
PRÁCTICA 4
SISTEMA DOMÓTICO

Tercer Curso - Grado en Ingeniería Informática
Curso 2019-2020

PEDRO SERRANO PÉREZ

Contents

1	Introducción	1
2	Servidor	1
3	Sensores	2
3.1	Capturas de pantalla	3
4	Usuarios	4
4.1	Capturas de pantalla	5
5	Anotaciones del contenido entregado	7

1 Introducción

En esta práctica se va a desarrollar un sistema domótico que va a estar compuesto por dos sensores, uno de luminosidad y otro de temperatura, un servidor, que incluye un agente que tomará ciertas decisiones en base a las medidas, y los actuadores (motor de la persiana y sistema de aire acondicionado).

La comunicación entre las distintas partes se va a realizar mediante nodejs y socket.io.

2 Servidor

Para el funcionamiento del servidor se van a crear dos servidores http (`http.createServer`).

Uno de los servidores estará escuchando en el puerto 8080 y se encargará de servir la página que simulará a los sensores. El otro servidor http escuchará en el puerto 8081 y servirá la página de los usuarios a estos.

El servidor utiliza una base de datos MongoDB (llamada `sistemaDomotico_P4` y las colecciones `temperatura` y `luminosidad`). En las colecciones de la base de datos se guardará el histórico con el valor de la medida y la fecha en la que fue emitida.

En cuanto a la funcionalidad del servidor, este se va a encargar de escuchar por ciertos eventos y actuar en consecuencia (emitiendo un evento, llamando al agente, ...).

En primer lugar, cuando sirve cualquiera de las páginas (sensores o usuarios) envía una información inicial a estas:

- Sensores: El servidor emite el estado actual de los actuadores.
- Usuarios: El servidor emite un mensaje con los datos actuales de las medidas (si no existen estos datos se emite el mensaje correspondiente), el estado de los actuadores, los valores actuales de los umbrales (que controla el Agente), las últimas 6 entradas del histórico de temperatura y luminosidad de la base de datos y, las alertes actuales de luminosidad y temperatura, si las hay.

El resto de aspectos de los que se encarga el servidor serían:

- Escuchar los cambios en las modificaciones de las medidas de temperatura y luminosidad. El servidor, tras obtener una nueva medida la inserta en el histórico de la colección de la base de datos correspondiente, emitiría esta medida a los usuarios y llamaría al agente para comprobar la nueva medida con los umbrales.
Además se obtienen de la base de datos los 6 últimos valores, con su fecha, y se envían a los usuarios (esto se ha hecho así para ver el funcionamiento de las consultas a una base de datos MongoDB, aunque, claramente, es más ineficiente que mandar el valor con la fecha que acaba de recibir).
- Escucha los cambios en los estados de los actuadores. Cuando se produce un cambio en el estado de uno de los actuadores, se notifica este cambio a usuarios y sensores.
- Escucha cambios en los umbrales mínimos y máximos de luminosidad y temperatura. En este caso se llama al agente, y este gestiona el proceso.

El agente del servidor es una clase que posee los métodos y atributos necesarios para la toma de decisiones en función de los valores de las medidas. Posee información de las medidas actuales y los umbrales máximos y mínimos de estas. Parte de esta información la envía a los usuarios cuando el servidor les sirve la página.

Cuando se produce algún cambio en las medidas se comprueban los siguientes aspectos:

- Comprobar la temperatura actual. Si la temperatura actual supera el umbral mínimo se enviará una alerta a los usuarios. Además, encenderá de forma automática el aire acondicionado (si está apagado) y lo notificará a los sensores y usuarios.
- Comprobar la luminosidad actual. Si la luminosidad actual supera el umbral mínimo se enviará una alerta a los usuarios.
- Comprobar máximos. Se comprueban los umbrales máximos. Si ambos se superan y la persiana está subida, esta se bajará y se alertará a los clientes. Se notifica el cambio de estado de la persiana.

Además, cuando se modifican los umbrales, se vuelven a comprobar las restricciones anteriores correspondientes al umbral modificado.

3 Sensores

En este caso tenemos dos sensores, uno de temperatura y otro de luminosidad. La página que proporciona el servidor permite el envío de nuevas medidas de las siguientes formas (siempre teniendo en cuenta que el mínimo de luminosidad es 0, y su máximo 100, y el mínimo de temperatura será 30, y su máximo 70):

- Escribiendo el valor y enviándolo.
- Activando una opción de generación automática. Se generan nuevos valores cada 5 segundos.

