

Desarrollo de un SE

Ingeniería del Conocimiento.

SBC para alertar de problemas de personas mayores que viven solas.

**UNIVERSIDAD DE GRANADA
E.T.S.I. INFORMÁTICA Y TELECOMUNICACIÓN**



**UNIVERSIDAD
DE GRANADA**

Departamento de Ciencias de la
Computación e Inteligencia Artificial

**Grado en Ingeniería Informática. Tercero.
Curso 2018-2019.**

Daniel Bolaños Martínez

76592621-E

danibolanos@correo.ugr.es

Grupo 1 - Jueves 17:30h

Índice

1. Resumen del Sistema Experto.....	1
2. Descripción del proceso seguido.....	1
Procedimiento construcción de la base de conocimiento.....	1
Procedimiento validación y verificación.....	1
3. Descripción del Sistema.....	2-7
Variables de entrada.....	2
Variables de salida.....	2
Conocimiento global del sistema.....	2-3
Especificación de los módulos desarrollados.....	3-4
Hechos y reglas de cada módulo.....	4-7
4. Breve manual de uso.....	7

1. Resumen del Sistema Experto

El Sistema Experto desarrollado es capaz de gestionar una casa donde vive una persona mayor sola, salvo una visita diaria de un/una asistente que la atiende de lunes a sábado de 10h a 14h.

El sistema ha sido implementado en CLIPS y, mediante una serie de reglas y hechos con distinta prioridad, será capaz de llevar a cabo estas actividades y funcionalidades:

- Gestión de luces inteligentes usando sensores de movimiento y luminosidad.
- Alertar sobre diferentes acciones que la persona mayor realice y sean un indicio de peligro para la salud de la misma.

Según los hechos que aparezcan en la base de conocimiento, decidirá qué acción realizar. Muchos de estos hechos serán introducidos por sensores de forma que el sistema sabrá lo que está ocurriendo en todo momento. Aun así el conocimiento no siempre será perfecto, por lo que deberá aprender a trabajar con un ligero grado de incertidumbre.

2. Descripción del proceso seguido

2.1. Procedimiento construcción de la base de conocimiento

La representación del conocimiento debe ser lo más intuitiva posible, con hechos que expliquen claramente qué conocimiento representan. Se ha de intentar mantener por medio de estos hechos un conocimiento total de lo que ocurra en la casa y así poder actuar acorde a éste.

Para obtener el conocimiento, además de los datos proporcionados por sensores, debemos observar el planteamiento del problema e introducir manualmente aquello que creamos relevante para representar el mismo.

Por ejemplo, será necesario definir los días de trabajo del asistente, el estado inicial de las luces, el estado de las habitaciones... Así como los pasos entre habitaciones que defina la estructura de nuestra casa.

2.2. Procedimiento validación y verificación

Para verificar el desarrollo he comprobado que la lógica seguida fuese completamente funcional y generase los resultados esperados.

Para la validación, aunque debería someterse a una aplicación en un problema real, disponemos de la simulación de la práctica anterior que podemos usar para comprobar que cumple las necesidades y funciona adecuadamente.

3. Descripción del Sistema

3.1. Variables de entrada

Las variables de entrada del problema que usaremos en todo el SBC, son los sensores.

Definimos sensores de tres tipos:

Sensor de movimiento: tiene dos estados ON y OFF.

- En estado OFF: no envía ninguna información hasta que no detecta movimiento.
- En estado ON: re-escanea cada 60s y si detecta movimiento envía *movimiento on* y la luminosidad en lux, y si no detecta movimiento envía *movimiento off* y pasa a estado OFF.

Pulsador de luz: reconoce una pulsación manual, tiene dos estados Conectado (luz encendida) y Desconectado (luz apagada), y ejecuta las órdenes de encender, apagar y cambiar (pasa de encendido a apagado y viceversa).

Sensor de luminosidad: mide luminosidad del exterior en lux.

3.2. Variables de salida

Como variables de salida, tenemos un informe de los distintos registros de los valores obtenidos a partir de los sensores de una habitación, además de los avisos por pantalla para advertir de que se cumplen las funcionalidades que se describieron anteriormente relacionadas con la actividad de la persona mayor en la casa.

