

**2016**

Trabajo con Raspberries

Iñigo Ezcurdia

Daniel García

Daliana Montada

Dayana Ruis

Mikel Uriz

Contenido

[Objetivo 2](#_Toc441505645)

[Trabajo sobre la raspberry centralita 3](#_Toc441505646)

[Preparación 3](#_Toc441505647)

[Toma de medidas mediante el sensor 3](#_Toc441505648)

[Servidor Web 5](#_Toc441505649)

[Recepción de datos de las raspberries remotas 8](#_Toc441505650)

[Trabajo sobre las raspberrys cliente 9](#_Toc441505651)

[Resultado 10](#_Toc441505652)

# Objetivo

Nuestro objetivo es implementar un sistema distribuido de raspberries que recopilen datos sobre la humedad y temperatura en distintos puntos de manera que esta información quede almacenada en un servidor central. Este servidor central, aparte de recopilar información sobre la humedad y temperatura en su emplazamiento, debe también ofrecer la posibilidad de consultar los datos de manera remota mediante una interfaz web. También ofrecerá la opción de consultar la temperatura y humedad de las raspberries remotas en tiempo real, sin necesidad de esperar al envió automático y periódico de estas.



Usuario

Petición realTime

JSON{id,humedad,temperatura,fecha}

Consultor Sensor

Receptor datos

Sensor

Sensor

Sensor

Raspberry medidora

Consultor Sensor

Raspberry medidora

Consultor Sensor

Servidor web

Raspberry servidor

# Trabajo sobre la raspberry centralita

## Preparación

* Se ha instalado el SO raspbian tal y como se indica en el tutorial de la [web oficial](https://www.raspberrypi.org/help/noobs-setup/).
* Se ha configurado accesos FPT y SSH para poder trabajar en ella de manera remota.
* Se ha instalado npm, node, mongo, zmq, forever…
* Se han abierto y redirigido puertos en el router al que está conectada para que puedan realizarse las conexiones a través de internet.



## File:RaspberryPiDHT11.pngToma de medidas mediante el sensor

Se conectó el sensor como se indica en el siguiente diagrama.

Hubo que instalar la [librería bcm283](http://www.airspayce.com/mikem/bcm2835/) para poder obtener datos del sensor.

También se instaló el módulo de node [node-dht-sensor](https://github.com/momenso/node-dht-sensor) para realizar las consultas al sensor de manera sencilla desde node.

Mediante el script ubicado en ServidorNode/sensor/activar.js se realiza la consulta al sensor y se almacena la información en la base de datos mongo de manera periódica.

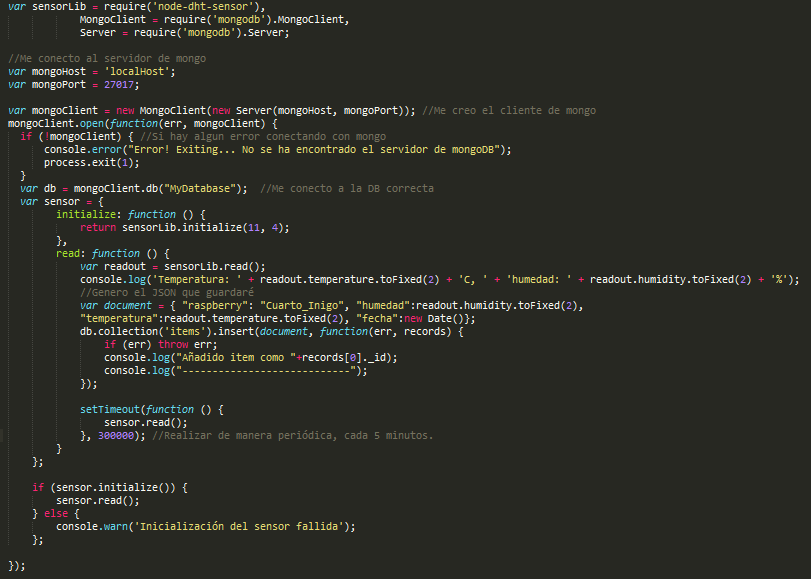


Ilustración ServidorNode/sensor/activar.js

Este script se lanza mediante el comando “ forever start ServidorNode/sensor/activar.js” para que continúe su ejecución indefinidamente aun cerrando la sesión en consola.

## Servidor Web

Se ha creado una API REST escuchando en el puerto 3000 mediante la cual se pueden solicitar datos al servidor, modificarlos, o borrarlos mediante los siguientes comandos:

**Añadir item:**  
curl -H "Content-Type: application/json" -X POST -d '{ContenidoJSON}' http://85.219.25.246:3000/items

**Modificar item:**  
curl -H "Content-Type: application/json" -X PUT -d '{NuevoContenidoJSON}' http://85.219.25.246:3000/items/{\_id}

**Eliminar item:**curl -H "Content-Type: application/json" -X DELETE http://85.219.25.246:3000/items/{\_id}

Todo esto está implementado en el fichero ServidorNode/index.js mediante express.

