



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH  
Escola d'Enginyeria de Barcelona Est

TRABAJO DE FINAL DE GRADO

**Grado en Ingeniería de la Energía**

**DISEÑO Y MONTAJE DE UNA SULFATADORA  
MÓVIL**



**Memoria y Anexos**

**Autor/a:** Álvaro Llamas Voltés  
**Director/a:** José López López  
**Convocatoria:** Enero 2022



## **Resumen**

Con este documento se pretende mostrar los pasos seguidos durante el diseño, con SolidWorks, y el montaje de una sulfatadora móvil. Se realizan los cálculos de dimensionamiento de componentes como la bomba de sulfatación, la batería a utilizar y el sistema de transmisión. Se presenta también el sistema de control utilizado, Arduino, así como el código diseñado y la APP implementada con MIT APP Inventor. Por último se presenta el coste del proyecto, así como las conclusiones.

***Palabras clave:** diseño, sulfatadora, dimensionamiento, bomba, batería, Arduino.*

## Resum

Amb aquest document es pretén mostrar els passos seguits durant el disseny, amb SolidWorks, i el muntatge d'una sulfatadora mòbil. Es realitzen els càlculs de dimensionament de components com la bomba de sulfatació, la bateria a utilitzar i el sistema de transmissió. Es presenta també el sistema de control utilitzat, Arduino, així com el codi dissenyat i l'APP implementada amb MIT APP Inventor. Finalment es presenta el cost del projecte, així com les conclusions.

***Paraules clau:** disseny, sulfatadora, dimensionament, bomba, bateria, Arduino.*

## **Abstract**

This document aims to show the steps followed during the design, with SolidWorks, and the assembly of an automated sulphating machine. Sizing calculations are made for components such as the sulphation pump, the battery to be used and the transmission system. It also presents the control system used, Arduino, as well as the code designed and the APP implemented with MIT APP Inventor. Finally, the cost of the project is presented, as well as the conclusions.

***Key words:*** *design, sulphating machine, sizing calculations, pump, battery, Arduino.*





## **Agradecimientos**

Me gustaría agradecer el apoyo de mi familia y amigos durante todo el proceso de montaje del sistema, y en especial a mi padre, que me ha acompañado en todo momento dando su opinión al respecto y ayudándome a ver diferentes perspectivas a los problemas que han ido surgiendo durante el proceso de montaje.





# Índice

<b>RESUMEN</b>	<b>I</b>
<b>RESUM</b>	<b>II</b>
<b>ABSTRACT</b>	<b>III</b>
<b>AGRADECIMIENTOS</b>	<b>V</b>
<b>1.- INTRODUCCIÓN</b>	<b>1</b>
<b>2.- LA SULFATACIÓN</b>	<b>3</b>
<b>3.- DIMENSIONAMIENTO</b>	<b>5</b>
3.1.- Cálculo bomba de sulfatación .....	5
3.2.- Cálculo Motor.....	9
3.3.- Cálculo Batería .....	13
3.4.- Cálculo de la sección para el cableado.....	16
3.5.- Cálculo del sistema de carga .....	17
<b>4.- DISEÑO EN SOLIDWORKS</b>	<b>19</b>
4.1.- Estructura de apoyo .....	19
4.2.- Estructura de desplazamiento.....	20
4.3.- Estación de carga.....	20
<b>5.- PROGRAMACIÓN EN ARDUINO</b>	<b>23</b>
5.1.- Control Motor .....	23
5.2.- Sensores utilizados .....	27
5.2.1.- Voltaje de batería. ....	28
5.3.- Comunicación bluetooth. ....	30
<b>6.- APLICACIÓN EN MIT APP INVENTOR</b>	<b>33</b>
6.1.- Conexión bluetooth con la APP. ....	33
6.2.- Control motor con APP.....	34
6.3.- Monitorización de sensores.....	36
6.4.- Cierre de la APP.....	37
<b>7.- MONTAJE Y CONSTRUCCIÓN</b>	<b>39</b>
<b>8.- MODO DE EMPLEO</b>	<b>47</b>
8.1.- Modo de operación.....	47
8.2.- Condiciones desfavorables.....	47



<b>9.- ANÁLISIS DEL IMPACTO AMBIENTAL</b>	<b>49</b>
<b>10.- PRESUPUESTO</b>	<b>51</b>
<b>11.- CONCLUSIONES</b>	<b>53</b>
<b>12.- BIBLIOGRAFÍA</b>	<b>54</b>
<b>ANEXO A: COMPONENTES</b>	<b>55</b>
Depósito 200 L.....	55
Base de apoyo.....	55
Ejes verticales.....	56
Ejes horizontales .....	56
Bomba de sulfatación.....	57
Motor eléctrico .....	57
Plato 38 dientes .....	58
Plato 11 dientes .....	58
Rueda.....	59
Batería .....	59
Caja de conexiones.....	60
Perfiles de apoyo .....	60
Carrete para manguera .....	60
Base de anclaje .....	61
Perfiles verticales .....	61
Sistema de bisagra.....	62
Ensamblaje de la máquina de sulfatación .....	62
Ensamblaje de la estación de carga.....	64
<b>ANEXO B: ESQUEMAS DE CONEXIÓN</b>	<b>66</b>
Control Motor.....	66
Voltaje de batería .....	69
Comunicación bluetooth .....	69
<b>ANEXO C: CÓDIGO ARDUINO</b>	<b>71</b>
<b>ANEXO D: RECOPIACIÓN DE CÁLCULOS EN EXCEL</b>	<b>75</b>

## Índice de ilustraciones

<b>Ilustración 1.-</b> Ejemplo de sistema de pulverización agrícola. [2]	3
<b>Ilustración 2.-</b> Sulfatadora de 50L con motor de gasolina. [Fuente Propia]	4
<b>Ilustración 3.-</b> Determinación grafica del coeficiente de fricción. [Fuente Propia]	8
<b>Ilustración 4.-</b> Diagrama de bloques del sistema. [Fuente propia]	9
<b>Ilustración 5.-</b> Gráfica de consumos teóricos para un día de funcionamiento. [Fuente Propia]	15
<b>Ilustración 6.-</b> Estructura de apoyo y deposito. [Fuente Propia]	19
<b>Ilustración 7.-</b> Estructura y componentes de desplazamiento. [Fuente propia]	20
<b>Ilustración 8.-</b> Vista de frente estación de carga. [Fuente Propia]	21
<b>Ilustración 9.-</b> Vista trasera estación de carga. [Fuente Propia]	21
<b>Ilustración 10.-</b> Inclinación de 15°. [Fuente Propia]	22
<b>Ilustración 11.-</b> Inclinación 35°. [Fuente Propia]	22
<b>Ilustración 12.-</b> Definición de los pines del driver para el control motor. [Fuente Propia]	24
<b>Ilustración 13.-</b> Definición del estado de los pines para el control motor. [Fuente Propia]	25
<b>Ilustración 14.-</b> Apartado “void leer_mensaje” en Arduino. [Fuente Propia]	26
<b>Ilustración 15.-</b> Apartado “void leer_mensaje” segunda parte. [Fuente Propia]	27
<b>Ilustración 16.-</b> Sentencia para leer repetitivamente los datos recibidos por bluetooth. [Fuente Propia]	27

<b>Ilustración 17.-</b> Esquema eléctrico del divisor de tensión. [Fuente Propia]	_____	28
<b>Ilustración 18.-</b> Definición de variables para medir voltaje. [Fuente Propia]	_____	29
<b>Ilustración 19.-</b> Definición del pin analógico. [Fuente Propia]	_____	29
<b>Ilustración 20.-</b> Calculo de voltaje con Arduino. [Fuente Propia]	_____	29
<b>Ilustración 21.-</b> Porcentaje asociado al voltaje. [Fuente Propia]	_____	30
<b>Ilustración 22.-</b> Carga de librería y asignación de pines en IDE Arduino. [Fuente Propia]		31
<b>Ilustración 23.-</b> Variables implicadas en el tratamiento de datos mediante bluetooth. [Fuente Propia]	_____	31
<b>Ilustración 24.-</b> Inicio del sistema bluetooth en Arduino. [Fuente Propia]	_____	31
<b>Ilustración 25.-</b> Almacenamiento de datos recibidos mediante bluetooth. [Fuente Propia]		31
<b>Ilustración 26.-</b> Código para el envío de datos. [Fuente Propia]	_____	32
<b>Ilustración 27.-</b> Captura del apartado conexión de la APP. [Fuente Propia]	_____	33
<b>Ilustración 28.-</b> Lista de dispositivos bluetooth. [Fuente Propia]	_____	33
<b>Ilustración 29.-</b> Captura una vez realizada la conexión. [Fuente Propia]	_____	33
<b>Ilustración 30.-</b> Bloques implicados en la sincronización bluetooth. [Fuente Propia]	__	34
<b>Ilustración 31.-</b> Captura de los botones implicados en el control motor. [Fuente Propia]		35
<b>Ilustración 32.-</b> Bloques implicados en el movimiento del motor. [Fuente Propia]	_____	35
<b>Ilustración 33.-</b> Captura de las etiquetas para sensores. [Fuente Propia]	_____	36
<b>Ilustración 34.-</b> Tratamiento de los datos recibidos mediante bluetooth. [Fuente Propia]		37

<b>Ilustración 35.-</b> Botón salir de la APP. [Fuente Propia]	37
<b>Ilustración 36.-</b> Bloques para cerrar la APP y desconectar bluetooth. [Fuente Propia]	38
<b>Ilustración 37.-</b> Perfiles base. [Fuente Propia]	39
<b>Ilustración 38.-</b> Pletina remachada. [Fuente Propia]	39
<b>Ilustración 39.-</b> Pilar en estructura. [Fuente Propia]	40
<b>Ilustración 40.-</b> Remachado del pilar. [Fuente Propia]	40
<b>Ilustración 41.-</b> Base con pilares. [Fuente Propia]	40
<b>Ilustración 42.-</b> Ubicación de barra para ejes. [Fuente Propia]	41
<b>Ilustración 43.-</b> Soldadura de barra para ejes. [Fuente Propia]	41
<b>Ilustración 44.-</b> Montaje de plato en rueda. [Fuente Propia]	42
<b>Ilustración 45.-</b> Montaje de rueda en eje. [Fuente Propia]	42
<b>Ilustración 46.-</b> Ubicaciones de los motores. [Fuente Propia]	43
<b>Ilustración 47.-</b> Perfiles que atraviesan la estructura. [Fuente Propia]	44
<b>Ilustración 48.-</b> Ubicación de la batería. [Fuente Propia]	44
<b>Ilustración 49.-</b> Bomba bajo estructura. [Fuente Propia]	44
<b>Ilustración 50.-</b> Caja de conexiones. [Fuente Propia]	45
<b>Ilustración 51.-</b> Corrugados de cableado. [Fuente Propia]	45
<b>Ilustración 52.-</b> Carrete para manguera. [Fuente Propia]	45

<b>Ilustración 53.-</b> Vista lateral del carrete. [Fuente Propia]	45
<b>Ilustración 54.-</b> Manguera de admisión. [Fuente Propia]	46
<b>Ilustración 55.-</b> Sistema acabado. [Fuente Propia]	46
<b>Ilustración 56.-</b> Plano 3D del depósito de 200 L. [Fuente Propia]	55
<b>Ilustración 57.-</b> Base del sistema. [Fuente Propia]	56
<b>Ilustración 58.-</b> Pilar. [Fuente Propia]	56
<b>Ilustración 59.-</b> Eje de ruedas. [Fuente Propia]	57
<b>Ilustración 60.-</b> Bomba. [Fuente Propia]	57
<b>Ilustración 61.-</b> Motor. [Fuente Propia]	58
<b>Ilustración 62.-</b> Palto. [Fuente Propia]	58
<b>Ilustración 63.-</b> Rueda. [Fuente Propia]	59
<b>Ilustración 64.-</b> Batería. [Fuente Propia]	59
<b>Ilustración 65.-</b> Caja estanca. [Fuente Propia]	60
<b>Ilustración 66.-</b> Perfiles de apoyo. [Fuente Propia]	60
<b>Ilustración 67.-</b> Carrete de manguera. [Fuente Propia]	61
<b>Ilustración 68.-</b> Base de anclaje. [Fuente Propia]	61
<b>Ilustración 69.-</b> Perfiles verticales. [Fuente Propia]	62
<b>Ilustración 70.-</b> Bisagra. [Fuente Propia]	62

<b>Ilustración 71.-</b> Maquina de sulfatación vista 1. [Fuente Propia]	_____	63
<b>Ilustración 72.-</b> Maquina de sulfatación vista 2. [Fuente Propia]	_____	64
<b>Ilustración 73.-</b> Estación de carga vista frontal. [Fuente Propia]	_____	65
<b>Ilustración 74.-</b> Estación de carga vista trasera. [Fuente Propia]	_____	65
<b>Ilustración 75.-</b> Conexionado del motor 1 con la placa Arduino. [Fuente Propia]	_____	67
<b>Ilustración 76.-</b> Conexionado del motor 2 con la placa Arduino. [Fuente Propia]	_____	68
<b>Ilustración 77.-</b> Divisor de tensión y conexión en Arduino. [Fuente Propia]	_____	69
<b>Ilustración 78.-</b> Conexión del módulo bluetooth. [Fuente Propia]	_____	70

## Índice de tablas

<b>Tabla 1.-</b> Datos implicados en el cálculo de la bomba. [Fuente Propia]	6
<b>Tabla 2.-</b> Datos para el cálculo de la potencia del sistema motor. [Fuente Propia]	10
<b>Tabla 3.-</b> Datos para el cálculo de la relación de transmisión. [Fuente Propia]	12
<b>Tabla 4.-</b> Consumo componentes de dimensionamiento para almacenamiento.	14
<b>Tabla 5.-</b> Resumen de las fases de operación.	15
<b>Tabla 6.-</b> Componentes implicados en el control motor. [Fuente Propia]	24



## Índice de ecuaciones

Ecuación 1	6
Ecuación 2	6
Ecuación 3	6
Ecuación 4	7
Ecuación 5	7
Ecuación 6	7
Ecuación 7	8
Ecuación 8	8
Ecuación 9	9
Ecuación 10	9
Ecuación 11	10
Ecuación 12	11
Ecuación 13	11
Ecuación 14	11
Ecuación 15	11
Ecuación 16	12
Ecuación 17	12



---

Ecuación 18	12
Ecuación 19	13
Ecuación 20	14
Ecuación 21	14
Ecuación 22	14
Ecuación 23	16
Ecuación 24	16
Ecuación 25	16
Ecuación 26	17
Ecuación 27	17
Ecuación 28	17
Ecuación 29	17
Ecuación 30	18
Ecuación 31	28
Ecuación 32	49
Ecuación 33	49
Ecuación 34	50

## **1.- Introducción**

Un día realizando un tratamiento a un campo de olivos surge la necesidad de resolver un problema en el método de sulfatación utilizado. Usando un depósito de 50 litros, son muchos los desplazamientos a realizar para tratar todos los árboles, perdiendo tiempo y esfuerzo en algo que podría hacerse más rápido. Por ello apareció la idea de crear un sistema con mayor capacidad, y que no precisase de fuerza física para desplazarlo.

En los siguientes apartados se abordarán los campos necesarios para crear el sistema de mayor capacidad comentado anteriormente. Este sistema debe permitir desplazar una gran cantidad de líquido fitosanitario sin realizar esfuerzo físico la distancia deseada, para poder ser pulverizado.



## **2.- La sulfatación**

La agricultura es la base para llevar una alimentación saludable y a medida que han ido avanzando los años, este campo ha evolucionado haciendo frente a retos como la producción a gran escala, el cambio climático, el uso eficiente de los recursos naturales y las afectaciones a los cultivos por diversas plagas, entre otros. Es esta última dificultad, sobre el control de plagas, lo que provoca en muchas ocasiones el desperdicio de una cosecha entera y la pérdida de los recursos empleados. Por ello es importante hacer frente a estas situaciones de forma rápida y eficiente.

Hasta el momento se han utilizado diferentes métodos de sulfatación a lo largo del tiempo y, sin dejar de lado otras medidas y métodos, aquí se centrará la atención en el control químico de plagas, más concretamente en la maquinaria a utilizar. [1]

Existen diferentes tipos de equipos utilizados en sulfatación dependiendo del estado del producto que se vaya a utilizar. Estos se dividen en:

Pulverizadores: sistemas utilizados para aplicar productos en estado líquido.

Empolvadores: sistemas para tratamientos con productos en estado sólido.

Fumigadores: sistemas empleados en lugares cerrados y para desinfección mediante productos en estado gaseoso.



*Ilustración 1.- Ejemplo de sistema de pulverización agrícola. [2]*

Es este primer concepto de pulverización del que parte este proyecto. Como se observa por la ilustración anterior, estos sistemas son utilizados a gran escala para grandes hectáreas de cultivo. También existen sistemas más pequeños para aquellos agricultores que no precisen de tratar tanta superficie de terreno, como el que se observa en la siguiente ilustración.



*Ilustración 2.- Sulfatadora de 50L con motor de gasolina. [Fuente Propia]*

Se trata de un sistema mucho más simple que para pequeños cultivos debería ser suficiente, pero cuando el agricultor se encuentra en un punto medio entre grande y pequeño, que no tiene suficiente con un depósito de 50 litros y que debe acarrearlo por toda su finca varias veces en un mismo día para tratar todo su terreno, esta capacidad se vuelve pequeña además de cansada. Por ello este trabajo pretende presentar un modelo de sistema de sulfatación de 200 litros de capacidad y semiautónomo, del que no haya que hacer fuerza para desplazar el depósito de un punto a otro y simplemente preocuparse por la pulverización.

### 3.- Dimensionamiento

Para poder realizar el proyecto de forma coherente y realista se deben hacer una serie de cálculos que permitan determinar la dimensión de los componentes más importantes del sistema, como son el motor eléctrico y la batería. En el Anexo D se recogen los cálculos realizados en excel.

#### 3.1.- Cálculo bomba de sulfatación

El aparato encargado de transportar el preparado fitosanitario será la bomba de sulfatación. En este apartado se justificarán los cálculos para dimensionado de este sistema. Estos procedimientos son similares a los aprendidos en la asignatura de Transporte y Distribución de Fluidos Energéticos (TDFE).[3]

El primer paso es contextualizar para conocer los parámetros base que puedan afectar a la bomba, como pueden ser: líquido utilizado, material de la manguera de transporte, distancia recorrida, etc.

Para esta aplicación se utilizará una manguera de 50 metros de largo y 20 mm de diámetro. El material escogido para ésta es el polietileno (PE), un compuesto que, entre otras propiedades, posee una destacable resistencia química y ante la corrosión, punto favorable para este proyecto dado que las mezclas utilizadas, aunque en mayor parte están compuestas por agua, tiene presencia de líquidos fitosanitarios agrícolas.[4]

Por otra parte, se supondrán algunos parámetros básicos de la bomba para realizar el cálculo, como la velocidad de bombeo, supuesta de máximo 15 litros por minuto, y un rendimiento aceptable del sistema del 60 %.

