

# Cálculo del centro de gravedad con Arduino para Android

Diego Sebastian Jiménez Gallegos

**Resum** Se presenta el diseño y la implementación de un sistema para medir el centro de gravedad de drones utilizando Arduino y visualizando los datos en un dispositivo con Android. Medir el centro de gravedad nos permitirá llevar un control de cómo se distribuye la carga en un dron, y por lo tanto, nos puede ayudar a identificar errores que se pueden producir en este a causa de una mala distribución e incluso problemas con la sujeción de la carga.

**Paraules clau**— Arduino, Android, Centro de gravedad, centro de masas, celdas de carga, bluetooth, dron.

**Abstract**— This Article describes how to create a system to measure the center of gravity of a dron using Arduino and Android. Measure the center of gravity allow us to monitor how the load of a dron is distributed. Having knowledge about the center of gravity can help us to identify some errors that could happen (before they happen) and even it can help us to identify problems in the fastening.

**Index Terms**— Arduino, Android, bluetooth, center of gravity, dron, load cell.



## 1 INTRODUCCIÓ

EN los últimos años la cantidad de drones y su uso en diferentes áreas (comercial, militar, personal, etc.) ha aumentado de manera notoria. Este aumento ha hecho que surjan nuevos proyectos de grandes empresas como Amazon, hasta ideas de uso para estos, como grabar carreras.

Los drones se han dado a conocer principalmente en las guerras. Se puede decir que los primeros sistemas no tripulados son del año 1849 cuando Austria intentó atacar Venecia con globos no tripulados, pero no es hasta la década de los años 70 cuando se tienen resultados parecidos a los drones actuales. A partir de allí, los drones han ido evolucionando hasta ser lo que tenemos hoy en día. Uno de los problemas de los drones es el peso, que afecta directamente a la autonomía de vuelo. Es importante elegir bien la colocación de las piezas, como la batería, y otros accesorios como las cámaras. Medir el centro de gravedad permitirá llevar un control de cómo se distribuye la carga en un dron, y, por lo tanto, puede ayudar a identificar errores que se pueden producir en este a causa de una mala distribución, cuando el centro de gravedad está desajustado puede forzar los motores para mantenerlo estable, también se pueden ocasionar problemas con la sujeción de la carga.

El documento sigue la siguiente organización:  
En el apartado 2, se describirán los objetivos del proyecto.

En el 3, se explicará el estado del arte, dónde se mostrará el por qué se han elegido los objetivos del apartado 2.

El apartado 4, explica cómo se realiza el cálculo del centro de gravedad, los componentes y software utilizados. En el apartado 5 se habla sobre la metodología seguida. En el apartado 6 se muestra los resultados obtenidos y en el 7 una prueba en conjunto. Finalmente se acaba con las conclusiones

## 2 OBJETIVOS

El objetivo de este trabajo es crear un sistema que se encargue de decirnos el estado de la distribución de la carga respecto al dron, y mostrarnos el mismo en una aplicación para android, así podemos realizar el proceso de ver cómo está distribuida la carga desde cualquier lugar y sin necesitar más que un dispositivo android con la aplicación instalada.

Teniendo esto en mente se definieron los siguientes objetivos para llevar a cabo el proyecto:

- Analizar los componentes que van a ser necesarios y comprarlos.
- Plantear cómo realizar el cálculo del centro de gravedad.
- Familiarización con las librerías que se van a utilizar y crear software para el cálculo del centro de gravedad.
- Realizar pruebas de funcionamiento del cálculo del centro de gravedad.
- Probar ejemplos para el módulo de bluetooth.
- Realizar configuración del módulo de bluetooth.
- Dar formato a los mensajes a transmitir.
- Crear aplicación que muestre los datos.

- E-mail de contacte: d.jimenezgallegos@gmail.com
- Menció realitzada: Enginyeria de Computadors / Computació / Enginyeria del Software / Tecnologies de la Informació.
- Treball tutoritzat per: nom i cognoms del tutor (departament)
- Curs 2015/16

- Realizar pruebas de funcionalidad entre los dos módulos.

### 3 ESTADO DEL ARTE

Hoy en día, es relativamente fácil conseguir un dron, es más, existe un proyecto muy interesante llamado nixie [12] que propone llevar un dron como pulsera, además de otros muchos como el de Amazon que quiere montar un sistema de reparto de paquetes con estos [13].

