

# ChipaSat

## **Colegio:**

Instituto Padre Agustín Elizalde

## **Miembros del equipo:**

- Ferrari, Valentín.
- Martinez, Román.
- Mainero, Aaron.
- Regner, Luka.
- Rodriguez, Bautista

## **Tutor responsable:**

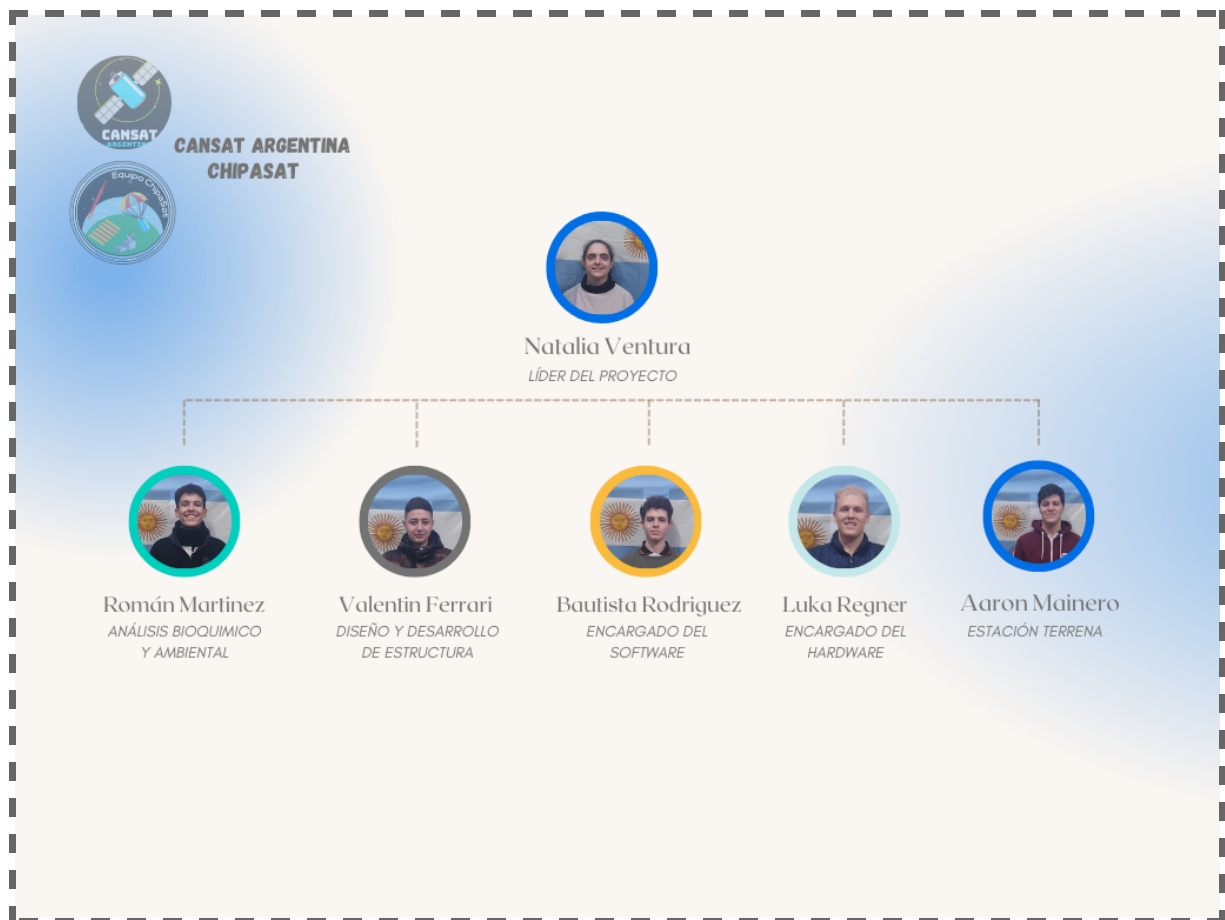
Natalia Ventura

## **Profesor asesor:**

Lucas Blanco



## **Organigrama**



## **Términos y siglas a utilizar**

No utilizamos terminología específica en el documento.

**Nombre de la misión:** Búsqueda de atmósferas habitables en exoplanetas.

**Parche:**



## **Objetivos**

### **Misión primaria**

Esta misión primaria es provista por la CONAE y consiste en el sensado de presión y temperatura y la transmisión de esos datos a la estación terrena al menos una vez por segundo.