También se incluyen en el cálculo otros parámetros como la gravedad, densidad, etc.

Todos los datos utilizados en este apartado se presentan a modo de resumen en la siguiente tabla:

<b>Longitud del sistema (S)</b>	50	m
<b>Diámetro tubo (D)</b>	20	mm
<b>Velocidad de bombeo (v)</b>	15	lpm

<b>Rendimiento global (<math>\eta</math>)</b>	60	%
<b>Gravedad</b>	9,81	m/s <sup>2</sup>
<b>Densidad del agua (<math>\rho</math>)</b>	998	kg/m <sup>3</sup>
<b>Viscosidad del agua a 20 °C (<math>\mu</math>)</b>	0,001	kg/m·s
<b>Rugosidad absoluta PE (<math>\epsilon</math>)</b>	0,0015	mm

*Tabla 1.- Datos implicados en el cálculo de la bomba. [Fuente Propia]*

Al final de este proceso se habrá determinado una bomba de potencia adecuada para esta aplicación. Los cálculos necesarios se desarrollan a continuación.

Se empieza determinado la rugosidad relativa ( $\epsilon_r$ ). La rugosidad ( $\epsilon$ ) de un compuesto nos ayuda a conocer el espesor medio de las irregularidades que este presenta en el interior de la tubería. El cálculo de la rugosidad relativa es el siguiente:

$$\epsilon_r = \frac{\epsilon}{D} = \frac{0,0015}{20} = 0,000075 \quad \text{Ecuación 1}$$

Este valor será útil para determinar el coeficiente de fricción más adelante.

El siguiente paso es determinar el caudal del sistema, conociendo la velocidad que se pretende de 15 lpm se pueden realizar factores de conversión:

$$Q = 15 \frac{l}{min} \cdot \frac{1 min}{60 s} \cdot \frac{1 m^3}{1.000 l} = 0,00025 m^3/s \quad \text{Ecuación 2}$$

Seguidamente, es importante conocer la sección de la tubería a partir del diámetro:

$$S = \pi \cdot r^2 = \pi \cdot 0,01^2 = 0,0003141 m^2 \quad \text{Ecuación 3}$$



Finalmente, se puede calcular la velocidad del fluido dentro de la tubería gracias al caudal y la sección:

$$v = \frac{Q}{S} = \frac{0,00025}{0,0003141} = 0,7958 \text{ m/s} \quad \text{Ecuación 4}$$

Un concepto importante de este apartado, para conocer la bomba adecuada, es saber que comportamiento tiene el fluido dentro de la tubería, esto se consigue gracias al número de Reynolds. Este valor adimensional, presentado en la siguiente ecuación, nos permite conocer la relación entre el rozamiento y las fuerzas viscosas del fluido.

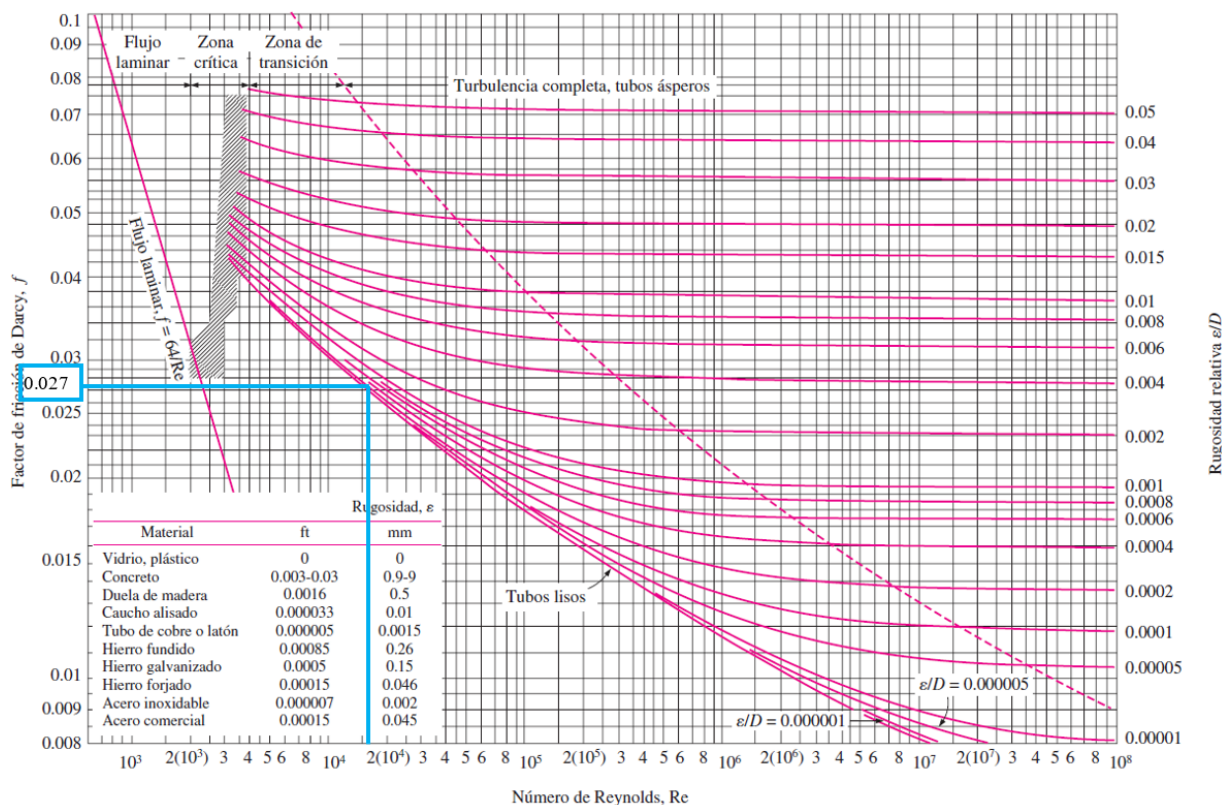
$$Re = \frac{v \cdot D \cdot \rho}{\mu} \quad \text{Ecuación 5}$$

Si se aplica esta fórmula a este sistema se obtiene:

$$Re = \frac{0,7958 \cdot 0,02 \cdot 998}{0,001} = 15.883,6633 \quad \text{Ecuación 6}$$

Dado que este valor es superior a 4.000, se trata de un flujo turbulento. Este concepto hace referencia al movimiento de las partículas de fluido dentro de la tubería, para el flujo turbulento estas partículas se desplazan de manera aleatoria y caótica.

Una vez conocidos los parámetros de rugosidad relativa ( $\epsilon_r$ ) y número de Reynolds (Re), se puede determinar el coeficiente de fricción de forma gráfica gracias al diagrama de Moody, que relaciona estos dos conceptos.



**Ilustración 3.-** Determinación grafica del coeficiente de fricción. [Fuente Propia]

Una vez determinado el coeficiente de fricción (f), de valor 0,027, se calculan las pérdidas de rozamiento expresadas en metros:

$$h_f = f \cdot \frac{L \cdot v^2}{2 \cdot g \cdot D} = 0,027 \cdot \frac{50 \cdot 0,7958^2}{2 \cdot 9,81 \cdot 0,02} = 2,178 \text{ m} \tag{Ecuación 7}$$

Este valor hace referencia a las pérdidas que provoca el rozamiento en el sistema. Para conocer el trabajo que debe desarrollar la bomba se utilizará la ecuación de Bernoulli. La siguiente ecuación relaciona la altura, velocidad y presión del fluido entre dos puntos, entrada y salida.

$$h_1 - h_2 + \frac{u_1^2 - u_2^2}{2 \cdot g} + \frac{p_1 - p_2}{\gamma} \pm W_0 - h_f = 0 \tag{Ecuación 8}$$

Para esta aplicación se considera que el fluido no cambia de altura entre el depósito y el punto de sulfatación, por tanto este elemento se elimina, tampoco hay cambios de velocidad en el circuito ni presión, por lo que la ecuación anteriormente presentada se reduce a:

$$h_1 - h_2 + \frac{u_1^2 - u_2^2}{2 \cdot g} + \frac{p_1 - p_2}{\gamma} \pm W_0 - h_f = 0 \rightarrow W_0 = h_f \quad \text{Ecuación 9}$$

Por lo cual, es fácil determinar que el trabajo a realizar por la bomba será igual a las pérdidas por rozamiento de 2,0173 m.

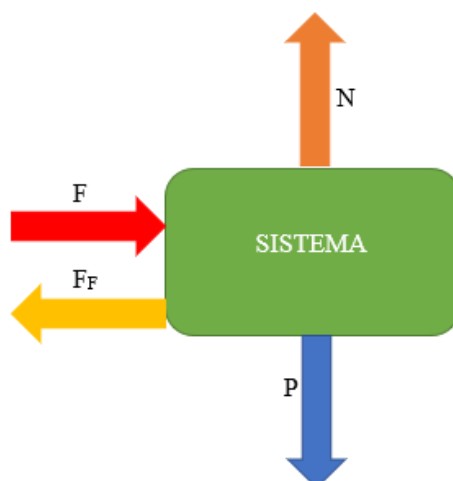
Finalmente, conociendo todos estos conceptos se puede determinar la potencia de bomba necesaria gracias a la siguiente ecuación:

$$N = \frac{Q \cdot W_0 \cdot \rho \cdot g}{75 \cdot \eta} = \frac{0,00025 \cdot 2,178 \cdot 998 \cdot 9,81}{75 \cdot 0,6} = 0,1185 \text{ CV} \quad \text{Ecuación 10}$$

En conclusión, la bomba para este sistema debe tener una potencia no menor a los 0,1185 CV, equivalentes a 87,15 W de potencia.

### 3.2.- Cálculo Motor

Para que el sistema se desplace se deben vencer una serie de fuerzas físicas. En el diagrama siguiente se observan la fuerza normal y el peso, que aparecen debido a la masa de la máquina en contacto con la superficie, la fuerza a ejercer y la fuerza de rozamiento.



**Ilustración 4.-** Diagrama de bloques del sistema. [Fuente propia]

Para el cálculo del motor se parte de una serie de parámetros básicos que se presentarán a continuación. En primer lugar encontramos la masa del sistema. Es importante determinar qué peso se debe desplazar para conocer la potencia que desarrollará el motor. Considerando la masa del depósito de 200 L y los diferentes componentes se determina que la masa total del sistema se encuentra en torno a 300 kg. En segundo lugar la velocidad de desplazamiento y el rozamiento. Se considera un coeficiente máximo de rozamiento de 1, esto quiere decir que la fuerza de rozamiento a vencer será exactamente igual a la fuerza normal que ejerce el sistema, por su propio peso. Dado que es un mecanismo del que no se precisa un desplazamiento rápido, para favorecer a la estabilidad del sistema y permitir que el motor realice más fuerza, se considera que una velocidad de 5 m/min es adecuada. Finalmente también se deben considerar parámetros como el coeficiente de rozamiento estático, la gravedad y la relación de transmisión.

Para esclarecer lo explicado anteriormente, se presentan a continuación una serie de cálculos. Para determinar la potencia del motor se utilizarán los datos que se muestran seguidamente.

<b>Masa del sistema (<math>m</math>)</b>	300	kg
<b>Gravedad (<math>g</math>)</b>	9,81	m/s <sup>2</sup>
<b>Coeficiente estático (<math>\mu_e</math>)</b>	1	
<b>Diámetro del eje (<math>d_e</math>)</b>	0,16	m
<b>Velocidad de desplazamiento (<math>v</math>)</b>	5	m/min
<b>Potencia bomba de sulfatación (<math>P_{sulf}</math>)</b>	204	W

*Tabla 2.- Datos para el cálculo de la potencia del sistema motor. [Fuente Propia]*

Se empieza determinando la fuerza de rozamiento ( $F_F$ ) que debe vencer el sistema:[5][6]

$$F_F = m \cdot g \cdot \mu_e = 300 \cdot 9,81 \cdot 1 = 2.943 \text{ N}$$

*Ecuación 11*

Conociendo esta fuerza de rozamiento ( $F_F$ ) se puede deducir que la fuerza que desarrolle el motor como mínimo debe ser igual a 2.943 N. Por ello, para el cálculo del momento se considera la fuerza a desarrollar como la fuerza de rozamiento, junto con el diámetro del engranaje final, que transmite la fuerza de la cadena a las ruedas gracias a estar sólido a esta última.

$$M = F \cdot d_e = 2.943 \cdot 0,16 = 235,44 \text{ Nm} \quad \text{Ecuación 12}$$

Como último paso antes de determinar la potencia motor, hay que conocer las revoluciones que desarrolla el sistema. En este aspecto considerando la velocidad arbitraria de 5m/min y los 160 mm de diámetro del eje, se dimensiona:[7]

$$n = \frac{v}{d_e \cdot \pi} = \frac{5}{0,16 \cdot \pi} = 9,95 \text{ rpm} \quad \text{Ecuación 13}$$

Finalmente, conociendo el par motor a desarrollar y las revoluciones por minuto del sistema, se puede dimensionar la potencia coherente del motor eléctrico.

$$P_{motor} = M \cdot \frac{n}{60/2\pi} = 235,44 \cdot \frac{9,95}{60/2\pi} = 245,25 \text{ W} \quad \text{Ecuación 14}$$

En conclusión la potencia a desarrollar por el motor eléctrico debe ser igual o superior a los 245 W calculados. Para este proyecto se plantea el uso de motores como los implementados en los patinetes eléctricos, que funcionan a 12 V y pueden administrar 250 W. Para facilitar el sistema de transmisión y giro se utilizarán cuatro de estos motores y, por tanto, la potencia necesaria por el sistema quedara perfectamente satisfecha con margen suficiente para más fuerza en momentos puntuales.

Como punto extra a tener en cuenta, la potencia total del sistema, considerando 4 motores y la bomba de sulfatación seria de 1.185 W.

$$P_{TOTAL} = 4 \cdot P_{motor} + P_{sulf} = 4 \cdot 245,25 + 204 = 1.185 \text{ W}_{pico} \quad \text{Ecuación 15}$$

Otro cálculo a tener en cuenta que permite dimensionar ciertos componentes del sistema, como son los engranajes sólidos al motor eléctrico y a las ruedas, es la relación de transmisión. Para este cálculo se presentan los datos en la tabla siguiente:

<b>Diámetro Motriz (<math>D_m</math>)</b>	31	mm
<b>Diámetro Salida (<math>D_s</math>)</b>	160	mm
<b>Rpm Motriz (<math>N_m</math>)</b>	2.700	rpm

**Tabla 3.-** Datos para el cálculo de la relación de transmisión. [Fuente Propia]

Se puede observar por los datos anteriores que se han escogido dos diámetros de engranajes a motor (motriz) y a ruedas (salida) de forma arbitraria para tener un primer cálculo aproximado. Destacar que las revoluciones de sistema motor (rpm motriz) son las que el fabricante informa en el motor eléctrico utilizado, que se presentara junto a otros componentes más adelante.

Este cálculo de relación de transmisión, en adelante RT, se basa en la concordancia entre el diámetro del engranaje y las revoluciones que desarrolla. Por este motivo, el cálculo de las revoluciones a la salida del sistema es el siguiente:[8]

$$D_m \cdot N_m = D_s \cdot N_s \rightarrow RT = \frac{D_m}{D_s} = \frac{N_s}{N_m} \quad \text{Ecuación 16}$$

Aislando las revoluciones a la salida en la ecuación anterior:

$$N_s = \frac{D_m \cdot N_m}{D_s} = \frac{31 \cdot 2.700}{160} = 523,125 \text{ rpm} \quad \text{Ecuación 17}$$

Finalmente, calculando RT:

$$RT = \frac{D_m}{D_s} = \frac{N_s}{N_m} = \frac{31}{160} = \frac{523,15}{2.700} = 0,19 \quad \text{Ecuación 18}$$

Dado que el resultado es menor a 1, el sistema RT es de reducción y, por tanto, nos favorece en esta aplicación para desarrollar más fuerza y menos velocidad durante las maniobras de desplazamiento. Por otra parte la reducción aplicada se puede calcular como:

$$\text{Reducción} = \frac{1}{RT} = \frac{1}{0,19} = 5,16 \quad \text{Ecuación 19}$$

El valor reducción, indicado en la ecuación 9, hace referencia al número de vueltas que da el eje motor respecto al eje de la rueda, en definitiva cuando un eje motor da 5,16 vueltas, el eje de la rueda ha dado 1 vuelta.

### 3.3.- Cálculo Batería

El componente esencial de este proyecto es la batería. Sin el sistema de almacenamiento el resto de los componentes, como motores y bomba, no podrían funcionar. Dado la importancia de este elemento, acertar con el proceso de dimensionamiento es fundamental. Para obtener un resultado de capacidad de batería (Ah) coherente con la aplicación se propone un día tipo de operación para ejemplificar los consumos de corriente por minuto.

El día tipo estudiado dura aproximadamente 2 horas y 10 minutos, y se puede dividir en 5 procesos diferentes:

Fase 1.- Inicializar los sistemas alimentando los motores y Arduino para monitorizar el estado del depósito y poder dirigir el sistema con la APP. Para este proceso tan simple se considera un tiempo de 1 minuto.

Fase 2.- Desplazarse desde un punto A, lugar inicial, hasta un punto B, zona de sulfatación. Considerando que el sistema se desplaza a una velocidad de 5 m/min y debe desplazarse 100 metros, el tiempo de desplazamiento será de 21 minutos.

Fase 3.- El proceso de sulfatación, que consiste en sulfatar un área de 1.105 m<sup>2</sup> teniendo en cuenta que se tardan 5 segundos en sulfatar 1 metro cuadrado. El tiempo total de esta fase es de aproximadamente 92 minutos.

Fase 4.- Desplazarse de nuevo del punto de sulfatación B al punto de partida A, considerando el mismo tiempo que su equivalente de 21 minutos.

Fase 5.- Apagar los sistemas de desplazamiento y Arduino para estacionar definitivamente el aparato, en un tiempo de 1 minuto.

La tabla siguiente muestra la corriente que consume cada uno de los sistemas implicados en el dimensionamiento de la batería:[9]

Componente	Consumo (A)
Motor eléctrico	20
Arduino Mega	0,093
Bomba de sulfatación	11

**Tabla 4.-** Consumo componentes de dimensionamiento para almacenamiento.

Debe tenerse en cuenta que los motores eléctricos de este proyecto son 4 y, por tanto, el consumo total será de 80 A. Por otra parte, cuando se traten la primera y última fase, de inicialización y paro de sistemas, se considerará un consumo remanente de los motores eléctricos equivalente al 10 % total.

Los cálculos de consumo para cada fase son los siguientes:

$F1 \text{ y } F5 = (0,1 \cdot (20 \cdot 4)) + 0,093 = 8,093 \text{ A}$	<i>Ecuación 20</i>
$F2 \text{ y } F4 = (20 \cdot 4) + 0,093 = 80,093 \text{ A}$	<i>Ecuación 21</i>
$F3 = 11 + 0,093 = 11,093 \text{ A}$	<i>Ecuación 22</i>

La siguiente tabla muestra, a modo resumen, los consumos en cada fase así como la duración de estas.

