

Servicio de domótica
Grupo 3

Omar Álvarez Mures
Noelia Luaces Fernández
Adrián Morán García
Alfonso Nishikawa Muñumer
David Torres Andreu

1. Descripción del trabajo

1.1. Descripción del entorno de negocio

El proyecto consistirá en el desarrollo de un sistema que monitorice el estado de los diferentes sistemas domóticos que contiene una vivienda determinada dotándola de los principales servicios de gestión energética, seguridad y bienestar. El sistema monitorizará cada sensor de cada habitación, organizándose por zonas.

En el interior de la casa, controlaremos la climatización, iluminación, persianas y humo, y en la zona exterior de la vivienda (patio) controlaremos iluminación y tendremos un sensor de movimiento.

1.2. Requisitos funcionales

Un cliente se podrá conectar al servicio de monitorización para consultar el estado de los distintos sensores. Para ello el sistema se encargará de comprobar qué elementos están activos y cuáles no, permitiendo obtener los distintos valores de los sensores. El sistema debe permitir la agrupación de monitores según criterios del usuario.

1.3. Requisitos no funcionales

El sistema tratará de garantizar, en la medida de lo posible, la disponibilidad del servicio de lectura de sensores, siempre y cuando no sea un fallo propio del sensor.

El sistema debe poder escalar la cantidad de sensores a monitorizar garantizando el óptimo rendimiento del servicio.

Para garantizar el rendimiento, cuando los esclavos mueren, se crea un nuevo esclavo.

Asimismo, el sistema podrá soportar facilidad de prueba debido a sus capas, ya que permite la realización de las pruebas, teniendo sensores virtuales.

2. Objetivos concretos del trabajo

2.1. Arquitectura a aplicar

La arquitectura a utilizar en el sistema será de tipo Maestro-Eslavo, donde habrá un maestro principal, encargado de interactuar con el cliente, que controlará una subcapa de maestros que organizarán a los esclavos por grupos.

El maestro principal podrá levantar a los maestros de la capa inferior que agrupan a los monitores de sensores, en el caso de que se encuentren caídos. Los maestros de la subcapa podrán levantar a los monitores de sensores que a su vez se encuentren caídos.

Los monitores de sensores aligerarán la carga de procesado a los maestros, haciendo un filtrado de los valores del sensor en caso de que haya una diferencia importante en los valores. Es decir, el esclavo (monitor de sensor) informará al maestro en el caso de que haya un salto importante en los valores de un sensor determinado.

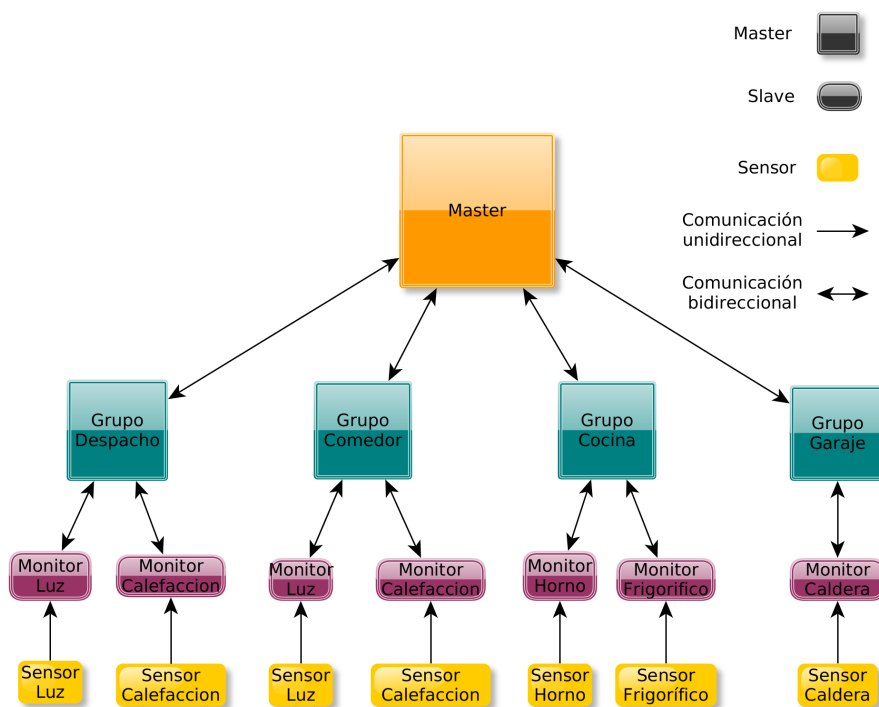


Figura 1: Esquema de la arquitectura a implantar

2.2. Tácticas a aplicar

1. Tácticas de disponibilidad

a) Detección de errores

Ping Se utilizará esta táctica entre los maestros y los grupos, para saber si es necesario levantarlos o no.

