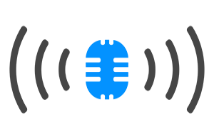
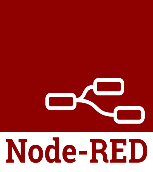
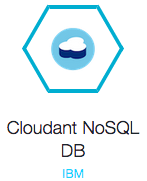
Informe final de la plataforma Cloud

# Despliegue

Llegados a este punto, hemos desplegado utilizando el Cloud de IBM la arquitectura necesaria para hacer funcionar nuestro sistema.

IBM Bluemix



IBM IoT Sensor

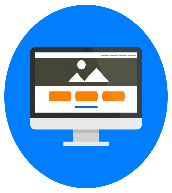
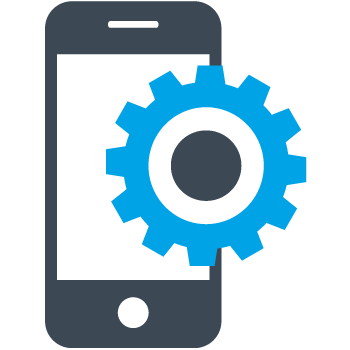


Figura 1 - Arquitectura desplegada

Como se muestra en la Figura 1, la arquitectura de nuestro sistema se compone de un sensor, un servidor node-red que se conecta a ese sensor para obtener los datos que captura, una base de datos en la que node-red almacena la información extraida y a la que se conecta un servidor web Node.js, en que se realizará todo el trabajo de procesamiento. Este servidor incluye un cliente HTML, que será el que se utilice en la pantalla informativa. Además, será necesario implementar una aplicación móvil que se conecte al servidor y que pueda obtener la información necesaria para cumplir con la funcionalidad que se especifica en los documentos anteriores. A continuación, vamos a analizar qué nivel de desarrollo hemos alcanzado en cada componente.

# Componentes

## IBM IoT Sensor

Este sensor es un sensor simulado, para el que se utiliza la herramienta de IBM publicada en <https://quickstart.internetofthings.ibmcloud.com/iotsensor/>.

Dicha herramienta permite crear un sensor IoT que se podrá conectar posteriormente a Node-Red, y que puede simular datos de Humedad o Temperatura. Como no hemos encontrado ningún sensor simulado de ruido, hemos usado el de temperatura, asumiendo que las unidades de trabajo serán decibelios en lugar de grados centígrados.

