

Space Six

CDR

Oscuro Angel

Eulogio Sorbas
<http://www.rtve.es/alacarta/videos/otros-documentales/otro-documentales-luna-nuestra-puerta-universo/5347281/> 16:00

Eulogio Sorbas

Otros documentales
La Luna muestra fiel compañera es mágica y misteriosa somos al
www.rtve.es/alacarta/videos/otros-documentales/otro-documentales-luna-nuestra-puerta-universo/5347281/ 16:01

SPACE SIX

CENTROS GUADALINFO SORBAS-ULEILA

HOY

Oscuro Angel

Recordatorio para Ben Jonney P. Azuaga a las 9:30 en el tanatorio

Antonio de Juanamínguez marzo 2020

Recordatorio para los de Uleila, a las 9:30 en guadalinfo 12:36

Tema	Página
Índice de contenidos	2
Introducción/antecedentes	3
Presentación y Plan de Organización de Equipo	5
Descripción de los objetivos para la misión primaria. Hardware y software elegido.	8
Diseño Mecánico del CanSat y de la Estación de Tierra	9
Prototipos desarrollados y esquema de conexión del desarrollo definitivo	11
Sistema de Aterrizaje de la misión primaria	16
Descripción de los objetivos para la misión Secundaria. <ul style="list-style-type: none"> • Misión secundaria Estandar • Misión secundaria Extra 	19
Programación. Lenguaje de Programación y diagrama de flujo	39
Análisis de los datos de telemetría del Satélite	49
Presupuesto Detallado	55
Sostenibilidad	56
Difusión del proyecto <ul style="list-style-type: none"> • Prensa • Divulgación insitu en el IES • Logo • YouTube • Blog • Twitter 	58
Anexo	65

Introducción/antecedentes

Los centros Guadalinfo de Uleila y Sorbas , de la Comarca Filabres-Alhamilla del interior de Almería, vienen realizando tareas de formación dirigidas a jóvenes de la comarca.

Estos centros, con el paso del tiempo, se han convertido en lugares de referencia para todos aquellos que disfrutan con las TIC y que ven en la ciencia y la tecnología una fuente inagotable para la diversión , investigación y conocimiento.

Las propuestas lúdicas y de formación que se realizan son de lo más variadas y van desde los cursos de programación en Scratch, Javascript , Python o C , hasta la participación en concursos provinciales de videojuegos o la organización de concursos de robots , e incluso se realizan talleres de biotecnología donde los alumnos analizan y cambian el ADN de bacterias.

Fué en este ambiente de investigación y diversión donde tuvimos conocimiento ,al final del verano de 2019 , del concurso CanSat, después de haber realizado un taller sobre la construcción de cohetes propulsados por presión de aire, también ese verano realizamos varios talleres para la observación del firmamento, para nosotros el momento no podía ser más propicio.

Después de varios encuentros para diseñar una suerte de plan estábamos todos tan emocionados que no dudamos ni un momento en ponernos manos a la obra.

Para ello estamos contando , además, con el apoyo de la asociación Científico-Tecnológica “Guadatech” , de los Ayuntamientos de Uleila y Sorbas , y del área de asuntos sociales de la Diputación de Almería.

No formamos parte de un instituto, ni de un colegio ,sencillamente somos un grupo de vecinos que, cada uno con motivaciones distintas queremos mostrar y demostrar que “la España vaciada” tiene muchas posibilidades y mucha gente de todas las edades interesadas en descubrir el futuro y formarnos para afrontarlo.

○ Proyectos en los que el equipo haya participado anteriormente o se hayan realizado su centro educativo, relacionados con la tecnología.

- Torneo de robots comarcal 2017
- Curso Python 2017-2018
- Curso de Raspberry Pi 2017
Jam Today Almería 2018 (ganadores del premio al mejor videojuego con Python)
- Curso JavaScript 2019
- JamToday 2019
- Encuentro de Biotecnología 2017
- Taller Cohetes de Agua 2019
- Talleres de observación estelar 2019

Presentación y Plan de Organización de Equipo

Somos un grupo de jóvenes pertenecientes a la comarca Filabres-Alhamilla (Almería, España), que se han embarcado en el proyecto CanSat con ilusión y muchas ganas de aprender.



Foto de Grupo



Beni, programación, análisis de datos



Alex y Iona , estación de tierra , telecomunicaciones

“Que el camino esté lleno de aventuras y conocimiento” C. Cavafis 6



Cristian , Robert y Fran, Los tres de la izquierda: Redes sociales, paracaídas , estructura del Satélite

Descripción de los objetivos para la misión primaria. Hardware y software elegido

La misión primaria consistirá en el envío de los datos de altitud y temperatura por radiofrecuencia una vez que el satélite sea desplegado.

En nuestro caso, como queremos que nuestro diseño sirva también para montarlo en nuestros cohetes de agua y sea ligero y pequeño, para ahorrar peso y espacio no utilizaremos un adaptador y una tarjeta SD para el guardado de Datos en el satélite , sino que utilizaremos la propia EEPROM del Arduino , el limitado espacio de la misma nos obliga a hacer solo una medida cada segundo, al límite de lo que marca el reglamento.

Arduino Nano : Elegimos este arduino debido a su tamaño ya que es necesario para que encaje en los cohetes de agua ecológicos que utilizamos para realizar parte de la misión secundaria “esto se explicara a detalle en la explicación de las misiones secundarias”

Batería reciclada de Litio – Testeada, Utilizamos esta batería por que es del tamaño idóneo para nuestros cohetes de agua además de que es reciclada / ecológica .Tiene entre 400-1000 mA

Sensor de Altitud, Presión y Temperatura/Bmp280-Testeado : Elegimos este sensor debido a que es económico , de pequeño tamaño y eficaz

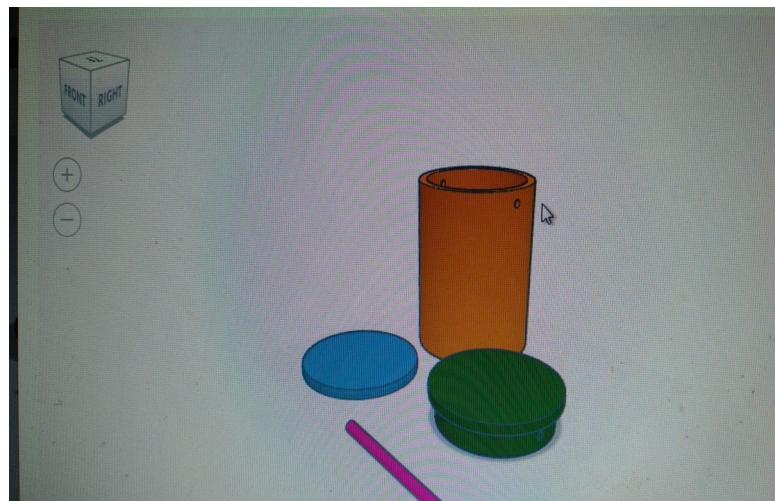
Zumbador, led - Testeados: Vamos a utilizar el zumbador y el led para comunicarnos con el procesador e interactuar con él .

APC220 - Construida pero sin testear (Fotos):La antena está construida, pero no la hemos probado todavía. Utilizaremos un emisor y receptor de radio frecuencia los cuales son los modelos estándares de las competiciones de cansat

Un interruptor de alimentación general perfectamente accesible :Lo vamos a usar para poder apagar y encender el arduino con la finalidad de ahorrar la batería.

Diseño Mecánico del CanSat

El diseño 3D fue hecho en una página llamada Tinkercad (www.tinkercad.com) con la colaboración de todos los integrantes del grupo en la siguiente imagen se puede ver un boceto inicial del diseño.



Boceto inicial del diseño

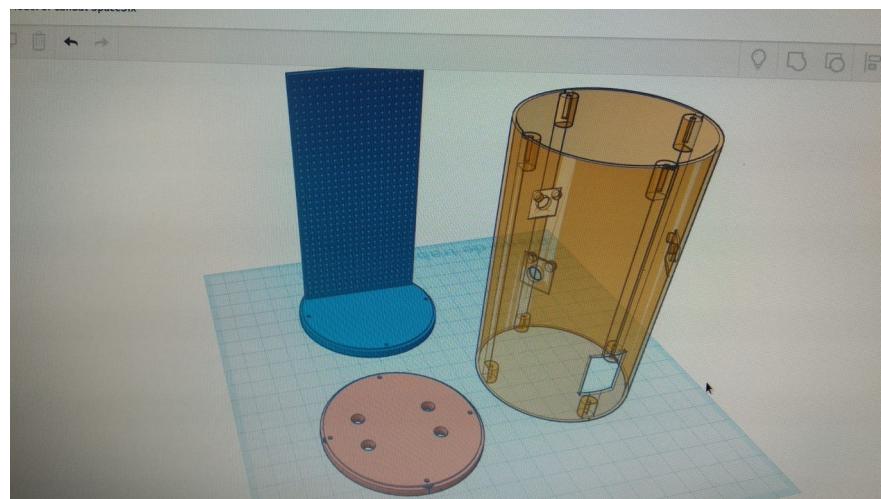
Tras realizar un primer diseño imprimimos una versión beta con una impresora 3D de manera rápida con el material PLA pero la versión final estará en ABS porque tiene una mayor resistencia mecánica a los golpes.

Hemos diseñado nuestra carcasa con tres agujeros equidistantes para poner los tres sensores ultravioleta de nuestra misión secundaria y que cumplan de manera óptima su misión de cubrir toda la radiación solar que rodee al satélite.

Para sujetar el paracaídas a la estructura hemos realizado cuatro agujeros en una de las tapas de la estructura, que irá atornillada al cilindro principal con tornillos metálicos.

La otra tapadera, la inferior , ha sido diseñada como apoyo de una pared vertical que nos ayudará a situar los componentes del Cansat y a compartimentar el interior del cilindro.

También hemos realizado una apertura de acceso mediante dedo al interruptor general de corriente del cansat , pero que tendremos que hacerla más grande en el diseño final , porque nos hemos dado cuenta que en la actual sólo cabe el dedo meñique de una persona adulta.



Diseño más avanzado

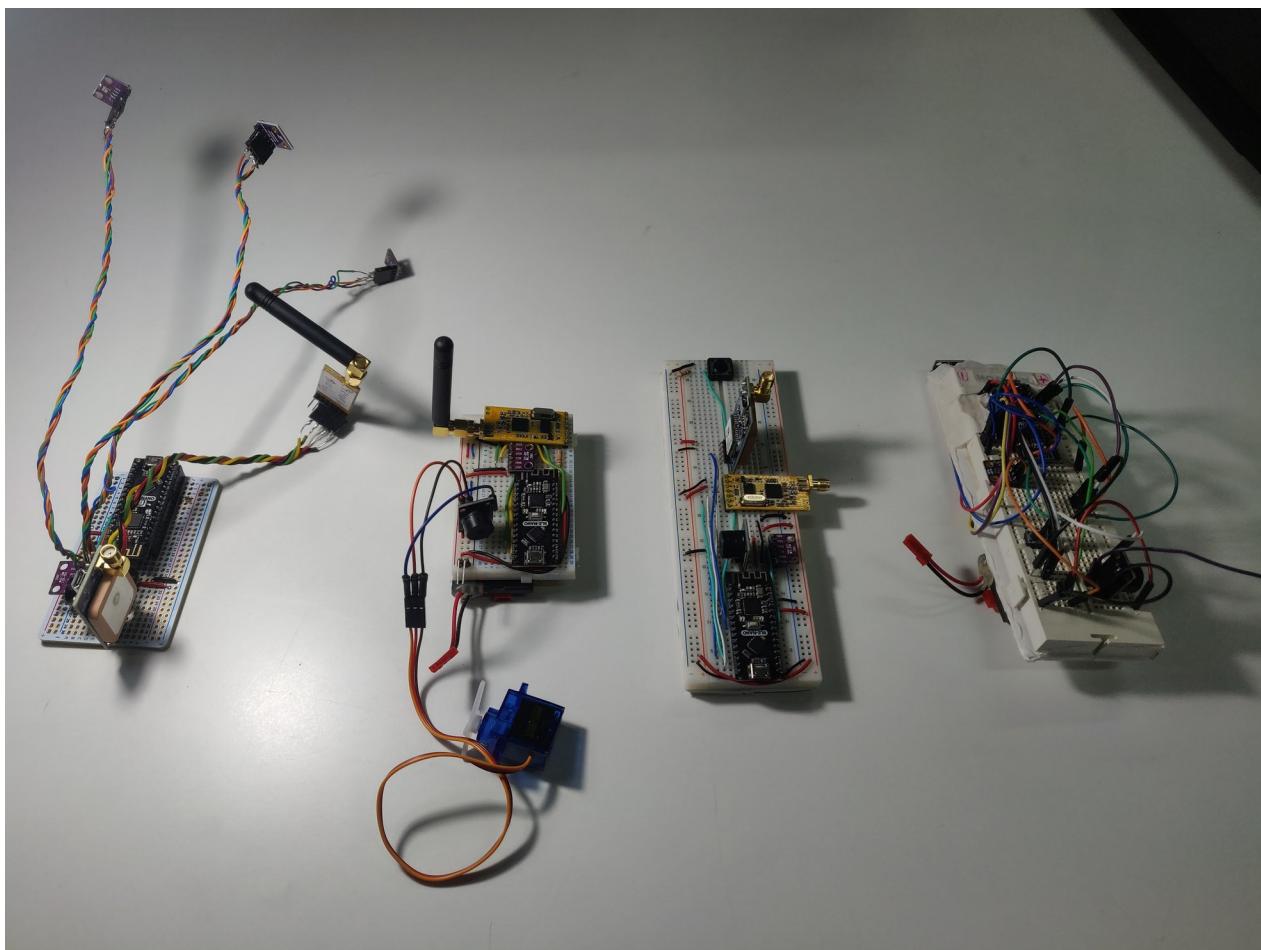


Vista general de los elementos de la estructura ya impresa

Prototipos desarrollados y esquema de conexión del desarrollo definitivo

En total hemos realizado 4 desarrollos : 3 prototipos y uno definitivo

En la siguiente imagen podemos ver los desarrollos , siendo el de la derecha el primero que hicimos y el de más a la izquierda el desarrollo final que va montado en nuestro Cansat.



