



UNIVERSIDAD DE GRANADA

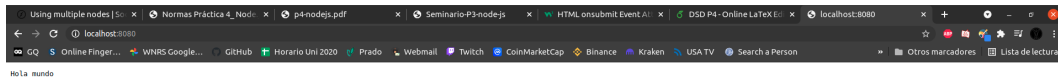
Práctica DSD: Node.js

Fernando Lojano Mayaguari

Universidad de Granada
Granada
Mayo 2021

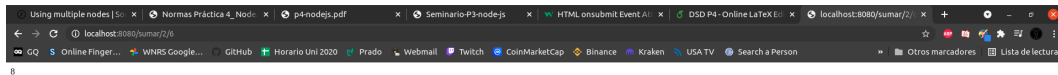
1 Parte 1: Implementación de los ejemplos

Los primeros ejemplos que se nos muestran como prueba para hacernos una idea son básicos. El primero de ellos, el ejemplo helloworld.js únicamente muestra en la dirección localhost:8080 la frase "Hola mundo". Se incluye una captura con el resultado:



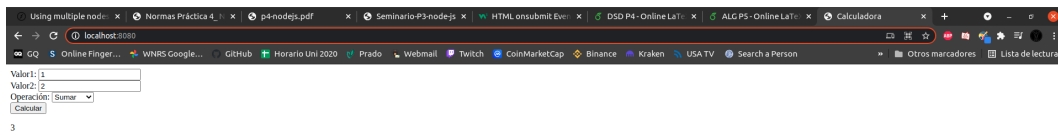
Helloworld.js

Posteriormente se nos muestran varios ejemplos de implementación de calculadoras. La primera implementación y la más básica versión consiste en una calculadora implementada en el servidor y la forma de introducir valores es mediante la url, de forma que si queremos realizar la suma de $6 + 2$ en la url deberíamos añadir: "suma/2/6". La captura del resultado obtenido es la siguiente:



calculadora.js

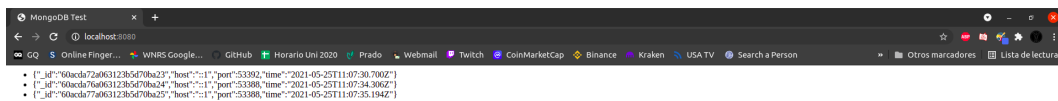
La versión mejorada de calculadora nos introduce un archivo html en el cual se implementa una interfaz de uso. Podemos ver en la captura siguiente como se muestra la calculadora con interfaz:



calculadoraWeb.js

El último ejemplo de prueba que se nos propone es para comprobar la integración de nodejs con la aplicación mongodb. En este caso, se nos muestra en el navegador todas las conexiones que se hacen a localhost mientras que la prueba se está ejecutando. Esto lo consigue ya que se crea una entrada en una base de datos de prueba cada vez cada vez que se accede a localhost:8080. De esta forma, podemos guardar un historial de todas las entradas que realicemos.

Se muestra a continuación una captura con los resultados mostrados por pantalla.



