

# Ingeniería del Conocimiento

---

SBC para el control de mascotas en una  
vivienda

---

Curso 2018-2019  
Tercer Curso del Grado en Ingeniería  
Informática

Laura Rabadán Ortega 79088745W  
laurarabadan@correo.ugr.es

# Contenido

- 1. Introducción ..... 3
- 2. Resumen de funcionamiento ..... 4
- 3. Descripción del proceso ..... 5
- 4. Descripción del Sistema ..... 6
- 5. Manual de uso ..... 20
- 6. Referencias..... 21

# 1. INTRODUCCIÓN

El problema que se quiere abordar con este SBC es el control de una o varias mascotas en una casa, con diferentes funciones si están o no solas.

Se ha creado un SBC para una casa en la que vive una persona con su perro. La persona trabaja fuera de la casa de 8h a 13h y de 16h a 20h de lunes a viernes. Los sábados pasa de 10h a 13h trabajando en el despacho de la casa. La tarea del SBC consiste en detectar y avisar si ocurre alguna de las siguientes situaciones:

1. La mascota no tiene agua.
2. La mascota ha entrado en la cocina/despacho.
3. La mascota ha orinado en el pasillo.
4. La mascota ha accedido al cubo de la comida.
5. La temperatura dentro de la casa es peligrosa.

Además, el sistema actuará cuando la mascota se encuentre sola y haya poca luz.

Para poder cumplir con la funcionalidad del sistema, se tienen los siguientes sensores:

- Sensor de movimiento: en todas las habitaciones de la casa.
- Sensor magnético: puerta entre la entrada y el jardín, puerta de la cocina y puerta del despacho.
- Sensor de luminosidad: en el salón.
- Sensor de humedad: en el pasillo.
- Sensor de temperatura: en el salón.
- Sensor de agua: en el bebedero del perro, colocado en el patio interior.
- Enchufe inteligente: en el salón.
- Detector de vibración: en el cubo de la comida del perro, ubicado en la despensa (cocina).

El mapa de la casa utilizada para el desarrollo del SBC es el mostrado en la *Ilustración 1*.

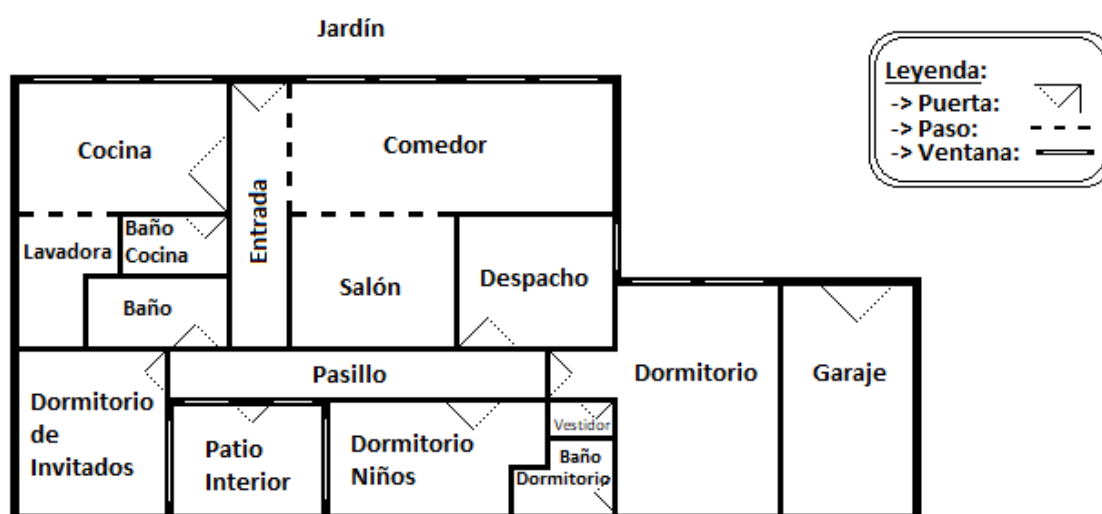


Ilustración 1 - Plano de la casa

## 2. RESUMEN DE FUNCIONAMIENTO

---

El SBC responderá de manera distinta en función de si la persona se encuentra o no sola, tanto en la casa como en la habitación, y de factores externos, como puede ser la temperatura.

### La mascota no tiene agua

---

Si la mascota se queda sin agua, siempre se mandará un aviso a la persona. El aviso será más importante si la temperatura es peligrosa y el perro no tiene agua. Además, si pasa una hora desde que se mandó el primer aviso, se aumentará la importancia del mensaje, ya que puede ser una situación de peligro.

### La mascota ha entrado en la cocina/despacho

---

Si la mascota entra en una de estas zonas, y no está el dueño en esa habitación, se mandará un aviso a la persona de que el perro ha entrado en una zona marcada como zona restringida.

### La mascota ha orinado en el pasillo

---

Si se detecta una humedad alta en el pasillo, se estudiará si es responsabilidad de la persona, porque, por ejemplo, ha fregado el pasillo, o es cosa del perro. En función de la causa, se mandará o no un aviso al dueño.

### La mascota ha accedido al cubo de la comida.

---

Si se registra una activación del detector de vibración y la mascota está sola en la cocina, se envía un aviso al dueño.