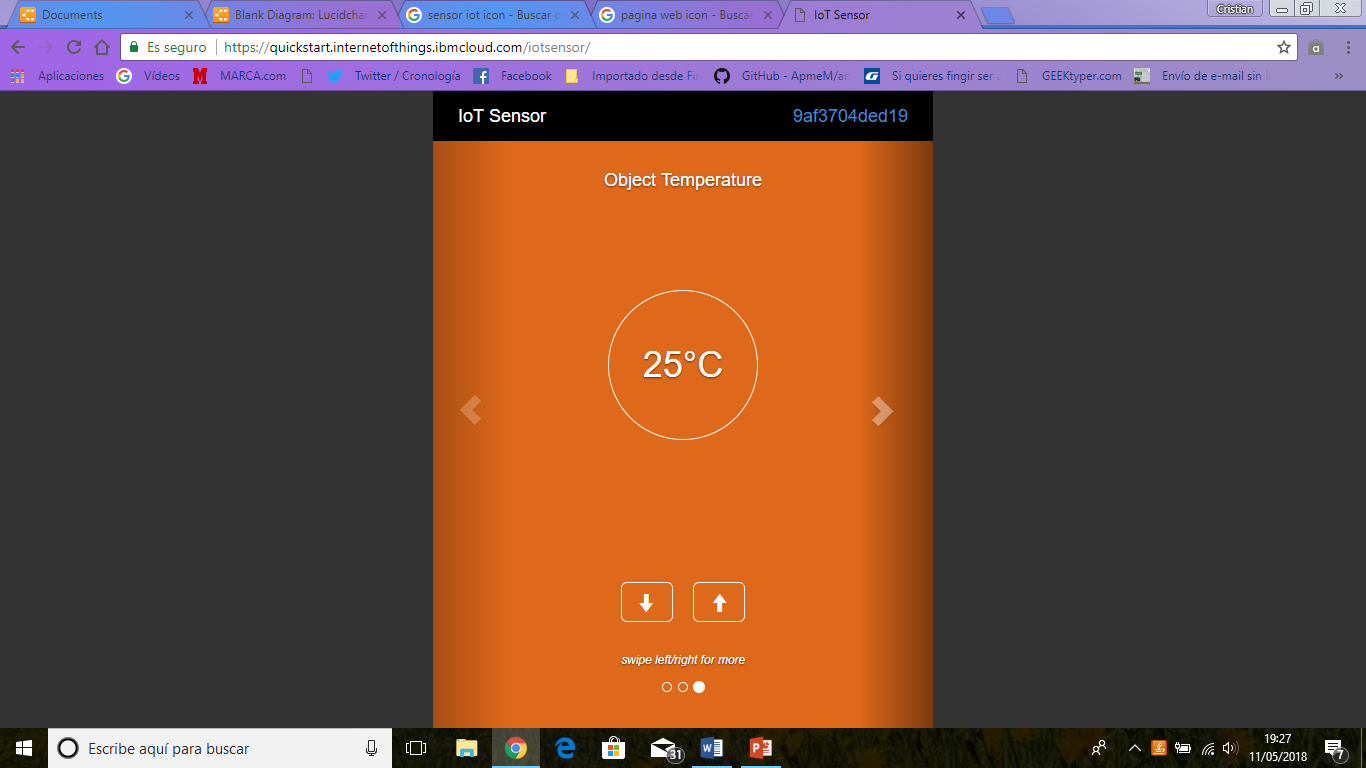


Figura 2 - Sensor IoT

## Node-Red

Hemos desplegado una aplicación Node-Red, que se conecta al sensor simulado mediante su ID, y en la que hemos definido una serie de patrones similares a los que teníamos en el proyecto desarrollado en Simulink para el software de la columna de la conciencia. Estos patrones, determinan un nivel de ruido (Suave, Medio, Alto, Muy Alto e Inseguro) en función del nivel de decibelios que reciben, como se muestra en la Figura 3.

