

INFORME DE REVISIÓN CRÍTICA DEL DISEÑO

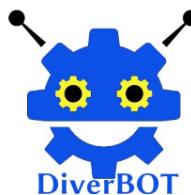


LAIKATA

EQUIPO: Laikata Team

MENTOR: Javier Gomez

Patrocinadores:



MADRE ARNAU



CRISTINA M.



LAIKATA

CanSat 2022

Índice

Organización y roles del equipo	3
Objetivo de la misión	4
Descripción del proyecto CanSat	5
Esquema de la misión	5
Diagrama de bloques	6
Objetivos de la misión	6
Demostración de tecnología	7
Diseño mecánico y estructural	8
Diseño electrónico	10
Software	11
Diagrama de flujo de la programación	11
Protocolo de comunicación con la estación de tierra	11
GPS	12
Imu	12
Temperatura humedad y presión	12
Información adicional	12
Cantidad de datos recopilados	13
Lenguajes de programación utilizados	13
Sistema de recuperación	13
Estación de tierra	13
Planificación	14
Planificación del proyecto CanSat	14
Estimación de recursos	15
Presupuesto	15
Apoyo externo	15
Pruebas realizadas	16
Primera prueba	16
La lanzadera	17
Hardware	17
Software	18
Programa de difusión y patrocinio	19
Bibliografía, referencias y recursos utilizados	20



Introducción

Todos nosotros estamos acostumbrados a aprender por proyectos. Lo hacemos en el instituto, y también en la academia de robótica a la que asistimos. Dejando un poco de lado los proyectos personales, terminamos el curso del año pasado haciendo un “Curiosity” a escala, con sus seis ruedas, distintos drivers para controlar servos y motores, tres placas de control para gobernar los distintos módulos, todas ellas conectadas con un mini-PC a través de I2C, al cual nos conectábamos por SSH y canales TCP/UDP para controlarlo. Fué todo un reto ya que era la primera vez que un proyecto requería de tantos sistemas. Aprendimos, y logramos ponerlo en funcionamiento.

Este año, 2021-2022, decidimos que, entre otros proyectos, nos presentaríamos al desafío CanSat. Nos pareció muy interesante el tener que lidiar con restricciones como el espacio, sistemas de alimentación, links de radio con no mucho ancho de banda, etc. Así es que tras hablarlo unos días y dar con una misión secundaria que nos motivaba a todos, nos pusimos manos a la obra, le pusimos nombre a nuestro pico-satélite (Laikata) y al equipo (Laikata Team).

1.1. Organización y roles del equipo

Laikata Team está formado por 10 personas, mayoritariamente somos todos estudiantes de secundaria, con alrededor de 16 años. Tenemos en común un gran interés por la programación y robótica, pero no llegamos a ponernos de acuerdo de si es mejor programar en lenguajes de alto nivel o ensamblador.

Para afrontar el proyecto y poder avanzar más rápido, decidimos dividir la empresa en departamentos. En cada uno de ellos nombramos a una persona responsable, la cual podía solicitar la colaboración de quien quisiese, por lo que todos terminaríamos haciendo de todo, pero con ciertos supervisores, que serían los máximos responsables en sus áreas. Los departamentos/equipos creados han sido los siguientes:

Equipo de diseño estructural: Tendrá como objetivo el diseño de Laikata para que ésta pueda integrar todos los sistema requeridos para afrontar las dos misiones, así como el sistema de parapente.

Equipo de diseño electrónico: Su misión será el diseño de todo circuito y conexión electrónica que requieran los distintos sistemas.

Equipo de desarrollo de software: Encargado de desarrollar tanto el software como el firmware necesario, seleccionar el IDE a utilizar, y asegurarse de que todos lo sabían manejar.

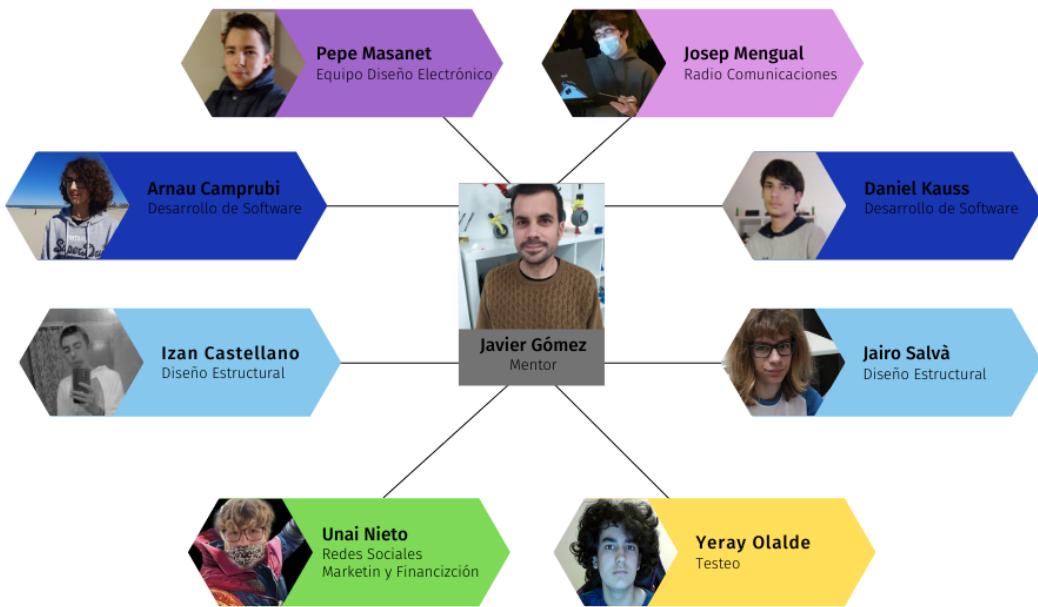
Equipo de radio comunicaciones: Encargado de todo el sistema de comunicación, tanto entre Laikata y la base, como del rover. También llevará la parte técnica de las comunicaciones internas entre los distintos componentes de los equipos.