### La temperatura de la casa es peligrosa.

---

Se determinan 3 niveles distintos en función a la temperatura registrada: temperatura ligeramente alta o baja, temperatura alta o baja y temperatura muy alta o baja. Los avisos irán en función del nivel de la temperatura.

### Poca luz

---

Si no hay luz suficiente en la casa y la mascota está sola, se encenderá una de las lámparas del salón. Si hay mucha luz y está encendida, se apagará.

### 3. DESCRIPCIÓN DEL PROCESO

---

Antes de implementar el sistema, se estudiaron las tareas que tenía que satisfacer y los medios con los que se contaba para ello. A partir de la información, se decidió la estructura modular del sistema, intentando que cada módulo estuviera especializado, pero evitando que tuviera un funcionamiento básico. Es por esta razón que hay un módulo que recoge 3 funcionalidades.

Para comprobar que el sistema cumplía las tareas impuestas, se implementaron los módulos por separado, comprobando que cada uno cumplía con la tarea para la que se diseñó de manera correcta. De esta forma, resultó más sencillo verificar el SBC.

Una vez los módulos funcionaban de manera correcta por separado, se probó el sistema a nivel global, para corroborar que los módulos se activaban y desactivaban cuando les correspondía.

Además, se verificó que las reglas de los módulos carecieran de inconsistencias, tanto dentro del módulo como a nivel global.

## 4. DESCRIPCIÓN DEL SISTEMA

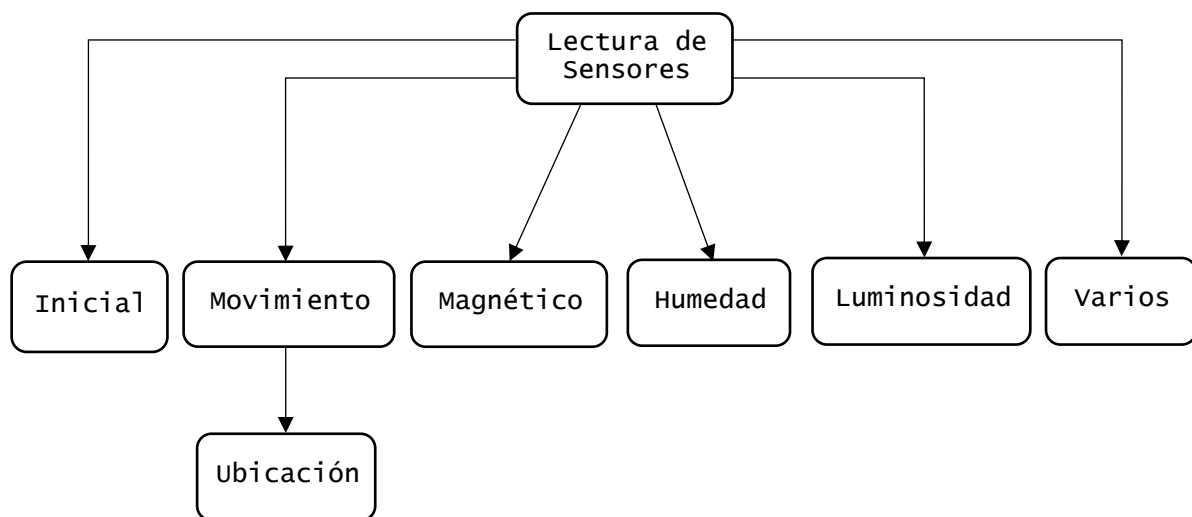
---

Para mayor comprensión del sistema, se ha dividido el SBC en distintos ficheros. Se utiliza una estructura de módulos para el sistema principal y, a parte:

- Dos ficheros con el conocimiento inicial.
- Un fichero para representar la casa en función del conocimiento inicial de la casa.
- Un fichero para el control del tiempo.
- Un fichero para la ejecución del sistema.

Los tres primeros ficheros aportan el conocimiento global del sistema, el fichero del control del tiempo se ejecuta siempre en paralelo al resto del sistema y el último fichero se encarga de ejecutar el sistema, por lo que solo se utiliza al principio de la ejecución del sistema.

Para la estructura de módulos se cuenta con 8 módulos distintos. Uno principal es el que se encarga de leer los datos y llamar a los distintos módulos. La jerarquía es la presentada en la *Ilustración 2*.



*Ilustración 2 - Estructura de Módulos*

‘*Varios*’ representa al conjunto de tareas que les corresponden a los sensores de agua, vibración y temperatura.

Para explicar el SBC se va a describir cada fichero por separado, explicando las tareas o reglas que tienen descritas, para separar las explicaciones por funcionalidades.

Es el batch encargado de cargar todos los ficheros necesarios para utilizar el sistema y de dejarlo preparado para empezar a ser utilizado por el usuario.

Este fichero recoge el conjunto de reglas encargadas del avance del tiempo. Funciona “*de forma paralela*” a la estructura de módulos, es decir, no es llamada por ningún otro módulo ni pertenece a la estructura.

El tiempo se modifica cuando al sistema no le quedan hechos que ejecutar, aumentando el tiempo en un minuto.

Se tiene un hecho, (*hora\_inicio*), que aumenta en 86400 (segundos de un día) cada vez que se cambia de día, para que el sistema funcione de manera correcta.

