

Práctica 4 - Sistema Domótico IoT

Jose Luis Gallego Peña - Desarrollo de Sistemas Distribuidos (DSD2)

Introducción

Este Sistema Domótico IoT consiste en un servicio web que monitoriza la temperatura y la luminosidad de una casa, con las siguientes características:

- Existen tres servicios:
 - **cliente.html** actúa para el usuario y permite monitorizar la temperatura y la luminosidad, apagar/encender el aire acondicionado y abrir/cerrar la persiana. Además, muestra alertas si se pasan ciertos límites en los parámetros o si todo va bien también se indica.

Recibe los datos del servidor, actualizando los parámetros que mostrará al cliente. También envía datos al servidor en el caso de actuar sobre alguno de los actuadores, mediante formularios que ejecutan funciones javascript, y el servidor le responderá con los datos actualizados tras usar los actuadores, así como las alertas.

Parámetros de la casa

Temperatura (°C): 0°C

Luminosidad (0 - 100): 0

Control de los actuadores de la casa

Estado aire acondicionado: **APAGADO**

Apagar/Encender Aire Acondicionado ☐

- Apagar el aire acondicionado implica tener más temperatura
- Encender el aire acondicionado implica tener menos temperatura

Estado persiana: **CERRADA**

Cerrar/Abrir Persiana ☐

- Cerrar la persiana implica tener menos luminosidad
- Abrir la persiana implica tener más luminosidad

- **iot.html** actúa para un administrador, haciendo a modo de sensor manual. Permite introducir los valores que captarían los sensores de temperatura y luminosidad, mandándolos al servidor para que este los devuelva al cliente. Además, cada vez que se mete un nuevo valor en los sensores, se notifica al servidor y este almacena junto a una marca de tiempo en una base de datos que gestiona.

Informa de lo mismo que el cliente: valores de los parámetros, alertas y estado de los actuadores, pero no puede actuar sobre ellos.

El histórico de todos los cambios en los valores de los sensores se obtiene del servidor haciéndole una llamada. El histórico puede hacerse muy grande, así que se ha incluido que si la lista se hace muy larga no ocupe toda la pantalla y se pueda hacer scroll. Además, existe un botón para borrar por completo el histórico.

Sensores

Temperatura (°C):

Luminosidad (0 - 100):

0 0
Estado persiana: **CERRADA**
Estado aire acondicionado: **APAGADO**

Histórico de cambios en las medidas de los sensores

- **iot.js**, es el servidor nodejs y es el que se ocupa de lanzar los dos servicios web anteriores. Al igual que los servicios web, define varios eventos de socketio para centralizar las distintas llamadas entre ambos servicios web. Se ocupa principalmente de que los clientes se puedan conectar satisfactoriamente y de despachar los distintos eventos.

Gestiona también una base de datos MongoDB con el histórico de las actualizaciones en los sensores. Además, almacena en una variable el estado actual de los parámetros.

Para ejecutar los ejemplos ha sido necesario instalar nodejs, socket.io y mongodb. Para iniciar el servidor IOT que tiene ambas páginas web se introduce el siguiente comando en la terminal:

```
node iot.js
```

Esto despliega dos servicios web, el cliente con `localhost:8080/cliente` y los sensores con `localhost:8080/iot`. Es decir, ambos servicios están en el puerto 8080, con distinta URI.

Eventos

Para entender bien el funcionamiento de todo, hay que hablar de los eventos de socketio que se usan para las comunicaciones entre las distintas partes del sistema.

- **connection** se llama cuando se conecta un cliente y el resto de eventos cuelgan de este. Es decir, si un cliente no está conectado, no puede usar ninguno de los otros eventos.
- **disconnect** se llama cuando un cliente se desconecta y se muestra un mensaje por terminal informando de esto. En el caso de los clientes, cuando esto pasa se informa mediante un mensaje de servicio de que el servidor está caído o ha habido una desconexión de este por cualquier razón.
- **medidas-casa**, se llama nada más se conecte un cliente para enviarle los datos de sensores que están activos en el momento. También se llama cuando los sensores captan un nuevo dato. Los clientes escuchan de este evento para actualizar sus datos.
- **obtener-medidas**, evento para que el cliente iot actualice la lista del histórico del sensor.
- **alerta**, evento que reciben los clientes cuando el servidor les manda un mensaje de alerta, en el caso de que se haya sobrepasado algún límite, especificando cual, o si no se ha sobrepasado ninguno especificando que está todo bien.
- **sensores**, es el evento más importante del sistema. El cliente iot cuando introduce nuevos valores, llama a una función que actualiza esos valores y los envía mediante este evento. El servidor, al recibir este evento, gestiona la inserción en la base de datos, comprueba de forma compleja si los valores de temperatura y luminosidad sobrepasan ciertos umbrales. Comprueba umbrales superiores, inferiores, o