Sin embargo, para este segundo caso, como para todos los drones que deben llevar una carga, aun no existen sistemas de confianza para distribuir bien el peso en los drones (al menos no de manera abierta) y llevar un control sobre los posibles fallos que se pueden producir por una mala distribución del mismo.

Los posibles errores que se pueden descubrir si implementamos un sistema de control para la distribución de la carga, son los siguientes:

- Puede haber motores más cargados que se desgasten antes y, por lo tanto, que fallen antes que el resto de motores.
- Los giros pueden ser más difíciles de hacer en función de como se distribuya la carga.
- El sistema de sujeción de la carga puede ser que no sea el adecuado y esta se desplace, dando fallos en el vuelo.

Por eso, lo que se propone es llevar a cabo un sistema que se encargue de darnos información sobre la distribución del peso, para así nosotros actuar en consecuencia (a poder ser, antes de realizar el vuelo).

### 4 METODOLOGÍA

Para conseguir cumplir los objetivos propuestos en el apartado 3, se ha seguido la metodología en Cascada. Esto quiere decir que se deben seguir las fases que se muestran en la siguiente imagen:

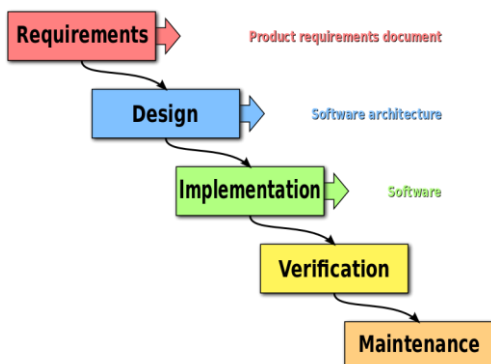


Figura 1 - Esquema metodología en cascada

En la primera fase, la de requerimientos, definimos que es lo que se busca obtener con el proyecto y se separa en diferentes objetivos. En nuestro caso crear un sistema que calcule el centro de gravedad de un dron, y el resultado de este lo pase por bluetooth a una aplicación para android, donde esta última muestre el resultado y los objetivos son los definidos en el apartado 3.

En la fase del diseño se define qué se va a utilizar para llevar a cabo el proyecto. En nuestro caso son el conjunto de módulos que se utilizan con el microcontrolador genuino mega y AppInventor para la creación de la aplicación.

En la fase de implementación, es cuando creamos el software, que en el caso de este trabajo se corresponde con la creación del programa para arduino que realizará el cálculo del centro de gravedad y con el software para android que se encargará de mostrar los resultados.

La fase de verificación consiste en probar que el sistema funciona correctamente, en nuestro caso consiste en hacer comprobaciones sobre el si está bien calculado el centro de gravedad y si se muestra correctamente los datos del centro de gravedad en la aplicación para android.

Por último, la fase de mantenimiento puede consistir en una fase de mejora y comprobar que no se produzcan fallos en el sistema. Esta última fase en nuestro caso es la única que no se cumple porque el trabajo es cerrado, es decir, cuando se ha acabado se le puede dar mantenimiento, pero en principio ya ha finalizado.

## 5 CÁLCULO DEL CENTRO DE GRAVEDAD, COMPONENTES Y SOFTWARE UTILIZADOS

### 5.1 Cálculo del centro de gravedad

Para realizar el cálculo del centro de gravedad, con diferentes masas en un sistema de 2 dimensiones (x e y) se deben tener en cuenta los *momentos de masa*.

Un *momento de masa*, definido de forma simple, es la capacidad de la masa de rotar sobre el origen. Para que se entienda mejor, la capacidad de girar sobre el eje X depende del *momento Y*, ya que el eje Y puede rotar sobre el eje X teniendo como punto fijo el origen de coordenadas, y el

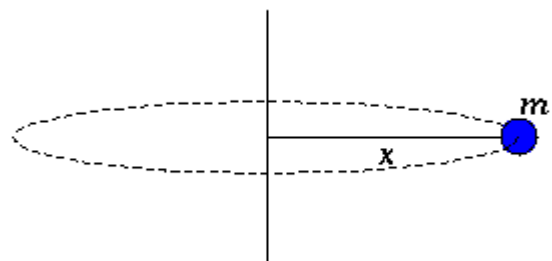


Figura 2 - El eje X (eje horizontal), tiene una masa que nos permite definir el  $M_x$  (momento x) que es la capacidad de la masa de girar (formado por guiones) sobre el eje Y (eje vertical).