La hora se controla con un hecho, (*hora ?hora ?minuto ?segundo*), que indica la hora del día actual del sistema.

El día de la semana se controla con el hecho (*semana ?dia*), en el que *?dia* es un valor numérico que hace referencia a un día de la semana (Lunes = 1, Martes = 2, ..., Sábado = 6, Domingo = 7).

Las reglas que se recogen en este fichero son:

- *InicializarSemana*: esta regla pide al usuario el día en el que se inicia el sistema, si no hay un hecho (*semana ?dia*) en la base de hechos.
- *InicializarHora*: similar a la regla anterior pero con la hora. Se le pide al usuario que introduzca la hora y el minuto en el que se inicia el sistema.
- *CambioDia*: se encarga de que, cuando sean las 00:00, se cambie de día de la semana.
- *AjustarDia*: en caso de que se cambie de domingo a lunes, se debe ajustar el identificador para que se mantenga en el intervalo [1,7].
- *CambioHora*: cuando no queden reglas que puedan activarse, se aumenta la hora del sistema un minuto.
- *AjusteHora\_segundos*: cuando se cambia la hora del sistema, se comprueba que los segundos estén dentro del intervalo [0,60[. Si no es así, se ajustan, aumentando el número de minutos correspondientes.
- *AjusteHora\_minutos*: si al ajustar los segundos el valor de los minutos se sale del intervalo [0,60[, se ajusta, aumentando el número de horas.
- *AjusteHora\_horas*: al igual que pasaba con los minutos y los segundos, si al actualizar la hora el valor de las horas se sale del rango [0,24[, se ajusta y se llama a la regla *CambioDia*.

Este fichero contiene el conocimiento global de la casa. En él, se declaran los hechos necesarios para poder representar la casa y sus partes. También, se especifica las habitaciones en las que hay instalado algún dispositivo y el tipo de dispositivo que es.

- *Habitaciones*: se declaran las distintas habitaciones de la casa, indicados en el plano de la página 3 (*Ilustración 1*).

(habitación ?habitacion)

- *Puertas*: habitaciones que están conectadas por medio de una puerta.

(puerta ?habitacion1 ?habitacion2)

- *Pasos*: habitaciones conectadas por un paso sin puerta.

(paso ?habitacion1 ?habitacion2)

- *Ventanas*: habitaciones que tienen ventanas al exterior.

(ventana ?habitacion)

- *SensoresMovimiento*: habitaciones que tienen instalado un sensor de movimiento. En este caso, todas las habitaciones cuentan con uno.

- *SensoresMagneticos*: habitaciones cuyas puertas, todas las que tenga, tiene instalado un sensor magnético. Se instala en la entrada, cocina y despacho.

- *SensoresHumedad*: habitaciones con un dispositivo para el control de la humedad del suelo. Solo está en el pasillo.

- *SensoresLuminosidad*: habitaciones con control de luminosidad. Solo está en el salón.

- *SensoresTemperatura*: habitaciones con sensor para el control de la temperatura. Solo hay en el salón.

- *SensoresAgua*: este dispositivo no se instala en una habitación, sino en el cuenco del agua del perro. Pero, para poder utilizar el mismo tipo de envío de datos para todos los sensores, se indica la habitación en la que se encuentra el cuenco. En principio se supone que el cuenco no se va a cambiar de habitación. Si se cambiara, habría que cambiar la ubicación del cuenco o añadir una regla que determinara la ubicación del cuenco en función a la ubicación de la persona cuando el sensor cambia de on a off.

Esta funcionalidad no se implementó, sino que se supone que el cuenco no va a cambiar.

- *SensoresVibracion*: el dispositivo se encuentra en el cubo de la comida de la mascota, que está en la despensa. Pasa lo mismo que con el sensor de agua: para poder utilizar el mismo formato de envío de valores, se le indica la habitación en la que se encuentra la despensa, es decir, la cocina.



- *EnchufeInteligente*: solo hay uno instalado en el salón, el cual se complementa con el sensor de luminosidad, como se explicará en el módulo correspondiente.

Todos los sensores se representan como (*dispositivo ?tipo ?habitacion*), donde *?tipo* equivale al identificador de dispositivo (movimiento, magnético, humedad, temperatura, luminosidad, agua o vibración).

### *DSensores.clp*

---

En este fichero se añaden los hechos necesarios para inicializar los sensores, en especial, aquellos que presentan un comportamiento binario (paso de on a off). Por defecto, todos los dispositivos empiezan desconectados.

Además, en este fichero se añaden los hechos necesarios para el correcto funcionamiento del sistema, como el estado de una habitación o de una puerta.

- *Estados*: indica el estado de una habitación, el cual puede ser activa, parece\_inactiva o inactiva. El estado de una habitación lo determinará el sensor de movimiento el número de personas que se encuentre en ella.

*(estado ?habitacion ?estado ?tiempo)*

- *NumeroPersonas*: indica el número de personas que hay en una habitación. En principio, no hay ninguna persona en ninguna habitación. A medida que el sistema vaya recibiendo datos, estos hechos serán más reales.

*(personas ?habitacion ?num\_personas)*

- *EstadoPuertas*: estado de las puertas que cuentan con un sensor magnético. Por defecto, todas las puertas se encuentran cerradas.