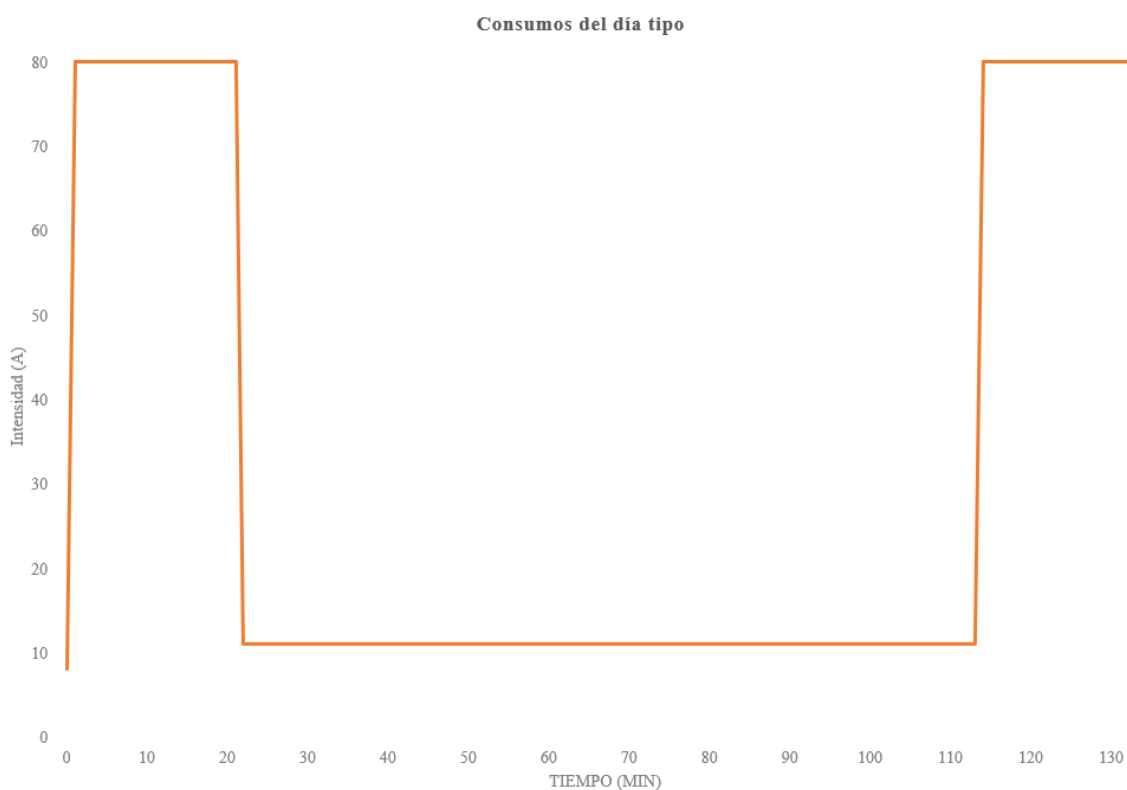
Fase	Consumo (A)	Duración (min)
1	8,093	1
2	80,093	21



<b>3</b>	11,093	92
<b>4</b>	80,093	21
<b>5</b>	8,093	1

**Tabla 5.-** Resumen de las fases de operación.

Por último, se muestra una gráfica resumen con el perfil de consumos de este día de ejemplo. Se puede observar que el momento de máximo consumo se encuentra en los extremos de la gráfica, cuando se desplaza el sistema y, por tanto, consumen energía los motores. Por otra parte, existe un momento de consumo lineal prolongado correspondiente al uso de la bomba de sulfatación.



**Ilustración 5.-** Gráfica de consumos teóricos para un día de funcionamiento. [Fuente Propia]

A partir de los cálculos realizados es fácil determinar la capacidad de la batería que se necesitará. Dado que se conoce el consumo y el tiempo se puede calcular la capacidad necesaria para cada fase. En las fases 1 y 5 con 8,093 A y 1 minuto de operación:

$$C_{F1 F5} = 8,093 \cdot \frac{1}{60} = 0,135 \text{ Ah} \quad \text{Ecuación 23}$$

En las fases 2 y 4 con 80,093 A de consumo y 20 minutos de operación:

$$C_{F2 F4} = 80,093 \cdot \frac{20}{60} = 26,698 \text{ Ah} \quad \text{Ecuación 24}$$

Por último, la fase 3 con 11,093 A y 92 minutos de operación:

$$C_{F3} = 11,093 \cdot \frac{92}{60} = 17,009 \text{ Ah} \quad \text{Ecuación 25}$$

En conclusión, gracias a los cálculos anteriores, se observa como el valor de capacidad máximo que debe soportar nuestra batería es de 17 Ah durante 1h y 30 minutos, y 27 Ah durante aproximadamente 20 minutos. Con el objetivo de asegurar otros posibles escenarios la batería a escoger será de una capacidad mayor a 27 Ah para 2 horas de descarga.

### 3.4.- Cálculo de la sección para el cableado

Para proteger el sistema de posibles fallos provocados por quemaduras de cable y cortocircuitos se incluye este apartado para justificar el uso de diferentes secciones de cableado en función de la aplicación.

En primer lugar se deben identificar las diferentes líneas de cableado del proyecto. Encontramos 2 líneas principales de consumo que destacan: la conexión entre motores y batería, y la conexión entre bomba de sulfatación y batería.

Para la conexión entre motores y baterías se considera una distancia de 2 m, correspondiente a la longitud de cableado, una intensidad de cada motor de 28 A, y una caída máxima admisible del 2 %, que considerando el sistema de 12 V corresponderían a 0,24 V. Conocidos los datos se puede calcular la sección mediante la siguiente ecuación:

$$S = \frac{2 \cdot L \cdot I}{cte_{CU} \cdot \%} \quad \text{Ecuación 26}$$

La ecuación se basa en los parámetros de voltaje, longitud e intensidad comentados anteriormente, además de una constante de conductividad que dependerá del material del cable, en este caso será cobre y la constante corresponde a 56 m/Ω·mm<sup>2</sup>.

De esta manera el cálculo para el primer tramo de cableado entre motores y batería sería el siguiente:

$$S_{MOT\_BAT} = \frac{2 \cdot 2 \cdot 28}{56 \cdot 0,24} = 8,34 \text{ mm}^2 \quad \text{Ecuación 27}$$

Siendo el resultado de 8,34 mm<sup>2</sup>, se buscará un cable de esa sección como mínimo para garantizar la seguridad del sistema.

El segundo tramo de cableado entre bomba y batería tendría las siguientes características: una longitud de cableado aproximada de 2 metros, el consumo de la bomba de 11 A y la caída de tensión máxima del 2%. El cálculo sería el siguiente:

$$S_{BOM\_BAT} = \frac{2 \cdot 2 \cdot 11}{56 \cdot 0,24} = 3,27 \text{ mm}^2 \quad \text{Ecuación 28}$$

Resultando una sección superior a eso 3,27 mm<sup>2</sup> para que no se produjeran cortocircuitos y quemaduras de cable.

### 3.5.- Cálculo del sistema de carga

Para garantizar un sistema totalmente autónomo, que no dependa de la red eléctrica, se diseñará un sistema de carga para la batería seleccionada basado en energía solar. Se selecciona una batería de 12 V y 90 Ah de capacidad, dado que cumple con el dimensionamiento realizado en apartados anteriores.

$$E_{BAT} = 12 \text{ V} \cdot 90 \text{ Ah} = 1.080 \text{ Wh} \quad \text{Ecuación 29}$$

Con los datos anteriores se puede determinar que la potencia necesaria para cargar dicha batería en un periodo de 1 hora debe ser de 1.080 W. Con el objetivo de diseñar una base de carga no excesivamente grande, se propondrá usar un panel de 380 W.

$$Tiempo_{1p\ 380W} = \frac{1.080\ Wh}{380\ W} = 2,85\ horas \quad \text{Ecuación 30}$$

Se observa por el cálculo anterior que cumpliendo los parámetros propuestos, la batería tardaría 2 horas y 50 min, aproximadamente, en cargarse.

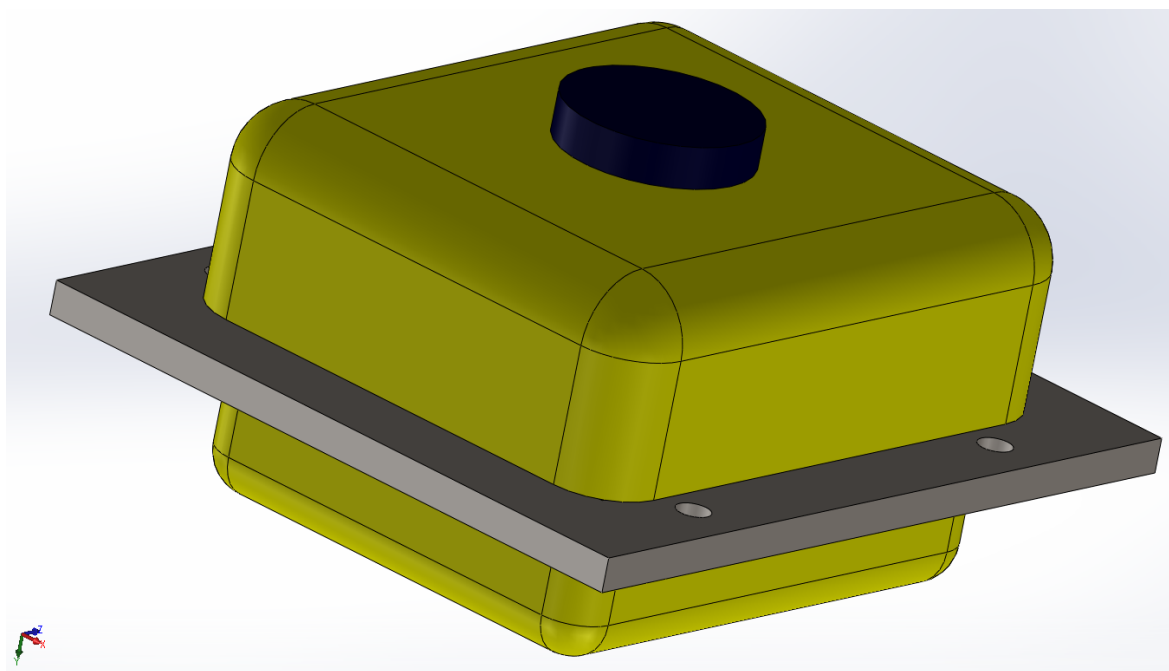
Este periodo de carga es bastante aceptable dado que la operación de estos sistemas se suele realizar por la tarde; el sol de la mañana puede cargar el aparato sin problemas, ya sea invierno o verano, evitando la falta de luz que se produce en los días más cortos del año.

## 4.- Diseño en SolidWorks

El diseño en todo proyecto de ingeniería es un aspecto fundamental para el correcto funcionamiento de la maquinaria. En una primera aproximación el planteamiento del trabajo depende más de un proceso imaginativo y de “*brainstorming*”, pero a medida que se avanza por el proyecto se toman decisiones que, de forma iterativa, van cambiando los componentes y estructuras pensadas en primera instancia. En este apartado se presentan los bloques principales del proceso de diseño, para una visión más concreta de cada pieza véase el anexo A. Estos bloques se dividen en: el sistema de apoyo, el sistema de desplazamiento y la estación de carga.

### 4.1.- Estructura de apoyo

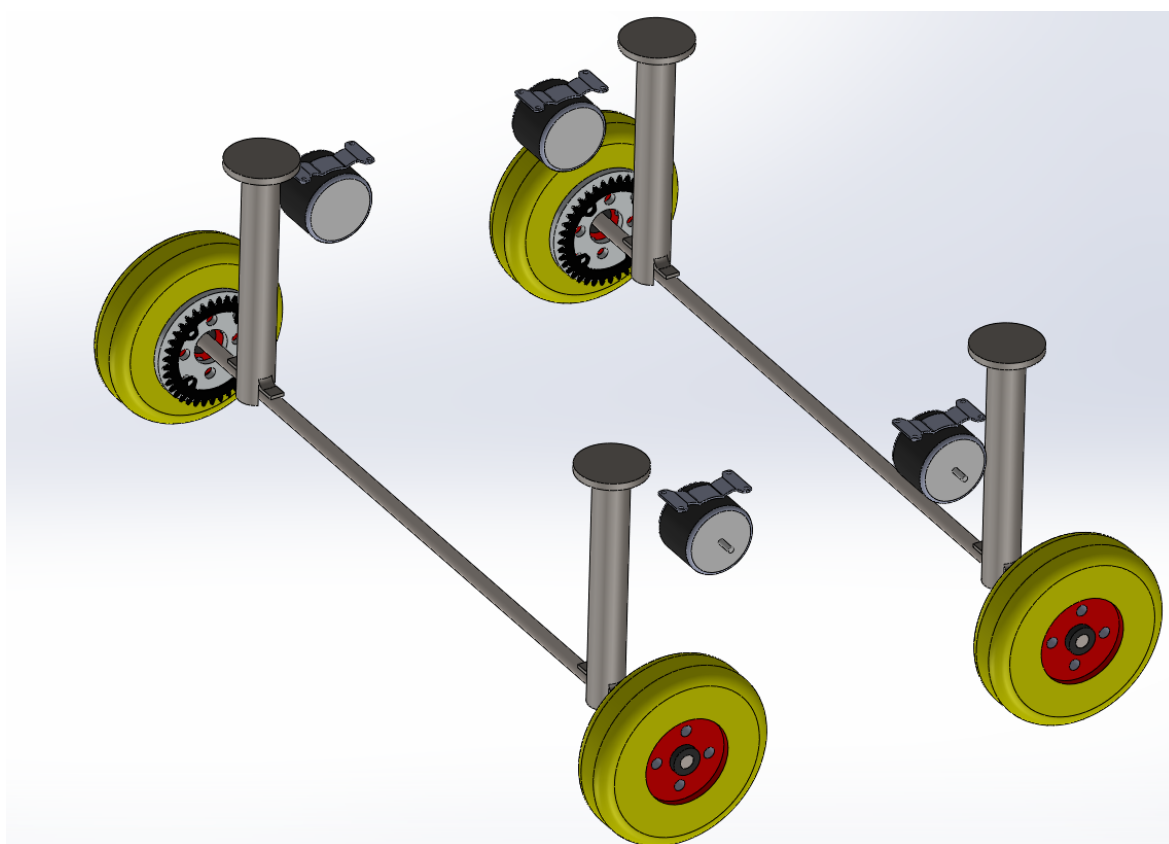
El objetivo de los componentes aquí nombrados está en permitir un apoyo estable del depósito de sulfatación. Como se aprecia en la ilustración, el depósito reposa en la estructura rectangular de aluminio. Por otra parte, la bomba de sulfatación también quedará anclada a esta estructura.



*Ilustración 6.- Estructura de apoyo y deposito. [Fuente Propia]*

## 4.2.- Estructura de desplazamiento

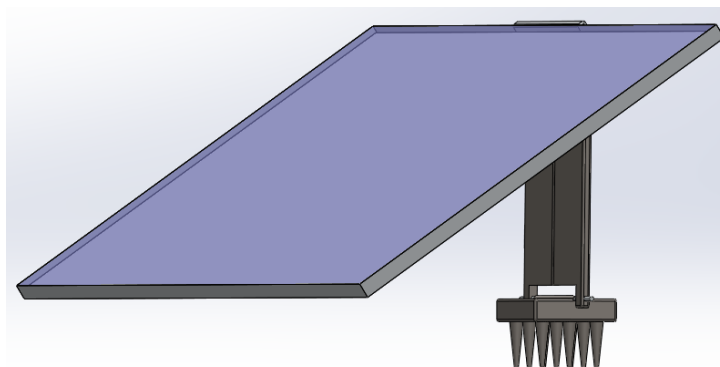
Las piezas contenidas en este apartado permiten al sistema moverse correctamente. Tal y como se observa en la ilustración siguiente, este apartado está compuesto por 5 piezas principales. Las ruedas, en primer lugar, permiten a la estructura moverse y estar en contacto con el suelo. En segundo lugar, encontramos las dos barras de eje que se unen a las ruedas. Seguidamente, los cuatro pilares que quedan asentados a la estructura de apoyo y a las barras de eje. Por último, los motores eléctricos, que quedan sólidos a la estructura de apoyo, y sus correspondientes engranajes, sólidos a las ruedas.



*Ilustración 7.- Estructura y componentes de desplazamiento. [Fuente propia]*

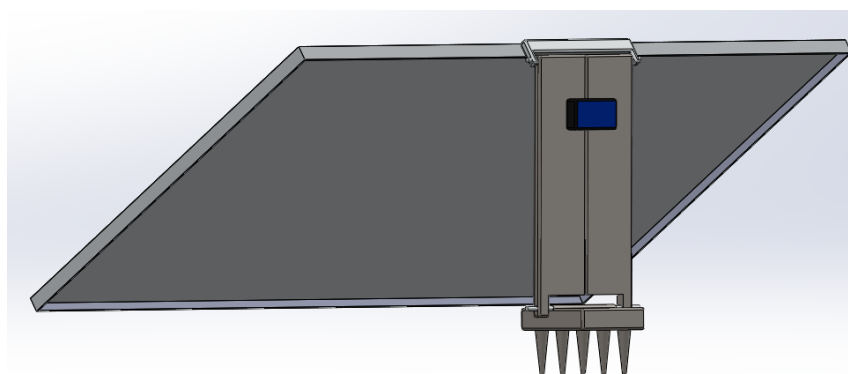
## 4.3.- Estación de carga

Seguidamente se presenta un ejemplo de diseño de estación de carga, cumpliendo con los parámetros especificados en el apartado de dimensionamiento.



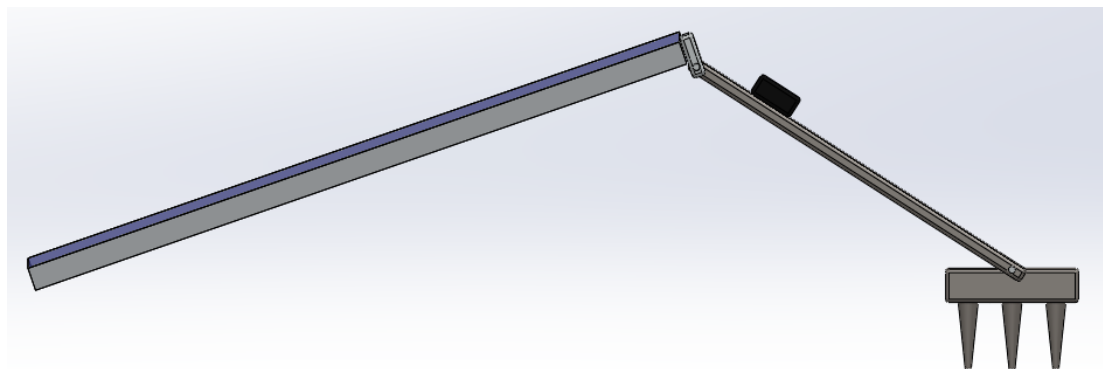
**Ilustración 8.-** Vista de frente estación de carga. [Fuente Propia]

Se observa un sistema sencillo de base de carga clavado al suelo. En primer lugar destaca el panel solar, que está anclado a dos perfiles verticales que lo unen con una pletina de fijación para césped o tierra. También se aprecia en la siguiente imagen la localización del regulador de carga en la parte trasera, esto permitirá acercar la máquina hasta los perfiles verticales y conectar fácilmente la batería.

