



Pontificia Universidad
JAVERIANA
Bogotá

SUSTENTACIÓN

PROFESORA: CECILE EUGENIE GLORIA GAUTHIER UMAÑA

PROYECTO 01 **PROLOG**

Luis Bravo
Camilo García
Fabio Buitrago

BASE DE HECHOS

Hechos estáticos



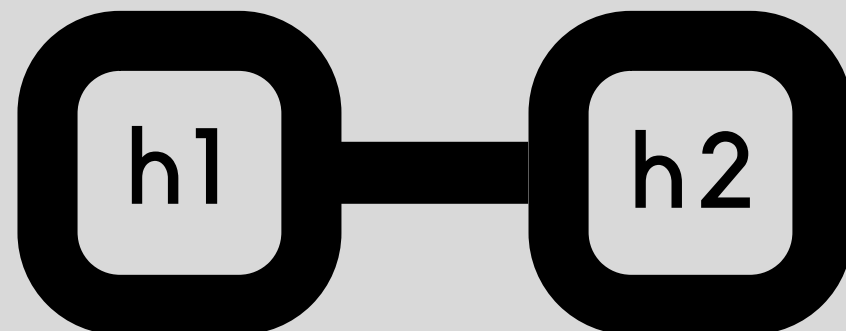
Ubicación inicial del robot

```
ubicacion_inicial(robot, h1).
```



Ubicación inicial cajas

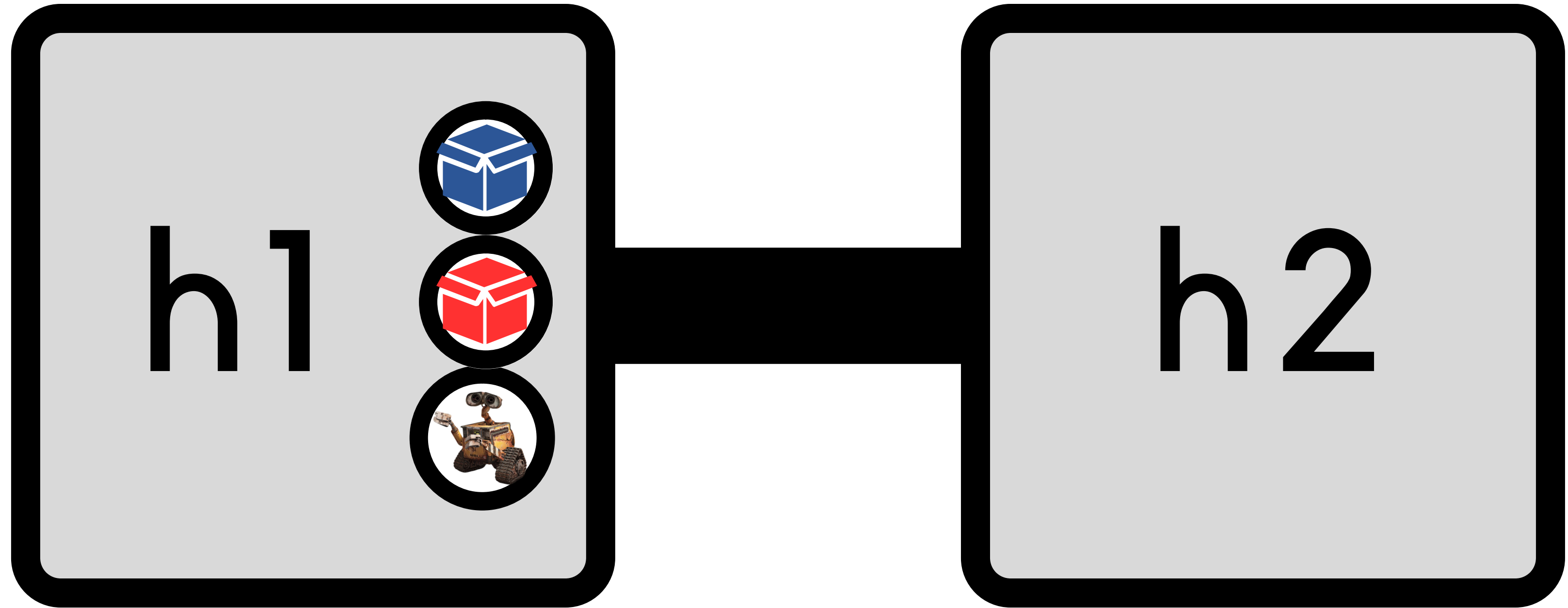
```
ubicacion_inicial(caja(azul), h1).  
ubicacion_inicial(caja(roja), h1).
```