*momentoX* afectará al eje Y, ya que el eje X puede girar sobre el eje Y teniendo como punto fijo el origen de coordenadas. En la figura 2, se muestra una imagen para ejemplificar lo dicho anteriormente:

Los *momentos de masa* son la suma de productos de masa por la coordenada correspondiente. Si miramos el ejemplo anterior, el momento 'X' ( $M_x$ ) es la distancia 'y' que en este caso es 0 multiplicada por la masa 'm'. Y el momento 'Y' ( $M_y$ ) se corresponde a la posición 'x' multiplicada por la masa 'm'.

Finalmente, para obtener la coordenada 'x' final del centro de gravedad, se utiliza el  $M_y$  dividido por la masa 'm', y para el punto 'y' se utiliza  $M_x$  dividido entre la masa 'm'.

## 5.2 Componentes

En el proyecto se han utilizado diferentes componentes que se detallan a continuación. Estos componentes han ido variando según el planteamiento del sistema, ya que al principio se quería diseñar un sistema integrado en el dron, pero al ver la poca viabilidad de hacerlo de esta forma (ya que el dron está sometido a factores ambientales), se decidió crear un sistema externo.

En el planteamiento inicial del problema se había pensando en utilizar un giroscopio/acelerómetro [13] para realizar el cálculo integrado en el dron. Sin embargo, como se ha dicho anteriormente, se tuvo que cambiar este planteamiento por uno más adecuado, adaptándonos al cálculo planteado en el apartado 5.

Para el nuevo planteamiento se ha utilizado el microcontrolador Genuino Mega (figura 3). Se ha elegido la casa de Arduino/Genuino, ya que cuentan con un gran soporte de la comunidad y por lo tanto, se puede obtener mucha información y ayuda en foros.



Figura 3 - Microcontrolador Genuino mega

Para la medición de las masas se han utilizado celdas de carga (figura 4). Las celdas de carga son transistores que cuentan con galgas extensométricas (metal que al flexionarse varía su resistencia) que generan una diferencia de voltaje que finalmente puede ser transformada en una medida de masa (gramos, kilogramos, etc.) a través de convertidores.

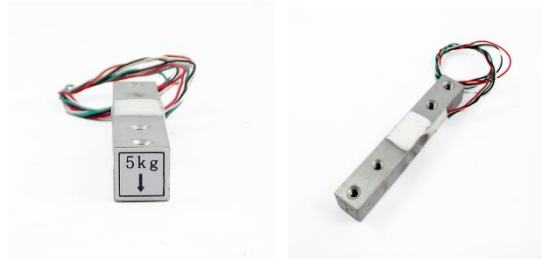


Figura 4 - Celdas de carga, parecidas a las utilizadas en el proyecto.

Para que las celdas de carga midan correctamente la carga (fuerza hacia abajo), tienen que estar separadas de la base con separadores, así pueden deformarse mejor para medir los cambios en las galgas.

Las celdas utilizadas en el proyecto tienen un rango de medición de 5 kg, como las que se muestran en la imagen. Sin embargo, existen de mayor rango en el mercado.

Para realizar la lectura de las galgas y convertir esta medida de voltaje a un valor digital, se ha utilizado un conector analógico digital (CAD), diseñado para sistemas de medida de peso, el componente HX711 [1] (figura 5) se comunica de forma serial con el microcontrolador Genuino a través de los pines de CLK y data.



Figura 5 - Módulo HX711

Para realizar la transmisión de datos de Arduino a Android, se ha utilizado el módulo hc-05. Se ha elegido este componente, por su precio y por que existen algunos tutoriales sobre como utilizarlo con arduino.



Figura 6 - Módulo hc-05

Este módulo realiza una comunicación en serie UART con la placa de Arduino.

## 5.2 Software utilizado

En la realización del software para Arduino se ha utilizado el IDE oficial de Arduino.

