

# INFORME CRÍTICO DE DISEÑO



**Mentora: Candelaria Palacios Calzado**

**Equipo: Julio Quintero Díaz**

**Rubén Vargas Núñez**

**Pablo Pacheco Reyes**

**Mateo Moreno Castaño**

**Patrocinadores:**

AMPA  
Wenceslao  
Benítez



# ÍNDICE

<b>INTRODUCCIÓN/ ANTECEDENTES .....</b>	<b>3</b>
Introducción .....	3
El equipo .....	3
Reparto de tareas .....	3
Organización del trabajo.....	4
<b>PROYECTO CANSAT .....</b>	<b>5</b>
Misiones .....	5
Misión primaria .....	5
Misión secundaria .....	5
Resultados esperados.....	7
<b>DISEÑO MECÁNICO DEL CANSAT Y DE LA ESTACIÓN DE TIERRA .....</b>	<b>8</b>
Estructura Externa .....	8
Estructura interna .....	10
Componentes .....	11
<b>DISEÑO ELÉCTRICO.....</b>	<b>14</b>
Componentes eléctricos .....	15
Placas PCB CanSat y Estación de Tierra .....	15
<b>PROGRAMACIÓN. LENGUAJE DE PROGRAMACIÓN Y DIAGRAMA DE FLUJO DEL PROGRAMA. BREVE EXPLICACIÓN DEL FUNCIONAMIENTO.....</b>	<b>15</b>
Diagrama de flujo.....	15
Programación.....	17
Cansat .....	18
Estación de tierra.....	21
<b>SISTEMAS IMPLICADOS EN LA ESTACIÓN DE TIERRA .....</b>	<b>21</b>
Explicación .....	21
Componentes .....	23
<b>SISTEMAS DE ATERRIZAJE. PARACAÍDAS U OTROS SISTEMAS. PLANTEAMIENTO DE LA IDEA. ....</b>	<b>23</b>
Diseño.....	23
Cálculos.....	23
Material .....	25
<b>PRESUPUESTO DETALLADO. DIFERENCIAS CON EL INICIAL .....</b>	<b>26</b>
Lista Presupuesto .....	26

<b>FINANCIACIÓN Y APOYO TÉCNICO .....</b>	<b>29</b>
<b>Financiación.....</b>	<b>29</b>
<b>Apoyo técnico.....</b>	<b>31</b>
<b>PROYECTO CIENTÍFICO. DESCRIPCIÓN DEL PLAN DE ACTUACIÓN DEL DÍA DEL LANZAMIENTO. PLANTEAR SOLUCIONES ANTE POSIBLES IMPREVISTOS.....</b>	<b>31</b>
<b>Pruebas .....</b>	<b>31</b>
<b>Funcionamiento del CanSat.....</b>	<b>31</b>
<b>Realización de la misión principal.....</b>	<b>31</b>
<b>Comunicación por radio .....</b>	<b>32</b>
<b>Realización de la misión secundaria .....</b>	<b>32</b>
<b>Pruebas del paracaídas.....</b>	<b>32</b>
<b>ESTADO DE LA DIFUSIÓN Y DIVULGACIÓN DEL PROYECTO. FOTOS, VÍDEOS, ENLACES DEL PROGRESO.....</b>	<b>32</b>
<b>AGRADECIMIENTOS .....</b>	<b>36</b>
<b>BIBLIOGRAFÍA: REFERENCIAS E INFORMACIÓN UTILIZADA .....</b>	<b>37</b>

## **INTRODUCCIÓN/ ANTECEDENTES**

### **Introducción**

En nuestro instituto, nuestros compañeros han participado con anterioridad a convenciones y eventos como las ferias de la ciencia y actividades de divulgación. En el curso pasado nos dieron la oportunidad de exponer en la feria de la ciencia de Sevilla, pero debido a la pandemia no pudimos participar, pero este curso, en 1º de bachillerato hemos decidido unirnos a un proyecto más grande teniendo en cuenta que no tendríamos la oportunidad de participar en próximos eventos, por eso elegimos el proyecto CanSat. Lo elegimos ya que sería la primera vez que hacíamos un proyecto de tal magnitud y así probar hasta dónde somos capaces de llegar.

### **El equipo**

El equipo está formado por cuatro miembros: Julio Quintero, Pablo Pacheco, Mateo Moreno y Rubén Vargas. Al principio decidimos en conjunto buscar información sobre el Proyecto CanSat, ejemplos de anteriores proyectos para tener una referencia y qué haríamos para la misión principal y secundaria con los respectivos sensores que necesitábamos, pero no los modelos específicos; todo ello para empezar con la lluvia de ideas y elegir en qué consistiría nuestro proyecto.

Después decidimos separar el trabajo de esta manera:

- ❖ Julio Quintero y Rubén Vargas se encargarían de la programación.
- ❖ Mateo Moreno Y Pablo Pacheco se encargarían de la búsqueda de los sensores específicos necesarios para el CanSat, de su diseño 3D, prototipado...

Nuestra mentora, tras informarle de nuestra distribución de trabajos, nos recomendó que lo mejor era buscar los sensores en conjunto para poder proseguir con la programación y el diseño 3D lo más pronto posible. El principal problema que tuvimos es que tardamos en decidir si participar en este proyecto o no, y a causa de ello, hemos perdido bastante tiempo y ha sido mucho más complicado trabajar. Sumado a la cantidad de problemas y fallos que hemos tenido que afrontar, han hecho que el camino hasta aquí haya sido arduo y complicado, pero finalmente satisfactorio.

### **Reparto de tareas**

Después de replantear las tareas teniendo en cuenta los conocimientos y capacidades de cada integrante para lograr una mayor eficiencia, la distribución quedó de esta manera que se muestra en la imagen:



## Organización del trabajo

Debido a la pandemia hemos decidido organizarnos para que pudiéramos hacer todo desde nuestras casas:

- ☐ Hemos creado un grupo de Telegram en conjunto con nuestra mentora para poder mantenerla informada de cualquier avance o imprevisto que sucediera a lo largo del proyecto, también es una manera sencilla de comunicarnos entre todos sin tener que reunirnos online de manera oral.
- ☐ Hemos realizado reuniones online desde la plataforma Discord, en la que compartimos ideas y resolvemos dudas que hayan surgido en la elaboración del proyecto, estas reuniones suelen ocurrir los lunes, miércoles y viernes de 6:30 a 8:00 para poder trabajar en conjunto, aunque a veces optamos por trabajar más tiempo cuando podemos. Esta es nuestra mejor opción ya que no podemos reunirnos de manera presencial debido a la pandemia.
- ☐ Para la elaboración del informe CDR, como teníamos que escribirlo entre todos por la distribución de tareas, hemos decidido trabajar en Google Docs., que es una plataforma en la que todos podemos escribir y trabajar a la vez para evitar estar utilizando documentos diferentes.

- ❑ A su vez, hemos creado un Trello para poder organizar los trabajos y partes que tenía que desarrollar cada uno. Además, es una buena forma para no olvidarnos de nada y tener todo ordenado.

## **PROYECTO CANSAT**

CanSat es un concurso diseñado para simular algunos de los objetivos principales de un satélite real, pero teniendo en cuenta una limitación importante: el tamaño de una lata. Es decir, hay que incluir los sensores, la alimentación y la comunicación necesaria en el espacio reducido de una lata.

En nuestro caso, el proyecto tiene un objetivo medioambiental. En la misión secundaria será posible comprobar nuestra preocupación por el mismo y de algún modo concienciar a los demás mediante la realización de una investigación basándonos en datos objetivos y probados. Además, hemos tomado como referencia el Acuerdo de París para en el que se hace hincapié en la necesidad de mantener el calentamiento global muy por debajo de los 2°C e intentar limitarlo todo lo posible hasta llegar a 1, 5° C.

### **Misiones**

A continuación, se desarrollará las dos misiones principales en las que se basa el proyecto:

#### **Misión primaria**

Su objetivo es medir dos parámetros durante el descenso del CanSat, que serán la presión y la temperatura. La información captada se transmite a la estación de tierra a través de telemetría durante el transcurso de la bajada.

Hemos elegido la placa Arduino UNO como principal componente de hardware, como lenguaje de programación Arduino (c/c++) y como entorno de desarrollo vscode.

#### **Misión secundaria**

En este caso, la misión no está elegida por la organización, sino por nuestro grupo de trabajo. En primer lugar, queríamos centrar esta misión secundaria en el medio ambiente, puesto que somos un grupo bastante preocupado por él y nos gustaría concienciar a los demás mediante la realización de una investigación basados en datos objetivos y probados.