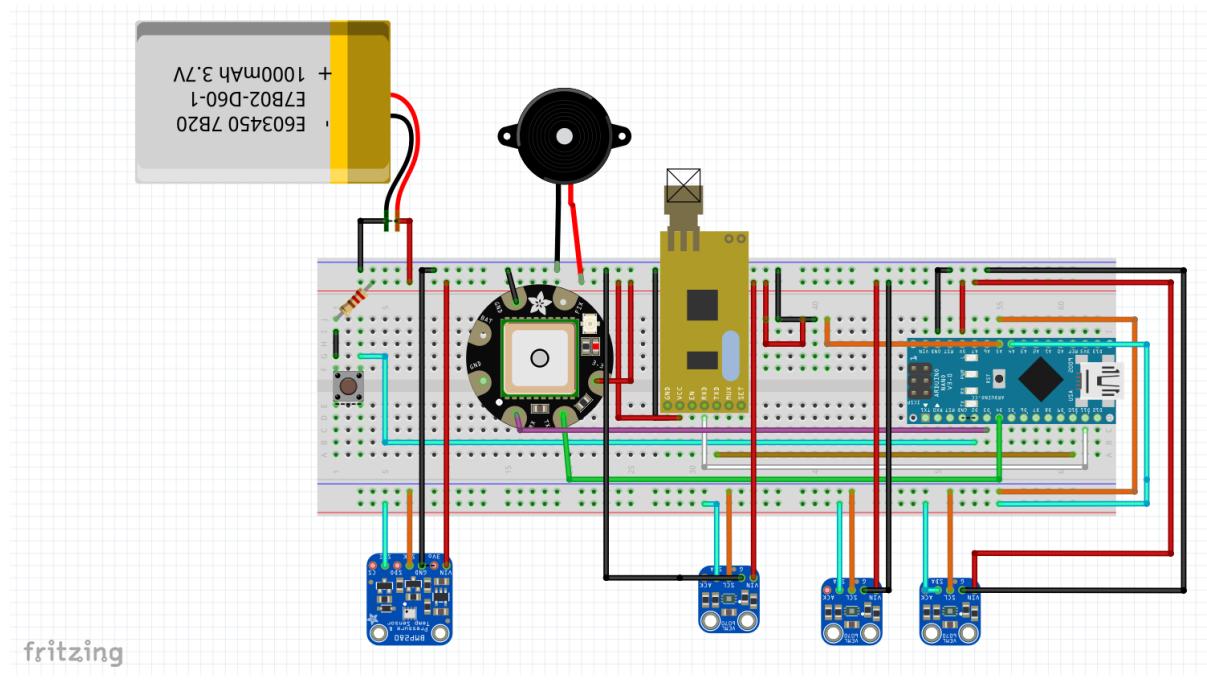
Desarrollo de los prototipos realizados para su inclusión en el CanSat

Como se puede apreciar hemos ido ganando experiencia con el prototipado y nuestro diseños han evolucionado desde un lio de cables que apenas cabía en el protobard a **un montaje muy liviano que es lo que pretendíamos conseguir ya que una de nuestras ideas era poder montar nuestra electrónica en nuestros cohetes de agua** , para lo cual necesitábamos reducir al máximo el peso.

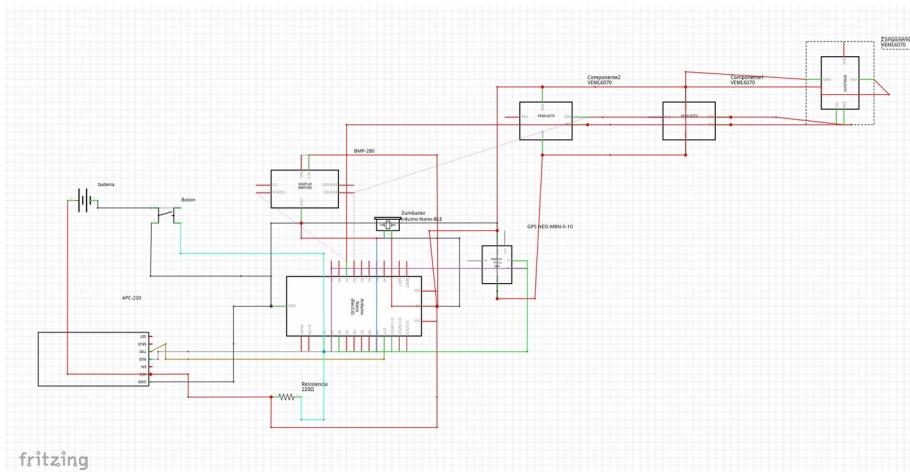
En este sentido, el tercer desarrollo contando por la derecha, en el que se incluye un servo está pensado precisamente para ir en nuestros cohetes de agua.

El desarrollo definitivo tiene 3 cables con sensores “al aire” porque estos son los sensores UV que se tienen que poder recolectar libremente por los distintos agujeros practicados en el cilindro del Cansat

A continuación se muestra el cableado definitivo en el que nos gustaría destacar el gran ahorro de cableado que nos ha supuesto el darnos cuenta que la mayoría de nuestros sensores utilizan el protocolo I2C y por lo tanto pueden compartir un par de cables , o como se ve en el protobard, un par de líneas , las que inicialmente están pensadas para el positivo y el negativo que las hemos reutilizado como bus común para nuestros sensores.



Protobard



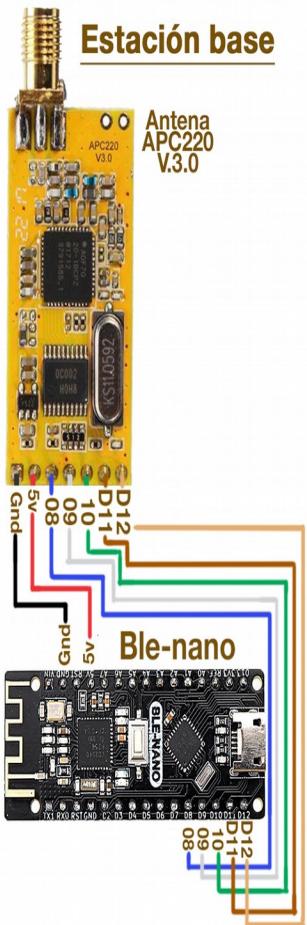
Esquematico



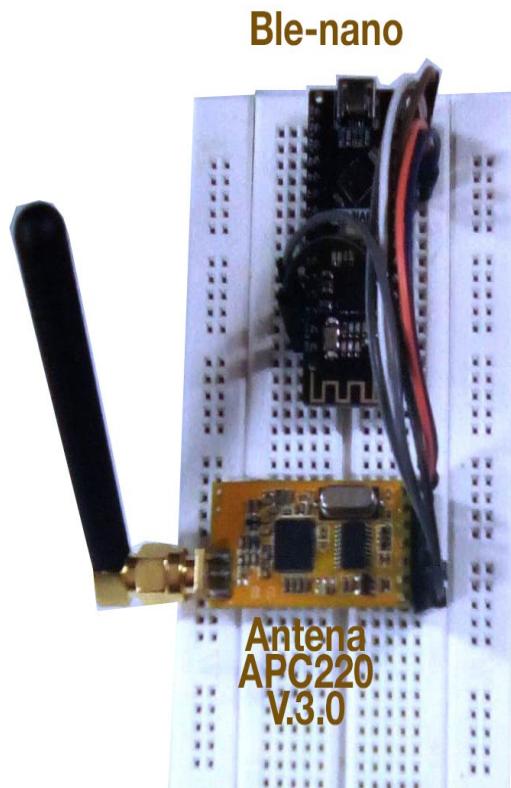
Miembros del equipo soldando los sensores al desarrollo definitivo

Estación de tierra

Para la estación de tierra hemos usado un Portatil, un Arduino Nano-Ble, un APC220 y una antena externa



Estación base



Esquema de estación de tierra

Estacion de tierra montada

Hemos usado dichos componentes por su versatilidad y reducido tamaño, además de que suponen un ahorro tanto en dinero como en energía, ya que se alimenta del puerto USB del portátil.

Código de la estación de tierra

```

ejemploEmisionRepcionSencilla

#include <SoftwareSerial.h>

SoftwareSerial comunicacionConSatelite(10, 11);

void setup() {

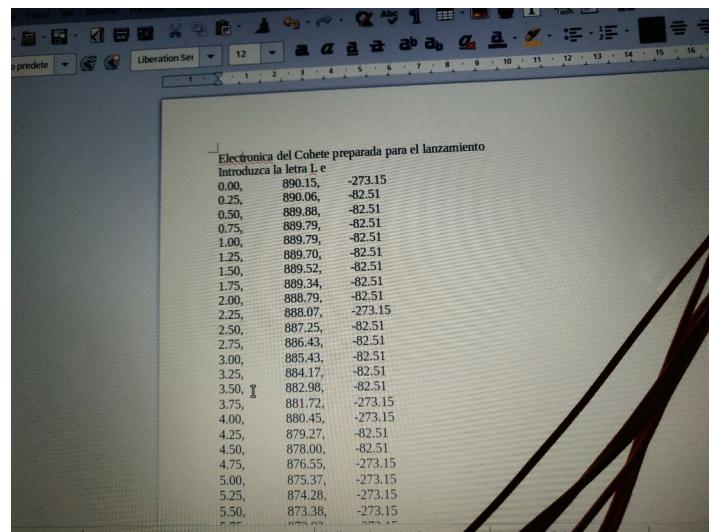
    Serial.begin(9600);
    Serial.println("Estacion de Tierra operativa");

    comunicacionConSatelite.begin(9600);
    comunicacionConSatelite.println("Mensaje inicial enviado desde la estacion de tierra al Satelite");
}

void loop() {
    while (comunicacionConSatelite.available()) {
        Serial.write(comunicacionConSatelite.read());
    }
    while (Serial.available()) {
        comunicacionConSatelite.write(Serial.read());
    }
}

```

Código del Arduino de la estación de tierra



Datos tomados por la estación de tierra en una de las pruebas realizadas

El “cerebro” del satélite usará los mismos componentes que la estación de tierra, es decir, un Arduino Nano-BLE y una antena ACP220.

El satélite envíá los datos al receptor y este, muestra los resultados obtenidos por la pantalla del portátil.

Además la estación de tierra puede mandar órdenes al satélite , como por ejemplo que se vaya preparando para un inminente lanzamiento.

Con la antena externa que le hemos acoplado a la estación de base hemos alcanzado un rango de recepción óptimo de los datos del satélite de hasta 2km



Prueba de campo de la antena externa de la estación base



Análisis de la cobertura con las distintas antenas a diferentes distancias

Sistemas de Aterrizaje
de la misión primaria del satélite
y misión secundaria “Extra” de los cohetes

Paracaídas para la misión Primaria del satélite



Imagen del paracaídas final con una barquilla de prueba

Empezaremos hablando sobre el diseño y construcción del paracaídas , comentaremos las virtudes y inconvenientes de su diseño y como llevamos acabo su construcción.

Comenzaremos con las ventajas y inconvenientes del diseño del paracaidas.

Algunas de las muchas virtudes son:

-La falta de costuras en el diseño lo hacen más fácil de construir o fabricar

-Además en el lugar en el cual se atan las cuerdas del paracaídas, se reforzó con cinta de fibra con el objetivo de reforzar el lugar de mayores esfuerzos y no se rompa

-Además de que con su diseño octagonal conseguimos que con un área de reducido tamaño el cansat descienda a una velocidad relativamente buena . Además de que al tener un tamaño menor conseguimos que pese y ocupe menos espacio.

Algunas de sus inconvenientes son:

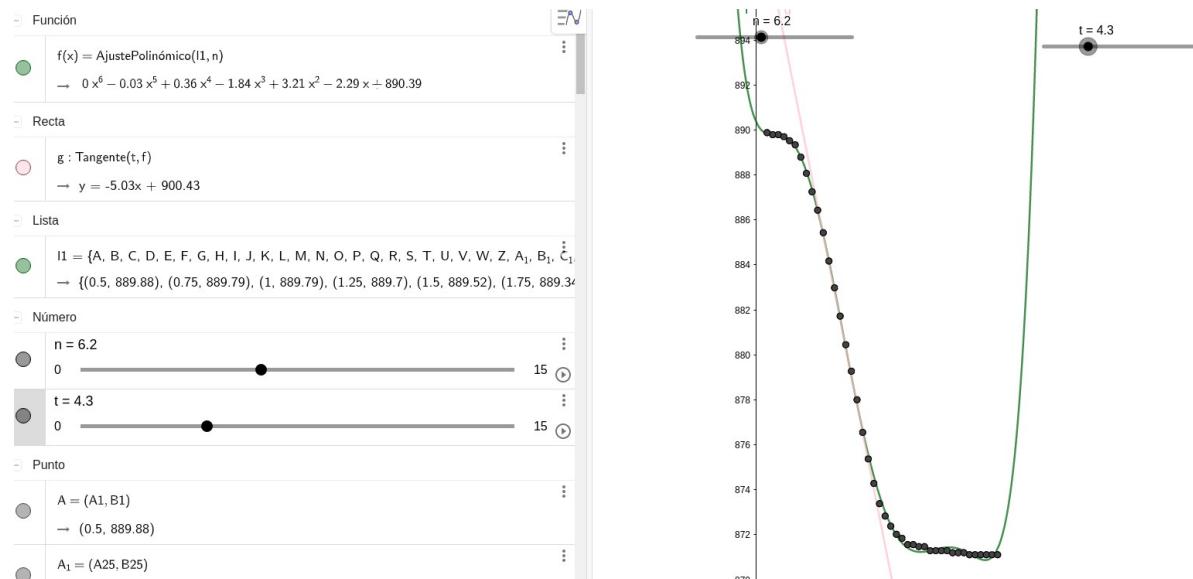
-los nudos que unen el paracaídas y el cansat son algo inestables claro está que esto es algo con fácil solución con solo llevar acabo un poco de investigación sobre su reforzamiento.

-Hay que decir que este es algo más personal ya que nos habría gustado que la forma del paracaídas fuera circular en vez de octagonal debido a que la eficiencia del paracaídas es algo mayor pero nuestro presupuesto y limitaciones técnicas nos lo impidieron.

Ahora hablaremos sobre su proceso de construcción:

Para guiarnos en la construcción del paracaídas utilizamos un tutorial de uswaterrockets que enseña cómo construir un paracaídas octogonal

como material base utilizamos una tela conocida como ripstop la ventaja de esta tela es resistencia a esfuerzos .



Análisis de los datos de una prueba de lanzamiento desde un edificio de 20 m de altura con una versión previa del paracaídas, con un lastre de 320 gramos , en el que cae con una velocidad de 5 m/s.
!Demasiado lento!

Descripción de los objetivos para la misión Secundaria.

Misión Secundaria: Hemos estructurado la misión secundaria en dos áreas, una misión secundaria “Estándar” centrada en el satélite y una “Extra” centrada en los cohetes de agua

Misión Secundaria “Estándar”

- **Estudio de las variables ambientales en relación a la altura,** para eso usaremos los sensores ultravioleta,

Misión Secundaria “Extra”

Estudio de la viabilidad del uso del cohetes de agua a presión como plataforma de pruebas para Cansats.

Las alternativas tradicionales para el testeo de CanSats han sido Drones, Aeromodelismo, Barrancos y Globos Aerostáticos.

Son soluciones caras e inaccesibles para un centro educativo estándar, algunas arriesgadas e insostenibles desde el punto de vista medioambiental.

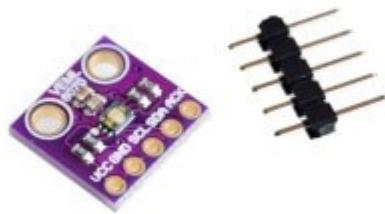
Los cohetes de agua, en cambio ofrecen una alternativa económica y factible a los centros educativos para introducir a los alumnos en los ámbitos científicos y tecnológicos.

Además de los cohetes y quizás mas importante ha sido el diseño y la construcción de dos diferentes tipos de lanzaderas de cohetes de Agua que nos han permitido lanzarlos de manera segura y estandarizada.