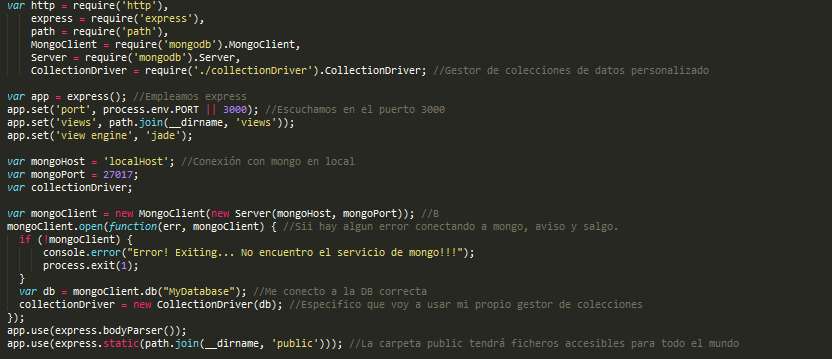


Ilustración ServidorNode/index.js (Comienzo)

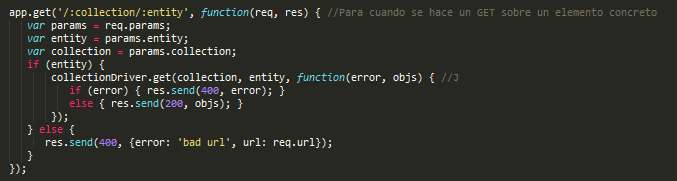


Ilustración ServidorNode/index.js (GET de un item)

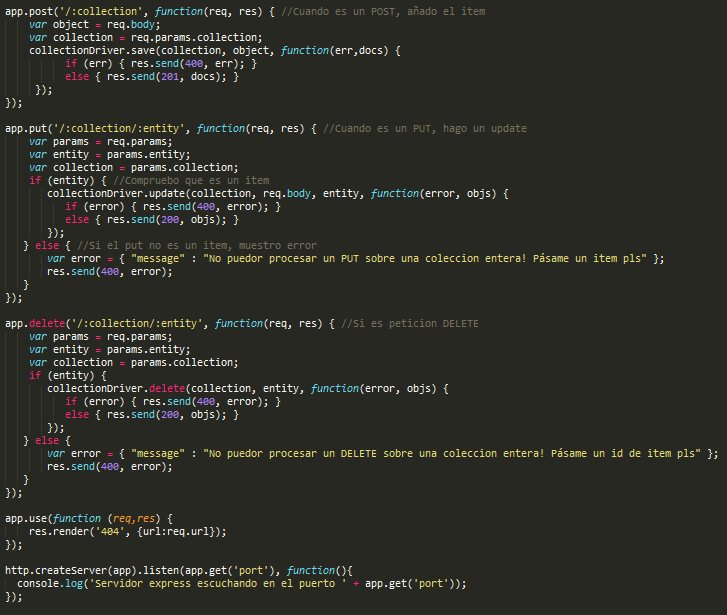


Ilustración ServidorNode/index.js (POST , PUT, DELETE y FINAL)

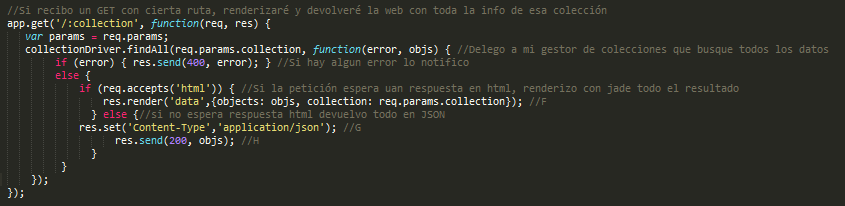
Sin embargo, lo que se hace para servir al usuario una bonita página web de consulta meteorológica queda implementado en el GET de una colección completa:

Ilustración ServidorNode/index.js (GET de una colección)

En este caso delegamos a nuestro CollectionDriver la consulta de los datos a mongo, y mediante Jade renderizamos en HTML (si el emisor de la petición lo permite) y lo enviamos de vuelta.