Conexión entre las conexiones h1 y h2

```
habitacion(h1).  
habitacion(h2).
```

```
conexion(h1, h2).  
conexion(h2, h1).
```

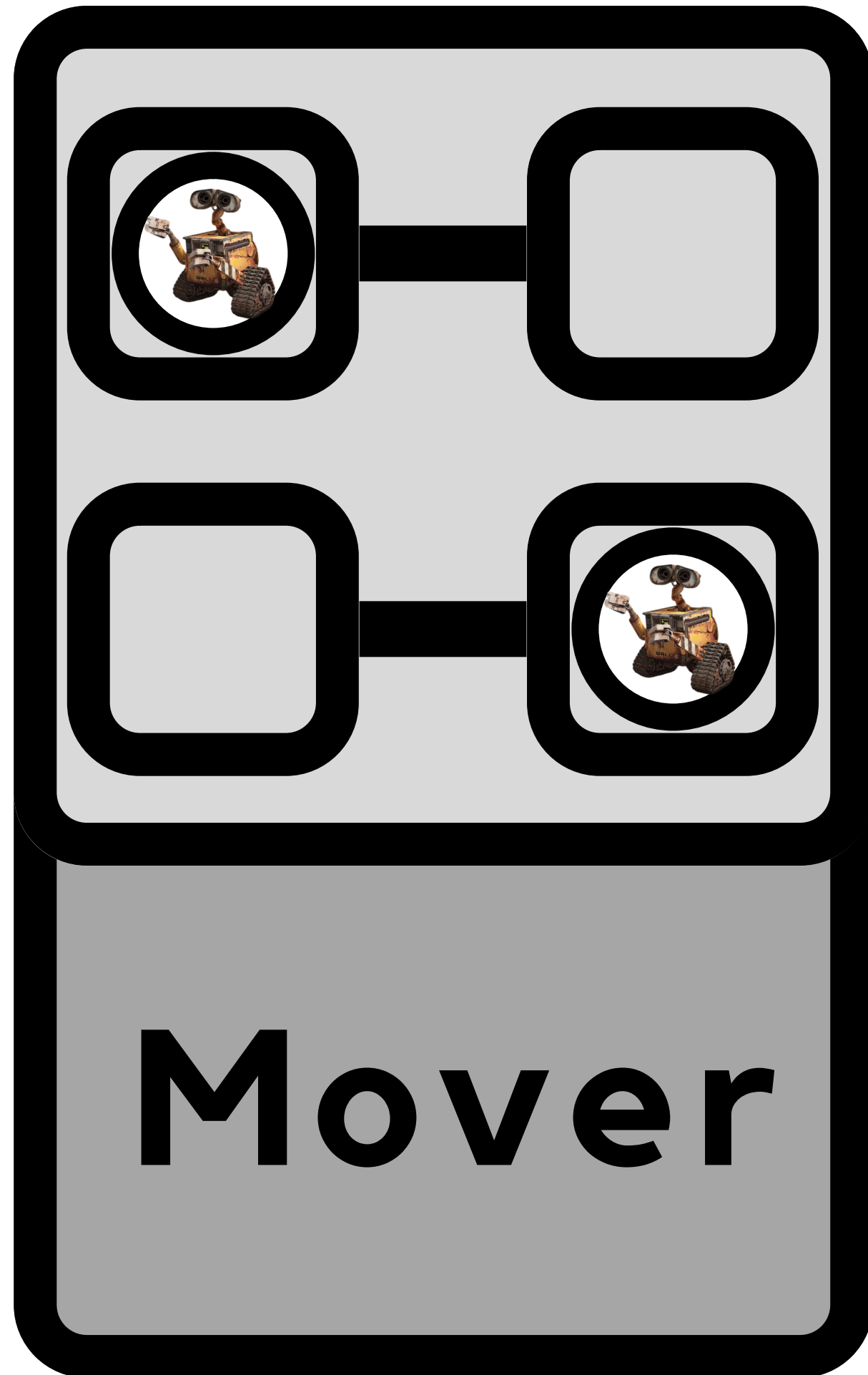


```
:- dynamic ubicacion, ②  
   ubicacion(robot, h1).  
   ubicacion(caja(azul), h1).  
   ubicacion(caja(roja), h1).
```

