

ENGINYERIA DEL SOFTWARE I

UNIVERSITAT ROVIRA I VIRGILI

PRÀCTICA 2: DESIGN APPROACH FOR REAL TIME SYSTEMS

1 Índex

<u>1 ÍNDEX.....</u>	<u>2</u>
<u>2 ENUNCIAT.....</u>	<u>3</u>
<u>3 DIAGRAMES DE FLUX DE DADES I CONTROL.....</u>	<u>7</u>
<u>4 DIAGRAMA ARQUITECTÒNIC DE TASQUES.....</u>	<u>8</u>
<u>5 DISSENY MODULAR DE CADA TASCA.....</u>	<u>9</u>
<u>.....</u>	<u>12</u>
<u>6 6 DISSENY DETALLAT: PSEUDOCODI.....</u>	<u>14</u>

3

2 Enunciat

ROBOT ASPIRADORA ROOMBA 2 SE

Descripción:

Roomba es una aspiradora automática que barre y aspira los suelos sin ningún tipo de supervisión. Basta con encenderla y marcharse, Roomba se encarga de limpiar el suelo, incluso las zonas de difícil acceso, tales como debajo de las camas, sofás y armarios. Con ROOMBA 2 SE la limpieza doméstica será más sencilla, rápida y eficaz.

Roomba, una limpieza inteligente que funciona!

La aspiradora Roomba realiza una limpieza en tres fases para eliminar el menor rastro de suciedad, dejando a su paso un suelo completamente limpio. Roomba empieza la limpieza de la habitación desplazándose en espiral, utilizando la tecnología de navegación inteligente diseñada por iROBOT. Toda la basura y el polvo se almacenan en un depósito sin bolsa, muy fácil de retirar, vaciar y volver a colocar. Se sirve de algoritmos de inteligencia artificial para abarcar todo el suelo durante el proceso de limpieza automática.

Gracias a su sensor de seguimiento de paredes, Roomba se desliza a lo largo de las paredes y alrededor de los muebles para limpiar toda la superficie del suelo.

No necesita accesorios adicionales, es extremadamente ligera (sólo 2,9 Kg) y muy fácil de transportar y guardar.

Roomba limpia de manera eficaz prácticamente todos los tipos de suelos incluyendo madera, linóleo, baldosas y laminados, así como moquetas de pelo corto y medio. La transición entre las diferentes superficies también es automática.

Roomba incluye una unidad de pared virtual que crea una pared invisible de hasta 4 m con infrarrojos, lo que permite bloquear puertas abiertas y dividir espacios grandes para que Roomba limpie solamente el área que se desee.

Gracias a la navegación inteligente y a un sensor de desnivel incorporado, Roomba detecta automáticamente escaleras y suelos en pendiente [o huecos de escalera], limpia hasta el borde y retrocede para evitar caer.

Porque tienes mejores cosas que hacer.

Roomba ha sido diseñada pensando en el ajetreado estilo de vida actual, para que cualquier suelo esté impecable todos los días y se disponga de más tiempo libre para otras actividades. Mientras Roomba aspira de manera autónoma, los que trabajan pueden disfrutar de su tiempo de ocio. Roomba también es ideal para personas mayores y personas con limitaciones físicas, puesto que limpia a conciencia el suelo, incluso zonas de difícil acceso. Se acabó el ir empujando la aspiradora. Ahora basta con pulsar un botón y Roomba se encarga del resto.

Los niños y mascotas están totalmente seguros con Roomba. De hecho, es perfecta para los dueños de animales de compañía, porque es muy eficaz recogiendo el pelo de las mascotas y deja el ambiente totalmente limpio.

Roomba nueva generación, nuevas características y accesorios.

La nueva línea Roomba incorpora principalmente características inteligentes que permiten una limpieza más autónoma que nunca. Entre ellas:

- Detector de suciedad que activa un sistema de respuesta: sus sensores especiales alertan a Roomba de la presencia de suciedad, para que dedique más tiempo a limpiar las zonas más sucias.
- Estación base: Roomba vuelve automáticamente a su estación de carga cuando termina de limpiar (incluida en el paquete Roomba 2 SE)
- Carga más rápida: Su innovador sistema de carga permite que Roomba 2 se cargue en menos de tres horas, comparado con las 7 de la versión anterior.
- Batería de larga duración: Roomba 2 tiene ahora una autonomía de limpieza de 120 minutos.
- Un único botón para el modo "Limpieza": Roomba 2 calcula el tamaño de la habitación y cuánto tiempo necesita para limpiarla.
- Contenedor de partículas más grande, que puede almacenar el triple de basura.
- Aspiradora más potente para una mejor limpieza de moquetas tupidas.
- Sensores inteligentes que detectan si Roomba se atasca [en su desplazamiento] y le ayudan a liberarse.
- Modificaciones de diseño que permiten una transición más fluida entre suelo y alfombra.
- Funcionamiento más silencioso.

El Roomba 2 SE se suministra con lo siguiente:

Cargador de 3 horas
 2 unidades de pared virtual
 Mando a distancia para el Roomba Home Base
 2 filtros
 Unidad de fijación en pared

Instrucciones en castellano.

Especificaciones técnicas:

- Diámetro: 33,65 cm
- Altura: 7,03 cm
- Peso: 2,9 Kg (con batería)
- Bolsas aspiradora: Sin bolsa (Recipiente partículas = 400cc, Recip. aspiradora = 100cc)
- Autonomía de limpieza: 90 - 120 minutos por carga, según el tipo de suelo

- Flujo de aire: 1.2 m / seg.
- Alimentación: 30 W (aspiradora)
- Nivel de ruido: <70dB
- Algoritmo de limpieza: Sistema de limpieza en 3 fases = 3 cepillos y una aspiradora
- Ancho trayect. Limpieza: 22 cm
- Sistema de navegación: Sensor de suciedad, Sensor de desnivel, Sensor de seguimiento de paredes, Sensor de contacto, Sensor de bloqueo y Sistema de pared virtual.
- Velocidad: 28 cm / seg.
- Área cubierta/ habitación: 99,97%

COMPRE YA!