Respecto al informe generado, cuando aparezca en la base de hechos (**informe ?h**) bajo demanda, el sistema devolverá una lista ordenada (más reciente a más antiguo) de los datos enviados por los sensores de la habitación **?h**, indexada por el instante en que fueron recibidos.

3.3. Conocimiento global del Sistema

El sistema debe conocer tanto la estructura de la casa como el estado de los interruptores de luz y de las habitaciones.

Para representar la casa, utilizaremos los hechos (**habitacion ?h**) identificando cada habitación por su nombre. El plano de la casa definida, sigue el siguiente dibujo:

Para representar los pasos y las puertas usaremos (**paso ?h1 ?h2**) y (**puerta ?h1 ?h2**). Además hemos definido dos reglas para extraer conocimiento de la estructura de la casa a partir de las habitaciones y sus puertas. Usaremos:

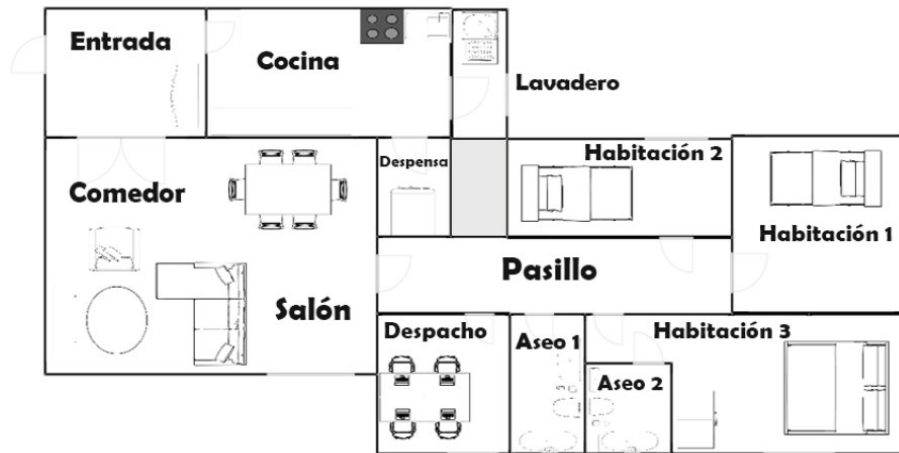


Figure 1: Estructura de la casa

(**posible_pasar ?h1 ?h2**): si se puede pasar directamente (por una puerta o por un paso) de una habitación a otra.

(**necesario_pasar ?h1 ?h2**): si para acceder a una habitación solo se puede pasar desde otra.

He definido un interruptor para cada habitación como:

(**interruptor ?h on/off**): dependiendo del estado de la luz de esa habitación.

Y el estado de la habitación podrá ser:

(**estado_habitacion ?h inactiva|activa|parece_inactiva**): representa el estado de la habitación; **activa** si se registra movimiento en la habitación, **parece inactiva** si no se detecta movimiento (se recibe un off en movimiento), pero no podemos asegurar que esté inactiva, **inactiva** si se ha producido un paso reciente (3 segundos) a una habitación colindante o pasan 10 segundos pareciendo inactiva.

Además tendremos conocimiento sobre las constantes de luz “ideal” en lux para cada habitación.

3.4. Especificación de los módulos desarrollados

En la realización de la práctica podemos encontrar dos grandes módulos, la gestión de luces inteligentes y la aplicación de las funcionalidades para advertir de las acciones de una persona mayor en la casa.

En este apartado explicaré a grandes rasgos las funcionalidades implementadas y en el siguiente me dispondré a explicar cada regla de forma más detallada.

En la **gestión de luces**, seguiremos tres reglas:

- Si una habitación no está vacía y hay poca luz la enciendo.
- Si una habitación está vacía y tiene la luz encendida, la apago.
- Si la luz está encendida y hay mucha luminosidad, apago la luz.

