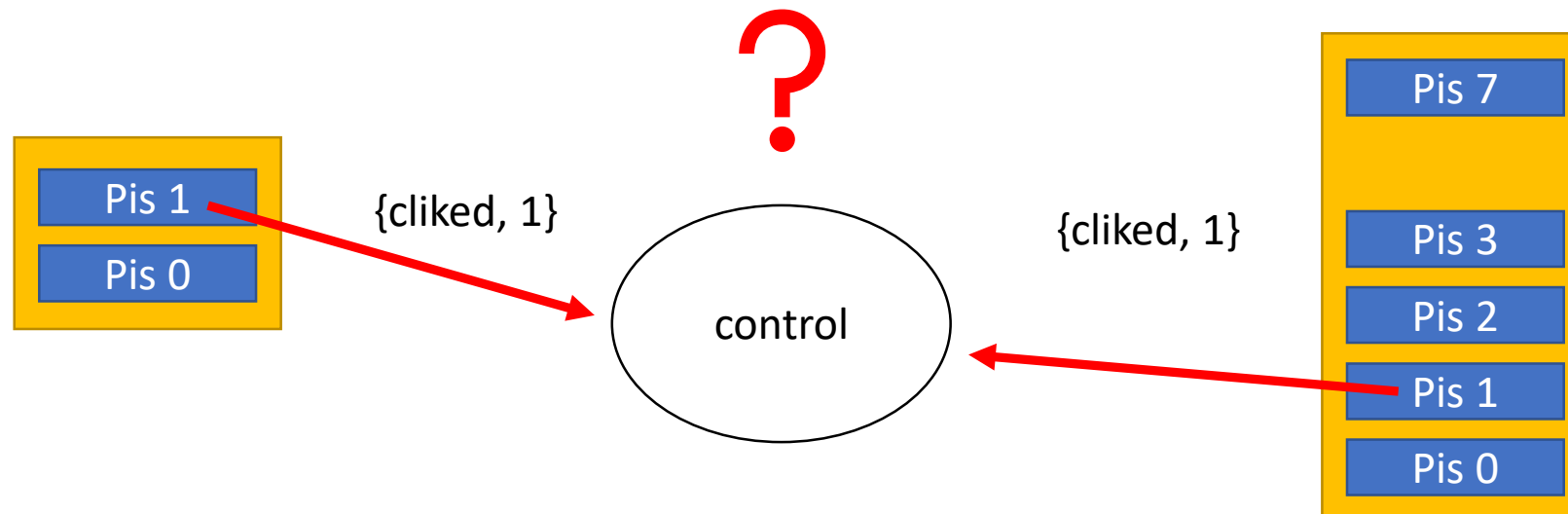
- De forma similar, la botonera pot enviar al procés que s'ha indicat en la seva creació els següents missatges:

{clicked, Pis}	Indica que s'ha polsat el botó corresponent al pis Pis.
abort	Indica que s'ha destruït la botonera per una acció de l'usuari (per exemple tancant la finestra).

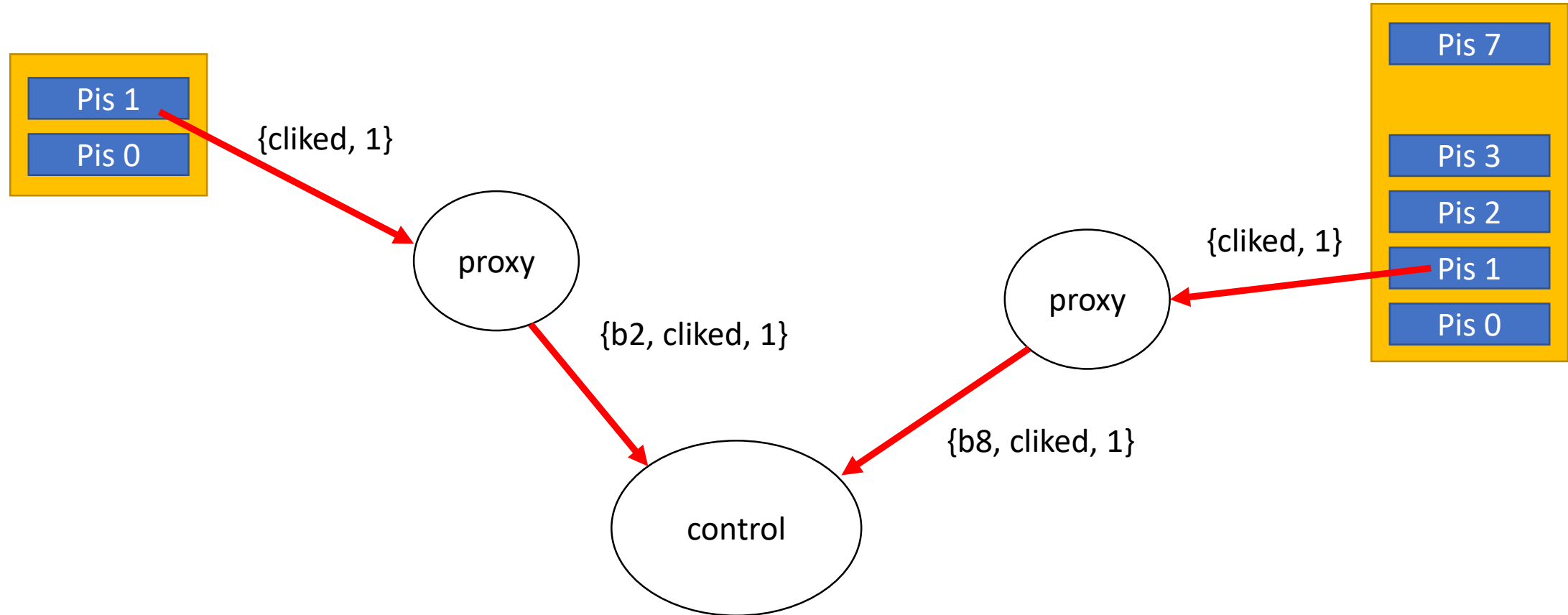
- Podeu comprovar la recepció dels missatges a la consola amb la comanda flush().