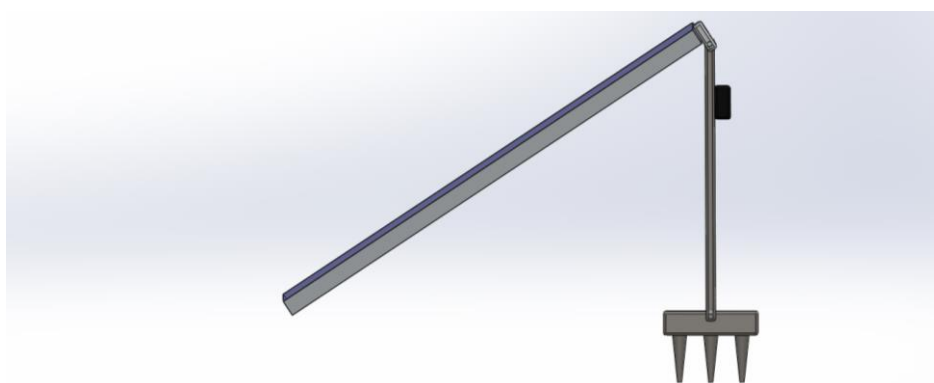


**Ilustración 9.-** Vista trasera estación de carga. [Fuente Propia]

Hay que tener en cuenta que como se oriente y con que inclinación queda a elección del usuario, permitiendo colocar la base haya donde se crea necesario, siempre orientándola al sur, y variando la inclinación dependiendo de la época del año. En la siguiente imagen se observa como el sistema llega permitir inclinaciones entre 15° y 35°.



*Il·lustració 10.- Inclinació de 15°. [Fuente Propia]*



*Il·lustració 11.- Inclinació 35°. [Fuente Propia]*



## **5.- Programación en Arduino**

Todo sistema necesita un método para poder ser controlado y tratar los datos que se puedan generar durante su uso. En este proyecto son muchos los datos que se pueden recopilar, desde caudal de pulverización hasta velocidad de movimiento, pasando por el nivel de mezcla en el depósito, entre otros. Arduino permite la recopilación de todos estos datos gracias a su compatibilidad con gran variedad de sensores y sencilla programación para el control de objetos.

Para esta aplicación se han usado diferentes componentes, entre ellos los dos más importantes son:

- Placa de control Arduino, tarjeta microcontrolador MEGA 2560.
- Módulo Bluetooth HC – 05.

Estos dos componentes permiten procesar la información que contiene el programa creado en IDE Arduino y comunicar el teléfono móvil con el sistema, respectivamente.

Los siguientes apartados muestran la aplicación concreta de Arduino a diferentes controles del sistema. Como información complementaria en el anexo B se incluyen esquemas eléctricos asociados a cada apartado para tener una visión clara sobre las conexiones realizadas y en el anexo C se recoge todo el código aquí comentado.

### **5.1.- Control Motor**

El apartado principal a tratar con Arduino es el movimiento del sistema. En el proceso creativo del proyecto se presentaron diferentes alternativas para el desplazamiento del sistema, finalmente se decidió que el diseño óptimo y adecuado para la aplicación era el uso de cuatro motores eléctricos que controlaran las ruedas de forma independiente. Se podría plantear el sistema de desplazamiento de manera sencilla, colocando botones en la máquina que al ser pulsados accionarían los motores para ir hacia delante, hacia atrás, etc. Pero esto haría del proyecto un sistema basto, poco fiable e incómodo; dado que se debería estar en todo momento cerca de la máquina para su control y se podría favorecer al descontrol del sistema. En este caso el programa creado en Arduino, junto cableado, la placa Mega y el driver específico para el motor eléctrico, se comunica con los motores para su control. La siguiente tabla muestra los componentes implicados en este apartado:

Componente	Unidades
Motor eléctrico	4
Batería 12 V	1
Arduino Mega	1
Driver JZK BTS7960B	2

**Tabla 6.-** Componentes implicados en el control motor. [Fuente Propia]

Seguidamente se presenta el programa utilizado para el control motor. El primer paso en todo programa es definir los pines de la placa microprocesador (Driver JZK BTS7960B) encargados de recibir y enviar información al sistema. En este caso se debe controlar cuatro motores al funcionar en grupos de dos se utilizan dos drivers, uno cada dos motores, que harán de intermediarios entre la placa Arduino y los motores eléctricos. En la ilustración siguiente se muestran los apartados “Motor 1” y “Motor 2”, lógicamente cada uno corresponde a un grupo motor y su driver característico, pero con los mismos conceptos.

```

/*Definición de los pines implicados en el GRUPO MOTOR 1*/

int R_IS_1 = 52; // El pin R_IS se encarga de asegurar que la corriente que pase en el sentido horario no supere el maximo, funciona a modo de alarma.
int R_EN_1 = 48; // El pin R_EN permite habilitar al driver el paso de corriente y, por tanto, el movimiento horario.
int R_PWM_1 = 44; // El pin R_PWM permite ajustar el valor de velocidad al que gira el motor en sentido horario.
int L_IS_1 = 50; // El pin L_IS se encarga de asegurar que la corriente que pase en el sentido horario no supere el maximo, funciona a modo de alarma.
int L_EN_1 = 46; // El pin L_EN permite habilitar al driver el paso de corriente y, por tanto, el movimiento antihorario.
int L_PWM_1 = 3; // El pin L_PWM permite ajustar el valor de velocidad al que gira el motor en sentido antihorario.

/*Definición de los pines implicados en el GRUPO MOTOR 2*/

int R_IS_2 = 49; // El pin R_IS se encarga de asegurar que la corriente que pase en el sentido horario no supere el maximo, funciona a modo de alarma.
int R_EN_2 = 41; // El pin R_EN permite habilitar al driver el paso de corriente y, por tanto, el movimiento horario.
int R_PWM_2 = 45; // El pin R_PWM permite ajustar el valor de velocidad al que gira el motor en sentido horario. // giro hacia derecha
int L_IS_2 = 51; // El pin L_IS se encarga de asegurar que la corriente que pase en el sentido horario no supere el maximo, funciona a modo de alarma.
int L_EN_2 = 47; // El pin L_EN permite habilitar al driver el paso de corriente y, por tanto, el movimiento antihorario.
int L_PWM_2 = 5; // El pin L_PWM permite ajustar el valor de velocidad al que gira el motor en sentido antihorario.

```

**Ilustración 12.-** Definición de los pines del driver para el control motor. [Fuente Propia]

La mayoría de los pines utilizados en la placa Mega para conectarse con el driver son digitales, esto implica que únicamente puedan tomar 2 valores posibles, estado HIGH o estado LOW. Pero en el caso del control de la velocidad es importante poder variar este estado entre diferentes valores para permitir una velocidad adecuada de desplazamiento, es por lo que los pines PWM de cada motor deben ir conectados a pines PWM de la placa Mega, en este caso los pines 44, 45, 3 y 5 cumplen dicha condición.

Los pines PWM de Arduino permiten regular el ciclo de trabajo de aquello que tengan conectados, estos pines se regulan con una resolución de 8 bits, por lo que se puede variar el valor entre 0 y 255, correspondiendo estos a un gradiente de voltaje entre 0 y 5 V. De este modo

si se define un pin PWM con un valor de 255, quiere decir que el ciclo de trabajo es máximo, 100 %, y se estará entregando 5 V, que una vez pasados por el correspondiente driver, y gracias a la batería, permite alimentar el motor con 12 V. Se variarán estos valores PWM para conseguir distintas velocidades de cada motor y permitir, desplazamientos lineales y giros.

Como se comentaba anteriormente, el estado de cada pin se debe definir, por este motivo la ilustración siguiente muestra el código de Arduino del apartado “*void setup*” para esta finalidad.

```

/*Definición de estado de los pines implicados en el GRUPO MOTOR 1*/

pinMode(R_IS_1, OUTPUT); // Se define el pin R_IS como salida, aunque no se va a utilizar en esta aplicación pero se conecta por seguridad.
pinMode(R_EN_1, OUTPUT); // Se define el pin R_EN como salida, dependiendo del estado que se le aplique estará activo o no.
pinMode(R_PWM_1, OUTPUT); // Se define el pin R_PWM como salida, dado que en función del valor que se envíe el motor irá más o menos rápido.
pinMode(L_IS_1, OUTPUT); // Se define el pin L_IS como salida, aunque no se va a utilizar en esta aplicación pero se conecta por seguridad.
pinMode(L_EN_1, OUTPUT); // Se define el pin L_EN como salida, dependiendo del estado que se le aplique estará activo o no.
pinMode(L_PWM_1, OUTPUT); // Se define el pin L_PWM como salida, dado que en función del valor que se envíe el motor irá más o menos rápido.

digitalWrite(R_IS_1, LOW); // Se aplica el estado LOW, y por tanto desconectado, al pin R_IS.
digitalWrite(L_IS_1, LOW); // Se aplica el estado LOW, y por tanto desconectado, al pin L_IS.
digitalWrite(R_EN_1, HIGH); // Se aplica el estado HIGH, por tanto activo, al pin R_EN para permitir el movimiento horario.
digitalWrite(L_EN_1, HIGH); // Se aplica el estado HIGH, por tanto activo, al pin L_EN para permitir el movimiento antihorario.

/*Definición de estado de los pines implicados en el GRUPO MOTOR 2*/

pinMode(R_IS_2, OUTPUT); // Se define el pin R_IS como salida, aunque no se va a utilizar en esta aplicación pero se conecta por seguridad.
pinMode(R_EN_2, OUTPUT); // Se define el pin R_EN como salida, dependiendo del estado que se le aplique estará activo o no.
pinMode(R_PWM_2, OUTPUT); // Se define el pin R_PWM como salida, dado que en función del valor que se envíe el motor irá más o menos rápido.
pinMode(L_IS_2, OUTPUT); // Se define el pin L_IS como salida, aunque no se va a utilizar en esta aplicación pero se conecta por seguridad.
pinMode(L_EN_2, OUTPUT); // Se define el pin L_EN como salida, dependiendo del estado que se le aplique estará activo o no.
pinMode(L_PWM_2, OUTPUT); // Se define el pin L_PWM como salida, dado que en función del valor que se envíe el motor irá más o menos rápido.

digitalWrite(R_IS_2, LOW); // Se aplica el estado LOW, y por tanto desconectado, al pin R_IS.
digitalWrite(L_IS_2, LOW); // Se aplica el estado LOW, y por tanto desconectado, al pin L_IS.
digitalWrite(R_EN_2, HIGH); // Se aplica el estado HIGH, por tanto activo, al pin R_EN para permitir el movimiento horario.
digitalWrite(L_EN_2, HIGH); // Se aplica el estado HIGH, por tanto activo, al pin L_EN para permitir el movimiento antihorario.
    
```

**Ilustración 13.- Definición del estado de los pines para el control motor. [Fuente Propia]**

Una vez se han definido correctamente los pines y estados del sistema, en IDE Arduino se debe programar qué debe hacer cada conexión en el apartado “*void loop*” pero en este caso, dado que el Arduino debe enviar y recibir datos, para no crear una incompatibilidad de la información y permitir la comunicación se crea un nuevo apartado “*void*” denominado “*void leer\_mensaje*” donde se definirán las órdenes que debe llevar a cabo el motor una vez la APP envíe determinados parámetros. En la siguiente ilustración se observa por el código que cuando la APP envía a través del módulo bluetooth una variable, esta se almacena en la variable de Arduino “*InfoAPP*” y se evalúa en función del código sucesivo. De esta manera, y tal y como se observa en la ilustración, dependiendo de la variable almacenada se avanzará, retrocederá, girará o parará. Como ejemplo se puede apreciar en el código que si se recibe la variable “*a*” se envía una señal PWM a cada motor, permitiendo que el sistema se desplace hacia delante, dado que un motor girará en sentido horario y otro en sentido antihorario. Para el giro, el funcionamiento es similar a avanzar pero, los motores de un lado girarán más rápido que el otro, esto provocará que la maquina se desplace más hacia el lado donde los motores giran más lentamente.

```
/*PROGRAMA GRUPO MOTOR*/

void leer_mensaje()
{
    InfoAPP = bluetooth.read(); // Se leen los datos que esta enviado el modulo bluetooth y se almacena en la variable InfoAPP

    if ( InfoAPP == 'a' ) // Cuando se recibe la variable "a" desde la APP se avanza
    {
        analogWrite(L_FWM_1, 150); // Se envia una señal PWM de 100 al GRUPO MOTOR 1 en sentido antihorario
        analogWrite(R_FWM_2, 150); // Se envia una señal PWM de 100 al GRUPO MOTOR 2 en sentido horario
    }

    if ( InfoAPP == 'v' ) // Cuando se recibe la variable "v" desde la APP se avanza a más velocidad
    {
        analogWrite(L_FWM_1, 250); // Se envia una señal PWM de 200 al GRUPO MOTOR 1 en sentido antihorario
        analogWrite(R_FWM_2, 250); // Se envia una señal PWM de 200 al GRUPO MOTOR 2 en sentido horario
    }

    if ( InfoAPP == 'r' ) // Cuando se recibe la variable "r" desde la APP se retrocede
    {
        analogWrite(R_FWM_1, 150); // Se envia una señal PWM de 100 al GRUPO MOTOR 1 en sentido horario
        analogWrite(L_FWM_2, 150); // Se envia una señal PWM de 100 al GRUPO MOTOR 2 en sentido antihorario
    }

    if ( InfoAPP == 't' ) // Cuando se recibe la variable "t" desde la APP se retrocede a más velocidad
    {
        analogWrite(R_FWM_1, 200); // Se envia una señal PWM de 100 al GRUPO MOTOR 1 en sentido horario
        analogWrite(L_FWM_2, 200); // Se envia una señal PWM de 100 al GRUPO MOTOR 2 en sentido antihorario
    }

    if ( InfoAPP == 'd' ) // Cuando se recibe la variable "d" desde la APP se gira a derecha
    {
        analogWrite(L_FWM_1, 150); // Se envia una señal PWM de 100 al GRUPO MOTOR 1 en sentido antihorario
        analogWrite(R_FWM_2, 50); // Se envia una señal PWM de 100 al GRUPO MOTOR 2 en sentido antihorario
    }
}
```

*Ilustración 14.- Apartado “void leer\_mensaje” en Arduino. [Fuente Propia]*

```

if ( InfoAPP == 'e' )           // Cuando se recibe la variable "e" desde la APP se gira a derecha con más velocidad
{
    analogWrite(L_PWM_1, 200); // Se envia una señal PWM de 100 al GRUPO MOTOR 1 en sentido antihorario
    analogWrite(R_PWM_2, 100); // Se envia una señal PWM de 100 al GRUPO MOTOR 2 en sentido antihorario
}

if ( InfoAPP == 'i' )           // Cuando se recibe la variable "i" desde la APP se gira a izquierda
{
    analogWrite(L_PWM_1, 50);  // Se envia una señal PWM de 100 al GRUPO MOTOR 1 en sentido horario
    analogWrite(R_PWM_2, 150); // Se envia una señal PWM de 100 al GRUPO MOTOR 2 en sentido horario
}

if ( InfoAPP == 'z' )           // Cuando se recibe la variable "z" desde la APP se gira a izquierda con más velocidad
{
    analogWrite(L_PWM_1, 100); // Se envia una señal PWM de 100 al GRUPO MOTOR 1 en sentido horario
    analogWrite(R_PWM_2, 200); // Se envia una señal PWM de 100 al GRUPO MOTOR 2 en sentido horario
}

if ( InfoAPP == 's' )           // Cuando se recibe la variable "s" desde la APP se para
{
    analogWrite(R_PWM_1, 0);    // Se envia una señal PWM de 0 al GRUPO MOTOR 1 en horario
    analogWrite(L_PWM_1, 0);    // Se envia una señal PWM de 0 al GRUPO MOTOR 1 en sentido horario

    analogWrite(R_PWM_2, 0);    // Se envia una señal PWM de 0 al GRUPO MOTOR 2 en horario
    analogWrite(L_PWM_2, 0);    // Se envia una señal PWM de 0 al GRUPO MOTOR 2 en sentido horario
}
}

```

**Ilustración 15.-** Apartado “void leer\_mensaje” segunda parte. [Fuente Propia]

Por último, para permitir estar constantemente actualizando el estado del desplazamiento se debe indicar a Arduino que debe leer el apartado “void leer\_mensaje” de forma continuada, es por ello que se incluye una sentencia “leer\_mensaje()” en el “void loop”.

```

leer_mensaje();                // para que se puedan ir leyendo todas las variables que se envien desde la APP se incluye
                                //esta sentencia para que cada vez que se ejecute el void loop, tambien se ejecute el void leer_mensaje.

Serial.println(InfoAPP);       // por ultimo en caso de que se este utilizando el puerto serial se imprime
                                //la variable recibida por bluetooth para comprobarla manualmente.
}

```

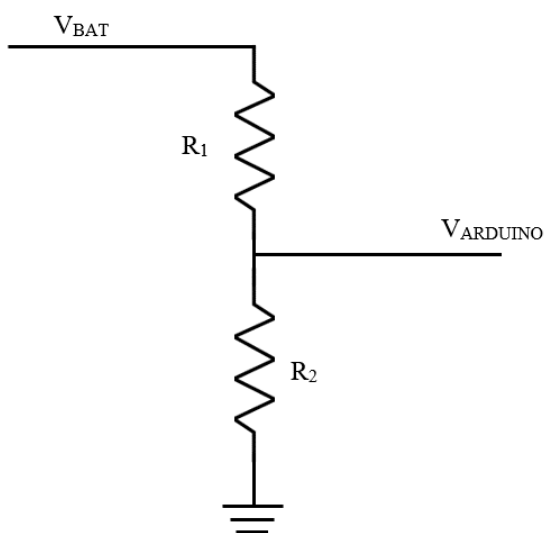
**Ilustración 16.-** Sentencia para leer repetitivamente los datos recibidos por bluetooth. [Fuente Propia]

## 5.2.- Sensores utilizados

Para el proyecto se utilizará un sensor para recibir datos sobre la carga de la batería. Para esto se utilizará un divisor de tensión que se desarrollarán a continuación.