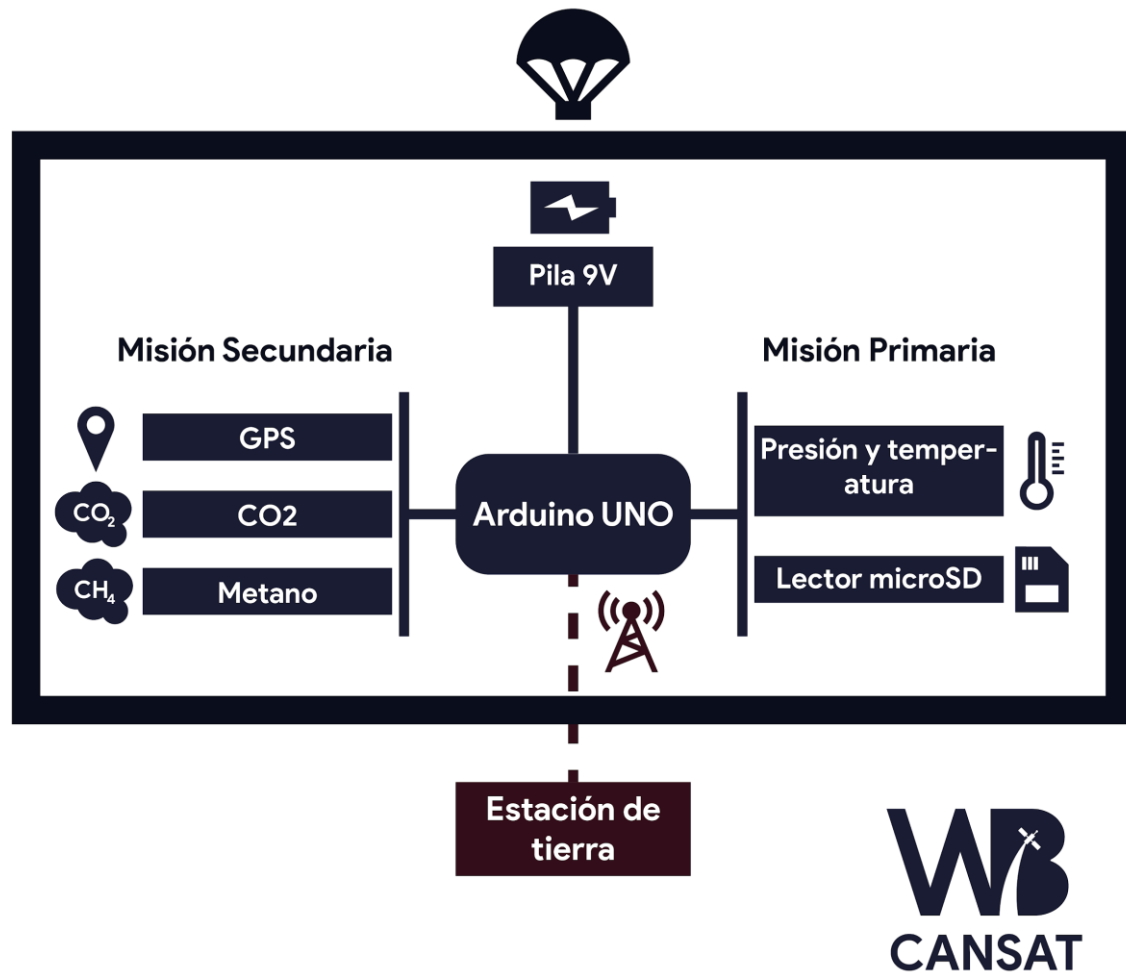
En un principio, la idea era examinar la vegetación de la zona desde el satélite. Para ello, mediríamos la eficiencia de la fotosíntesis de las plantas, basándonos en la fluorescencia de la clorofila que éstas emiten. Tras una exhaustiva investigación sobre este estudio, rechazamos la idea debido a la complejidad para encontrar

sensores que no estuvieran en desarrollo y que se adaptaran al presupuesto máximo impuesto en las bases del concurso.

La segunda idea era comprobar el metano y otros gases nocivos emitidos por los vertederos e industrias, mediante un espectrómetro de imágenes aéreas de alto rendimiento. Pero nos ocurrió lo mismo que con la primera idea: los sensores necesarios suponían un alto costo.

Finalmente, decidimos adaptar la segunda idea utilizando sensores MQ, pero en un futuro nos gustaría utilizar infrarrojos ofreciendo una mayor precisión en la medición de los parámetros. Todo ello, sin perder el objetivo principal de nuestra misión secundaria: identificar el aumento de emisiones de gases de efecto invernadero en vertederos, industrias, ciudades concurridas y otro tipo de zonas peligrosas que perjudiquen el medio ambiente y que estén por encima de las adoptadas en el Acuerdo de París.

También podríamos extrapolar la idea explicada anteriormente, suponiendo que el CanSat fuese un satélite espacial, para investigar las atmósferas de otros planetas y comparar la presencia de los gases que producen contaminación en ellos, determinando así si sería factible la vida en éstos.



## Resultados esperados

Con nuestra misión esperamos obtener durante el descenso los valores de concentración de CH<sub>4</sub> (metano) y CO<sub>2</sub> (dióxido de carbono) cada segundo enviando los datos por telemetría a la estación de tierra. Con los datos obtenidos seremos capaces de determinar los niveles de concentración para cada uno de ellos y comprobar que valores no cumplen con los límites establecidos. Además de observar la relación que pueden tener estos parámetros con la temperatura de la zona específica.



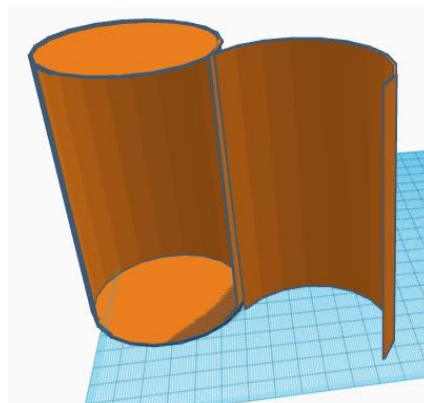
## DISEÑO MECÁNICO DEL CANSAT Y DE LA ESTACIÓN DE TIERRA

Al empezar a diseñar el CanSat nos dimos cuenta que tenía que tener unas medidas específicas, nuestro satélite no podía superar una altura de 115mm y 65mm de diámetro. Esto nos dio un poco de dificultad, pero con trabajo pudimos encajar todo. Aprovechamos que nuestro instituto disponía de una impresora 3D, para construir el diseño mediante diversas aplicaciones, hasta acabar con el resultado más profesional posible. Hemos decidido que nuestra estructura interna y externa están separadas, debido a que a la hora de manipularlos y trabajar con ellos va ser mucho más fácil que trabajar en un recipiente hermético de paredes.

### Estructura Externa

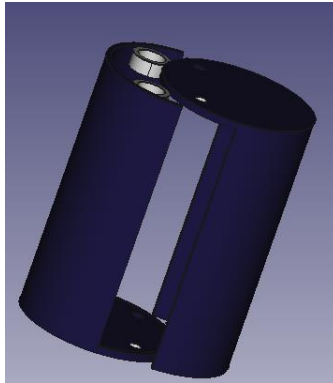
Para empezar con la estructura externa, nos dirigimos a un programa que estábamos bastante familiarizados (Tinkercad), debido a que ya habíamos estado diseñando con él antes. Es un programa bastante sencillo y fácil para plasmar las ideas antes de pasarla a Freecad.

La estructura la hemos diseñado varias veces, para ver cuál se ajusta más a nuestras necesidades. Primero pensamos en hacer un prototipo que se encaja mediante deslizamiento hacia arriba, pero lo descartamos porque pensábamos que podía llevarse los cables. Ese primer prototipo lo dejamos en un croquis a mano y no lo pasamos a los programas de edición. Después pensamos en hacer uno con bisagras, pero lo vimos muy difícil para ponerlas en la carcasa de plástico. Aun así, no lo descartamos como el otro y lo pasamos al programa de Tinkercad, junto a dos prototipos más.



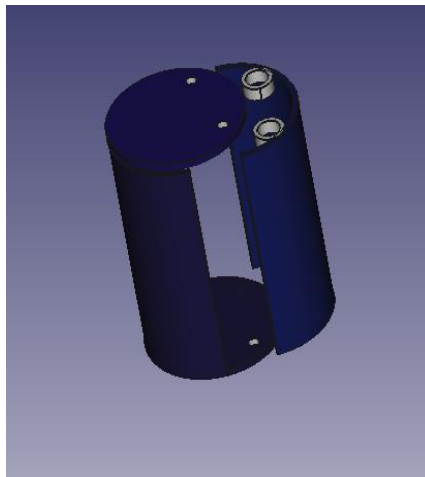
**Prototipo 1 (Tinkercad)**

Los dos últimos fueron los mejores, debido a que iban fijados con tornillos y era mucho más fácil a la hora de manipularlos. Estos dos los pasamos a un programa un poco más profesional, pero antes de poder llevarlo a cabo, tuvimos que ver bastantes tutoriales para poder empezar a diseñar en FreeCAD.



**Prototipo 2 (FreeCAD)**

Hemos decidido poner las paredes laterales de 2 mm y las inferiores de 3mm en las bases.



**Prototipo 3 (FreeCAD)**

Para la impresión en 3D hemos probado con dos tipos de filamento, PLA y ABS, de 1.75 mm con el software CURA. Hemos tenido dificultades para imprimir ya que la impresora 3D del centro no soportaba bien la alta temperatura del ABS. Sin embargo, hemos conseguido imprimir un prototipo en PLA, gracias a Manuel y su impresora.

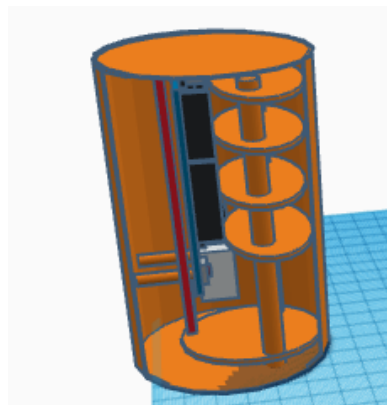
### Prototipo 3(Impreso)



### Estructura interna

Para realizar la estructura interna al igual que para realizar la carcasa, lo esbozamos todo en papel y después lo pasamos todo a un programa de edición 3D.