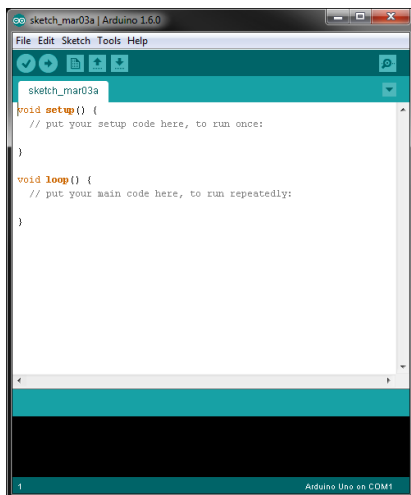


Figura 7 - Arduino IDE

Este está adaptado a las placas de Arduino, por lo tanto, nos deja elegir con qué placa estamos trabajando y verificar el código antes de subirlo a esta.

Para facilitar el uso del módulo HX711, se ha utilizado una librería (librería hx711) creada por el usuario *bodge* en Github.

Para la realización de la aplicación para android se ha utilizado App Inventor (figura 8 y figura 9), que es un aplicativo web que nos permite realizar aplicaciones para Android de manera simple.

App Inventor cuenta con dos partes, una para el diseño y otra para crear la lógica de la aplicación. Esta segunda parte esta formada por *bloques*, que se pueden unir como si fueran piezas de puzzle.

Se ha elegido realizar la aplicación con App Inventor, ya que han habido problemas con librerías y la instalación de

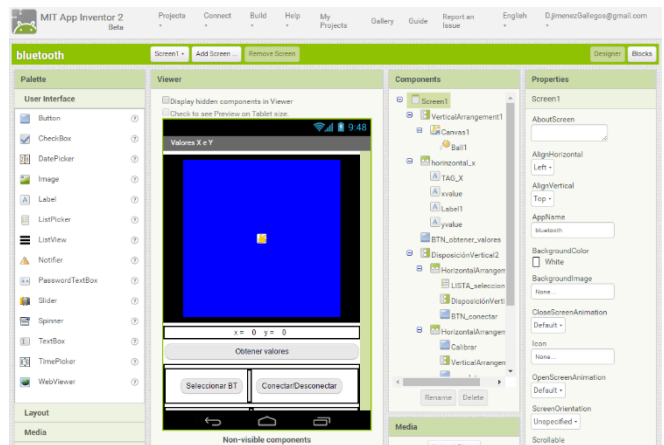


Figura 8 - Parte de diseño de App Inventor



Figura 9 - Parte de bloques App Inventor

Android Studio (aplicación oficial de google para realizar aplicaciones para Android). Así que, por problemas y falta de tiempo, App Inventor ha sido una solución simple y asequible para alcanzar los objetivos esperados en el proyecto.

## 6 RESULTADOS

Los resultados obtenidos de la aplicación desarrollada son los siguientes.



## 6.1 Prototipo cálculo centro de gravedad

En la siguiente imagen podemos ver el prototipo obtenido para realizar el cálculo del centro de gravedad.

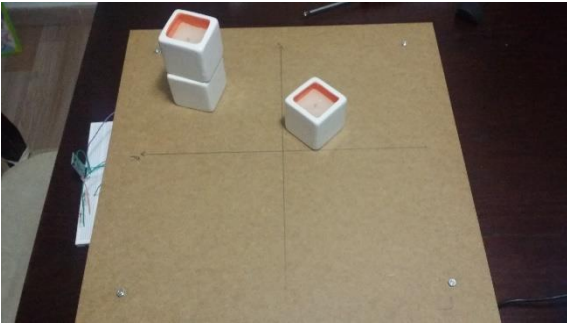


Figura 10 - prototipo final del sistema para medir el centro de gravedad con masa encima.

El prototipo planteado inicialmente, se había diseñado con placas de plástico como base (figura 11). El problema que hubo con las bases de plástico es que no eran lo suficiente rígidas, y, por lo tanto, cedían dando resultados que no eran lo suficiente fiables.

Así que, finalmente las bases de plástico las sustituí por bases de madera (figura 10), éstas pese a ser mas pesadas que las de plástico, no comportaron ningún problema ya que el peso (800 gramos aproximadamente) se reparte entre las 4 celdas de carga, y por lo tanto estamos perdiendo unos 200 gramos de medición por cada celda, lo cual no es demasiado, teniendo en cuenta que estas miden hasta 5 kg. Además, el peso inicial, es decir el de las bases es tomado como tara (peso que no se tendrá en cuenta), así que este no afecta en la medición del sistema.