Equipo de testeo: Su misión, preparar cuantas pruebas sean necesarias para verificar que el diseño supera las especificaciones requeridas, tanto de software como de hardware. Será el encargado de planificar las posibles salidas para realizar pruebas de vuelo.

Equipo de redes sociales, comunicación, marketing, branding y financiación: Este equipo se encargará de las publicaciones en redes sociales, comunicaciones con los distintos medios, y el posible marketing de la misión para conseguir financiación.

ORGANIGRAMA LAIKATA TEAM



1.2. Objetivo de la misión

El desafío CanSat se compone de dos misiones:

Misión principal: viene dada por la organización, y consiste en que tras la liberación y descenso del CanSat, este deberá enviar por telemetría hasta la estación de tierra, mediciones de temperatura y presión atmosférica al menos una vez por segundo.

Misión secundaria: elegida por nosotros. Consiste en una demostración de tecnología. Vamos a conseguir que Laikata vuelva navegando hasta un punto determinado mediante el control de un parapente. Aprovecharemos también el canal Down-link de Laikata para enviarnos una gran cantidad de datos, como la posición exacta mediante GPS, el valor de los 9 ejes de un sensor inercial, la tensión de la batería, así como información sobre las decisiones tomadas por el navegador de forma autónoma. Con todo ello, crearemos un



sistema de monitorización muy completo de nuestro Laikata en la estación de control de tierra.

2. Descripción del proyecto CanSat

2.1. Esquema de la misión

Nuestra misión consiste en hacer que nuestro CanSat navegue de forma autónoma hasta ciertas coordenadas de destino. A su vez, enviaremos una gran cantidad de datos de telemetría, entre los que se incluirán temperatura y presión, para cumplir con la misión principal, e información sobre sensores iniciales, monitorización de batería, e información sobre decisiones tomadas para la navegación. Todos estos datos se recibirán en la estación de tierra, donde se implementará un software para el registro y visualización de todos ellos. La estación de tierra dispondrá de dos antenas con posicionamiento automático síncrono, una de ellas con polarización horizontal y la otra vertical. Las dos antenas recibirán las señales enviadas por Laikata y enviarán los datos recibidos al software de monitorización, y éste será el encargado de coger únicamente las tramas que hayan llegado de forma íntegra, tras realizar la verificación.



Diagrama de bloques

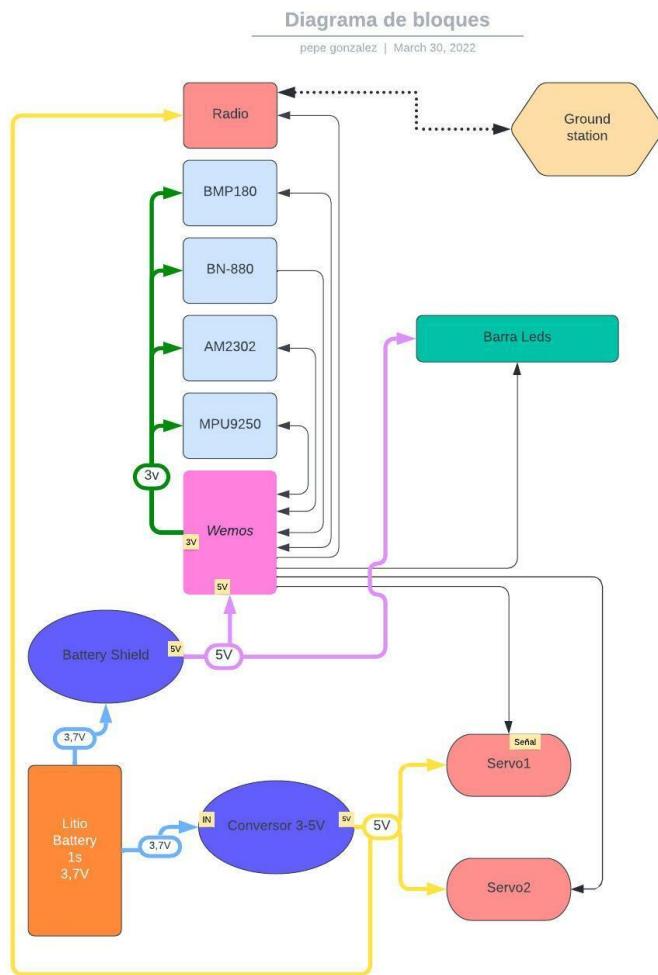


Diagrama de bloques de las conexiones del CanSat

Objetivos de la misión

El objetivo de la misión de nuestro CanSat es que mediante el gps, el CanSat maniobre con el parapente para dirigirse a una ubicación exacta donde se sitúa nuestro apoyo terrestre. Durante el lanzamiento la radio pasará continuamente tramas con telemetría que nos ayudará a saber en todo momento donde está ubicado y que está pasando.

Daremos por cumplidos los objetivos de nuestra misión secundaria siempre y cuando Laikata logre aterrizar en un radio de 10m de su punto de destino. Si las condiciones meteorológicas impidiese el correcto desempeño de la misión, siempre nos quedará una baliza de radio en la que se enviarán tramas con la localización exacta para su recuperación.