Un cop vist el funcionament de les botoneres, es demana que dissenyeu un programa que faci el següent:

1. crea dues botoneres, una de dos pisos i una altra de 8 pisos.
2. Cada vegada que es polsa un botó de la botonera petita, s'il·luminen els pisos parells o senars (segons si s'ha polsat el 1 o el 2) de la botonera gran.
3. Si es pitja un botó que correspon als pisos il·luminats aleshores s'apaguen.



Necessitem un: proxy



control pot diferenciar entre els missatges de la botonera 2 i els de la botonera 8

Proxy:

Necessitem el pid
de la botonera
per poder-li
enviar missatges

```
proxy(N,Nom,Pid)->%N es la mida total  
  B = spawn(modul,bucle,[Nom,Pid]),  
  bots:nou(N,B).
```

```
bucle(Nom,Pid)->  
  receive  
    {clicked,Pis} ->  
      Pid!{Nom,clicked,Pis},  
      bucle(Nom,Pid);  
  abort ->  
    Pid!{Nom,abort},  
    ok  
end.
```

botonera()->

```
B = proxy(8,b8,self()),  
L = proxy(2,b2,self()),  
bucle(B,L).
```

bucle(B,L)->

```
receive
```

```
{b2,clicked,Pis} ->
```

```
if (Pis == 0) -> encen(B, 0);  
   (Pis == 1) -> encen(B, 1)
```

```
end,
```

```
bucle(B,L);
```

```
{b8,clicked,Pis} ->
```

```
if (Pis rem 2 == 0) -> apaga(B, 0);  
   (Pis rem 2 == 1) -> apaga(B, 1)
```

```
end,
```

```
bucle(B,L);
```

```
{b8, abort} ->
```

```
L!close,
```

```
receive _ -> ok end;
```

```
{b2, abort} ->
```

```
B!close,
```

```
receive _ -> ok end
```

end.

encen(B,Pis) ->

```
if
```

```
Pis < 6 -> %Mientras no has acabat ves incrementant Pis
```

```
B!{light_on,Pis},
```

```
encen(B, Pis+2);
```

```
Pis >= 6 -> % finalitza si val 6 o més
```

```
B!{light_on,Pis}
```

```
end.
```

apaga(B, Pis)->

```
if
```

```
Pis < 6 ->
```

```
B!{light_off,Pis},
```

```
apaga(B, Pis+2);
```

