TRABAJO DE PRUEBA

SISTEMA DE DOMÓTICA

José Manuel Candilejo Egea
Álvaro de Castro García

Sistemas Operativos
GITT

Curso **2020/21**

ÍNDICE

LA IDEA

Introducción

Para la realización del presente proyecto nos apoyamos en el concepto de domótica: el conjunto de **sistemas capaces de automatizar una vivienda u edificio** de cualquier otro carácter, aportando diversos servicios para aumentar la comodidad de vida de los distintos usuarios.

Teniendo en cuenta la definición expuesta anteriormente y el auge de las denominadas “Smart homes” y de productos como Amazon Echo o Google Home, se nos vino a la mente una idea que pudiésemos implementar tanto físicamente como a nivel de software: un sistema de **control de la potencia consumida en un hogar** en cada instante.

Inspiración

Tras informarnos a cerca de la situación actual del desarrollo de hogares inteligentes, tanto a nivel industrial como a nivel de proyectos individuales, dimos con múltiples ideas de proyectos sobre los cuales poder inspirarnos; principalmente, proyectos pensados para dispositivos como *Raspberry Pi* (ordenadores de placa de bajo coste).

Estos proyectos solían implementar software que permitía el uso de diversos productos, tales como controlar LEDs remotamente o hacer uso de *Google Assistant*, sin embargo, nuestra idea era implementar nuestro propio software haciendo uso de los conocimientos adquiridos en la asignatura, por lo que decidimos atacar el proyecto desde un ángulo distinto.

El plan

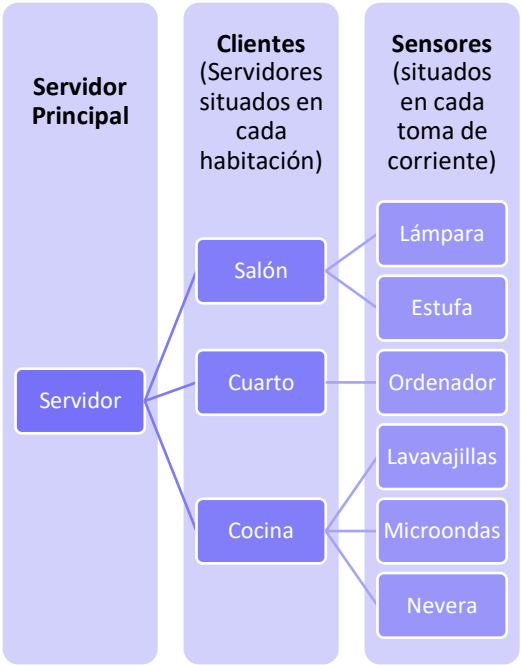
Para poder lograr el objetivo que nos propusimos, debíamos tener bien clara la idea sobre qué era exactamente lo que queríamos diseñar; es por esto, que empezamos a barajar diversas opciones, muchas de las cuales, por más que nos hubiese gustado ponerlas en práctica, se salían de la materia de la asignatura y habrían supuesto un coste económico y de tiempo que harían plantearnos si se trataba de un trabajo de fin de grado.

El proyecto lo planteamos de la siguiente forma: Se trataría de una **serie de sensores** situados en cada toma de corriente de las diversas habitaciones de un edificio (en principio, una casa); éstos, colectarían información sobre el consumo (en kWh) de los diversos electrodomésticos de cada habitación y enviarían dichos datos a unos dispositivos situados cada uno en una sala para que fuesen estos los que recolectasen la información sobre todos ellos y la pusiesen en un formato predefinido.

Estos dispositivos, enviarían la información recibida a cerca de los distintos sensores a un servidor central, el cual mostraría por un *display* una tabla que enumerase **cuántos dispositivos de cada tipo** habían conectados en cada momento, así como el consumo individual de cada uno y el consumo total de todos los mismos.

Planteamiento

Para llevar a cabo la idea, en el caso de tener acceso a todos los recursos que hubiésemos deseado, habríamos utilizado una serie de sensores conectados a diversas tomas de corriente, así como uno o varios ordenadores de placa de bajo coste (en principio, las conocidas *Raspberry Pi*) dispuestos de la siguiente manera:



PROCESO DE DISEÑO

Ideas Previas

Antes de comenzar con el desarrollo del proyecto, barajamos diversas ideas, de las cuales acabamos fusionando varias entre sí para lograr llegar al conjunto de programas que finalmente acabamos desarrollando.