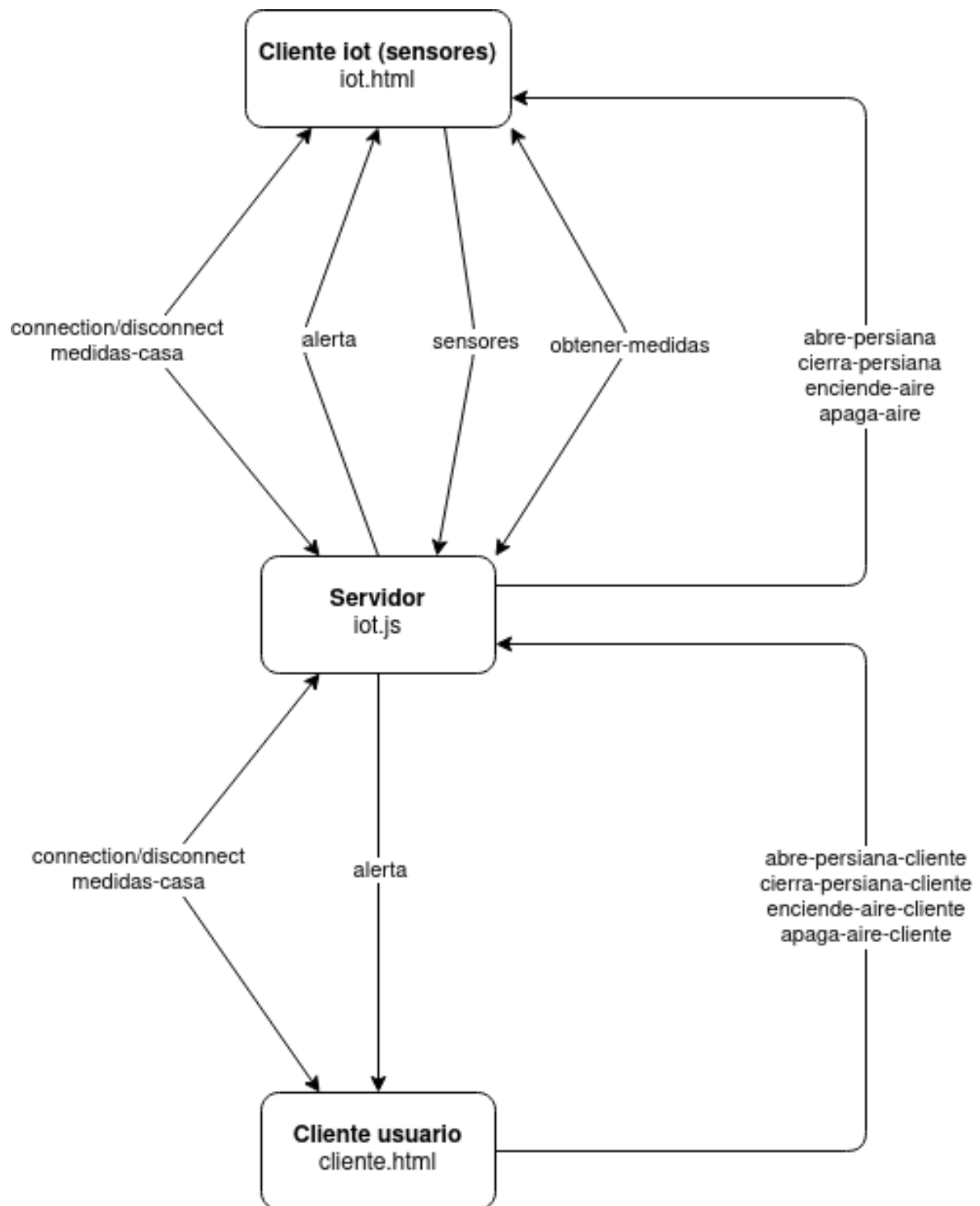
ambos, teniendo en cuenta ambos parámetros o sólo uno, y en función de eso envía varios eventos, que son los siguientes:

- **cierra-persiana**, se envía cuando se detecta una luminosidad muy alta, para así bajarla un número aleatorio de unidades.
- **abre-persiana**, se envía cuando se detecta una luminosidad muy baja, para así aumentarla un número aleatorio de unidades.
- **apaga-aire**, se envía cuando se detecta una temperatura muy baja, para así aumentarla un número aleatorio de unidades.
- **enciende-aire**, se envía cuando se detecta una temperatura muy alta, para así disminuirla un número aleatorio de unidades.

Todos estos eventos descritos cambian lo que se ve en pantalla, indicando el estado de los actuadores. Se pueden llamar uno o más de uno en función de los parámetros que reciba el servidor. Además, estos eventos se pueden “forzar”, es decir, de normal estos se activan cuando se sobrepasan ciertos valores (esto es el agente) sin embargo el cliente puede deciri apagarlos o encenderlos cuantas veces quiera para aumentar o disminuir los valores cuantas veces quiera. Se hace con los siguientes eventos:

- **cierra-persiana-cliente**
- **abre-persiana-cliente**
- **apaga-aire-cliente**
- **enciende-aire-cliente**

En el siguiente diagrama se resume el flujo de los distintos eventos que se han explicado, con flechas indicando quién llama a quién y con qué evento:




Ejemplo de funcionamiento

Como hemos visto en las anteriores imagenes, el estado inicial es 0, vamos a introducir nuevos valores en los sensores.

Parámetros de la casa

Las medidas están en unos valores correctos

Temperatura (°C):  50°C

Luminosidad (0 - 100):  50

Control de los actuadores de la casa

Estado aire acondicionado: APAGADO

Apagar/Encender Aire Acondicionado ☐

Aplicar

- Apagar el aire acondicionado implica tener más temperatura
- Encender el aire acondicionado implica tener menos temperatura

Estado persiana: CERRADA

Cerrar/Abrir Persiana ☐

Aplicar

- Cerrar la persiana implica tener menos luminosidad
- Abrir la persiana implica tener más luminosidad

Sensores

Temperatura (°C):

Luminosidad (0 - 100):

Cambiar valores

50 50

Estado persiana: CERRADA

Estado aire acondicionado: APAGADO

Las medidas están en unos valores correctos

Histórico de cambios en las medidas de los sensores

Borrar histórico

- Temperatura: 50 Luminosidad: 50 Tiempo: 2020-06-01T12:33:51.217Z

Al hacerlo, se actualiza en ambos y se ve que los valores están dentro de los límites. Si ahora por ejemplo encendemos el aire acondicionado, vemos como la temperatura baja. Cada vez que le demos a aplicar, bajará un poco más, y en el caso de que sobrepase el límite, irá aumentando la temperatura poco a poco de forma automática hasta llegar a un valor normal, pero esto es más difícil de enseñar con capturas.

Parámetros de la casa

Las medidas están en unos valores correctos

Temperatura (°C):  35.24°C

Luminosidad (0 - 100):  50

Control de los actuadores de la casa

Estado aire acondicionado: ENCENDIDO

Apagar/Encender Aire Acondicionado ☒

Aplicar

- Apagar el aire acondicionado implica tener más temperatura
- Encender el aire acondicionado implica tener menos temperatura

Estado persiana: CERRADA

Cerrar/Abrir Persiana ☐

Aplicar

- Cerrar la persiana implica tener menos luminosidad
- Abrir la persiana implica tener más luminosidad

Sensores

Temperatura (°C):

Luminosidad (0 - 100):

Cambiar valores

35.24 50

Estado persiana: CERRADA

Estado aire acondicionado: ENCENDIDO

Las medidas están en unos valores correctos

Histórico de cambios en las medidas de los sensores

Borrar histórico

- Temperatura: 50 Luminosidad: 50 Tiempo: 2020-06-01T12:33:51.217Z
- Temperatura: 35.24 Luminosidad: 50 Tiempo: 2020-06-01T12:37:06.851Z