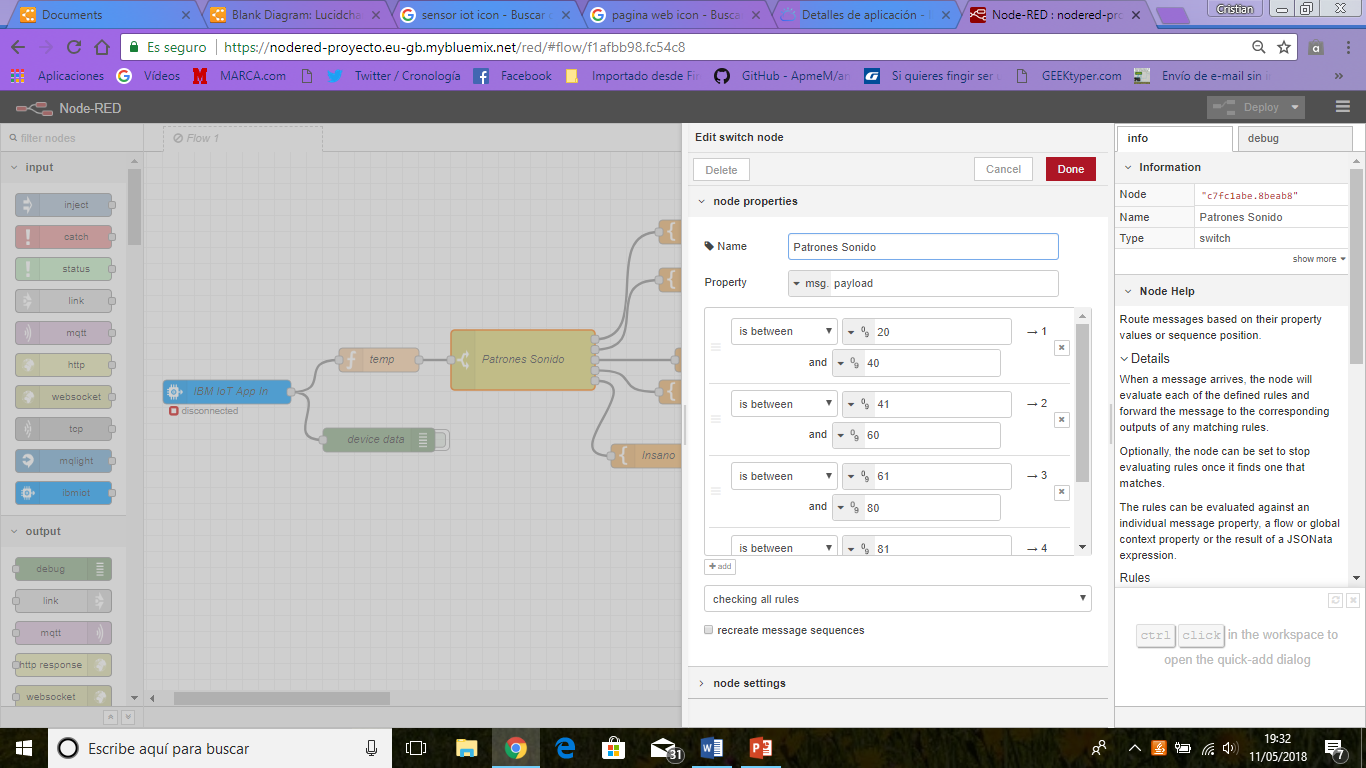


Figura 3 - Patrones de ruido en Node-Red

Una vez que se clasifica el nivel de ruido recibido por el sensor, se procede a almacenarlo en la base de datos.