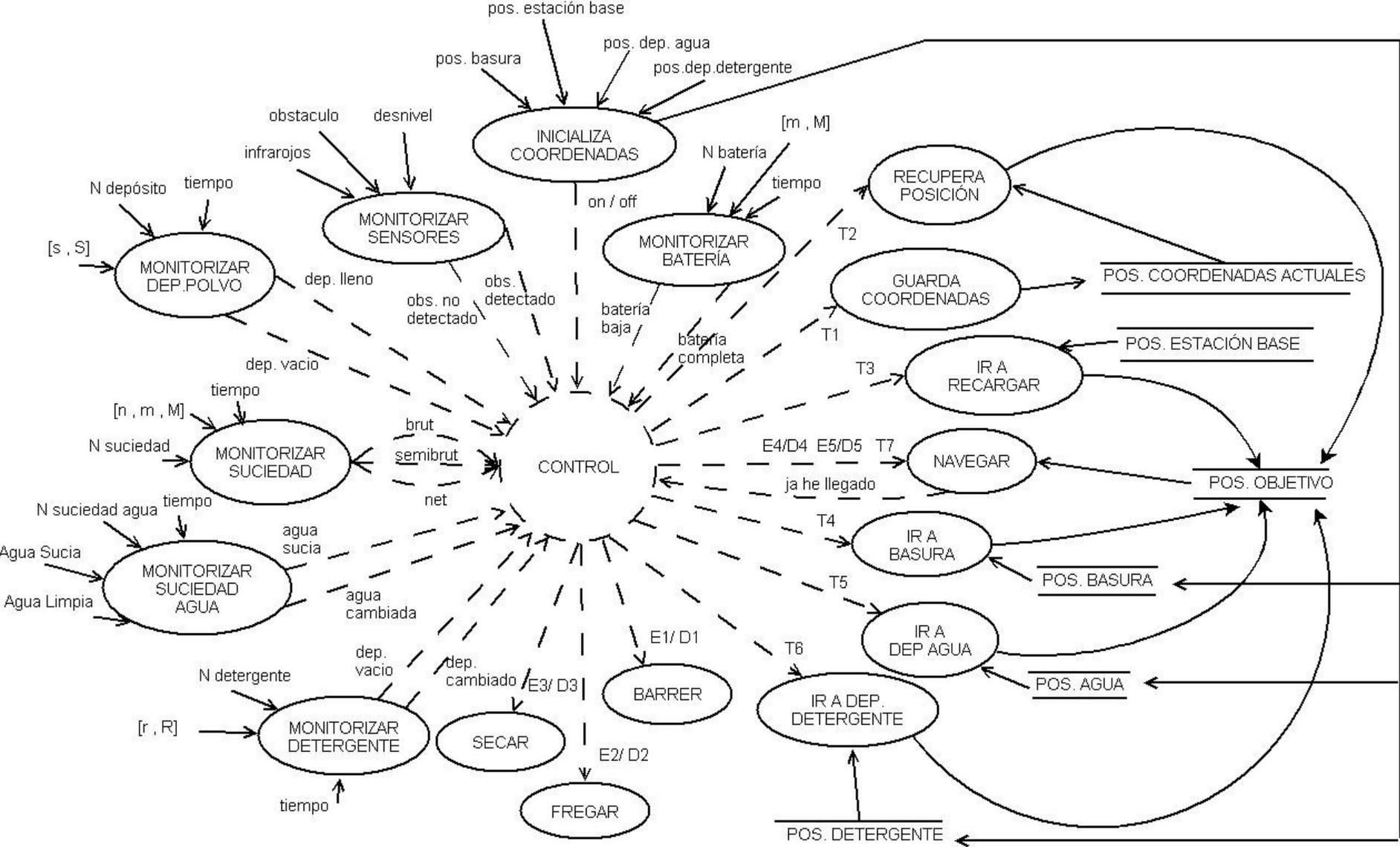
Aquesta és la descripció real de ROOMBA 2SE, el robot netejador de l'empresa iROBOT. Malgrat les bones crítiques rebudes pel producte iROBOT hi ha una sèrie d'aspectes que són millorables però que comporten un anàlisi i disseny nous del sistema de control de robot, encara que hi ha certs elements del disseny actual que poden ser reutilitzats:

aspectes que es mantenen
<ul style="list-style-type: none"> • activació: Roomba s'activa/desactiva amb el botó "Limpieza". • navegació: en el moment en que l'activem, Roomba comença un recorregut en espiral (de dins cap a fora) fent servir la tecnologia de navegació "NAVEGA" proporcionada per iROBOT. Quan qualsevol dels sensors "OBSTACLE", "DESNIVELL" o "INFRARROJOS" s'activa cal informar "NAVEGA" que actua pertinentment sobre el motor de desplaçament que pot rebre cinc tipus de senyals: "endavant", "enrera", "gir-dreta", "gir-esquerra", o "parada". • manteniment bateria: un cop es detecta "bateria-baixa" el robot es guarda les coordenades actuals i s'encamina a les coordenades del dispositiu extern "RECARREGADOR BATERIA" per recarregar-se automàticament, moment que es detecta amb el senyal "recarregat". A continuació el Roomba s'encamina a les coordenades inicialment guardades i procedeix amb les seves tasques.
aspectes que cal millorar (ampliacions del sistema de control)