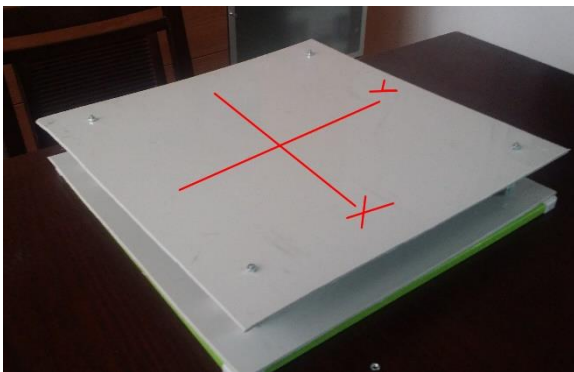


Figura 11 - Prototipo con bases de plástico

El funcionamiento de este prototipo es prácticamente como una balanza con cuatro puntos de medición, en los que en función de las mediciones del peso podemos obtener un (x,y) aplicando las fórmulas para el cálculo del centro de gravedad, teniendo en cuenta la distancias de las masas y la medición de estas.

Como se ha mencionado en el apartado 4 del documento, las celdas de carga no pueden tocar directamente la base, por lo tanto, se han tenido que levantar un poco a través de 3 placas de metal como las siguientes:



Figura 12 - Placas para levantar las celdas de carga de la base

En la siguiente imagen se puede ver como estan levantadas las celdas de carga.



Figura 13 - Celda de carga separada de la base

En la figura 14, se muestra como queda montado el sistema con el microcotrolador de arduino mega.

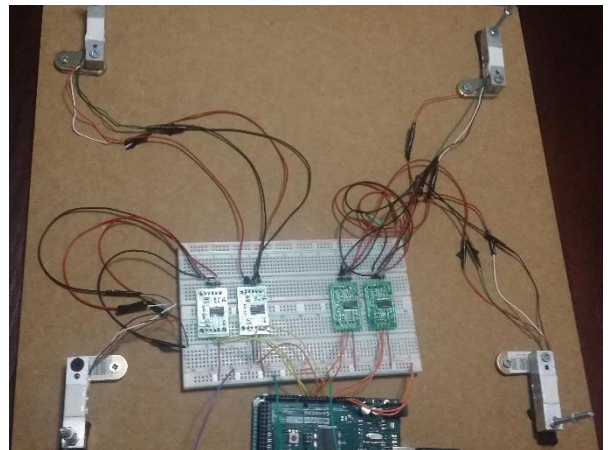


Figura 14 - Arduino mega unido a los módulos HX711 y estos a las celdas de carga

Por último, respecto a los datos que se estan obteniendo, se pueden ver en la siguientes imagen, en la que se muestran los resultados por el COM5 (monitor) del IDE de Arduino.

```

48.69
65.75
91.63
104.24
xy: 3.06;8.40
48.98
66.85
92.67
105.45
xy: 3.12;8.39
49.27
67.94
93.67
106.58
xy: 3.18;8.37

```

Figura 15 - Monitor de arduino nos muestra los valores en gramos que están leyendo las celdas de carga y el centro de gravedad donde la 'x' y la 'y' están separadas por un ';'.

## 6.2 Prototipo de bluetooth y Aplicación Android

### 6.2.1 Prototipo de bluetooth

El prototipo de bluetooth tal y como se ha mencionado se ha realizado con el módulo hc-05.

A continuación, se muestra como queda montado:

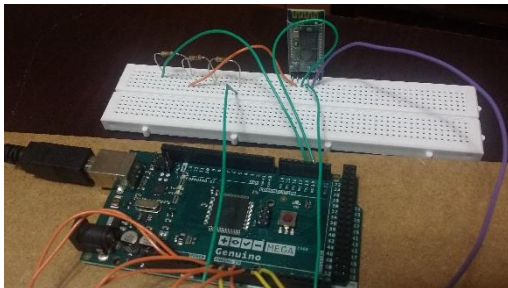


Figura 16 - módulo hc-05 conectado a arduino mega

Este prototipo se encarga de enviar datos a cualquier dispositivo Android que tenga la aplicación que se ha realizado, para así poder interpretar los datos que se reciben. El módulo se ha configurado en modo slave, es decir que espera que otro dispositivo se conecte a él.