En la **gestión de avisos**, implementaremos nueve funcionalidades. Avisando por pantalla cuando:

- La persona ha salido de la casa.
- Siendo de día, la persona no se ha movido por la casa en las últimas 3 horas.
- La persona se ha levantado durante más de 15 minutos por la noche.
- Estando sola, la persona lleva más 20 minutos en el baño.
- La persona no ha ido al baño en las últimas 12 horas.
- La persona ha ido al baño varias veces en las últimas 3 horas.
- El asistente tendría que haber llegado, pero no lo ha hecho.
- Es tarde y la persona no se ha acostado.
- La persona tiene menos actividad de lo normal o hace desplazamientos por la casa que no hace normalmente en una franja horaria determinada.

3.5. Hechos y Reglas de cada módulo

3.5.1. Gestión de Luces

Para gestionar los interruptores definidos, implementaremos dos reglas para tener información en cada momento del estado de cada interruptor. Esta información la obtenemos gracias al registro *estado_luz* que aporta el *pulsador de luz*.

- **EnciendeInterruptor:** Crea un hecho interruptor a *on* y borra, si lo hay, el antiguo a *off*.
- **ApagaInterruptor:** Crea un hecho interruptor a *off* y borra, si lo hay, el antiguo a *on*.

Una de las claves de este SBC es como se definen los estados de las habitaciones según su *movimiento*, ya que son clave importante tanto para el *manejo de luces* como para el resto de funcionalidades.

Para ello, definimos dos reglas para pasar la habitación de un estado a otro cuando sea conveniente.

- **HabitacionActiva:** Si se registra *movimiento* en una habitación y estaba en estado *inactiva* o *parece_inactiva*, pasa a estar *activa*.
- **HabitacionPareceInactiva:** Si **no** se detecta *movimiento* en la habitación y está *activa* pero no se detecta un *paso reciente* a una habitación colindante, la habitación queda *activa*. Si la habitación está en el estado *parece_inactiva*, pasará a *inactiva* en 3 segundos si existe el paso reciente o en 10 segundos en otro caso.

Describiremos ahora cómo se han implementado las tres funcionalidades de este módulo.

Si una habitación no está vacía y hay poca luz la enciendo.

- **HabitacionActivaPocaLuz:** Si la habitación está *activa* y el valor de luminosidad es menor que la necesaria, enciendo el interruptor (pasa a *on* si no lo estaba).

Si una habitación está vacía y tiene la luz encendida, la apago.

- **HabitacionInactiva10seg:** Si la habitación *parece_inactiva* durante 10 segundos, pasa a *inactiva*.
- **PasoReciente3seg:** Se detecta un *paso reciente* (3 segundos) a habitación colindante. Entonces la habitación pasa a *inactiva*.
- **ApagoLuzHabitacionInactiva:** Si la habitación está *inactiva*, apago la luz.

Si la luz está encendida y hay mucha luminosidad, apago la luz. (Mucha es el doble de la necesaria).

- **ApagoLuzDobleLuminosidad:** Si el *interruptor* está encendido y el valor de luminosidad es mayor que el doble de la necesitada, apago el interruptor (pasa a *off* si no lo estaba).

3.5.2. Gestión de Avisos

Para gestionar los avisos de control de las acciones y movimientos de la persona mayor, debemos definir antes algunos hechos y reglas que nos servirán para la implementación de las funcionalidades pedidas:

- Lista de hechos de los días de la semana. Pudiendo seleccionar el día que queremos simular.
- Estado de la jornada (día o noche).
- Estado del asistente (*on* si está en la casa, *off* si no está, *deberia_estar* si no ha llegado pero debería haberlo hecho).
- Número de visitas a cualquiera de los dos aseos en un intervalo de 3 horas.
- Actividad media de la persona en la casa definida como el número de movimientos que detecta el sensor.

Algunas reglas que regulan estos hechos son:

- **DiaNoche/NocheDia:** Controla el paso del día a la noche y viceversa. Suponiendo que amanece a las 6:00 y anochece a las 19:00,
- **TrabajoAsistente:** Controla los días de trabajo del asistente. De lunes a sábados. De 10h a 14h. Deja el hecho *asistente* en el estado *deberia_estar*.