### **Misión secundaria**

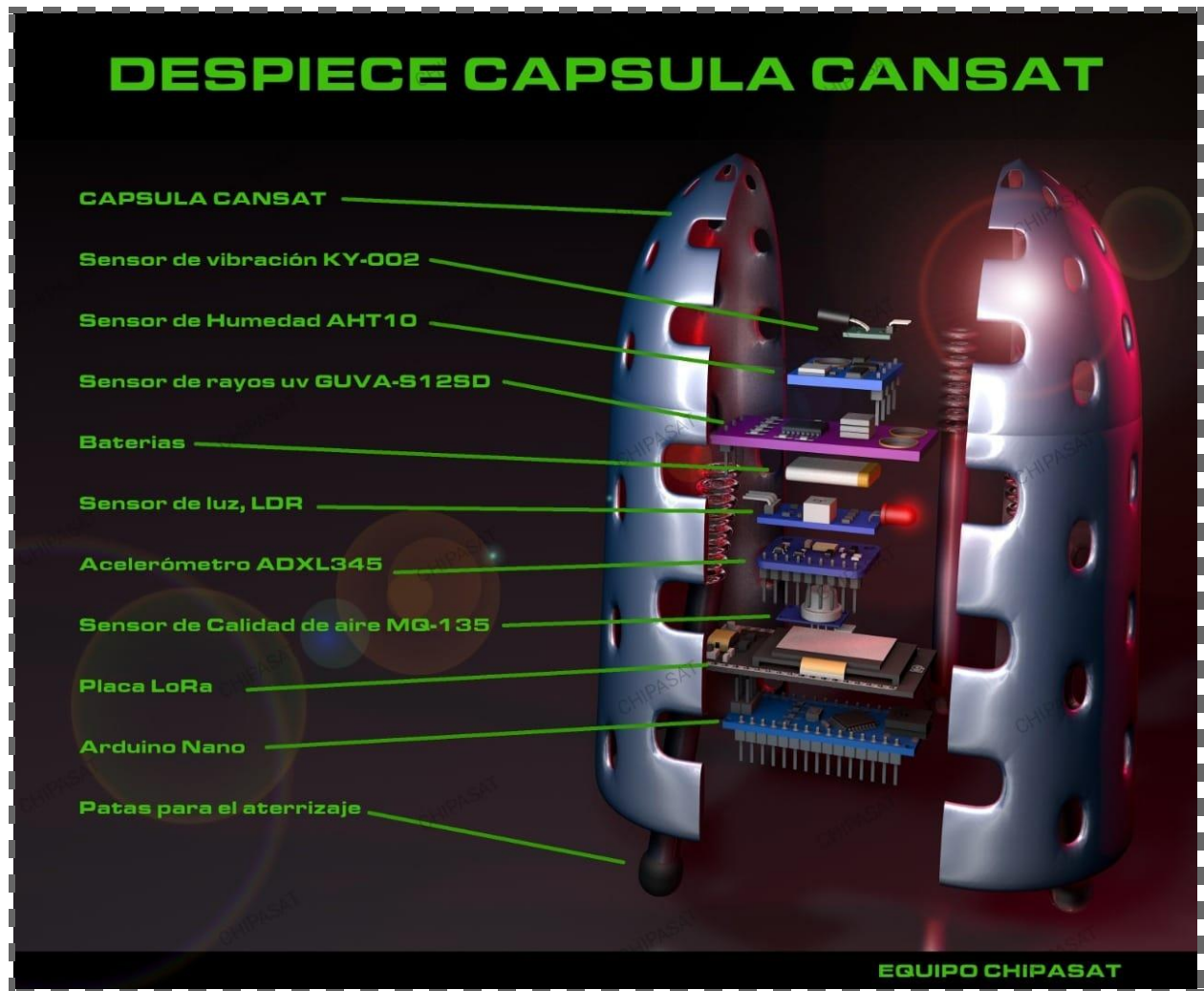
La misión secundaria que ideamos consiste, esencialmente, en la búsqueda de ambientes habitables por el ser humano en exoplanetas que fueron previamente analizados.

Buscamos crear un satélite que de ser enviado a estos planetas, potencialmente Pseudo-tierras, pueda hacer un análisis y paneo general sobre las condiciones atmosféricas, capa de ozono (si las hay), y otros factores que se detallarán a lo largo del documento.

Se debe comprender que el modelo que se presentará en el concurso del Cansat es meramente un prototipo de lo que podría llegar a ser si se utiliza para la finalidad propuesta. Por lo tanto, podrá y deberá ser sometido a algunos cambios menores si se deseara utilizar para enviarlo a otro planeta.