Misión secundaria “Estándar”

Idea de nuestra misión secundaria estándar, hipótesis del trabajo

Nuestra misión secundaria se trata de poder medir los rayos ultravioletas con sensores de rayos ultravioleta para poder observar , medir y analizar los niveles de rayos ultravioletas a diferentes alturas.



*El sensor que usaremos es el
VEML6070*

Nuestro montaje experimental

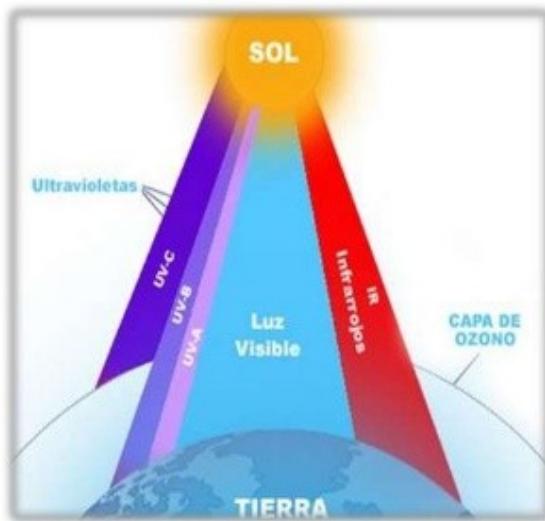
Nuestro montaje experimental en el cohete se compone de tres sensores en el cohete para poder observar el grado de rayos ultravioletas y si el paracaídas o el cohete tapa uno que el resto pueda calcular los rayos ultravioletas para observar el nivel.

¿Qué es la radiación ultravioleta?

Se denomina radiación ultravioleta o radiación UV a la [radiación electromagnética](#) cuya [longitud de onda](#) está comprendida aproximadamente entre los 10 [nm](#) y los 400 nm. Su nombre proviene de que su rango empieza desde longitudes de onda más cortas de lo que los humanos identificamos como el color [violeta](#), pero dicha luz o longitud de onda, es invisible al ojo humano al estar por encima del espectro visible. Esta radiación es parte integrante de los rayos solares y produce varios efectos en la salud al ser una radiación entre no-ionizante e ionizante.

¿Qué es la radiación solar?

- Es un tipo de onda electromagnética que viaja desde el sol hasta la tierra.
- Esta compuesta de rayos UVA y UVB.
- El daño dependerá de la intensidad, tiempo de exposición y de la piel del individuo.



¿Qué es la radiación UV?

Fuente: <https://sites.google.com/site/lampahoy2/fuego/radiacion-uv>

¿Cómo se mide la radiación Ultravioleta?

El índice UV es un indicador de la intensidad de radiación ultravioleta proveniente del Sol en la superficie terrestre en una escala que comienza en 0 y no está acotado superiormente. El índice UV también señala la capacidad de la radiación UV solar de producir lesiones en la piel.

Efectos de los UV en los seres vivos.

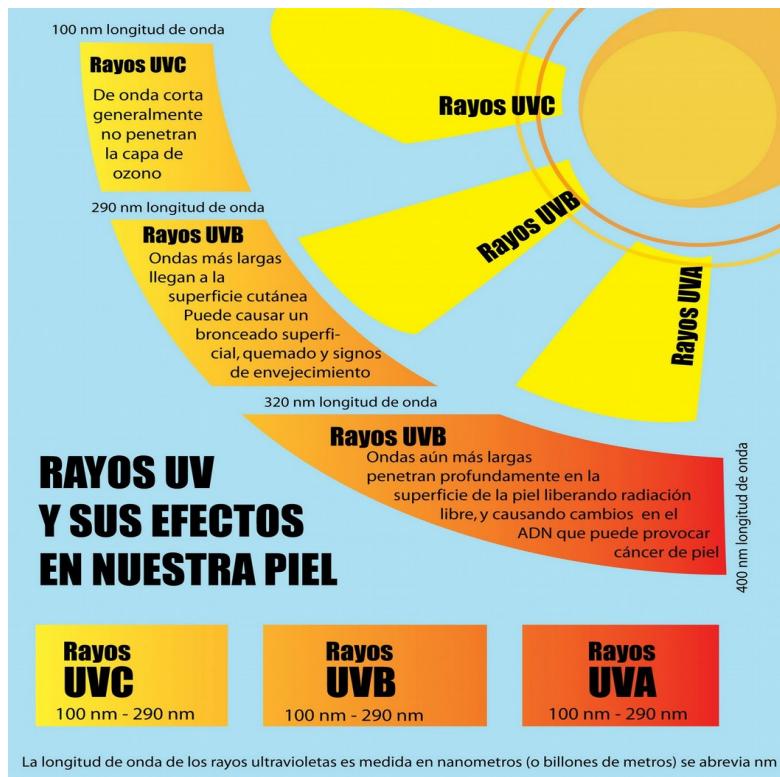
Las radiaciones UV entre 290 nm y 320, se denominan B (UVB) y son las responsables de los efectos biológicos más importantes de dichas radiaciones sobre el ser humano. Sobre la piel, tienen efectos nocivos a corto y a mediano plazo. El enrojecimiento de la piel (eritema solar), desde leve a quemaduras importantes, es el principal efecto nocivo inmediato. Los efectos a mediano plazo, destaca la mayor frecuencia de

cánceres cutáneos y el envejecimiento prematuro de la piel, y las modificaciones en el ADN de los seres vivientes.

Es importante recalcar, que la radiación UV solo constituye un riesgo para la salud

cuando el ser humano se somete repetidamente, durante años, a exposiciones excesivas para su tipo de piel.

El riesgo ante la radiación UV disminuye a medida que aumenta el grado de pigmentación natural de la piel del ser humano, siendo máximo en pieles muy blancas y
mínimo en personas de piel negra.

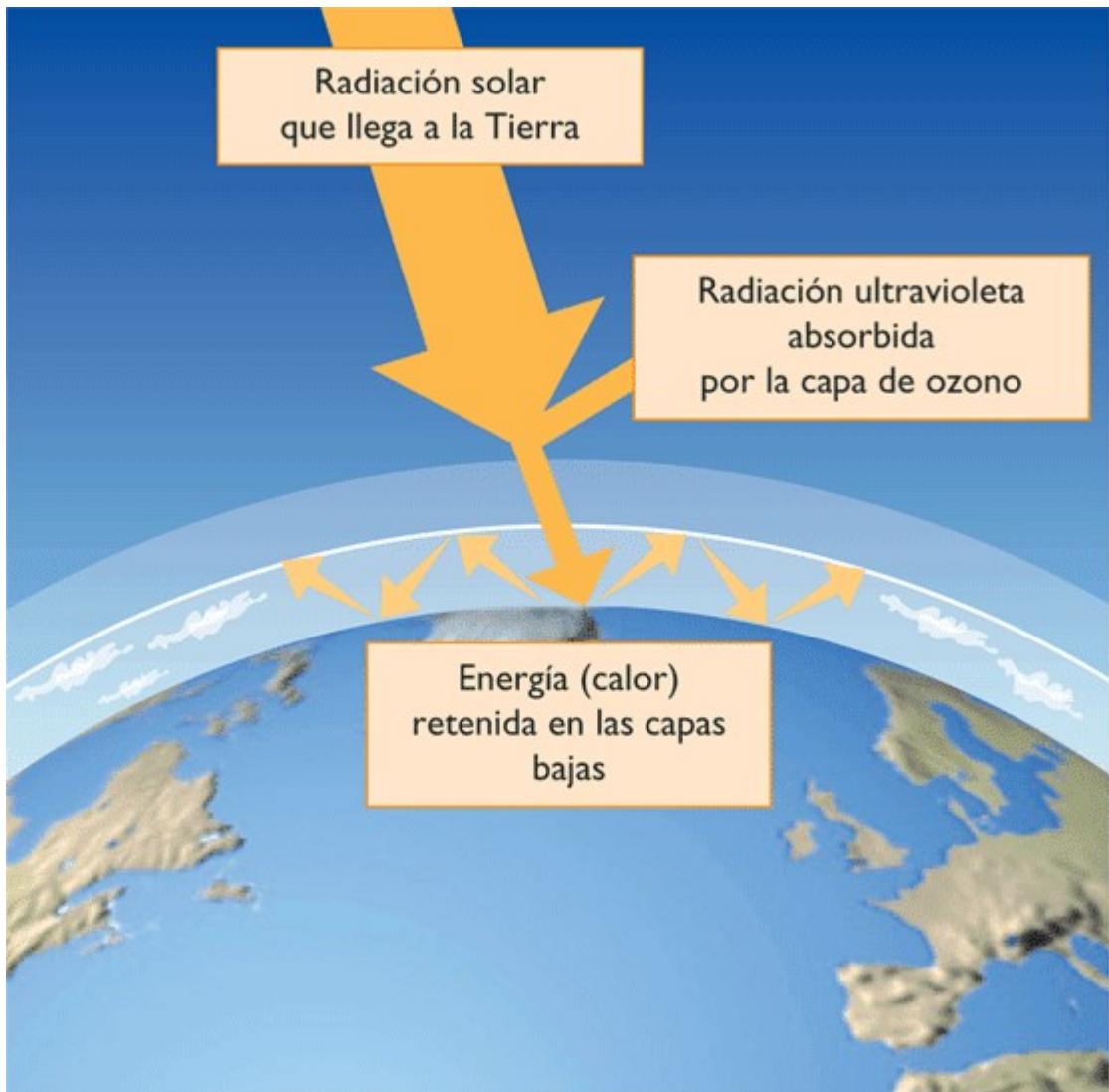


Efectos de la radiación UV en los seres humanos

Fuente: <http://anacarmenmarina.blogspot.com/2013/11/las-radiaciones-ultravioleta.html>

Propagación de los UV en la atmósfera terrestre.

La capa de ozono filtra la mayor parte de la radiación ultravioleta proveniente del sol, especialmente los rayos ultravioleta B, dejando pasar los rayos ultravioleta A, necesarios para la vida en la tierra.



propagación UV en la superficie terrestre

La radiación ultravioleta en otros planetas rocosos

Para explicar cómo afecta la radiación ultravioleta en planetas rocosos , pondré ejemplos de cómo afecta esta a planetas que todos conocemos como Marte , Venus y Mercurio .

Como ejemplo pondré a marte

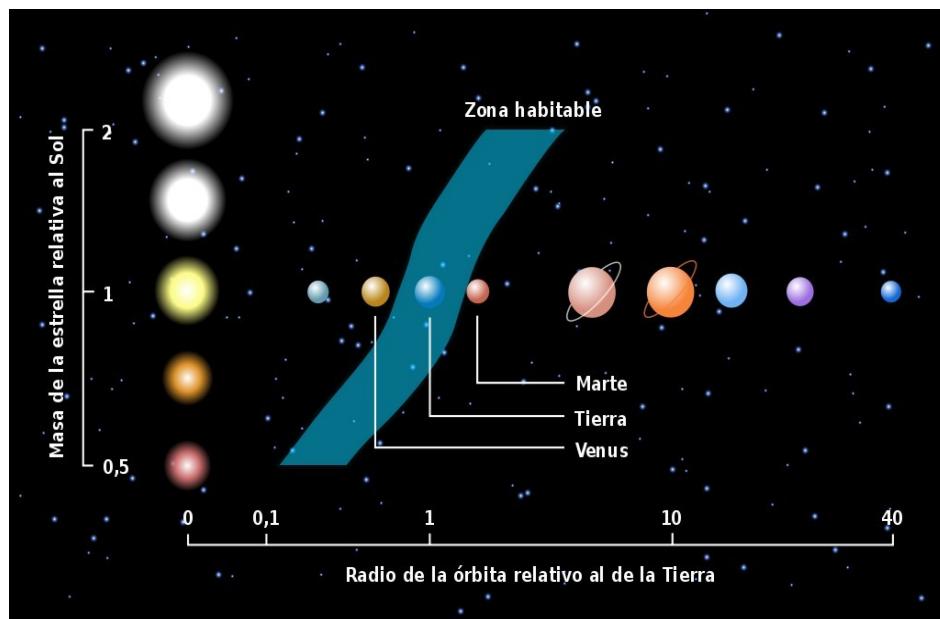
Marte es un planeta que recibe en su superficie radiación ultravioleta(UV) solar con una fuerte componente biológicamente muy perjudicial (UV-C y UV-B), lo que influye notoriamente en el deterioro de la superficie en vistas de poder encontrar algún signo de vida

Por lo tanto, Marte puede ser considerado un planeta auto-esterilizante, es decir, la combinación de la radiación UV solar que alcanza la superficie, la extrema sequedad del suelo y la naturaleza oxidante de la química del suelo impiden la formación de organismos vivos en el suelo marciano.

La radiación UV solar que alcanza la superficie de Marte está fundamentalmente determinada por su atmósfera, tanto en magnitud como en rangopectral. La atmósfera de Marte es bastante diferente de la atmósfera de la Tierra. Está compuesta fundamentalmente por dióxido de carbono (CO₂) con pequeñas cantidades de otros gases este tipo de atmósfera permite que entre una mayor cantidad de rayos ultravioletas que en la tierra

La curva de radiación UV en un planeta como un posible indicador de vida extraterrestre

Estas radiaciones son necesaria en pequeñas cantidades, para el desarrollo de la vida: fotosíntesis de las plantas, síntesis de vitamina D en los humanos, etc.

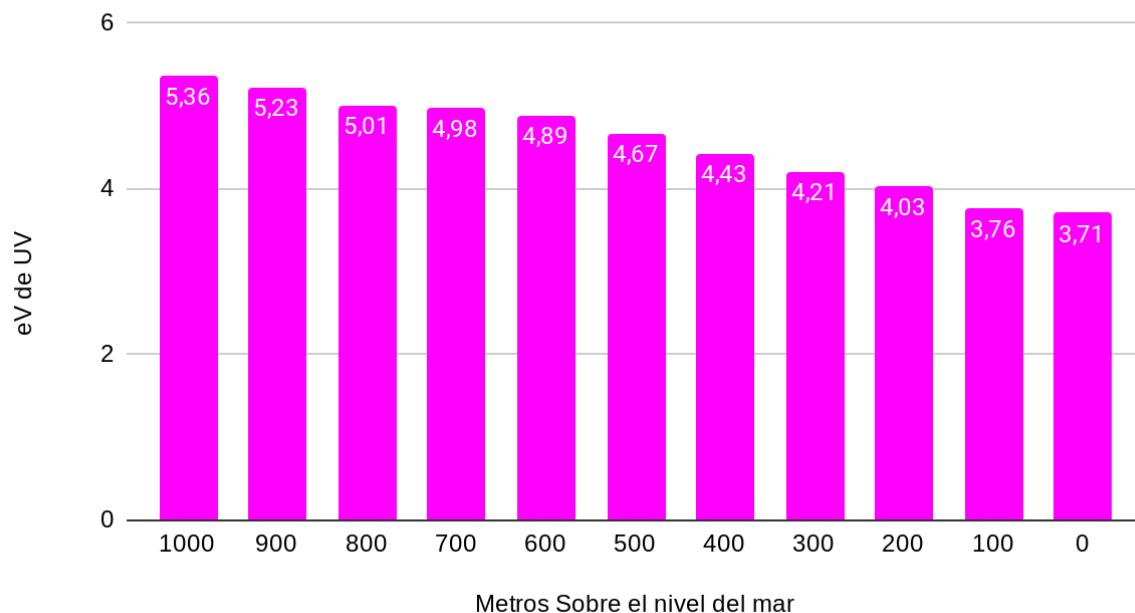


Cuanto más cerca este un planeta a una fuente UV, menor es su habitabilidad

Pero en grandes cantidades estas radiaciones resultan nocivas para la vida imposibilitandola pero al mismo tiempo la nula presencia de la misma también imposibilita la vida. Por ello el detectar una correcta cantidad de uv en un planeta es un buen indicador de vida extraterrestre.