Para la misión secundaria se necesitan de varios componentes como son, los servos para frenar y girar el parapente; un módulo de comunicaciones por radio para comunicarse con la



estación base y un GPS para conocer la ubicación del CanSat y que así el mismo pueda dirigirse al punto de caída. Para alimentar el microcontrolador ESP8266 y todos los sensores usamos el módulo “battery shield” de Wemos. Los servos y el módulo de radio se alimentan con una fuente de alimentación independiente.

Todo el sistema está alimentado con una batería de Litio de una celda con capacidad de 1400mAh. Se ha estimado que durante el descenso Laikata consumirá unos 800mA durante 200s. Una vez en tierra, se desactivarán, y se reducirán las transmisiones a una baliza cada 5 segundos, por lo que su consumo descenderá a unos 100mA. Haciendo una estimación calculamos que la batería nos permitirá unas 9h de autonomía tras el despegue, tiempo más que suficiente para recuperar a Laikata aún y cuando no haya podido cumplir su misión secundaria con éxito.

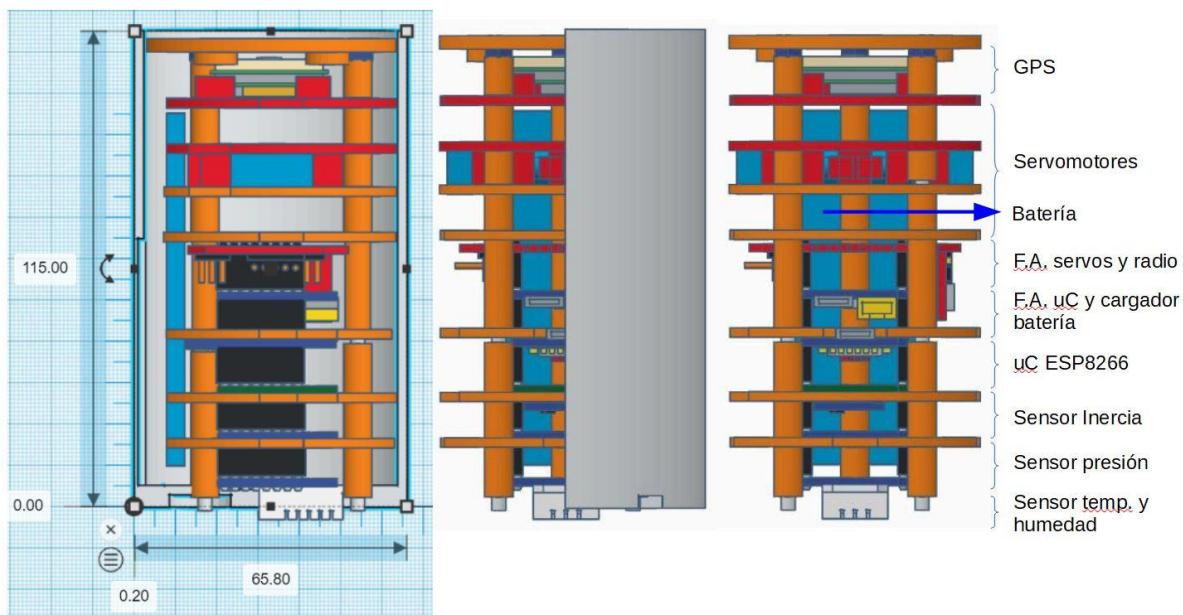
2.2. Demostración de tecnología

Se demostrará con el CanSat que se es capaz de, mediante cálculos hechos en tiempo real, calcular el vector actual del CanSat respecto a la zona indicada de donde debe aterrizar. Y con este vector mover los servos para que respectivamente se mueva el paracaídas y dirigir la nave al lugar de aterrizaje.

Si, por lo tanto todo se produce con normalidad el CanSat aterrizará en la zona de aterrizaje sin desviarse. Además usando el GPS incorporado podremos localizar dónde se encuentra en todo momento el cansat y así poder recuperarlo.

2.3. Diseño mecánico y estructural

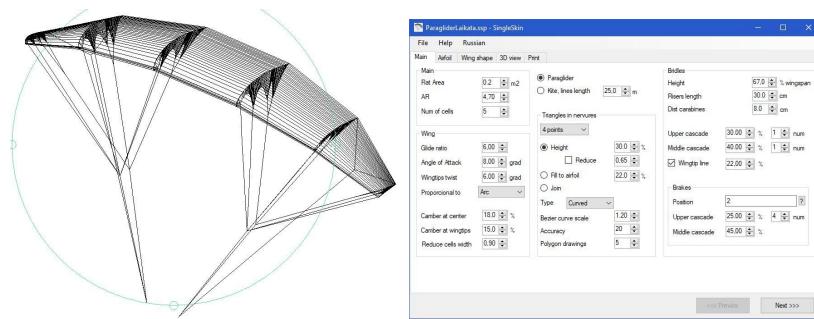
Para el diseño de nuestra lata hemos utilizado un diseño modular, ya que nos permite de forma fácil trabajar con los distintos módulos, pudiendo modificar alguno de ellos sin afectar al conjunto.. Como herramienta para el diseño se ha utilizado Tinkercad de Autodesk. Y como material se ha optado por la utilización de PLA, por ser más sostenible con el medio ambiente que el ABS, aunque sea un poco menos robusto. Unas varillas de acero de 3mm serán las utilizadas para unir los distintos módulos que posteriormente se introducirán dentro de la lata, la cual tiene un espesor de 2.3mm para darle la suficiente solidez.. Cada módulo contiene una parte diferente de la lata, como los sensores, los módulos de radio, o los servos. Estos últimos son los utilizados para regular los frenos del parapente para su guiado.