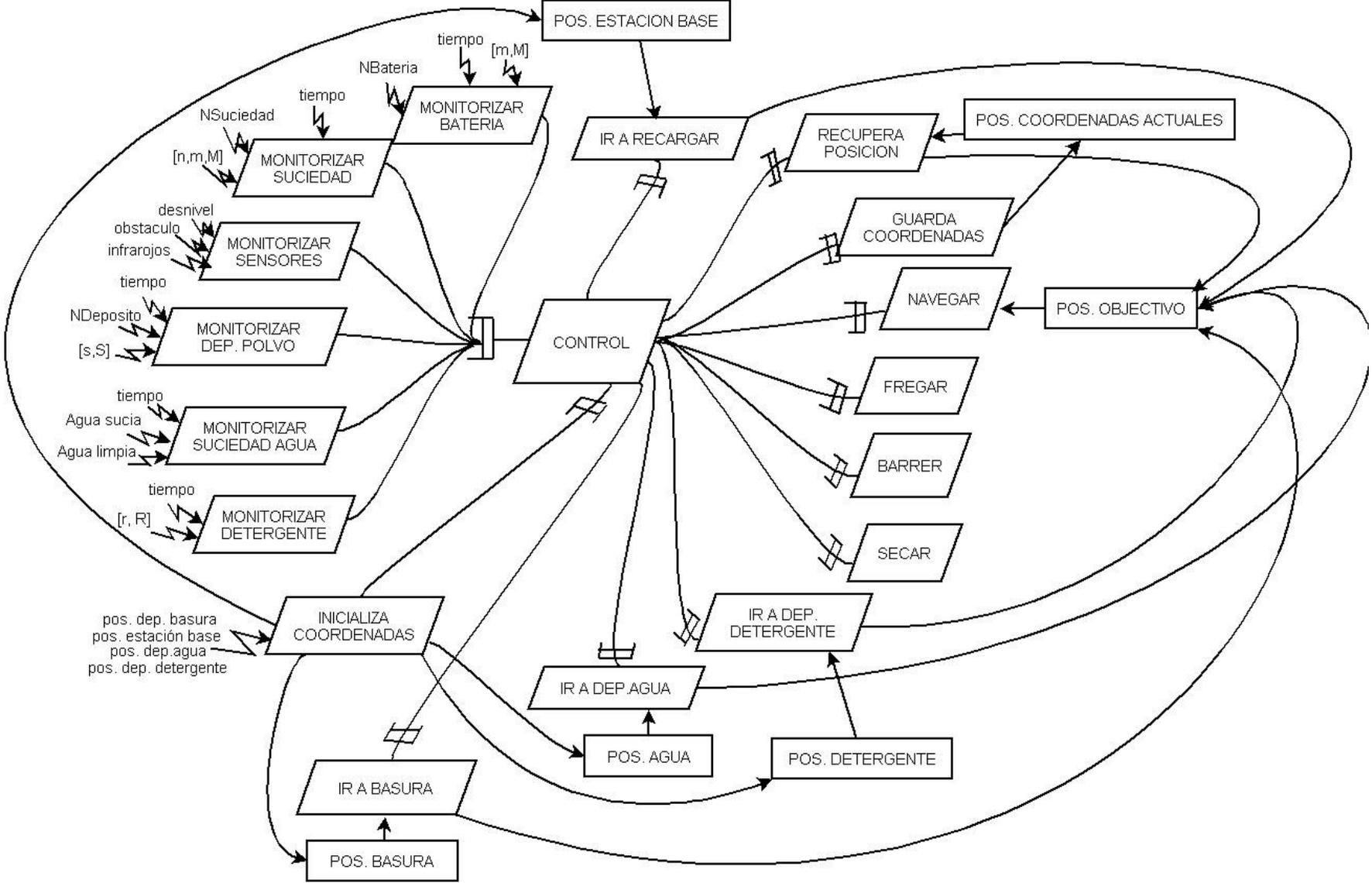
• **neteja**: amplia el sistema d'escombrat en tres fases existent amb un posterior procés doble de fregat i assecat. El sensor/detector de brutícia del terra informa del nivell de netedat sota la superfície del robot (amb valors "brut", "semi-brut" i "net"). Quan està "brut" cal escombrar fins que es detecta un nivell "semi-brut", moment en que es passa a fregar fins assolir nivell "net", en que comença l'assecament que dura, com a molt, 5 segons i que pot ser interromput per nous senyals de "brut", "semi-brut" o "net".

• **manteniment**: Roomba incorpora un procés autònom de manteniment. Així, un cop el "DIPÒSSIT DE POLS" està ple, el robot s'encamina al dispositiu extern "ESCOMBRARIA", el qual el buida i l'informa, un cop buidat, amb el senyal "dipòsit_buidat". El dipòsit d'aigua de fregar disposa d'un detector de brutícia de l'aigua (diferent del de brutícia del terra) que quan informa de "aigua bruta" provoca que el robot s'encamini al dispositiu extern "CANVIADOR AIGUA" que la renova. Un cop canviada l'aigua bruta per aigua neta el Roomba rep el senyal "aigua_canviada". El mateix passa amb el dipòsit de detergent del Roomba "DETERGENT" que, quan es buida fa que vagi al "DISPENSADOR DETERGENT". Penseu que **manteniment** **bateria** ja està desenvolupat i que aquests nous comportaments són estructuralment equivalents.