### **Diseño preliminar (Layout físico)**

Partes y componentes del prototipo:



### **Cómo funciona: Operaciones**

La carga útil, una vez en el aire e iniciando el descenso, se mantendrá constantemente midiendo los datos de su entorno, utilizando los sensores que posee, para que esta información sea posteriormente transmitida a la estación terrena, donde será procesada y plasmada en gráficos. Ambas acciones, se realizarán de manera automática, gracias a la programación del satélite.

Mediante los componentes del satélite y los resultados obtenidos, se podrá determinar, si un humano soporta las condiciones del ambiente en exploración (indicios como presencia de agua, buena presión, temperatura, capa de ozono), o si estas son nocivas para el mismo debido a extremas temperaturas o presiones, gases nocivos, ausencia de agua, mucha actividad sísmica o radiación.

El conjunto del satélite con la estación terrena, se encargaran de este proceso y de proveer la respuesta acerca del tema tratado. En las siguientes hojas se detallarán los componentes y el conjunto de operaciones a realizar.

### **Subsistemas**

El satélite solo presenta el arduino, junto a su respectivo módulo LoRa y sensores.

### **Comunicaciones**

Optamos por utilizar el método de comunicación brindado en el Kit de Cansat, es decir el LoRa. Esto se debe a que es más barato, ligero y se adapta más a los términos propuestos por el concurso.

No buscamos enviar datos desde la estación terrestre, sino exclusivamente recibirlos. Estos datos posteriormente los almacenaremos en una base de datos, con los cuales podremos ver las condiciones climáticas y atmosféricas registradas durante el descenso del satélite. Analizamos la posibilidad de realizar una serie de gráficos que muestran la fluctuación de los datos censados sobre el tiempo.

### **Sensores**

Además del sensor provisto en el kit (que mide presión y temperatura), mediremos una serie de variables atmosféricas, geológicas y ambientales con los sensores que se detallan a continuación:

- Sensor de rayos uv, modelo GUVB-S12SD.
  - Los rayos UV nos sirven para detectar si la atmósfera filtra o no estos rayos, y así a su vez determinar la presencia de una capa de ozono. También se utilizará para saber si afectará en el humano esta radiación.
- Acelerómetro, modelo ADXL345.
  - Mediremos la aceleración para ver el estado de la caída del satélite
- Sensor de calidad de aire, modelo MQ-135.
  - La medición de la calidad del aire sirve para determinar si el ambiente contiene, o no, gases nocivos para el humano.
- Sensor de vibración, modelo KY-002.
  - Este sensor, una vez en tierra, servirá para saber si hay cierta actividad sísmica en dicho planeta.
- Sensor de humedad, modelo AHT10.
  - Medir humedad en el aire nos es útil para saber si hay presencia de agua, y un posible ciclo de la misma.
- Sensor de luz, LDR
  - Saber la intensidad de la luz, primero nos servirá para detectar cuando el satélite ha salido del cohete, y segundo, si el ambiente recibe luz de su estrella.

### **Unidad de potencia eléctrica:**

Para nuestro satélite, necesitaremos 2 baterías LiPo de 3.7v, para que al ponerlas en serie, tengamos el doble de voltaje, y así nos alcance para alimentar el arduino nano, el

cual tiene un voltaje de funcionamiento mínimo de 5v. En caso de que no nos puedan dar las dos baterías, emplearíamos un conversor DC-DC step-up.

Periférico	Vin	Cin	Fuente
LORA 32U4 II	3.5–5.5V	- 90mA (en transmisión) - 3.9uA(en modo sleep)	<a href="#">Link</a>
ADXL345 (Acelerómetro)	150µA	3,6v(max)	<a href="#">Link</a>
MQ-135 (Calidad de aire)	2,5 - 5v	150mA	<a href="#">Link.</a>
GUVA-S12SD (Sensor de rayos ultravioleta)	2.5v - 5.5v	5mA	<a href="#">Link</a>
KY-002 (Vibraciones)	3.3-5.0v	-0mA en estado natural. - 0,5mA en caso de vibraciones.	<a href="#">Link</a>
AHT10 (Sensor de temperatura)	1.8~3.6v	23µA	<a href="#">Link</a>
LDR	5v	0,5mA	-
Mecanismo de despliegue	7v	150mA (durante solo un segundo)	-
Arduino Nano	5v	15mA	<a href="#">Link</a>

