

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

1. INTRODUCCIÓN/ ANTECEDENTES.....	3
1.1. Introducción	3
1.2. El equipo	3
1.4. Organización del trabajo.....	3
2. PROYECTO CANSAT.....	5
2.1. Misiones	5
2.1.2. Misión secundaria	5
2.2. Resultados esperados	6
3. DISEÑO MECÁNICO DEL CANSAT Y DE LA ESTACIÓN DE TIERRA.....	6
3.1. Estructura Externa	7
.....	8
.....	8
3.2. Estructura interna	8
3.3. Componentes	9
4. DISEÑO ELÉCTRICO	9
4.1. Componentes eléctricos	11
5. PROGRAMACIÓN. LENGUAJE DE PROGRAMACIÓN Y DIAGRAMA DE FLUJO DEL PROGRAMA. BREVE EXPLICACIÓN DEL FUNCIONAMIENTO ...	11
5.1. Diagrama de flujo	12
5.2. Programación	13
5.3. CanSat.....	13
5.4. Estación de tierra.....	17
6. SISTEMAS IMPLICADOS EN LA ESTACIÓN DE TIERRA.....	17
6.1. Explicación	17
6.2. Componentes	18
7. SISTEMAS DE ATERRIZAJE. PARACAÍDAS U OTROS SISTEMAS. PLANTEAMIENTO DE LA IDEA.	19
7.1. Diseño	19
7.2. Cálculos	19
7.3. Material	20
8. PRESUPUESTO DETALLADO. DIFERENCIAS CON EL INICIAL.....	20
8.1. Lista Presupuesto	21
9. FINANCIACIÓN Y APOYO TÉCNICO.....	22

9.1. Financiación.....	22
9.2. Apoyo técnico.....	23
10. PROYECTO CIENTÍFICO. DESCRIPCIÓN DEL PLAN DE ACTUACIÓN DEL DÍA DEL LANZAMIENTO. PLANTEAR SOLUCIONES ANTE POSIBLES IMPREVISTOS.....	23
10.1. SOLUCIONES A POSIBLES PROBLEMAS	23
10.1.1. Funcionamiento del CanSat.....	23
10.1.2. Realización de la misión principal.....	24
10.1.3. Comunicación por radio	24
10.1.4. Realización de la misión secundaria	24
10.1.5. Pruebas del paracaídas.....	24
11. ESTADO DE LA DIFUSIÓN Y DIVULGACIÓN DEL PROYECTO. FOTOS, VÍDEOS, ENLACES DEL PROGRESO.....	24
12. AGRADECIMIENTOS.....	26

1. INTRODUCCIÓN/ ANTECEDENTES

1.1. Introducción

Este año, cursando 1º de bachillerato, nos embarcamos en este proyecto reconociendo que desafortunadamente no tendremos oportunidad de participar en los próximos debido a la complejidad de cursos superiores y el escaso tiempo para invertir en estos. Por eso elegimos CanSat, un proyecto a gran escala con el fin de desafiar nuestro conocimiento.

1.2. El equipo

El equipo se conforma de cinco miembros: Julio Quintero, Pablo Pacheco, Mateo Moreno, Rubén Vargas y Diego Iglesias.

1.3. Reparto de tareas

Las tareas fueron asignadas teniendo en cuenta los conocimientos y capacidades de cada integrante, para lograr la máxima eficiencia.