### 5.2.1.- Voltaje de batería.

Un punto fundamental para la operación del aparato es conocer cómo va evolucionando la batería y poder decidir cuando hay que cargarla. Para esta aplicación de medida de voltaje no se ha utilizado un sensor como tal, sino que se ha aplicado un divisor de tensión para ajustar la entrada a Arduino. El divisor de tensión presenta el esquema que se ve a continuación:



*Ilustración 17.- Esquema eléctrico del divisor de tensión. [Fuente Propia]*

En el esquema,  $V_{bat}$  corresponde al borne positivo de la batería y  $V_{arduino}$  al punto de conexión con Arduino.

Dado que la batería tiene un voltaje de flotación, se considerará que esta está cargada cuando se registre un voltaje de 13,5 V, y estará descargada con un voltaje de 11,9 V. Para ajustar la entrada a Arduino se debe conseguir pasar 13,5 V de la batería a 5 V admisibles por la placa electrónica. Se utiliza la siguiente ecuación para el cálculo de  $R_1$ , suponiendo una  $R_2$  de 10 k $\Omega$ .

$$R_1 = \frac{R_2 \cdot (V_{bat} - V_{arduino})}{V_{arduino}} = \frac{10.000 \cdot (13,5 - 5)}{5} = 17 \text{ k}\Omega \quad \text{Ecuación 31}$$

Gracias a este divisor de tensión el voltaje máximo que llegará a la palca Arduino será de 5 V, de esta manera se pueden tratar los datos en formato analógico sin dañar la placa. Una vez hecha

la conexión llega a la entrada analógica de Arduino un valor de voltaje que el propio Arduino ajustará entre 0 y 1023 niveles, con una precisión de 2,44 mV. [11]

El código que se utilizará para esta aplicación se expone seguidamente. El primer paso es definir las variables para el almacenamiento y tratamiento de los datos.

```
/*Definición de variables para medir el voltaje*/  
  
float valorBAT; // Variable analógica que almacenará el valor del voltaje registrado por el divisor de tensión  
float VBAT; // Variable decimal que almacenará el valor del voltaje calculado  
float PBAT; // Porcentaje de batería calculado a partir del voltaje
```

**Ilustración 18.- Definición de variables para medir voltaje. [Fuente Propia]**

Básicamente se observan 3 variables: “*valorBAT*” encargada de almacenar el valor que se recibe por la entrada analógica de entre 0 y 1023 niveles. “*VBAT*” encargada de almacenar el voltaje y “*PBAT*” encargada de almacenar el porcentaje de batería restante.

El siguiente paso es definir el pin de Arduino al cual se conecta la salida *Vout* del divisor de tensión, se trata de una entrada analógica.

```
/*Definición del pin analógico para el registro de voltaje*/  
  
pinMode(A0, INPUT); // Pin analógico para determinar carga de batería
```

**Ilustración 19.- Definición del pin analógico. [Fuente Propia]**

Seguidamente se muestra el proceso para determinar el porcentaje de batería. En primer lugar se registra un valor analógico por el pin A0 definido anteriormente, y a continuación se calcula el voltaje asociando el valor de 13.5 V a 1023.

```
// Apartado para el calculo de porcentaje de batería restante  
  
valorBAT = analogRead(A0); // Variable que registra el voltaje recibido por la entrada analógica  
  
VBAT = valorBAT * 13.5 / 1023; // Calculo del voltaje de la batería, transformación de número analógico a voltaje
```

**Ilustración 20.- Calculo de voltaje con Arduino. [Fuente Propia]**

Finalmente, conocido el voltaje se asocia un porcentaje en función del rango en el que se encuentre *VBAT*.

```

//En función del voltaje calculado se asociará un porcentaje de batería

if (13.5 <= VBAT < 13.3){
    PBAT = 100;
}

if (13.3 <= VBAT < 13.1){
    PBAT = 90;
}

if (13.1 <= VBAT < 12.9){
    PBAT = 75;
}

if (12.9 <= VBAT < 12.7){
    PBAT = 60;
}

if (12.7 <= VBAT < 12.5){
    PBAT = 45;
}

if (12.5 <= VBAT < 12.3){
    PBAT = 30;
}

if (12.3 <= VBAT < 12.1){
    PBAT = 15;
}

if (12.1 <= VBAT < 11.9){
    PBAT = 5;
}

if (11.9 <= VBAT{
    PBAT = 0;
}

```

*Ilustración 21.- Porcentaje asociado al voltaje. [Fuente Propia]*

### 5.3.- Comunicación bluetooth.

Todos los datos registrados por los sensores y las acciones que se deben llevar a cabo para controlar los motores deben enviarse y recibirse a través de bluetooth. El primer paso es cargar la librería correspondiente en Arduino para utilizar el módulo y también asignar los pines.



```
/*Librería y asignación de pines para utilizar el modulo bluetooth*/  
  
#include <SoftwareSerial.h> // Librería que permite el uso del módulo bluetooth.  
  
SoftwareSerial bluetooth(10, 11); // Los pines de recepción y envío de datos del módulo se conectan a la placa Mega  
// El RX del módulo debe ir al pin 11 configurado como TX en la placa, se debe alternar
```

**Ilustración 22.-** Carga de librería y asignación de pines en IDE Arduino. [Fuente Propia]

Seguidamente se crean las variables implicadas en el envío y recepción. La siguiente ilustración muestra como la variable “InfoAPP” guardará los datos recibidos para el control motor.

```
/*Definición de caracteres para la lectura y envío de los datos a la APP*/  
  
char InfoAPP; // variable que recibirá los datos de la APP para el control motor
```

**Ilustración 23.-** Variables implicadas en el tratamiento de datos mediante bluetooth. [Fuente Propia]

Esta variable se define como carácter puesto que debe permitir almacenar letras, tal y como se ha explicado en el ámbito de control motor.

A continuación, se definen los pines y se inicializa el módulo bluetooth. Por la siguiente ilustración se observa cómo se resetea la información que pueda contener el bluetooth, gracias al comando “flush”, para evitar errores de comunicación.

```
/*Inicialización del módulo bluetooth.*/  
  
bluetooth.flush(); // Se inicia la comunicación bluetooth  
pinMode(10, INPUT); // El pin 10 se encargará de recibir datos por parte del módulo  
pinMode(11, OUTPUT); // El pin 11 se encargará de enviar datos por parte del módulo  
bluetooth.begin(38400); // Configuración del bluetooth a 38400 bits por segundo
```

**Ilustración 24.-** Inicio del sistema bluetooth en Arduino. [Fuente Propia]

Por otro lado, se definen los pines de recepción y envío de datos, así como se configura la velocidad de comunicación en 38.400 bits por segundo. La velocidad de comunicación afecta a la cantidad de datos recibidos, de manera proporcional.

En el caso de la recepción de datos, solo hay una sentencia a tener en cuenta dentro del código que ha sido comentada anteriormente. Se trata del comando “bluetooth.read” que permite saber que acción debe llevar a cabo el motor.

```
InfoAPP = bluetooth.read(); // Se leen los datos que está enviando el módulo bluetooth y se almacena en la variable InfoAPP
```

**Ilustración 25.-** Almacenamiento de datos recibidos mediante bluetooth. [Fuente Propia]

Por otro lado, para el envío de datos se ha creado una sentencia que enviará el porcentaje de batería a la APP. Para esta operación se observa el siguiente código:

```
// Tratamiento de datos mediante bluetooth  
  
bluetooth.println(PBAT);           // Se envia al bluetooth el dato de la bateria  
  
delay(1000);                       // Demora de 1 segundo
```

**Ilustración 26.-** Código para el envío de datos. [Fuente Propia]

También se crea una espera de 1 segundo, antes de repetir tanto la recepción de datos como el envío.

## 6.- Aplicación en MIT App Inventor

Como método de comunicación entre usuario y sistema se propone el uso de una aplicación móvil, ya que es un método sencillo de utilizar por cualquiera y que permite una gran variedad de funciones. Para este proyecto la APP diseñada tiene como objetivo el control de los motores mediante una sencilla pantalla de dirección, así como la monitorización de los sensores implicados.

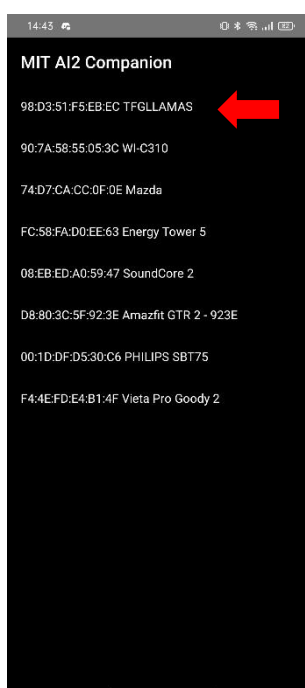
Esta APP ha sido desarrollada con MIT APP Inventor, que utiliza un lenguaje de programación por bloques para que se puedan crear comandos complejos de forma sencilla.

### 6.1.- Conexión bluetooth con la APP.

El primer paso para permitir la comunicación entre estructura y usuario es la conexión bluetooth. Como se observa en las ilustraciones, el botón conexión nos inicializa una pantalla que muestra los dispositivos bluetooth cercanos, permitiendo la conexión con Arduino.



*Ilustración 27.- Captura del apartado conexión de la APP. [Fuente Propia]*



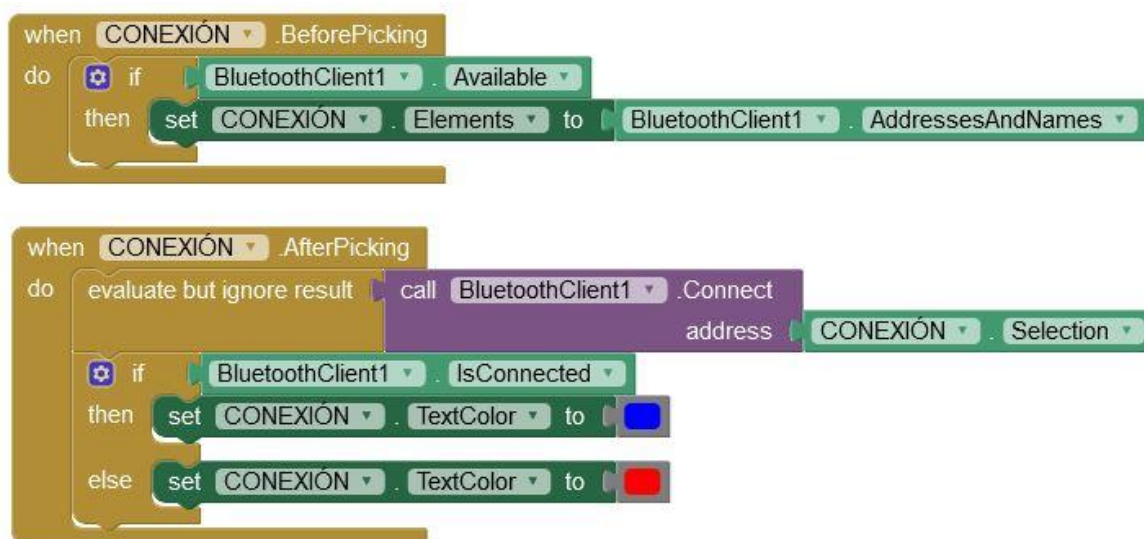
*Ilustración 28.- Lista de dispositivos bluetooth. [Fuente Propia]*



*Ilustración 29.- Captura una vez realizada la conexión. [Fuente Propia]*

Una vez se ha seleccionado el dispositivo adecuado de la lista, en este caso “TFG LLAMAS”, se retorna a la pantalla principal pero la palabra conexión a cambiado de color rojo a azul para indicar que se ha realizado correctamente la conexión.

Detrás de dicho botón existe una programación mediante bloques que configura la APP. Por la ilustración siguiente se observan dos bloques característicos. En primer lugar, se crea una sentencia para permitir mostrar los dispositivos bluetooth disponibles en la lista comentada anteriormente, tal y como ejemplifican el grupo de primeros bloques.



*Ilustración 30.- Bloques implicados en la sincronización bluetooth. [Fuente Propia]*

En segundo lugar, se crea un grupo de bloques que permita conocer si se ha realizado la conexión correctamente, en caso afirmativo se debe cambiar de color la palabra del botón a azul y en caso contrario dejarla de color rojo.

## 6.2.- Control motor con APP.

Para el control motor, la interfaz es muy sencilla. Únicamente con 5 botones se conseguiría controlar todo el movimiento del sistema. Como se observa en la siguiente ilustración cada botón esta nombrado según su funcionalidad, siendo el de “AVANZAR” para moverse hacia delante, “RETROCEDER” para ir hacia atrás, “IZQ” y “DER” controlan el giro y “STOP” detiene el movimiento. Estos botones referenciados anteriormente se pueden observar en la siguiente ilustración:

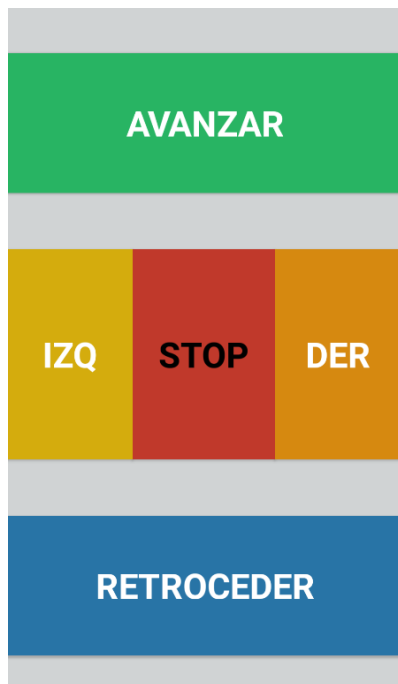


Ilustración 31.- Captura de los botones implicados en el control motor. [Fuente Propia]

Los bloques implicados en este apartado se muestran a continuación:

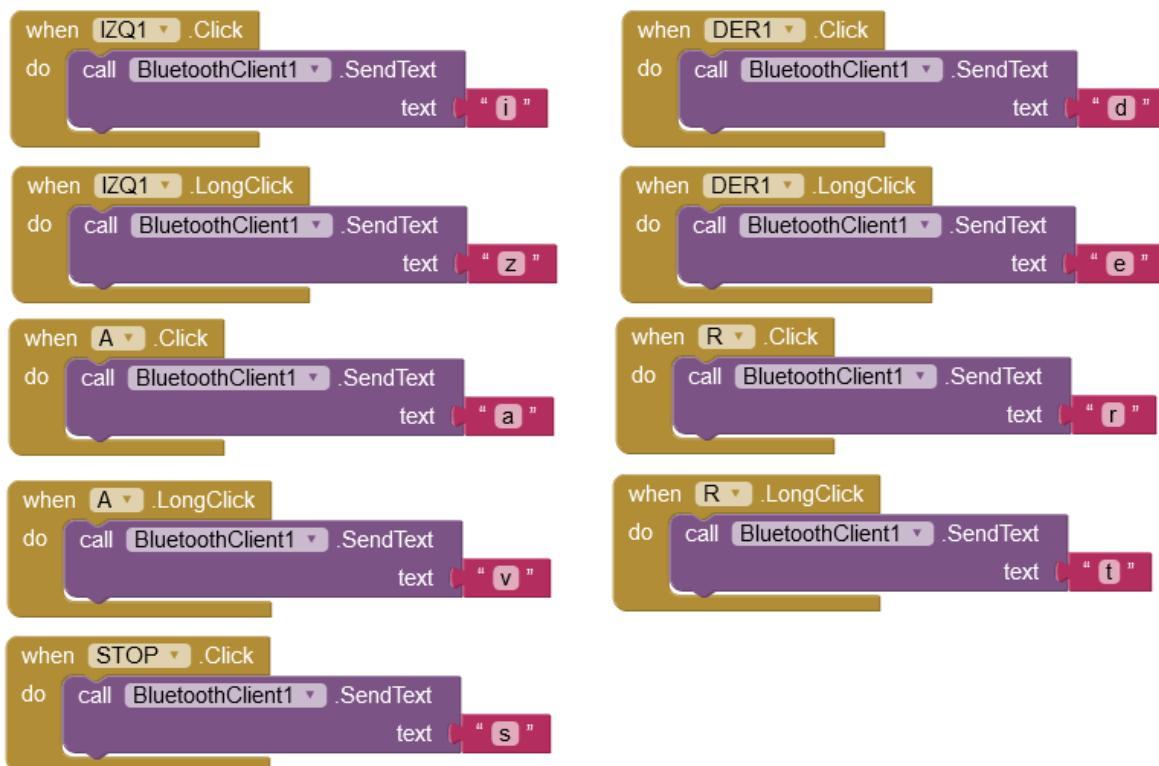


Ilustración 32.- Bloques implicados en el movimiento del motor. [Fuente Propia]

Se observa cómo hay 9 bloques. Cada uno de estos se encarga de enviar una variable a través del bluetooth hasta el Arduino para que el programa realice lo que conlleve cada variable. Como ya se ha visto en apartados anteriores, cada letra activaba un movimiento distinto es por ello que cuando se utiliza el bloque morado de “*call BluetoothClient Send Text*” seguido de un bloque purpura con una letra, por ejemplo “a”, la placa Arduino está recibiendo la variable “a” y por tanto el sistema avanzará. Solo comentar que el bloque marrón “*when xxx .Click do*”, permite separar cada botón y enviar una letra diferente, así mismo se incluye otro tipo de bloque, con la sentencia “*when xxx .LongClick do*”, que diferencia entre un clic corto y uno largo sobre el mismo botón y poder obtener diferentes velocidades de desplazamiento

### 6.3.- Monitorización de sensores.

Otra funcionalidad por destacar de la APP es su capacidad para, no solo enviar datos para el control motor, sino también recibir datos de los sensores conectados a Arduino. Concretamente, y como se ha explicado en el apartado sobre Arduino, la APP recibirá datos sobre la carga de la batería.

Tal y como se observa en la siguiente ilustración, los datos se muestran en etiquetas, justo encima del botón de conexión.



*Ilustración 33.- Captura de las etiquetas para sensores.  
[Fuente Propia]*

Seguidamente se deben tratar los datos recibidos, para ello se evalúa con la herramienta “*clock1*” si se reciben datos a través del bluetooth cuando éste esté conectado. Como se ve en

la ilustración, cuando hay datos mayores a 0 se envían a la etiqueta “Label9” para mostrarlos en pantalla.

```
when Clock1.Timer
do
  if BluetoothClient1.IsConnected
  then
    set CONEXIÓN.TextColor to blue
    if BluetoothClient1.BytesAvailableToReceive > 0
    then
      set Label9.Text to call BluetoothClient1.ReceiveText
      numberOfBytes call BluetoothClient1.BytesAvailableToReceive
    else
      set CONEXIÓN.TextColor to red
    end if
  end if
end do
```

Ilustración 34.- Tratamiento de los datos recibidos mediante bluetooth. [Fuente Propia]

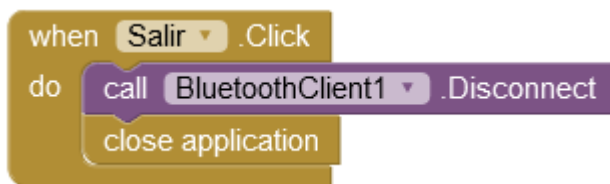
### 6.4.- Cierre de la APP.