Durante la realización del envío de datos desde Arduino a Android hubo problemas de sincronización, ya que, la aplicación Android revisa cada un segundo si tiene un dato que leer, y si es así actualiza los valores 'x' e 'y'. Mientras que el software de Arduino envía datos cada 8 segundos (aproximadamente), ya que, a parte de realizar el envío, debe hacer los cálculos del centro de gravedad.

Este problema hizo que si enviábamos primero el valor 'x' y luego el 'y', la aplicación puede que actualizara dos veces el valor 'x' o incluso, si los datos se recibían al instante estos se podían introducir seguidos en una misma variable.

A partir de este problema, se planteó pasar los datos en una cadena de caracteres, donde estos se pueden separar a través de un 'separador' en nuestro caso se ha utilizado un ';'. Así que, ahora la aplicación está recibiendo una cadena de caracteres formada por los valores 'x' e 'y' separados por un ';'.

### 6.2.2 Aplicación Android

La aplicación realizada para Android nos permite seleccionar el dispositivo con el que conectarnos, a través del botón "Seleccionar BT". A continuación, se puede ver una captura de la pantalla principal de la aplicación.

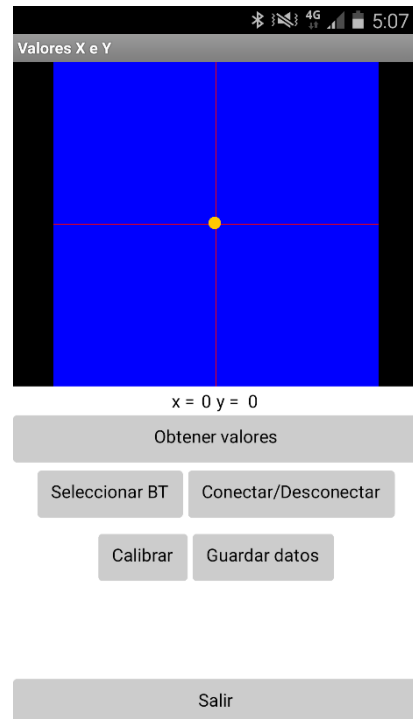


Figura 17 - Pantalla principal aplicación de Android

Una vez hemos seleccionado el dispositivo con el que conectarnos clicamos sobre el botón "conectar" (figura 18), y se deberá mostrar un mensaje conforme estamos conectados. Luego para empezar a ver los resultados del cálculo del centro de gravedad, se clic sobre "Obtener valores" (figura 19).

Por último, para dejar de recibir datos, se clicla sobre el botón “Desconectar” o si queremos salir de la aplicación se clicla sobre el botón salir.

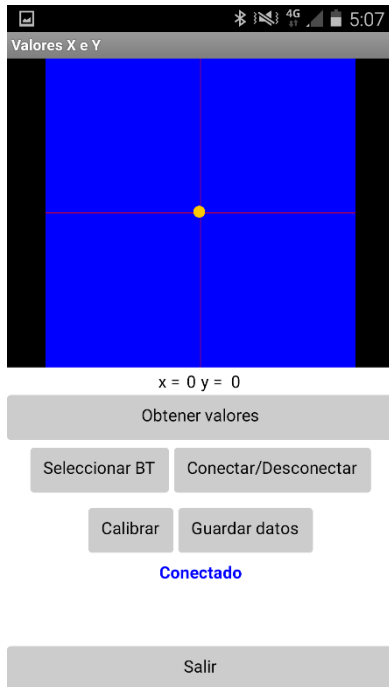


Figura 18 - Ya se ha conectado con el dispositivo hc-05

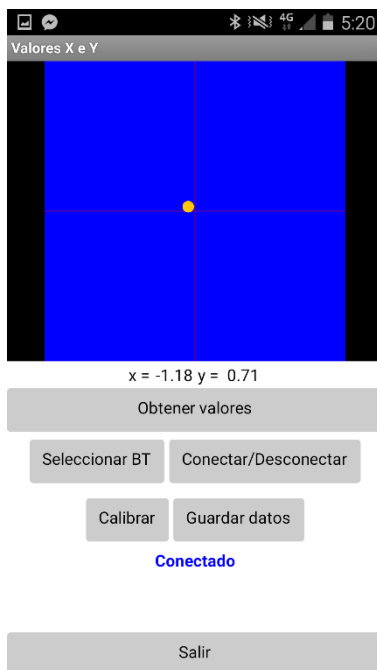


Figura 19 - Se ha clicado sobre obtener valores y vemos como  $x$ , e  $y$  han cambiado en función de la masa que se ha colocado

## 7 FUNCIONAMIENTO Y PRUEBAS

El funcionamiento del sistema es simple, primero ponemos a funcionar el sistema de Arduino con el IDE, así podemos ver que los valores generados coinciden con los que se pasan a la aplicación.