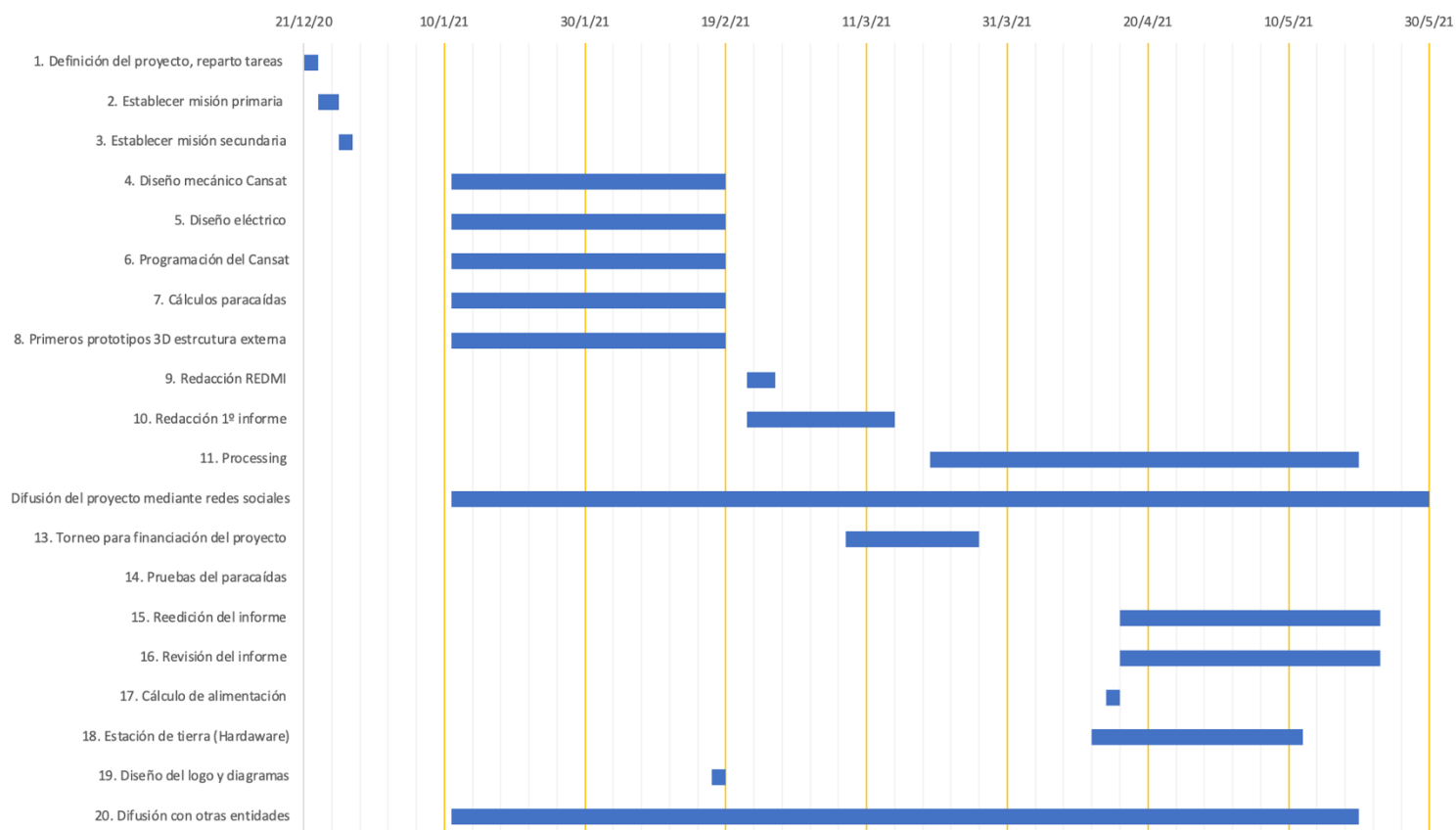
1.4. Organización del trabajo

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

- ☐ Hemos creado un grupo de Telegram con nuestra mentora para mantenerla informada del avance proyecto.
- ☐ Realizamos reuniones desde en Discord, plataforma en la que compartimos nuestras ideas y resolvemos dudas.
- ☐ Para la elaboración del informe CDR, trabajamos en Google Docs, que nos permite editar el texto de forma simultanea
- ☐ Por último, hemos creado un WBS para organizar los trabajos que tenía que desarrollar cada uno. Además, es una buena forma para no olvidarnos de nada y tener todo ordenado. A continuación, se muestra el diagrama que ilustra las actividades fundamentales del proyecto:

Nombre de la actividad	Fecha inicio	Duración en días	Fecha fin	Integrantes
1. Definición del proyecto, reparto tareas	21/12/20	2	23/12/20	Julio, Pablo, Rubén, Mateo, Diego
2. Establecer misión primaria	23/12/20	3	26/12/20	Julio, Pablo, Rubén, Mateo, Diego
3. Establecer misión secundaria	26/12/20	2	28/12/20	Julio, Pablo, Rubén, Mateo, Diego
4. Diseño mecánico Cansat	11/1/21	39	19/2/21	Mateo
5. Diseño eléctrico	11/1/21	39	19/2/21	Rubén
6. Programación del Cansat	11/1/21	39	19/2/21	Julio
7. Cálculos paracaídas	11/1/21	39	19/2/21	Pablo
8. Primeros prototipos 3D estructura externa	11/1/21	39	19/2/21	Mateo
9. Redacción REDMI	22/2/21	4	26/2/21	Pablo
10. Redacción 1º informe	22/2/21	21	15/3/21	Julio, Pablo, Rubén, Mateo, Diego
11. Processing	20/3/21	61	20/5/21	Diego
12. Difusión del proyecto mediante redes sociales	11/1/21	139	30/5/21	Rubén
13. Torneo para financiación del proyecto	8/3/21	19	27/3/21	Diego, Julio
14. Pruebas del paracaídas	17/5/21	0	17/5/21	Pablo, Rubén
15. Reedición del informe	16/4/21	37	23/5/21	Julio, Pablo, Rubén, Mateo, Diego
16. Revisión del informe	16/4/21	37	23/5/21	Rubén
17. Cálculo de alimentación	14/4/21	2	16/4/21	Pablo
18. Estación de tierra (Hardware)	12/4/21	30	12/5/21	Pablo, Julio, Mateo
19. Diseño del logo y diagramas	17/2/21	2	19/2/21	Julio
20. Difusión con otras entidades	11/1/21	129	20/5/21	Rubén
		Inicio proyecto	21/12/20	
		Fin proyecto	20/5/21	

DIAGRAMA GANTT



2. 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.

2.1. Misiones

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

2.1.1. 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.

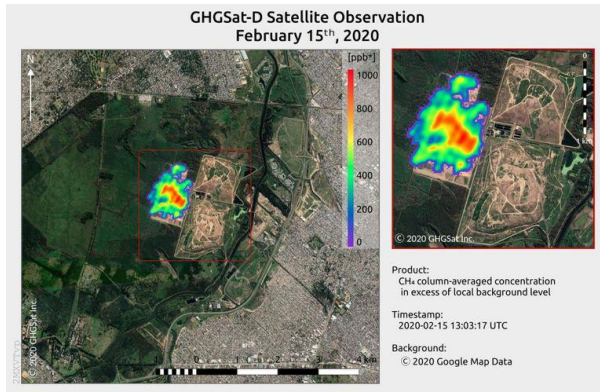
Hemos elegido la placa Arduino UNO como principal componente de hardware, como lenguaje de programación Arduino (c/c++) y como entorno de desarrollo vscode.