→ Número de argumentos

} Inicialización

DEFINICIÓN DE REGLAS



```
mover(ObjetoAMover, HabitacionActual,  
HabitacionFinal) :-  
  connexion(HabitacionActual, HabitacionFinal),  
  retract(ubicacion(ObjetoAMover,  
HabitacionActual)),  
  assertz(ubicacion(ObjetoAMover,  
HabitacionFinal)).
```



Recoger

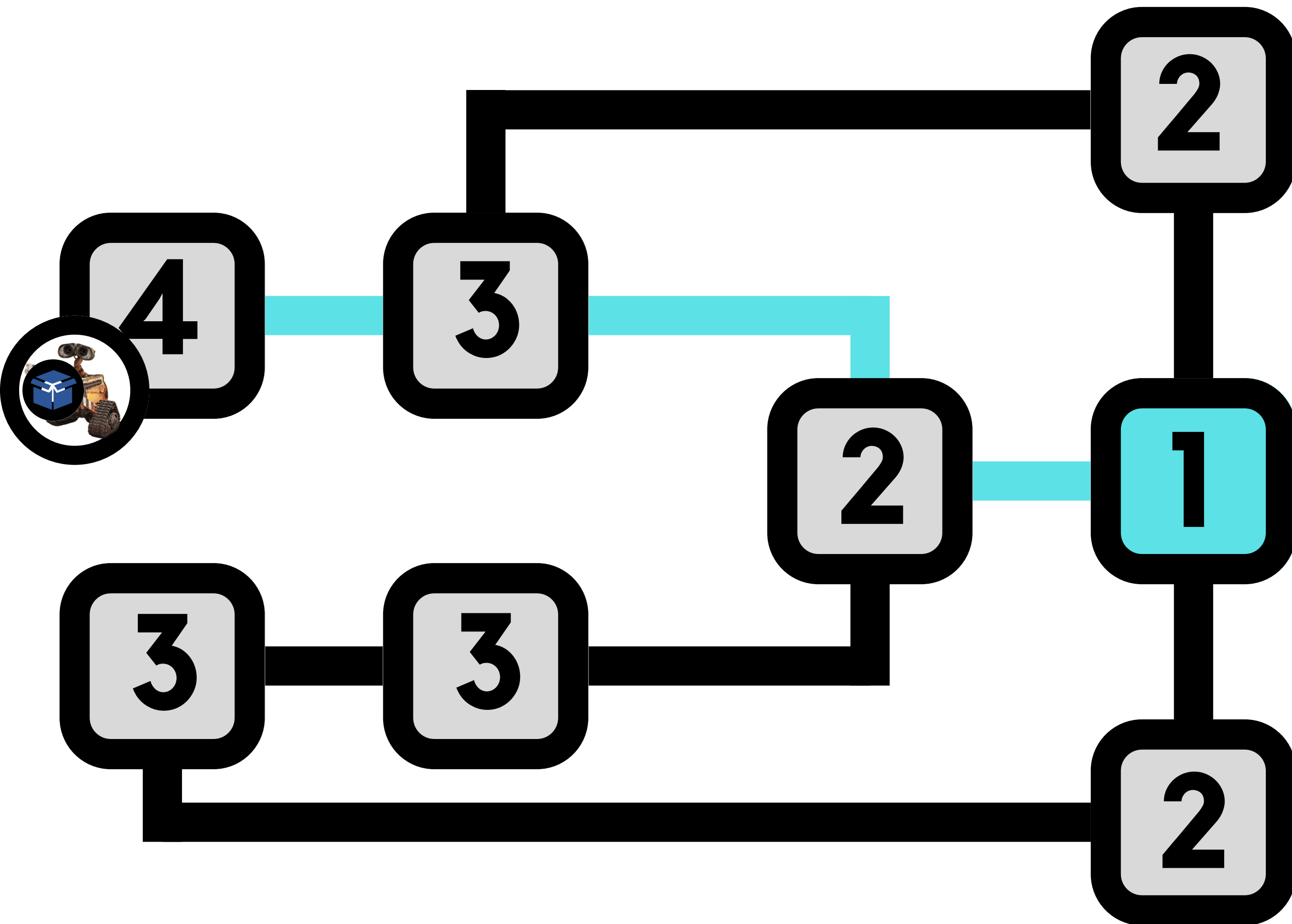
```
recoger(Robot, Caja) :-  
    ubicacion(Caja, UbicacionCaja),  
    ubicacion(Robot, UbicacionCaja),  
    forall((ubicacion(OtraCaja, Robot),  
OtraCaja \= Caja), fail),  
    retract(ubicacion(Caja, UbicacionCaja)),  
    assertz(ubicacion(Caja, Robot)).
```