Luego arrancamos la aplicación de Android y conectamos con el sistema. Clicamos sobre *obtener valores* en la aplicación, esto enviará una señal al módulo de bluetooth de Arduino y este empezará a hacer los cálculos del centro de gravedad teniendo en cuenta la masa colocada sobre el sistema. Estos datos serán enviados a la aplicación de Android, que moverá el punto que se muestra, de manera proporcional a los datos obtenidos.

En la figura 20, se muestra una prueba con imágenes del monitor, la disposición de las masas en el sistema físico del cálculo y en la aplicación de Android.

Como se puede ver en todos los casos, los valores generados se están pasando correctamente y coinciden con la realidad (la disposición de las masas sobre el sistema físico).



Figura 20 - Se han colocado tres masas en  $x$  e  $y$  negativas

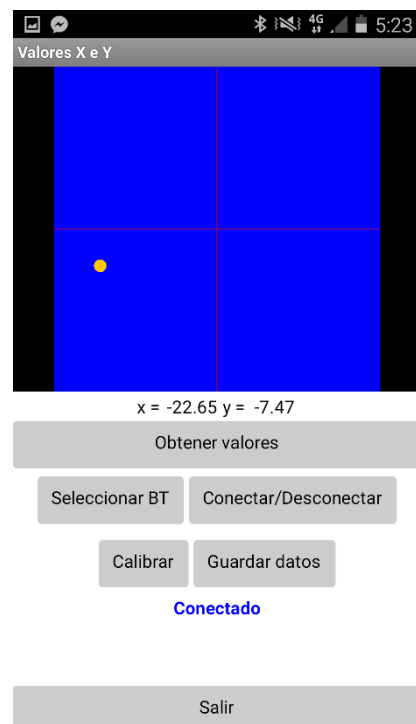


Figura 21 - Se muestra el desplazamiento en  $x$  e  $y$  de las masas colocadas

Por último, en la aplicación hay un boton calibrar que sirve para cuando ponemos mucho peso sobre el sistema

```
154.03
46.73
126.22
0.47
xy: -22.85;-7.29
155.36
47.63
126.78
0.66
xy: -22.75;-7.39
156.42
48.52
127.37
0.90
xy: -22.65;-7.47
```

☒ Autoscroll

*Figura 22 - Monitor de Arduino donde se muestran que los valores que se pasan a la aplicación, son los mismos que se generan en Arduino*

físico, ya que este puede que no vuelva a su estado inicial y, por lo tanto, si queremos calcular el centro de gravedad sobre otro dron obtendremos valores erróneos. Así que, lo que hace el botón calibrar, es coger la nueva disposición como tara.

```
Calibrando de nuevo
Ya se puede colocar de nuevo el
0.26
0.24
0.34
0.27
xy: 0.00;0.00
0.52
0.71
0.85
0.63
xy: 0.00;0.00
0.82
1.40
1.28
1.20
xy: 0.00;0.00
```

*Figura 23 - Mensajes que se producen al clicar sobre el botón calibrar y los pesos nuevos vemos que estan entorno a '0'.*

## 8 CONCLUSIÓN

El proyecto se ha podido llevar a cabo casi en su totalidad, es cierto que se han tenido que hacer cambios, como la tecnología utilizada para la creación de la aplicación, pero se a podido abarcar lo esencial en el tiempo estipulado.

Además, se ha conseguido lo que se quería inicialmente que es obtener datos sobre la distribución de la masa en un dron, y mostrarlos por una aplicación Android. Por otro lado, aunque no era un requerimiento se ha hecho un sistema portable y con un coste bajo, lo cual aumenta la facilidad de reproducción del trabajo.

Sin embargo, aun hay mucho margen de mejora como puede ser utilizar un acelerómetro para tener en cuenta las pendientes, y así cuando estemos haciendo el cálculo en un plano inclinado, cambiar el tipo de cálculo a realizar para que este sea correcto, o generar gráficas de como varía el centro de gravedad desde la aplicación de android.

