

PEC2

COMPUTACIÓN UBICUA

Cabanillas Prudencio Luis Alejandro

Losada Fernández Miguel Ángel

Maestre Santa Álvaro

Zamorano Ortega Álvaro

Contenido

ARQUITECTURA 2

ARDUINO 2

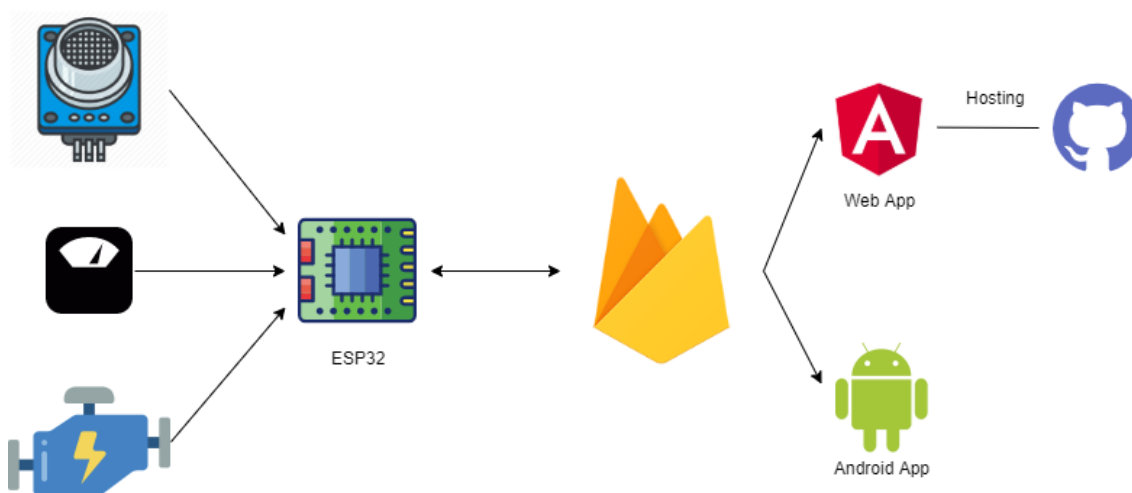
WEB 3

ANDROID 4

ARQUITECTURA

Hemos cambiado la arquitectura respecto a la expuesta en el documento de la primera entrega de la práctica. Los cambios más significativos es el uso de Angular8 en lugar de NodeJs como herramienta principal de la página web. La aplicación Android se conecta directamente a nuestra página web mediante el componente WebView que nos proporciona AndroidStudio.

La forma de funcionamiento de nuestro sistema es muy sencilla. Nuestro cubo inteligente que contiene la ESP32 manda información a Firebase (servicio en la nube), entonces nuestras aplicaciones, tanto la web con Angular como la aplicación Android leen los datos en tiempo real y se actualizan automáticamente. Para las aplicaciones Web y Android el comportamiento respecto a Firebase es unidireccional ya que únicamente leen datos, para el caso de la ESP32 es bidireccional, puesto que además de mandar los datos del cubo, también lee un índice de referencia para insertar los datos de forma ordenada.



ARDUINO

En primer lugar, describiremos los pasos realizados en la construcción e inserción de los sensores en el prototipo de cubo de basura:

1. Para empezar, se comenzó probando el movimiento del servomotor y si éste era capaz de mover la tapa del cubo. Para ello, se unió una pieza metálica en el servo para facilitar la abertura y posteriormente se adhirió el servo en el interior del cubo. Al intentar mover la tapa, el servo no consiguió abrirla, por lo que se tuvo que quitar peso de la tapa y más tarde montarla con un cartón de color.
2. En la montura de la celda de carga, ésta se instaló en la parte de debajo del cubo con unos tornillos, para después poner en la parte posterior de la celda una base de la siguiente forma:



3. En la instalación de los sensores de distancia HC-SR04 no hubo mucha dificultad. Se adhirió uno de ellos en la parte frontal del cubo (que mide la distancia para abrir automáticamente el cubo) y otro de ellos en la tapa (que mide la distancia al fondo).

En segundo lugar, describiremos la parte de código de Arduino. Las librerías usadas son las siguientes:

- Wifi.h: establece conexión wifi desde la ESP32 a un router.
- Firebase ESP32 Client: permite el envío y recepción de datos con una base de datos firebase.
- WifiLocation: permite obtener las coordenadas (latitud y longitud).
- NTPClient: permite obtener la fecha y hora actuales mediante una solicitud al servidor NTP.
- Servo.h: implementa las funciones necesarias para el movimiento del servomotor.
- HX711.h: implementa las funciones necesarias para el cálculo y obtención del peso de la celda de carga y la placa HX711.

El funcionamiento del método loop es el siguiente: el sensor de distancia frontal calcula cada 5 segundos la distancia, y si esta es menor de 30 cm, el servo motor gira y mueve la tapa. Éste gira de forma instantánea a la posición de 50 grados, y pasados 5 segundos, vuelve a girar a la posición inicial de una forma más lenta (esto se realiza mediante un bucle for).

Una vez cerrada la tapa, se procede al cálculo de la distancia y del peso. Una vez calculados, se envía a firebase, enviándose a su vez la fecha y hora actuales, la latitud y la longitud. Estos datos se insertan en el último histórico. Esto se realiza guardando en una variable global el número de históricos que contiene la base de datos, que inicialmente leemos en la inicialización de la ESP32.

WEB

En lo que respecta a la aplicación web utilizamos Angular e implementamos diversos módulos que nos permiten entre otras cosas conectarnos a Firebase, realizar gráficos mediante ChartJs y mostrar mapas a través de Google Maps. Los más importantes son:

- **angularfire2**: conexión con la base datos en Firebase. Está conexión nos permite utilizar funciones CRUD.
- **chart.js**: permite realizar gráficos en tiempo real.
- **agm** (Angular Google Maps): permite hacer uso de la herramienta de Google Maps en nuestra aplicación web.

Nuestra aplicación recoge los datos de la base de datos de Firebase y muestra información sobre el cubo asociado en tiempo real mediante el uso de datos y gráficos, además mostramos una

gráfica con información histórica del cubo como son los datos del peso y la distancia libre al fondo de este. También mostramos la ubicación del cubo y proporcionamos una serie de información estadística interesante para el usuario.

La aplicación se encuentra hosteada en GitHub gracias a la herramienta **gh-pages** que nos proporciona esta plataforma. El dominio en el que se encuentra hosteada nuestra aplicación web es <http://www.trashqube.com/>.

ANDROID

Con respecto a la aplicación, hemos utilizado Android Studio, que es un entorno de desarrollo, para desarrollar aplicaciones Android. La función principal de esta aplicación es conectarse a la página web y darle formato a la página para que sea visible en el móvil.

En este caso, hemos decidido que se ejecute solo en horizontal, ya que la visualización de las gráficas mejora considerablemente con respecto a cuando está en vertical. Cabe destacar que nuestra aplicación estará disponible para la versión de android 5.0 en adelante. De esta forma, facilitamos al usuario una herramienta para que esté al tanto de lo que está ocurriendo en su dispositivo trashqube desde el móvil, ya que meterse a internet puede ser más engorroso en el día a día, pero el teléfono móvil es un dispositivo que siempre llevamos encima.

Hemos querido que nuestra aplicación sea reconocible en el Play Store, por lo que le hemos asignado un icono y un nombre que la distinguirán del resto:



Al pulsar sobre el icono se nos abrirá directamente la página web que monitoriza el estado de nuestro cubo inteligente.