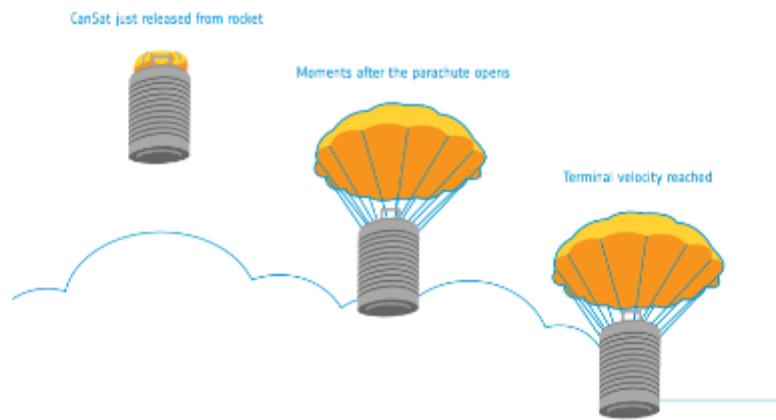
Diseño gráfico del CanSat

Antes de fabricar el parapente, tuvimos que calcular el área necesaria para cumplir los requisitos de la ESA. Se nos exigió que la velocidad durante el descenso no superara los 10 m/s, pero tampoco fuera menor que 4 m/s. Con estos requisitos en mente, y unas ecuaciones que nos proporcionó la ESA, obtuvimos que el área de nuestro parapente debía de ser de $0,11\text{m}^2$.

Con estos datos procedimos a realizar el diseño definitivo del parapente, utilizando el software “SingleSkin”, el cual permite diseñar parapentes de piel única, siendo estos más sencillos de implementar que los de cámara.



Visualización del parapente diseñado, y software “SingleSkin”



Etapas del lanzamiento del CanSat

$$mg - \frac{1}{2}C_D \rho A v^2 = 0$$

$$\begin{aligned} V_{\max} &= 11 \text{ m/s} \\ V_{\min} &= 8 \text{ m/s} \end{aligned}$$

Fuerza de gravedad

$$g = 9,81 \text{ m/s}^2$$

Masa de Laikata

$$m = 0,35 \text{ kg}$$

Velocidad de descenso

$$V = 8 \text{ m/s}^2$$

Coeficiente de viscosidad

$$C_d = 0,75$$

Densidad del aire

$$\rho = 1,225 \text{ kg/m}^3$$

Área del paraglider

$$A = 0,116785714 \text{ M}^2$$

$$A_{\min} = 0,0617$$

Dimensiones del paraglider

$$\text{Ancho} = 0,778571429 \text{ m}$$

$$\text{Largo} = 0,15 \text{ m}$$

$$\text{Área} = 0,116785714 \text{ M}^2$$

Cálculos realizados para obtener el área necesaria del parapente



2.4. Diseño electrónico

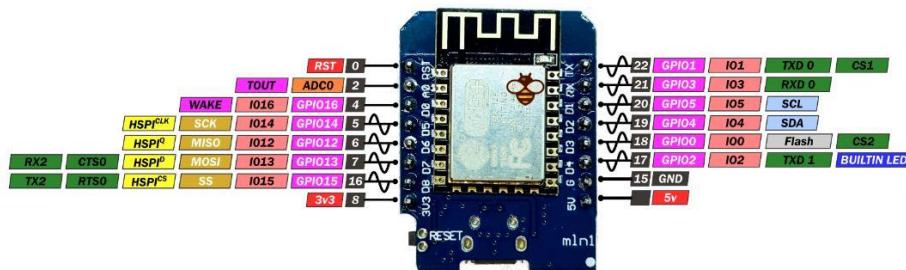
Para la electrónica nos hemos basado en la familia Wemos D1 Mini, que contiene un ESP8266. Nos hemos decidido por esta placa ya que con ella se puede implementar fácilmente un sistema modular. Dispone de distintos “hats” como el de baterías, sensores de presión, de temperatura, etc. Su formato es muy reducido en tamaño, lo que nos ayudaría a conectar cuantos módulos necesitásemos en vertical.

Para la misión principal se ha escogido el sensor BMP180 para medidas de presión, con un margen de error de error de 0.02hPa, el AM2302 para realizar medidas de temperatura con una precisión de $\pm 0.5^{\circ}\text{C}$. Como módulo de radio para transmitir la telemetría se ha escogido el módulo HC12.

Para la misión secundaria, hemos añadido el MPU9250 como sensor inercial de 9 ejes, como GPS utilizamos el BN-880, que soporta GPS,GLONASS, Galileo,BeiDou, QZSS y SBAS, dos microservos.

Para completar nuestro sistema se añadió el “hat” de baterías, que nos permite tanto cargar la batería como proporcionar los 5V que necesita la placa de nuestro microcontrolador para trabajar, así como una pequeña fuente de alimentación comutada para proporcionar 5V a los servos y al módulo de radio.

A falta de realizar el circuito electrónico pormenorizado, a continuación se muestra como han quedado los distintos sensores conectados con el microcontrolador.



ServoD	BMP180	HC12	AM2302	MPU9250	BN-880	Pin wemos	Pin wemos	BN-880	MPU9250	AM2302	HC12	BMP180	Servol
	Presión	Radio	Hum/Temp	IMU	GPS	Micro. RST	Micro. TX	GPS RX	IMU	Hum/Temp	Radio	Presión	
						A0	D0	D1	SCL			TX	
servo						D5	D2	D2	SDA	SCL		SCL	
						D6	D3	D3	SDA	SDA		SDA	
						D7	D4	D4			DATA		
						D8	GND	GND	GND	GND	GND		
	VCC	VCC	VCC	VCC		3.3V	5V					GND	

Diseño de los pines utilizados para cada placa

Como se puede apreciar, alguno de ellos usan el bus I2C, otros la UART, y otros un protocolo propio. El GPS y el módulo de radio comparten UART, ya que la comunicación con ambos es unidireccional. De este modo, nos ha sido suficiente con la única UART por hardware de que dispone el ESP8266. En caso de requerir en un futuro comunicación

bidireccional, por ejemplo con el enlace de tierra, se implementaría una UART por software en dos puertos no utilizados.