La opción de generación automática consta de dos partes:

- Luminosidad. Inicialmente comienza en 50. Si la persiana está bajada el valor de esta será de 0. Si está subida aumentará o decrementará de forma aleatoria en un rango (entre $\text{luminosidadActual}-5$ y $\text{luminosidadActual}+5$).
- Temperatura. Para obtener la temperatura se hace uso de una API de OpenWeather: Current Weather Data. Se realizará una petición http a la url:

`http://api.openweathermap.org/data/2.5/weather?id=idCiudad&appid=key`

Donde idciudad se va a corresponder con el id de Granada y key será una clave asignada una vez te registras en la página.

La página devuelve la información en formato JSON, por lo que, una vez obtenemos la temperatura, aplicamos un filtro para variarla en función del estado del aire acondicionado. Si está encendido la temperatura se bajará hasta 10 grados (de forma aleatoria) y, si está apagado, aumentará hasta 4 grados.

Por último los sensores escuchará por cambios en los estados de los actuadores.

3.1 Capturas de pantalla

A continuación se muestran algunas capturas de pantalla de la página de sensores.

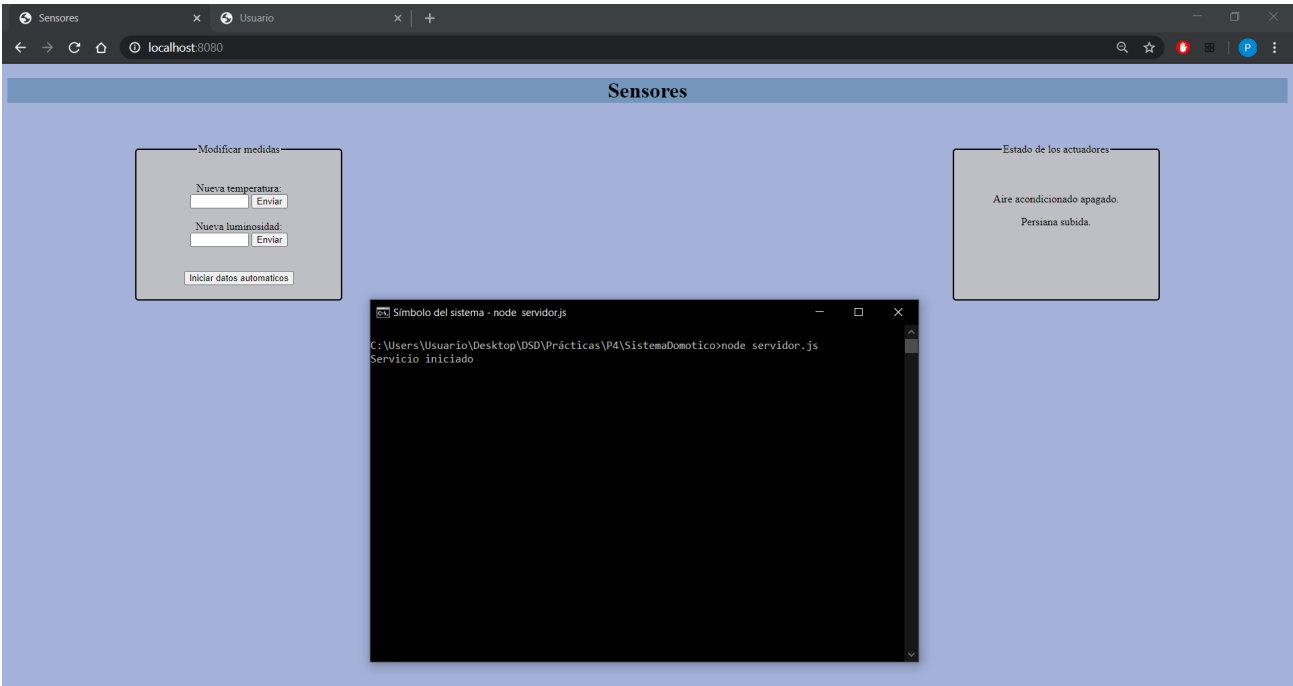


Figure 1: Captura de pantalla del estado inicial de la página de sensores

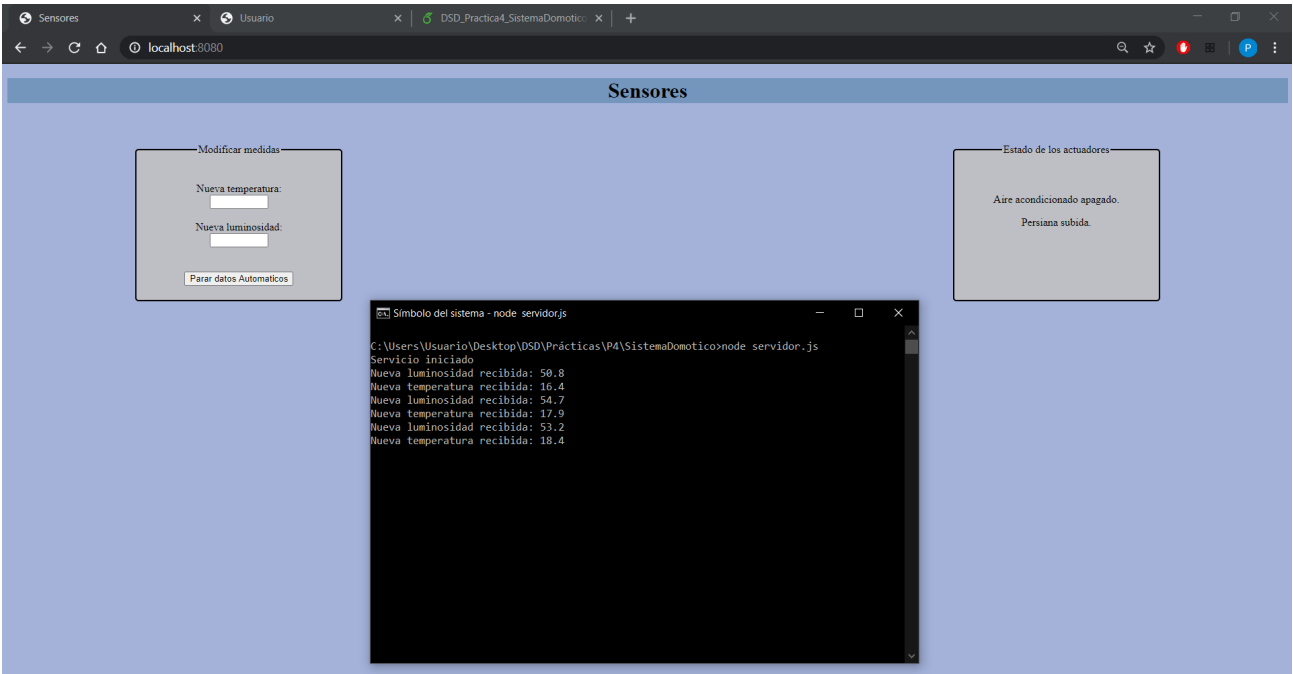


Figure 2: Captura de pantalla con la generación automática de medidas

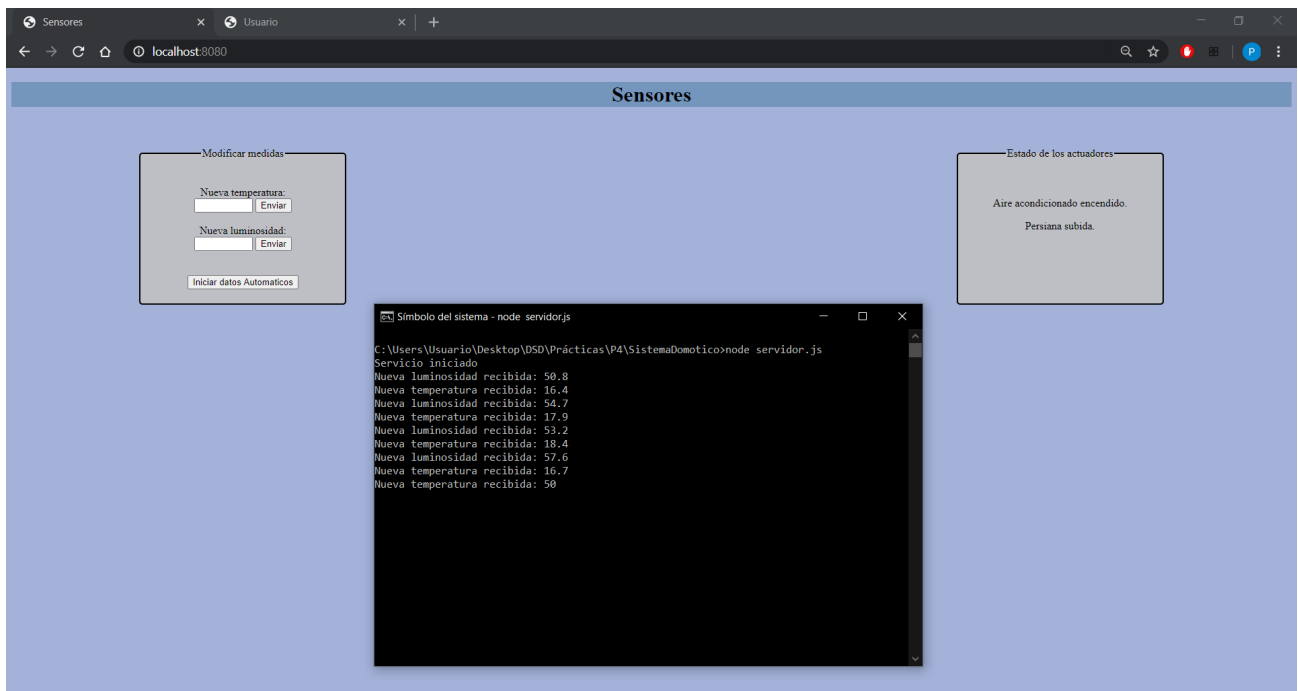


Figure 3: Captura de pantalla con temperatura que produce la activación del aire acondicionado

4 Usuarios

La principal finalidad de esta página es informar a los usuarios y permitir el control remoto, tanto de los actuadores en sí, como de los umbrales máximos y mínimos de las medidas.

Esta parte se va a encargar de;

- Escuchar las medidas actuales de temperatura y luminosidad.
- Escuchar los historiales de los valores de las 6 últimas medidas (con los que generará y actualizará unas gráficas).
- Escuchar el estado actual de los actuadores y emitirlo si desea cambiarse.
- Escuchar por las modificaciones de los umbrales y enviar estas modificaciones, si desean cambiarse los umbrales actuales. En caso de modificar los umbrales, la página no permite enviar un umbral mínimo mayor que el máximo (ni viceversa), ni establecer un umbral por debajo del mínimo de la medida o por su máximo (valores comentados en la sección de sensores).
- Escuchar por las alertas que envía el Agente del servidor.

Como ya se ha mencionado esta página representa los historiales de luminosidad y temperatura en gráficas. Estas gráficas se generan utilizando **Chart.js**. Los valores que puede tomar la medida se muestran en el eje Y, y la fecha en la que fue emitida en el eje X. Estas gráficas se actualizan de forma animada cada vez que se recibe un nuevo historial, utilizando el metodo **update()** que proporciona Chart.js

4.1 Capturas de pantalla

A continuación se muestran algunas capturas de pantalla de la página de usuarios.