Algunas pautas a tener en cuenta:

- La *habitacion1* es la habitación donde duerme la persona mayor.
- Para simplificar el problema, supondremos que el asistente tiene llaves de la casa y la persona mayor, por desconfianza, nunca se acercará a la entrada para abrir la puerta. (Evitamos que se confunda la llegada del asistente con la detección de movimiento de la persona mayor en la entrada).

Ahora pasaremos a describir las reglas que describen las nueve funcionalidades que hemos listado anteriormente:

La persona ha salido de la casa.

- **PersonaSaleCasa:** Si la entrada *parece_inactiva*, es decir, ha registrado *movimiento* en ella, aún no se puede saber si se ha movido de la habitación, pero no se registra *movimiento* en ninguna habitación colindante (*posible_pasar*), supondremos que la persona ha salido de casa.

Siendo de día, la persona no se ha movido por la casa en las últimas 3 horas.

- **DiaCasa3Horas:** Siendo de día, si la diferencia entre la última activación de *movimiento* en la casa con la hora actual es mayor a 3 horas, la persona no se ha movido en ese tiempo.

La persona se ha levantado durante más de 15 minutos por la noche.

- **Noche15Min:** Siendo de noche, si la diferencia entre la última activación de *movimiento* en la habitación 1 con la hora actual es mayor a 15 minutos y la habitación 1 está inactiva. Lleva fuera de su habitación más de 15 minutos.

Estando sola, la persona lleva más 20 minutos en el baño.

- **Sola20MinAseo1/Sola20MinAseo2:** Haremos esta regla útil para cualquiera de los dos baños existentes. El asistente no está, el aseo está *activo* o *parece_inactivo* y el último movimiento de la habitación obligatoria de acceso al baño, fue hace 20 minutos.

La persona no ha ido al baño en las últimas 12 horas.

- **NoAseo12Horas:** La última activación de movimiento en alguno de los dos aseos fue hace 12 horas.

La persona ha ido al baño varias veces en las últimas 3 horas.

- **VisitaAseoVariasVeces3Horas:** Usando el contador de hechos de los ejercicios propuestos. Contamos el número de veces que se detecta movimiento en un aseo. Si el número supera a 2 en menos de 3 horas, se informará de ello y se volverán a poner a 0 los contadores de visitas al baño.
- **NoVisitaAseoVariasVeces3Horas:** Si no ocurre, no se informa pero los contadores vuelven a cero para hacer un *reset* cada 3 horas.

La asistenta tendría que haber llegado, pero no lo ha hecho.

- **AsistenteNoLlega/AsistenteLlega:** Si no existe una última activación de movimiento en la entrada con un tiempo inferior al actual y el asistente, debería estar trabajando deduciremos que el asistente no ha llegado y pondremos el estado del asistente a *off*. En caso contrario, el asistente estará presente y su estado cambiará a *on*.

Es tarde y la persona no se ha acostado.

- **PersonaNoAcostada:** Definiremos tarde para la persona mayor a partir de las 23:00. Son más de las 23:00 y existe alguna habitación distinta al dormitorio (donde duerme la persona mayor) con actividad o que parece que tiene actividad.

La persona tiene menos actividad de lo normal o hace desplazamientos por la casa que no hace normalmente en una franja horaria determinada.

- **ActividadMediaIrregular:** Vamos a utilizar un contador de hechos para calcular la actividad restringida a los domingos por la mañana, día en el que no trabaja el asistente. Incrementaremos el contador en uno, cada vez que se detecte movimiento en una habitación y consideraremos como actividad media normal un valor de 40 movimientos por la casa al día. Si el número de movimientos es menor, el sistema informará de ello.

4. Breve manual de uso

Se utilizará un simulador y los archivos `FuncionesTiempo.clp` y algunas funciones de `CicloControlado.clp` como *HoraActualizada* u *hora_actual* de forma similar a la de la práctica 1.

Para conocer más detalladamente el funcionamiento, sugiero leer los comentarios incluidos en `CasaInteligente.clp` que describen cada regla usada y algunos aspectos importantes de la misma.