De entre las diversas ideas que barajamos caben destacar:

Servidor de un Banco

La primera idea que tuvimos fue la de basarnos en la práctica final de la asignatura de Fundamentos de Programación II, pero realizarla en C en vez de en Java, de forma que implementaríamos conceptos como la memoria compartida o las colas.

Sin embargo, consideramos que la idea era muy poco original y que, al haber realizado ya el proyecto anteriormente (aunque fuese con otro enfoque), no nos aportaría realmente gran cosa desde un punto de vista académico.

A pesar de esto, consideramos importante mencionar esta idea, pues fue gracias a ella que nos decantamos por realizar un proyecto que siguiese un modelo cliente-servidor mediante el uso de colas de mensajes y de zonas de memoria compartida.

Servidor de Chat Local

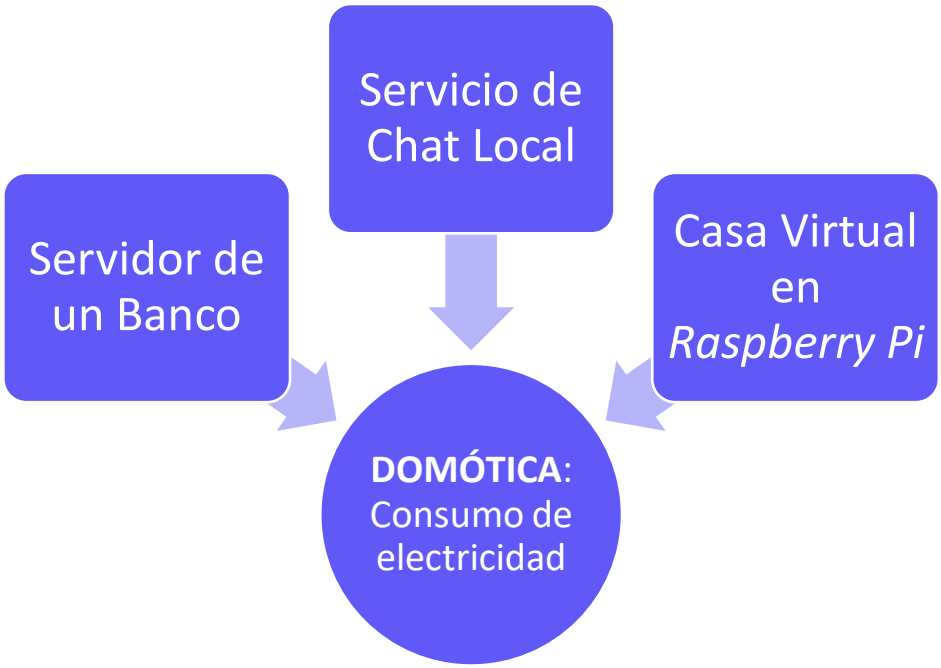
Como bien se puede intuir por el nombre, se trataría de un sistema de mensajería que permitiría compartir no sólo mensajes de texto, si no que, haciendo uso de diversos tipos de señales, se podría notificar entre varias terminales que se iba a proceder a transmitir otro tipo de datos (como imágenes o archivos) entre diversas carpetas en un mismo ordenador haciendo uso de los conocimientos adquiridos en la práctica 9 de la asignatura.

A pesar de haber podido hacer uso de múltiples recursos estudiados en la asignatura, consideramos que la idea de un servidor de chat que sólo funcionase de manera local no tenía una utilidad real y que, por tanto, a pesar de poder haber sido un proyecto interesante a realizar, no tendría mucho sentido su implementación sin hacer uso de conocimientos adquiridos en otras asignaturas que permitiesen la comunicación entre distintas máquinas.

Casa Virtual en Raspberry Pi

Esta idea ya se iba acercando cada vez más a lo que finalmente ha acabado siendo el proyecto que hemos realizado, la idea era tener varios dispositivos periféricos conectados a una misma *Raspberry* que nos permitiesen variar diferentes entradas de un programa que simulase ser un servidor de domótica (temperatura, luminosidad, ruido, …) y que actuase en consecuencia.

A pesar de todo, desconocíamos como utilizar el hardware y no poseíamos ninguna *Raspberry*, por lo que tuvimos que descartar esta idea, aunque nos quedó claro que el concepto de las *Smart Homes* era algo en lo que queríamos profundizar.