Figure 4: Captura de pantalla de la página del usuario

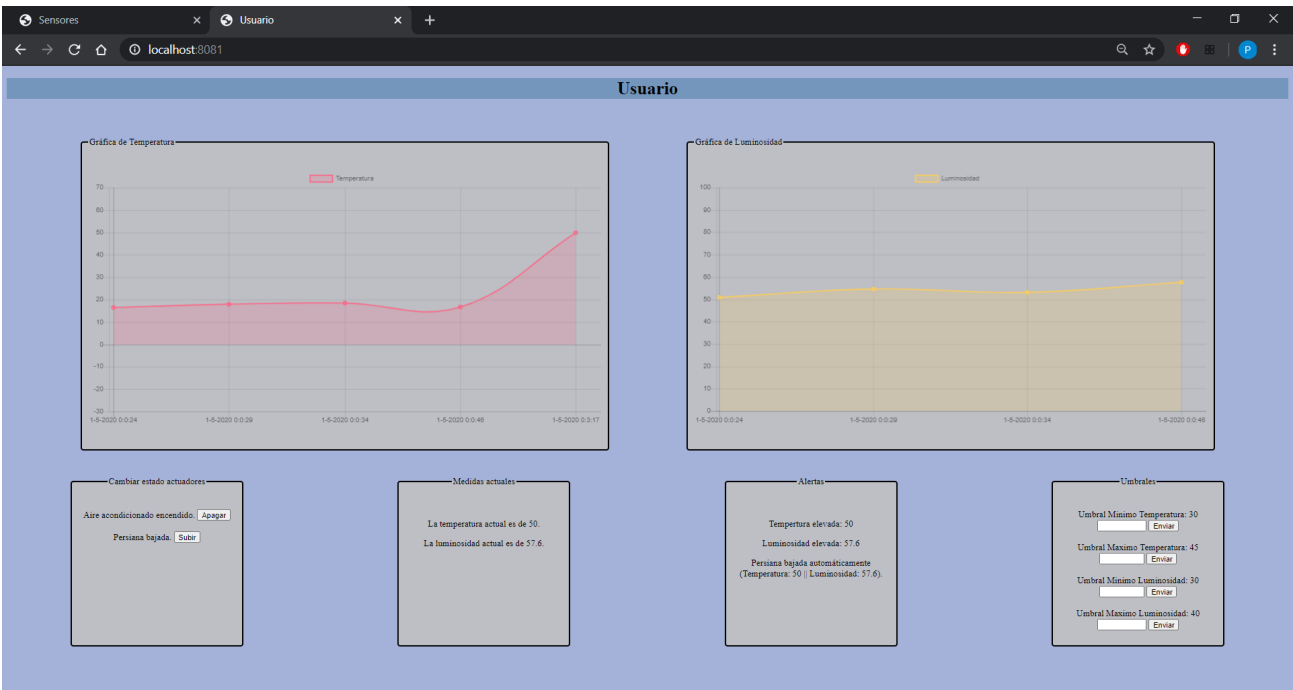


Figure 5: Página del usuario modificando umbrales para que se baje de forma automática la persiana