2.5. Software

2.5.1. Diagrama de flujo de la programación

Para ilustrar mejor el funcionamiento del código principal del CanSat, hemos representado los procedimientos de este en un diagrama de flujo.

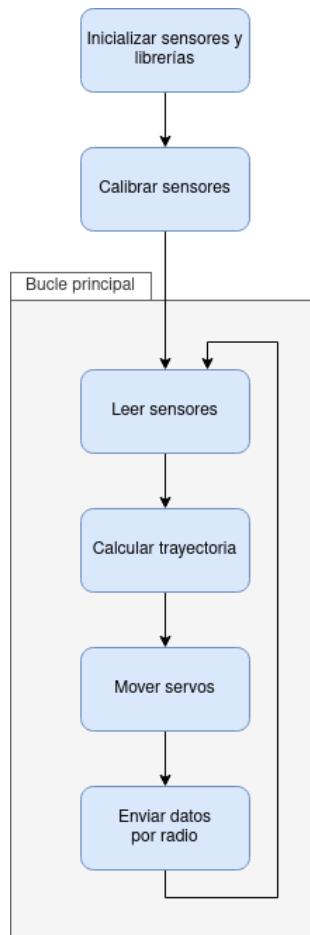


Diagrama de flujo del código principal del CanSat

2.5.2. Protocolo de comunicación con la estación de tierra

Hemos creado un protocolo ligero, que empaqueta los datos a enviar y añade un *checksum* en caso de que haya interferencias en el canal de radio.

Para enviar datos a través del protocolo se envían paquetes que contienen lo siguiente:

- Un carácter que marca el inicio del paquete (usamos SYN o 0x16)



- Un *header* que indica el tamaño de los datos en bytes. El *header* mide un *byte* (rango 0 a 256)
- Un contador que nos ayuda a determinar cuándo perdemos un paquete. Asciende uno por cada paquete enviado.
- Los datos a transmitir (del tamaño especificado en el *header*)
- El CRC-32 de los datos (cuyo tamaño es de 32 *bits* o 4 *bytes*)

S Y N	HE AD ER	CO UN T	DATA	CHECKSU M
1	1	1	###	4

Diagrama que representa un paquete

Utilizando este protocolo como base hemos definido un conjunto de subprotocolos para agilizar el desarrollo del software de comunicación entre la estación base y el CanSat. Hay un total de cuatro subprotocolos:

1. GPS

HEADER	LONGITUD	LATITUD	ALTITUD
0x01	<i>float</i> (4 bytes)	<i>float</i> (4 bytes)	<i>float</i> (4 bytes)

2. Imu

HEADER	MAGNETÓMETRO	ACELERÓMETRO	ALTITUD
0x02	3x <i>float</i> (12 bytes)	3x <i>float</i> (12 bytes)	3x <i>float</i> (12 bytes)

3. Temperatura humedad y presión

HEADER	TEMPERATURA	HUMEDAD	PRESIÓN
0x03	<i>float</i> (4 bytes)	<i>float</i> (4 bytes)	<i>float</i> (4 bytes)

4. Información adicional

HEADER	VOLTAJE	SERVO A	SERVO B
0x04	<i>float</i> (4 bytes)	<i>float</i> (4 bytes)	<i>float</i> (4 bytes)



2.5.3. Cantidad de datos recopilados

El CanSat contiene un IMU que recopila la aceleración, la señal magnética y orientación del CanSat en tres ejes cada uno. También recopilamos la temperatura y humedad a través de placas dedicadas. Toda esta información se transmite en tiempo real a través del módulo de radio.

2.5.4. Lenguajes de programación utilizados

Hemos usado el *subset* de C++ del [framework de Arduino](#) para la programación del ESP8266. El código se ha compilado y subido a las placas utilizando [PlatformIO](#).

Todo el código se encuentra en el [GitHub del proyecto](#). Hemos usado [git](#) para el control de versiones.

2.6. Sistema de recuperación

Una vez que el CanSat haya aterrizado en el terreno con la ayuda del paracaídas, una matriz LED delataría la posición de la lata. Además, para poder recuperarlo mejor se usaría el recorrido recibido del GPS para saber la zona aproximada de aterrizaje, incluso en el caso en el que no aterrice en la zona designada. También se seleccionó para la tela del paracaídas colores que contrastan tanto con el terreno del suelo, como con el cielo cuando aún está en el aire. Estos colores son azul oscuro y amarillo. Gracias a esto se facilita el poder visualizar dónde se encuentra con la propia vista, así se puede recuperar en caso de que todo lo anterior falle.