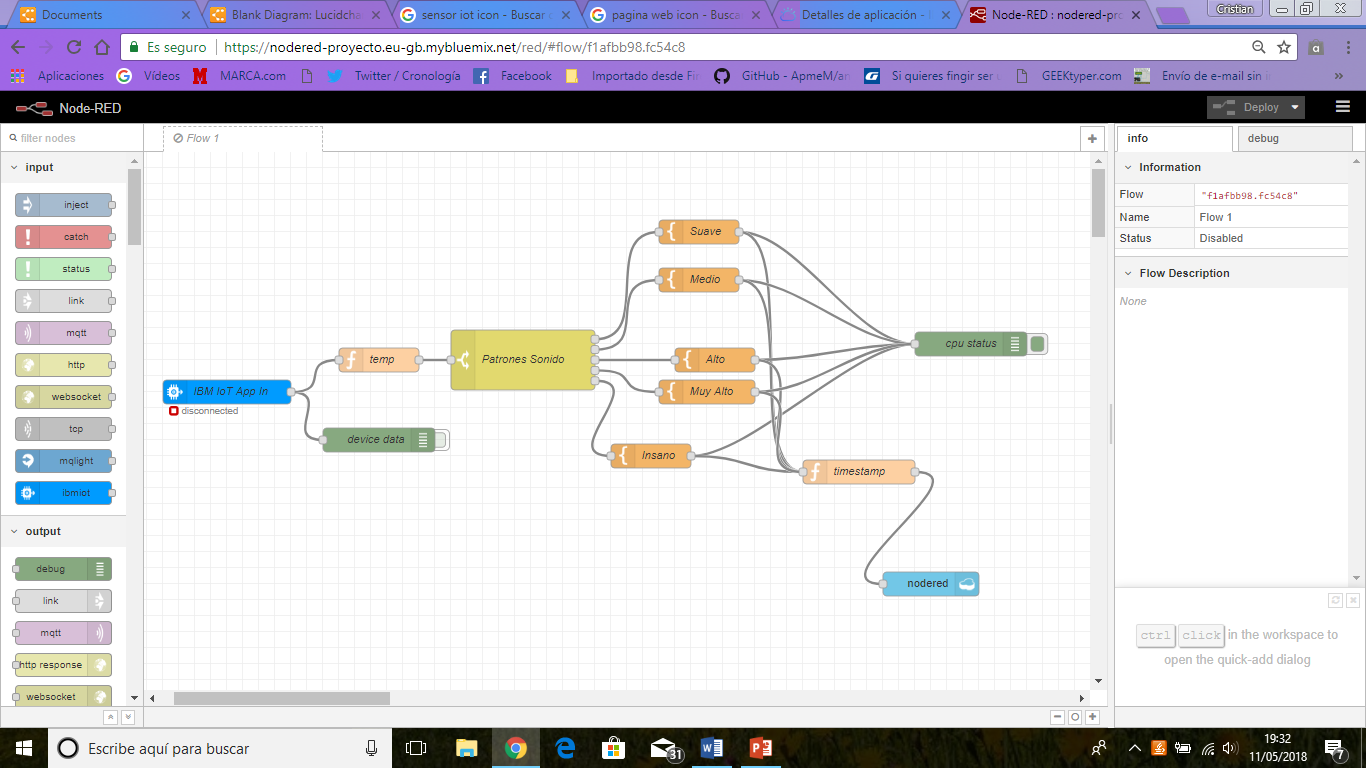


Figura 4 - Vista global de la arquitectura Node - Red

## Cloudant NoSQL

La base de datos se genera automáticamente, bien cuando creas la aplicación Node-Red o bien cuando creas el servidor web. En nuestro caso, la hemos creado junto a la aplicación Node-Red, y hemos creado una nueva conexión también con la aplicación de Node.js