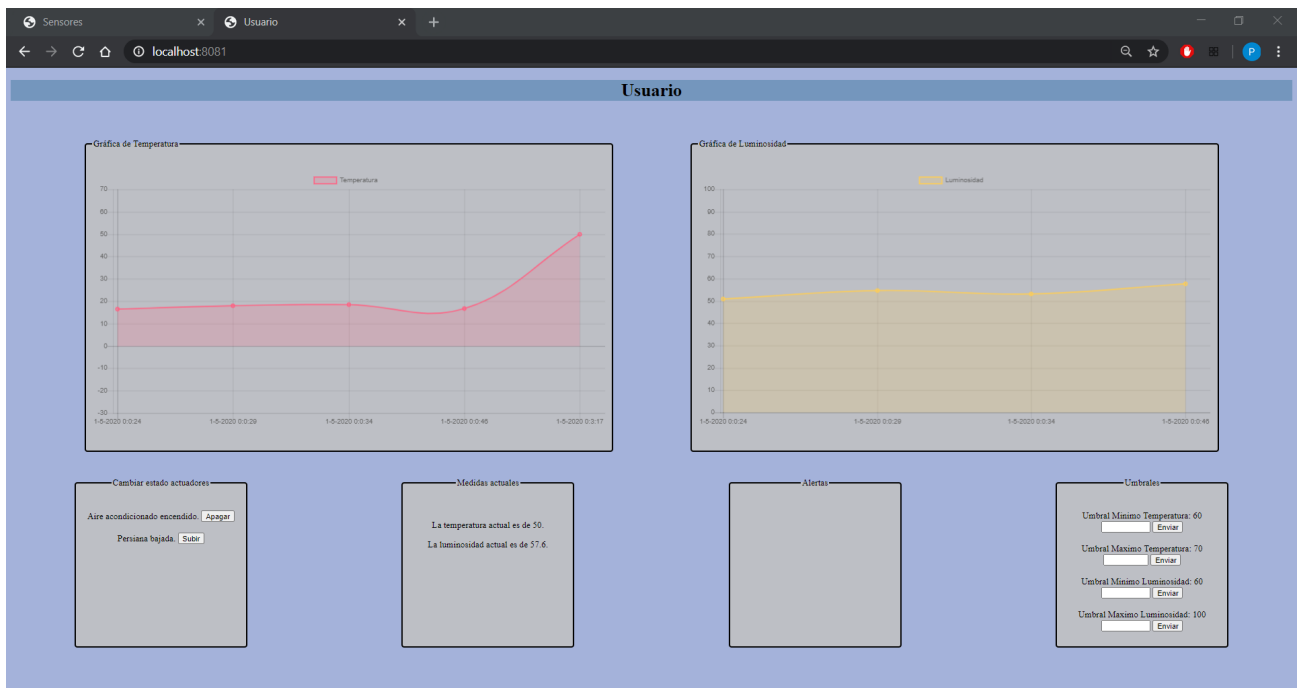


Figure 6: Página del usuario modificando umbrales alejarlos de las medidas actuales y eliminar alertas

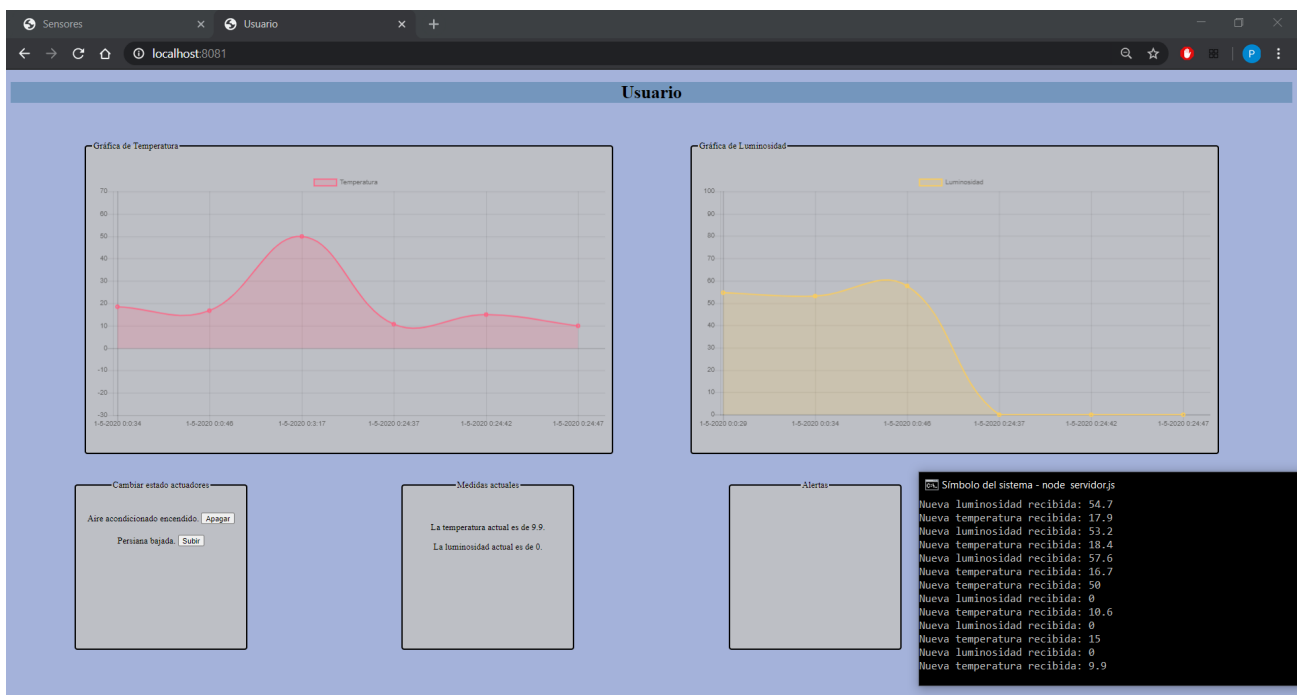


Figure 7: Actualización de las gráficas con sensores en generación automática (persiana bajada)

5 Anotaciones del contenido entregado

- Para evitar errores en el servidor con las peticiones se ha añadido un archivo favicon.ico (en la carpeta icon), aunque este no se utiliza.
- Se proporcionan las dos colecciones utilizadas para la práctica (luminosidad y temperatura) en formato csv.
- Se proporciona la carpeta node_modules.