2.7. Estación de tierra

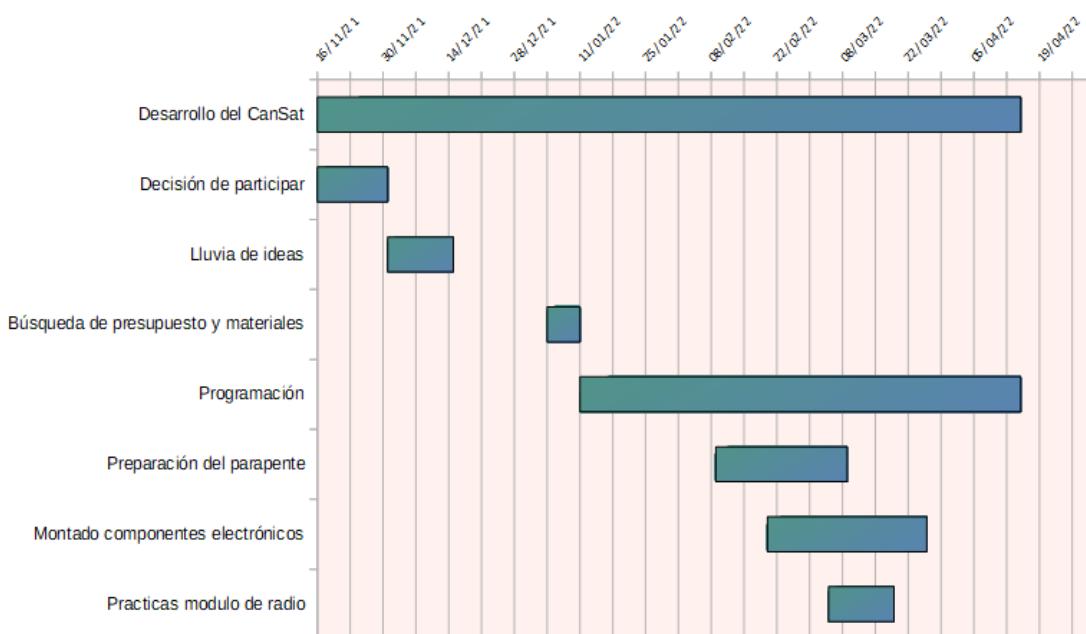
La estación de tierra consiste en dos receptores de radio (HC12) conectados a dos antenas con polarización cruzada. Éstos estarán conectados a un Arduino, el cual se comunica con un portátil sobre usb. El software principal de la estación de tierra está programado en Unity y C#. El arduino recibirá los datos sobre radio, y los mandará al portátil, donde se procesarán, visualizarán y finalmente guardarán en un archivo. Usamos [Ardity](#) para la comunicación entre Unity y Arduino.

3. Planificación

3.1. Planificación del proyecto CanSat

A continuación se encuentran varios diagramas representando la economización del tiempo disponible

Actividad	Fecha		Duración
	Inicio	Fin	
Desarrollo del CanSat	16/11/21	15/04/22	150
Decisión de participar	16/11/21	01/12/21	15
Lluvia de ideas	01/12/21	15/12/21	14
Búsqueda de presupuesto y materiales	04/01/22	11/01/22	7
Programación	11/01/22	15/04/22	94
Preparación del parapente	09/02/22	09/03/22	28
Montado componentes electrónicos	20/02/22	26/03/22	34
Prácticas modulo de radio	05/03/22	19/03/22	14



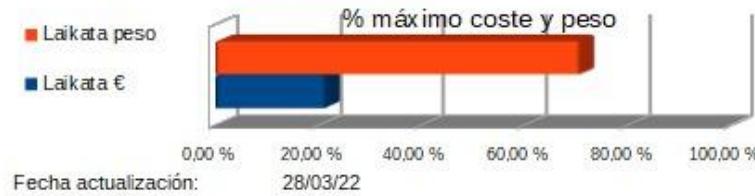
Representación mediante diagrama de Gantt del tiempo invertido en cada departamento



3.2. Estimación de recursos

3.2.1. Presupuesto

Proyecto Laikata



Presupuesto

Unidad	Descripción	Coste unitario	Cantidad	Coste total	Peso Unitario	Peso Total
Microcontroladores						
WemosD1mini ESP8266		4,95 €	1	4,95 €	7,00 g	7,00 g
Sensores						
MPU9250 IMU 9 ejes		4,95 €	1	4,95 €	4,00 g	4,00 g
AM2302 Humedad y Temperatura		5,95 €	1	5,95 €	8,00 g	8,00 g
BMP180 Presión		2,95 €	1	2,95 €	7,00 g	7,00 g
BN-880 GPS		24,95 €	1	24,95 €	13,00 g	13,00 g
				0,00 €		0,00 g
Actuadores						
FS0403-FB Servo		6,95 €	2	13,90 €	7,00 g	14,00 g
8x-ws2812 Barra leds		6,10 €	1	6,10 €	5,00 g	5,00 g
Sistemas de alimentación				0,00 €		
Bateria Lipo		6,50 €	1	6,50 €	12,00 g	12,00 g
Battery Shield		4,12 €	1	4,12 €	7,00 g	7,00 g
Conversor 3-5V		3,30 €	1	3,30 €	6,00 g	6,00 g
				0,00 €		0,00 g
Componentes electrónicos discretos						
Placa islas		1,00 €	1	1,00 €	10,00 g	10,00 g
Placa islas Wemos		2,00 €	1	2,00 €	5,00 g	5,00 g
Condensadores		1,00 €	1	1,00 €	3,00 g	3,00 g
Conectores		3,00 €	1	3,00 €	3,00 g	3,00 g
Estructura						
Nylon impermeable	Parapente	15,00 €	1	15,00 €	15,00 g	15,00 g
Dyneema 1m	Cuerda	0,20 €	15	3,00 €	0,90 g	13,50 g
Carcasa exterior	PLA		1	2,00 €	70,00 g	70,00 g
Separadores			4	0,00 €	3,00 g	12,00 g
Varilla 3mm		1,00 €	1	1,00 €	28,00 g	28,00 g
Tuercas varias		1,00 €	1	1,00 €	5,00 g	5,00 g
	Total			106,67 €	Total	247,5
	Laikata €			21,33 %	Laikata peso	70,71 %

Presupuesto y peso ilustrado por partes

3.2.2. Apoyo externo

Hemos hablado con los centros educativos de la zona incluyendo colegios e institutos pero estos tienen múltiples dificultades para patrocinar un proyecto de este estilo debido a la escasez de recursos con todos los gastos. No hemos recibido por el momento respuestas definitivas, pero la búsqueda de apoyo por parte de instituciones públicas sigue en pie.