*(estado\_puertas ?habitacion ?estado)*

- *InicializarSensores*: se inicializan los sensores binarios, es decir, aquellos que solo pueden estar en on o en off.

- Los sensores de movimiento comienzan a off, es decir, no hay movimiento en ninguna habitación.
- Los sensores magnéticos, a on, es decir, la puerta está cerrada.
- El sensor de agua a off, es decir, no hay agua.

*(valor ?tipo ?habitacion ?valor)*

Esta es la manera en la que se representa la información que se recibe de los dispositivos. Todos presentan el mismo formato, independientemente del dispositivo que sea.

En este fichero se recogen las reglas iniciales para definir la casa y poder trabajar con ella. A estas reglas se le especifica la máxima prioridad, para que la casa esté definida antes de que el sistema empiece a trabajar. Las reglas son:

- *Posible\_pasar\_puerta*: si hay una puerta entre dos habitaciones y aún no se ha indicado que se puede pasar de una a otra, se añade el hecho  
(*posible\_pasar* ?habitacion1 ?habitacion2)
- *posible\_pasar\_paso*: si hay un paso sin puerta entre dos habitaciones y no se ha indicado aún la conexión entre ellas, se añade el hecho  
(*posible\_pasar* ?habitacion1 ?habitacion2)
- *necesario\_pasar*: si solo se puede llegar a una habitación desde otra concreta, es decir, solo aparece esa habitación en un hecho *posible\_pasar*, se añade el hecho  
(*necesario\_pasar* ¿?habitacion1 ¿?habitacion2)
- *habitaciones\_interiores*: si una habitación no tiene ventanas, y no es la habitación *jardín*, se indica que es una habitación interior con el hecho  
(*habitación\_interior* ?habitacion)

Este es el módulo principal. Se encarga de recoger los valores de los sensores y llamar al módulo correspondiente si es necesario. Por tanto, las reglas recogidas en este fichero son:

- *Registrar\_valor*: cuando recibe una lectura de uno de los sensores de una habitación y en la base de hechos se tiene registrado que en esa habitación está instalado ese sensor, se guarda el valor recibido y el momento en el que se recibe.

El hecho que se recibe con la información captada con el sensor es

(*valor* ?*tipo* ?*habitacion* ?*valor*)

En la que ?*tipo* indica el sensor que está enviando la información. Para registrar el valor de un sensor, se utilizan los hechos

(*valor\_registrado* ?*tiempo* ?*tipo* ?*habitación* ?*valor*)  
(*ultimo\_registro* ?*tipo* ?*habitación* ?*tiempo*)

El valor de ?*tiempo* se consigue sumándole al valor de (*hora\_inicio* ?*t*) el número de segundos que se corresponden a la hora en la que se recibe la lectura del sensor, la cual se consigue con el hecho (*hora* ?*hora* ?*minuto* ?*segundo*).

- *limpiar\_registros*: cuando se registra un valor, se añade el hecho *ultimo\_registro*, ya que es la última información que se tiene. Para que los hechos de la base de hechos sean correctos, se tiene que borrar cualquier hecho del tipo *ultimo\_registro* de un sensor en una habitación que aparezca más de una vez.

Para saber el hecho que hay que borrar, se comparan las marcas de tiempo y se borra la más antigua (el hecho que tenga una marca de tiempo más pequeña).

- *activacion\_sensor*: solo se utiliza con los sensores que hemos denominado binarios, es decir, aquellos cuyos valores pueden ser on u off. Esta regla se activa cuando el nuevo valor recibido por un sensor de este tipo es on y el último valor guardado del sensor es off. Este evento se guarda con el hecho

*(ultima\_activacion ?tipo ?habitacion ?tiempo)*

El valor de *?tiempo* se consigue de la misma manera que en la regla *registrar\_valor*.

- *desactivacion\_sensor*: tiene un funcionamiento similar a la regla *activación\_sensor*, pero guardando cuando un sensor pasa de on a off. EL evento se refleja con el hecho

*(ultima\_desactivacion ?tipo ?habitacion ?tiempo)*

Dónde el valor de *?tiempo* se consigue igual que antes.

- *eliminar\_activacion*: igual que pasa con los hechos *ultimo\_registro*, solo interesa un solo hecho para cada sensor de cada habitación. Por esta razón se deben eliminar todos los registros antiguos que no correspondan con la activación más reciente del sensor. Para ello, se comparan las marcas temporales.
- *eliminar\_desactivacion*: igual a la regla *eliminar\_activación* pero para los hechos del tipo *ultima\_desactivacion*.
- *llamada\_inicial*: si no se tiene un hecho que indique la ubicación exacta de la persona y del perro en la casa, se llama al módulo encargado de conseguirlas. Para activar el módulo, se añade a la base de hechos

*(activa iniciar)*

- *llamada\_movimiento*: si se registra la activación de una habitación, se detecta movimiento en una habitación en la que no había movimiento, se activa al módulo de movimiento con el hecho

*(activa movimiento)*

- *llamada\_magnetico*: de la misma manera, si se detecta que una puerta estaba cerrada y se abre, desactivación del sensor magnético, se llama a su módulo con el hecho