Domótica: Consumo de Electricidad

Finalmente, nos decantamos por el que acabó siendo el proyecto que desarrollamos: un sistema de domótica que permitiese conocer el consumo de electricidad en una casa, haciendo uso de herramientas de comunicación entre procesos tales como las colas de mensajes o las zonas de memoria compartida, logrando así implementar conceptos de la asignatura a un sistema con utilidad práctica y que pudiese ser llevado a cabo en físico con las menos modificaciones posibles.

DESARROLLO

Introducción

Tras un largo proceso de diseño y muchas pruebas y errores, finalmente nos decantamos por realizar un sistema compuesto por tres programas distintos: un servidor central, un programa que hiciese de gestor de las diversas salas, y un tercer programa que controlase a los sensores de cada habitación. La comunicación entre estos programas se realiza mediante colas de mensajes y zonas de memoria compartida, haciendo así uso de los recursos aprendidos en la asignatura.

A continuación, expondremos uno por uno el funcionamiento de los distintos programas que constituyen al proyecto a un nivel técnico, explicando las diversas herramientas que se le dan al usuario para poder controlar al sistema.

Servidor

El servidor es el "núcleo" del sistema, se trata del programa que inicializa tanto los semáforos como crea las zonas de memoria compartida y la cola de mensajes. El usuario visualiza el resultado de todas las operaciones que vaya realizando a los diversos sensores y a las salas en la terminal del servidor, de forma que funciona como una central donde se recogen los datos resultantes de la ejecución de todos los demás programas.

La estructura "disp."

Antes de entrar en detalles con respecto a la ejecución del programa, hay que tener en cuenta que se va a trabajar con estructuras de tipo "disp.", que simbolizan a los diversos electrodomésticos o dispositivos que pudiesen estar conectados a la toma de corriente y que están definidas de la siguiente forma:

```
typedef struct{
  char nombre[MAX_TAM_NOMBRE];
  float consumo;
  int opciones;
  bool ON;
}disp;
```

Cuyos parámetros son:

Parámetro	Definición
Nombre	Se trata del nombre del dispositivo. Por ejemplo “bombilla” o “lavadora”
Consumo	Mide el consumo del dispositivo en cuestión en kWh.
Opciones	A la hora de recibir un “disp.” por cola de mensajes, el servidor mira, dependiendo de la opción que se le indique, qué debe realizar con la estructura recibida.
ON	Marca si el dispositivo está encendido o apagado.

Donde “Opciones” puede valer:

- **ANADIR:** Al recibir un disp. con esta opción, el servidor añade dicho dispositivo a un buffer de tamaño MAX_TOTAL donde almacena los dispositivos registrados.
- **ELIMINAR:** Si el servidor recibe un dispositivo con esta opción, buscará en su buffer de dispositivos a un dispositivo cuyo nombre coincida con el indicado, para, posteriormente, eliminarlo de la misma.
- **EXIT:** Esta opción indica al servidor que termine su ejecución, eliminando los semáforos y las zonas de memoria compartida, así como la cola.
- **CONMUTA:** Mediante esta opción, se cambia la opción “ON” de un dispositivo cuyo nombre coincida con el suministrado.
- **SIN_DEFINIR:** Tras recibir una estructura con esta opción marcada, el servidor imprime por pantalla un mensaje de error.

Cabe destacar que no pueden existir dos estructuras “disp.” que compartan nombre, pero difieran en su consumo; esta propiedad está garantizada mediante una de las zonas de memoria compartida de las que hablaremos a continuación.

La cola de mensajes

La cola de mensajes es el principal elemento utilizado para la comunicación entre el servidor y los diversos gestores. Su función es almacenar estructuras “disp.” enviadas por los gestores de forma que el servidor pueda leerlas y realizar las gestiones oportunas sobre su buffer.

Está controlada por el semáforo “cola” y el servidor se encuentra en un bucle leyendo constantemente de ella, para así actuar conforme a la información recibida.

La cola únicamente permite estructuras del tipo “disp.”, por lo que, para poder gestionar el comportamiento del servidor desde los diversos gestores, se envían las operaciones a realizar en el campo “opciones” de las diversas estructuras, siendo así, por ejemplo, que si el servidor recibe una estructura con nombre “bombilla” y con opción “ELIMINA”, buscará en su buffer una estructura bombilla y la eliminará.

Memoria Compartida: Gestores

Esta es una de las dos zonas de memoria creadas por el servidor. Se trata de un número (en este caso, 5, pero puede ser ampliado o reducido) de bits que los gestores consultan para ver si pueden conectarse o no al servidor.