Además de lo descrito, se han recaudado 20€ de la madre de uno de los integrantes del equipo que decidió que era un buen momento para patrocinar el proyecto.

La empresa Pineda Electrodomésticos ha considerado que era un proyecto interesante al que patrocinar, así como la academia DiverBOT.

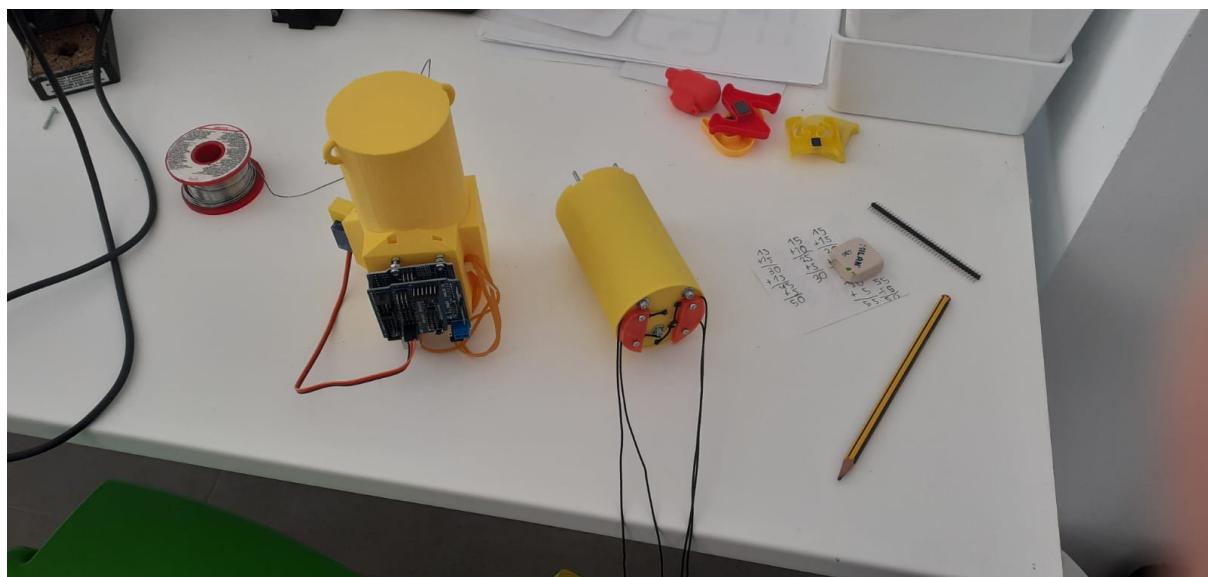
3.3. Pruebas realizadas

Hemos realizado múltiples pruebas con el fin de refinar nuestro proyecto hasta el punto de mayor rendimiento y precisión. Algunas de ellas acabaron en la inutilización parcial de nuestro CanSat, que en esta fase aún no contenía electrónica por lo que no se produjo ninguna pérdida.

3.3.1. Primera prueba

En la primera prueba de nuestro CanSat, elaboramos un módulo de lanzamiento (más detalles a continuación) para poder realizar lanzamientos con un drone.

El propósito de esta prueba era el de comprobar el funcionamiento de nuestro parapente, ya que no había sido probado hasta el momento. Ya que tampoco teníamos el CanSat en un estado funcional, decidimos lanzar una réplica en su lugar. La réplica consistía de la carcasa del CanSat y objetos en su interior que servían de peso (ya que tenía que imitar el peso del Cansat verdadero).



La lanzadera (izquierda) y la réplica del CanSat (derecha)

Al intentar volar el drone, tuvimos un problema técnico. Durante el montaje acelerado del drone esa misma tarde, una pieza fue colocada de forma incorrecta. Dicho fallo hizo que una de las hélices se soltara en el momento del despegue, resultando en una caída repentina del drone.



Una vez arreglado, despegó sin problemas. Pero otro problema ocurrió. La lanzadera (o módulo de lanzamiento) estaba programada para soltar el CanSat 60 segundos después del lanzamiento, pero, debido a un error en el código de esta no realizó su función. Se bajó el CanSat a tierra firme para poder solucionar el *bug*.

En el tercer intento, por fin se soltó el CanSat. Se desplegó el parapente correctamente y sin enredarse. Pero por desgracia, debido a no estar el parapente bien calibrado, este no voló en línea recta, como era de esperar. Más tarde, ajustando las cuerdas del parapente, se consiguió que el radio de la trayectoria del parapente fuese más grande. Lo dimos por válido, ya que ese pequeño desajuste se podrá compensar con los servos, una vez estos controlen los frenos.

En su último vuelo, el CanSat sufrió una caída sobre una zona de cemento, por lo que acabó destrozado.



Estado del CanSat al final de su primera prueba

Esta prueba fue muy útil ya que nos preparó para las siguientes pruebas al mismo tiempo que nos ayudó a ajustar el parapente para conseguir un vuelo óptimo.

3.3.2. La lanzadera

Para las pruebas desarrollamos el módulo de lanzamiento detallado a continuación.