Soltar

```
soltar(Robot, Caja) :-  
  retract(ubicacion(Caja, Robot)),  
  ubicacion(Robot, UbicacionRobot),  
  assertz(ubicacion(Caja, UbicacionRobot)).
```


HEURÍSTICA



Heurística

Cantidad de habitaciones a recorrer antes de llegar al destino.

Calculado

a partir del algoritmo de Breadth First Search

Generalidades del algoritmo

```
resolver_heuristica :-
```

```
    habitacionDeMinimaHeuristica([h1,h2], h1, Solucion, MinRoom),  
    resolver_heuristica_recursivo(MinRoom, Solucion).
```

```
resolver_heuristica_recursivo(Habitacion, Solucion) :-
```

```
    heuristica(Habitacion, Solucion, Heuristica),
```

```
    (Heuristica == 1 ->
```

```
        writeln("Llegamos al destino! " + Habitacion),  
        true
```

```
    ;
```

```
    writeln("NO Hemos llegado al destino (Actualmente en " + Habitacion + ")..."),
```

```
    conseguirSiguienteHabitacion(Habitacion, SiguienteHab),
```

```
    ubicacion(robot, L),
```

```
    mover(robot, L, SiguienteHab),
```

```
    resolver_heuristica_recursivo(SiguienteHab, SolucionAux)
```