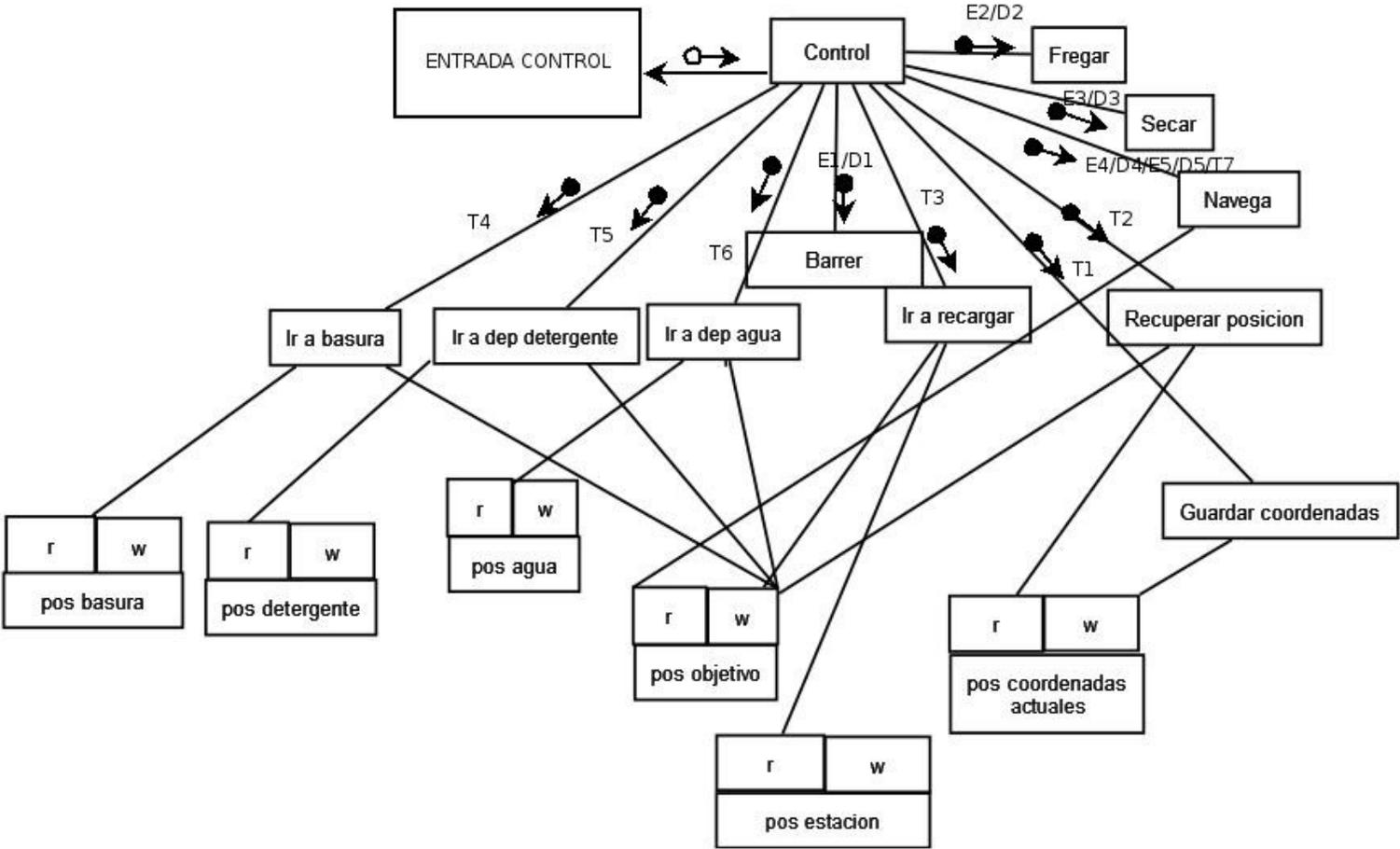
3 Diagramas de Flux de Dades i Control

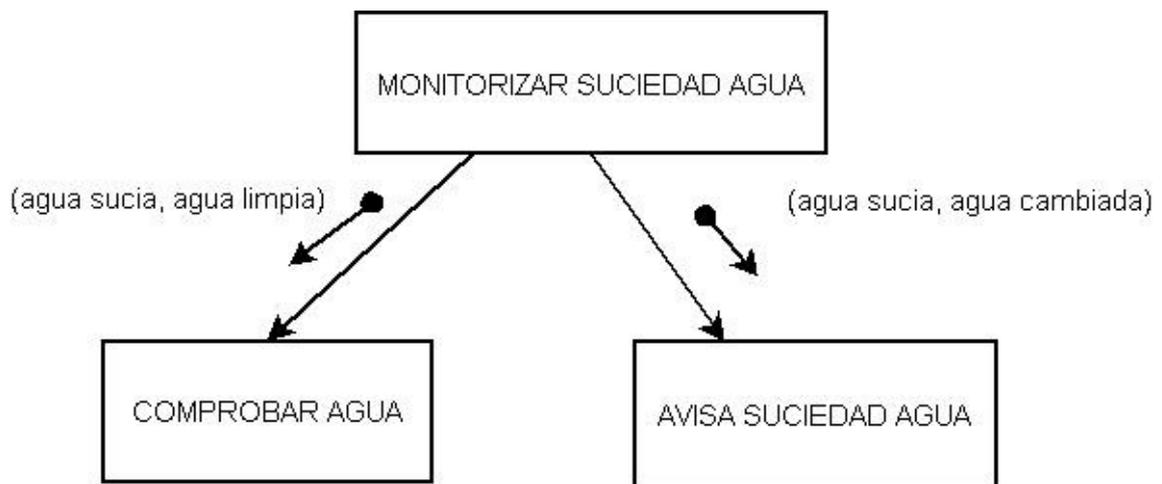
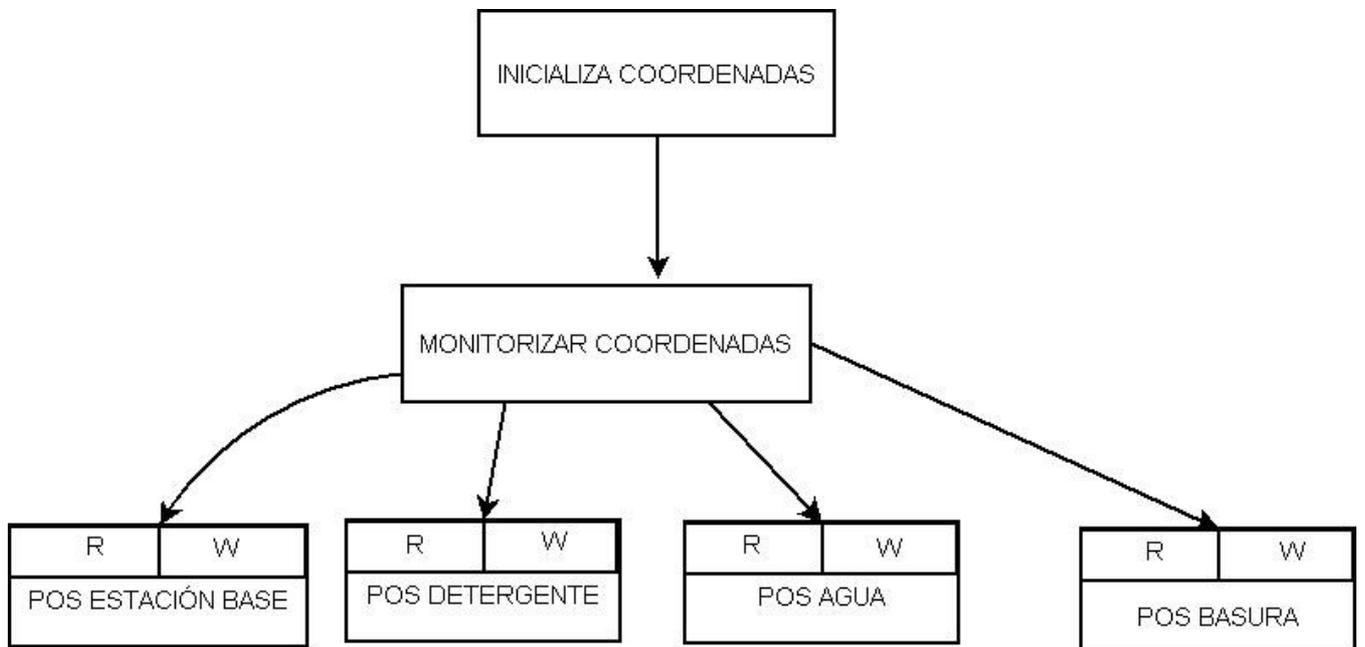