*(activa magnético)*

- *llamada\_luminosidad\_baja*: cuando se detecta una luminosidad baja, menor a 200 lux, se llama al módulo encargado con el hecho

*(activa luminosidad)*

- *llamada\_luminosidad\_alta*: de la misma manera que en la regla anterior, si se detecta mucha luz, un valor por encima de 400 lux, se llama al módulo con el mismo hecho que en la regla *llamada\_luminosidad\_baja*.
- *llamada\_humedad*: si el último registro del sensor de humedad es un valor menor a 600, a partir de 650 es ligeramente húmedo, se activa el módulo encargado con el hecho

*(activa humedad)*

- *llamada\_varios\_agua*: cuando el último registro del sensor de agua es una desactivación, ha pasado de on, hay agua, a off, no hay agua, se llama al módulo con el hecho

*(activa varios)*

- *llamada\_varios\_vibracion*: si el detector de vibración manda una señal, significa que se ha interactuado con el cubo de la comida, por lo que se llama al módulo encargado de su estudio con el mismo hecho que en la regla anterior,

*(activa varios)*

- *llamada\_varios\_temperatura\_alta*: de media, a partir de 20 grados Centígrados hay que empezar a controlar al perro, ya que para él empieza a ser una temperatura alta. Por esta razón, cuando se recibe un valor de temperatura superior a 20 grados, se activa el módulo con el hecho

*(activa varios)*

- *llamada\_varios\_temperatura\_baja*: tiene un funcionamiento similar a la regla anterior, *llamada\_varios\_temperatura\_alta*, pero se activa cuando la temperatura es menor a 5 grados Centígrados. Llama al mismo módulo con el mismo hecho.

### MIniciarUbicacion.clp

---

Este módulo se encarga de ubicar a la persona y el perro en la casa, para facilitar las tareas posteriores. Solo se activa una vez, y, en cuanto se consigue ubicar a los individuos, se desconecta.

Para poder saber en que habitación está cada uno, se utiliza el horario de trabajo de la persona. Si de lunes a viernes en el horario laboral del dueño solo hay una habitación activa, se le asigna al perro la ubicación de la habitación y al dueño la ubicación *exterior*, que significa que no está en la casa. Para ello, se utilizan las reglas:

- *NoDuenio\_maniana*: para el intervalo de horas entre las 8 de la mañana y las 13 del mediodía.
- *NoDuenio\_tarde*: para el intervalo de horas entre las 16 de la tarde y las 20 de la noche.

En cambio, si es sábado, se utilizará también el horario en la que la persona se encuentra en el despacho. Si hay dos habitaciones activas y una de ellas es el despacho, se le asigna a la persona la ubicación *despacho* y al perro la otra habitación activa. Para esta tarea se utiliza la regla *NoDuenio\_sabado*, que se activa si el día de la semana es sábado, la habitación *despacho* está activa y la hora está dentro del intervalo de las 10 de la mañana y las 13 del mediodía.

Todas las reglas añaden a la base de hechos los hechos

(ubicación\_perro ?habitacion)    (ubicación\_persona ?habitacion)

### MMovimiento.clp

---

Este módulo se activa cada vez que se activa el sensor de movimiento de una habitación, es decir, pasa de off a on. Se mantiene activado hasta que se cambie el estado de las habitaciones implicadas, es decir, cuando el estado de las habitaciones sea *activa* o *inactiva*. Además, este módulo llama al módulo *MUbicacion.clp* para actualizar las ubicaciones de la persona y del perro al moverse por las habitaciones.

Las reglas que se utilizan son las mismas que las de la práctica 1, que son:

- |   |                                    |
|---|------------------------------------|
| - <i>Habitación_activa.</i>               | - <i>habitación_cero_personas.</i> |
| - <i>habitación_parece_inactiva.</i>      | - <i>posible_paso_habitacion.</i>  |
| - <i>habitación_inactiva.</i>             | - <i>no_posible_paso.</i>          |
| - <i>paso_habitacion_una_persona.</i>     | - <i>no_parece_inactiva.</i>       |
| - <i>paso_habitacion_varias_personas.</i> |                                    |

Para llamar al módulo encargado de actualizar la ubicación, se utiliza el hecho

*(activa ubicación)*

Además, se añaden otros hechos para indicar las habitaciones que se han activado y desactivado y facilitar la tarea del módulo. Esos hechos son

*(activación ?habitación) (desactivación ?habitación)*

Estos tres hechos se añaden solo en las tareas en las que se cambia el estado de la habitación, es decir, en las reglas: *habitación\_activa*, *habitación\_inactiva*, *habitación\_cero\_personas* y *no\_parece\_inactiva*.

*MUbicacion.clp*

---

Cuando se actualiza el estado de una habitación, significa que o la persona o el perro o los dos se han cambiado de habitación. Por tanto, hay que actualizar la ubicación para que la información de la base de hechos sea correcta. Para esta tarea, se utilizan los hechos creados en el módulo de movimiento, *activación* y *desactivación*.