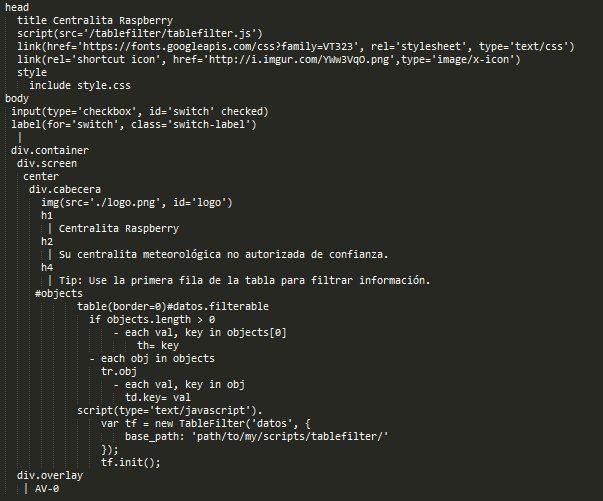


Ilustración ServidorNode/views/data.jade Plantilla Jade mediante la que se construye la respuesta HTML

Al final del informe se puede consultar el resultado.

CollectionDriver hace de puente gestor de colecciones con MongoDB.



Ilustración ServidorNode/collectionDriver.js

## Recepción de datos de las raspberries remotas

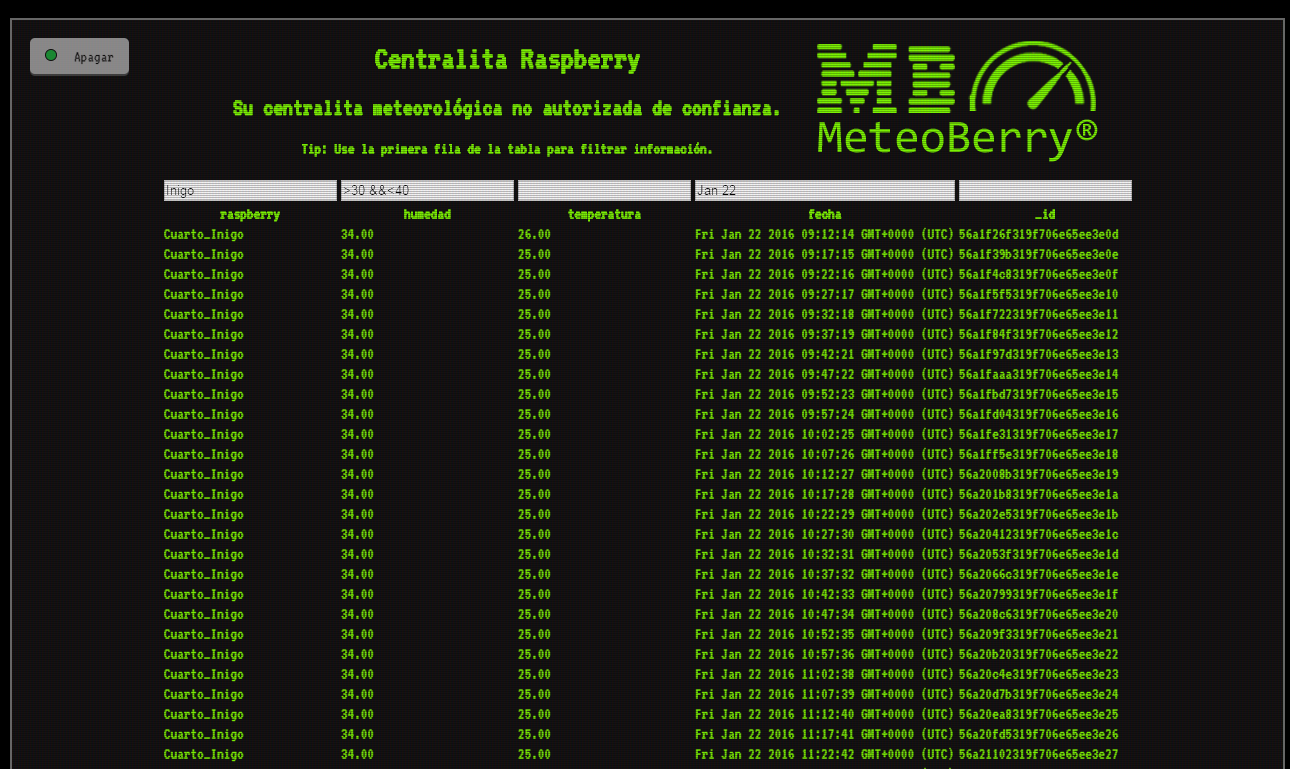
**[PENDIENTE]**

# Trabajo sobre las raspberrys cliente

**[PENDIENTE]**

# Resultado

Se ha intentado dar un aspecto de consola antigua, de computadora viejísima, a modo de homenaje a la serie LOST (perdidos).

La primera fila de la tabla de datos permite [filtrar](http://tablefilter.free.fr/) los datos mostrados y permite operadores como “<”,“>”, “>=”, “=”, “!”, “||”,”&&” y expresiones regulares mediante “rgx:<expresión>$”