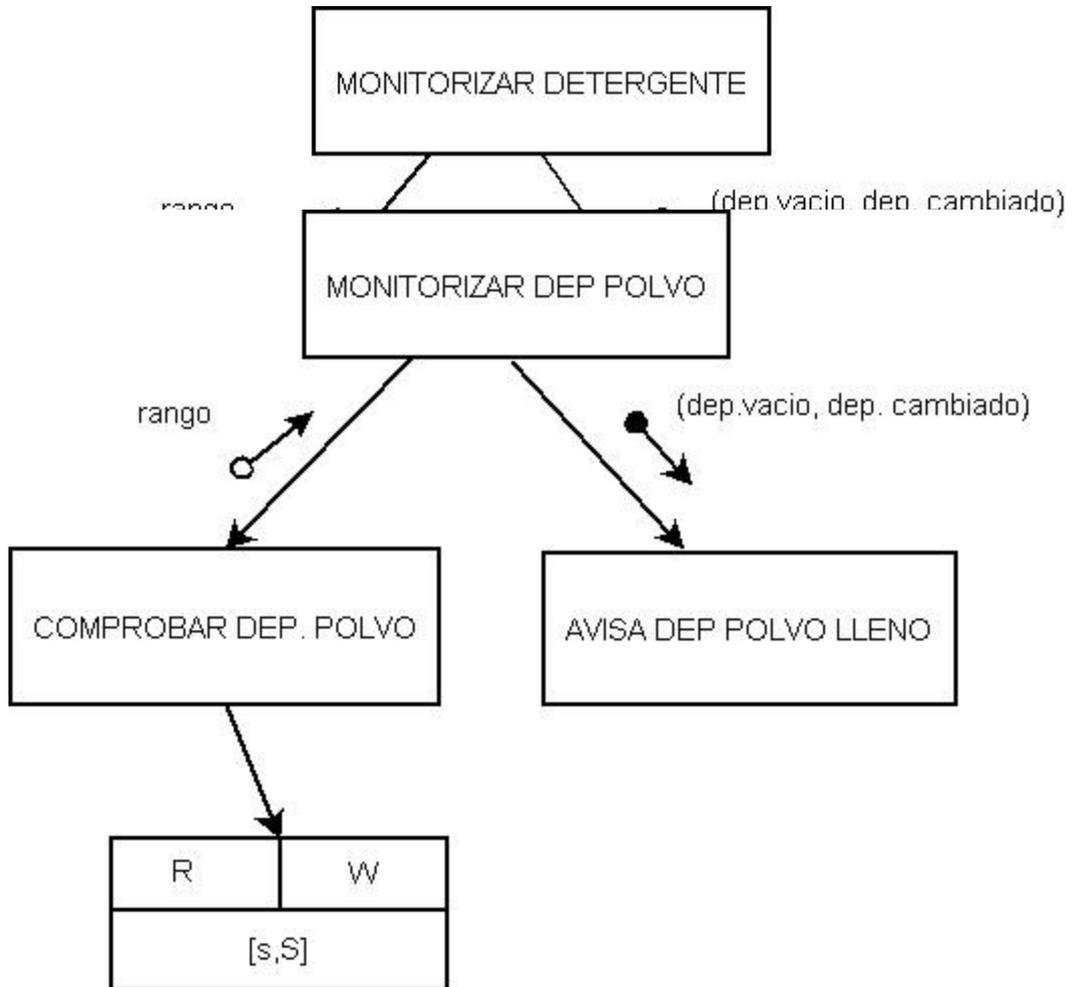
4 Diagrama Arquitectònic de Tasques

