Martín Beltrán Díaz – UO276244 LAURA PERNÍA BLANCO – UO276264 STELIAN ADRIAN STANCI – UO277653 Grupo 3-7

SOFTWARE PARA ROBOTS

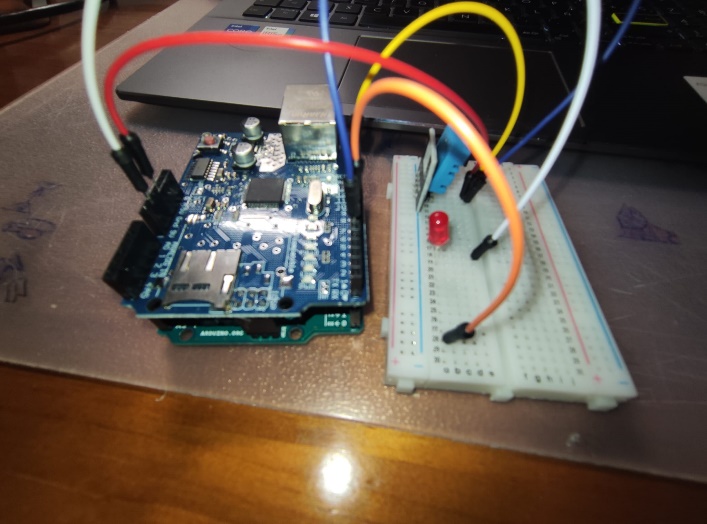
Internet de las cosas

# IoT10.1 – Red de sensores de temperatura y humedad.

Para desarrollar la aplicación de monitorización hemos utilizado la API de Google Maps para JavaScript, en su versión de pago, puesto que el bono gratuito lo utilizamos el año anterior para la asignatura de SEW. La web se ha construido en base a uno de los ejemplos avanzados que Google Cloud Platform ofrece en su documentación: <https://console.cloud.google.com/google/maps-apis/build/locator-plus?project=decent-being-369415>

Se ha editado dicho ejemplo para adaptarlo a nuestras necesidades.

El servidor web que corre en el Arduino responde a peticiones GET siempre con JSON, devolviendo la humedad, temperatura y el estado del led en ese momento. En función de los parámetros en la petición GET, enciende/apaga el led, o solo devuelve los estados. El sensor DHT está conectado al pin 5 y el led al 7.



Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteLa web mostrará la localización de los Arduino en función de la configuración que se le pase, y relacionará cada localización con una IP determinada, en nuestro caso, pese a mostrar más de un Arduino en la web, ambos apuntan a la misma IP. Este es un ejemplo de configuración:

En el ejemplo podemos ver que se configuran dos localizaciones, ambos pedirán datos a la misma IP. También se establecen alertas, para que la web nos avise de alguna forma si las mediciones sobrepasan valores límite.

Mapa

Descripción generada automáticamente

Al pulsar sobre los marcadores rojos, se desplegará un ventana de información que mostrará los datos recogidos por el Arduino, así como un botón para interactuar con el led:

Mapa

Descripción generada automáticamente

Las medidas tendrán 3 colores en función de las alertas: rojo si sobrepasa el valor límite y el led de aviso está apagado, naranja si el valor sobrepasa el límite, pero el led de notificación ya ha sido encendido, y verde, que indica que el valor está por debajo del límite.

Cabe destacar que el mapa y la barra de búsqueda es completamente funcional, por lo que responderá a las búsquedas del usuario.

Para terminar, puesto que no disponíamos del SHIELD ETHERNET en casa durante los primeros días, desarrollamos una API con Python y Flask que simulaba la respuesta del Arduino: https://phoenixnap.com/kb/install-flask

Se incluye dentro de la carpeta IoT10.1 una carpeta extra con instrucciones y el código de la API por si quiere revisarse.