Los bits de esta zona de memoria tienen dos estados: 0 si están disponibles para ser llenados o 1 en caso contrario. Cada gestor debe, previamente a hacer ningún tipo de operación, consultar si el servidor tiene un espacio disponible para conectarse a él, en caso contrario, no podrá realizar la conexión.

Conforme se van conectando los diversos gestores, estos bits se irán llenando en orden hasta que no permitan más conexiones, y en caso de liberarse uno o más huecos, otros gestores podrán conectarse, llenando estos huecos en orden.

Memoria Compartida: Dispositivos

Existe una última zona de memoria que crea el servidor, aunque esta no es consultada por los gestores, sino por el programa que simula el comportamiento de los sensores; se trata de una zona de memoria que contiene una serie de estructuras “disp.” predefinidas y que garantiza que no se almacenen estructuras con el mismo nombre, pero distinto consumo.

Esta zona de memoria compartida tiene seis dispositivos predefinidos (bombilla, ventilador, lavadora, lavavajillas, TV y nevera), y conforme el usuario vaya definiendo más, estos se irán almacenando aquí para permitir utilizarlos en cualquier otra sala.

El servidor, además, consulta esta zona de memoria compartida a la hora de imprimir por pantalla los diversos datos proporcionados por los gestores, aunque eso se tratará en profundidad a continuación.

Función imp_tabla

La función que utiliza el servidor para mostrar por pantalla en todo momento el resultado de la ejecución de los diversos programas se llama “imp_tabla”. Utiliza el buffer interno del servidor en el cual se van almacenando los dispositivos conectados en cada momento, así como su estado (ON/OFF), gracias al cual hace un recuento del número de dispositivos de cada tipo que están encendidos y su consumo individual y total.

Representa en una tabla los siguientes datos:

- Nombre del dispositivo.
- Número de dispositivos de este tipo encendidos.
- Consumo individual de cada dispositivo de este tipo.
- Consumo total de todos los dispositivos de este tipo.
- Consumo total entre todos los dispositivos.

Ejemplo de ejecución de la función “imp_Tabla”:

```
+---------------+---------------+---------------+---------------+
|Nombre         |Num. Disp.     |Consumo I(kWh) |Consumo T(kWh) |
+---------------+---------------+---------------+---------------+
|     bombilla  |             3 |      0.10     |      0.30     |
|   ventilador  |             2 |      3.30     |      6.60     |
|     lavadora  |             0 |    255.00     |      0.00     |
| lavavajillas  |             0 |    246.00     |      0.00     |
|           TV  |             1 |    263.00     |    263.00     |
|       nevera  |             1 |    662.00     |    662.00     |
|           PC  |             1 |    300.50     |    300.50     |
+---------------+---------------+---------------+---------------+
Consumo total:    1232.40 KWh
```

Gestor

Los diferentes gestores simbolizan a las pequeñas terminales que deberían gestionar la conexión de diversos dispositivos en las distintas habitaciones. El número de gestores que pueden estar conectados a un mismo servidor está determinado en la zona de memoria anteriormente comentada.

En el sistema que hemos diseñado, la tarea del gestor es poder conmutar el estado de los dispositivos que se le conectan (encendido/apagado), dar la opción de añadir o eliminar un nuevo dispositivo y ofrecer al usuario poder cerrar el servidor desde cualquiera de los gestores.

```
TERMINAL DEL GESTOR 0
Seleccione que desea hacer:
1.-Listar mis dispositivos
2.-Eliminar gestor y dispositivos del servidor
3.-Eliminar gestor y cerrar sevidor

Opción:
```

Listar mis dispositivos

La opción uno hace uso de una función distinta de “imp_Tabla”. En esta, se listan únicamente los tipos de dispositivos que estén añadidos uno por uno, por lo que, en caso de que hubiese dos o más unidades de un dispositivo de un mismo tipo, se imprimirían por pantalla dicho número de veces.

A demás, únicamente se mostrará el consumo individual (en kWh) y el estado del dispositivo en cuestión:

NOMBRE	CONSUMO	WORKING
bombilla	0.10	TRUE
ventilador	3.30	FALSE
TV	263.00	TRUE
PC	300.50	TRUE
bombilla	0.10	TRUE

Dado a que las otras dos opciones se explican por sí mismas, pasaremos a continuación a explicar el medio de comunicación entre el gestor y los sensores.

Memoria compartida: Sensores

Para permitir la comunicación con el programa que simula los datos obtenidos de los diversos sensores, el gestor crea una zona de memoria compartida de la que recibe el resultado de las diversas operaciones que se realizan en los sensores.