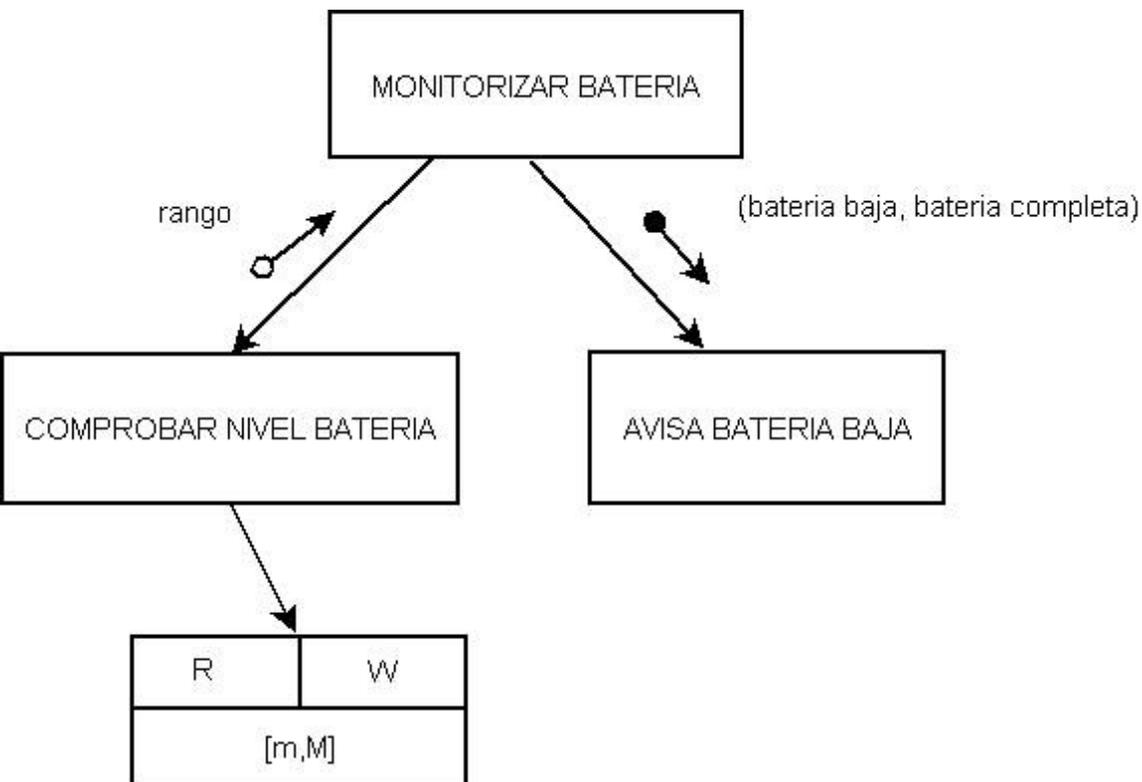
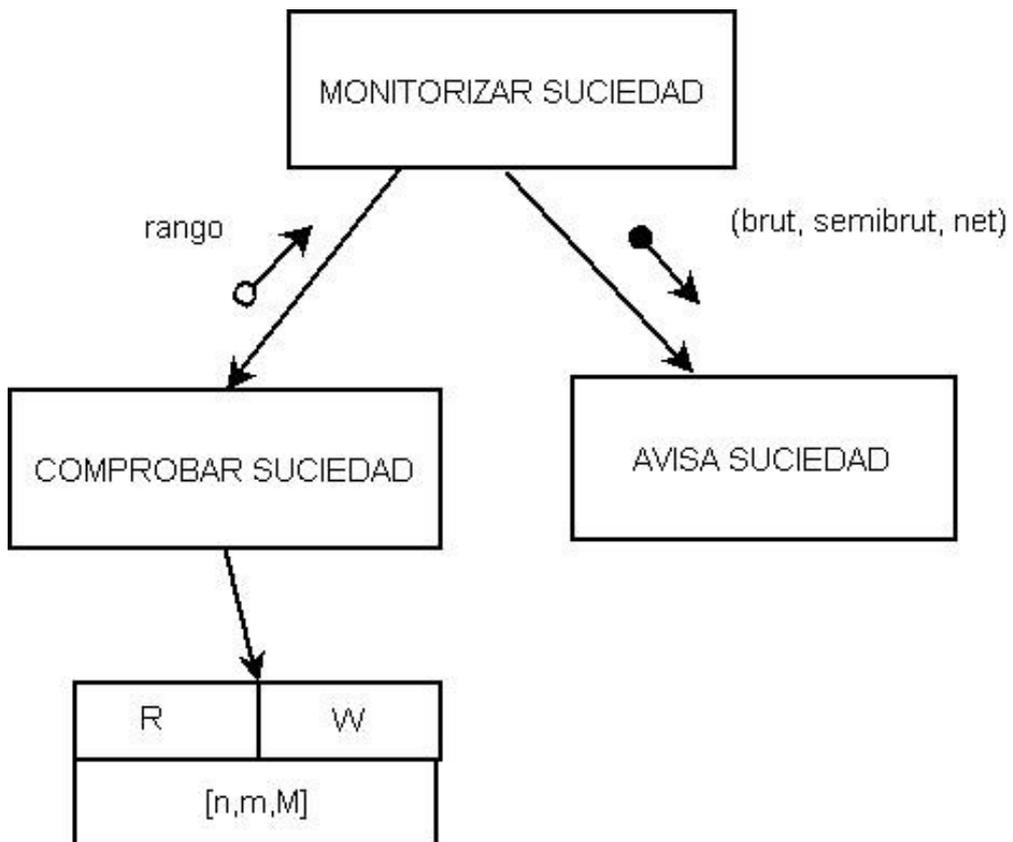