Resultados esperados.

eV de UV frente a Metros Sobre el nivel del mar



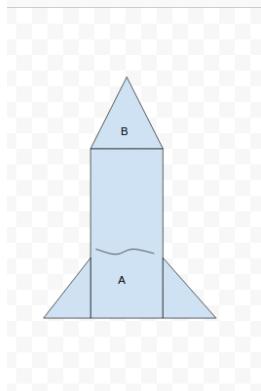
Estos son los datos de la gráfica que nosotros esperamos, debido que cuanta más altura tengamos mayor será la radiación ultravioleta y cuanto más bajo menor será esta radiación.

Misión secundaria “extra”, la física detrás de los cohetes de agua**Ley de la Conservación del Movimiento**

La ley de conservación del movimiento en cohetes de Agua Empezaré hablando sobre en qué consiste la ley de la conservación del movimiento , esta ley dice que que si la resultante de las fuerzas que actúan sobre un cuerpo o sistema es nula, su momento lineal permanece constante en el tiempo.

Ahora la comentare pero aplicándola en los cohetes de agua

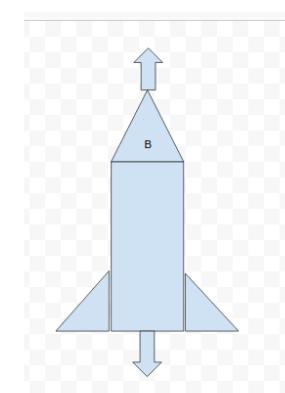
Tenemos un cohete de agua lleno de agua en el cual el agua y el cohete no tienen movimiento por lo tanto la cantidad de movimiento en el sistema es =



Instante Inicial, previo al lanzamiento

$$P(\text{Total}) = P(A) + P(B) = 0$$

Nosotros mediante una bomba de aire metemos presión en el cohete de agua al llegar a cierta presión deseada liberamos el cohete



Instante final , tras el lanzamiento

$$P(\text{Total}) = P(A)(-) + P(B)(+)$$

Al liberar el cohete , el agua (A) , sale disparada hacia abajo con una velocidad negativa (-) esto provoca que el cohete salga disparado hacia arriba con una velocidad (+) ya que la suma de ambas cantidades de movimiento tienen que seguir dando 0 que era la cantidad de movimiento inicial que es constante

La ley de conservación del movimiento de Newton dice , que , siendo dos cuerpos “a” y “b” que interactúan sin interferencias de fuera se cumple :

$$\begin{aligned} Ma \cdot Va + Mb \cdot Vb &= Ma \cdot V_a' + Mb \cdot V_b' \\ P(\text{antes}) &= P(\text{después}) \end{aligned}$$

¿A qué temperatura queda el aire dentro del cohete justo cuando se ha producido toda la expulsión del agua y se ha expulsado también todo el aire que puede salir de la botella? ¿A una temperatura mayor o menor que la del ambiente que rodea al cohete ? ¿Mucho mayor o mucho menor?

Para llegar a la respuesta hay que pensar primero a qué nos referimos exactamente cuando hablamos del término “Temperatura”, para esto el artículo correspondiente de la wikipedia puede ser un buen punto de partida

<https://es.wikipedia.org/wiki/Temperatura>

“ La temperatura de un gas ideal monoatómico es una medida relacionada con la energía cinética promedio de sus moléculas al moverse ”

También conviene tener en cuenta que pasa cuando las partículas se paran del todo, técnicamente se dice que alcanzaríamos el cero absoluto de temperatura (-273.15 °C) https://es.wikipedia.org/wiki/Cero_absoluto

Ahora realicemos el siguiente experimento mental:

Vivimos inmersos en un fluido que es el aire que está , digamos , a unos 20°C de temperatura porque las partículas del mismo se mueven con una energía cinética media que le da esa temperatura.

Pero si nos vamos al vacío del espacio interestelar , en una zona alejada de cualquier estrella y demás objetos celestes , donde apenas haya partículas ni fotones de luz ni nada, ¿que temperatura tendremos allí? Pues una muy cercana al cero absoluto. No llegaría al cero porque en realidad no existe el vacío “total” , siempre hay partículas, y en particular habrá fotones en el rango de longitud de ondas de las microondas que son el remanente del big bang y constituyen el “Fondo cósmico de Microondas” que da al vacío una temperatura mínima de 2.7 grados sobre el cero absoluto.

Luego , en nuestro cohete lo que estamos creando al expulsar el agua y el aire de manera violenta es un vacío, que hará que la temperatura interior disminuya en una cantidad apreciable.

Respuesta:

Resulta que al expandirse el gas de las botellas de los cohetes de agua se enfriá hasta los -75.9 grados , o más frío incluso con la condensación del agua que queda dentro.

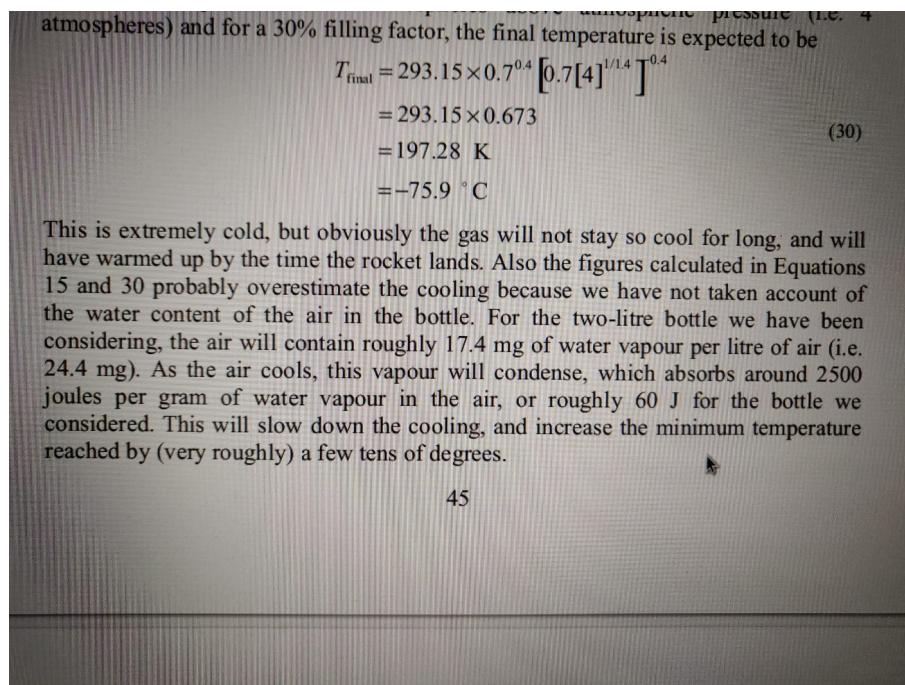
Desde cierto punto de vista estamos haciendo “mini frigoríficos super potentes”.

atmospheres) and for a 30% filling factor, the final temperature is expected to be

$$\begin{aligned}
 T_{\text{final}} &= 293.15 \times 0.7^{0.4} [0.7[4]^{1/14}]^{0.4} \\
 &= 293.15 \times 0.673 \\
 &= 197.28 \text{ K} \\
 &= -75.9^{\circ}\text{C}
 \end{aligned} \tag{30}$$

This is extremely cold, but obviously the gas will not stay so cool for long, and will have warmed up by the time the rocket lands. Also the figures calculated in Equations 15 and 30 probably overestimate the cooling because we have not taken account of the water content of the air in the bottle. For the two-litre bottle we have been considering, the air will contain roughly 17.4 mg of water vapour per litre of air (i.e. 24.4 mg). As the air cools, this vapour will condense, which absorbs around 2500 joules per gram of water vapour in the air, or roughly 60 J for the bottle we considered. This will slow down the cooling, and increase the minimum temperature reached by (very roughly) a few tens of degrees.

45



<https://drive.google.com/open?id=1oNneJDXINZGWtqz3Izku6aKj7-WRHVSk> (pag 42)

A propósito ¿Sabes como funciona un frigorífico?



"Cuando el líquido refrigerante atraviesa la válvula de expansión, disminuye su presión, pasando de un estado de más alta presión y temperatura a uno de menor presión y temperatura. Debido a este proceso, el líquido refrigerante se

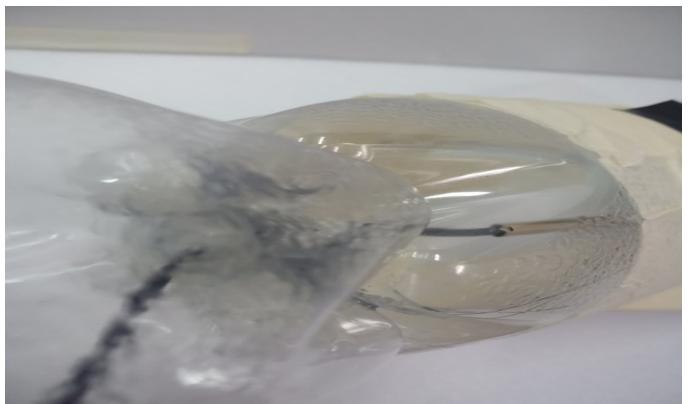
evaporará, y conseguirá reducir la temperatura del frigorífico. Es decir, el líquido refrigerante al entrar en el serpentín interior (el del frigorífico) se evaporará debido a la disminución de presión y al calor que recoge de los elementos del frigorífico.

Al salir del evaporador, el gas refrigerante (ya no es un líquido) se introduce en el compresor. Este dispositivo se encarga de aportar energía al gas, aumentando su presión (al contrario que la válvula de expansión) y su energía cinética, impulsándolo a fluir. Gracias a este aumento de presión, el gas refrigerante **se convierte de nuevo en líquido**, y al atravesar el serpentín exterior, cede su calor a la atmósfera a través de las paredes del tubo condensador.

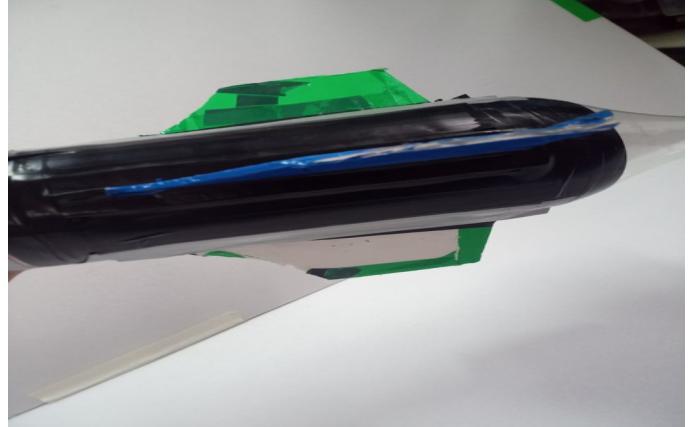
Este **ciclo se repite** constantemente hasta que el termostato de la orden de parada al compresor, momento en que el frigorífico habrá alcanzado la temperatura deseada y el líquido dejará de fluir por el sistema.”

Explicación tomada de Xataka Ciencia (<https://www.xatakaciencia.com/sabias-que/como-funciona-un-frigorifico-2>

Misión secundaria “extra”, fichas técnicas de los cohetes desarrollados

FICHA TÉCNICA		
Nombre	Lf-803el	Foto
Peso en vacío	91g	
Peso al Despegue	241g	
Envergadura	58cm.	
Número de etapas	1.	
Combustible	Agua y aire.	
Motor	Boquilla de botella.	
Presión de la cámara de expansión al despegue	8 bares.	
Materiales de construcción	Botellas recicladas, depron, adhesivos y pelota de corcho.	
Estado de construcción	En proceso.	
Sistema de Anclaje	Lanzadera One.	
Sistema de Paracaídas	No.	
Electrónica	Si.	
Altura Alcanzada (estimada)	90m.	
Comentarios técnicos	Este es un cohete de prueba de la temperatura de enfriamiento al expandirse el aire en la cámara de expansión. Tiene un sensor de temperatura, integrado en la cámara de expansión.	

FICHA TÉCNICA		
Nombre	At-890	Foto
Peso en vacío	215g	
Peso al Despegue	585g	
Envergadura	88cm.	
Número de etapas	1.	
Combustible	Agua y aire.	
Motor	Boquilla de botella.	
Presión de la cámara de expansión al despegue	6 bares.	
Materiales de construcción	Botellas recicladas, depron y cartón pluma, recámara de bicicleta reciclada,gomas elásticas,paracaídas de plástico,cuerda, punta de corcho,adhesivos y pelota de corcho.	
Estado de construcción	En proceso.	
Sistema de Anclaje	Lanzadera Zero.	
Sistema de Paracaídas	Si.	
Electrónica	Si	
Altura Alcanzada (estimada)	70m.	
Comentarios técnicos	Este cohete es de testeo múltiple. Tiene un novedoso sistema de almacenaje de electrónica "ElectroSpaceSix", además de que tiene el sistema americano de retención de paracaídas y las alas, que permiten que el cohete suba casi sin rozamiento con el aire.	

FICHA TÉCNICA		
Nombre	Sm-54	Foto
Peso en vacío	107g	
Peso al Despegue	407g	
Envergadura	48cm	
Número de etapas	1	
Combustible	Agua y aire.	
Motor	Boquilla de botella.	
Presión de la cámara de expansión al despegue	4 bares.	
Materiales de construcción	Botella reciclada, depron, recámara de bicicleta reciclada y adhesivos.	
Estado de construcción	Acabado.	
Sistema de Anclaje	Lanzadera Zero.	
Sistema de Paracaídas	Ninguno	
Electrónica	Ninguna	
Altura Alcanzada (estimada)	50m	
Comentarios técnicos	Es un cohete de prueba para pruebas de lanzadera Zero.	

FICHA TÉCNICA		
Nombre	Robin-470a	Foto
Peso en vacío	434g	
Peso al Despegue	934g	
Envergadura	130cm.	
Número de etapas	3.	
Combustible	Agua y aire.	
Motor	Boquilla de botella.	
Presión de la cámara de expansión al despegue	6 bares.	
Materiales de construcción	Botellas recicladas, depron, paracaídas de tela, servo, goma elástica, adhesivos y pelota de corcho.	
Estado de construcción	Acabado.	
Sistema de Anclaje	Lanzadera One.	
Sistema de Paracaídas	Si.	
Electrónica	Si.	
Altura Alcanzada (estimada)	50m.	
Comentarios técnicos	<p>Este es un cohete de testeо, donde va a ir el cansat, cumpliendo la misión principal, que mide la altura y la presión y la misión secundaria, que mide la radiación de rayos ultravioletas.</p> <p>Cuenta con alas en dos secciones que hace que sea más aerodinámico y nos hemos basado en el diseño aerodinámico de misiles militares.</p>	

Paracaídas para nuestros cohetes, la misión secundaria “Extra”

Hemos realizado dos sistemas distintos de expulsión del paracaídas de los cuales los dos funcionan correctamente y con clara precisión, a uno le hemos llamado sistema americano y al otro sistema europeo.