2.1.2. Misión secundaria

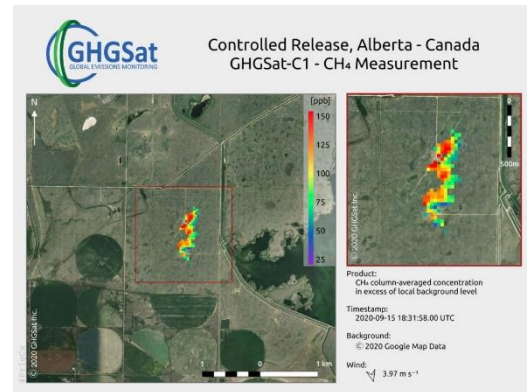
En un principio, la idea era examinar la vegetación de la zona desde el satélite. Para ello, mediríamos la eficiencia de la fotosíntesis de las plantas, basándonos en la fluorescencia de la clorofila que éstas emiten, con ello podríamos determinar la salud de las zonas verdes, bosques, reservas naturales... Sin embargo, tras una exhaustiva investigación sobre este estudio, rechazamos la idea debido a la complejidad para encontrar sensores que se adaptaran al presupuesto máximo impuesto en las bases del concurso.

Así pues, aunque optamos por utilizar sensores MQ, el objetivo seguiría estando centrado en el medioambiente: 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. Todo ello para poder reducirlas cuando sean detectadas. Como se graficarán y representarán los datos se ampliará en el epígrafe 6.

Esta idea puede extrapolarse de distintas formas. Primeramente, aunque usemos sensores MQ que tienen baja resolución, se podrían utilizar espectrómetros de imagen de alta resolución (high-resolution imaging spectrometer) para poder observar estas variaciones desde la órbita terrestre. Un ejemplo a esto sería el satélite Iris de GHGSat, que puede observar altas concentraciones de metano.

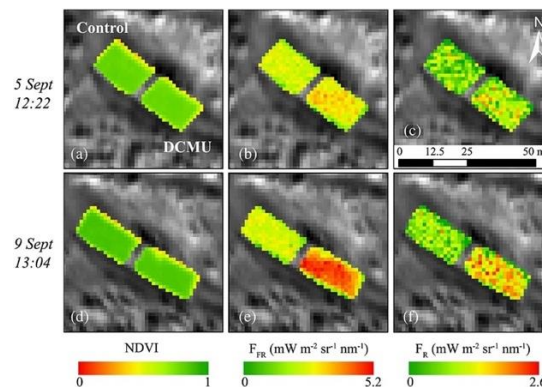


(Anon., 2021)



(Bishton, 2020)

A su vez, se pueden utilizar sensores que midan la concentración de la fluorescencia de la clorofila, en vez de aquellos que miden gases de efecto invernadero. Esta extrapolación estaría conectada con nuestra primera idea.



(Anon., 2015)

Por último, suponiendo que el Cansat fuese un satélite espacial, se podría utilizar para investigar las atmósferas de otros planetas y comprobar la presencia de gases que propicien la vida, tanto extraterrestre como para posibles colonias humanas.

2.2. Resultados esperados

Con nuestra misión esperamos obtener durante el descenso los valores de concentración de CH₄ (metano) y CO₂ (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.

3. 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.

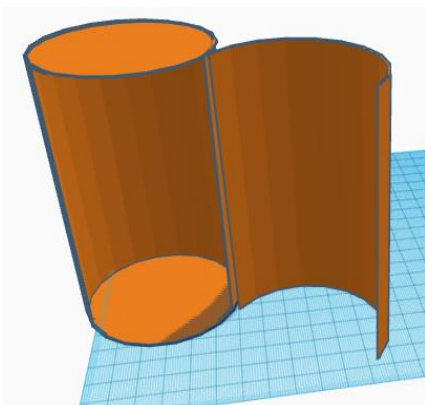
3.1. Estructura Externa

Para empezar con la estructura externa, nos dirigimos a un programa que con el que nos encontramos 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 un prototipo que se encajase mediante deslizamiento hacia arriba, pero lo descartamos porque pensábamos que podía ocasionar problemas con los cables. Ese primer prototipo solo fue una idea y que no se trasladó al ordenador a diferencia de los otros. Después, se nos ocurrió fabricar uno con bisagras, pero lo vimos muy difícil para ponerlas en la carcasa de plástico. Aun así, no se descartó y se pasó al programa de Tinkercad, junto a dos prototipos más.

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 se pasaron a Tinkercad. Para el diseño, tardamos unas 5 semanas, debido a que hicimos un par de pruebas para ver la resistencia de la estructura externa.

Tras las pruebas decidimos poner las paredes laterales de 2 mm y las inferiores de 3mm en las bases, porque fue el que más resistencia tenía para que no se rompiese



Prototipo 1(TINKERCAD)

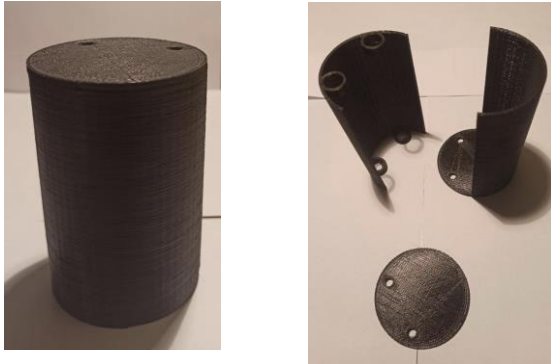


Prototipo 2(freecad)



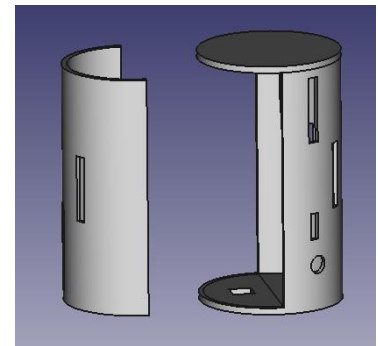
Prototipo 3(freecad)