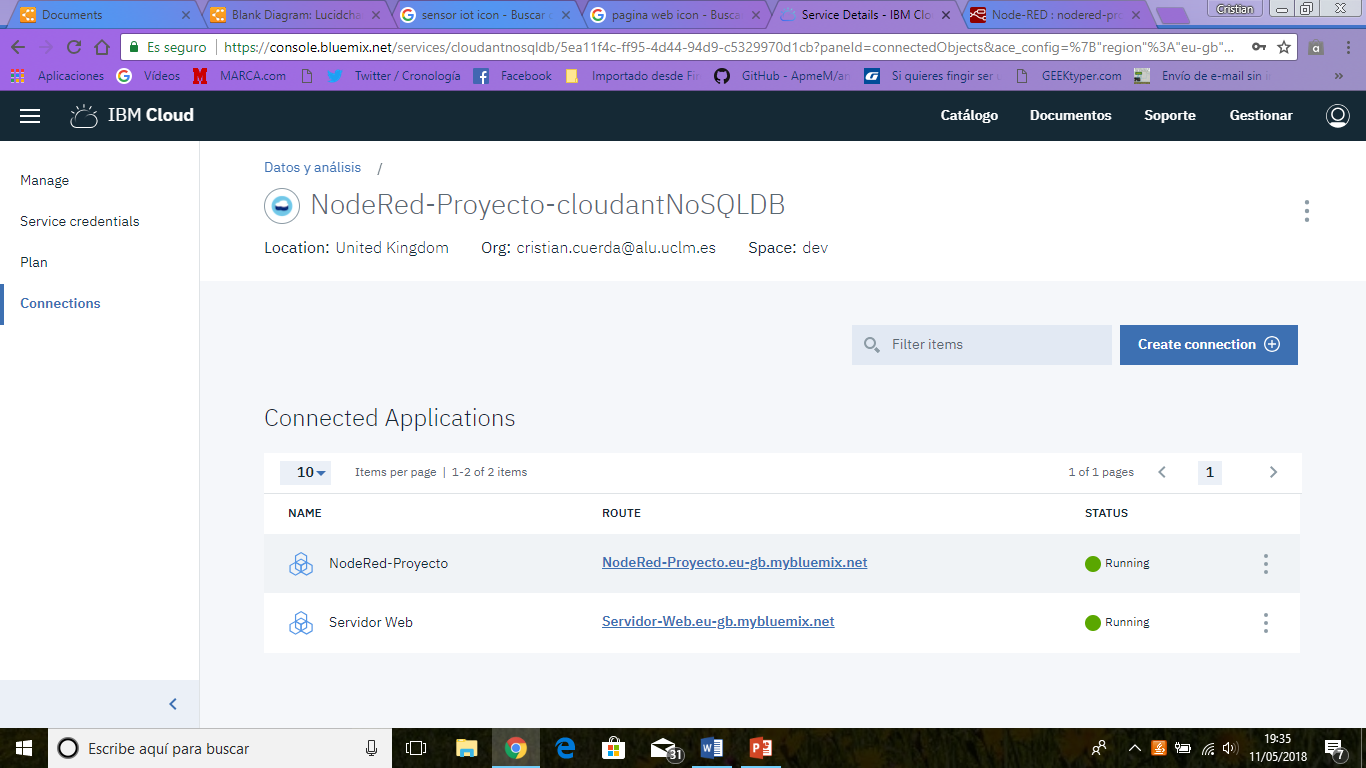


Figura 5 - Conexiones de la base de datos

Dentro almacenamos documentos JSON con el nivel de ruido medido por el sensor. En un futuro, sería necesario almacenar el ID del sensor que envía el ruido, ya que habría varios sensores, y el instante de tiempo en el que se recogen los datos.