Integracion MongoDB

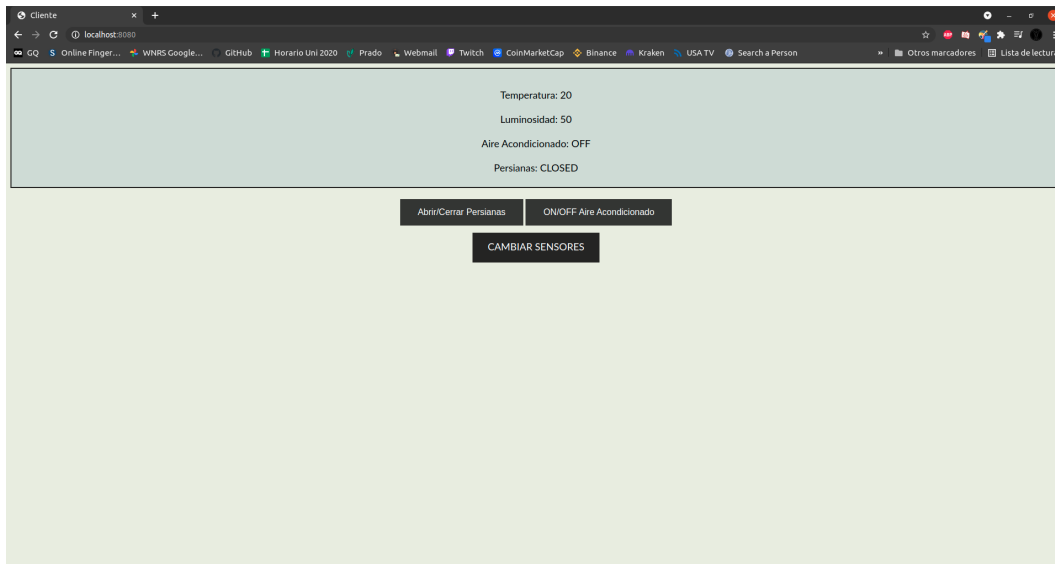
2 Parte 2: Sistema domótico

En este apartado, se nos pide que implementemos un simulador de un sistema domótico, de forma que, por pantalla del cliente se muestre un seguimiento de los estados de la temperatura y luminosidad de una casa, así como la monitorización de la apertura y cerradura de las ventanas y el apagado y encendido del aire acondicionado.

Con este fin, se han creado cuatro ficheros diferentes:

2.1 cliente.html

En este fichero se realiza la monitorización de los diferentes estados que pueden tomar alguno de las variables que estamos monitorizando. Además, se incluyen en esta página las opciones de cerrar/abrir las ventanas y de apagar/encender el aire acondicionado. A simple vista no se ve pero en la parte inferior de la página se encuentran dos apartados de notificaciones que se disparan cuando intentamos introducir unos valores para los sensores que se encuentran fuera del rango permitido. A continuación, se incluye una captura de como se ve la pagina Cliente.



Cliente.html

En este archivo, se han incluido los siguientes eventos de escucha:

- `socket.on('updateTemperatura', function(data)`
- `Socket.on('updateLuminosidad', function(data)`
- `socket.on('updateAire', function(data)`
- `socket.on('updatePersianas', function(data)`
- `socket.on('TemperaturaNotificacion', function(data)`
- `socket.on('updateTemperatura', function(data)`
- `socket.on('LuminosidadNotificacion', function(data)`

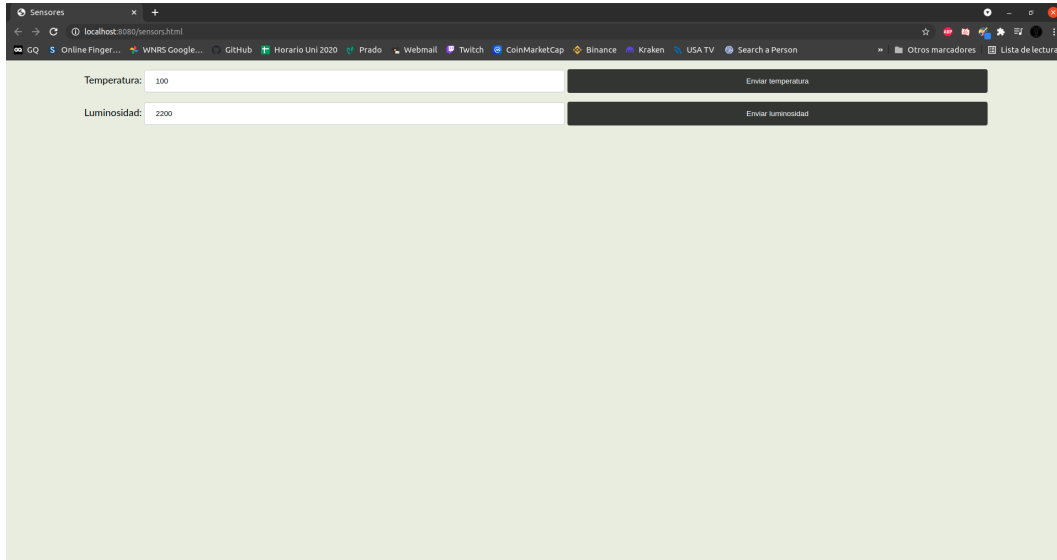
Todos estos sockets de escucha sirven para poder actualizar los apartados cada vez que se realiza algún cambio sobre esos datos.