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

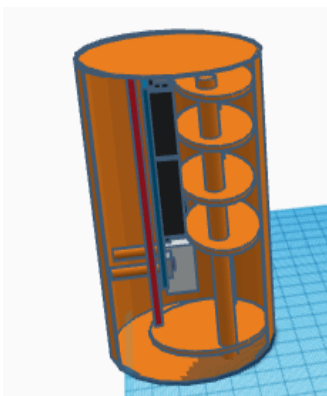
Para finalizar nuestra estructura, modificamos el tercer prototipo realizando algunos orificios de ventilación para que los sensores tomen la muestra de gases en la atmósfera continuamente, y de este modo, comprobar la variación de los mismos. En ellos se inserta el led, el interruptor y el sensor del GPS



Prototipo final







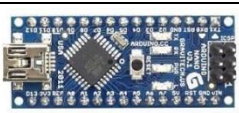
3.2. Estructura interna

Para realizar la estructura interna, esbozamos un esquema en papel y después pasamos todo a un programa de edición 3D.



Para crear la estructura, nos focalizamos en dos factores principales. Primero, que fuera fácil de manipular. Y segundo en poner cada uno de los componentes sin ningún problema y 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 crear un sitio específico para ubicar cada sensor. De este modo, en caso de que existiese algún problema con un sensor no sería necesario desconectar todos los cables

3.3. Componentes

	Nombre	Masa	Dimensiones	Voltaje	Amperaje	Sensibilidad
	BMP280	1.07g	1.9x1cm	4V	20mA	Presión: +-1Hpa Temperatura: +-0.5°C
	MQ-4	6.8g	3.2x2cm	3.5V-5V	150mA	
	GPS-NEO6	16.8g	3.5x2.5 cm	2.7V-3.6V	37mA	+ -2.5m en velocidad 0.1m/s
	Zumbador	1.2	3cm r	3.5V-5.5V	25mA	+ -500Hz
	MQ-135	6.8	3.2x2cm	4V	150mA	
	Lector Sd	4.38	4.2x2.3cm	3.3V-5V	15 mA	X
	Radio apc 220	11.4	3.7x1.8cm	3.5V-5V	30mA	X
	Arduino nano	5.89	4.3x1,7cm	5V	20mA	X

4. DISEÑO ELÉCTRICO



4.1. Componentes eléctricos

Hemos utilizado la placa Arduino Nano uno para conectar los componentes que se encuentran ilustrados en el diagrama superior debido al bajo costo, su menor dimensión y peso respecto a otros modelos, la facilidad para encontrar documentación y soporte y la amplia variedad de dispositivos con los que se puede conectar.

4.2. Placas PCB CanSat y Estación de Tierra

○Sistema de alimentación del CanSat

Al medir el voltaje y amperaje de nuestro CanSat con un polímetro, obtuvimos un amperaje de 0,26 A y voltaje de 6,41 V de media. Por lo que la potencia es:

Por lo que, para alimentarlo hemos decidir usar una pila de 9 V y 550 mAh, con una energía almacenada de 4,95 WH. Ya que teniendo en cuenta estos valores, podemos calcular la autonomía del CanSat:

Tiempo suficiente para hacer pruebas antes del lanzamiento y para este mismo. A su vez, al tener 9 V puede ser conectada al Arduino Nano como se ve en el esquema, sin necesidad de regular el voltaje. Y son fáciles de encontrar y su coste es muy reducido.

○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. La antena utilizada es una antena omnidireccional funcionando a una frecuencia de 433 MHz.

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

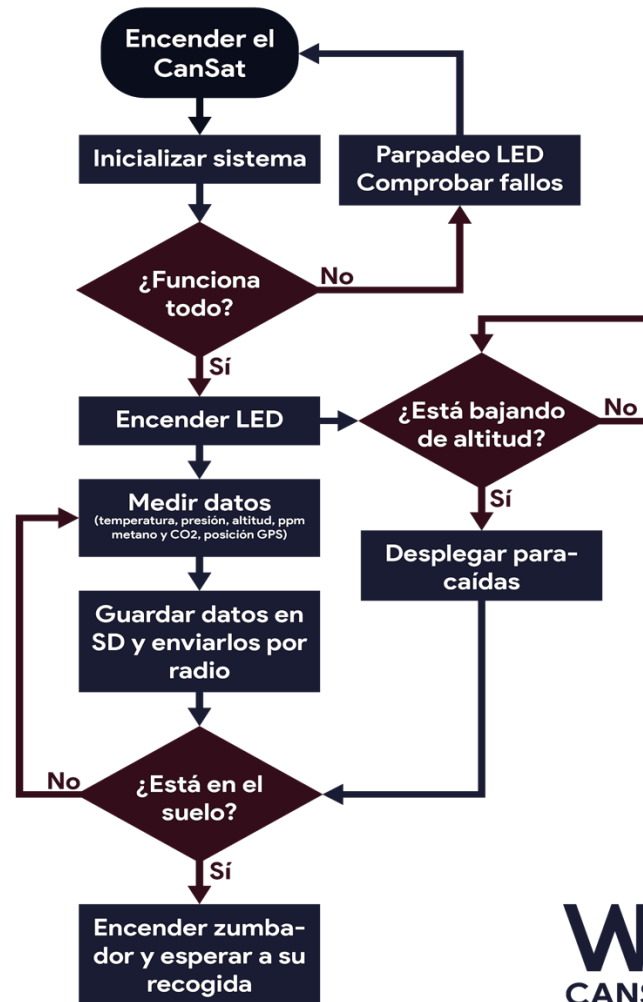
El lenguaje de programación que hemos utilizado para nuestro proyecto es el propio lenguaje de Arduino, que consiste en una mezcla entre C y C++; ya que como se ha mencionado anteriormente, hemos utilizado una placa Arduino UNO R3 como placa. Sin embargo, el entorno de desarrollo ha sido Visual Studio 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.