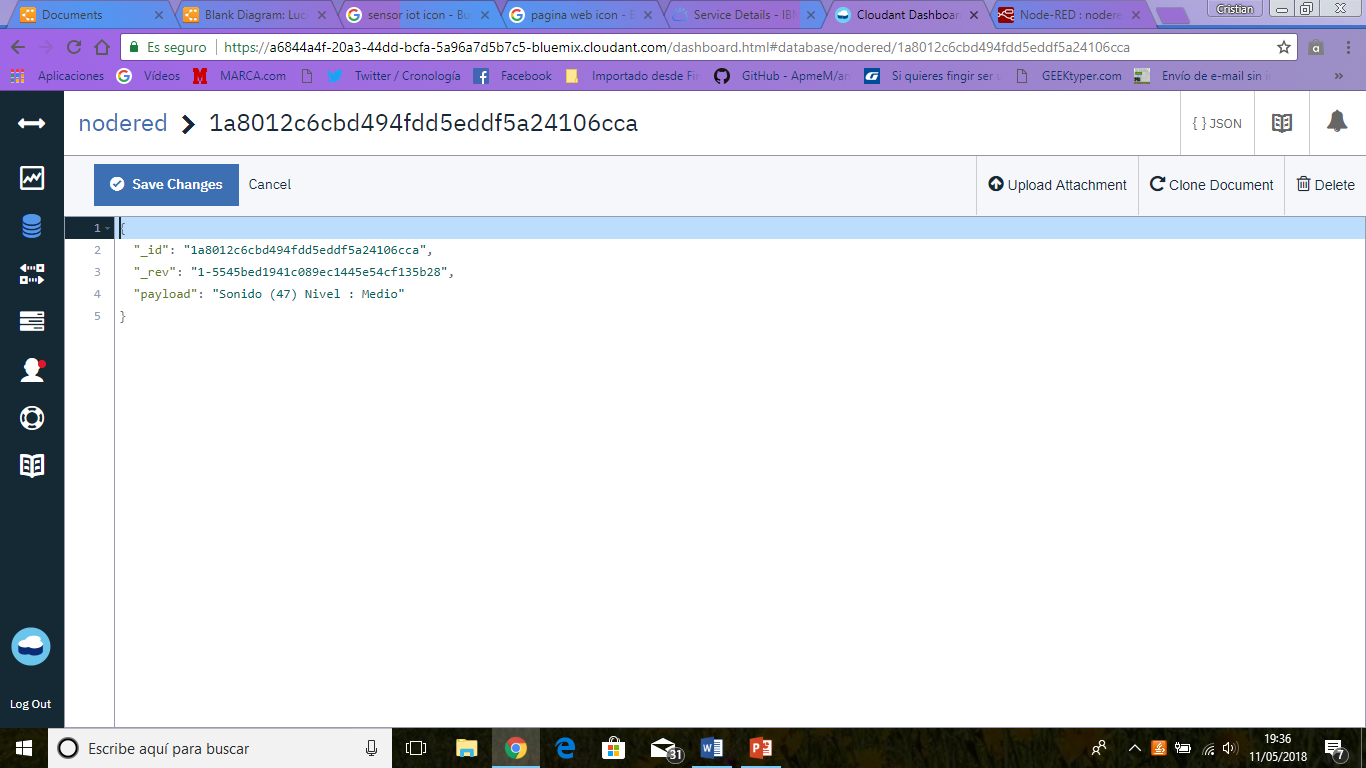


Figura 6 - Ejemplo documento JSON en la base de datos

## Servidor Web Node.js

Para realizar el procesamiento de los datos almacenados en la base de datos, hemos creado una aplicación web utilizando Node.js. Esta aplicación se conecta a la base de datos, extrae los datos que hay almacenados, y permite realizar el procesamiento necesario para que el Cliente Web (la pantalla informativa) o las aplicaciones móviles puedan funcionar adecuadamente. En este caso, no hemos implementado nada, ya que se crean por defecto las funciones para conectarse a la base datos, listar documentos, modificar, añadir o eliminar. Habría que desarrollar las distintas rutas GET y POST a las que los clientes se conectaran para obtener la información procesada.

## Cliente Web

Por defecto, al crear el servidor Node.js, se genera un cliente web, con sus funciones Javascript, el código HTML y los ficheros de estilo CSS. No hemos trabajado con este cliente, ya que no disponemos de tiempo suficiente para realizar la implementación, pero sí que se queda realizada la arquitectura sobre la que solo quedaría desarrollar el código. Se adjuntan prototipos de cómo debería quedar el cliente, que será el que se utilice para mostrar en la pantalla informativa.



Figura 7 - Prototipo de la pantalla informativa

En la pantalla, se mostraría un marquesina en la parte inferior, en la que irían apareciendo los distintos establecimientos monitorizados y el nivel de ruido actual que hay. En la parte izquierda, aparecen los tres mejores establecimientos, los que más tiempo llevan en un nivel de ruido adecuado. A esos establecimientos, se les permite mostrar su publicidad en la pantalla.

## Aplicación Móvil

La aplicación móvil queda fuera del dominio de IBM Bluemix, y sería necesario desarrollarla para los distintos dispositivos móviles. Podría realizarse mediante un nuevo Cliente Web que se enmascararía dentro de una app para cada tipo de dispositivo (Android, IOS, Windows, etc) o bien desarrollando una aplicación en código nativo para cada plataforma. Esta segunda opción, aunque es más costosa, se beneficiaría mejor de las características propias de los dispositivos móviles y podría realizar un acceso más fácil a hardware como el GPS. Se adjuntan prototipos del resultado esperado, tanto de la aplicación móvil para establecimientos, como la aplicación móvil para clientes de establecimientos.



Figura 8 - Prototipo App Móvil Clientes y Establecimiento

La aplicación móvil para clientes muestra los puntos que han acumulado y en que pueden canjearlos, mientras que la aplicación para establecimientos cuenta con opciones para personalizar la publicidad del local, ver el nivel de ruido actual del local (por si el propietario no está en el local y no puede ver la columna de la conciencia, por ejemplo) y la información histórica del local.