Para crear la estructura, nos centramos en dos factores importantes. El primero, que fuera fácil de manipular y la segunda que pudiéramos poner cada uno de los componentes sin ningún problema y que todo estuviera muy bien organizado. Al igual que con la carcasa comenzamos trasladando el diseño a digital con Tinkercad e intentamos mejorarlo lo máximo posible antes de pasarlo a FreeCAD. Nuestra intención ha sido hacer un sitio específico para colocar cada sensor para a la hora de ubicarlo y ver si hay algún fallo tenerlo localizado y evitar tocar los otros sensores debido a que podríamos desconectar algún cable.



## Componentes

### BMP280

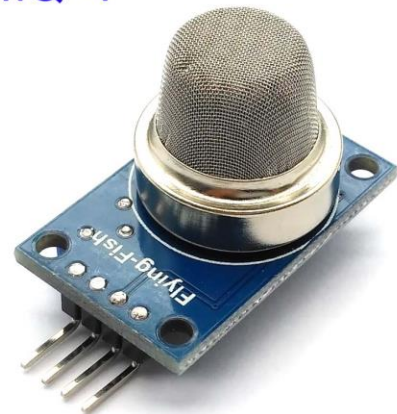


**Rango de humedad relativa:** 0 -100%  
**El rango de temperatura:** es de -40 °C-+ 85 °C  
**Medición de humedad tiempo de respuesta:** 1S  
**Lag:** humedad relativa 2%  
**Rango de presión:** 300-1100 hPa  
**Precisión de presión absoluta:** Valor absoluto  $\pm 1$ hPa (después de soldar)  
**Precisión absoluta de la temperatura:**  $\pm 0,5$  °C (a 25 °C)  
**Interfaz de comunicación:** I2C, SPI  
**Voltaje de funcionamiento:** DC3.3V

### MQ-4

**Tipo de sensor:** Semiconductor.  
**Requisitos de energía:** VCC - 5V  $\pm 0.1$   
**Salida DO:** TTL digital 0 y 1 (0,1 y 5 V)  
**Salida AO:** 0,1-0,3 V (en relación con la contaminación), la concentración máxima de un voltaje de aproximadamente 4 V.  
**Gas de detección:** gas natural / metano  
**Concentración de detección:** 200-10000ppm (gas natural / metano)  
**Interfaz:** 1 entrada compatible TTL (HSW), 1 salida compatible TTL (ALR)  
**Consumo de calefacción:** menos de 750 mw  
**Temperatura de funcionamiento:** 14 a 122 ° F (-10 a 50 ° C)  
**HR:** Humedad relacionada con menos del 95% de HR  
 La concentración de oxígeno O2 es del 21% (condición estándar): la concentración de oxígeno puede afectar la sensibilidad  
**Resistencia de carga:** 20 K $\Omega$   
**Resistencia de detección Rs:** 10K $\Omega$ - 60K $\Omega$  (1000ppm CH4)  
**Tiempo de precalentamiento:** más de 24 horas

### MQ-4



## GPS NEO-6



**Intensidad de corriente:** 37 mA  
**Tensión de alimentación:** 2.7V-3.6V  
**Tiempo encendido cold/warm:** 30s  
**Frecuencia medición:** 5Hz  
**Presión de medición:**  $\pm 2.5m$  en velocidad 0,1m/s  
 orientación 0, 5°

## Zumbador

**Voltaje:** 3.5V-5.5V  
**Amperaje:** 25mA/5VDC  
**Frecuencias:** 2300Hz ~ 4500Hz  $\pm$  500Hz  
**Sonido:** entre 2kHz y 5KHz



## MQ-135

**Detección de gas:** Amoníaco / Sulfuro / Vapor de benceno/CO<sub>2</sub>  
**Detección de Concentración:** 10-1000ppm  
**Voltaje de entrada:** DC5V  
**Consumo de energía (corriente):** 150mA  
**DO salida:** TTL cantidades digitales 0 y 1 (0.1 y 5V)  
**Salida AO:** 0.1-0.3V (relativamente no contaminante), el voltaje de concentración más alto es aproximadamente 4V  
**Usando el chip:** AT89S52  
**Cristal:** 11.0592MHZ  
**Velocidad de transmisión:** 9600  
**Entorno de compilación:** Keil

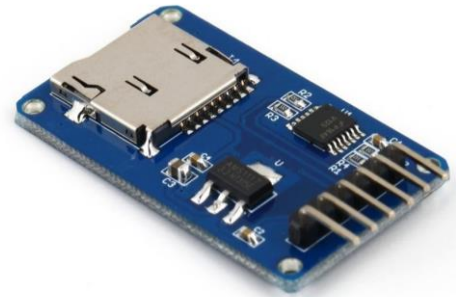
### Lector microSD Arduino

Soporte para tarjeta: Micro SD ( $\leq 2G$ ), tarjeta Micro SDHC ( $\leq 32G$ ) (Tarjeta de alta velocidad)

Voltaje: 5V o 3.3V

La fuente de alimentación: 4.5V  $\sim$  5.5V, placa de circuito del regulador de voltaje 3.3V

La interfaz de comunicación es una interfaz SPI estándar  
4 orificios de posicionamiento del tornillo M2



### Radio apc 220

**Distancia de comunicación:** 1000m (2400bps)

**La potencia de salida:** 20 mW

**La frecuencia:** 418MHz a 455MHz

**Canales:** más de 100 canales

**Modulación:** GFSK

**Interfaz:** UART / TTL

**Supera el búfer de datos :** 256 bytes

Apto para grandes transferencias de datos

### Arduino Uno

**Microcontrolador:** ATmega328P

**Voltaje de funcionamiento:** 5 V

**Voltaje de entrada (recomendado):** 7-12 V

**Voltaje de entrada (límite):** 6-20 V

Pines de E / S digitales 14 (de los cuales 6 proporcionan salida PWM)

Pines de E / S digitales PWM 6

Pines de entrada analógica 6

**Corriente CC por pin de E / S:** 20 mA

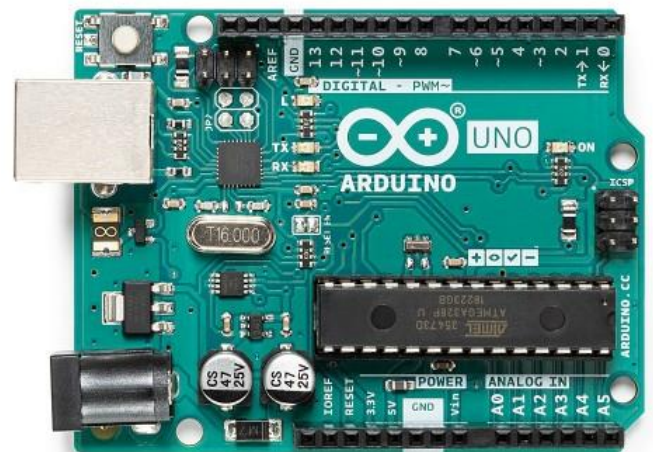
**Corriente CC para pin:** 3.3V 50 mA

**Memoria Flash:** 32 KB (ATmega328P) de los cuales 0,5 KB utilizados por el gestor de arranque

**SRAM:** 2 KB (ATmega328P)

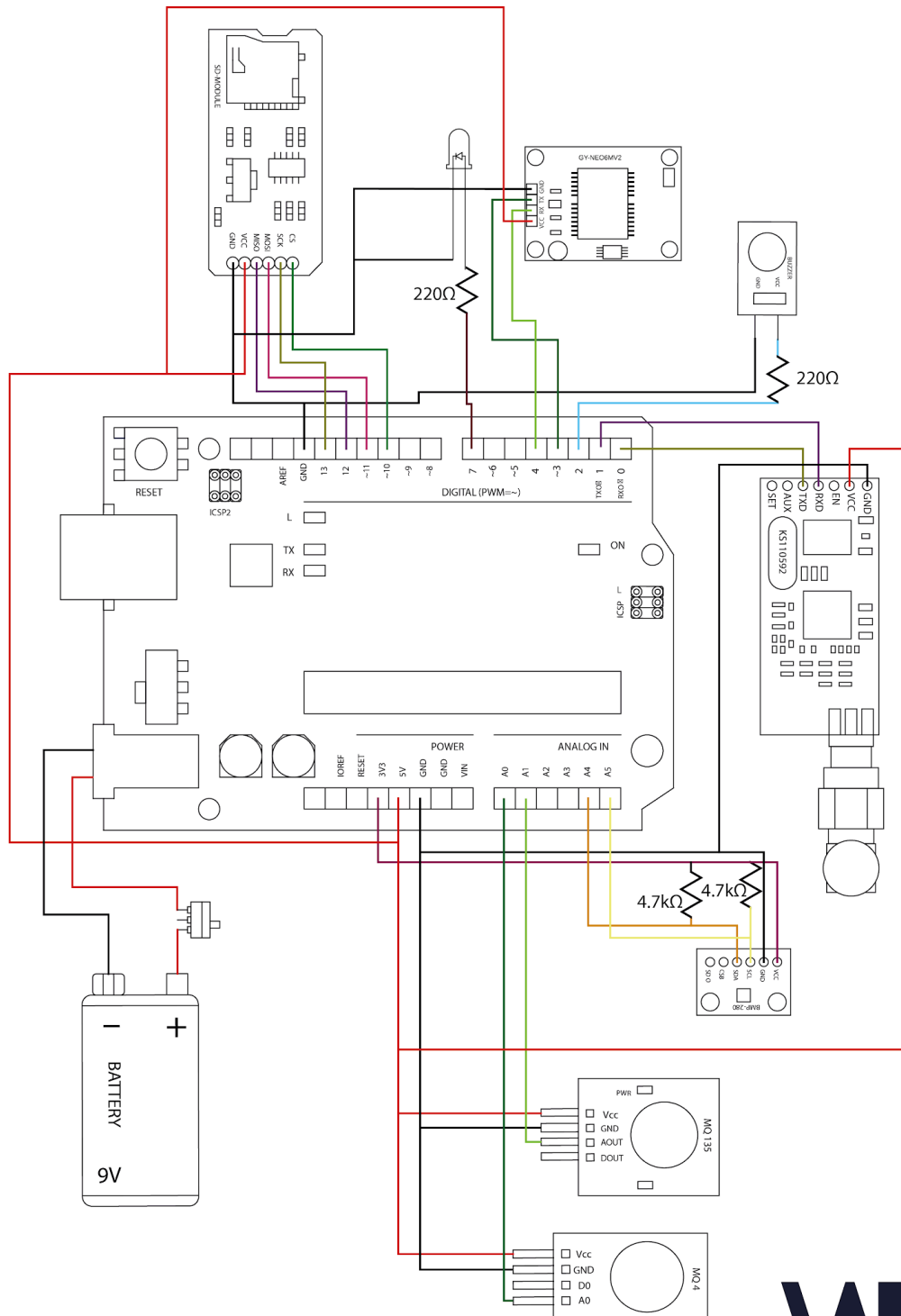
**EEPROM:** 1 KB (ATmega328P)