Una vez se haya finalizado la actividad se debe cerrar la APP. Con tal de facilitar al usuario esta acción se ha creado un último botón llamado “Salir”.



Ilustración 35.- Botón salir de la APP. [Fuente Propia]

Los bloques implicados en esta operación desconectan el bluetooth y cierran la APP.



*Il·lustració 36.- Bloques para cerrar la APP y desconectar bluetooth. [Fuente Propia]*



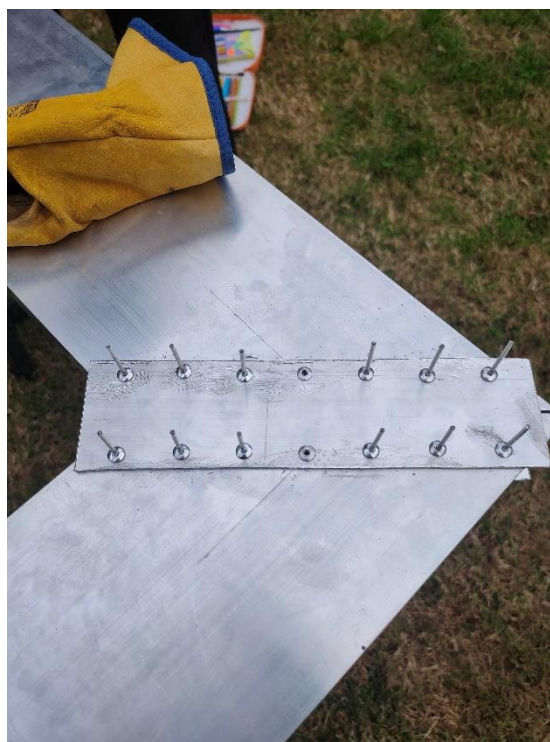
## 7.- Montaje y construcción

El objetivo de este apartado es documentar los pasos seguidos durante la construcción del aparato, así como justificar y presentar las modificaciones que difieren en el montaje respecto el diseño original.

En primer lugar se utilizaron perfiles de aluminio con un corte en diagonal para permitir crear una forma rectangular adaptada al depósito a utilizar. Como se observa en las siguientes ilustraciones, la unión entre cada perfil se hizo remachando una pequeña pletina en cada esquina.



**Ilustración 37.-** Perfiles base. [Fuente Propia]



**Ilustración 38.-** Pletina remachada. [Fuente Propia]

El siguiente paso fue agujerear la estructura principal, anteriormente comentada, para poder ubicar los pilares de la estructura. Estos pilares son de hierro, permitiendo así la soldadura de una pieza circular, que permite remachar los pilares a los perfiles de aluminio, quedado así los cuatro componentes sujetos.



*Il·lustració 39.- Pilar en estructura. [Fuente Propia]*



*Il·lustració 40.- Remachado del pilar. [Fuente Propia]*

Una vez decidida la altura final de los pilares, se cortaron y remacharon todos a la base, tal y como se muestra en la siguiente ilustración.



*Il·lustració 41.- Base con pilares. [Fuente Propia]*



Seguidamente se colocaron las barras de eje para las ruedas, soldadas a los pilares.

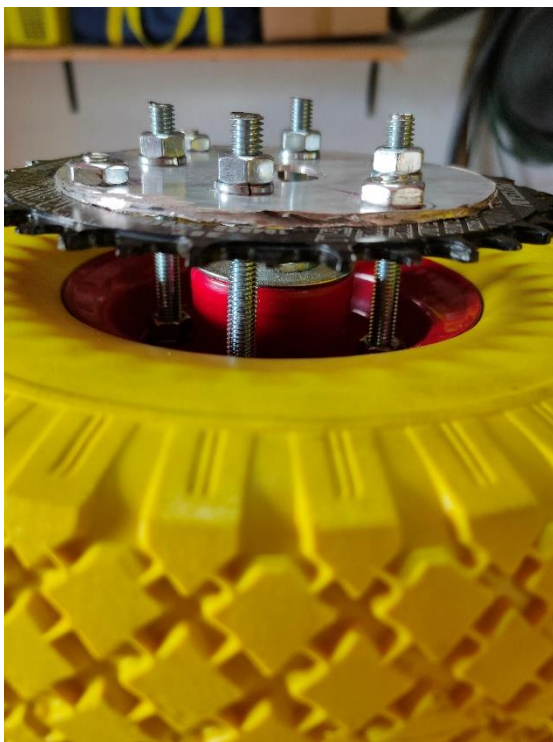


**Ilustración 42.-** Ubicación de barra para ejes.  
[Fuente Propia]

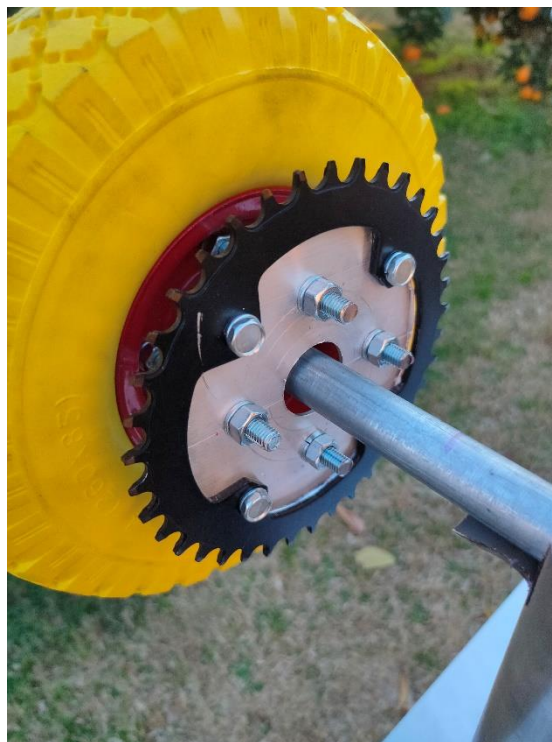


**Ilustración 43.-** Soldadura de barra para ejes.  
[Fuente Propia]

A continuación se montaron los componentes encargados de la transmisión localizados en las ruedas. Para poder colocar los platos de 38 dientes se atornilló una pletina circular de aluminio a la rueda y al propio plato, permitiendo una unión más sólida.



*Ilustración 44.- Montaje de plato en rueda.  
[Fuente Propia]*



*Ilustración 45.- Montaje de rueda en eje.  
[Fuente Propia]*

Una vez ubicadas las ruedas en la zona correspondiente del eje de la estructura, y sujetas con un pasador por ambos lados para que no se desplacen por la barra libremente, se colocaron los motores eléctricos en la estructura, siguiendo la posición del diseño. En las siguientes imágenes se observa la localización de los motores lo más cercanos posibles a las ruedas, para evitar fuerzas que puedan tensionar y dañar la estructura, así como favorecer una tracción óptima.



**Ilustración 46.-** Ubicaciones de los motores. [Fuente Propia]

La batería utilizada tiene un peso aproximado de 25 kg, y se decidió colocarla lo más cerca del suelo posible a la vez que de forma simétrica se compensarán los pesos con otros componentes, como son la bomba o la caja de conexiones. Se observa seguidamente por las ilustraciones, como la batería descansa bajo la estructura en dos perfiles de hierro unidos a las barras que hacen de eje para las ruedas.





**Ilustración 47.-** Perfiles que atraviesan la estructura. [Fuente Propia]



**Ilustración 48.-** Ubicación de la batería. [Fuente Propia]

Por otro lado, para compensar el peso de la batería, la bomba se localizó bajo la estructura principal y en el lado opuesto a la batería, para mantener un punto de gravedad bajo y centrado.



**Ilustración 49.-** Bomba bajo estructura. [Fuente Propia]

Por último, se pasaron tubos corrugados por el interior de la estructura principal para que el cableado del sistema quedase cubierto y estanco ante cualquier humedad. También se ubicó la caja de conexiones sobre la bomba.

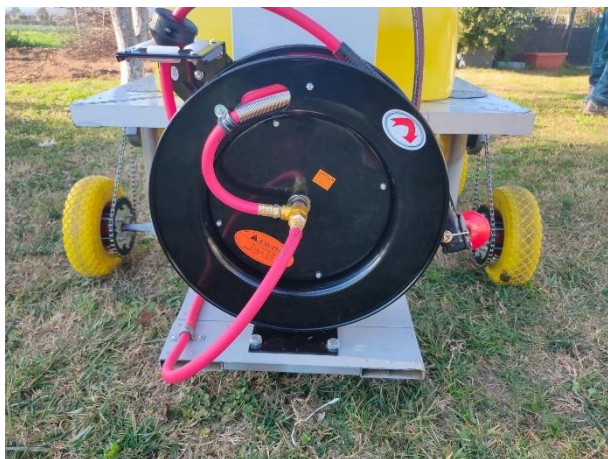


*Ilustración 50.- Caja de conexiones. [Fuente Propia]*



*Ilustración 51.- Corrugados de cableado. [Fuente Propia]*

Respecto al sistema de sulfatación, se añadió un carrete con la manguera a utilizar que reposa sobre los perfiles rectangulares bajo la estructura, y esta conectado a la bomba mediante la manguera roja.



*Ilustración 52.- Carrete para manguera. [Fuente Propia]*



*Ilustración 53.- Vista lateral del carrete. [Fuente Propia]*



Para el sistema de admisión se agujereó la parte superior del depósito y se introdujo una manguera que se une a la bomba.



*Ilustración 54.- Manguera de admisión. [Fuente Propia]*

Para tener una idea clara sobre el montaje final se presenta una imagen del sistema terminado.



*Ilustración 55.- Sistema acabado. [Fuente Propia]*



## **8.- Modo de empleo**

Para garantizar el uso correcto del dispositivo y evitar situaciones desfavorables o fallos, en este apartado se introducirán los pasos a seguir para un ejemplo de operación así como las condiciones en las que no se recomienda funcionar.

### **8.1.- Modo de operación**

Seguidamente se enumeran los pasos a seguir para la puesta en marcha, uso y parada del sistema.

Paso 1: Inicialización del sistema. El primer paso es alimentar todo el sistema mediante la batería, pulsando el botón 2 localizado en la parte superior de la caja estanca. Esto permite alimentar tanto motores como el sistema Arduino. Seguidamente se inicia la APP y se hace la conexión bluetooth con el sistema.

Paso 2: Desplazamiento. Utilizando los botones de la APP se avanza, retrocede o gira en función del lugar al que se desee llegar.

Paso 3: Sulfatación. Una vez tomada una posición correcta se inicia la sulfatación, para ello simplemente se pulsa el botón una localizado en la parte superior de la caja estanca para activar la alimentación a la bomba. Seguidamente se utiliza la pistola para pulverizar como se crea conveniente. Al acabar se debe pulsar de nuevo el botón 1 para apagar la bomba.

Paso 4: Desplazamiento. Si se quiere volver a la zona de partida para guardar la máquina, simplemente se utiliza la APP para desplazarse al lugar necesario

Paso 5: Apagado del sistema. Una vez llegado al punto de estacionamiento, se cierra la APP y se pulsa el botón 2 para cortar la alimentación del sistema.

### **8.2.- Condiciones desfavorables**

Con el fin de evitar situaciones que pueden traducirse en problemas para el uso o la integridad del aparato, se definirán seguidamente una serie de condiciones bajo las cuales no se recomienda operar.

Suelo: no se recomienda el uso del sistema en zonas embarradas o recientemente humedecidas por fuertes lluvias, ya que el propio peso del sistema podría provocar quedarse atascado en el barro. Como superficie óptima se recomienda un suelo firme, ya sea césped o tierra compacta.

Pendiente: evitar en la medida de lo posible subir pendientes mayores al 35%, dado que el depósito, estando lleno, podría volcar toda la estructura al desplazar el centro de gravedad hacia atrás.

Peso: con tal de no sobrecargar la estructura y asegurara la integridad de los perfiles empleados, no colocar pesos extra en ninguna zona del sistema y no subirse a la estructura.

## 9.- Análisis del impacto ambiental

Uno de los objetivos del proyecto estaba plantear un sistema de operación que, durante su uso, sea considerado limpio. Este hecho se consigue gracias al uso de una bomba y todos los componentes eléctricos. El sistema de sulfatación presentado en la ilustración 2, es un sistema que utiliza gasolina como combustible. En este apartado se desea presentar el ahorro en carburante y en emisiones de CO<sub>2</sub> entre el motor de combustión y el sistema eléctrico aquí presentado.

El sistema de combustión es un motor Honda modelo GX25, un motor pequeño que consume 0,54 L por hora. [12]

Para la sulfatación se considerará que, aproximadamente, se sulfata una media de 50 veces año, con una duración de 1 hora cada vez. De esta manera se habrá sulfatado anualmente 50 horas.

Considerando que un litro de gasolina emite 2,37 kg de CO<sub>2</sub> [auto10], se pueden determinar tanto las emisiones por hora del motor de combustión como las emisiones anuales.

Multiplicando el consumo por las emisiones de 1L de gasolina:

$$CO_2 \text{ por hora} = \frac{2,37 \text{ kg } CO_2}{1 \text{ L}} \cdot \frac{0,54 \text{ L}}{1 \text{ hora}} = 1,27 \text{ kg } CO_2/\text{hora} \quad \text{Ecuación 32}$$

Se obtiene que las emisiones de CO<sub>2</sub> por hora de operación son de 1,27 kg. Si se consideran todas las operaciones que se llevan a cabo durante el año:

$$CO_2 \text{ anual} = 1,27 \text{ kg } \frac{CO_2}{\text{hora}} \cdot 50 \text{ horas} = 64 \text{ kg } CO_2/\text{anual} \quad \text{Ecuación 33}$$

Se obtiene que las emisiones anuales son de 64 kg de CO<sub>2</sub>.

Por otro lado, si se considera el coste de la gasolina, aproximadamente 1,5 €/litro, el coste de operación sería de 40,5 €/anuales.

$$\text{Coste} = 50 \text{ horas} \cdot 0,54 \frac{\text{L}}{\text{hora}} \cdot 1,5 \frac{\text{€}}{\text{L}} = 40,5 \text{ €/anuales} \quad \text{Ecuación 34}$$

Si se comparan las emisiones obtenidas con las de un vehículo que consume 7l/100km, los 64 kg de CO2 se podrían emitir a los 100 km.

Se trata de un sistema pequeño y es lógico obtener bajo coste y bajas emisiones, pero este podría ser el caso de una zona agrícola pequeña, si se considerasen grandes hectáreas estos valores serían más significativos.

## 10.- Presupuesto

Seguidamente se presenta en una tabla el presupuesto del proyecto, donde se contemplan materiales y componentes utilizados, así como costes de ingeniería relacionados con el diseño y los costes de montaje.

Artículo	Precio [€]	Uds.
<b>Componentes y Materiales</b>		
Deposito 200 L	103,2	1
Rueda D25 cm	13,9	4
Plato 38 dientes	14,99	4
Motor Eléctrico 12V 250W	80,59	4
Bomba DC 12V	92	1
Placa Arduino Mega	14,99	1
Driver control motor	11,79	2
Cableado	15	1
Caja de conexiones	14,23	1
Perfil 150x40	222	1
Tubo redondo	6,652	4
Batería AGM 12V 90Ah	120	1
Conector para Arduino	0,799	1
Botones	0,879	4
Remaches 4x8mm	0,089	100
Cadena	13,75	4
Tornillería	50	1
Perfil Redondo 20 mm	6,5	2
Adaptador tornillos Arduino Mega	14,99	1
<b>Diseño</b>		
Costes de Ingeniería	30	400
<b>Montaje</b>		
Costes de Montaje	15	200

<b>BASE</b>		16.222,88 €
<b>IVA</b>	21 %	3.406,80 €
<b>TOTAL</b>	<b>19.629,68 €</b>	

Entre los materiales más caros utilizados, destacan los perfiles utilizados para la base del sistema, también el propio depósito y los motores. Por otra parte, los conceptos más caros son los costes de diseño y montaje. Para estos conceptos se han contado aproximadamente las horas dedicadas al proyecto, donde la parte de diseño tiene mayor peso. En un proceso automatizado de fabricación estos costes se podrían ajustar mucho más y el precio bajaría bastante.

El precio final del producto sería de 19.629,68 €, un precio final bastante caro, pero hay que remarcar que se trata de un prototipo. Por otra parte, aunque el precio de venta al público podría disminuir, sería un sistema orientado a gente del sector con plantaciones de dimensiones medias y no ha pequeños jardines.

## **11.- Conclusiones**

El objetivo principal de este trabajo era crear un sistema capaz de transportar una gran cantidad de agua desde un punto a otro sin necesidad de hacer fuerza física por parte del usuario. Tras realizar este trabajo se ha llegado a la conclusión de que este sistema es posible.

Iniciando el trabajo en el dimensionamiento, los componentes utilizados se ajustan bien a las condiciones del sistema una vez vista la máquina en funcionamiento. Referente al desplazamiento, el uso de 4 motores ha permitido tener un mayor control en el movimiento. La bomba de sulfatación utilizada aunque pequeña permite realizar la sulfatación cumpliendo los parámetros especificados en su cálculo. En relación con la batería se acabó escogiendo una mayor a la calculada, siempre dejando cierta tolerancia por si se decidía actualizar a componentes de mayor consumo.

Con relación al apartado de diseño, hay que destacar que se partió de un proyecto de SolidWorks rudimentario y se fue evolucionando hasta llegar al presentado en este proyecto, un sistema que utiliza materiales livianos, de fácil obtención y que permiten crear una estructura sólida y sencilla para unir todos los componentes del proyecto. Este apartado ha sido útil para valorar todo proceso creativo que hay detrás de cualquier producto, ya sea grande o pequeño, puesto que siempre pueden surgir problemas que se van solucionado durante el desarrollo del proyecto, como fue en el caso de la ubicación de la batería o el paso del cableado, entre otros.

Respecto al código creado en Arduino y la aplicación, aunque se utilizaron sistemas de programación sencillos, se consiguió crear una programación bastante compleja que permite el control de los componentes del sistema, como la batería y los motores eléctricos, de forma sencilla y accesibles a cualquier usuario gracias a la APP.

Como futuras mejoras al sistema, sería interesante plantear la localización del motor en la propia rueda para poder introducir un sistema de amortiguación. También incluir un sistema GPS y un pitorro de pulverización fijo por encima del depósito para que el usuario pudiera marcar el recorrido y se sulfatase de forma totalmente autónoma el espacio.

Por último, el montaje de la estructura ha sido una experiencia que satisface todas las expectativas que se tenían cuando todo esto era solo una idea.