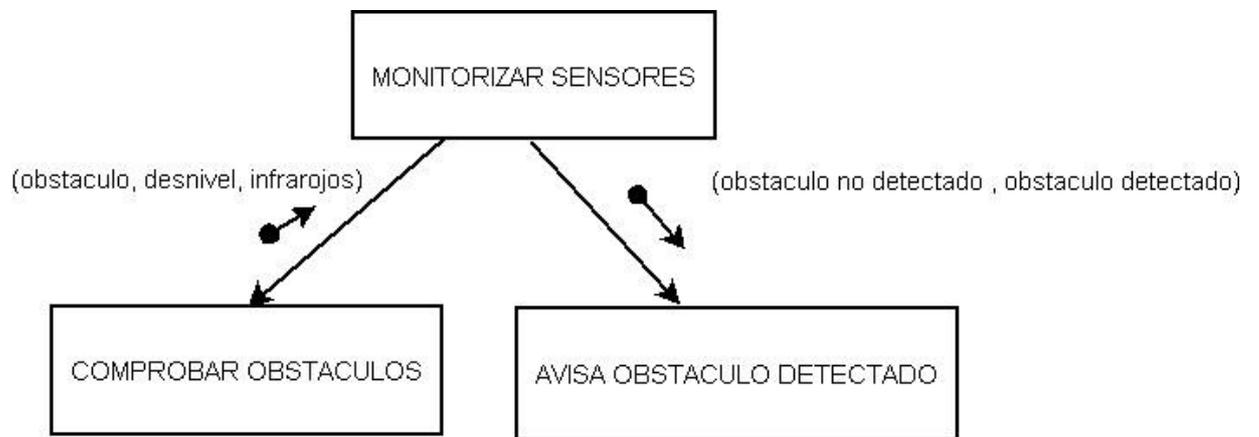
5 Disseny Modular de cada Tasca











```
module sistema is  
uses control, e/s, inicializar;  
begin  
    inicializar.inicializar_coordenadas();  
    repeat  
        control.actua(esdevenimineto);  
    forever  
end sistema;
```

```
nodule control is  
uses ir_a_recargar ,ir_a_basura, ir_a_dep_agua, ir_a_dep_detergente, guarda_co-  
ordenadas, recupera_posicion;  
type entradas = monitor_suciedad, monitor_sensores,  
monitor_deposito,monitor_agua, monitor_detergente, monitor_bateria;
```

```

procedure actua (esdevenimiento: in entradas) do
  Case esdevenimiento do
    dip_ple: ir_a_basura.irbasura(); navega.irposicion();
    dip_vacio: navega.espiral();
    agua_sucia: Ir_a_dep_agua.irdepaua(); navega.irposicion();
    agua_cambiada: navega.espiral();
    bateria_baja: ir_a_recargar.recargar(); navega.irposicion();
    bateria_completa: navega.espiral();
    obstaculo_detectado: navega.esquiva();
    obstaculo_no_detectado: navega.espiral();
    detergente_vacio: Ir_a_dep_detergente.irdepdetergente(); navega.irpo-
    sicion();
    detergente_cambiado: navega.espiral();
  end case;
end actua;
begin
  Actua();
end control;

module MonitorBateria is
uses NIVEL_BATERIA, [m,M];
type nivel_bateria = {bateriaBaja, bateriaAlta};
procedure determina_situación_bateria (esdeveniment: out nivel_bateria) do
  leer(bateria);
  NIVEL_BATERIA.escribir(bateria); [m,M].leer([m,M]);
  if bateria < 5 then
    esdeveniment := batbaja;
  else if bateria == 100 then
    esdeveniment := batalta;
end determina_situación_bateria;
begin
  null;
end MonitorBateria;
module MonitorSuciedad is
uses NIVEL_SUCIEDAD, [n, m, M];
type nivel_suciedad = {brut, semibrut, net};
procedure determina_situación_suciedad (esdeveniment: out nivel_suciedad) do
  leer(suciedad);
  NIVEL_SUCIEDAD.escriure(suciedad);
  [m,M].leer([n, m, M]);
  If suciedad < n then
    esdeveniment := brut;
  else if suciedad > n and suciedad < m then
    esdeveniment := semibrut;
  else if suciedad > m and suciedad < M then
    esdeveniment := net;
end determina_situación_suciedad;
begin
  null;

```

end MonitorSuciedad;

module MonitorSensors **is**

uses DESNIVEL, OBSTACULO, INFRAROJOS;

type obstaculo = {obs.detectado, obs.no_detectado};

procedure determina_obstaculos (esdeveniment: **out** obstaculo) **do**

 leer(desnivel);

 DESNIVEL.escribir(desnivel);

 leer(obstaculo);

 DESNIVEL.escribir(obstaculo);

 leer(infrarojos);

 DESNIVEL.escribir(infrarojos);

if desnivel **or** obstaculo **or** infrarojos **then**

 esdeveniment := obs.detectado;

else

 esdeveniment := obs.no_detectado;

end determina_obstaculos;

begin

 null;

end MonitorSensors;

module MonitorDepositoPolvo **is**

uses NIVEL_DEPOSITO, [s, S];

type nivel_deposito = {dep_lleno, dep_vacio};

procedure determina_situación_deposito (esdeveniment: **out** nivel_deposito) **do**

 leer(deposito);