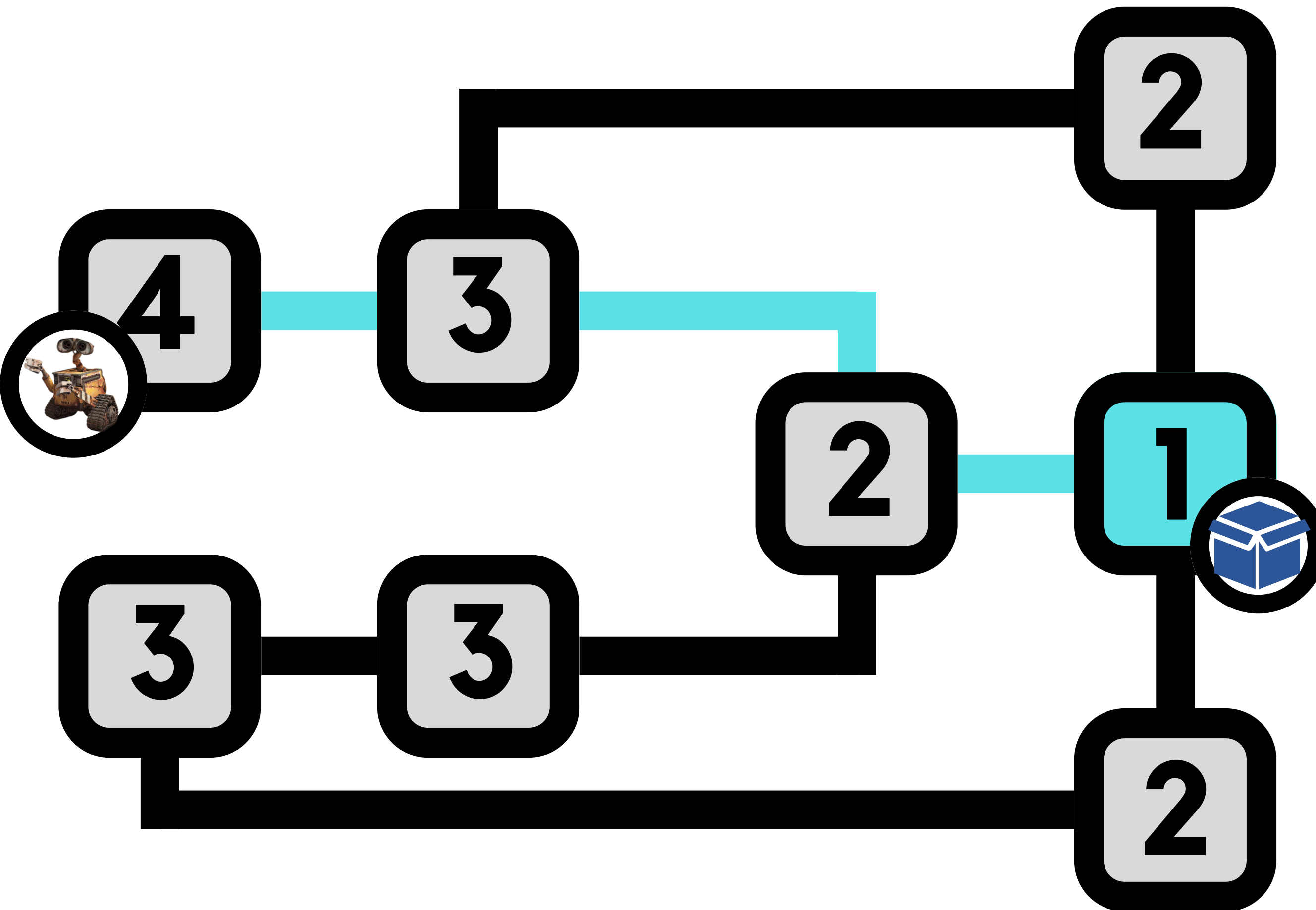
```
).
```

Resolución final del ejercicio

```
resolver :-  
writeln("Luis Bravo, Camilo Garcia, Fabio Buitrago. Proyecto 1 IIA.  
Prolog"),  
writeln("La caja azul se encuentra inicialmente en H1. El robot la  
recoge y se mueve a h2."),  
recoger(robot, caja(azul)),  
resolver_heuristica,  
soltar(robot, caja(azul)),  
ubicacion(caja(azul), L),  
writeln("Tras soltar la caja, estando en h2, la ubicacion de la caja  
azul es " +L).
```

Adicional 01 - Varias habitaciones

```
resolver_heuristica :-  
    habitacionDeMinimaHeuristica([h1,h2,h3,h4,h5,h6,h7], h1, Solucion, MinRoom),  
    resolver_heuristica_recursivo(MinRoom, Solucion).  
  
resolver_heuristica_recursivo(Habitacion, Solucion) :-  
    heuristica(Habitacion, Solucion, Heuristica),  
    (Heuristica == 1 ->  
        writeln("Llegamos al destino! " + Habitacion),  
        true  
    ;  
        writeln("NO Hemos llegado al destino (Actualmente en " + Habitacion + ")..."),  
        conseguirSiguienteHabitacion(Habitacion, SiguienteHab),  
        ubicacion(robot, L),  
        mover(robot, L, SiguienteHab),  
        resolver_heuristica_recursivo(SiguienteHab, SolucionAux)  
    ).
```