## 12.- Bibliografia

- [1] G. de C. Departament d'agricultura alimentació i acció rural, Curs de nivell qualificat de manipulador i aplicador de productes fitosanitaris, Barcelona, 2009.
- [2] Descubre todo sobre el Pulverizador Agrícola y mucho más, (s. f.). <https://hablemosdeculturas.com/pulverizador-agricola/> (accedido 27 de noviembre de 2021).
- [3] F.. Estrany Coda, M. Sánchez Jiménez, Transporte y Distribución de Fluidos Energéticos, (2021).
- [4] Plásticos PE - TECAFINE PE | Ensinger, (s. f.). <https://www.ensingerplastics.com/es-es/semielaborados/plasticos-industriales/pe> (accedido 27 de noviembre de 2021).
- [5] Ejercicio: El rozar se va a acabar, (s. f.). <https://www.fiscalab.com/ejercicio/866> (accedido 26 de octubre de 2021).
- [6] El rozamiento por deslizamiento, (s. f.). <http://www.sc.ehu.es/sbweb/fisica/dinamica/rozamiento/general/rozamiento.htm> (accedido 27 de noviembre de 2021).
- [7] Calculadora del número de revoluciones PFERD, (s. f.). <https://www.pferd.com/es-es/servicios/calculadora-numero-revoluciones-pferd/#> (accedido 27 de noviembre de 2021).
- [8] Cómo calcular la relación de reducción en tus proyectos con engranajes – Blog CLR, (s. f.). <https://clr.es/blog/es/calcular-relacion-de-reduccion/> (accedido 27 de noviembre de 2021).
- [9] ¿Cuánto consume Arduino? | Tienda y Tutoriales Arduino, (s. f.). <https://www.prometec.net/consumos-arduino/> (accedido 28 de octubre de 2021).
- [10] Medir distancia con Arduino y sensor de ultrasonidos HC-SR04, (s. f.). <https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/> (accedido 20 de noviembre de 2021).
- [11] Entradas analógicas en Arduino, (s. f.). <https://www.luisllamas.es/entradas-analogicas-en-arduino/> (accedido 25 de noviembre de 2021).
- [12] Engine model details - Honda Engines, (s. f.). <https://www.honda-engines-eu.com/es/web/eec-public-site/engine-model-details?e=3> (accedido 8 de diciembre de 2021).

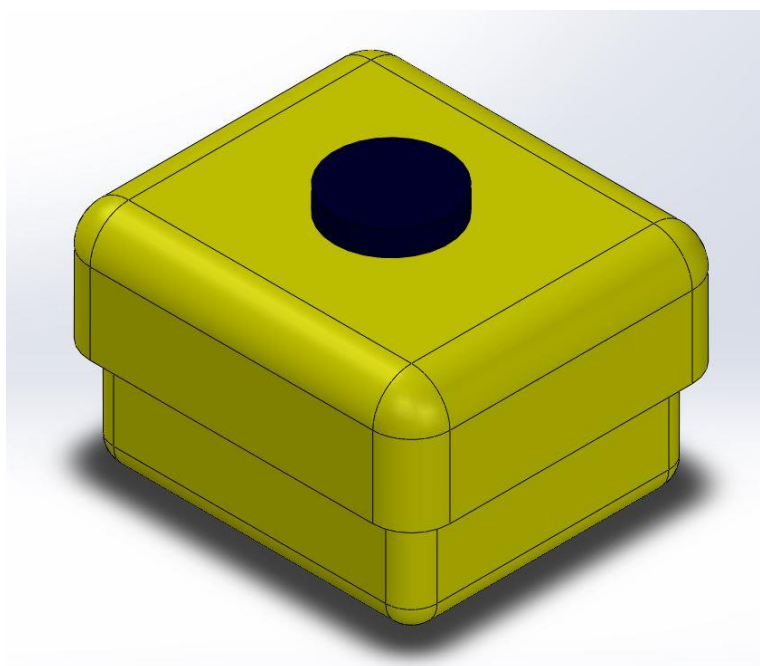


## Anexo A: Componentes

En este primer anexo se tiene como objetivo presentar todas las piezas y componentes diseñados en el proyecto, por si fuera necesario tener un concepto más claro de estos elementos.

### Depósito 200 L

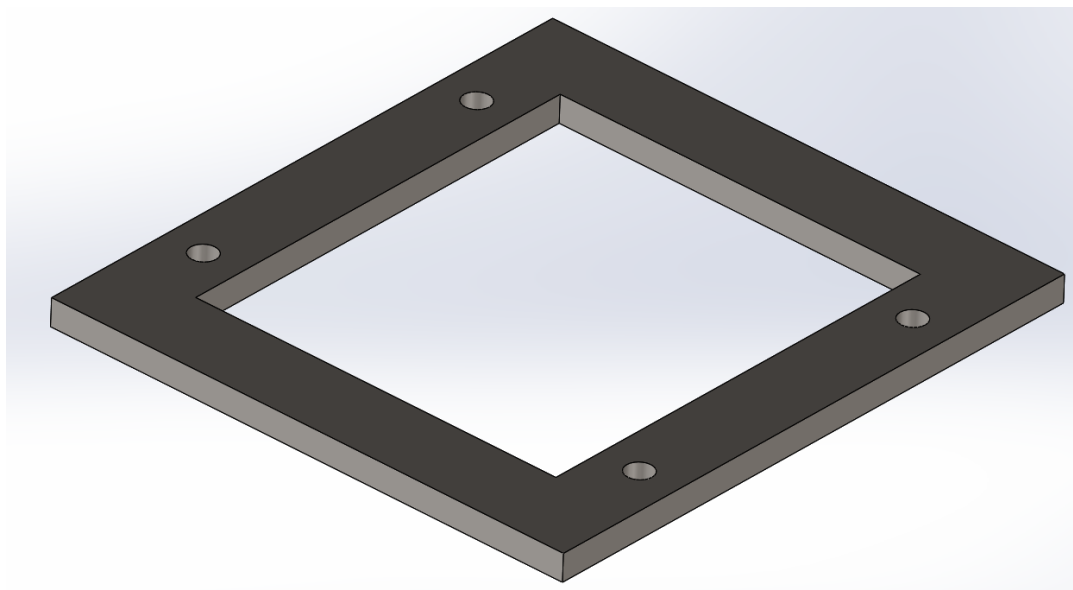
El depósito de este proyecto tiene una capacidad de 200 L, aproximadamente dimensiones de 890x790x560 mm y un peso de 200 kg. Su función consiste en contener el líquido de sulfatación.



*Ilustración 56.- Plano 3D del depósito de 200 L. [Fuente Propia]*

### Base de apoyo

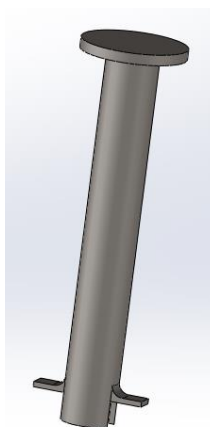
Esta estructura hace de punto de unión para la mayoría de los componentes del sistema. El depósito reposa sobre ella, la bomba de sulfatación y los motores están acollados a sus laterales y los pilares están remachados por la cara superior.



*Ilustración 57.- Base del sistema. [Fuente Propia]*

## Ejes verticales

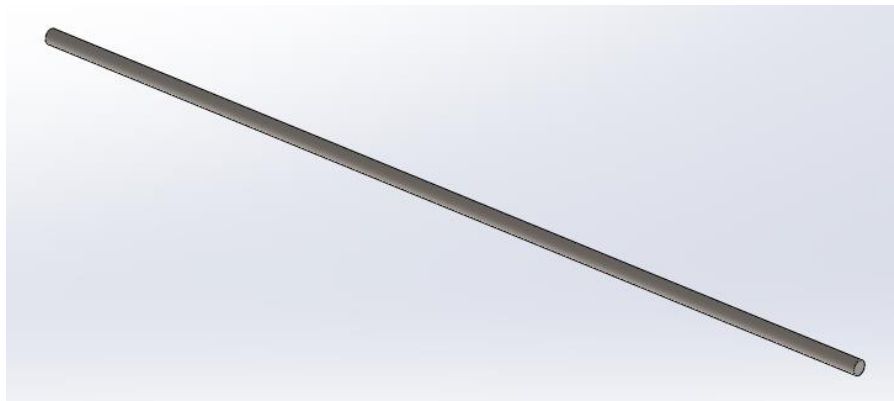
Este componente se encuentra duplicado 4 veces en el sistema y se encarga de hacer la función similar a un pilar, ya que permite unir los ejes de las ruedas con la base de apoyo de forma vertical.



*Ilustración 58.- Pilar. [Fuente Propia]*

## Ejes horizontales

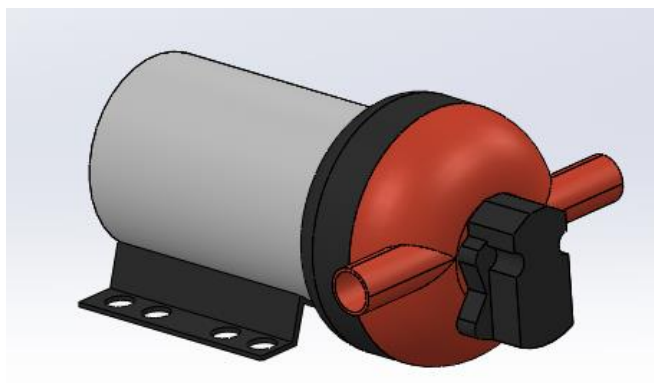
Los ejes horizontales se encargan de unir las ruedas dos a dos con los correspondientes ejes verticales, permitiendo así que el sistema de transmisión este sujeto a la estructura.



*Ilustración 59.- Eje de ruedas. [Fuente Propia]*

## Bomba de sulfatación

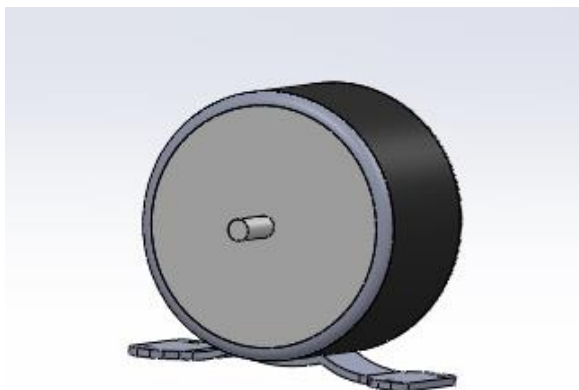
Como componente destacado del proyecto encontramos la bomba de sulfatación. Este aparato se utiliza para propulsar el fluido de sulfatación desde el depósito hasta la aplicación concreta, a través de la manguera.



*Ilustración 60.- Bomba. [Fuente Propia]*

## Motor eléctrico

Cuatro elementos principales del sistema son los motores eléctricos, estos se encargan de transformar la energía eléctrica en movimiento y transmitirlo a las ruedas gracias a los platos y cadenas.



*Il·lustració 61.- Motor. [Fuente Propia]*

### **Plato 38 dientes**

El plato sólido a las ruedas es el más grande del proyecto y tiene 38 dientes, permite comunicar el movimiento a las ruedas gracias a la cadena que le une al motor eléctrico.



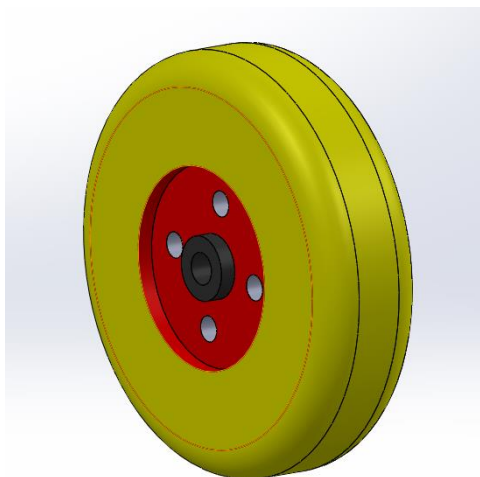
*Il·lustració 62.- Plato. [Fuente Propia]*

### **Plato 11 dientes**

El plato pequeño del sistema se encuentra sólido a los motores eléctricos y permiten la comunicación entre este y la cadena.

## Rueda

Un componente esencial del sistema es la rueda, 4 unidades de este elemento permiten a la estructura estar en contacto con el suelo y desplazarse.



*Ilustración 63.- Rueda. [Fuente Propia]*

## Batería

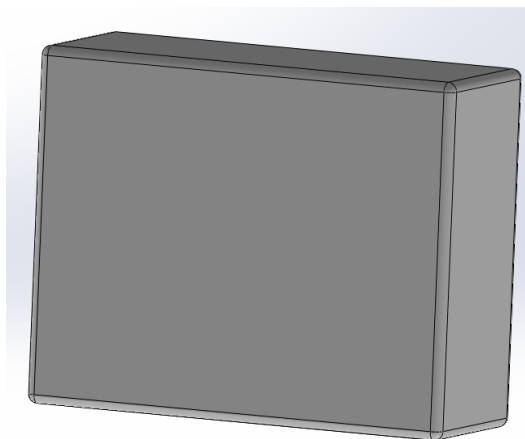
Este elemento permite alimentar todo el sistema, se localiza bajo la estructura para guardar un punto de gravedad bajo y compensar los pesos del sistema.



*Ilustración 64.- Batería. [Fuente Propia]*

## Caja de conexiones

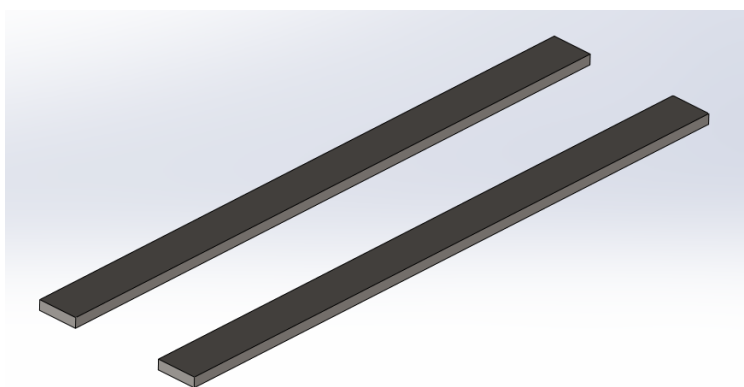
El punto donde se une todo el cableado, se colocan drivers y placa Arduino es la caja de conexiones de plástico que además es estanca.



*Ilustración 65.- Caja estanca. [Fuente Propia]*

## Perfiles de apoyo

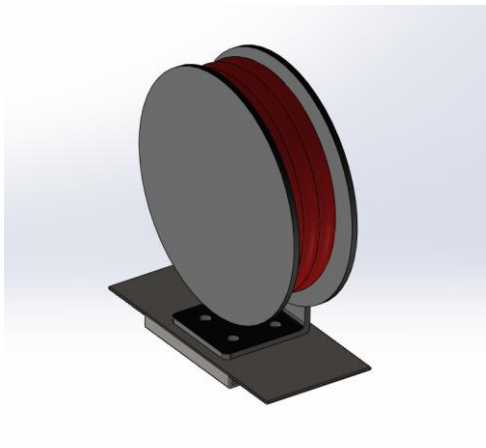
El lugar donde la batería quedará sujeta es en dos perfiles que están sujetos a los ejes de las ruedas.



*Ilustración 66.- Perfiles de apoyo. [Fuente Propia]*

## Carrete para manguera

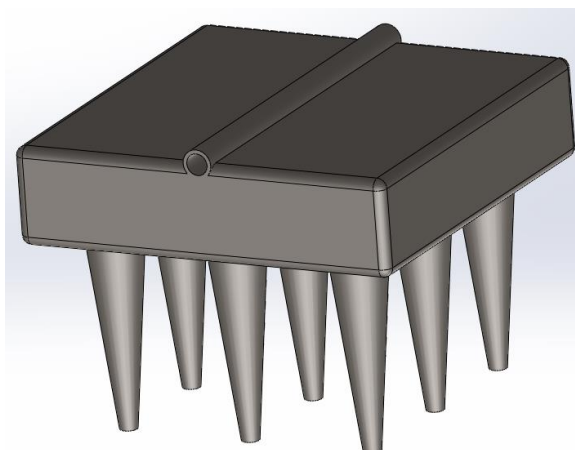
Componente encargado de recoger la manguera de sulfatación.



*Ilustración 67.- Carrete de manguera. [Fuente Propia]*

## Base de anclaje

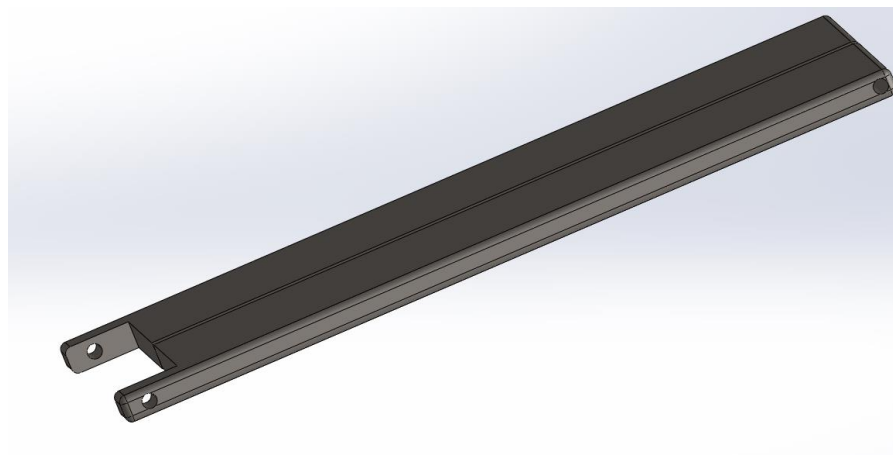
Este componente permite unir los perfiles verticales de la base de carga con el suelo, para tener toda la estructura bien anclada.



*Ilustración 68.- Base de anclaje. [Fuente Propia]*

## Perfiles verticales

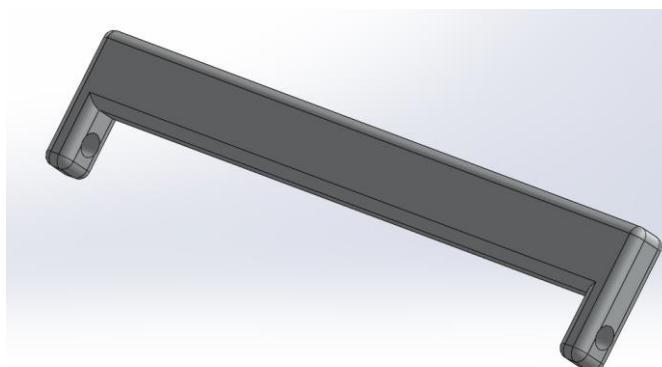
Estos perfiles son la estructura principal de la base de carga, donde se acopla el cargador y la placa solar. Unidos a la base de anclaje por una varilla que permite el movimiento.



