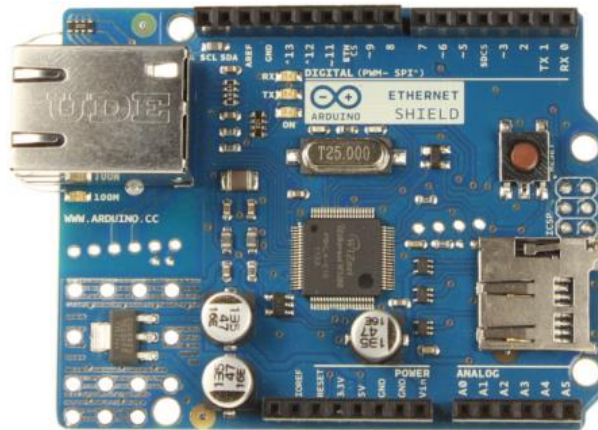


# Internet de las Cosas



Práctica 10 – Actividades (v1.3.3 octubre 2022)

## Software para robots

Cristian González García

[gonzalezcristian@uniovi.es](mailto:gonzalezcristian@uniovi.es)

Basado en el material original de Jordán Pascual Espada

# Índice

Introducción.....	2
Actividades normales.....	2
Estructura.....	2
(IoT10.1) Red de sensores de temperatura y humedad (1 punto) .....	2
Ampliaciones.....	4
(IoT10.2) Recolector de basura (0,5 Puntos) .....	4

El total de las actividades tienen un valor de 1 punto obligatorio y 0,5 puntos optativos dentro del bloque 5.

## Introducción

Lista de reproducción en YouTube:

<https://www.youtube.com/watch?v=07iYgehBpTo&list=PLpe5dTl2xCy8CNbYdZPkCwvCwpLV4IbkW>

## Actividades normales

**Cuidado** con el error «Access-Control-Allow-Origin» de «Cross-Origin Resource Sharing» y de activarlo en el Arduino: <https://www.html5rocks.com/en/tutorials/cors/>

### Estructura

Las actividades deben constar de los siguientes apartados:

- Un **prototipo funcional** de un dispositivo recolector de datos conectado a Internet. Se requiere el software y el circuito de conexiones hardware. El proyecto implantado podrá constar de cientos de dispositivos, pero en nuestro caso solo construiremos un único dispositivo, que serviría como prototipo para comenzar la producción.
- **Aplicación de monitorización:** se debe desarrollar una aplicación simple en JavaScript que consulte los datos de los dispositivos conectados y los muestre de la manera más adecuada posible. En la actividad, la aplicación se conectará a un único dispositivo, que es el prototipo. Cada dispositivo debería tener una IP y una posición en la representación (por ejemplo, en un mapa).
- **Documentación:** resumir los servicios/páginas soportados por el objeto y las funcionalidades de la aplicación web.
- **Vídeo de demostración:** un vídeo en el que se muestre la funcionalidad de todo el proceso.

## (IoT10.1) Red de sensores de temperatura y humedad (1 punto)

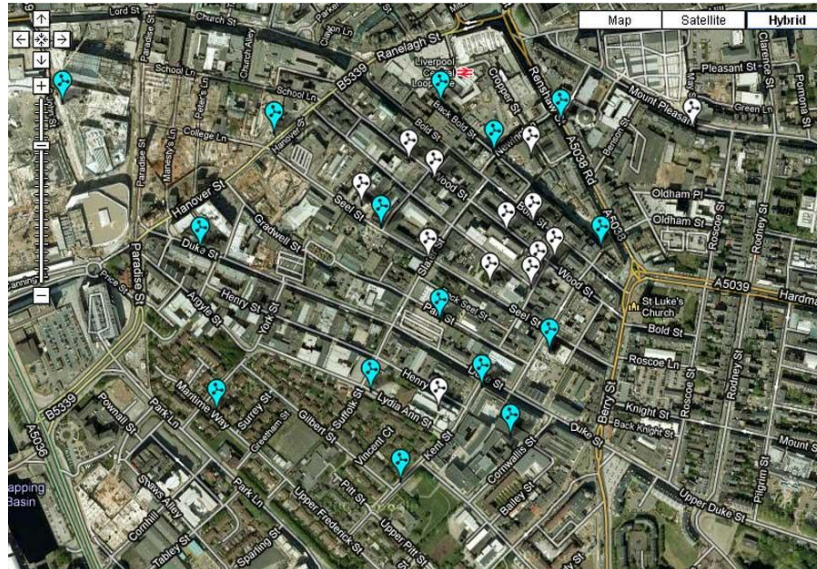
Diseñar un sistema basado en una red de sensores distribuidos y un panel web de monitorización/control.

La ciudad de Oviedo plantea colocar una red de sensores por la ciudad. Los dispositivos captan la **temperatura** y humedad del aire. Cada dispositivo tiene un identificador único (su IP) y una localización fija pregrabada. Es decir, sabemos de antemano las coordenadas GPS de donde hemos colocado el dispositivo con IP X.X.X.X. Los dispositivos se colocarán en lugares donde ya haya corriente eléctrica, como luces, y aprovecharán la alimentación eléctrica de las mismas, luego, no requieren batería, y pueden conectarse a Internet.

Se requiere una aplicación que permita consultar en tiempo real la información de cada sensor y ver donde están ubicados en el mapa de la ciudad. Además, **la aplicación deberá mostrar una notificación especial cuando un sensor rebase un valor prefijado.**

Desde la aplicación web también **se podrá enviar una señal a un dispositivo concreto para que encienda o apague LED situado en dicho dispositivo para poder indicar alertas.**

Este LED podría ser utilizado para indicar la situación del aparato, indicar visualmente su mal funcionamiento, etc.