Sistema Americano.

Para este sistema nos hemos basado en esta página de internet, que la verdad tienen unas ideas magnificas:

http://www.uswaterrockets.com/construction_&_tutorials/Radial_Deploy/_tutorial.htm



Este es el sistema que hemos llamado sistema americano, como se puede ver son dos botellas unidas y en medio un paracaídas rodeado por un trozo de botella atado a una goma elástica que se enrolla alrededor del trozo de botella, claro está que debe estar tenso.



La goma se desenreda por la tensión haciendo que el trozo de botella se separe del cohete liberando el paracaídas, que cae solo debido al empuje del trozo de botella.



Así quedaría el sistema tras su apertura, con el trozo de botella separado y el el paracaídas desplegado.

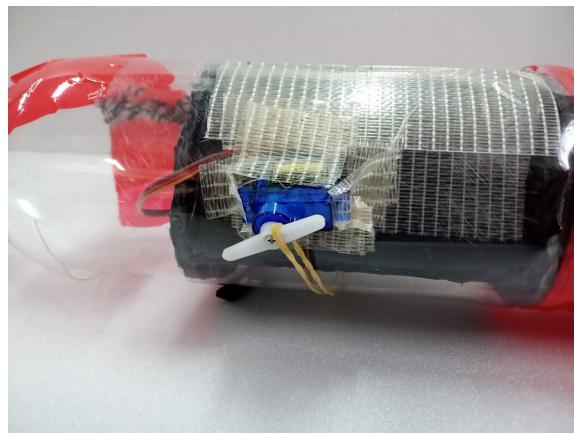
Sistema Europeo.

Para este sistema nos hemos basado en un vídeo de YouTube:

<https://www.youtube.com/watch?v=ieyiWYC9-yE>



Este es el sistema que hemos llamado sistema europeo que se usa un trozo de botella como muelle y otro trozo como compuerta con una goma elástica atada a esta puerta, el soporte donde va el muelle está hecho de depron (Cartón pluma).



Este servo sirve para abrir la puerta del paracaídas la cual como dijimos con anterioridad está atada a una goma y esta a la vez se engancha al servo como se ve en la imagen, este último siendo activado por la electrónica.



Una vez el servo se activa, la puerta se abre, el trozo de botella que actúa como muelle expulsa el paracaídas y este sale disparado desplegándose.



Así quedaría el sistema tras su apertura, con la compuerta abierta, el muelle expandido y el paracaídas desplegado.

Sistemas de almacenamiento de electrónica en la misión secundaria “Extra” de los cohetes de Agua

Hemos realizado por ahora dos sistemas de almacenaje de la electrónica en los cohetes, que hasta hoy no dan fallos y la electrónica tiene espacio para entrar correctamente, a uno lo hemos llamado ElectroSpaceSix ya que lo hemos inventado nosotros y a la otra forma de almacenamiento la hemos llamado sistema clásico .

ElectroSpaceSix.



Este es un sistema nuevo llamado ElectroSpaceSix para conseguir introducir la electrónica.



En esta imagen podemos ver claramente como funciona este nuevo sistema de almacenamiento de electrónica, le hemos anclado una goma elástica a los

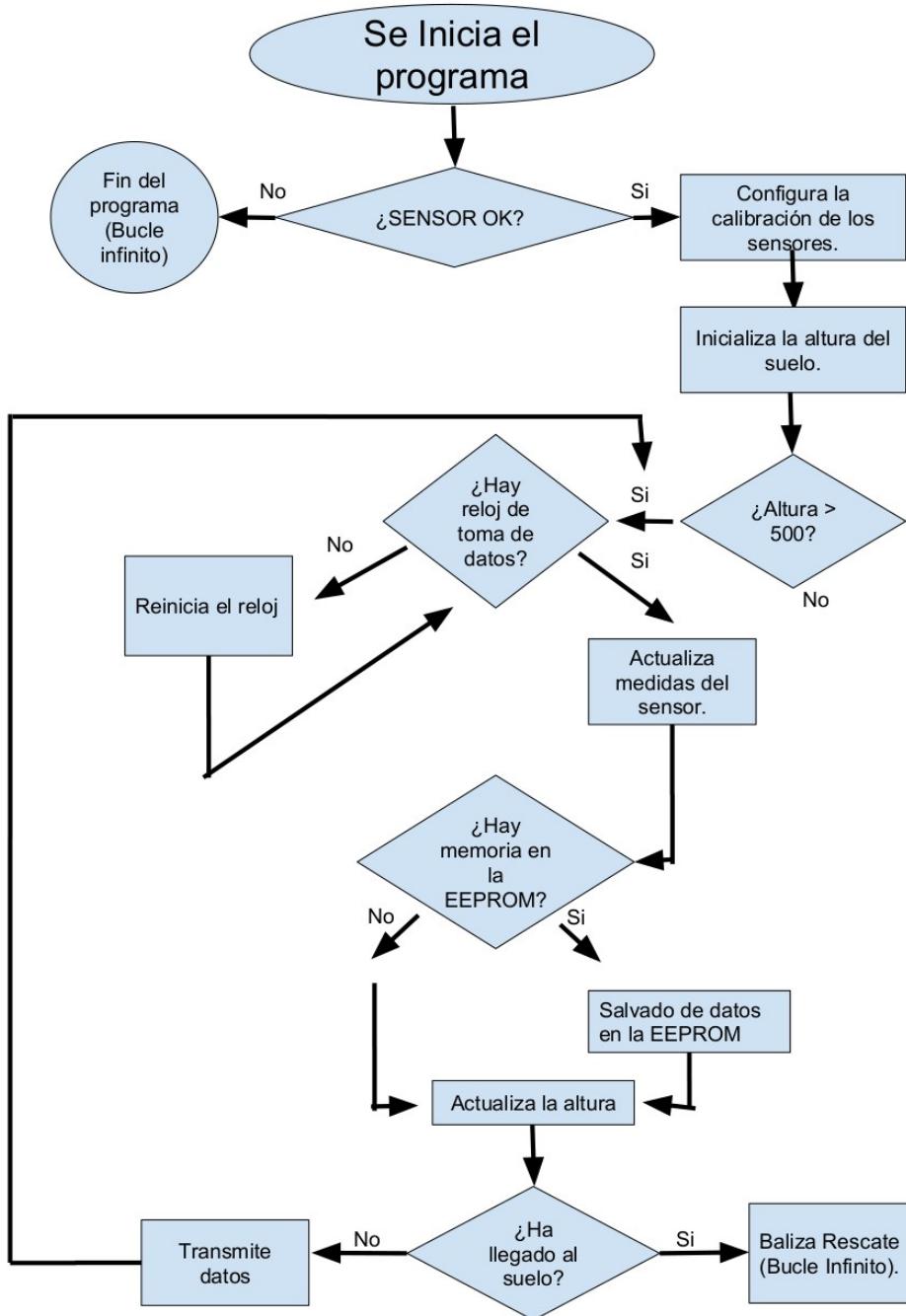
tapones de dos botellas distintas, para cuando estiremos la cabeza del cohete, esta vuelva a la normalidad y se cierre de forma correcta.

De esta manera conseguimos un acceso limpio y espacioso a la cavidad donde va la electrónica y de manera muy elegante y natural queda cerrado el compartimento por la tensión de la goma, una vez introducida la electrónica.

Pt: Esta forma de almacenamiento de electrónica fué inventada por nuestro equipo SpaceSix.

Programación. Lenguaje de Programación y diagrama de flujo del programa.

Diagrama de Fluzo



Capturas de pantalla del código del programa

```

Cansat_beta_v2
/*
Programa para el Cansat del equipo SpaceSix
Versión Beta v2

Versión especialmente preparada para trabajar tanto en el cohete oficial Cansat como con cohetes de agua

En esta versión se añade:

- Se añaden condicionales para marcar si estamos en depuración o si estamos en un lanzamiento de cohete de agua o no
- Inicialización del proceso de lanzamiento mediante señal de radiofrecuencia enviada por la estación base
- Se añade un servo de liberación del paracaidas del cohete de agua
*/

/*=====
//      SECCION PARA IMPORTACION DE LIBRERIAS
=====*/
#include <Wire.h>          // Utilidades para comunicaciones I2C
#include <i2c_BMP280.h>      // Biblioteca de funciones para el barometro
#include <EEPROM.h>
#include <Adafruit_VEML6070.h>
#include <SoftwareSerial.h>
#include <Temporizador_inopya.h>
#include <Universal_GPS_inopya.h>
#include <Servo.h>

/*=====
//      SECCION DE DEFINICION DE PINES
=====*/
#define PIN_PULSADOR    2
#define GPS_RX          3
#define GPS_TX          4
#define UV_1             5
#define UV_2             6

```



```

Cansat_beta_v2
/*
===== SECCION DE DEFINICION DE PINES =====
=====

#define PIN_PULSADOR    2
#define GPS_RX          3
#define GPS_TX          4
#define UV_1             5
#define UV_2             6
#define UV_3             7
#define PIN_ALTAZOZ     9
#define ANTENA_RX        10
#define ANTENA_TX        11
#define PIN_SERVO        12

===== SECCION DE DEFINICION DE CONSTANTES DEL PROGRAMA =====
=====

#define ALTURA PARA EMPEZAR A MEDIR 500 //Se empezarán a tomar medidas a los 500 metros
#define INTERVALO_MUESTRAS   1000 // tiempo en milisegundos entre cada muestra
// Recordad que disponemos de memoria para 204 muestras,
// dependiendo del intervalo, tendremos mayor o menor tiempo de grabacion

#define EN_DEPURACION     1 // Si se pone a 1 (true) es que estamos en depuración y a 0 (false) es que no

===== SECCION DE CREACION DE OBJETOS =====
=====

/* Creacion del objeto servo */
Servo servo;

/* creación de un puerto serie para el GPS */

```



```
Cansat_beta_v2
//*****
//      FUNCION DE CONFIGURACION
//*****
//*****



void setup()
{
    servo.attach(PIN_SERVO);
    pinMode(PIN_PULSADOR, INPUT);
    pinMode(UV_1, OUTPUT);
    pinMode(UV_2, OUTPUT);
    pinMode(UV_3, OUTPUT);
    pinMode(PIN_ALTAVOZ, OUTPUT);

    /* inicializamos el puerto serie para el PC (pruebas) */
    if(EN_DEPURACION){
        Serial.begin(9600);
    }

    /* Pone el servo en posicion cerrada de 0 grados para asegurar el paracaidas */
    servo.write(0);

    /* inicializamos el puerto serie para el emisor RF */
    radioLink.begin(9600);

    /* inicializamos el puerto serie para el GPS */
    NEO_gps.begin(9600); //iniciamos el GPS a la velocidad standard
    NEO_gps.set_mode(0);

    /* cortamos la alimentacion a todos los sensores UV */
    for(uint8_t n=0;n<3;n++){
        digitalWrite(uvSensorList[n], LOW);
    }
}
```

```
Cansat_beta_v2
NEO_gps.set_mode(0);

/* cortamos la alimentacion a todos los sensores UV */
for(uint8_t n=0;n<3;n++){
    digitalWrite(uvSensorList[n], LOW);
}

/* inicializamos el barometro */
if (bmp280.initialize()){
    if(EN_DEPURACION){
        Serial.println(F("Sensor ok"));
    }
}
else{
    if(EN_DEPURACION){
        Serial.println(F("Fallo de Sensor"));
    }
    while (true) {} //si no hay barmetro el programa queda en bucle infinito
}

/* Configuracion de calibracion */
bmp280.setPressureOversampleRatio(8); //2
bmp280.setTemperatureOversampleRatio(1); //4
bmp280.setFilterRatio(4); //4
bmp280.setStandby(0); // 0=desactivado, por tanto el sensor esta activo.

/* Medidas bajo demanda */
bmp280.setEnabled(0); //0-Desactivamos el modo automatico. Solo obtendremos respuesta
//del sensor tras una peticion con 'triggerMeasurement()'


digitalWrite(uvSensorList[0], HIGH); //alimentamos uno de los sensores Uv
delay(10); //pausa para estabilizar la alimentacion
sensor_UV6070.begin(VEML6070_1_T); //reiniciamos el sensor (si no no funciona correctamente)
```

```

Cansat_beta_v2
delay(10);
sensor_UV6070.begin(VEML6070_1_T); //pausa para estabilizar la alimentacion
//reiniciamos el sensor (si no no funciona correctamente)

delay(2000); // segundos para dar tiempo a reprogramar la placa cuando sea necesario (Necesario en arduino Micro)

/* El programa en cualquier momento estará atento al estado del pulsador */
attachInterrupt(digitalPinToInterrupt(PIN_PULSADOR), atenderInterrupcion, RISING);

/* Permite el uso de un pulsador para iniciar la toma de muestras (util en pruebas) */
if(EN_DEPURACION){
  // El programa espera en esta linea hasta que sea pulsado el pulsador
  while(FLAG_estado_pulsador == false){}
  FLAG_pararZumbador = false;
  Serial.println("PULSADOR ACTIVADO");
}

/* Detección AUTOMATICA de cohete en el aire e inicio de toma de muestras */
medirAlturaYTemperatura();
altura_suelo = altura;

//Esta parte espera a que el cohete despegue y alcance una altura de más de ALTURA_PARA_EMPEZAR_A_MEDIR metros .
//Se queda en la espera de que la altura menos la altura del suelo sea mayor de ALTURA_PARA_EMPEZAR_A_MEDIR .
while((altura - altura_suelo) < ALTURA_PARA_EMPEZAR_A_MEDIR) {

  //Si recibimos por radio la orden de lanzamiento salimos de aquí para activar el protocolo de toma de medidas
  if(ordenLanzamientoPorRadio){
    break;
  }
  atenderPuertoSerie(); //para poder recuperar los datos de la eeprom
  medirAlturaYTemperatura();
  delay(1000);
  if(EN_DEPURACION){
    Serial.println("esperando a ser lanzado");
  }
}

```



```

Cansat_beta_v2
medirAlturaYTemperatura();
delay(1000);
if(EN_DEPURACION){
  Serial.println("esperando a ser lanzado");
}

FLAG_uso_eeprom = true;
relojControlSuelo.begin(15000);

if(EN_DEPURACION){
  Serial.println("iniciado el lanzamiento");
}
FLAG_uso_eeprom = false;
}

////////////////////////////////////////////////////////////////
// BUCLE PRINCIPAL DEL PROGRAMA
////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////

void loop()
{
  /* control de tiempo para tareas de toma de datos y transmisión a tierra */
  if(relojMuestras.estado()== false){
    relojMuestras.begin(1000); //Reinicio del reloj de toma de muestras y otras tareas.

    /* Actualizar datos de altura y temperatura */
    medirAlturaYTemperatura();

    /* Actualizar el índice UV */
    indiceUV = obtener_UV_max();
}