3.3.2.1. Hardware

El módulo de lanzamiento estaba formado por un microcontrolador *Atmega328*, un servomotor *SG90* y una batería de litio.

Las conexiones electrónicas son bastante sencillas.

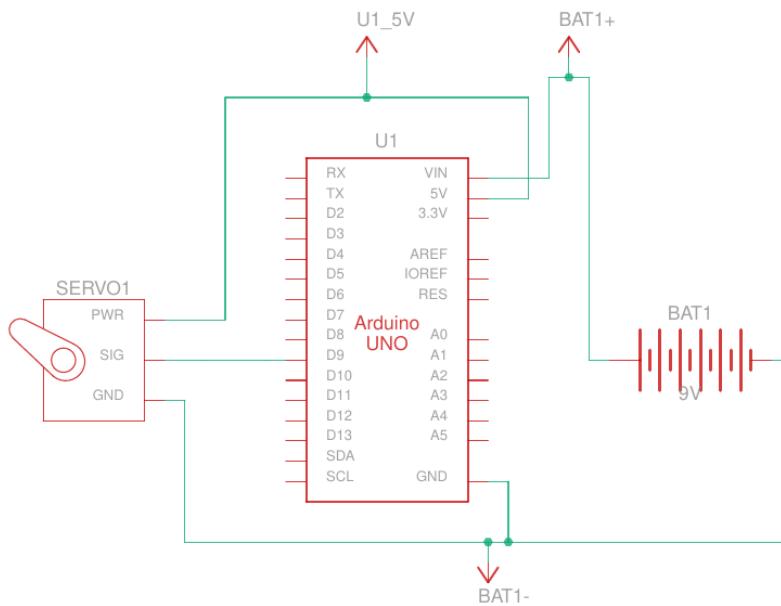


Diagrama de las conexiones eléctricas

Las diferentes partes mecánicas del módulo de lanzamiento fueron impresas con una impresora 3D y montadas con tornillería, bridales y gomas (que sirven como resortes).

3.3.2.2. Software

Para la programación de la lanzadera se utilizó C++ en el framework de Arduino.

El código simplemente se espera 30 segundos (o la cantidad configurada) y acciona el servomotor que libera el CanSat. A continuación se encuentra un diagrama de flujo de dicho código junto con una simplificación de este.

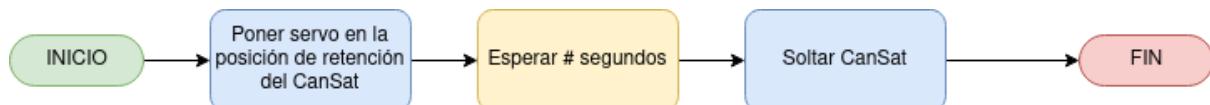


Diagrama de flujo del código de la lanzadera



4. Programa de difusión y patrocinio

La difusión mediática del proyecto se está realizando a través de la plataforma Instagram y una página web propia para el proyecto. A través de Instagram se publicaron imágenes y videos sobre el avance y las pruebas realizadas, así mostrando de forma simple y rápida los objetivos del proyecto al público. Mediante la página web se detalla de mejor forma, gracias a lo que permite poder escribir texto, los avances de la lata en un blog. Además, se explicaba en detalle la misión y quienes somos. Esto permitió ser una entrada, y una forma de enseñar en que consiste Laikata a los patrocinadores.

Para el patrocinio se llevaron a cabo campañas en los colegios, para pedir a estos el patrocinio a cambio de mostrar para el alumnado los objetivos de la misión y cómo llegamos a ello. También se organizó una recolecta de dinero de particulares, donde la gente donaba al proyecto a cambio de participar en un concurso que se realizará una vez acaben los lanzamientos. Por último se le ofreció a marcas y empresas locales el enseñar su marca en zonas como la camiseta del equipo a cambio de patrocinio.

La página web: <http://laikata.diverbot.es/>

El instagram: https://www.instagram.com/laikata_team/?hl=es

El GitHub: <https://github.com/Laikata>



5. Bibliografía, referencias y recursos utilizados

A continuación se hallan los recursos utilizados durante la elaboración del proyecto Laikata

Software para creación de esquemas de paraglider:

<http://www.laboratoridenvol.com/projects/singleskin/sk-nr.html>

<https://sourceforge.net/projects/singleskin/>

Explicación de funcionamiento, materiales etc de un paraglider de doble piel:

https://www.skynomad.com/articles/paraglider_construction.html

Modelos en venta de paragliders para RC de single skin:

https://www.opale-paramodels.com/gb/rc-paramotor-paraglider-wing/37-oxy-05-3760245320_116.html

Foro sobre construcción de un paraGlider singleSkin, aunque muy complejo por que tiene muchas celdas:

<https://forum.flitetest.com/index.php?threads/an-r-c-paramotor-wing-you-can-build.35994/>

Construcción de un paraglider para uso a mayores escalas:

<https://www.youtube.com/watch?v=Ku9XCtRHAqA>

Series de videos de “bandarra” en el que intenta hacer un paraglider a mano, y luego basándose en el software singleskin:

https://www.youtube.com/results?search_query=making+your+own+paraglider+bandarra

Proyecto R2Home en el que intenta hacer precisamente lo que pretendemos:

<https://hackaday.io/project/176621-r2home>

Otro proyecto similar, en el que se basa el anterior:

<https://hackaday.io/project/176736-project-bird-fish-free-flight-robotics>

El repositorio de Github, donde se puede encontrar toda la documentación relacionada con el proyecto: <https://github.com/Laikata/Laikata>

Recursos de ayuda ofrecidos por la ESA

https://www.esa.int/Education/CanSat/Cansat_resources