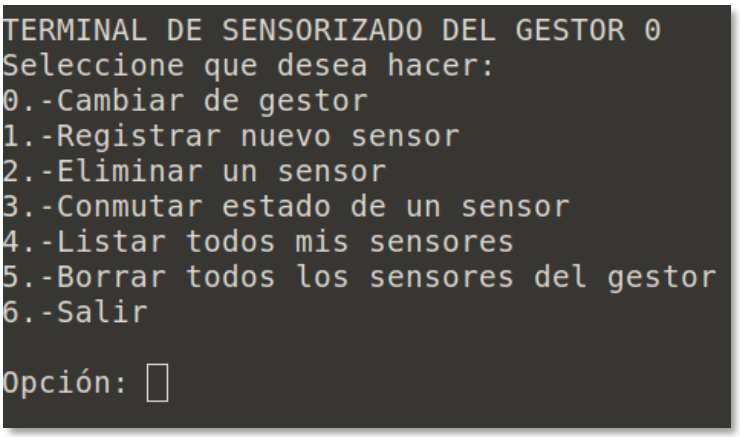
En el próximo apartado, nos adentraremos en el simulador de sensores, las diversas opciones que ofrece y el uso que puede hacer de él el usuario.

Simulador de Sensores

La función de este programa es la de enviar al gestor diversos cambios en cuanto a los sensores presentes en cada habitación, pudiendo añadir sensores nuevos, eliminarlos o conmutar su estado.

Al iniciar el programa, se nos abre una lista con los diversos gestores que están en ese instante conectados al servidor (esto lo hace consultando la zona memoria compartida de la que hemos hablado en el punto “memoria compartida: Gestores”) para decidir sobre cuál queremos trabajar.

Una vez hemos introducido el número del gestor en cuestión (en este caso, el número estará comprendido entre 0 y 4), se nos abre una interfaz con una serie de opciones de entre las cuales podemos escoger:



Opciones

Las opciones listadas anteriormente hacen lo siguiente una vez son seleccionadas:

- **0 – Cambiar de gestor**: Como el nombre indica, esta orden hace que se vuelvan a mostrar por pantalla los diversos gestores disponibles, y permite al usuario cambiar sobre el cual se está trabajando. Hay que tener en cuenta que la lista de dispositivos de cada gestor se almacena en el mismo y, por tanto, al cambiar de gestor la lista no se vacía en el anterior.
- **1 – Registrar un nuevo sensor**: pide al usuario que se introduzca por línea de comandos el nombre del dispositivo a añadir. El programa consulta la memoria compartida que lista los dispositivos existentes y, en caso de que el nombre introducido no coincida con ninguno de los ya existentes, se le pide al usuario que introduzca también su consumo en kWh.
- **2 – Eliminar un sensor**: cada sensor que se haya introducido tiene una ID. Se introduce la ID del sensor a eliminar y el programa lo borra.
- **3 – Conmutar estado de un sensor**: Se introduce la ID del sensor en cuestión, cuyo estado cambiará de ON a OFF o de OFF a ON.
- **4 – Listar todos mis sensores**: imprime por pantalla una lista con todos los sensores registrados en el gestor en cuestión que tiene la siguiente estructura:

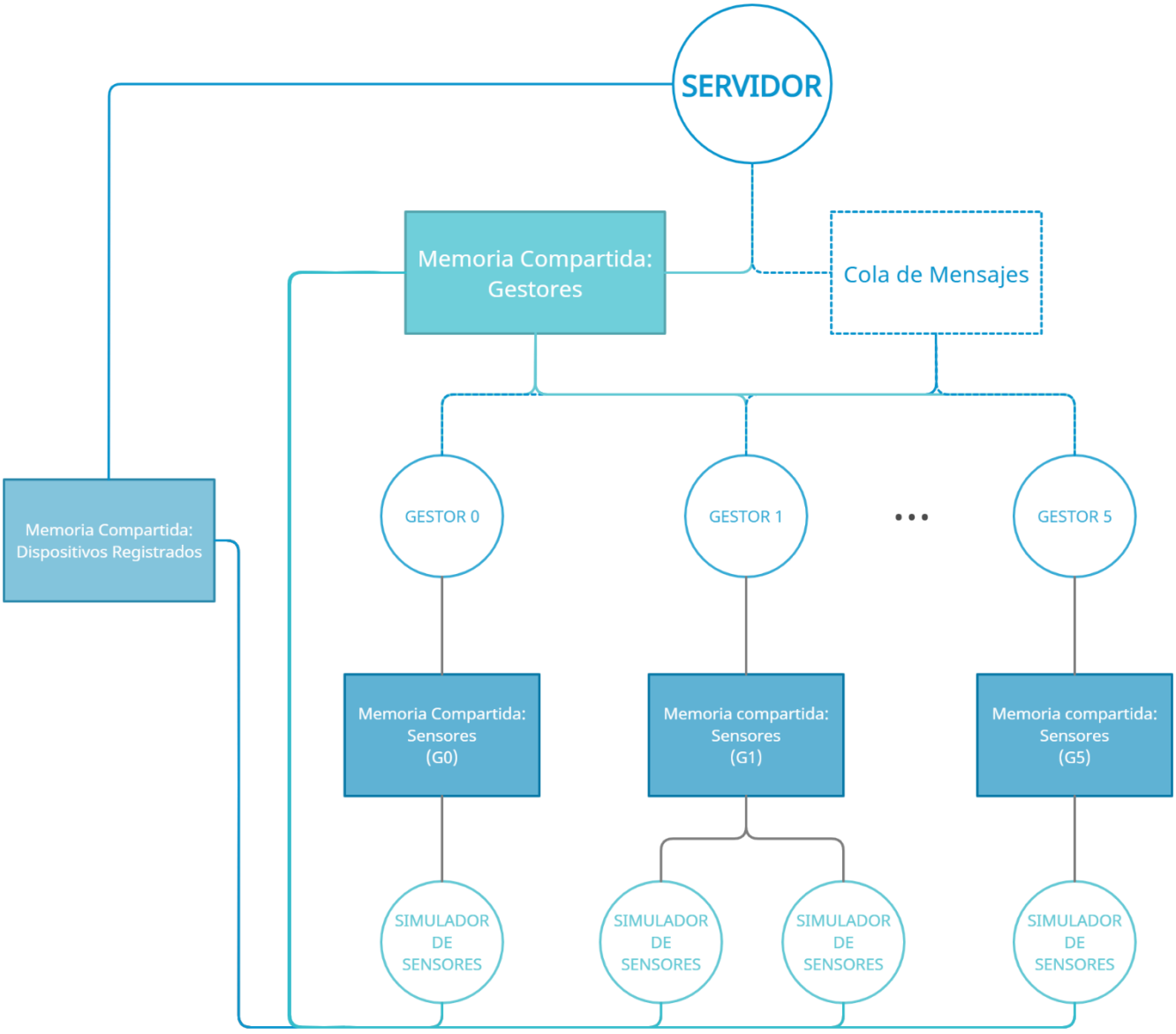
```
+--------+----------------+----------------+----------------+
| ID     | NOMBRE         | CONSUMO        | WORKING        |
+--------+----------------+----------------+----------------+
| 0      |     bombilla   |       0.10     |   TRUE         |
| 1      |   ventilador   |       3.30     |  FALSE         |
| 2      |            TV  |     263.00     |   TRUE         |
| 3      |            PC  |     300.50     |   TRUE         |
| 4      |     bombilla   |       0.10     |   TRUE         |
+--------+----------------+----------------+----------------+
```

- **5 – Borrar todos los sensores del gestor**: El propio nombre lo explica.
- **6 – Salir**: El programa termina. La tabla, al estar almacenada en el gestor, no se reinicia, por lo que múltiples programas pueden estar conectados a un mismo gestor.

RESUMEN

El sistema está compuesto por tres programas independientes que simulan el consumo de potencia de una serie de dispositivos en diversas habitaciones de una casa, siendo el servidor el encargado de mostrar por pantalla el resumen de todas las operaciones realizadas.

El sistema tiene la siguiente estructura, señalando las zonas de memoria compartida y colas como medios de comunicación entre procesos:



BIBLIOGRAFÍA

Proyectos que nos sirvieron de inspiración:

- https://www.home-assistant.io/
- http://misterhouse.sourceforge.net/
- http://www.openhab.org/

Empresas que nos ayudaron a descubrir la utilidad y las aplicaciones de la domótica:

- https://kimbosmart.com/
- https://es.futurehome.co/

Artículos destacados:

- https://computerhoy.com/listas/tecnologia/como-montar-smart-home-gastar-mucho-dinero-estos-5-dispositivos-suman-menos-200eu-526143
- https://www.electromaker.io/blog/article/9-best-raspberry-pi-smart-home-software-options