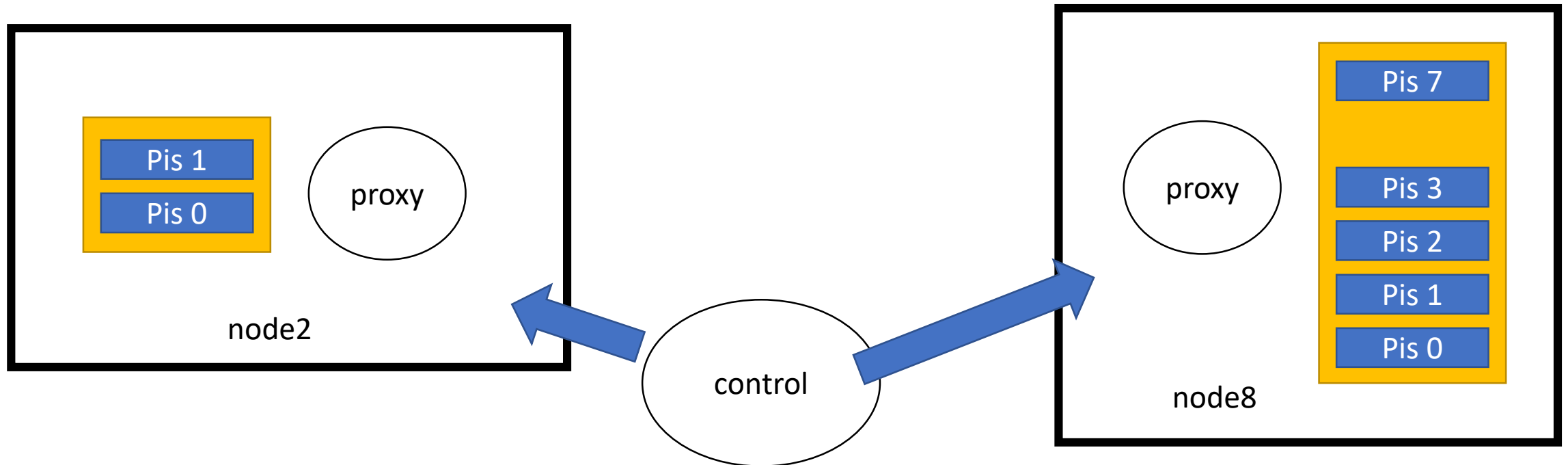
```
Pis >= 6 ->
```

```
B!{light_off,Pis}
```

```
end.
```

- Modifiqueu l'exercici anterior per executar les botoneres en diferents nodes. Feu que en un dels nodes s'iniciï la botonera de 2 i a l'altre node la de 8. Recordeu que per poder comunicar dos processos ubicats en nodes diferents, s'ha de saber el nom del node i el seu identificador. Us serà útil registrar els processos. Per evitar problemes d'accés a la xarxa, podeu obrir dos nodes diferents al mateix ordinador.

- a) Engegar les dues botoneres des del mateix node (node2), registrant el procés de l'altre node(node8)




```
-module(ex3a).  
-export([start/0,botonera/0, proxy/3,bucle_proxy/2]).
```

```
start() -> spawn(ex3a, botonera, []).
```

```
botonera()->
```

```
    % Iniciar la botonera de 8 al node8
```

```
    spawn( node8@maquina, ex3b, proxy, [8, b8, self()]), % Crear la boto
```

```
    %Iniciar la botonera de 2
```

```
    L = proxy(2,b2,self()), % L -> Pid de la botonera de 2
```

```
    bucle(L).
```

```
-module(ex3b).
```

```
-export([proxy/3,bucle_proxy/2]).
```

```
proxy(N,Nom,Pid)->%N es la mida total
```

```
    B = spawn(ex3b,bucle_proxy,[Nom,Pid]),  
    register(bot8,bots:nou(N,B)).
```

```
bucle_proxy(Nom,Pid)->
```

```
    receive
```

```
{clicked,Pis} ->
```

```
    Pid!{Nom,clicked,Pis},
```

```
    bucle_proxy(Nom,Pid);
```

```
abort ->
```

```
    Pid!{Nom,abort},
```

```
    ok
```