Dividiremos el consumo en 2 partes, standby y caída

Consumo total = consumo en stand-by + consumo en caída

+ Consumo en standby =  $5v \cdot 15mA = 75 \text{ mW}$

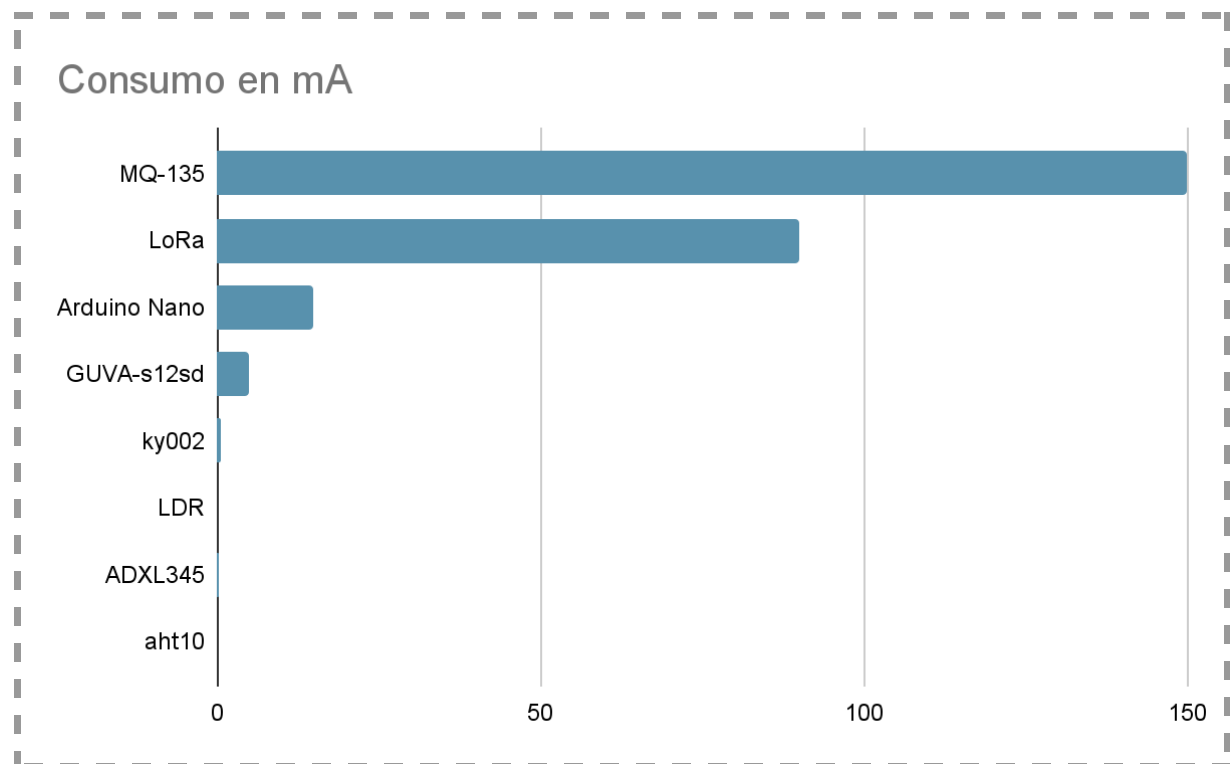
Para el consumo en stand-by consideramos únicamente el consumo del arduino Nano en reposo ya que el resto de los periféricos van a estar apagados.

+ Consumo en caída:

Mecanismo de despliegue ( $7v \cdot 150mA$  durante 1s por única vez, equivalentes a aproximadamente 68 ppm del total de la batería, despreciable por su breve tiempo al total de energía que pueden aportar las baterías).

El consumo del resto de periféricos que estarán encendidos durante toda la caída:

- 150mA (MQ-135)
- 90mA (Lora)
- 15mA (Arduino Nano)
- 5mA (GUVA-S12SD)
- 0,5mA (ky002)
- 0,1mA (LDR)
- 150μA (ADXL345)
- 23μA (AHT10)



> Corriente de suma de periféricos  $\approx$  260mA

> Con este consumo las baterías durarán aproximadamente 4hs

### **Control de descenso**

Pensamos en un método sencillo y eficaz, donde el descenso sea realizado por un paracaídas; Esto, además de asegurar la integridad de la carga, nos brinda información acerca de la atmósfera que presenta el planeta. Cuando el satélite toque tierra, con fin de mantener la integridad y estabilidad del mismo, se desplegarán 3 puntos de apoyo.