2.1.1 Sensor.html

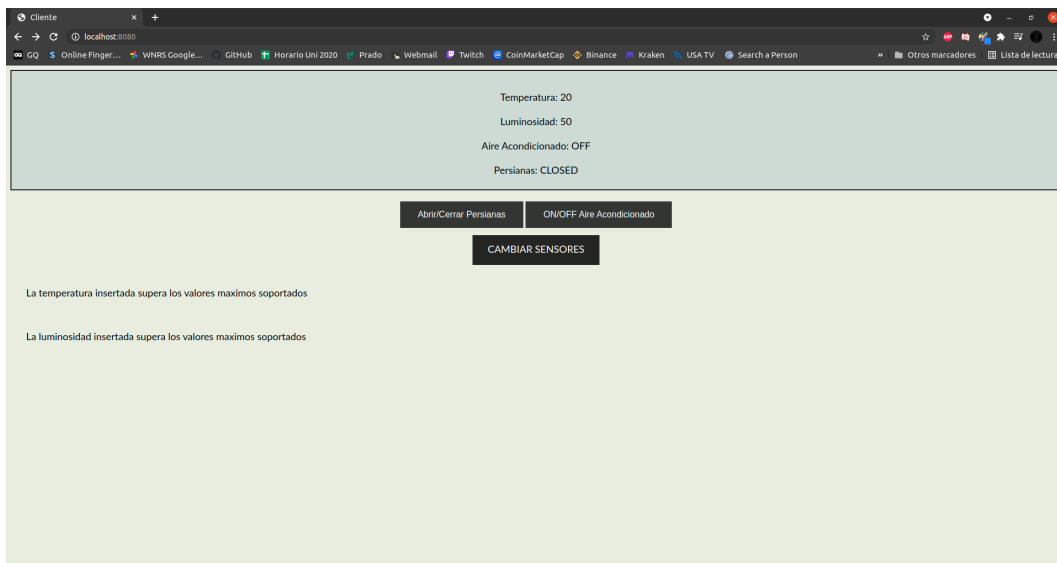
En esta sección se muestran las acciones disponibles sobre los sensores. En este caso, se disponen de dos apartados, uno de introducción de valor para la temperatura y otro para la introducción de un valor de la luminosidad. En este archivo se han incluido las funciones de envío de datos de temperatura y luminosidad:

- function sendTemperatura()
- function sendLuminosidad()

Estas funciones nos sirven para enviar los valores que deseamos al servidor, en este caso, si introducimos un valor que sobrepasa el rango establecido para los valores (para temperatura: 10-50, para luminosidad:0-100), en la sección de cliente se disparan las notificaciones que hemos nombrado anteriormente.



sensor.html



Salto notificaciones

Como podemos ver, las dos notificaciones han saltado pues los dos valores se encuentran fuera de rango. En este caso, los valores no cambian.

2.2 server.js

En esta parte se realiza la mayor parte de las funcionalidades, aquí se han incluido varios sockets para el envío de mensajes (socket.emit) y varias funciones para poder recibir los cambios que se realicen. Además, se pedía en la

práctica que se guardasen los cambios realizados en los sensores, para esto, se hace uso de la integración de mongodb con nodejs. La conexión a la base de datos se realiza de la misma forma que se presenta en el seminario ofrecido para esta práctica. Para poder introducir los valores de los sensores que se están introduciendo en la página sensor.html se han implementado las funciones insertTemperatura e insertLuminosidad. Dentro de estas mismas funciones se realiza la comprobación para poder decidir si el valor está dentro de los rangos permitidos. Como hemos dicho antes, en caso de que los valores estén fuera de rango, se enviará el mensaje de alerta a la página cliente mediante la función emit de socket. En la página cliente se disparará la alerta puesto que hemos implementado en ella un socket de escucha para estas notificaciones.

Además de estas funciones, se implementaron los métodos necesarios para poder cambiar el estado de las persianas y del aire acondicionado. En este caso, para determinar como cambiar el estado de los aparatos se hace una comprobación de los valores actuales de los estados y se cambian por los valores opuestos, es decir, si el ventilador estaba apagado (OFF), se enciende (ON) y si las persianas estan abiertas(OPENED), se cierran(CLOSED). Pasa lo mismo para el caso inverso. Para notificar de los cambios que se produzcan, también se ha usado la función emit en las dos funciones.