5.1. Diagrama de flujo

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

El código para saber si está bajando de altitud o si está en el suelo sería este:



```

if (!reach_ground){

    //Read sensors and save data
}

if (altitude_val - pre_altitude_val > 3 && !top_alt && altitude_val > 200){

    top_alt = true
    //Deploy parachute and save maximum altitude
}

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.

5.2. 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.

5.3. CanSat

5.3.1. Misión primaria

```
1  #include <Adafruit_BMP280.h>
2  #include <SPI.h>
3  #include <SD.h>
4  #include <SoftwareSerial.h>
5  #include <TinyGPS.h>
6  #include <MQ135.h>
7
8  #define CS_pin 9
9  #define LED_pin 7
10 #define buzz_pin 2
11 #define RX_pin 5
12 #define TX_pin 6
13 #define MQ4_pin A0
14 #define MQ135_pin A1
15
16 float seconds, altitude_val, pre_altitude_val, temperature_val, pressure_val, latitude_val, longitude_val, co2_val;
17 float MQ4_volt, MQ4_RSgas, MQ4_ratio, MQ4_value, flat, flon;
18 float MQ4_R0 = 11.820;
19 float slope = -0.318;
20 float y_intercept = 1.133;
21 double methane_val, methane_log;
22 char gps_data;
23 boolean top_alt, reach_ground, new_data;
24 unsigned long age;
25
26 File myFile;
27 File topFile;
28 Adafruit_BMP280 bmp;
29 TinyGPS gps;
30 SoftwareSerial softSerial(RX_pin, TX_pin);
31 MQ135 gasSensor = MQ135(A1);
32
33
```

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.

```
35 void setup(){
36     //Initialize sensors and board
37
38     Serial.begin(9600);
39     softSerial.begin(9600);
40     pinMode(LED_pin, OUTPUT);
41     pinMode(CS_pin, OUTPUT);
42     pinMode(buzz_pin, OUTPUT);
43     pinMode(MQ135_pin, INPUT);
44     pinMode(MQ4_pin, INPUT);
45
46     //Check if it is working
47
48     if (bmp.begin() && SD.begin(CS_pin) && softSerial.available()) {
49         digitalWrite(LED_pin, HIGH);
50     }
51     else {
52
53         if(!bmp.begin()){
54             Serial.print("bmp");
55         }
56         else if(!SD.begin()){
57             Serial.print("SD");
58         }
59         else if(!softSerial.available()){
60             Serial.print("GPS");
61         }
62         else{
63             Serial.print("Pégate un tiro");
64         }
65     }
66
67     while(1) {
68
69         digitalWrite(LED_pin, HIGH);
70         delay(500);
71         digitalWrite(LED_pin, LOW);
72         delay(1000);
73     }
74 }
75
```

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.

```
75
76 //Create datalog file
77
78 if(!SD.exists("datalog.csv")) {
79
80     myFile = SD.open("datalog.csv", FILE_WRITE);
81     if (myFile) {
82
83         myFile.println("Time(s),Temperature(C),Pressure(hPa),Altitude(m),Latitude(),Longitude(),Methane(PPM),CO2(PPM)");
84         myFile.close();
85     }
86 }
87
88 }
```

Una vez comprobado, aún en el setup, creamos el archivo en el que se guardarán los datos por la SD.

```

89 void loop(){
90
91   if (!reach_ground) {
92     // Read sensors:
93
94     //GPS
95     for (unsigned long start = millis(); millis() - start < 1000){
96
97       while(softSerial.available()){
98         gps_data = softSerial.read();
99         if (gps.encode(gps_data)) {
100           new_data = true;
101         }
102       }
103     }
104
105     if (new_data) {
106
107       gps.f_get_position(&flat, &flon, &age);
108       latitude_val = (flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);
109       longitude_val = (flon == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6);
110     }
111
112     //BMP280
113     altitude_val = bmp.readAltitude(1013.25);
114     temperature_val = bmp.readTemperature();
115     pressure_val = bmp.readPressure()/100;
116
117     //MQ135
118     co2_val = gasSensor.getPPM();
119
120     //MQ4
121     MQ4_value = analogRead(MQ4_pin);
122     MQ4_volt = MQ4_value * (5.0 / 1023.0);
123     MQ4_RSgas = ((5.0 * 10.0) / MQ4_volt) - 10.0;
124     MQ4_ratio = MQ4_RSgas / MQ4_R0;
125
126     methane_log = (log10(MQ4_ratio) - y_intercept) / slope;
127     methane_val = pow(10, methane_log);
128
129     //Time
130     seconds= millis()/1000;

```

Ahora sí, comenzaríamos 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.