Heartbeat Los monitores de sensores harán uso de esta táctica para informar de que están levantados, y a mayores proporcionar información adicional, como por ejemplo el cambio brusco de valores de los sensores.

Excepciones Los componentes que se mueren lanzan excepciones y los componentes enlazados tratan dicha excepción y relanzan de nuevo los componentes que la han lanzado.

b) Recuperación de errores

Repuesto Se incluye un componente equivalente en función de si falla o no.

c) Prevención de errores

Monitor de procesos El master monitoriza el estado de los grupos y cada grupo monitoriza el estado de cada monitor de sensor.

2. Tácticas de flexibilidad al cambio

a) Localización de modificaciones

Coherencia semántica Los servicios comunes tienen el mismo tipo de componentes, como por ejemplo los grupos o los monitores de sensores.

Anticipación de cambios esperables Al aislar la lógica de la monitorización de sensores, su cambio no afecta a los grupos o al master, debido a que éstos sólo gestionan el uso de los monitores de sensores.

Generalización de cambios Se utiliza en conjunto con la anterior táctica, debido a que no podemos anticiparnos a todos los cambios. Intentamos generalizar lo máximo posible para finalmente reducir el impacto de los mismos.

b) Prevención de efecto dominó

Ocultación de información El maestro y los grupos delegan la lógica de los monitores de los sensores, esperando únicamente la respuesta; sin conocer el API concreto.

Uso de intermediarios El cliente se comunicará con los monitores a través del master, y éste con los sensores a través de los monitores.

c) Postergación de la ligadura

Ligadura dinámica Carga de código en caliente. Los módulos incluirán sentencias para actualizar a la última versión compilada del código.

3. Tácticas de rendimiento

a) Demanda de recursos

Gestión del ratio de eventos Reducimos la frecuencia con la que se comprueba el valor de los sensores por los monitores de sensores sólo para aquellos casos cuyo umbral de variación sea significativo.

Control de frecuencia de muestreo Se reduce la frecuencia de los eventos que le llegan a los grupos sin que éstos tengan control sobre su ratio de llegada.

Limitación del tiempo de ejecución Si se realiza una petición y ésta supera un timeout determinado, la petición se deshecha.

Limitación de colas En cada proceso se va a limitar el número de mensajes que pueden estar encolados.

b) Arbitraje de recursos

FIFO Las peticiones del cliente al master serán atendidas en el orden de llegada, garantizando que todas serán satisfechas.

4. Tácticas de facilidad de prueba

a) Gestión de entradas y salidas

Separación de interfaz/implementación Los sensores serán simulados.

5. Tácticas de usabilidad

a) En tiempo de ejecución

Soporte a la iniciativa del sistema El sistema enviará al usuario información sobre el estado de los distintos grupos y monitores de sensores.

3. Plan inicial

1. Tareas para lograr los objetivos

- Estudio de los requisitos
- Análisis de las tácticas a utilizar
- Análisis y diseño de la arquitectura escogida
- Desarrollo del proceso cliente
- Desarrollo del proceso maestro
- Desarrollo del proceso grupo
- Desarrollo del proceso monitor de sensor
- Desarrollo del simulador del sensor
- Desarrollo de los casos de prueba
- Ejecución de las pruebas
- Elaboración de la documentación

2. Herramientas y tecnologías a utilizar

- Lenguaje de programación Erlang
- Eclipse
- EUnit
- EDoc
- Cover
- LaTeX

3. Estimación de esfuerzo de cada tarea

Estudio de los requisitos	8h
Análisis de las tácticas a utilizar	8h
Análisis y diseño de la arquitectura a implantar	9h
Desarrollo del proceso cliente	20h
Desarrollo del proceso maestro	20h
Desarrollo del proceso grupo	20h
Desarrollo del proceso monitor de sensor	20h
Desarrollo del simulador del sensor	20h
Desarrollo de los casos de prueba	8h
Ejecución de las pruebas	8h
Elaboración de la documentación	16h

4. Lista de posibles riesgos durante el desarrollo

- Tareas subestimadas para el equipo actual por una falta de dominio en el lenguaje.
- Dentro de las tácticas escogidas, que haya tácticas que anulen parcialmente o resten beneficios a otras.