**Velocidad de reloj:** 16 MHz





## DISEÑO ELÉCTRICO



## Componentes eléctricos

Hemos utilizado la placa Arduino uno para conectar los componentes que se encuentran ilustrados en el diagrama superior.

### Placas PCB CanSat y Estación de Tierra

#### ○Sistema de alimentación del CanSat

El CanSat está alimentado por una batería de 9V, ya que hay que tener en cuenta el consumo energético debido a la gran cantidad de sensores que van conectados al mismo. teniendo en cuenta el consumo de los sensores, creemos que esto bastará para garantizar la energía suficiente durante el viaje. Sin embargo, nos falta el conector Jack para unir la batería a la placa.

#### ○Sistema de Transmisión y recepción de datos elegido

Para establecer la comunicación entre el CanSat y la estación de tierra hemos elegido el módulo apc220, debido a su bajo costo, y a su característica estrella: la capacidad para enviar información a más de un kilómetro de distancia transmitiendo a una frecuencia de alrededor 433 MHz, esto dependerá de la banda que se nos ceda en el momento del despegue.

## PROGRAMACIÓN. LENGUAJE DE PROGRAMACIÓN Y DIAGRAMA DE FLUJO DEL PROGRAMA. BREVE EXPLICACIÓN DEL FUNCIONAMIENTO

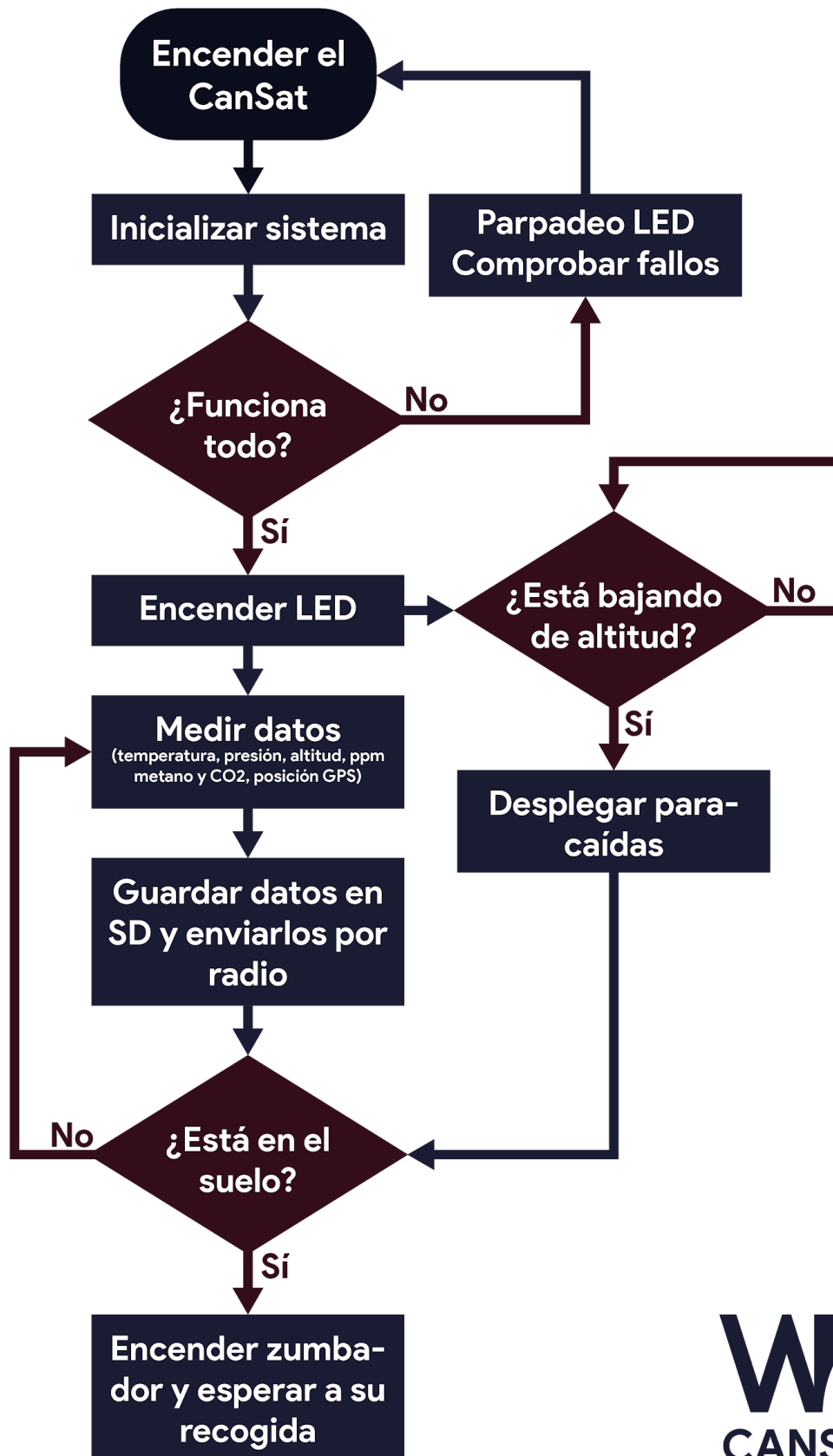
El lenguaje de programación que hemos utilizado para nuestro proyecto es el propio lenguaje de Arduino, que consiste en una mezcla entre C y C++; ya que como se ha mencionado anteriormente, hemos utilizado una placa Arduino UNO R3 como placa. Sin embargo, el entorno de desarrollo ha sido Visual Studio Code junto con las extensiones para Arduino y C/C++, puesto que trae muchas funcionalidades y herramientas de las que carece el entorno propio de Arduino.

A su vez, para la estación de tierra usaremos Processing, un lenguaje de programación de código abierto basado en java, que utilizaremos para tratar los datos recogidos por radio. El entorno para este programa seguirá siendo Visual Studio Code.

### Diagrama de flujo

En la siguiente página se expondrá el diagrama de flujo y se explicará detenidamente.





Como se puede ver en el diagrama de flujo de la página anterior, cuando el CanSat sea encendido verificará que todos los sensores y componentes funcionan perfectamente. Si es así, una bombilla LED se encenderá para mostrar el correcto funcionamiento; si detecta algún fallo, ésta parpadeará intermitentemente.

Una vez encendido el CanSat este estará listo para ser lanzado, y empezará a tomar medidas constantemente, que serán guardadas en la SD y se enviarán por radio a la estación de radio (cuando sea posible). Si detecta que la nueva medida es inferior que la anterior (significaría que el CanSat está bajando) y que la altura supera los 500 metros, este desplegará el paracaídas y guardará que ha llegado a la altura máxima.

Para detectar que el sensor ha llegado al suelo y deje de enviar datos, se compararán la altitud medida en el ciclo con la medida del anterior ciclo. Si es distinta significa que sigue bajando por lo que seguirá mandando datos, pero si es igual y anteriormente ha alcanzado la altitud máxima significará que ha llegado al suelo y dejará de enviar los datos a la estación de tierra y guardarlos en la SD, encendiendo a su vez un zumbador para ayudar a localizarlo.

El código para estas decisiones sería este:

```
if (!reach_ground){  
  
    //Read sensors and save data  
}  
  
if (pre_altitude_val > altitude_val && altitude_val > 500 && !top_alt){  
  
    top_alt = true  
    //Deploy parachute  
}  
  
else if (pre_altitude == altitude && top_alt){  
  
    reach_ground = true  
    //Turn on the buzzer  
}
```

Aunque el código al completo se mostrará a continuación, este fragmento puede servir de ayuda para entender el diagrama en mayor profundidad.