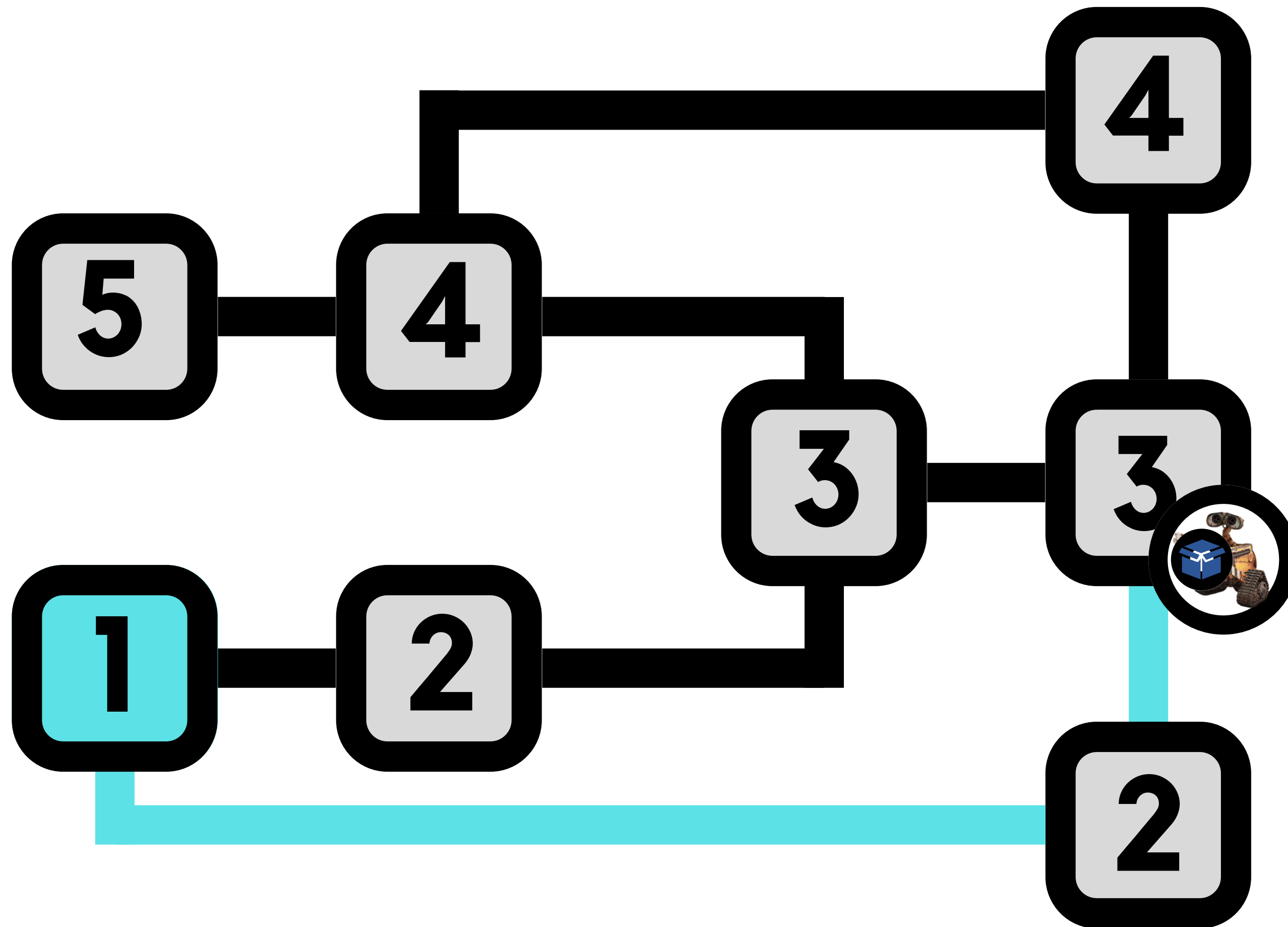


Heurística

Cantidad de habitaciones a recorrer antes de llegar al destino.

