

Memoria Sistema Domótico

(NodeJS y Socket.io)

En esta práctica se nos pedía simular un sistema domótico en el que hubiese dos sensores, uno de temperatura y otro de luminosidad, y dos botones, uno para abrir/cerrar las persianas y otro para encender/apagar el aire acondicionado.

Para esto vamos a utilizar nodeJS y Socket.io comunicando el servidor con el cliente y los sensores.

En el servidor creado con `http.createServer()` se renderiza la página del cliente. Tras esto lanzamos el cliente de mongoDB en el puerto 27017 con:

```
MongoClient.connect("mongodb://localhost:27017/", function(err, db) {
```

Con esto vamos a empezar una sucesión de call-backs anidados.

También dejamos el servidor a la escucha en el puerto 8080: `httpServer.listen(8080);`

Creamos una base de datos llamada “sistema_somotico” y dentro una colección “sensores”.

Tras esto el servidor queda a la escucha hasta que reciba una conexión, una vez que se hayan conectado al servidor, este vuelve a quedar a la escucha esperando diferentes llamadas:

- **client.on('poner-temperatura', function (data)):** En esta función metemos en la base de datos una entrada con la temperatura que ha enviado el sensor y la hora en la que ha registrado dicha temperatura. Se realiza también un emit llamado “actualizar-temperatura” al que se le pasa la temperatura para que el cliente pueda actualizar el visor en el que se muestra este dato. También se comprueba que no se superen los límites de temperatura, en caso de ser así se envía una alerta al cliente. Si se supera tanto el límite de temperatura como de luminosidad se cierran las persianas en caso de estar abiertas.
- **client.on('poner-luminosidad', function (data)):** Similar al anterior pero con la luminosidad. Metemos en la base de datos una entrada con la luminosidad que ha enviado el sensor y la hora en la que ha registrado dicha luminosidad. Se realiza también un emit llamado “actualizar-luminosidad” al que se le pasa la luminosidad para que el cliente pueda actualizar el visor en el que se muestra este dato. También se comprueba que no se superen los límites de luminosidad, en caso de ser así se envía una alerta al cliente. Si se supera tanto el límite de temperatura como de luminosidad se cierran las persianas en caso de estar abiertas.
- **client.on('cambiar-persianas', function ()):** Se llama a una función que alterna el estado de las persianas entre abiertas y cerradas. Después se hace un emit “actualizar-persianas” para enviar la actualización al cliente y que refleje el cambio.
- **client.on('cambiar-aire', function ()):** Se llama a una función que alterna el estado del aire acondicionado entre encendido y apagado. Tras esto, se hace un emit “actualizar-aire” que envía el estado del aire acondicionado al cliente.
- **client.on('obtener-temperatura', function ()):** Se hace un emit “actualizar-temperatura” para enviar el dato al cliente.
- **client.on('obtener-luminosidad', function ()):** Se hace un emit “actualizar-luminosidad” para enviar el dato al cliente.

- **client.on('obtener-persianas', function ()):** Se hace un emit “actualizar-persianas” para enviar el dato al cliente.
- **client.on('obtener-aire', function ()):** Se hace un emit “actualizar-aire” para enviar el dato al cliente.

El cliente es un fichero html con código javascript. El html muestra la información correspondiente al estado de la temperatura, luminosidad, persianas y aire acondicionado. También muestra dos botones para que el usuario pueda alternar los estados de las persianas y el aire acondicionado. Estos botones ejecutan funciones que envían emit “cambiar-persianas” y “cambiar-aire” respectivamente.

En el código JS obtenemos el socket que comunica con el servidor con las siguientes líneas de código:

```
var serviceURL = document.URL;  
var socket = io.connect(serviceURL);
```

Inmediatamente le pedimos al servidor que nos proporcione los datos correspondientes para poder actualizarlos en el html realizando cuatro emits: “obtener-temperatura”, “obtener-luminosidad”, “obtener-persianas”, “obtener-aire”.

Luego el cliente se queda a la escucha esperando llamadas desde el servidor con la información que actualizará en el html mediante la función “innerHTML”.

El usuario puede pulsar un botón que le redirige a la página de sensores donde podrá introducir unos valores en los sensores de temperatura y luminosidad simulando así que esos valores los han captado realmente los sensores.

En esta página de sensores habrá dos formularios, uno para la temperatura y otro para la luminosidad, al enviarlos se ejecutará una función que obtendrá la hora y enviará la temperatura o luminosidad (dependiendo del formulario) con su correspondiente hora de captación al servidor mediante un emit “poner-temperatura” o “poner-luminosidad”.