## Programación

Una vez expuesto el diagrama de flujo vamos a mostrar el código que hemos programado. Antes de empezar decir que el código al completo se puede ver en nuestro GitHub (<https://github.com/WBCanSat/CanSatProject>), donde, además, se pueden encontrar más documentos e información sobre el proyecto; sin

embargo, aquí explicaremos prácticamente la totalidad de este para que sea posible su comprensión.

Dicho esto, comenzaremos con la programación que estará presente en el CanSat.

## CanSat

### Misión primaria

```
1  #include <Adafruit_BMP280.h>
2  #include <SPI.h>
3  #include <Wire.h>
4  #include <SD.h>
5  #include <SoftwareSerial.h>
6  #include <TinyGPS.h>
7
8  #define CS_pin 10
9  #define LED_pin 7
10 #define buzz_pin 2
11 #define RX_pin 4
12 #define TX_pin 3
13 |
14 int seconds;
15 float altitude_val, pre_altitude_val, temperature_val, pressure_val;
16 boolean top_alt = false;
17 boolean reach_ground = false;
18
19 File myFile;
20 File topFile;
21 Adafruit_BMP280 bmp;
22 TinyGPS gps;
23 SoftwareSerial softSerial(RX_pin, TX_pin);
24
```

Primeramente, incluimos todas las librerías que utilizaremos (en el README están descritas cada una de ellas), que serían la librería de Adafruit para el sensor BMP280, las librerías SPI y Wire que son utilizadas para administrar las conexiones I2C, SPI y poder crear puertos con varios pines y la SD para la escritura de los datos en la tarjeta SD.

A continuación, definimos los pines que estaremos usando en las conexiones, algunos usan por defecto ciertos pines, por lo que no será necesario definirlo; y también definiremos ciertas variables que necesitaremos, además de ciertas clases y valores para los sensores y componentes del CanSat.

```

25 void setup(){
26
27     //Initialize sensors and board
28
29     Serial.begin(9600);
30     pinMode(LED_pin, OUTPUT);
31     pinMode(CS_pin, OUTPUT);
32     pinMode(buzz_pin, OUTPUT);
33
34     //Check if it is working
35
36     if (bmp.begin(0x76) && SD.begin(CS_pin)) {
37         digitalWrite(LED_pin, HIGH);
38     }
39     else {
40
41         while(1) {
42
43             digitalWrite(LED_pin, HIGH);
44             delay(500);
45             digitalWrite(LED_pin, LOW);
46             delay(1000);
47         }
48     }
49
50 }
51

```

Comenzamos el setup inicializando el puerto serie y los pines que necesitamos, además comprobamos si funcionan todos los sensores, si es así, se encenderá un LED de forma constante y pasará a la siguiente parte del código; sino el LED empezará a parpadear en bucle, que no dejará que el código continúe para poder comprobar el motivo de los fallos.

```

52 //Create datalog file
53
54 if(!SD.exists("datalog.csv")) {
55
56     myFile = SD.open("datalog.csv", FILE_WRITE);
57     if (myFile) {
58
59         myFile.println("Time(s),Temperature(C),Pressure(hPa),Altitude(m)");
60         myFile.close();
61     }
62 }
63
64
65 //Set parameters of BMP280
66
67 bmp.setSampling(Adafruit_BMP280::MODE_NORMAL, /* Operating Mode. */
68               Adafruit_BMP280::SAMPLING_X2, /* Temp. oversampling */
69               Adafruit_BMP280::SAMPLING_X16, /* Pressure oversampling */
70               Adafruit_BMP280::FILTER_X16, /* Filtering. */
71               Adafruit_BMP280::STANDBY_MS_500); /* Standby time. */
72
73 }

```

Una vez comprobado, aún en el setup, creamos el archivo en el que se guardarán los datos por la SD y establecemos los parámetros de varios sensores.

```

75 void loop(){
76
77   if (!reach_ground) {
78     // Read sensors
79
80     altitude_val = bmp.readAltitude(1013.25);
81     temperature_val = bmp.readTemperature();
82     pressure_val = bmp.readPressure()/100;
83
84
85     //Save information SD
86
87     myFile = SD.open("datalog.csv", FILE_WRITE);
88
89     if(myFile) {
90       myFile.print(seconds);
91       myFile.print(",");
92       myFile.print(temperature_val);
93       myFile.print(",");
94       myFile.print(pressure_val);
95       myFile.print(",");
96       myFile.println(altitude_val);
97
98       myFile.close();
99     }
100
101
102     //Send information Radio
103
104     Serial.print(seconds);
105     Serial.print(",");
106     Serial.print(temperature_val);
107     Serial.print(",");
108     Serial.print(pressure_val);
109     Serial.print(",");
110     Serial.println(altitude_val);
111   }
112

```

Ahora sí, comenzaremos con el loop (la parte del código que se repetirá constantemente), primeramente, comprobamos si ha llegado al suelo, puesto que cuando llegue queremos que deje de enviar datos (la comprobación se verá más adelante); y si no lo ha hecho leerá los sensores y los escribirá en la SD y los mandará por radiofrecuencia a la estación de tierra.

```

106 //check if it has reached the top
107
108 if (pre_altitude_val > altitude_val && !top_alt && altitude_val > 500) {
109
110   top_alt = true;
111
112   if(!SD.exists("top_alt.txt")) {
113
114     topFile = SD.open("top_alt.txt", FILE_WRITE);
115     if (topFile) {
116
117       topFile.print("The CanSat has reached the top at ");
118       topFile.print(altitude_val);
119       topFile.print("m");
120       topFile.close();
121
122     }
123   }
124 }

```

Después de ello, comprobará si ha llegado a la altitud máxima, si es así creará un nuevo archivo en el que escribirá la altitud máxima y definirá la variable booleana top\_alt como cierta (esta se refiere a si ha llegado a la altura máxima). Como en el diagrama de flujo ya se explicó el criterio para comprobar la altura máxima, no nos detendremos en ello.

```
126 //check if it's on the ground and turn on the buzzer
127
128 else if(pre_altitude_val == altitude_val && top_alt) {
129     tone(buzz_pin, 600);
130     reach_ground = true;
131 }
132
133 pre_altitude_val = altitude_val;
134
135 delay(1000);
136 seconds ++;
137 }
138 }
```

Finalmente comprobará si ha llegado al suelo, y si es así encenderá el zumbador para facilitar la búsqueda del lugar de caída del CanSat y establecerá la variable booleana `reach_ground` como verdadera para que no envíe más datos, ni los escriba en la SD. Por último, establecerá el valor de la altitud de ese ciclo como el anterior, esperará durante un segundo, sumará un segundo al contador y volverá a empezar de nuevo el bucle (loop).

### Misión secundaria

La misión secundaria todavía no está programada, seguimos en su desarrollo; sin embargo, será simplemente añadir las librerías necesarias para cada sensor y medir sus variables; además de enviarlos por radiofrecuencia y escribirlos en la SD. El cuerpo del programa se mantendrá como se ha visto anteriormente.

### Estación de tierra

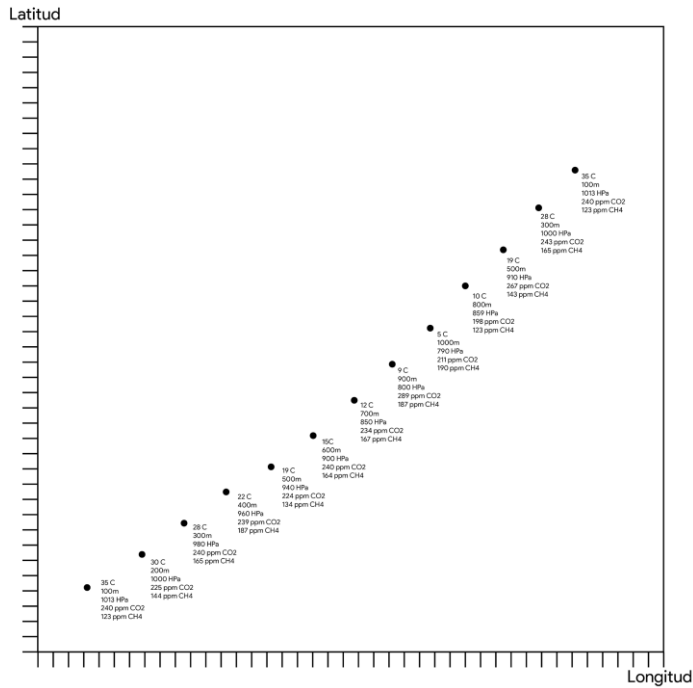
Para la estación de tierra estamos desarrollando un programa utilizando Processing, sin embargo, todavía no está completo. La intención es que muestre las gráficas de los datos y el mapeo de los mismos en tiempo real.