*Ejemplo, representación mapa con sensores*

Resumen:

- **Crear una aplicación web para:**
  - Recibir continuamente datos y visualizarlos en una página web, es decir, se deben ir actualizando.
    - Puede ser al clicar en las chinchetas
    - En una tabla aparte
    - Una combinación de ambas
    - Etc.
  - Mostrar la situación de los sensores sobre un mapa.
  - Encender y apagar un LED a distancia según conveniencia.
- **Se recomienda:**
  - Hacer una aplicación web con HTML, CSS y JS externa al Arduino en la que se consulte el Arduino y se le hagan peticiones como en la teoría explicada en las prácticas. De esta manera, la gran carga de la lógica de la web estará fuera del Arduino y no le supondrá un gran peso de espacio.
- El hardware constará de un sensor de DHT11, un LED, una placa Arduino Uno o similar y un Shield de Ethernet.
- Utilizar mapas (Google, OpenLayers, etc.).
  - Necesitaréis crearos la API de acceso al servicio para utilizar los mapas.
  - Ejemplos de Google Maps:  
<https://developers.google.com/maps/documentation/javascript/examples/>

- Si os muestra un mensaje de «Only for development purposes» significa o que no habéis activado la clave o que no habéis introducido la «key API».

Vídeo del ejercicio: <https://www.youtube.com/watch?v=5VcZ121k7Nc>

## Ampliaciones

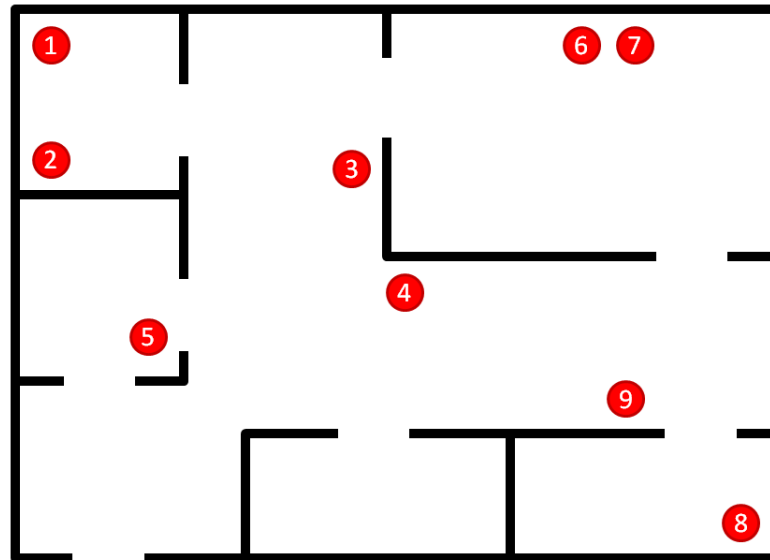
### (IoT10.2) Recolector de basura (0,5 Puntos)

Una empresa de alimentación plantea colocar una red de sensores en sus instalaciones. En las instalaciones de la empresa hay muchos contenedores y cubos que se utilizan para tirar el producto no aprovechable. Se colocará un **sensor de distancia** en la parte superior de los cubos para comprobar la capacidad de la que disponen. También se colocará un **sensor de colisión** en la tapa para comprobar cuando están abiertos o cerrados. Ambos sensores estarán integrados en un único dispositivo conectado.

Cada dispositivo tiene un identificador único (su IP y una localización fija pregrabada). Por ejemplo, sabemos de antemano en que habitación hemos colocado el dispositivo con IP 1.1.1.1. Los dispositivos se alimentarán directamente de la corriente eléctrica, así que no requieren batería y pueden conectarse a Internet utilizando la red Wifi o un cable de Ethernet de la empresa.

Se requiere una aplicación que permita consultar en tiempo real la información de cada sensor y ver donde están ubicados en las instalaciones de la empresa. Además, la aplicación deberá mostrar una notificación especial cuando un sensor de distancia detecte que un cubo está muy lleno (detecciones de distancias menores a 10 cm).

Desde la aplicación web también se podrá enviar una señal a un dispositivo concreto para que encienda una señal luminosa (LED) que indique a los operarios que deben dejar de utilizar ese cubo, porque van a vaciarlo próximamente.



*Ejemplo: representación de un mapa con sensores*

Resumen:

- **Crear una aplicación web para:**
  - Recibir continuamente datos y visualizarlos en una página web, es decir, se deben ir actualizando.
    - Puede ser al clicar en los sensores
    - En una tabla aparte
    - Una combinación de ambas
    - Etc.
  - Mostrar la situación de los sensores sobre un mapa.
  - Encender y apagar un LED a distancia según conveniencia.
- **Se recomienda:**
  - Hacer una aplicación web con HTML, CSS y JS externa al Arduino en la que se consulte el Arduino y se le hagan peticiones como en la teoría explicada en las prácticas. De esta manera, la gran carga de la lógica de la web estará fuera del Arduino y no le supondrá un gran peso de espacio.
- El hardware constará de un sensor de ultrasonidos, un sensor de colisión, un LED, una placa Arduino Uno o similar y un Arduino Ethernet Shield.
- Sugerencia
  - Utilizar la etiqueta canvas de HTML5 para pintar el mapa y la situación de los elementos dentro del edificio.