```

Cansat_beta_v2

```

void loop()
{
    /* control de tiempo para tareas de toma de datos y transmision a tierra */
    if(relojMuestras.estado()== false){
        relojMuestras.begin(1000); //Reinicio del reloj de toma de muestras y otras tareas.

        /* Actualizar datos de altura y temperatura */
        medirAlturaYTemperatura();

        /* Actualizar el indice UV */
        indiceUV = obtener_UV_max();

        /* Salvado de datos (MIENTRAS QUEDE MEMORIA) -- */
        if(FLAGS_uso_eeprom==true){
            /* altura con dos decimales, como un entero */
            int altura_int = int(altura*100);
            /* temperatura con dos decimales, como un entero */
            int temperatura_int = int(temperatura*100);
            /* indice UV con 1 decimal, como un entero */
            uint8_t indiceUV_int = indiceUV*10;

            /* empaquetado de los datos de interes en un 'struct' */
            CanSatDatos datos_actuales = { altura_int, temperatura_int, indiceUV_int };
            saveData(puntero_eeprom, datos_actuales);           //enviamos un dato del tipo 'CanSatDatos'
            puntero_eeprom+=5;                                //incrementamos el puntero para escritura
            /* Si llenamos la eeprom, dejamos de grabar y desactivamos los permisos de acceso*/
            if(puntero_eeprom > 1020 || puntero_eeprom < 5){
                FLAG_uso_eeprom = false; // bloqueo de acceso para evitar sobreescribir
            }
        }

        /* Actualizamos el dato de altura maxima alcanzada, por si lo usamos para algo */
        if (altura > altura_maxima){
            altura_maxima = altura;
        }
    }
}

```

Cansat_beta_v2

```

if (altura > altura_maxima){
    altura_maxima = altura;
}

/* Si la altura registrada es menor que la altura_maxima mas un intervalo de seguridad de 5 metros
 * activamos el paracaidas
 */
if (altura + 5 < altura_maxima){
    /* Pone el servo en posicion abierta de 180 grados para abrir el paracaidas */
    servo.write(180);
}

/* Controlar la llegada al suelo */
if (abs(altura - altura_anterior)<1){
    contador_suelo++;
    if(contador_suelo>5){ //Cambiado por Iona para estar más seguro de que está en el suelo.
        //llegada al suelo!! emitir los datos de GPS
        baliza_Rescate();
    }
}

/* Transmitir datos a tierra */
radioLink.print(altura);
radioLink.print(";");
radioLink.print(temperatura);
radioLink.print(";");
radioLink.println(indiceUV);

/* mostar datos por serial (solo apra pruebas, eliminar del programa final */
if(EN_DEPURACION){
    Serial.print(altura_anterior);
    Serial.print(";");
    Serial.print(altura);
    Serial.print(";");
    Serial.print(temperatura);
}

```

```
Cursat_beta_v2

if(EN_DEPURACION){
    Serial.print(altura_anterior);
    Serial.print("'''");
    Serial.print(altura);
    Serial.print("'''");
    Serial.print(temperatura);
    Serial.print("'''");
    Serial.println(indiceUV);
}
}

/* control de alitud para detectar llegada al suelo */
if(relojControlsuelo.estado()== false){
    relojControlsuelo.begin(10000);
    altura_anterior = altura;
}
}

/*
#####
##### BLOQUE DE FUNCIONES: LECTURAS DE SENsoRES, COMUNICACIONES...
#####
*/
// ACTUALIZAR DATOS DE ALTITUD Y TEMPERATURA
//



void medirAlturaYTemperatura()
{
    bmp280.triggerMeasurement();      //peticion de nueva medicion
    bmp280.awaitMeasurement();        //esperar a que el sensor termine la medicion
    bmp280.getAltitude(altura);
    bmp280.getTemperature(temperatura);
```

```

/*
 * Cansat beta v2
 */

// BLOQUE DE FUNCIONES: LECTURAS DE SENSORES, COMUNICACIONES...
// ACTUALIZAR DATOS DE ATITUD Y TEMPERATURA
// EEPROM

void medirAlturaYTemperatura()
{
    bmp280.triggerMeasurement();      //peticion de nueva medicion
    bmp280.awaitMeasurement();        //esperar a que el sensor termine la medicion
    bmp280.getAltitude(altura);
    bmp280.getTemperature(temperatura);
}

// EEPROM

=====

// LEER DATOS ALMACENADOS EN EEPROM
=====

CanSatDatos loadData(int posicion)
{
    CanSatDatos muestra;
    EEPROM.get(posicion, muestra);
    return muestra;
}

// SALVAR DATOS EN LA EEPROM

```

```
Cansat_beta_v2

//=====
// SALVAR DATOS EN LA EEPROM
//=====

void saveData(int posicion, CanSatDatos muestra)
{
    EEPROM.put(posicion, muestra);
}

//=====
// LISTAR EL CONTENIDO DE LA EEPROM
//=====

void listar_datos()
{
    int puntero_lectura = 0;
    float contador = 0;

    Serial.println();
    Serial.println(F("orden, Altura (m), Temperatura (C), indice UV"));

    while(puntero_lectura < 1020){
        /* recuperar datos de la eeprom */
        CanSatDatos dato_leido;
        EEPROM.get(puntero_lectura, dato_leido);

        /* tratar los datos recuperados */
        float altura_float = float(dato_leido.altura)/100.0;
        float temperatura_float = float(dato_leido.temperatura)/100.0;
        float indiceUV_float = float(dato_leido.uv)/10.0;

        /* mostar datos por puerto serie */
        Serial.print(contador++); Serial.print(F(","));
        Serial.print(altura_float); Serial.print(F(","));

        /* mostar datos por puerto serie */
        Serial.print(contador++); Serial.print(F(","));
        Serial.print(temperatura_float); Serial.print(F(","));
        Serial.print(indiceUV_float);

        /* incrementar el puntero de lectura de la eeprom */
        puntero_lectura +=5;
    }
}

/*=====
// PUERTO SERIE
//=====
// FUNCION PARA LECTURA DE CARACTERES POR PUERTO SERIE
//=====

int atenderPuertoSerie()
{
    char orden_recibida = ' ';
    while(Serial.available()) {
        orden_recibida = Serial.read();

        /* listar datos de eeprom */
        if(orden_recibida == 'L' or orden_recibida == 'l'){
            Serial.flush();
            listar_datos();
        }
    }
    return 0;
}
```

```

Cansat_beta_v2
return 0;
}

/* ACTIVACION POR RADIO ENLACE
// FUNCION PARA LECTURA DE CARACTERES POR RADIO ENLACE
=====

int ordenLanzamientoPorRadio()
{
    char orden_recibida = ' ';
    while(radiolink.available()){
        orden_recibida = radiolink.read();
        if(orden_recibida == 'I' or orden_recibida == 'L'){
            return 1;
        }
    }
    return 0;
}

// ATENDER INTERRUPCIONES DEL PUSADOR
//=====

void atenderInterrupcion()
{
    FLAG_estado_pulsador = !FLAG_estado_pulsador;
    delay(500); //Para evitar rebotes
    FLAG_pararZumbador = true;
}

/* LOCALIZACION DURANTE EL RESCATE
//=====

void baliza_Rescate()
{
    while(true){
        /* Generar tono para localizacion */
        if (FLAG_pararZumbador == false) { //BORRAR EL IF.
            tone(PIN_ALTAZO, 2100); //frecuencia que emite un sonido bastante estridente
            delay (450);
            noTone(PIN_ALTAZO);
        }

        /* obtener datos del GPS */
        NEO_gps.get();

        /* Transmitir datos de POSICION GPS */
        radiolink.print(NEO_gps.longitud,6);
        radiolink.print(F(" , "));
        radiolink.println(NEO_gps.latitud,6);
        radiolink.print(F(" , "));

        /* Transmitir datos de HORA GPS */
        if(NEO_gps.hora < 10){ radiolink.print(F("0")); }
        radiolink.print(NEO_gps.hora); radiolink.print(F(":"));
        if(NEO_gps.minuto < 10{ radiolink.print(F("0")); }
        radiolink.print(NEO_gps.minuto); radiolink.print(F(":"));
        if(NEO_gps.segundo < 10{ radiolink.print(F("0")); }
        radiolink.print(NEO_gps.segundo);
        radiolink.print(F(" , "));

        /* Transmitir datos de FECHA GPS */
        if(NEO_gps.dia < 10{ radiolink.print(F("0")); }
        radiolink.print(NEO_gps.dia); radiolink.print(F("/"));
        if(NEO_gps.mes < 10{ radiolink.print(F("0")); }
        radiolink.print(NEO_gps.mes); radiolink.print(F("/"));
}

```

```
Cansat_beta_v2

/*
// OBTENER INDICE UV
// */

float obtener_UV_max()
{
    float indice_UV_max = 0.0; //reseteamos el indice uv maximo

    for(uint8_t j=0;j<3;j++){
        digitalWrite(uvSensorList[j], LOW); //cortamos la alimentacion a todos los sensores
    }
    delay(10);

    for(uint8_t n=0;n<3;n++){

        digitalWrite(uvSensorList[n], HIGH); //alimentamos los sensores secuencialmente
        delay(10); //pausa para estabilizar la alimentacion
        sensor_UV6070.begin(VEML6070_1_T); //reiniciamos el sensor (si no no funciona correctamente)
        int lecturaUV_RAW = sensor_UV6070.readUV(); //obtener el valor de luz ultravioleta
        float indice_UV = lecturaUV_RAW/280.0; //procesamos para obtener indice de radiacion UV

        digitalWrite(uvSensorList[n], LOW);
        if (indice_UV > indice_UV_max){
            indice_UV_max = indice_UV;
        }

        // !!! eliminar todos los print del programa final !!!! 
        Serial.print(F("Sensor UV(")); Serial.print(n);
        Serial.print(F(") RAW:")); Serial.print(lecturaUV_RAW);
        Serial.print(F(" , UV index:")); Serial.println(indice_UV,2);
    }
    Serial.print(F("UV max. = ")); Serial.println(indice_UV_max); Serial.println();
    return indice_UV_max;
}

.
```

Análisis de Datos de telemetría del Cansat con GeoGebra



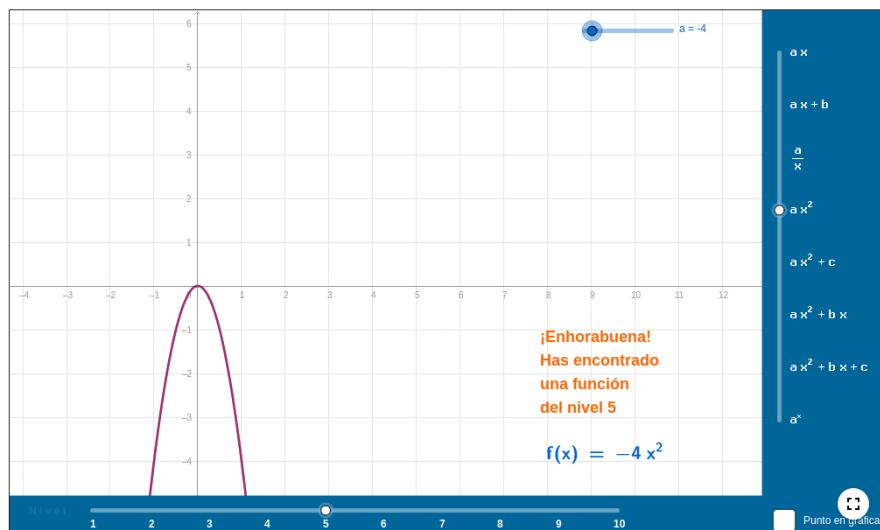
¿Porque utilizar GeoGebra como herramienta principal de análisis de los datos entregados por el Cansat ?

- Es una herramienta gratuita y fácilmente accesible desde la web
- Se puede usar directamente online sin instalarla
- Muy fácil de aprender y con gran cantidad de recursos y tutoriales disponibles en castellano
- Cada vez se está estandarizando más su uso como herramienta curricular en las asignaturas de ciencias en el temario de la ESO y Bachillerato, por lo que todo lo que aprendamos en ella con nuestra experiencia en Cansat nos será de utilidad para aplicarlo en dichas asignaturas.

Introducción a los conceptos matemáticos y físicos

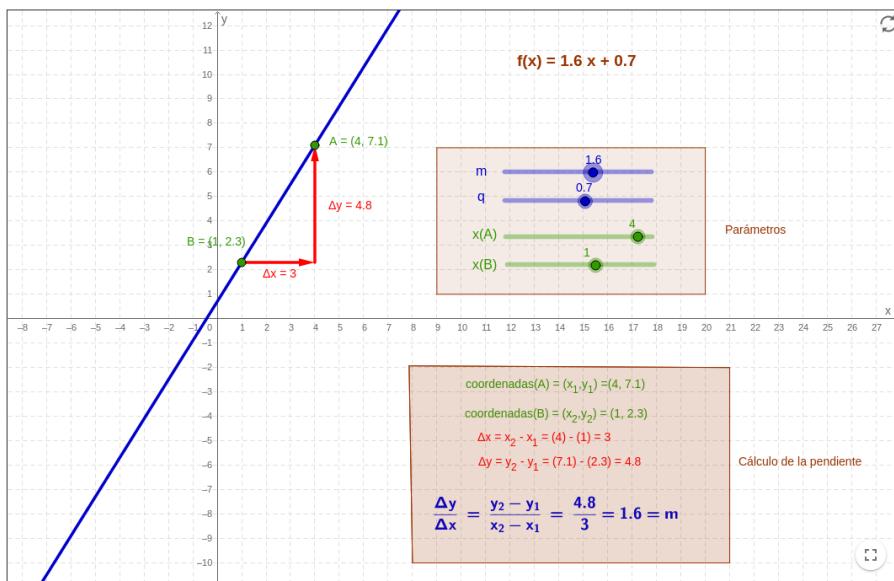
1. Concepto matemático de gráfica y función
 - Ejemplos de funciones elementales
<https://www.geogebra.org/m/vhSQ1Hhg#material/fUz5wACv>
 - El Juego de las Funciones <https://www.geogebra.org/m/fNwyEGKY>

- Ahora realizaremos el siguiente ejercicio, vamos a intentar inferir cual de las funciones básicas que hemos visto se corresponden con la caída libre de un objeto dejado caer desde cierta altura.
 - Comienza haciendo unos ejes de coordenadas con el eje X representando el tiempo y el eje Y representando la altura.
 - Ahora, intenta inferir cual será la grafica aproximada de este movimiento
 - Contrasta tus impresiones con la realidad:
Visor de caída libre <https://www.geogebra.org/m/ztarftge>
 - Ahora intenta hacer una grafica de la misma situación con el eje X como el tiempo , pero ahora el eje Y como la velocidad ¿Que forma tendría esta gráfica?