## SISTEMAS IMPLICADOS EN LA ESTACIÓN DE TIERRA

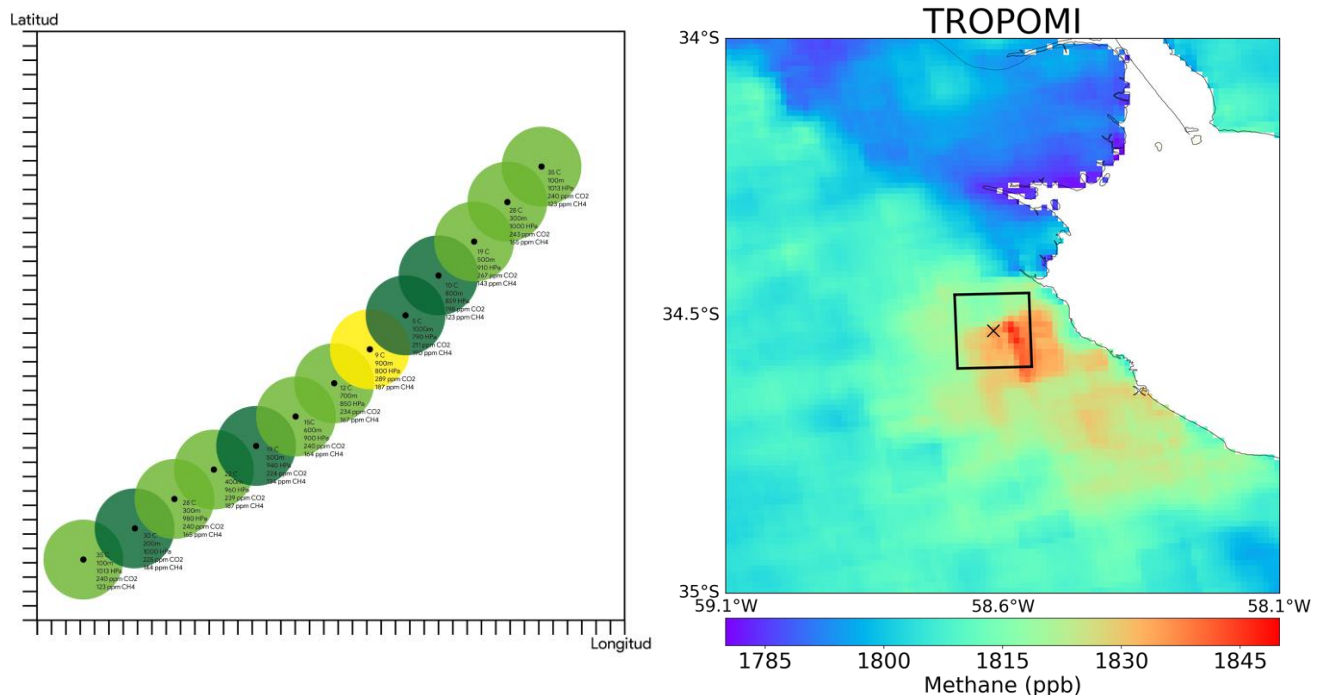
### Explicación

La estación de tierra es la encargada de recibir y procesar los datos enviados por el CanSat durante su misión. Los datos que recibirá serán los recogidos por los sensores (presión, altitud, temperatura, posición GPS, ppm CO2, ppm metano...). Estos datos se tratarán y graficarán en tiempo real, la intención es tener una gráfica con cada dato por separado y, además, utilizar la posición GPS para poner un plano en el que se mapearán los datos, según la posición en la que sean recogidos. A continuación, se muestra un ejemplo de lo que queremos

conseguir de mapeo (los datos son aleatorios y no representan los valores que esperamos obtener):



También nos gustaría resaltar con colores según los datos obtenidos, para que se vean más fácilmente y para simular, salvando las distancias, como se verían si se usasen sensores de espectrometría de imagen de alta resolución. Se muestran un ejemplo de lo que queremos conseguir y una imagen captada con un espectrómetro aéreo de imagen de alta resolución:



## Componentes

Los componentes utilizados para la recepción de la señal enviada por el CanSat serán:

1. Módulo de radio: APC220. Será el encargado de recibir la señal enviada por el otro módulo APC220 a bordo del CanSat. La señal se enviará en una frecuencia que oscilará entre los 433 y los 435 MHz.
2. Portátil: tendrá conectado el módulo de radio y cargado el programa desarrollado con Processing, donde se procesarán, graficarán y mapearán los datos recibidos.

## SISTEMAS DE ATERRIZAJE. PARACAÍDAS U OTROS SISTEMAS. PLANTEAMIENTO DE LA IDEA.

Queremos que cuando empiece el descenso desde los aproximadamente 1000 metros se despliegue el paracaídas y queremos que el tiempo de vuelo máximo sea de 120 segundos. Queremos que el paracaídas tenga una forma de semiesfera para conseguir que el CanSat baje a una velocidad entre 8-11 m/s. El paracaídas se almacenará desde un principio encima del CANSAT y si hay algún inconveniente se pondrá en el lugar más adecuado.

### Diseño

Hemos decidido que el paracaídas tenga forma de semiesfera presentada de la siguiente manera como se muestra en la imagen:



La idea era crear un cálculo basado en el radio, peso y velocidad de caída del CanSat formado por gores formulados a continuación:

### Cálculos

Para los cálculos si igualamos la fuerza de resistencia del paracaídas ( $F_r$ ) a el peso que mueve el paracaídas ( $F_g = m \cdot g$ ) hacia el suelo. Cuando estén igualadas deja de acelerar y la velocidad se convierte en constante, esa es la velocidad de caída.

$$\Sigma F = m \cdot a, F_r - F_g = m \cdot a, \text{ como } F_r = F_g, 0 = m \cdot a; a = 0$$

$$F_g = m \cdot g \text{ (peso CanSat)}$$



$F_r = \frac{1}{2} \cdot r \cdot C_d \cdot A \cdot V^2$  (fuerza de resistencia del paracaídas).

- **r** = densidad del aire ( $1,22 \text{ Kg/m}^3$ ).
- **Cd** = coeficiente de resistencia aerodinámico en paracaídas de forma semiesférico, su valor es 0,62.
- **A** = superficie del paracaídas (en nuestro caso semiesfera).
- **V** = velocidad de descenso.

1. Igualando las dos expresiones:  $m \cdot g = \frac{1}{2} \cdot r \cdot C_d \cdot A \cdot V^2$

De estas dos expresiones puede despejar el área del paracaídas,

$$A = 2 \cdot (m \cdot g) / r \cdot C_d \cdot V^2$$

2. Si la superficie del paracaídas es una semiesfera:  $A = 2 \cdot \pi \cdot R^2$

Por tanto, igualando:

$$2 \cdot \pi \cdot R^2 = 2 \cdot (m \cdot g) / r \cdot C_d \cdot V^2$$

3. Despejando R (radio) podemos calcular el diámetro de la superficie del paracaídas:

$$R = \sqrt{\frac{m \cdot g}{r \cdot C_d \cdot V^2 \cdot \pi}}$$

Para determinar la velocidad de caída y si la velocidad es constante, tenemos que:

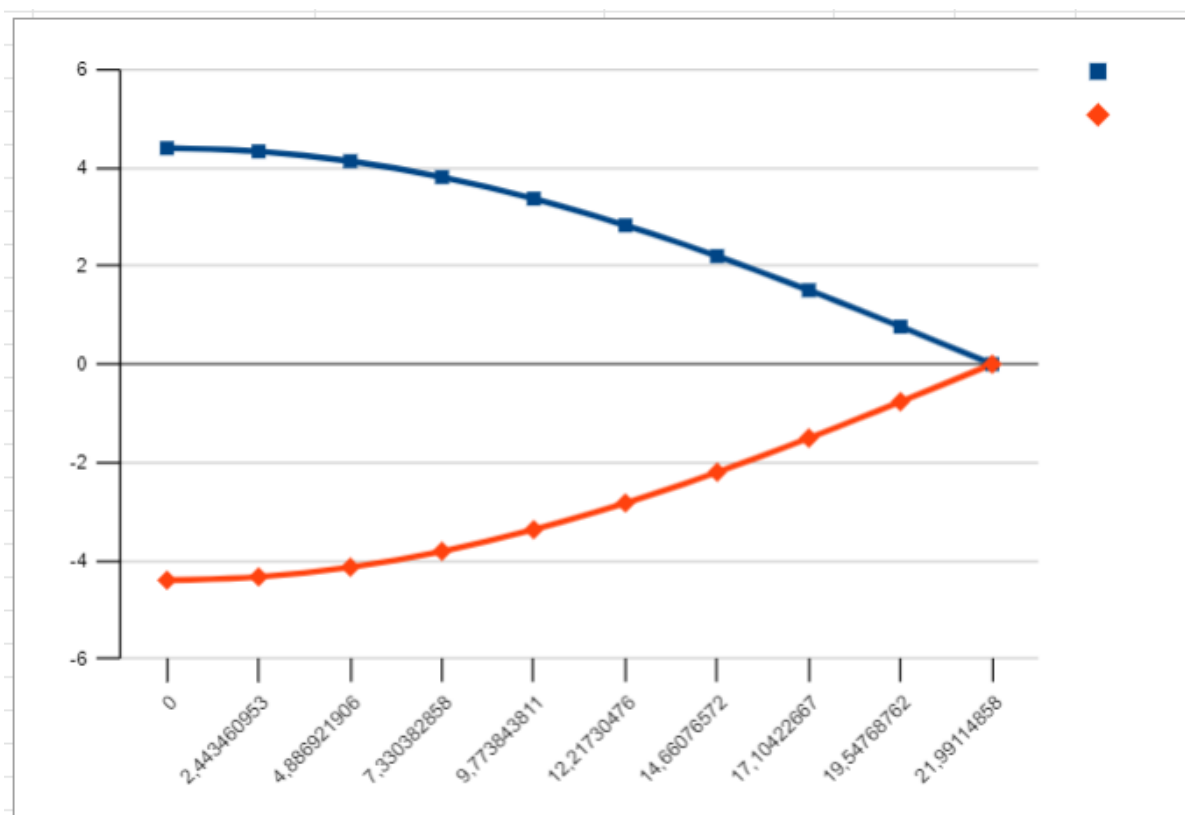
$$V = e / t.$$

El cohete alcanza entre 1000 metros de altura y el tiempo de descenso queremos que sea de 120 segundos entonces, ya tenemos la V.