```

132 //Save information SD
133
134 myFile = SD.open("datalog.csv", FILE_WRITE);
135
136 if(myFile) {
137   myFile.print(seconds);
138   myFile.print(",");
139   myFile.print(temperature_val);
140   myFile.print(",");
141   myFile.print(pressure_val);
142   myFile.print(",");
143   myFile.print(altitude_val);
144   myFile.print(",");
145   myFile.print(latitude_val);
146   myFile.print(",");
147   myFile.print(longitude_val);
148   myFile.print(",");
149   myFile.print(methane_val);
150   myFile.print(",");
151   myFile.println(co2_val);
152   myFile.close();
153 }
154
155

```



```
156     //Send information Radio
157
158     Serial.print(seconds);
159     Serial.print(",");
160     Serial.print(temperature_val);
161     Serial.print(",");
162     Serial.print(pressure_val);
163     Serial.print(",");
164     Serial.print(altitude_val);
165     Serial.print(",");
166     Serial.print(latitude_val);
167     Serial.print(",");
168     Serial.print(longitude_val);
169     Serial.print(",");
170     Serial.print(methane_val);
171     Serial.print(",");
172     Serial.println(co2_val);
173 }
```

Una vez ha recogido los datos, los guardará en la SD y los enviará por radio.

```
166 //check if it has reached the top
167
168 if (altitude_val - pre_altitude_val > 3 && !top_alt && altitude_val > 200) {
169
170     top_alt = true;
171
172     if(!SD.exists("top_alt.txt")) {
173
174         topFile = SD.open("top_alt.txt", FILE_WRITE);
175         if (topFile) {
176
177             topFile.print("The CanSat has reached the top at ");
178             topFile.print(altitude_val);
179             topFile.print("m");
180             topFile.close();
181
182         }
183     }
184 }
```

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).