Para esta tarea se cuenta con las siguientes reglas:

- *Posibles\_ubicaciones*: si la persona o el perro están en una habitación que se ha desactivado, se agregan como *posible\_ubicacion* del individuo todas las habitaciones que se han activado y desde las que se puede acceder desde la habitación en la que se encuentra.
- *una\_posible\_ubicacion*: si solo hay un posible paso activo desde esa habitación a otra, se deduce que el individuo está en esa habitación.
- *dos\_posibles\_ubicacion*: si hay dos posibles habitaciones a las que se puede haber movido, pero el otro individuo está en una de ellas, se supone que el individuo está en la otra habitación, por lo que se actualiza su ubicación.
- *ninguna\_posibilidad*: si se ha desactivado una habitación, pero no se ha activado ninguna habitación contigua, no se modifica la ubicación y se supone que el individuo sigue en la misma habitación.
- *desactivar\_modulo\_ubicacion*: cuando no quedan más opciones que probar, se desactiva el módulo.

Este módulo se activa cuando se detecta que una puerta que da a una zona restringida se abre. Las reglas recogidas en este módulo se encargan de estudiar quien abrió la puerta, si la persona o el perro. En caso de que la haya abierto el perro, mandar un aviso al dueño de que el perro ha entrado en una zona restringida, la habitación en la que ha entrado y la hora a la que lo ha hecho.

Además, este módulo también actualiza la ubicación de los individuos al detectar la puerta que abren y la habitación en la que estaban.

- *Perro\_solo\_entra*: si se abre una puerta y el dueño no está en la casa, se deduce que es el perro el que abre la puerta, por lo que se manda un aviso a la persona.
- *perro\_entra*: si el perro y el dueño están en la casa y se abre una puerta que solo se podía abrir desde la habitación del perro, se deduce que fue el perro y se manda un aviso a la persona.
- *persona\_entra*: mismo caso que el anterior pero con la persona en vez de con el perro. En este caso, no se manda ningún aviso.
- *perro\_persona\_puede\_entrar*: si tanto el perro como la persona pueden haber abierto la puerta, se ponen los dos posibles movimientos con el hecho

*(posible\_puerta ?individuo ?habitación\_origen ?habitación\_destino)*

- *semi\_descartar\_perro*: si había una posibilidad de que fuera el perro pero se tiene un nuevo registro de que la puerta se cerró, se le baja más la probabilidad de que fuera la persona, con el hecho

*(poco\_posible\_perro ?habitación\_origen ?habitación\_destino)*

- *semi\_descartar\_persona*: si el ultimo registro de movimiento de la habitación en la que está el dueño es una *activación*, se baja la probabilidad de que fuera la persona con el hecho

*(poco\_posible\_persona ?habitación\_origen ?habitación\_destino)*

- *misma\_habitacion*: si el perro y el dueño están en la misma habitación, se deduce que es el dueño el que ha abierto la puerta, ya que no hubiera dejado al perro hacerlo.
- *descartar\_perro*: si había una posibilidad baja de que fuera el perro, la puerta se cerró después de abrirse y la persona estaba en la misma habitación, se descarta al perro.
- *no\_descartar\_perro*: si solo queda la posibilidad de que la haya activado el perro, se deduce que fue él y se manda un aviso al dueño.
- *no\_descartar\_persona*: si solo queda la posibilidad de que haya sido el dueño, se deduce que fue él.

- *resuelto\_persona*: si se ha deducido que la ubicación de la persona es la destino, se borran los hechos de *posible\_puerta* y se deduce que fue la persona.
- *resuelto\_perro\_solo*: si se ha deducido que la ubicación del perro es la de destino y la persona no está ahí, se borran los hechos de *posible\_puerta* y se deduce que fue el perro y se le manda un aviso al dueño.
- *persona\_sale\_casa*: si se activa la puerta de la entrada y la persona estaba en la entrada, se deduce que la persona ha salido de la casa, es decir, su ubicación es *exterior*.
- *persona\_entra\_casa*: si se activa la puerta de la entrada y la persona estaba fuera de la casa, se deduce que la persona ha entrado en la casa y se cambia su ubicación a *entrada*.

---

### *MHumedad.clp*

Este módulo se encarga de detectar y avisar si el perro se ha orinado en el pasillo. Antes de poder mandar un aviso, tiene que descartar que no haya sido el dueño porque, por ejemplo, ha fregado el pasillo. Esta tarea se divide en 3 casos distintos: solo un individuo, perro o persona, está en el pasillo, los dos están en el pasillo o no hay nadie en el pasillo.

Para el primer caso, solo uno de los dos está en el pasillo, se deduce que es el individuo que es encuentra en la habitación el culpable. Las reglas que se utilizan son:

- *Perro\_solo\_pasillo*: crea un hecho para indicar que está resuelto y manda un aviso al dueño.
- *Persona\_sola\_pasillo*: crea un hecho para indicar que está resuelto.

El hecho que se crea es (*mensaje\_pis*) e indica que se ha descubierto al responsable. Se usará en todos los casos.

El segundo caso, se encuentran el perro y el dueño en el pasillo, se resuelve de manera automática. Si el perro se ha orinado, el dueño ya se ha dado cuenta, por lo que es innecesario mandarle un aviso.

Para este caso, se utiliza la regla *todos\_pasillo*, que, al igual que las reglas anteriores, crea el hecho (*mensaje\_pis*) para indicar que ya se ha resuelto.