Además, tenemos la masa de nuestro CanSat (de 300g a 350g), con lo cual ya tenemos todo para calcular nuestro radio o diámetro de paracaídas.

Estas son unas imágenes de cómo han quedado los cálculos para los gores del paracaídas (Se puede acceder al desarrollo completo aquí: <https://github.com/WBCanSat/CanSatProject/tree/main/CDR> ):

Radio (cm)	Ángulos (radio) ( $\alpha$ )	Ángulos	x	y	y <sup>-</sup>	Base
14	0	0	0	4,398229715	-4,398229715	8,79645
Número de gores	0,1745329252	10	2,443460953	4,331410723	-4,331410723	
10	0,3490658504	20	4,886921906	4,132984008	-4,132984008	
	0,5235987756	30	7,330382858	3,808978665	-3,808978665	
	0,6981317008	40	9,773843811	3,369239433	-3,369239433	
	0,872664626	50	12,21730476	2,827127565	-2,827127565	
	1,047197551	60	14,66076572	2,199114858	-2,199114858	
	1,221730476	70	17,10422667	1,504283158	-1,504283158	
	1,396263402	80	19,54768762	0,763744575	-0,763744575	
	1,570796327	90	21,99114858	0	0	



## Material

Para la fabricación del paracaídas hemos elegido basarnos en un material doméstico, es decir, un material que tuviera los requisitos mencionados posteriormente que se pudiera encontrar por casa. En este caso elegimos tras mucha búsqueda, una bolsa que estaba fabricada de una tela que era perfecta para la fabricación de nuestro paracaídas, incluyendo la cantidad necesaria para tener todos los gores.

La idea principal sobre la tela es que fuera solo de entre 2,5 y 3 milímetros de grosor para que en el descenso del CanSat sea de entre 8-11 m/s. Después se conectarán los extremos del paracaídas con tanza que es un hilo fuerte, fino y ligero. Tras mucha búsqueda decidimos utilizar este material y esta tanza:



La bolsa que se muestra en la imagen estaba compuesta del material que necesitábamos para el paracaídas, está compuesto de nylon. El único inconveniente que hemos podido tener es que, al ser de un color tan oscuro, era un poco más complicado patronar los gores. Encima de la bolsa se muestra la tanza que hemos utilizado para el paracaídas.




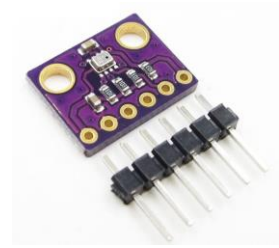
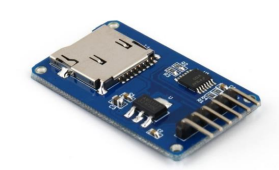

Una idea dónde se podría almacenar el paracaídas es de colocar una pequeña anilla de metal en la zona superior del CanSat, donde se unirán las tanzas que estarán conectadas al paracaídas.

## **PRESUPUESTO DETALLADO. DIFERENCIAS CON EL INICIAL**



Al principio del proyecto no sabíamos con seguridad cómo sería el precio final tras obtener todos los sensores y el equipamiento necesario, asumimos que íbamos a gastar menos de 200€ en total en el proyecto. Tras hacer mucha búsqueda y hacer un recuento de los precios de cada producto, hemos reducido el precio final a 95.06€ que es más barato que lo que en un principio asumimos. Estas diferencias se deben a que hicimos esta suposición al principio del proyecto, solo teniendo en cuenta que el presupuesto máximo era de 500€ y que no teníamos aún información sobre los sensores que acabaríamos utilizando.

### **Lista Presupuesto**

Aquí está la lista de los sensores y otros materiales con sus respectivos precios:

Material	Imagen Material	Cantidad	Precio unitario	Precio total
MQ-135		1	2.95€	2.95€
MQ-4(Sensor metano)	MQ-4 	1	2.53€	2.53€
GY-NEO6MV2 NEO-6M 3V-5V (GPS)		1	9.99€	9.99€
BMP280 (Presión y temperatura)		1	4.59€	4.59€
2 lectores Micro-SD Arduino		1	6.99€	6.99€
4022 6F22 6LR61 (pila 9V)		1	1.69€	1.69€

Tela paracaídas		1	Metro de Nylon:2€	2€
Cordajes (Tanza)		1	2€	2€
Zumbador Pasivo 5V		2	0.59€	1.19€
Conector/Adaptador para pila 9V 6LF22		1	0.99€	0.99€
Kit Arduino compatible UNO Mini Básico		1	16.36€	16.36€
Rollo de filamento ICEFIL1ABS085		1	16.94€	16.94€
Kit de soldadura		1	11.88€	11.88€

Interruptor 2P2T-50V-05A		1	0.49€	0.49€
Radio apc 220		1	17.44€	17.44€
			Total:	95.06

## FINANCIACIÓN Y APOYO TÉCNICO

### Financiación

Para la financiación hemos optado seguir dos estrategias distintas de forma simultánea:

1. Patrocinio de una empresa local llamada TechMakers (cuya página web es: <https://techmakers.es>). Contactamos con ella, le explicamos el proyecto y nos facilitó varios sensores y componentes (Lector SD, GPS, módulo de radio y sus impresoras 3D para imprimir la carcasa de nuestro CanSat); por lo que estamos eternamente agradecidos por su colaboración.
2. Financiación organizando actividades culturales, ya que vamos a organizar un torneo de videojuegos online que se celebrará entre el 26 y 27 de marzo. Con él planeamos recaudar beneficios para pagar los gastos que suponen la creación del CanSat. Para ello contactamos con nuestro centro que ha aportado 90€ junto con el AMPA (45 euros el centro y 45 euros el AMPA) para los premios del torneo, no hemos conseguido contactar con ninguna empresa que quisiera facilitarnos algún producto para los premios, aunque lo hayamos intentado. Sin embargo, con esto creemos que será suficiente para atraer a jóvenes al torneo. Para la organización estamos contando con la ayuda de Diego Iglesias, compañero de nuestro centro. Los ingresos recaudados se describirán más adelante, puesto que ahora estamos en una fase inicial y todavía no hemos abierto las inscripciones a este. En la siguiente página le mostramos una versión de los carteles que estamos creando para promocionar el torneo, no es la versión definitiva, pero se puede apreciar el camino artístico que se está tomando:



**90€ EN PREMIOS**

Juega con tus amigos en

**EQUIPOS DE 3**

**26-27MARZO**

📷 📺 @wbcansat



## **Apoyo técnico**

Para la creación del CanSat hemos contado y contamos con varias personas que nos están apoyando:

- Manuel Trigueros, profesor del IES Wenceslao Benítez, que nos está ayudando con las dudas que tenemos relacionadas con la electrónica y la programación.
- María del Mar Díaz, familiar, que ha accedido a coser el paracaídas, pues tiene bastante experiencia en la costura y hará un mejor trabajo que nosotros (aunque los gores están ya diseñados, el paracaídas todavía no está hecho).
- Diego Iglesias, estudiante de 1º de Bachillerato y compañero, que nos está ayudando con la organización del torneo y la financiación del proyecto.
- Candelaria Palacios, nuestra mentora, que nos ayudó a contactar con TechMakers, a organizarnos, a buscar ideas, a redactar este informe y a un sinfín de cosas más.

## **PROYECTO CIENTÍFICO. DESCRIPCIÓN DEL PLAN DE ACTUACIÓN DEL DÍA DEL LANZAMIENTO. PLANTEAR SOLUCIONES ANTE POSIBLES IMPREVISTOS.**

### **Pruebas**

#### **Funcionamiento del CanSat**

Si al hacer pruebas el CanSat funciona, entonces estará preparado para su lanzamiento en el cohete, pero si al hacer pruebas vemos que el CanSat no funciona, comprobaremos todo lo que constituye la construcción del CanSat, su diseño su programación y construcción en busca de errores que hagan que nuestro satélite no funcione. Tras arreglar cualquier posible fallo que tenga que ver con algún sensor o parte del diseño que no funcione realizaremos otras pruebas de nuevo para poder perfeccionar nuestro CanSat.

#### **Realización de la misión principal**

Cuando realicemos las pruebas con nuestro satélite, queremos comprobar con prioridad si hemos sido capaces de llevar a cabo la misión principal, que es comprobar si nuestro CanSat puede medir la presión y la temperatura. No es solo comprobar que sus sensores funcionen correctamente, también tenemos que comprobar si los datos recogidos llegan a la estación de tierra donde los observadores estarán posicionados. Por ello, si ocurre algún error o imprevisto en las pruebas, tendremos que comprobar si este error viene del propio CanSat, que sus sensores no funcionen correctamente, por ejemplo, o si este error tiene que ver con que los datos no llegan a la estación de tierra o son datos que no figuran con lo que normalmente se debería obtener tras las pruebas de lanzamiento. Para



poder prevenir estos errores garantizamos que todo esté bien colocado y que funcione antes de hacer ninguna prueba en conjunto con el lanzamiento.