```
195     //check if it's on the ground and turn on the buzzer
196
197     else if(pre_altitude_val == altitude_val && top_alt) {
198
199         tone(buzz_pin, 600);
200         reach_ground = true;
201     }
202
203     pre_altitude_val = altitude_val;
204 }
```

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 y volverá a empezar de nuevo el bucle (loop).

5.3.2. 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.

5.4. Estación de tierra

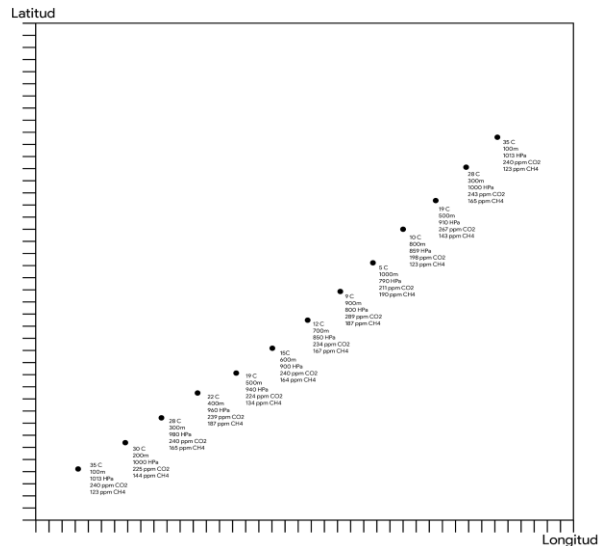
Para la estación de tierra, se programó una interfaz en processing la cual al recibir del puerto serial los valores procedentes de los sensores aportados por el código de Arduino reproduce cinco funciones con cada uno de los sensores con sus valores correspondientes. Con este programa se pueden visualizar en la estación de tierra en tiempo real los datos registrados por los sensores durante la caída recibidos gracias a la antena. El código está subido en GitHub y se puede visualizar en el subsiguiente enlace:

<https://github.com/WBCanSat/CanSatProject/tree/main/CanSatCode/Ground%20Station>

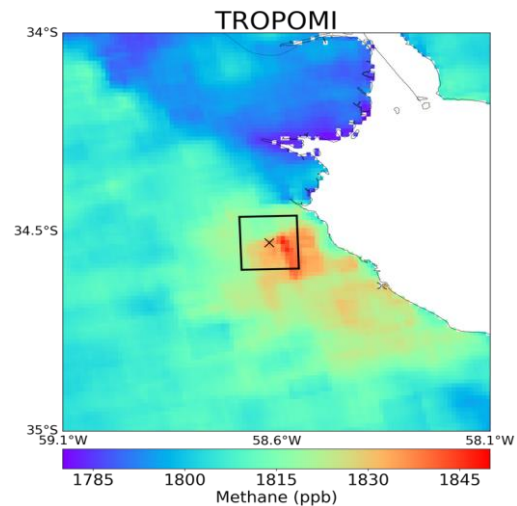
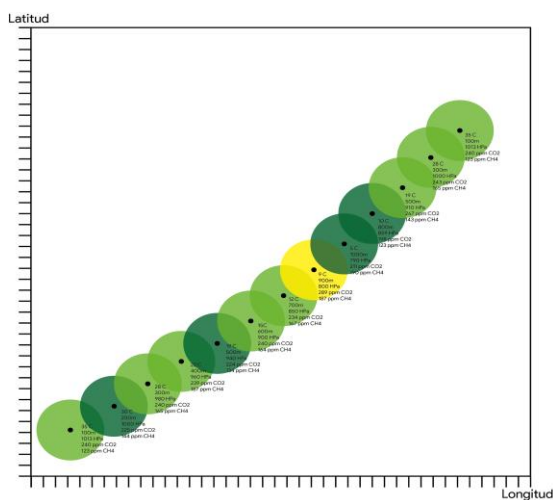
6. SISTEMAS IMPLICADOS EN LA ESTACIÓN DE TIERRA

6.1. 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 los datos obtenidos, para que se vean más fácilmente y para simular, como se verían si se usasen sensores de espectrometría de imagen de alta resolución. Se muestra un ejemplo de lo que queremos conseguir y una imagen captada con un espectrómetro aéreo de imagen de alta resolución



(León, 2020)

6.2. 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 antena utilizada es una antena omnidireccional tipo duck. 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, graficará y mapearán los datos recibidos.

7. 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 cruz 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.

7.1. Diseño

Según la organización, la altitud a la que llegará el CanSat será de unos 1000 metros aproximadamente y su tiempo de vuelo máximo no puede superar los 120 segundos, por lo que queremos que el CanSat baje a una velocidad de entre 8-11 m/s. Para ello, hemos diseñado un paracaídas de tipo cruz que se encuentra en la parte superior del CanSat y que se desplegará cuando comience el descenso.



7.2. Cálculos

Para realizar los cálculos para averiguar el área necesaria para el paracaídas utilizamos esta fórmula para durante el descenso:

$$A = \frac{2 \cdot m \cdot g}{C \cdot p \cdot v^2}$$

- **m** = masa del paracaídas = 350g
- **g** = gravedad = 9.8 m/s²
- **A** = superficie del paracaídas (en nuestro caso de cruz).
- **v** = velocidad de descenso = 8-11 m/s (se realizan dos cálculos, uno para el área más pequeña con 8 m/s y otro para el área más grande con 11 m/s)
- **C** = coeficiente rozamiento = 0.8
- **P** = coeficiente rozamiento del aire = 1.225 kg/m³

$$A = \frac{2 \cdot 0,350 \cdot 9,8}{0,8 \cdot 1,225 \cdot 8^2} = 0,636$$

$$A = \frac{2 \cdot 0,350 \cdot 9,8}{0,8 \cdot 1,225 \cdot 11^2} = 0,875$$

- Resultado área más pequeña: 0,636 m²
- Resultado área más grande: 0,875 m²

Los videos realizando las pruebas se pueden ver en el GitHub. Se adjunta, a continuación, el enlace:

<https://github.com/WBCanSat/CanSatProject/tree/main/FDR/Videos>

De ellas extraemos la conclusión de que el cansat tendrá un aterrizaje seguro.

Peso (g)	Tiempo (s)	Velocidad media (m/s)
100	4,28	2,803738318
200	3,54	3,389830508
300	2,96	4,054054054
400	2,78	4,316546763
500	2,62	4,580152672
Altura (m)	12	

- Resultado área más pequeña: 0.3181 m²
- Resultado área más grande: 0.4375 m²

7.3. Material

Para la fabricación del paracaídas hemos utilizado bolsas hechas de nylon, un material resistente y ligero ideal para paracaídas. Para conectar los extremos y coser el paracaídas hemos utilizada tanza.



8. 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 111.53€ 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.

8.1. 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)		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	Chubasquero de Nylon:2€	3.50
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€
Arduino Nano		1	12.00€	12.00€
			Total:	111.53€

9. FINANCIACIÓN Y APOYO TÉCNICO

9.1. 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 a través de un torneo de videojuegos online que celebrado entre el 26 y 27 de marzo para obtener beneficios cubriendo, de este modo los gastos que suponen la creación del CanSat. Para ello, contactamos con nuestro centro el cual aportó 90€ junto con el AMPA (45 euros el centro y 45 euros el AMPA) para facilitar los premios del torneo. No conseguimos contactar con ninguna empresa que quisiera facilitarnos algún producto

para los premios. Sin embargo, con el dinero aportado por las dos instituciones anteriores fue suficiente para atraer a jóvenes al torneo. Una vez finalizado, el concurso extrajimos 90 euros de beneficios. A continuación, se muestra los carteles que creamos para promocionar el torneo mediante las redes sociales.



9.2. 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).
- 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.

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

10.1. SOLUCIONES A POSIBLES PROBLEMAS

10.1.1. Funcionamiento del CanSat

Ante cualquier imprevisto que tenga que ver con el funcionamiento del CanSat, buscaremos el origen del problema (si falla algún sensor etc...) y lo solucionaremos lo más rápido posible y posteriormente hacer pruebas para ver si se ha solucionado

el problema ya que el propio CanSat es la parte más importante de nuestro proyecto. En caso de que ocurra el mismo día del lanzamiento y el problema es alguna parte física del CanSat, tendremos partes de repuesto preparadas para cambiarlas por las partes inservibles.