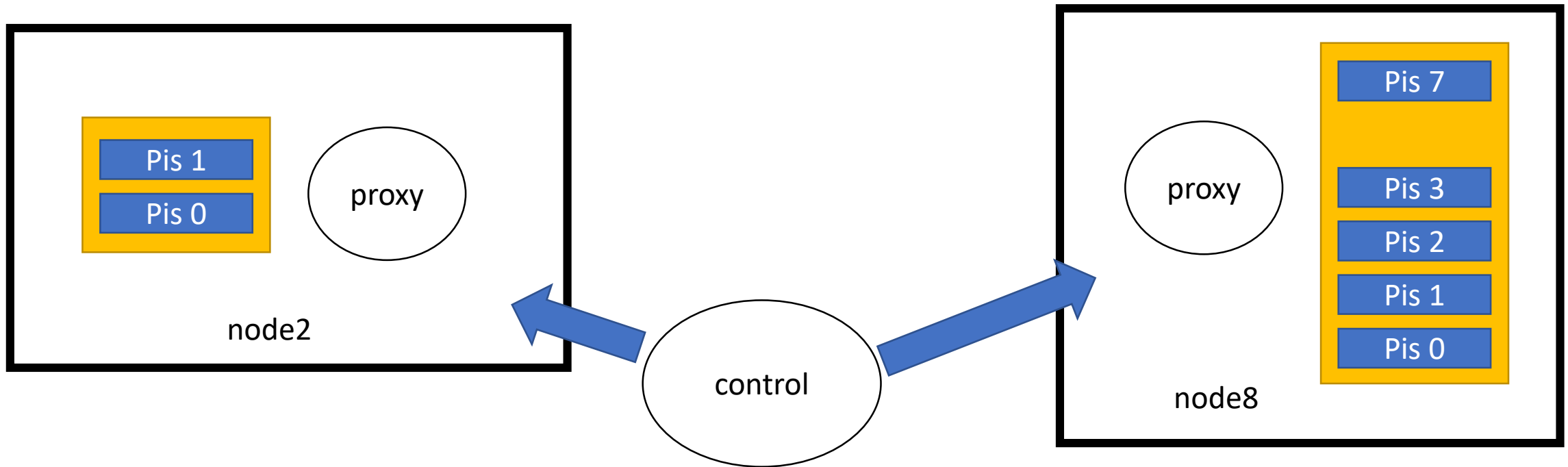
```
end.
```

```
bucle(L)->
  receive
{b2,clicked,Pis} ->
  If
    (Pis == 0) -> encen( 0);
    (Pis == 1) -> encen( 1)
  end,
  bucle(L);
{b8,clicked,Pis}->
  If
    (Pis rem 2 ::= 0) -> apaga(0);
    (Pis rem 2 ::= 1) -> apaga(1)
  end,
  bucle(L);
{b8, abort} ->
  L!close,
  receive _ -> "ok" end;
{b2, abort} ->
  {bot8, node8@maquina}!close,
  receive _ -> "ok" end
end.
```

```
encen(Pis) ->
  if
Pis < 6 ->
    {bot8, node8@maquina}!{light_on,Pis},
    encen(Pis+2);
Pis >= 6 ->
    {bot8, node8@maquina }!{light_on,Pis},
  end.
```

```
apaga(Pis)->
  if
Pis < 6 ->
    {bot8, node8@maquina }!{light_off,Pis},
    apaga(Pis+2);
Pis >= 6 ->
    {bot8, node8@maquina }!{light_off,Pis},
  end.
```

a) Engegar cada botonera des del seu node



```

-module(ex3a).
-export([start/0, botonera/0, proxy/3, bucle_proxy/2]).

start() -> register(bot2, spawn(?MODULE, botonera, [])).

botonera()->
    %Iniciar la botonera de 2
    B2 = proxy(2,b2,self()),
    bucle(B2).

```

```

encen(Pis) ->
    if
    Pis < 6 ->
        {bot8, node8@maquina }!{light_on,Pis},
        encen(Pis+2);
    Pis >= 6 ->
        {bot8, node8@maquina }!{light_on,Pis},
    end.

```

```

apaga(Pis)->
    if
    Pis < 6 ->
        {bot8, node8@maquina }!{light_off,Pis},
        apaga(Pis+2);
    Pis >= 6 ->
        {bot8, node8@maquina }!{light_off,Pis},
    end.

```

```

bucle(L)->
    receive
    {b2,clicked,Pis} ->
        If
            (Pis == 0) -> encen( 0);
            (Pis == 1) -> encen( 1)
        end,
        bucle(L);
    {b8,clicked,Pis}->
        If
            (Pis rem 2 == 0) -> apaga(0);
            (Pis rem 2 == 1) -> apaga(1)
        end,
        bucle(L);
    {b8, abort} ->
        L!close,
        receive _ -> "ok" end;
    {b2, abort} ->
        {bot8, node8@maquina}!close,
        receive _ -> "ok" end
    end.

```

```
-module(ex3b).  
-export([start/0, proxy/2,bucle_proxy/1]).
```

```
start() -> spawn(?MODULE, proxy, [8, b8]).
```

```
proxy(N,Nom)->%N es la mida total  
B = spawn(ex3b,bucle_proxy,[Nom]),  
register(bot8,bots:nou(N,B)).
```

```
bucle_proxy(Nom)->  
receive  
  {clicked,Pis} ->  
    {bot2, node2@maquina}!{Nom,clicked,Pis},  
    bucle_proxy(Nom);  
  
  abort ->  
    {bot2, node2@maquina}!{Nom,abort},  
    ok  
end.
```