 NIVEL_DEPOSITO.escribir(deposito);

 [s,S].leer([s,S]);

if deposito < s **then**

 esdeveniment := dep_vacio;

else if deposito == S **then**

 esdeveniment := dep_lleno;

end determina_situación_deposito;

begin

 null;

end MonitorDepositoPolvo;

module MonitorSuciedadAgua **is**

uses SENSORSUCIEDAD, [t, T];

type nivel_agua = {agua_sucia, agua_cambiada};

procedure determina_situación_agua (esdeveniment: **out** nivel_agua) **do**

 leer(agua);

 SENSORSUCIEDAD.escribir(agua);

 [t, T].leer([t, T]);

if agua < t **then**

 esdeveniment := agua_cambiada;

else if deposito == T **then**

 esdeveniment := agua_sucia;

```
end determina_situación_agua;  
begin  
    null;  
end MonitorSuciedadAgua;
```

```
module MonitorDetergente is  
uses NIVEL_DETERGENTE, [r, R];  
type nivel_detergente = {dep_vacio, dep_cambiado};  
procedure determina_detergente(esdeveniment: out nivel_detergente) do  
    leer(detergente);  
    NIVEL_DETERGENTE.escribir(detergente);  
    [r,R].leer([r,R]);  
    if detergente < r then  
        esdeveniment := dep_vacio;  
    else if detergente == R then  
        esdeveniment := dep_cambiado;  
end determina_detergente;  
begin  
    null;  
end MonitorDetergente;
```

```
module Ir_a_Recargar is  
uses pos_estacion;  
uses pos_objetivo;  
procedure recargar() do  
    pos_objetivo.escribir(pos_estacion.leer());  
end recargar;  
begin  
    null;  
end Ir_a_Recargar;
```

```
module Ir_a_basura is  
uses pos_basura;  
uses pos_objetivo;  
procedure irbasura do  
    pos_objetivo.escribir(pos_basura.leer());  
end irbasura;  
begin  
    null;  
end Ir_a_basura;
```

```
module Ir_a_dep_agua is  
uses pos_dep_agua;  
uses pos_objetivo;  
procedure Irdepagua do  
    pos_objetivo.escribir(pos_dep_agua.leer());  
end irdepagua;
```

```
begin  
    null;  
end lr_a_dep_agua;
```

```
module lr_a_dep_detergente is  
uses pos_dep_detergente;  
uses pos_objetivo;  
procedure irdepdetergente do  
    pos_objetivo.escribir(pos_dep_detergente.leer());  
end irdepdetergente;  
begin  
    null;  
end lr_a_dep_detergente;
```

```
module guarda_coordenadas is  
uses pos_coordenadas_actuales;  
uses navega;  
procedure gcoordenadas do  
    pos_coordenadas_actuales.escribir(navega.coordenadas_actuales());  
end gcoordenadas;  
begin  
    null;  
end guarda_coordenadas;
```

```
module recupera_posicion is  
uses pos_coordenadas_actuales;  
uses pos_objetivo;  
procedure recuperap do  
    pos_objetivo.escribir(pos_coordenadas_actuales.leer());  
end recuperap  
begin  
    null;  
end recupera_posicion;
```

```
module pos_estacion  
    var PO: coordenada;  
procedure escribir (t: in coordenada) do  
    PO := t;  
end escribir;  
procedure leer (t: out coordenada) do  
    t := PO;  
end leer;  
begin  
    null;  
end pos_estacion;
```

```
module pos_objetivo
```

```

    var PO: coordenada;
procedure escribir (t: in coordenada) do
    PO := t;
end escribir;
procedure leer (t: out coordenada) do
    t := PO;
end leer;
begin
    null;
end pos_objetivo;

```

```

module pos_detergente
    var PO: coordenada;
procedure escribir (t: in coordenada) do
    PO := t;
end escribir;
procedure leer (t: out coordenada) do
    t := PO;
end leer;
begin
    null;
end pos_detergente;

```

```

module pos_coordenadas_actuales
    var PO: coordenada;
procedure escribir (t: in coordenada) do
    PO := t;
end escribir;
procedure leer (t: out coordenada) do
    t := PO;
end leer;
begin
    null;
end pos_coordenadas_actuales;

```

```

module pos_basura
    var PO: coordenada;
procedure escribir (t: in coordenada) do
    PO := t;
end escribir;
procedure leer (t: out coordenada) do
    t := PO;
end leer;
begin
    null;
end pos_basura;

```

```

module pos_agua
    var PO: coordenada;
procedure escribir (t: in coordenada) do
    PO := t;
end escribir;
procedure leer (t: out coordenada) do
    t := PO;
end leer;
begin
    null;
end pos_agua;

module E/S is
uses monitor_suciedad, monitor_Sensores, monitor_dispositivos, monitor_agua, moni-
tor_detergente, monitor_bateria;

procedure monitorsuciedad (esdeveniment: out monitor_suciedad.suciedad()) do
    monitor_suciedad.determina_situacion_suciedad(esdevenimiento);
end monitorsuciedad;

procedure monitorsensores (esdeveniment: out monitor_sensores.obstaculo()) do
    monitor_sensores.determina_situacion_obstaculos(esdevenimiento);
end monitorsensores;

procedure monitoragua (esdeveniment: out monitor_agua.vacio ()) do
    monitor_agua.determina_situacion_dep_agua(esdevenimiento);
end monitoragua;

procedure monitordetergente (esdeveniment: out monitor_detergente.vacio ()) do
    monitor_detergente.determina_situacion_dep_detergente(esdevenimiento);
end monitordetergente;

procedure monitorbateria (esdeveniment: out monitor_bateria.baja ()) do
    monitor_bateria.determina_situacion_bateria(esdevenimiento);
end monitorbateria;

end E/s;

```