El diseño de nuestro paracaídas, se realizará en base a otros paracaídas de juguete de menor escala que poseemos, mediante la utilización de la regla de 3 y otros cálculos para poder sacar la correcta escala correspondiente al peso de nuestro satélite. Además, contamos con una regla de medidas para paracaídas, obtenida de esta página, <http://jcoppens.com/globo/teoria/caida.php>, la cual nos permitirá verificar o corregir los cálculos que obtuvimos en base a los modelos de juguete que tenemos.

### **Sistemas mecánicos**

En el satélite aplicamos un solo sistema mecánico, con el fin de abrir las patas que servirán de apoyo para la carga útil. Se utilizarán patas con resortes, las cuales se encontrarán retenidas dentro del satélite, mediante un cable que las rodee. Tiempo después de que el paracaídas empiece a funcionar, el sistema de patas se desplegará cuando el cable que las rodea se corte, mediante un pequeño circuito con un transistor. Al ocurrir esto, los propios resortes de las patas harán que se desplieguen por la fuerza que hacen, permitiendo tener así un aterrizaje controlado sin importar la superficie.

### **Estructuras**

Con respecto a los materiales estaremos utilizando material de impresión 3D, es decir ABS, para el layout; Esto es así debido al peso, practicidad y volatilidad que nos brinda a la hora de realizar el diseño. Como la carga útil no será sometida a altas temperaturas ni a radiación, consideramos que este material es óptimo para la fabricación del satélite.

### **Presupuesto de masas** (cuadro donde se detallan todos los componentes y su peso)

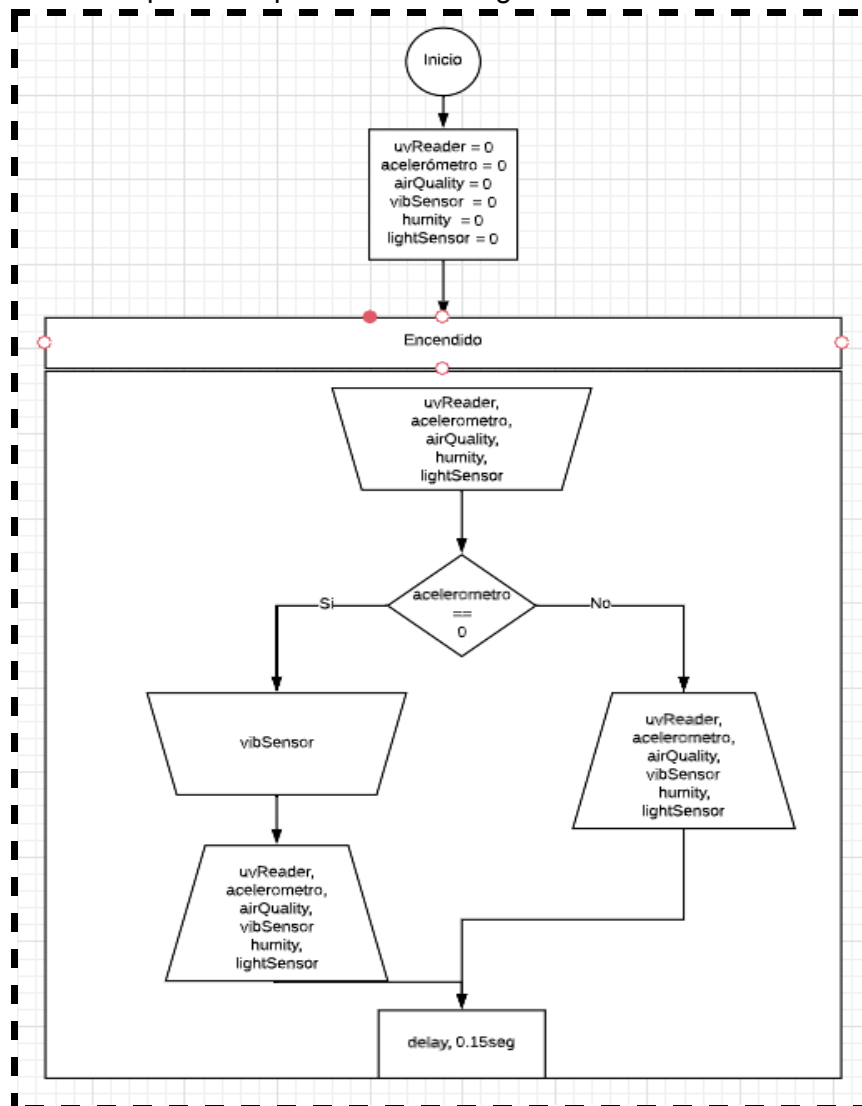
Componente	Peso	Cantidad de unidades
Arduino Nano	7 gramos	1
Placa de desarrollo LORA 32U4 II + antena	20 gramos	1
Acelerómetro ADXL345	6 gramos	1
Sensor rayos UV GUVVA-S12SD	4,5 gramos	1
Sensor calidad de aire MQ-135	10 gramos	1
Sensor de vibración KY-002	4 gramos	1
Sensor de humedad AHT10	2 gramos	1
Sensor de presión y temperatura BMP180	3 gramos	1
Sensor de luz LDR	4 gramos	1