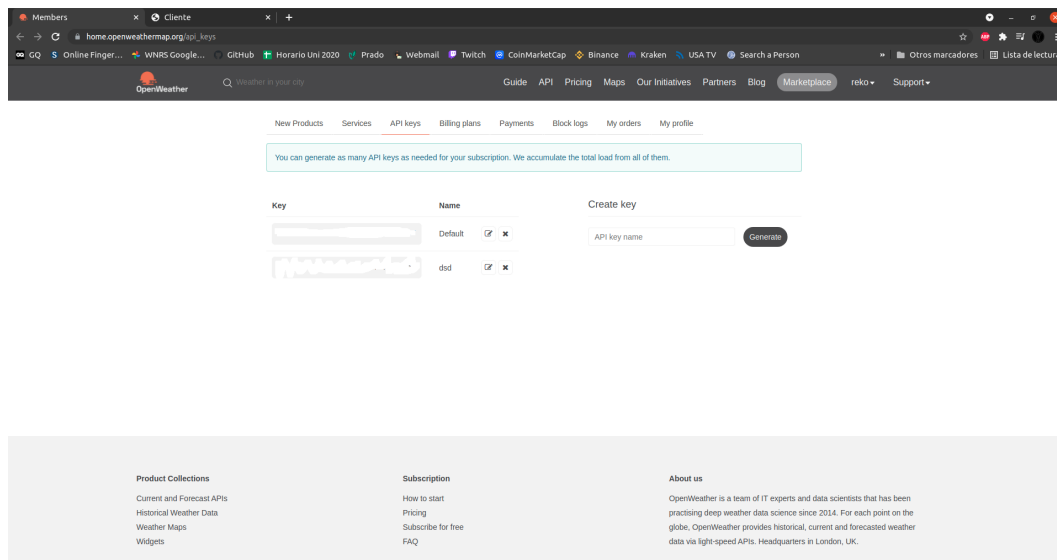
Por último, se han añadido varios métodos para obtener los datos de las variables que tenemos:

- getTemperatura
- getLuminosidad
- getAire
- getPersianas

Todas las funciones nombradas anteriormente se han realizado mediante la escucha de mensajes del cliente(client.on).

3 Partes Opcionales

Como parte opcional he añadido una sección de meteorología en la ventana del cliente. Este imprime los valores de temperatura, sensación térmica, temperaturas máximas, temperaturas mínimas, humedad y presión atmosférica. Todos estos datos se consiguen mediante el uso de la Api que openweather nos proporciona. Para poder hacer uso de dicha opción es necesario crear una cuenta y solicitar una llave Api.