Para el tercer caso, ni el perro ni el dueño están en el pasillo, se necesitan más reglas que para los casos anteriores, ya que se tiene que trabajar con probabilidades. Para ello, se utilizan dos hechos nuevos:

- (*posible\_pis ?habitación ?tiempo*): se indican todas las habitaciones que están conectadas al pasillo y están activas.
- (*muy\_posible\_pis ?individuo ?habitación ?tiempo*): la posibilidad de que uno de los dos sea más responsable que el otro aumenta.



Estos dos hechos nuevos se utilizan para ayudar en la deducción del culpable. Las reglas que se utilizan son:

- *nadie\_pasillo*: si no hay nadie en el pasillo, se guardan las habitaciones que están activas y conectadas al pasillo.
- *solo\_perro*: si solo está el perro en la casa, se deduce que es él y se envía un aviso al dueño.
- *es\_duenio*: si solo hay una habitación posible y en ella se encuentra la persona, se aumenta la probabilidad de que sea el responsable.
- *es\_perro*: igual que la regla anterior pero para el perro.
- *no\_es\_duenio*: si la habitación en la que se encuentra el dueño es la única habitación activa que está conectada al pasillo pero lleva más tiempo activa que la habitación en la que se encuentra el perro, se descarta el dueño como responsable y se aumenta la probabilidad de que sea el perro.
- *no\_es\_perro*: igual que la regla anterior pero quitándole responsabilidad al perro. En este caso, se descarta que sea el perro, por lo que no interesa saber si es el dueño o un error.
- *solo\_queda\_perro*: si solo como posible opción que sea el perro, se deduce que fue él y se le manda un mensaje al dueño.
- *mas\_posible\_perro*: si hay dos habitaciones activas conectadas al pasillo y en la que está el perro lleva menos tiempo activa, se deduce que fue el perro.
- *mas\_posible\_persona*: si hay dos habitaciones activas conectadas al pasillo y la que habitación en la que está el dueño lleva menos tiempo activa, se deduce que fue el dueño.
- *Indecidible*: si están los dos en la misma habitación y no hay manera de descartar, se manda un mensaje de aviso para indicar que puede que el perro se haya orinado en el pasillo.
- *error\_sensor*: si tras 10s no se ha podido deducir quien es el responsable, no se deduce nada.

Por último, una vez se ha deducido el culpable, es decir, hay un hecho del tipo (*mensaje\_pis*) en la base de hechos, se desactiva el módulo con la instrucción *desactivar\_modulo*. El módulo no se desactiva hasta que el suelo no esté seco, para evitar más avisos.

Este módulo se encarga de determinar si hay mucha o poca luz en la casa y actuar en consecuencia si el perro está solo en el piso. Es una funcionalidad trivial, la parte de la incertidumbre la resuelve otro módulo, pero se quiso tener en un módulo a parte para poder responder a las peticiones de este módulo.

Si se activa el módulo, significa que hay poca o mucha luz en la casa. Por lo que primero se comprueba si el perro está solo. Si el dueño también está en la casa, se activa la regla *no\_perro\_solo*, que simplemente desactiva el enchufe en caso de que esté conectado.

Este enchufe se utiliza para una lámpara auxiliar para evitar que el perro esté a oscuras, es su único uso, por lo que, cuando el dueño está en la casa, no hay necesidad de usarla.

Si el perro está solo en la casa y hay mucha o poca luz, se activan las reglas *poca\_luz* y *mucha\_luz*, las cuales mandan la orden de conectar o desconectar el enchufe. Para eso, se utiliza el hecho

*(acción enchufe ?habitación ?accion)*

En la que la acción puede ser *desconectado*, para desconectar el enchufe en caso de que haya mucha luz (también utilizado en la regla *no\_perro\_solo*), o *conectado*, para conectar el enchufe. En ambos casos se manda un aviso al dueño para indicarle que se ha conectado o desconectado el enchufe.

Para poder ejecutar las acciones de conectar y desconectar el enchufe, se utilizan las siguientes reglas:

- *conectar\_enchufe*: si se recibe el hecho *accion* con el valor de *?accion* como *conectado*, se manda unos nuevos datos de que se ha conectado el enchufe y del nuevo valor de la luminosidad de la habitación. Los hechos son:

*(valor enchufe ?habitación conectado) (valor luminosidad ?habitación (+ ?valor 200))*

La intensidad de la bombilla es de 200, por eso se le suma al valor que había antes. Además de esos dos hechos, se añade *(luz\_encendida ?habitación 200)* para indicar que en la habitación *?habitación* la luz está encendida y la bombilla tiene una intensidad de 200.

- *desconectar\_enchufe*: es similar al anterior pero para el valor *desconectado*. En vez de sumarle 200, le resta 200 y elimina el hecho de *luz\_encendida*.

En este fichero se recogen las tareas de los sensores que son triviales, es decir, no cuentan con ningún tipo de incertidumbre o la incertidumbre la resuelve otro módulo.

En primer lugar, están las tareas relacionadas a la temperatura. Para esto, se cuenta con tres tipos de avisos distintos, en función al valor de la temperatura.