Baterias LiPo	8 gramos	2
Carcaza	50 gramos	1
Cables	1 gramo	25 (Puede variar)
Tornillos	1 gramo	20 (Puede variar)
Total	<b>171,5 gramos</b>	

## Software

El software utilizado para llevar a cabo la tarea del satélite es Arduino. Como consecuencia, el lenguaje de programación a utilizar es C, es decir el lenguaje en que el software se basa para interpretar las órdenes. Consideramos Arduino como la plataforma ideal debido a la sencillez en su sintaxis, conveniente comunicación entre los sensores y la placa, y por último la experiencia previa de los integrantes sobre este software.



## Estación terrena

### Configuración:

La unidad central de procesos que utilizaremos será una computadora conectada al módulo LoRa de manera continua, con el objetivo de recibir la información enviada por el satélite y, posteriormente, utilizarla para hacer un gráfico con estos mismos valores.

La antena será, como ya mencionamos, el LoRa. Encargado de recibir la información y enviarla a uno de los puertos (COM) del pc. La información será enviada en tiempo real y con un intervalo de 0.15 segundos.

Utilizaremos el lenguaje de programación Python para transformar estas mediciones arbitrarias en algo significativo para las personas encargadas de analizar el ambiente del exoplaneta.

Las conexiones serán bastante simples, ya que solo debemos contar con una PC y el módulo previamente mencionado.

Los datos entrantes en la estación terrena serán importantes para el cumplimiento de la misión secundaria, debido a que con estos se harán las hipótesis correspondientes acerca de que si el ambiente es posiblemente habitable por un ser humano o no.

El código de la estación terrena es el siguiente:

```
Lectura de datos.py > ...
1  import serial,time,collections
2  import matplotlib.pyplot as plt
3  import matplotlib.animation as animation
4  from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
5  from threading import Thread
6  from tkinter import Tk, Frame, StringVar, Label, Button, Entry
7
8  isReceiving= False
9  isRun = True
10 datos = 0.0
11 muestraD = 100
12 data = collections.deque([0]*muestraD, maxlen=muestraD)
13 xmin = 0
14 xmax = muestraD
15 ymin = -5
16 ymax = 5
17
18 #"/dev/tty/USB0"
19
20 try:
21     arduino = serial.Serial("COM6", 9600 , timeout=1)
22 except:
23     print("Error de coneccion con el puerto")
24
25
26
27 def Iniciar():
28     global datos
29     global isReceiving
30     global isRun
31     isReceiving = True
32     isRun = True
33
34     thread.start()
35     anim = animation.FuncAnimation(fig, plotData, fargs=(muestraD,lines),interval = 100, blit = False )
36     plt.show()
37
```

```

Lecturadedatos.py > ...
36     plt.show()
37
38
39 def DatosA():
40     time.sleep(1)
41     arduino.reset_input_buffer()
42
43     while (isRun):
44         global isReceive
45         global datos
46         datos = float(arduino.readline().decode('utf-8'))
47         isReceive = True
48
49 def plotData(self,muestraD,Lines):
50     data.append(datos)
51     Lines.set_data(range(muestraD), data)
52
53     labelx.set("VOL:" + str(datos))
54
55 thread = Thread(target = DatosA)
56
57 def Salir():
58     global isRun
59     isRun = False
60     thread.join()
61     arduino.close()
62     time.sleep(1)
63     raiz.destroy()
64     raiz.quit()
65     print("proceso finalizado")
66
67
68 def Terminar():
69
70     global isRun
71     global isReceiving
72     isRun = False
73     isReceiving = False
74
75     time.sleep(0.5)
76     thread.join(timeout=0.3)
77     arduino.close()
78     datos=00.0
79
80 # , figsize=(6, 5) tamaño / , dpi=75 zoom / plt.cla() borra nombres x e y /
81
82 fig = plt.figure(facecolor="0.55",figsize=(6, 4), clear=True, dpi=100)
83 ax = plt.axes(xlim=(xmin,xmax),ylim=(ymin,ymax))
84 plt.title("Grafica - 0 - 5 Voltios",color='red',size=16, family="Tahoma")
85 ax.set_xlabel("Muestras")
86 ax.set_ylabel("Señal")
87
88 lines = ax.plot([],[], 'r')[0]
89
90 def Limpiar():
91     fig.clf()
92
93
94 raiz = Tk()
95 raiz.protocol("WM_DELETE_WINDOW",Salir)
96 raiz.config(bg = "black")
97 raiz.title(" \t\t\t\t GRAFICA SEÑAL ANALOGICA")
98 raiz.geometry("738x402")
99 raiz.resizable(1,1)
100
101 lienzo = FigureCanvasTkAgg(fig, master = raiz )
102 lienzo._tkcanvas.grid(row = 0,column = 0, padx = 1,pady = 1)
103
104 frame = Frame(raiz, width = 130,height = 402, bg = "#7003FC")
105 frame.grid(row = 0,column = 1, padx = 1,pady = 2)
106 frame.grid_propagate(False)
107 frame.config(relief = "sunken")