Calculado

a partir del algoritmo de Breadth First Search

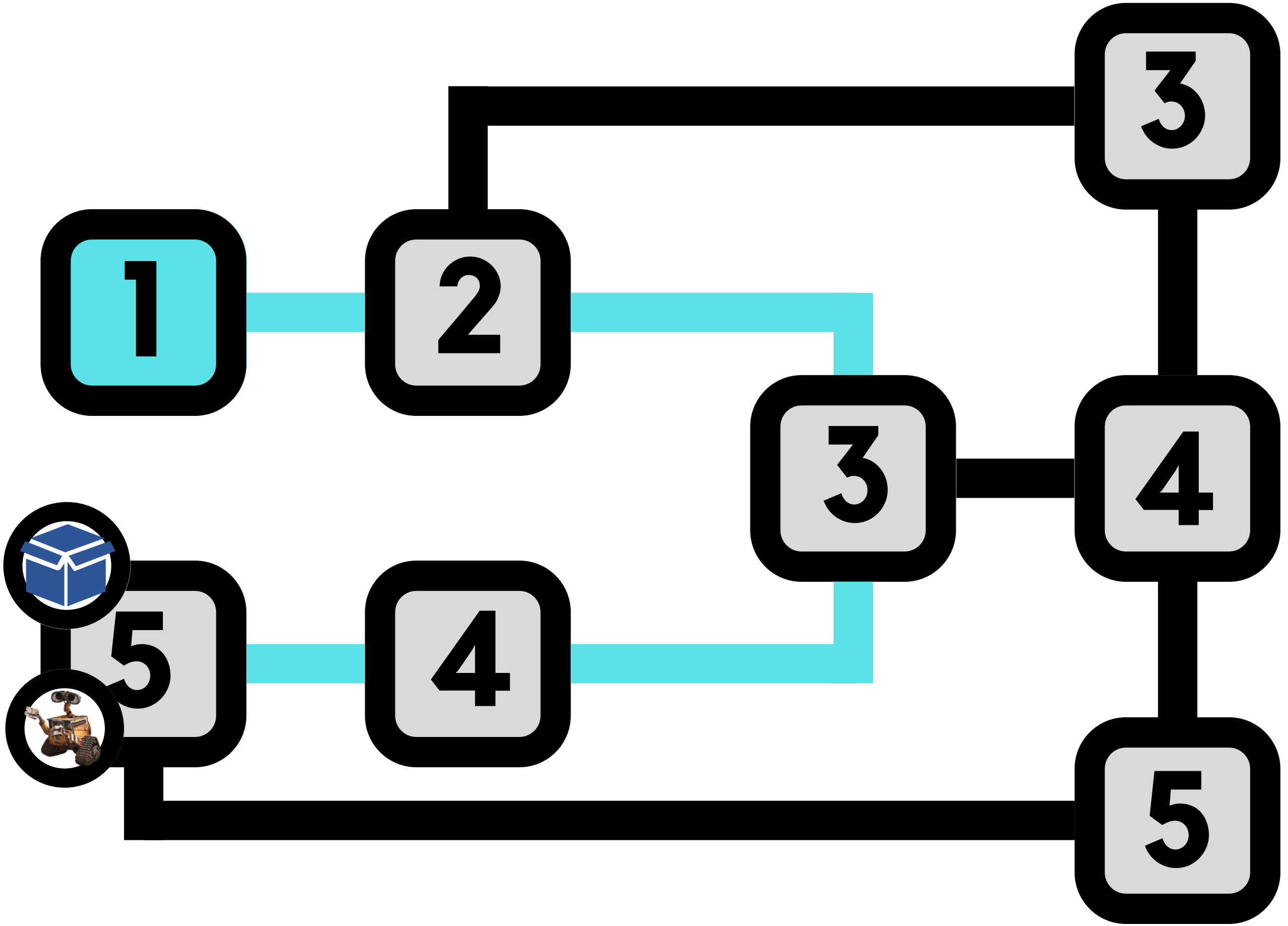


Heurística

Cantidad de habitaciones a recorrer antes de llegar al destino.

Calculado

a partir del algoritmo de Breadth First Search



Heuristica

Cantidad de habitaciones a recorrer antes de llegar al destino.

Calculado

a partir del algoritmo de Breadth First Search

Adicional 02 - Cambio de ubicaciones

```
resolver(HabitacionIRobot,HabitacionICaja,ColorCaja,HabitacionFCaja,HabitacionFRobot):-
writeln("Luis Bravo, Camilo Garcia, Fabio Buitrago. Proyecto 1 IIA. Prolog"),
inicializarPosicion(robot,HabitacionIRobot),
inicializarPosicion(caja(ColorCaja),HabitacionICaja),
ubicacion(robot, M),
writeln("El robot se encuentra inicialmente en " +M),
ubicacion(caja(ColorCaja), L),
writeln("La caja se encuentra inicialmente en " +L),
resolver_heuristica(h1,HabitacionIRobot, HabitacionICaja),
recoger(robot, caja(ColorCaja)),
ubicacion(robot, N),
writeln("El robot se encuentra en " +N),
writeln("Caja recogida"),
resolver_heuristica(HabitacionICaja,HabitacionICaja, HabitacionFCaja),
soltar(robot, caja(ColorCaja)),
ubicacion(caja(ColorCaja), 0),
writeln("La caja se encuentra en " +0),
ubicacion(robot, P),
writeln("El robot se encuentra en " +P),
writeln("Caja soltada"),
resolver_heuristica(HabitacionFCaja,HabitacionFCaja, HabitacionFRobot),
ubicacion(caja(ColorCaja), Q),
writeln("Tras soltar la caja, la ubicacion de la caja es " +Q),
ubicacion(robot, R),
writeln("Además, finalmente, el robot se encuentra en la posicion final " +R).
```