10.1.2. Realización de la misión principal

Si llega a haber algún problema al realizar la misión principal en las pruebas de nuestro CanSat, el problema será que alguno de los sensores que utilizamos en esta misión no funcione correctamente, en este caso tendremos sensores de repuesto para poder sustituirlos por los sensores que fallen. Se hará lo mismo en el caso de que este imprevisto ocurra el mismo día de lanzamiento

10.1.3. Comunicación por radio

Si cuando realizamos las pruebas, falla la comunicación por radio, miraremos si es debido a la frecuencia, si la radio está rota o si la comunicación no llega lo suficiente lejos; en cuyo caso solucionaremos ese problema y volveremos a realizar las pruebas. En el caso de que alguno de estos problemas ocurre en el día de lanzamiento, si es la frecuencia la ajustaremos a la adecuada, si es la propia radio tendremos una de repuesto preparada y si es por la distancia, llevaremos un pequeño amplificador de señal que hará que lleguemos a la distancia necesaria.

10.1.4. Realización de la misión secundaria

En el caso de que no podamos realizar la misión secundaria a la hora de hacer las pruebas, será porque alguno de los sensores que utilizamos no funciona adecuadamente, en cuyo caso tendremos sensores de repuesto para cambiar por los sensores que no funcionen, incluyendo si este imprevisto ocurre el mismo día de lanzamiento.


10.1.5. Pruebas del paracaídas

Si en el momento de hacer las pruebas del paracaídas no funciona como queremos (no tiene la velocidad que buscamos), revisaremos los cálculos y la tela para ajustar un paracaídas adecuado para aguantar al CanSat. En caso de que el paracaídas se rompa antes del lanzamiento, se sustituirá por un paracaídas de repuesto igual al original.

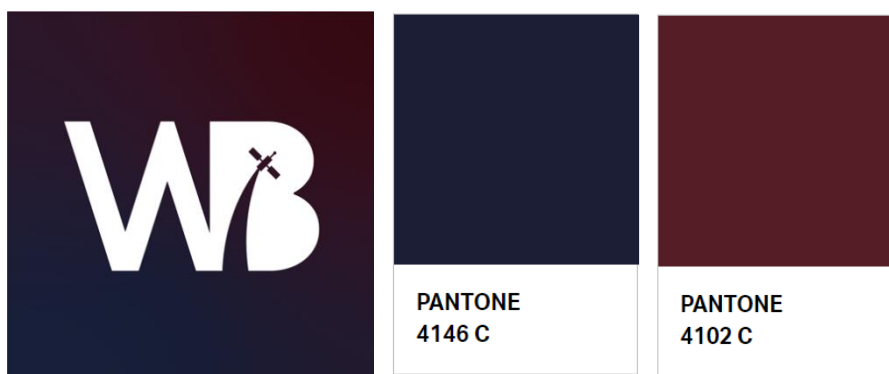
11. ESTADO DE LA DIFUSIÓN Y DIVULGACIÓN DEL PROYECTO. FOTOS, VÍDEOS, ENLACES DEL PROGRESO.

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:

	Red social	Usuario	Enlace web
	GitHub	Wbcansat	https://github.com/WBCanSat/CanSatProject
	Instagram	wbcansat	https://www.instagram.com/wbcansat/?igshid=1b64ue1rplmxx
	Twitter	WBCansat	https://twitter.com/wbcansat?s=21
	Facebook	WB Cansat	https://www.facebook.com/wb.cansat

Hemos creado un logo personalizado para una mejor identificación del equipo. 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, lanzar un satélite. El color principal es el Pantone 4146 C y el secundario el Pantone 4102C.



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.

11.1. Difusión con el ayuntamiento

Para divulgar los conocimientos científicos y tecnológicos pensamos en 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. Aunque finalmente, no nos ha sido posible conseguir difusión con el Ayuntamiento de San Fernando, se adjunta, a continuación, un enlace al documento que hace referencia a la propuesta de difusión que solicitamos: <https://github.com/WBCanSat/CanSatProject/blob/main/CDR/Townhall%20request.pdf>

12. 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. 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 a través de las redes sociales.

12.1. BIBLIOGRAFÍA: REFERENCIAS E INFORMACIÓN UTILIZADA

- Anon., 2015. *AGU*. [En línea]
Available at: <https://agupubs.onlinelibrary.wiley.com/doi/full/10.1002/2014GL062943>
- Anon., 2019. *Esero*. [En línea]
Available at: <http://esero.es/cansat/>
- Anon., 2020. *Luis Llamas*. [En línea]
Available at: <https://www.luisllamas.es/medir-temperatura-y-presion-barometrica-con-arduino-y-bmp280/>
- Anon., 2021. *Barnacle News*. [En línea]
Available at: <https://thebarnaclenews.com/ngc-signs-game-changing-mou-with-orbital-eye-b-v-in-actioning-its-green-agenda/>
- Anon., s.f. *Shield o módulo GPS con Arduino: Introducción*. [En línea]
Available at: <https://www.geekfactory.mx/tutoriales/tutoriales-arduino/shield-o-modulo-gps-con-arduino/>
- Bishton, D., 2020. *Spatial Source*. [En línea]
Available at: <https://www.spatialsource.com.au/gis-data/microsatellite-records-smallest-methane-leak-detected-from-space/attachment/sfl-ghgsat-image>
- CanSat, E., 2019. *Edición 2019*. [En línea]
Available at: <http://esero.es/cansat/ediciones-anteriores/cansat-edicion-2019/>
- Cansat, E., 2020. *Edición 2020*. [En línea]
Available at: <http://esero.es/cansat/ediciones-anteriores/cansat-edicion-2020/>
- Esa, 17/09/2020. *High-resolution methane emissions data for waste management*. [En línea]
Available at:
https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Sentinel-5P/High-resolution_methane_emissions_data_for_waste_management
- ESA, 2015. *Plant power from above*. [En línea]
Available at:
http://www.esa.int/Applications/Observing_the_Earth/Plant_power_from_above
- ESA, 2015. *Plant power from above*. [En línea]
Available at:
http://www.esa.int/Applications/Observing_the_Earth/Plant_power_from_above
- Kate Maxwell, G. N. J., 2000. *Oxford Academy*. [En línea]
Available at: <https://academic.oup.com/jxb/article/51/345/659/652534>
- León, F. M., 2020. *Meteored*. [En línea]
Available at: <https://www.tiempo.com/ram/datos-de-emisiones-de-metano-de-alta-resolucion.html>
- Meteored, 18/09/2020. *Datos de emisiones de metano de alta resolución*. [En línea]
Available at: <https://www.tiempo.com/ram/datos-de-emisiones-de-metano-de-alta-resolucion.html>

- Meteored, 18/09/2020. *Datos de emisiones de metano de alta resolución*. [En línea]
Available at: <https://www.tiempo.com/ram/datos-de-emisiones-de-metano-de-alta-resolucion.html>
- W. W. Biggs, A. R. E. J. D. E. K. W. B. J. W. M. a. M. D. C., 1971. *JSTOR*. [En línea]
Available at: <https://www.jstor.org/stable/1934743?seq=1>