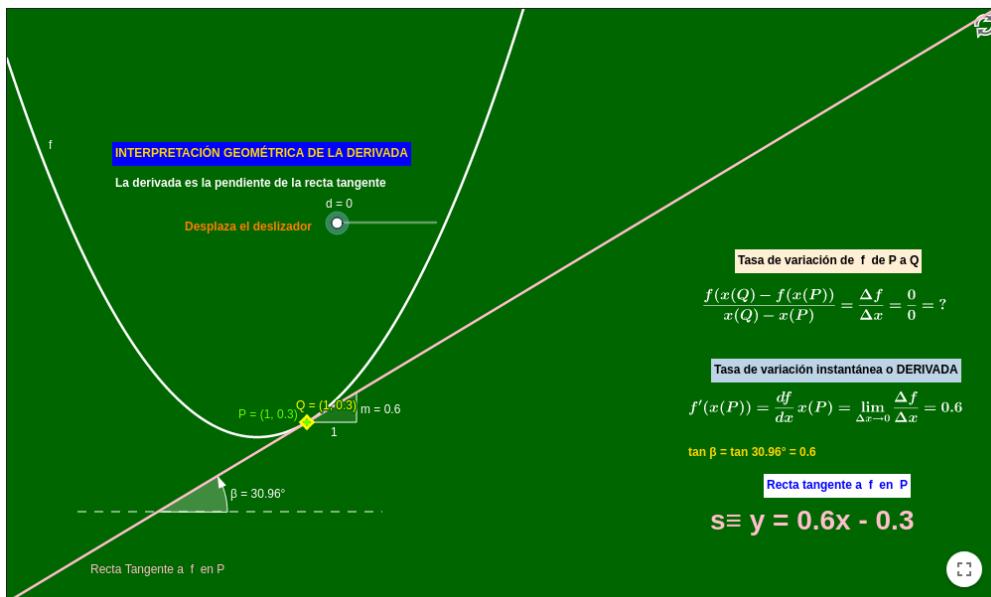


2. Ahora vamos a desviarnos un poco para hablar del concepto trigonométrico de tangente , que nos puede servir para medir lo inclinada que esta la pendiente
 - Mira en la siguiente actividad <https://www.geogebra.org/m/QjkfFJwc> como al cambiar la inclinación de cuesta representada por el triangulo al aumentar el ángulo agudo del mismo la tangente aumenta
 - Observa también que , si el angulo no cambia, la tangente permanece constante al hacer mas grande el triangulo ¿Cómo es esto posible?
 - Una última pregunta sobre la tangente, ¿Cuanto crees que valdrá cuando el ángulo sea de 90º?

3. Al representar la velocidad de caída de un objeto respecto al tiempo vimos que su función era una linea que se puede representar como “ $y = mx + n$ ” , pues bien, resulta que “ m ” es el valor de la pendiente de dicha linea tal y como se puede ver en <https://www.geogebra.org/m/SNMwQ9wW>



4. Pues bien, resulta que dada una gráfica podemos calcular el crecimiento de la misma en un punto hallando la pendiente de la recta tangente a la gráfica en ese punto , esto se llama técnicamente la Derivada
- Tal y como se puede ver aquí <https://www.geogebra.org/m/ZZVNDp8t>
 - La Derivada de un movimiento es precisamente la velocidad del mismo , para ello se utiliza el movimiento de caida libre que ya sabes que da una gráfica parabólica y una velocidad lineal como puedes comprobar aquí <https://www.geogebra.org/m/q9HwzhdC>
 - Aquí <https://www.geogebra.org/m/mrUUA5aG> puedes ver ejemplos de derivadas de distintas funciones, observa que cuanto menos crece la función menor es la derivada

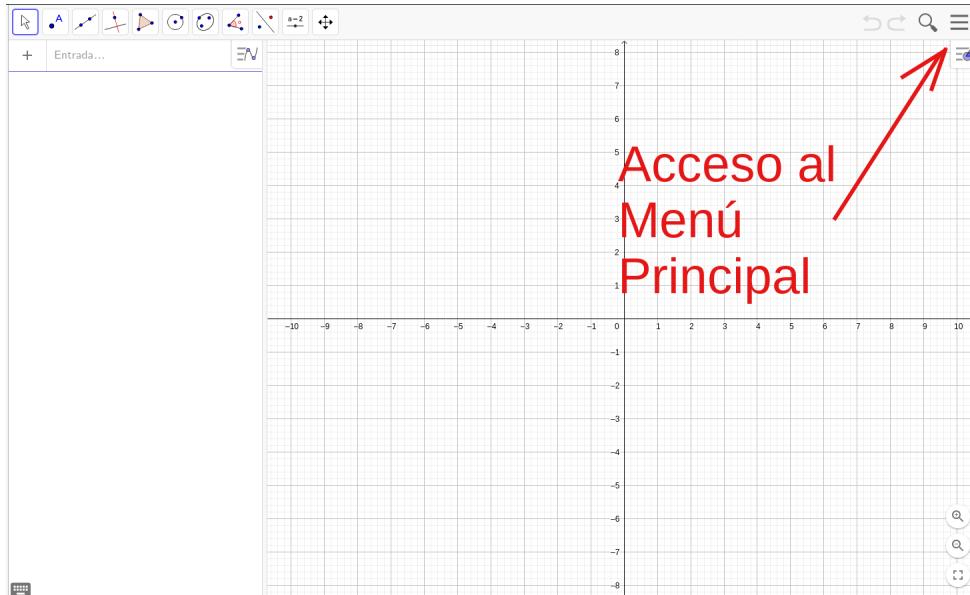


5. Para estudiar el movimiento de caída del satélite Cansat , en principio no se dispone de una gráfica continua , sino de una serie de puntos aislados que hay que conectar de alguna manera para poder tener una grafica más o menos suave a la que poder sacarle la derivada y por tanto la velocidad de descenso. Para ello contamos con la herramienta de GeoGebra de ajustePolinomial que usa la característica que tienen los polinomios de poder adaptarse a estas nubes de puntos sin mas que cambiar sus coeficientes y añadirles grados a los mismos tal y como se puede ver aquí <https://www.geogebra.org/m/kfsc6jnn>

Análisis de los datos propiamente dicho

6. Recopilar los datos del vuelo Guardados en la EEPROM con el monitor serial (Introducir “L” para “listar” los datos)
7. Crear un documento de texto con extensión “.csv” y pegar allí los datos
8. Abrir ese documento csv con el programa LibreOffice Calc, que descompondrá los datos en dos columnas
9. Abrir la web de GeoGebra online <https://www.geogebra.org/classic>

10. Breve explicación sobre un uso básico de GeoGebra, acceso a los diferentes menús y vistas



11. Seleccionar la Vista “Hoja de Cálculo” , y copiar en ella las dos columnas del punto 8

12. Seleccionar ambas columnas y en el menú que se despliega al pulsar el botón derecho sobre ellas selecciona “Crear Lista de Puntos”

13. Para ver la lista de puntos creada hay que activar la vista “Algebraica” , además ya podemos cerrar la vista de “Hoja de Cálculo” para tener mas espacio.

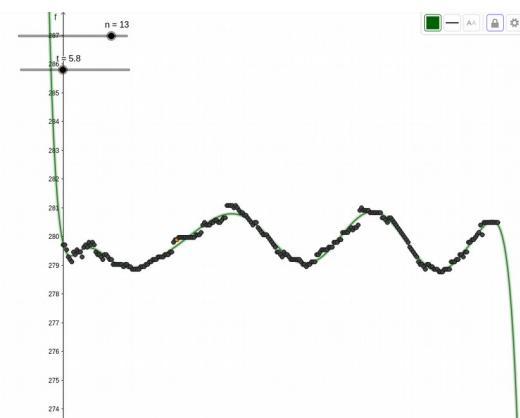
14. Ademas hay que activar la “Barra de Entrada” en el menú principal de GeoGebra

15. Ahora tenemos que crear dos deslizables,

- Uno para controlar el grado de aproximación del polinomio y al que llamaremos “n” y que tiene que ser un entero entre 0 y 15
- Y otro para controlar el punto desde el que calculamos la tangente al que llamaremos “t” que tiene que ser un número entre 0 y 15

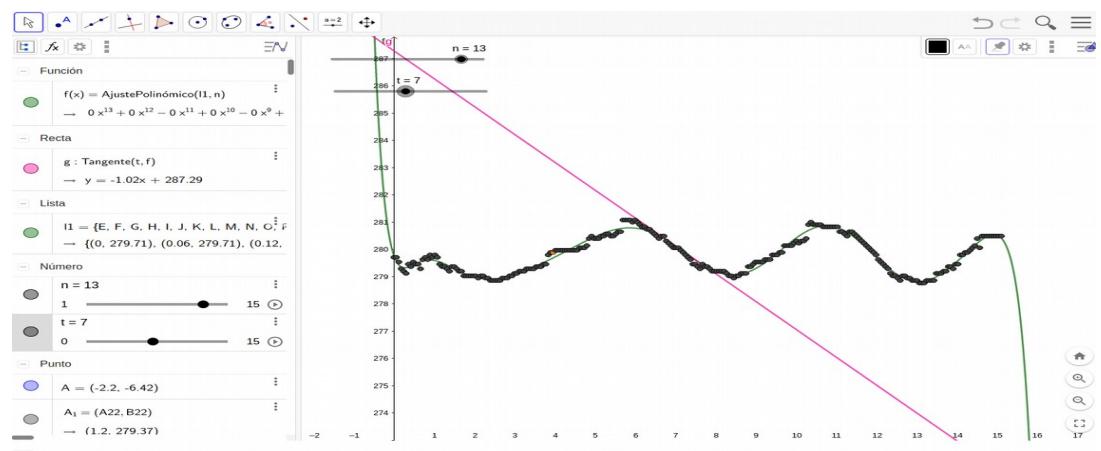
16. Tras esto , en la barra de entrada tecleamos “

AjustePolinómico(listaPuntos, n)” y una vez le demos al intro podremos ir deslizando el valor de n para ir viendo el polinomio que mejor se ajusta a nuestros puntos



17. Luego, en la barra de entrada tecleamos “Tangente(t, f)” siendo “f” el nombre de la función creada por el ajuste polinómico y que habrá que cambiar en consecuencia si el ajuste nos ha dado otro nombre

18. Ahora , moviendo el deslizable t podemos ir viendo como varía la tangente del ajuste polinómico por los distintos puntos y en la vista algebraica saldrá e irá variando la fórmula de esta tangente según nos desplazamos y de ahí podemos coger el valor de la velocidad. Por ejemplo , en la siguiente imagen , donde la tangente en $t = 7$ aparece como una recta rosa con fórmula $y= -1.02x + 287.29$, el valor de la derivada , y por tanto la velocidad en ese punto es de -1.02



Presupuesto del proyecto

Teniendo en cuenta el número total de prototipos desarrollados el presupuesto final ha aumentado en 78.5€ respecto a nuestra estimación inicial de 103.50€

Concepto	Adquirido en	Gasto (En €)
4x Bluetooth Nano para Arduino Ble Nano V3.0	Amazon	$12 * 4 = 48$
4x BMP280 Sensor de presión barométrica	Aliexpress	$4 * 0.50 = 2$
Módulo GPS NEO-8M	Aliexpress	5
cargador de batería	Aliexpress	2
GY-VEML6070 sensor de luz UV (3 Unidades)	Aliexpress	8
3x APC220 módulos de datos en serie inalámbricos RF	Aliexpress	$16 \times 3 = 48$
Tela impermeable Ripstop	Amazon	14
Brotree Paracord 2mm Cuerda de Nailon Paracaídas	Amazon	9
Gasto estimado en la impresión de la estructura del Cansat en 3D , hardware y cableado electrónico	Varios	30
Compensacion CO2	CeroCO2	6
	Total	181,50 €

SOSTENIBILIDAD

En Space Six como toda nuestra sociedad estamos comprometidos con el medio ambiente y hemos querido que nuestro proyecto fuera lo mas sostenible posible, dentro de los parámetros internacionalmente consensuados sobre la sostenibilidad

Económico:

Hemos querido implementar el “factor 4” en nuestro proyecto, o lo que es lo mismo hacer lo mismo con la cuarta parte del presupuesto o gasto habitual establecido. Creo que nuestro presupuesto, y sobre todo la ausencia de caros dispositivos para las pruebas y la utilización de recursos compartidos explica por si solo esta cuestión

Social:

Si bien tenemos una carencia de genero al no haber podido incluir mujeres en nuestro equipo, en parte producida por la desconfianza de sus padres, al no ser nuestra entidad dentro de ningún ámbito reglado, aunque estamos bajo el amparo municipal y de la Junta de Andalucía, si creemos que hemos cumplido con la multiculturalidad en nuestro equipo, pues si bien todos los integrantes de mismo son españoles, el 67 por ciento son de origen Rumano, una población que abunda bastante en nuestra región.

Quizás podríamos sacar de este dato la conclusión de que los habitantes del antiguo bloque soviético mantienen un interés por la ciencia y la cultura que los padres han transmitido a los hijos, cosa que el capitalismo no ha conseguido....pero bueno es solo una hipótesis de café...

Ambiental

Nuestra mayor preocupacion en este campo eran los desplazamientos entre las dos poblaciones que distan unos 15 kilometros entre ellas, pero hemos decidido compensar las emisiones de CO₂ de nuestros desplazamientos aportando el coste economico de UNA TONELADA a una entidad de total confianza que lleva años realizndo proyectos con esta finalidad



Difusión del proyecto

Teleprensa:

<http://www.teleprensa.com/es/sorbas-uleila-del-campo/jovenes-usuarios-de-los-guadalinfo-de-sorbas-y-uleila-participan-en-un>

The screenshot shows a news article from Teleprensa. The main title is "Jóvenes usuarios de los Guadalinfo de Sorbas y Uleila participan en un concurso de la ESA para diseñar y construir un mini satélite". Below the title, there is a photograph of several people working at computers in a lab setting. On the left side of the article, there are social sharing options (Facebook, Twitter, Email) and a reading progress bar (11% Leido). On the right side, there is a sidebar titled "NOTICIAS RELACIONADAS" with links to other local news stories.

La Provincia:

<https://www.lavozdealmeria.com/noticia/3/provincia/187848/jovenes-de-sorbas-colaboran-en-el-diseno-de-un-satelite-en-miniatura>

The screenshot shows a news article from La Provincia. The main title is "Jóvenes de Sorbas colaboran en el diseño de un satélite en miniatura". Below the title, there is a subtitle: "El concurso 'CanSat', a nivel europeo, tiene por objetivo simular el lanzamiento de un satélite". Below the subtitle, there is a photograph of two young people working at a computer in a classroom setting. At the bottom of the article, there is a caption: "El grupo de jóvenes de los centros Guadalinfo - LA VOZ".

La Vanguardia:

<https://www.lavanguardia.com/local/sevilla/20200219/473661824264/economia--jovenes-de-guadalinfos-de-sorbas-y-uleila-participan-en-concurso-de-la-esa-para-disenar-un-satelite.html>

The screenshot shows a news article from LAVANGUARDIA | Andalucía. The headline reads: "Economía.- Jóvenes de Guadalinfos de Sorbas y Uleila participan en concurso de la ESA para diseñar un satélite". The article discusses six young people from Guadalinfo centers in Sorbas and Uleila del Campo, Almería, participating in the 'CanSat' competition organized by the European Space Agency (ESA) for secondary school students. It mentions that they simulate all aspects of a real mission to launch a satellite. The text is in Spanish.

La mañana del 24 de febrero fue la presentación del equipo en el IES Rio de Aguas de Sorbas.

Esa mañana empezó con una charla de Juanma, en la cual explicaba en que consistía el proyecto CanSat de Esero, para después, uno de los componentes del equipo explicaba cual era el rol de cada uno dentro del equipo, tras lo que se realizaron una serie de talleres sobre la carrera espacial y la construcción de cohetes que luego fueron lanzados en las pistas deportivas del IES.