### **Comunicación por radio**

Una de las cosas más importantes que debemos comprobar es si funciona la comunicación por radio, esto será muy importante para poder comprobar si ha habido algún tipo de imprevisto en las pruebas de lanzamiento, por ello queremos prevenir que la comunicación funcione a la perfección haciendo pruebas, aunque no sean con el propio dispositivo CanSat para poder garantizar que nuestra comunicación por radio estará preparada para las pruebas con nuestro satélite.

### **Realización de la misión secundaria**

Una vez realicemos pruebas, nos gustaría comprobar que nuestra misión secundaria funcione correctamente porque nos indicaría que nuestra idea fue bien planteada y realizada incluso tras los cambios que realizamos a lo largo del proyecto, por ello es muy importante para nosotros que esta misión funcione. Pero si llegamos a tener errores en las pruebas, primero comprobaremos que este error proceda de un sensor que no funciona o que los datos no se recogen correctamente, pero si este error llega hasta un punto en el que tengamos que replantear nuestra misión secundaria, aunque sería algo desafortunado, todos los miembros del grupo trabajaremos para buscar una nueva solución con la que poder trabajar y llevar a cabo pruebas que podamos realizar. Por suerte hemos tenido la oportunidad de comprobar uno de nuestros sensores, el de aire, y hemos podido probar que funciona correctamente.

### **Pruebas del paracaídas**

El paracaídas fue creado en referencia a una semiesfera, queremos ver si funciona tanto su despliegue como si su forma es adecuada para amortiguar el peso del CanSat, pero estas pruebas serán realizadas previamente en alturas muy bajas para prevenir dañar nuestro Satélite y así si descubrimos algún error tras hacer pruebas, como que no aguanta el peso o el despliegue no funciona, podremos arreglarlo y volver a hacer otras pruebas antes de comprobar alturas más mayores, que cuando lleguemos a ellas habremos garantizado que el paracaídas funcione.

### **ESTADO DE LA DIFUSIÓN Y DIVULGACIÓN DEL PROYECTO. FOTOS, VÍDEOS, ENLACES DEL PROGRESO.**

Para llevar a cabo la divulgación del proyecto, nos hemos apoyado principalmente en el uso de las redes sociales. La información ha estado en continua

retroalimentación aportando diariamente nuestro progreso y toda aquella documentación que creíamos necesaria que se conociese. Twitter, Instagram y Facebook han sido nuestras herramientas de comunicación con el público. También, hemos creado un GitHub registrando todo el progreso en cuanto a la programación y otros documentos, para cualquier persona que desee consultarlo. A continuación, se adjunta el enlace para poder acceder a nuestras redes sociales y sitios webs:



<https://github.com/WBCanSat/CanSatProject>



**wbcansat** <sup>2</sup>



4 DE 5 PASOS COMPLETADOS ▾



4  
Publicaciones

24  
Seguidores

4  
Siguiendo

**WBCansat**

Ciencia, tecnología e ingeniería

Twitter: @WBCansat

[github.com/WBCanSat/CanSatCode](https://github.com/WBCanSat/CanSatCode)

Usuario: wbcansat

<https://instagram.com/wbcansat?igshid=1b64ue1rplmxx>



Usuario: WBCanSat

<https://twitter.com/wbcansat?s=21>

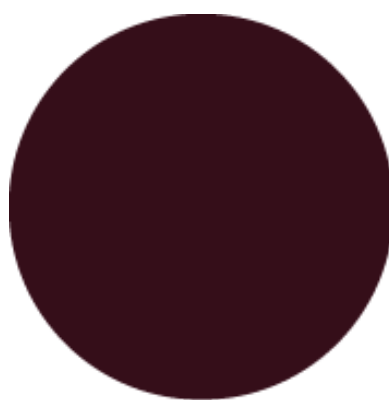


Usuario: WB CanSat

<https://www.facebook.com/wb.cansat>

Hemos creado un logo para nuestro equipo personalizado, el cual ha sido compartido por redes sociales para una mejor identificación del equipo. Este logo se ha creado a partir de las iniciales "WB" de nuestro centro Wenceslao Benítez, que han sido modificadas para tener una apariencia más moderna, innovadora y conectada con el mundo de las tecnologías. En el interior de la "B" se observa un satélite despegando, que simboliza el propósito de nuestro proyecto, que es lanzar

un satélite. El color elegido como principal es el Pantone 4146 C, un azul muy oscuro que refleja el espacio, además se asocia con la seriedad, la estabilidad, la verdad y la moderación. El secundario que se utilizará en algunos casos es un color burdeos similar al Pantone 4102C, aunque un poco más oscuro, que evoca la pasión y la devoción.



Por otro lado, estamos realizando una campaña de difusión en el instituto, a través de la cual presentaremos el proyecto a otros alumnos con el fin de motivar a los más jóvenes para que en un futuro se animen a participar en este tipo de concursos.

### **Difusión con el ayuntamiento**

Dado la importancia de divulgar los conocimientos científicos y tecnológicos hemos decidido recurrir a entidades que proporcionen una mejor visibilidad al proyecto, y que al mismo tiempo nos ayuden a acercarnos a un mayor número de usuarios.

Nuestro excelentísimo Ayuntamiento de San Fernando lideraría, de este modo, la difusión del trabajo, ya no solo en nuestra ciudad sino a nivel internacional. A continuación, se adjunta un enlace al documento que hace referencia a la propuesta de difusión que solicitamos al Ayuntamiento: <https://github.com/WBCanSat/CanSatProject/blob/main/CDR/Townhall%20request.pdf>

## **AGRADECIMIENTOS**

Finalmente, queremos agradecer a TechMakers por aportar material a nuestro proyecto (Lector SD, GPS, módulo de radio y la posibilidad de imprimir en 3D las piezas del CanSat). A Manuel Trigueros, profesor de nuestro centro IES Wenceslao Benítez, que nos ha ayudado con algunos de los problemas que hemos y estamos encontrando por el camino; además de ser nuestra principal fuente para resolver dudas relacionadas con la electrónica. A Diego Iglesias, alumno del centro que ha aceptado colaborar en la campaña de financiación para el proyecto y que nos está ayudando a organizar un torneo para recaudar fondos, gracias al cual podremos pagar ciertos sensores y componentes del proyecto. A Candelaria Palacios, nuestra mentora, a la cual le debemos esta increíble oportunidad que se nos ha ofrecido y que nos ha estado apoyando en todo lo que ha podido, tanto mental como económicamente. Al ayuntamiento de San Fernando, nuestra ciudad, que ha cedido a publicitar nuestro proyecto y a incrementar la expectación alrededor de este. A María del Mar Díaz, familiar que nos va a ayudar en la costura del paracaídas, debido a su experiencia en este hábito. Al IES Wenceslao Benítez y al AMPA por financiar los premios para el torneo. Y, sobre todo, a todos aquellos que nos están siguiendo y apoyando en todo momento.

## **BIBLIOGRAFÍA: REFERENCIAS E INFORMACIÓN UTILIZADA**

Para obtener referencias sobre cómo hacer el proyecto CANSAT, nos basamos en buscar información sobre pasados proyectos, ver su distribución, las misiones, para poder tener una idea clara de lo que teníamos que hacer:

Esero CanSat (edición 2019) <http://esero.es/cansat/ediciones-anteriores/cansat-edicion-2019/>

Esero CanSat (edición 2020) <http://esero.es/cansat/ediciones-anteriores/cansat-edicion-2020/>

Datos de emisiones de metano de alta resolución para la gestión de residuos (17/9/2020)

[https://www.esa.int/Applications/Observing\\_the\\_Earth/Copernicus/Sentinel-5P/High-resolution\\_methane\\_emissions\\_data\\_for\\_waste\\_management](https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Sentinel-5P/High-resolution_methane_emissions_data_for_waste_management)

Datos de emisiones de metano de alta resolución (18/9/2020)

<https://www.tiempo.com/ram/datos-de-emisiones-de-metano-de-alta-resolucion.html>

Medidor y sensor de luz de fotosíntesis (Enero 1971)

<https://www.jstor.org/stable/1934743?seq=1>

Fluorescencia de clorofila: una guía práctica (1/4/2000)

<https://academic.oup.com/jxb/article/51/345/659/652534>

Planta de energía desde arriba (3/2/2015)

[http://www.esa.int/Applications/Observing\\_the\\_Earth/Plant\\_power\\_from\\_above](http://www.esa.int/Applications/Observing_the_Earth/Plant_power_from_above)

Cartas de investigación geofísica (20/2/2015)

<https://agupubs.onlinelibrary.wiley.com/doi/full/10.1002/2014GL062943>

Un rojo simple: sensor de rojo lejano que utiliza detectores de fosforo de arseniuro de galio (Agosto 1996)

<https://www.jstor.org/stable/2389947?seq=1>

Esero CanSat (información CanSat)

<http://esero.es/cansat/>

Medir temperatura y presión barométrica con Arduino y BMP280 (31/3/2020)

<https://www.luisllamas.es/medir-temperatura-y-presion-barometrica-con-arduino-y-bmp280/>

Módulo GPS con Arduino (28/8/2017)

<https://www.geekfactory.mx/tutoriales/tutoriales-arduino/shield-o-modulo-gps-con-arduino/>