Es más, si se parte de este proyecto implementar las mejoras no debería ser demasiado complicado, a menos que se decida utilizar una tecnología diferente. Aun así, partir de una idea ya es una parte importante que se puede aprovechar, por lo tanto, pienso que este trabajo, puede dar un impulso a nuevas ideas y desarrollos.

## AGRADECIMIENTOS

Me gustaría agracer a Dolores Rexachs que me haya guiado en la realización del trabajo. También me gustaría agradecer a David Matanzas Rexachs por la ayuda prestada en el planteamiento del trabajo, y por la idea de como hacer el paso de datos entre arduino y android.

## BIBLIOGRAFÍA

- [1] 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales.  
[https://cdn.sparkfun.com/datasheets/Sensors/Force-Flex/hx711\\_english.pdf](https://cdn.sparkfun.com/datasheets/Sensors/Force-Flex/hx711_english.pdf)  
[Última consulta: 11/06/2016]
- [2] Arduino library HX711 – Bodge.  
<https://github.com/bodge/HX711>  
[Última consulta: 11/06/2016]
- [3] Genuino Mega.  
<https://store.arduino.cc/product/GBX00067?language=en>  
[Última consulta: 11/06/2016]
- [4] Tutorial transmisor de celda de carga HX711, Balanza Digital.  
[http://www.naylampmechatronics.com/blog/25\\_Tutorial-trasmisor-de-celda-de-carga-HX711-Ba.html](http://www.naylampmechatronics.com/blog/25_Tutorial-trasmisor-de-celda-de-carga-HX711-Ba.html)  
[Última consulta: 11/06/2016]
- [5] Aplicación AppInventor.  
<http://ai2.appinventor.mit.edu/>  
[Última consulta: 11/06/2016]
- [6] Configuración del módulo HC-05 usando comandos AT  
[http://www.naylampmechatronics.com/blog/24\\_Configuraci%C3%B3n--del-m%C3%B3dulo-bluetooth-HC-05-usa.html](http://www.naylampmechatronics.com/blog/24_Configuraci%C3%B3n--del-m%C3%B3dulo-bluetooth-HC-05-usa.html)  
[Última consulta: 11/06/2016]
- [7] HC-05 – Bluetooth to serial port module  
[http://biblioteca.geekfactory.mx/Bluetooth\\_Bee\\_Pro/datasheet\\_hc-05.pdf](http://biblioteca.geekfactory.mx/Bluetooth_Bee_Pro/datasheet_hc-05.pdf)  
[Última consulta: 11/06/2016]
- [8] Android Studio.  
<https://developer.android.com/studio/index.html>



- [9] [Última consulta: 11/06/2016]  
Arduino with HC-05 (ZS-040) Bluetooth module – AT MODE.  
<https://developer.android.com/studio/index.html>  
[Última consulta: 11/06/2016]
- [10] Crear App para Arduino con AppInventor.  
<http://diymakers.es/crear-app-para-arduino-con-app-inventor/>  
[Última consulta: 11/06/2016]
- [11] ¿Cómo funcionan las celdas de carga?  
[http://www.basculaspoise.com/Soporte/Celdas\\_de\\_Carga.html](http://www.basculaspoise.com/Soporte/Celdas_de_Carga.html)  
[Última consulta: 21/06/2016]
- [12] Nixie  
<http://flynixie.com/>  
[Última consulta: 22/06/2016]
- [13] Amazon Prime Air  
<https://www.amazon.com/b?node=8037720011>  
[Última consulta: 22/06/2016]
- [13] Sensor inercial BNO055  
<http://tienda.bricogeek.com/medicion-inercial-imu/802-sensor-inercial-absoluto-9-dof-bno055.html>  
[Última consulta: 25/06/2016]

## APÉNDICE

### A.1 Tabla de costes de los componentes

A continuación, se muestra una tabla con los precios aproximados de cada componente, estos pueden variar en función de donde se compren y los gastos de envío, pero en general los podemos encontrar por los precios de la tabla.

| Componente          | Precio |
|---------------------|--------|
| 1 - Arduino Mega    | 34.50€ |
| 4 - Celdas de carga | 13 €   |
| 4 - Hx711           | 5 €    |
| 1 - HC-05           | 7,20   |
| Coste total         | 59.70€ |