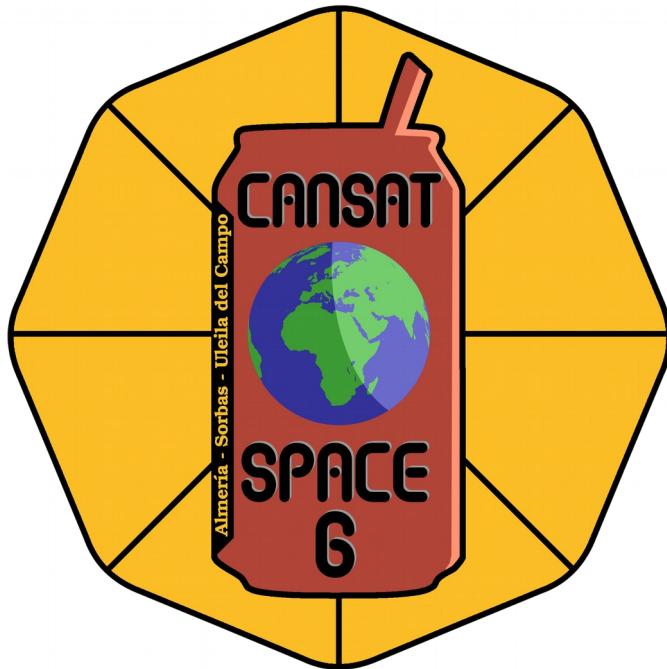




Las redes sociales son importantes porque enseñan o compartir con el mundo lo qué estás haciendo, y poder comunicarte con otras personas.

Las redes sociales que más hemos usado han sido Twitter e Instagram en las que hemos ido publicando nuestros avances en el proyecto, como la creación de la carcasa del satélite y las pruebas de vuelo con cohetes de propulsión fría . Principalmente hemos usado estas dos redes sociales ya que nos proporciona mayor difusión entre los equipos rivales y los creadores del concurso CanSat.

Logo



Este logo lo hemos creado en la página gimp. el logo se compone de una lata en la zona central con el nombre del concurso y el nombre de nuestro equipo que es space 6 . Para crear este logo tuvimos que pensar en un dibujo sencillo y que llame la atención al receptor.

Nuestro trabajo en las redes sociales

Red social	Publicaciones	Nombre/Red Social
Twitter	41 tweets	@SpaceSix1
Instagram	3 publicaciones	@project.cansat
Blogger	20 Publicaciones	cansatspacesix.blogspot.com
Youtube	11 videos	Space Six

Nuestro principal medio de difusión es via Twitter, pero previamente subimos las cosas al blog del grupo y a nuestro canal de Youtube.

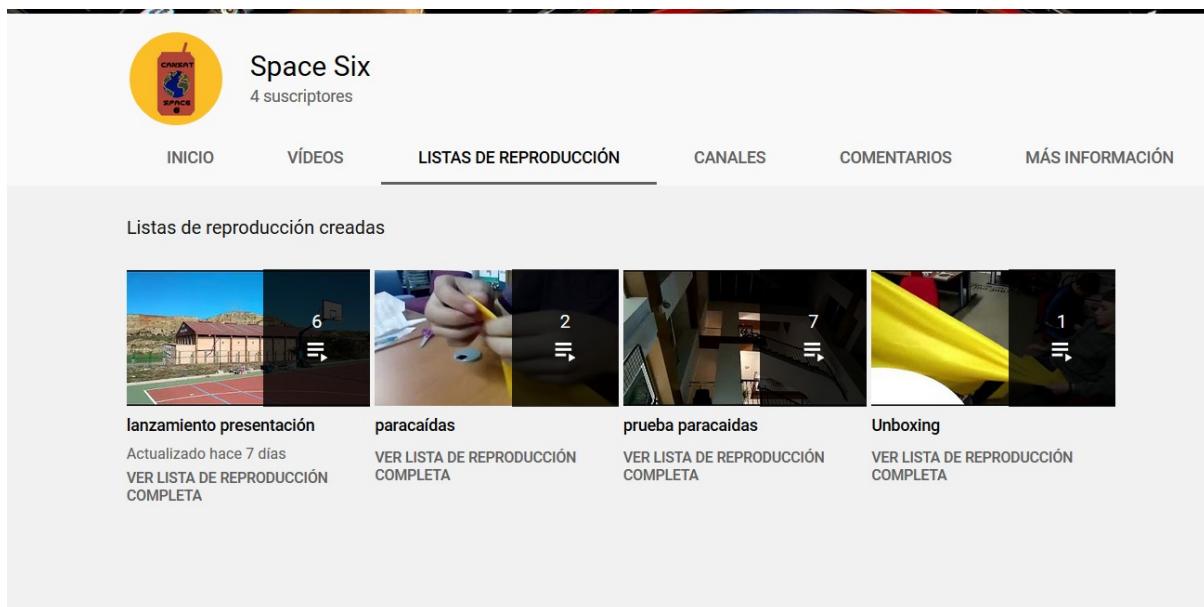
YOUTUBE

En nuestro canal hemos ido subiendo y creando listas de reproducción de todas las pruebas que hemos ido haciendo, así como de la confección del paracaídas y de la gran presentación del grupo ante los profesores y alumnos del instituto “IES Río Aguas”

La finalidad de este evento fue la de dar a conocer el proyecto CanSat e intentar captar gente para poder participar los años venideros.

Enlace a nuestro canal de Youtube:

https://www.youtube.com/channel/UCID6Y71hDKX_VGAhojaH4cg



vista general del canal

BLOG

Enlace a nuestro blog: <https://cansatspacesix.blogspot.com/>

El objetivo del blog era el de ir manteniendo un diario de los avances que íbamos haciendo.

The screenshot shows the homepage of the SpaceSix blog. At the top, there's a banner with the text "SPACESIX. DIARIO DE ABORDO". Below the banner, a short description in Spanish and English states: "Somos un grupo de jóvenes pertenecientes a la comarca Filabres-Alhamilla que se han embarcado en el proyecto CanSat con ilusión y muchas ganas de aprender. // We are a group of young people belonging to the Filabres-Alhamilla region who have embarked on the CanSat project with enthusiasm and eagerness to learn." The main content area features a section titled "Resumen (05/03/2020) -Actualización-", which includes a photo of two team members working on a satellite component and a caption: "Tras unos días de no parar, al fin podemos sentarnos frente al blog y actualizarlo. La semana previa a quedar fue un no parar: abocetar el diseño definitivo del la carcasa del satélite, diseñar y soldar diferentes "cerebros" para el CanSat, confeccionar el paracaídas que usaremos y pegarle un buen avance al CDR." To the right, there are two sidebar boxes: "Datos personales" (with a profile picture and a link to "Ver todo mi perfil") and "Archivo del blog" (listing posts from March 2020, February 2020, and January 2019). The background of the sidebar boxes features a stylized image of Earth.

TWITTER

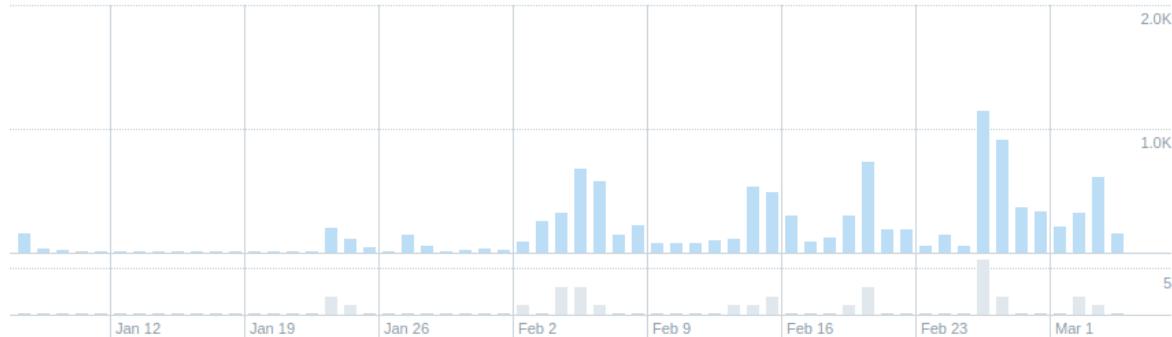
Enlace al Twitter del grupo: <https://twitter.com/SpaceSix1>

Como bien se ha podido ver en el gráfico de las fuentes externas, nuestra fuente principal radica en Twitter, ya que es una de las redes sociales más usadas y la mayor parte de personas tiene una cuenta en dicha red social .

Nuestro objetivo era el de usar dicha red para dar a conocer nuestro trabajo y usarla como complemento del blog y el canal de youtube ya que en los tweets ponemos enlace al blog y/o al canal de Youtube.

A continuación se muestran las estadísticas:

Sus Tweets consiguieron **10.8K impresiones** en este período de **58 días**



Distintos tipos de interacciones

Interacciones

Mostrar 58 días con frecuencia diaria

Tasa de interacción

2.1%

Mar 4
0.6% tasa de interacción



Clics en el enlace

24

Mar 4
0 clics en el enlace



En promedio, consiguió **0 clics en el enlace** por día

tasa de interacción y clics en el enlace

Retweets

107

Mar 4
1 retweet

En promedio, consiguió **2 Retweets** por día

Me gusta

163

Mar 4
0 me gusta

En promedio, consiguió **3 me gusta** por día

retweets y "me gusta"

ANEXO

Nuestro colaborador Eulogio López Cayuela lleva tiempo estudiando y trabajando por gusto en la creación, implementación y desarrollo de diversos programas y procedimientos basados en Linux para comunicación entre dispositivos, comunicación entre plataformas y otro que utiliza para gestionar adecuadamente su negocio de venta y reparación de productos informáticos e informática en general, podemos decir sin lugar a dudas que su negocio es el centro tecnológico de la comarca donde cualquier empresario, agricultor o ciudadano con necesidades informáticas acude.

Como digo el lleva tiempo desarrollando estos programas y cuando conoció nuestra iniciativa no dudo ni un segundo en ofrecernos su ayuda y adaptar algunos de los programas que tenía subidos a github

(<https://github.com/inopya/>) para que sirvieran de base a los chavales para realizar sus propios desarrollos.

A continuación publicamos las capturas de pantalla de estos desarrollos para que sirvan de apoyo a nuestro proyecto y de ayuda para cualquier grupo que los necesite

The screenshot shows a GitHub repository interface. At the top, there are buttons for 'Branch: master', 'New pull request', 'Find file', and 'Clone or download'. Below this, a table lists repository files:

File	Description	Last Commit
inopya Update parpadeo_tiempo_asimetrico.ino	Update parpadeo_tiempo_asimetrico.ino	Latest commit 615dc2d 22 days ago
TemporizadorInopya	Update parpadeo_tiempo_asimetrico.ino	22 days ago
LICENSE	Initial commit	23 days ago
README.md	Update README.md	23 days ago

Below the table, the 'README.md' file is shown in its entirety:

```

Temporizadores virtuales basados en la funcion Arduino millis()

Temporizadores versatiles y relativamente precisos basados en millis().

Para crear un temporizador: --> Temporizador_inopya nombreTemporizador;
Los temporizadores virtuales disponen de tres metodos: begin(), stop() y estado()

Una vez definido un temporizador se usa de la siguiente forma:

*nombreTemporizador.begin(tiempoEnMilisegundos) --> carga tiempo en el temporizador y lo inicia.

Si un temporizador esta en marcha no se puede sobreescribir su valor.
Debe ser parado previamente o esperar a que acabe para poder cargarle un nuevo valor.
* nombreTemporizador.stop() --> detiene el temporizador, pasando su estado a ser 'false'.

* nombreTemporizador.estado() --> retorna un valor booleano:
    true = temporizador aun activo;
    false = temporizador terminado

```

inopya Update library.properties

	Add files via upload	
examples		
src		
LICENSE	Initial commit	
README.md	Update README.md	27 days ago
keywords.txt	Add files via upload	27 days ago
library.properties	Update library.properties	6 days ago

Downloading...
Want to be notified of new releases in **inopya/I2C_EEPROM?**

Sign in **Sign up**

README.md

I2C_EEPROM_inopya V1.0 para Arduino

Autor: Eulogio López

Funciones

```
I2C_EEPROM_inopya.write( posicion, byte )
Este metodo permite leer de la eeprom un byte
```

```
I2C_EEPROM_inopya.read( posicion )
Este metodo permite guardar en la eeprom un byte.
```

```
I2C_EEPROM_inopya.save( posicion, dato )
Este metodo permite guardar en eeprom datos de cualquier tipo: byte, int, float, struct...
```

```
I2C_EEPROM_inopya.load( posicion, dato )
Este metodo permite leer de la eeprom datos de cualquier tipo: byte, int, float, struct...
```

Librería Generica para GPS NEO6, NEO7 y NEO8

18 commits 1 branch 0 packages 0 releases 1 contributor GPL-3.0

Branch: master New pull request Find file Clone or download

Inopya Update README.md

	Add files via upload	
readNEO8		
LICENSE	Initial commit	
README.md	Update README.md	4 days ago
Universal_GPS_inopya.cpp	Add files via upload	4 days ago
Universal_GPS_inopya.h	Add files via upload	4 days ago
keywords.txt	Add files via upload	4 days ago

Downloading...
Want to be notified of new releases in **inopya/inopya_Universal_GPS?**

Sign in **Sign up**

README.md

inopya_Universal_GPS

Librería básica para GPS NEO6, NEO7 y NEO8 Esta pensada para escuchar al GPS en pines que no sean puerto serie Hardware, por tanto necesita de la librería softSerial

Para usarla debemos crear un puerto serie que destinaremos al GPS.

```
SoftwareSerial gpsPort(PIN_Rx, PIN_Tx);
```

e inicializaremos una instancia del GPS pasandole como argumento el puerto serie software

```
Universal_GPS_inopya NEO_gps(&gpsPort);
```

En el **setup()** debemos inicializar el GPS de la siguiente forma:

```
NEO_gps.begin(9600); //iniciamos a la velocidad standard
```

Configurador para APC220 con interfaz grafica TKinter compatible Linux/Windows

APC220 GUI config

Configurador para APC220 con interfaz gráfica TKinter compatible Linux/Windows

Sencillo programa en Python/Tkinter para facilitar la configuración de los módulos de comunicaciones APC220 de DFrobot.

Como casi todo en esta vida... surge de la necesidad.

Me dejaron un par de estos módulos para probarlos y no pude hacer funcionar la utilidad RF-Magic que ofrece DFRobot. Así que hoja de características en mano hice una primera versión muy tosca para programarlos usando un Arduino que enviaba los comandos serie adecuados. Podeis ver dicha versión aquí: https://github.com/inopya/APC220_Transceiver Me sacó del apuro pero entendí que no era demasiado amigable para que la usasen otros. En este caso he optado por un poco de python y una sencilla interfaz gráfica en Tkinter de modo que sea intuitivo y sobre todo compatible con linux.

Requisitos: Modulos APC220, Arduino Uno (Nano, Micro, Mega...), Python y tener instalada la librería pyserial: <https://pypi.org/project/pyserial/>