```

```

main.py  Lecturadedatos.py X
Lecturadedatos.py > ...
102 lienzo._tkcanvas.grid(row = 0, column = 0, padx = 1, pady = 1)
103
104 frame = Frame(raiz, width = 130, height = 402, bg = "#7003FC")
105 frame.grid(row = 0, column = 1, padx = 1, pady = 2)
106 frame.grid_propagate(False)
107 frame.config(relief = "sunken")
108 frame.config(cursor = "heart")
109
110 labelx = StringVar(raiz, "VOL: 0.00")
111
112 label = Label(frame, textvariable = labelx, bg = "#5CFE05", fg = "black", font = "Helvetica 13 bold", width = 11, justify = "center")
113 label.grid(row = 0, column = 0, padx = 5, pady = 8, pady = 10)
114
115 Iniciar = Button(frame, command = Iniciar, text = "Iniciar ", bg = "blue", fg = "white", font = "Helvetica 14 bold", width = 9, justify = "center")
116 Iniciar.grid(row = 1, column = 0, padx = 5, pady = 5)
117
118 terminar = Button(frame, command = Terminar, text = "Terminar", bg = "blue", fg = "white", font = "Helvetica 14 bold", width = 9)
119 terminar.grid(row = 2, column = 0, padx = 5, pady = 5)
120
121 limpiar = Button(frame, command = Limpiar, text = "Limpiar ", bg = "blue", fg = "white", font = "Helvetica 14 bold", width = 9, justify = "center")
122 limpiar.grid(row = 3, column = 0, padx = 5, pady = 5)
123
124 salir = Button(frame, command = Salir, width = 9, text = "SALIR", bg = "red", font = "Helvetica 14 bold", justify = "center")
125 salir.grid(row = 4, column = 0, padx = 5, pady = 125)
126
127 raiz.mainloop()

```

## Integración y pruebas

Se hará una comprobación individual de cada sensor para verificar que todos funcionen correctamente y asegurarnos de que los datos medidos vayan a ser tan precisos como sea posible.

El paracaídas, las patas y la carcasa serán probados varias veces arrojándolos del quinto piso de nuestra escuela. (Lo tiraremos en el patio interior de la escuela asegurándonos de que no haya nadie en este para evitar problemas).

## Secuencia de integración

El proceso de armado del Cansat inicia con la impresión 3D de la estructura. Debido al diseño elegido, vamos a realizar dos partes, para poder manipular de una mejor manera el interior del mismo. Una vez tengamos la carcasa impresa, pasaríamos al ensamblaje de los componentes electrónicos (Hardware en general), con sus respectivas conexiones.

Tras realizar el ensamblaje interno, pasaríamos al armado e integración de las patas, cerrar la carcasa (la cual irá atornillada) y por último, colocar el paracaídas en su respectivo lugar.

## Pruebas de integridad

Haremos nuevamente la prueba de arrojarlo desde altura para ver que ningún cable o componente tenga problemas por las fuerzas resistidas. Además de esto, probaremos la comunicación de LoRa con la estación terrena dejando el satélite nuevamente en el quinto piso y colocando nuestra estación terrena en el patio interior del colegio, en esta prueba evaluaremos de forma tanto individual como grupal las transmisiones (Las mediciones de cada sensor y las de todo el conjunto correspondiente).