- *temperatura\_alta\_leve / temperatura\_baja\_leve*: estas dos reglas se activan cuando la temperatura está en ligeramente alta o baja para el perro. Mandan un aviso de tipo *AVISO*, informativo.
- *temperatura\_alta\_fuerte / temperatura\_baja\_fuerte*: cuando la temperatura alcanza el rango en el que empieza a ser peligroso, se envía un aviso de tipo *ALERTA* para indicar que se tiene que estar pendiente.
- *temperatura\_alta\_peligrosa / temperatura\_baja\_peligrosa*: cuando se superan los umbrales de peligro, se manda un aviso de tipo *PELIGRO* para indicar que la temperatura pone en riesgo al perro y se actúe de manera consecuente.

Los rangos de temperatura son, medidos en grados Centígrados:

	<b>AVISO</b>	<b>ALERTA</b>	<b>PELIGRO</b>
<b>FRIO</b>	[-3, 5[	[-6, -3[	< -6
<b>CALOR</b>	]20, 25]	]25, 30]	> 30

El segundo caso es el nivel del agua del bebedero. Para este caso se tienen 3 casos distintos, con una regla para cada uno:

- *agua\_leve*: si se desactiva el sensor de agua, es decir, no hay agua, y la temperatura es menor a 25 grados Centígrados, se manda un aviso de tipo *AVISO* de falta de agua.
- *agua\_necesaria\_temperatura*: si se desactiva el sensor de agua y la temperatura es mayor o igual a 25 grados Centígrados, se envía un aviso de tipo *ALERTA* por falta de agua.
- *agua\_necesaria\_tiempo*: si el cuenco lleva vacío más de una hora, se manda un aviso de tipo *ALERTA* para indicar que el perro lleva una hora sin posibilidad de beber.

Por último, si el sensor de vibración se activa, significa que se ha interactuado con el cubo de la comida. Por tanto, con la instrucción *perro\_cocina* se comprueba si es cosa del perro. Si está solo en la cocina, se manda un aviso al dueño de tipo *AVISO* de que el perro está en la despensa. Si es otra situación, se ignora la señal del sensor.

Una vez se haya rellenado el cuenco de agua y la temperatura esté en el intervalo [5, 20], se desactiva el módulo con ayuda de la regla *eliminar\_activa*.

## 5. MANUAL DE USO

---

Para poder ejecutar y probar el sistema, se tiene que cargar el fichero “*Practica.bat*” como batch. El fichero está preparado para limpiar todos hechos y reglas que se tuvieran, cargar los hechos y reglas del SBC y ejecutar las reglas necesarias para probar el sistema.

Al ejecutar estas reglas iniciales, se le preguntará al usuario el día de la semana y la hora en la que se inicia el sistema, para poder funcionar de manera correcta. Una vez se introduce esta información, el sistema está listo para usarse.

Para poder probarlo, se debe ir introduciendo la información que envían los sensores al sistema e ir ejecutando. Se recomienda no usar (*run*) sin especificar el número de ejecuciones, ya que se tiene una regla que se ejecuta cuando no quedan instrucciones para ir actualizando la hora, por lo que se entraría en un bucle infinito.

Si se quiere utilizar un simulador que vaya generando las instrucciones, se puede incorporar al batch o, una vez se ejecuten las reglas iniciales, cargarlas en el sistema.

Para que la hora se actualice, debe dejarse que el sistema resuelva las reglas que tenga hasta que no le queden reglas que activar. Si cuenta constantemente con alguna activación, la hora será siempre la misma.

Los valores de todos los dispositivos siguen el siguiente formato:

*(valor ?tipo ?habitación ?valor)*

Dónde,

- *?tipo* es el identificador del dispositivo que envía la señal. Los identificadores son: movimiento, magnético, humedad, luminosidad, temperatura, agua, vibración y enchufe.
- *?habitación* es la habitación en la que se encuentra el dispositivo que envía la señal.
- *?valor* es el valor que recibe, en función del dispositivo desde el que se reciba la información. Ese valor puede ser desde un valor, correspondiente a la cantidad de humedad, luminosidad o temperatura que se detecte, hasta una etiqueta que especifique un evento, como vibración, desconectado u on.

Si se introduce un valor de un sensor en una habitación en la que no hay ningún sensor de ese tipo, se ignora ese hecho.

## 6. REFERENCIAS

---

*Bajas temperaturas, altas temperaturas: los perros sufren los efectos del clima.* (s.f.). Obtenido de SrPerro.com: <https://www.srperro.com/consejos/salud-perruna/bajas-temperaturas-altas-temperaturas-los-perros-sufren-los-efectos-del-clima>

Gómez, M. (2013). *Golpe de calor en perros*. Obtenido de Camino con mi perro : [http://www.caminoconmiperro.com/cuidados/golpe\\_de\\_calor\\_en\\_perros.html](http://www.caminoconmiperro.com/cuidados/golpe_de_calor_en_perros.html)

*Labrador Retriever.* (s.f.). Obtenido de Petplan: <https://www.petplan.co.uk/pet-information/dog/breed/labrador/>

Smyth, D. K. (03 de Abril de 2015). *How hot is too hot for your dog?* Obtenido de PetPlan: <https://www.gopetplan.com/blogpost/hot-weather-and-dogs>

Smyth, D. K. (08 de Noviembre de 2018). *How cold is too cold for your dog?* Obtenido de PetPlan: <https://www.gopetplan.com/blogpost/cold-weather-and-dogs>