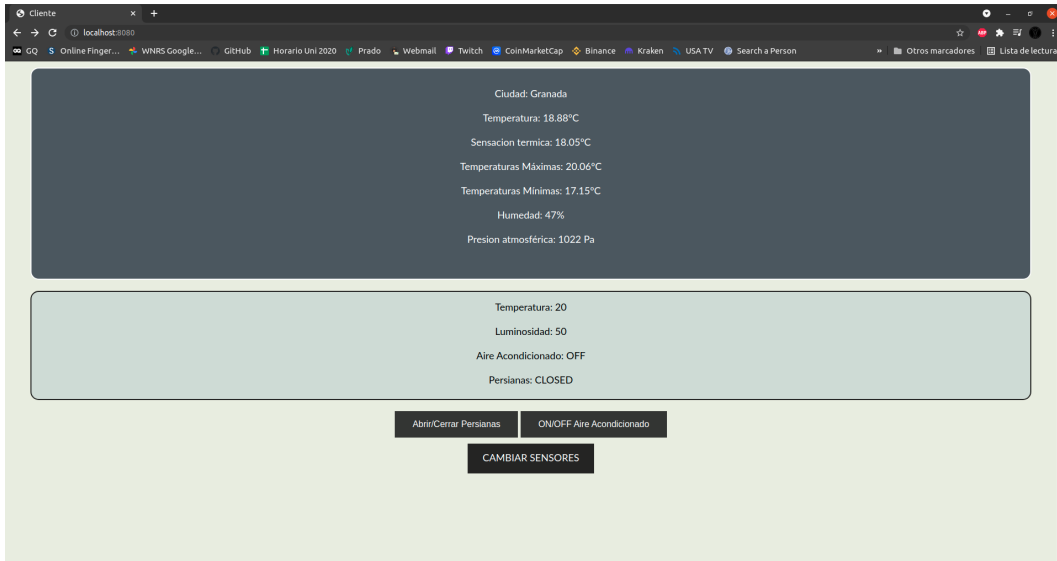


Claves Api

Una vez hayamos obtenido la clave, hacemos uso de ella para poder obtener la información que solicitamos para la aplicación. En este caso, he añadido la opción de poder solicitar la ciudad de la cual queremos obtener la información, siempre y cuando pasemos el nombre junto con la orden de ejecución del programa. No obstante, por defecto, si no pasamos ningún argumento, se muestra la información meteorológica de Granada. De esta forma:

```
reko@R99: ~/repos/Desarrollo-de-Sistemas-Distribuidos/Practica_5/Entrega
reko@R99:~/repos/Desarrollo-de-Sistemas-Distribuidos/Practica_5/Entrega$ node server.js
init
body: {"coord":{"lon":-3.25,"lat":37.25},"weather":[{"id":800,"main":"Clear","description":"clear sky","icon":"01n"}],"base":{"stations":{"main":{"temp":18.88,"feels_like":18.05,"temp_min":17.15,"temp_max":20.06,"pressure":1022,"humidity":47,"sea_level":1022,"grnd_level":893},"visibility":10000,"wind":{"speed":0.31,"deg":119,"gust":0.51},"clouds":{"all":0},"dt":1623009705,"sys":{"type":2,"id":2008325,"country":"ES","sunrise":1622955170,"sunset":1623007443},"timezone":7200,"id":2517112,"name":"Granada","cod":200}}
```

orden default

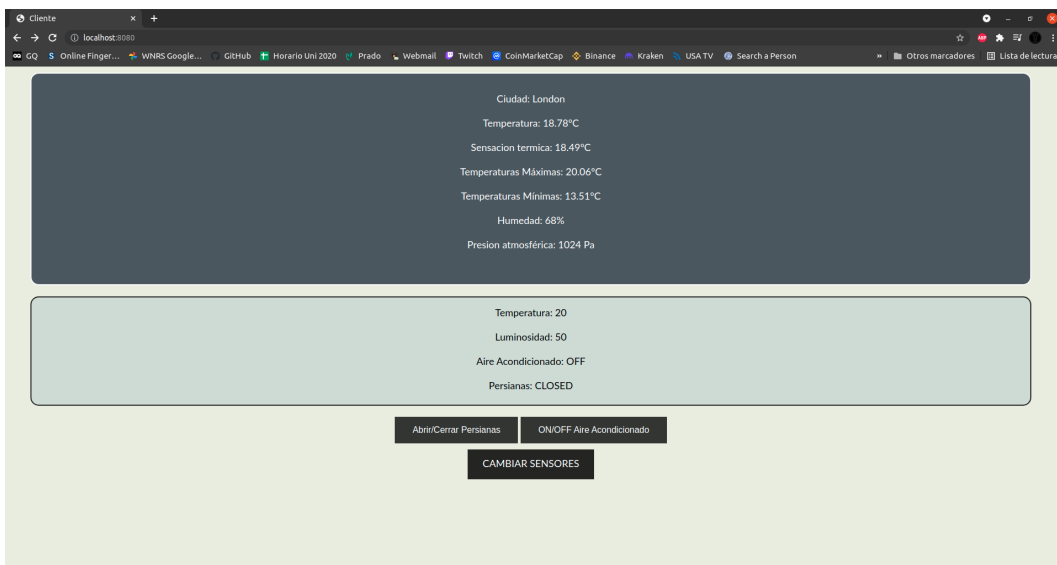


pantalla default

Si ejecutamos por ejemplo de la forma: node server.js -c London:

```
reko@R99: ~/repos/Desarrollo-de-Sistemas-Distribuidos/Practica_5/Entrega
reko@R99:~/repos/Desarrollo-de-Sistemas-Distribuidos/Practica_5/Entrega$ node server.js -c London
init
body: {"coord":{"lon":0.1257,"lat":51.5085},"weather":[{"id":803,"main":"Clouds","description":"broken clouds","icon":"04d"}],"base":{"stations":{"main":{"temp":18.78,"feels_like":18.49,"temp_min":13.51,"temp_max":20.06,"pressure":1024,"humidity":68},"visibility":10000,"wind":{"speed":1.97,"deg":208,"gust":4.48},"clouds":{"all":75},"dt":1623010275,"sys":{"type":1,"id":1414,"country":"GB","sunrise":1622951143,"sunset":1623010379},"timezone":3600,"id":2643743,"name":"London","cod":200}}
```

orden Londres



pantalla Londres