### **Plan de validación**

Dado el diseño a emplear, las únicas evaluaciones que haremos serán altura y peso. El profesor de la institución Pablo Chiarelli, encargado de los laboratorios de físico/química del colegio, puso a nuestra disposición una balanza digital con una sensibilidad de 0,01 gramos, con la que realizaremos mediciones sobre el peso del satélite y sus componentes. Además, en los talleres contamos con distintos elementos de medición para realizar las evaluaciones correspondientes

### **Operación de la misión**

Secuencia de operaciones:

1. El satélite es expulsado del cohete
2. El sensor de luz LDR detecta un cambio grande en la iluminación
3. Los sensores de temperatura y presión, humedad, rayos UV y el acelerómetro se activan
4. El paracaídas y las patas se despliegan, mientras el satélite desciende
5. Comienzan a enviar a la estación terrena los datos 4 veces por segundo
6. La estación terrena empieza a recibir los datos
7. El software en la estación terrena toma los datos y comienza a procesarlos y escribirlos en los gráficos
8. Cuando el satélite toque tierra, el acelerómetro lo detectará, avisará a la estación terrena
9. El sensor de vibración se activará y comenzará a enviar lo que registre a la estación terrena para que sea graficado.

### **Roles y responsabilidades**

- Encargado de la estación terrena  
Aaron Mainero
- Encargado del desarrollo de estructura y diseño  
Valentín Ferrari
- Encargado de Hardware  
Luka Regner
- Encargado de Software  
Bautista Rodriguez
- Encargado de análisis bioquímico ambiental  
Román Martinez

**Presupuesto**

Cuadro:

Material	Costo (Aproximado)	Cantidad de unidades
Arduino Nano	\$2000	1
Acelerómetro ADXL345	\$500	1
Sensor rayos UV GUVA-S12SD	\$830	1
Sensor calidad de aire MQ-135	\$600	1
Sensor de vibración KY-002	\$219	1
Sensor de humedad AHT10	\$519	1
Sensor de luz LDR	\$260	1
Baterías LiPo	\$1000	2
Filamento para impresión	El Instituto Padre Elizalde comprará el filamento para los equipos	-
Tornillos	\$5	20
Hilo de paracaídas (Tanza)	El Instituto Padre Elizalde comprará el rollo para los equipos del mismo	-
Transistor	\$80	2
Total	\$7200	

**Capacidades**

Venimos de una escuela técnica en la cual contamos con talleres de soldadura y todas las herramientas para poder llevar a cabo el diseño y construcción del satélite. Estamos capacitados para desarrollar todas las tareas necesarias como puede ser programación del arduino, soldado de cables y componentes, diseño de la estructura, posterior impresión de la misma y probar finalmente el satélite y su correcto funcionamiento.

En cuanto a materiales necesitamos los sensores del kit para poder llevar a cabo el armado físico del proyecto. Para la misión secundaria compraremos los sensores específicos que nos harán falta.

## **Agenda**

Tabla:

Tareas	Duración aproximada	Fecha inicio	Fecha límite	Dependencia
Capacitación Virtual	4 días	06/06/22	09/06/22	-
Escritura y presentación del proyecto de satélite	22 días	10/06/22	01/07/22	Capacitación Virtual
Construcción del satélite	65 días	16/07/22	18/09/22	Escritura y presentación del proyecto de satélite
Pruebas, verificaciones del satélite	65 días	16/07/22	18/09/22	Escritura y presentación del proyecto de satélite
Validación del satélite	5 días	19/09/22	23/09/22	Pruebas, verificaciones del satélite, Construcción del satélite
Campaña de lanzamiento	7 días	04/10/22	10/10/22	Validación del satélite

Diagrama:

[Diagrama de Gantt](#)

## **Plan de documentación**

Pensamos registrar el proceso de construcción mediante fotos y algunos videos, parte de los procesos y logros serán subidos a nuestra cuenta de instagram.

## **Plan de publicacion y difusion de resultados**

Para dar a conocer nuestro proyecto y cómo vamos con los avances, contamos con una cuenta en la red social Instagram, @chipasat, donde estaremos subiendo y comentando el progreso del proyecto. Además, pensamos publicar los resultados obtenidos en la revista “Viva el Oeste”, para tener un mayor alcance en nuestra localidad. Además a través del sistema de gestión institucional, estaremos comunicando a las familias de los alumnos del colegio los avances de la misión.

## **Video**

[Presentación Chipasat](#)