*Ilustración 69.- Perfiles verticales. [Fuente Propia]*

## Sistema de bisagra

Componente encargado de unir la placa solar con los perfiles verticales y que, gracias a una varilla interior, permite el movimiento.

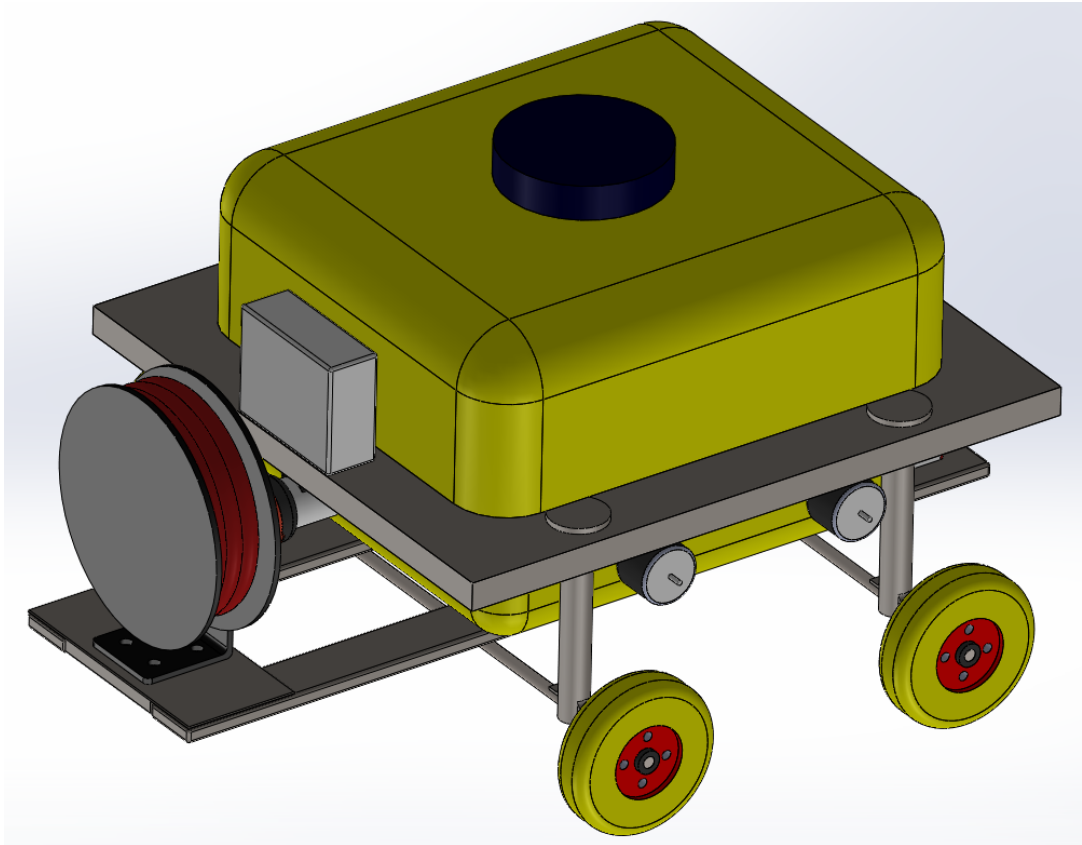


*Ilustración 70.- Bisagra. [Fuente Propia]*

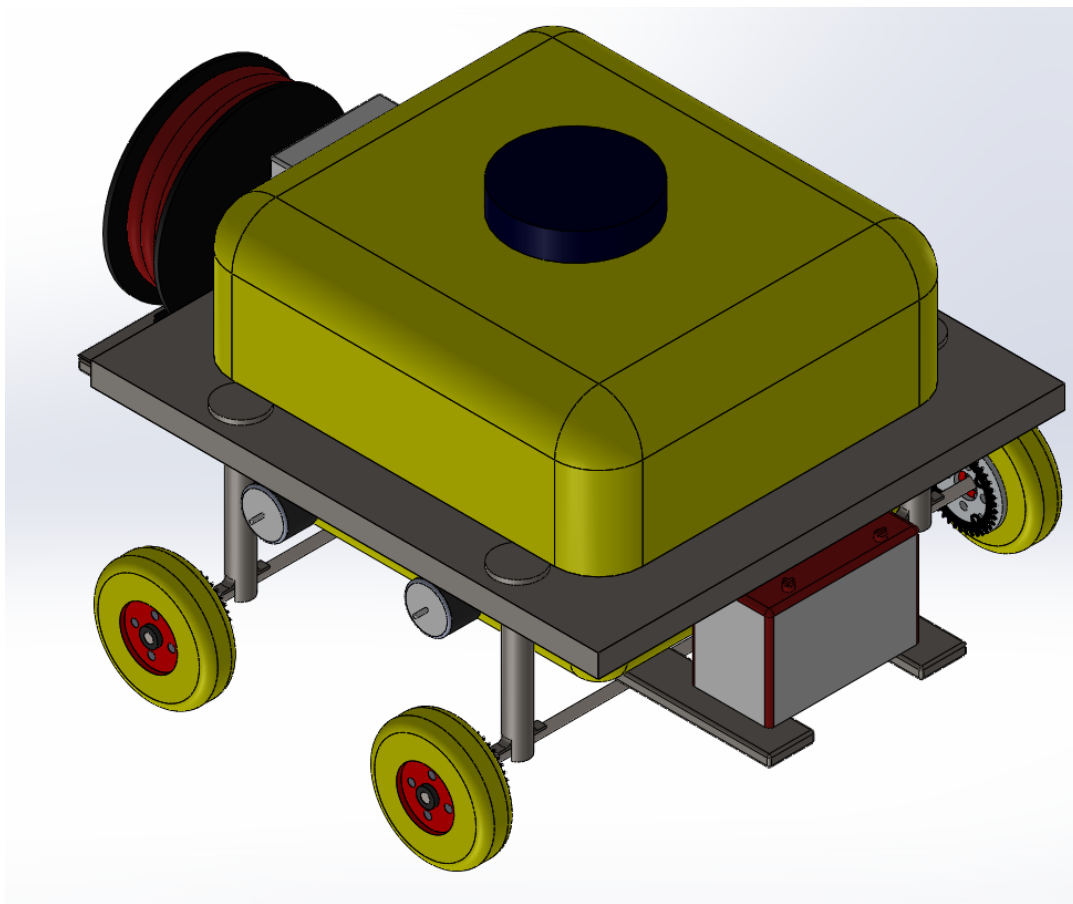
## Ensamblaje de la máquina de sulfatación

Por último se incluyen diferentes vistas con todo el sistema montado diseñado en SolidWorks.





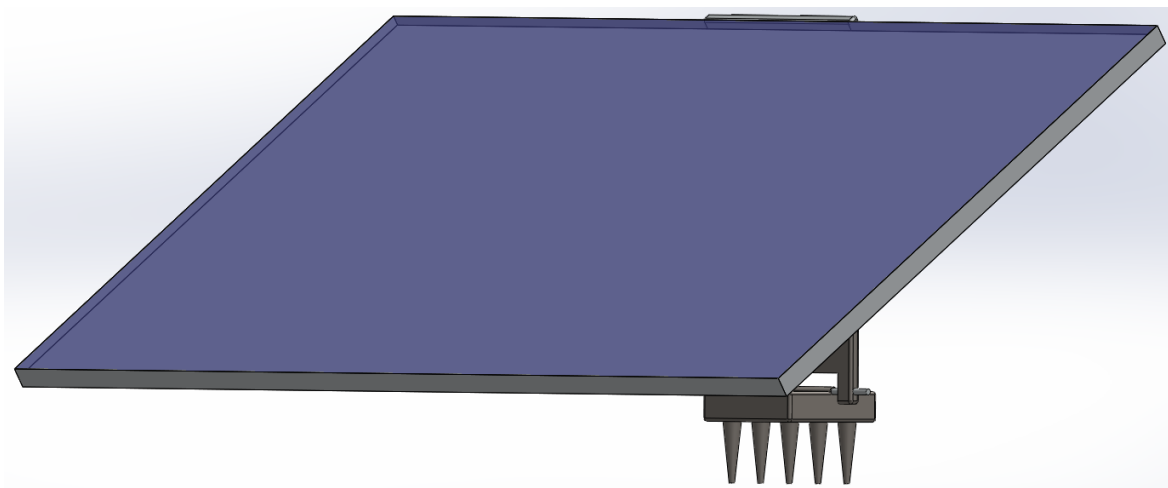
*Ilustración 71.- Maquina de sulfatación vista 1. [Fuente Propia]*



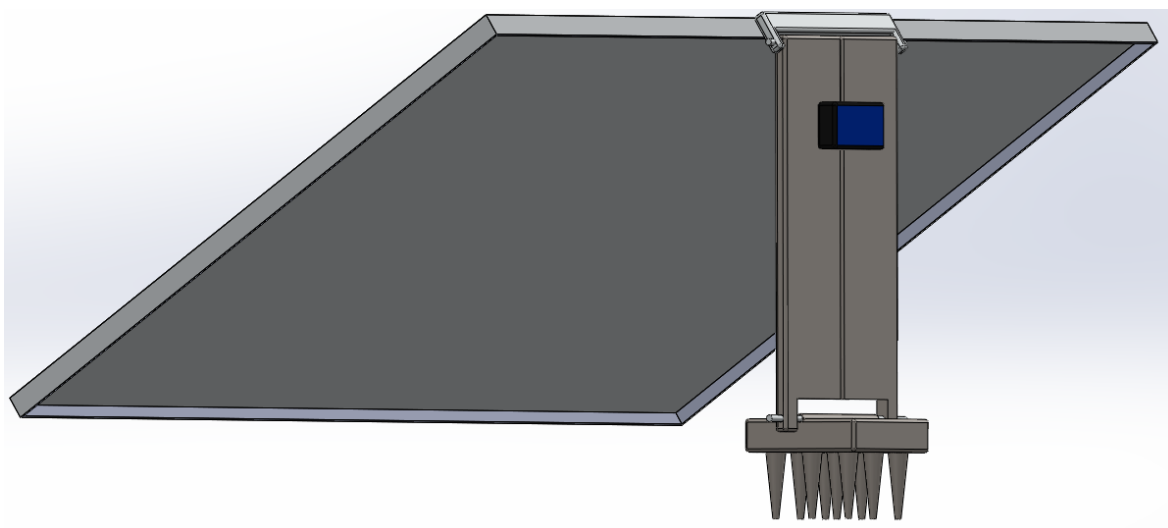
*Ilustración 72.- Maquina de sulfatación vista 2. [Fuente Propia]*

## **Ensamblaje de la estación de carga**

También se incluyen diferentes vistas del ensamblaje propuesto para la estación de carga.



*Ilustración 73.- Estación de carga vista frontal. [Fuente Propia]*



*Ilustración 74.- Estación de carga vista trasera. [Fuente Propia]*

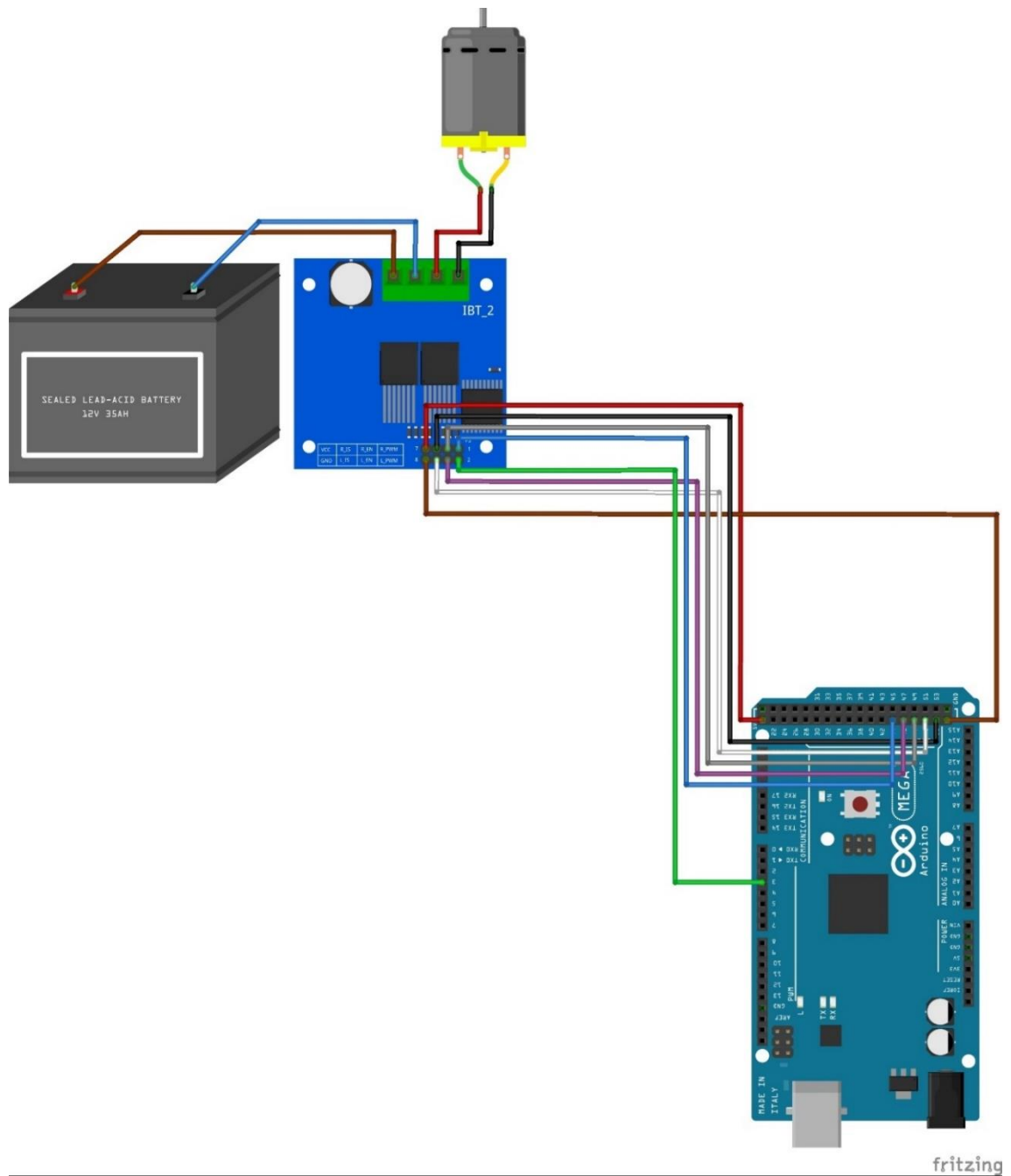
## **Anexo B: Esquemas de conexión**

Con tal de no entorpecer la lectura del documento se incluyen en el siguiente anexo los esquemas eléctricos hechos con el programa Fritz, para complementar la explicación del apartado sobre Arduino y la asignación de pines.

### **Control Motor**

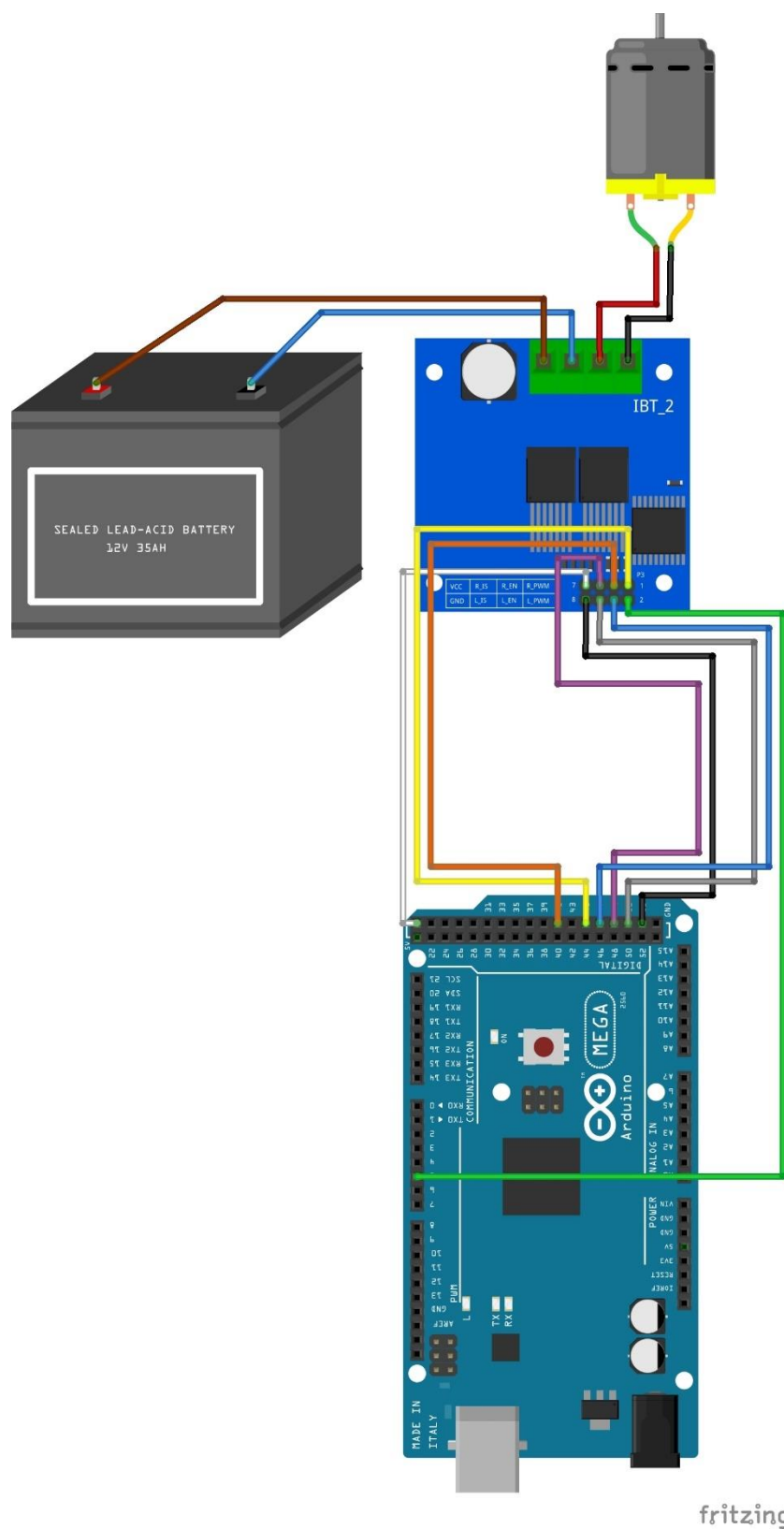
El esquema de conexión en la palca Arduino Mega para ambos motores se presenta a continuación, siguiendo las indicaciones del código.

Motor 1:



*Ilustración 75.- Conexión del motor 1 con la placa Arduino. [Fuente Propia]*

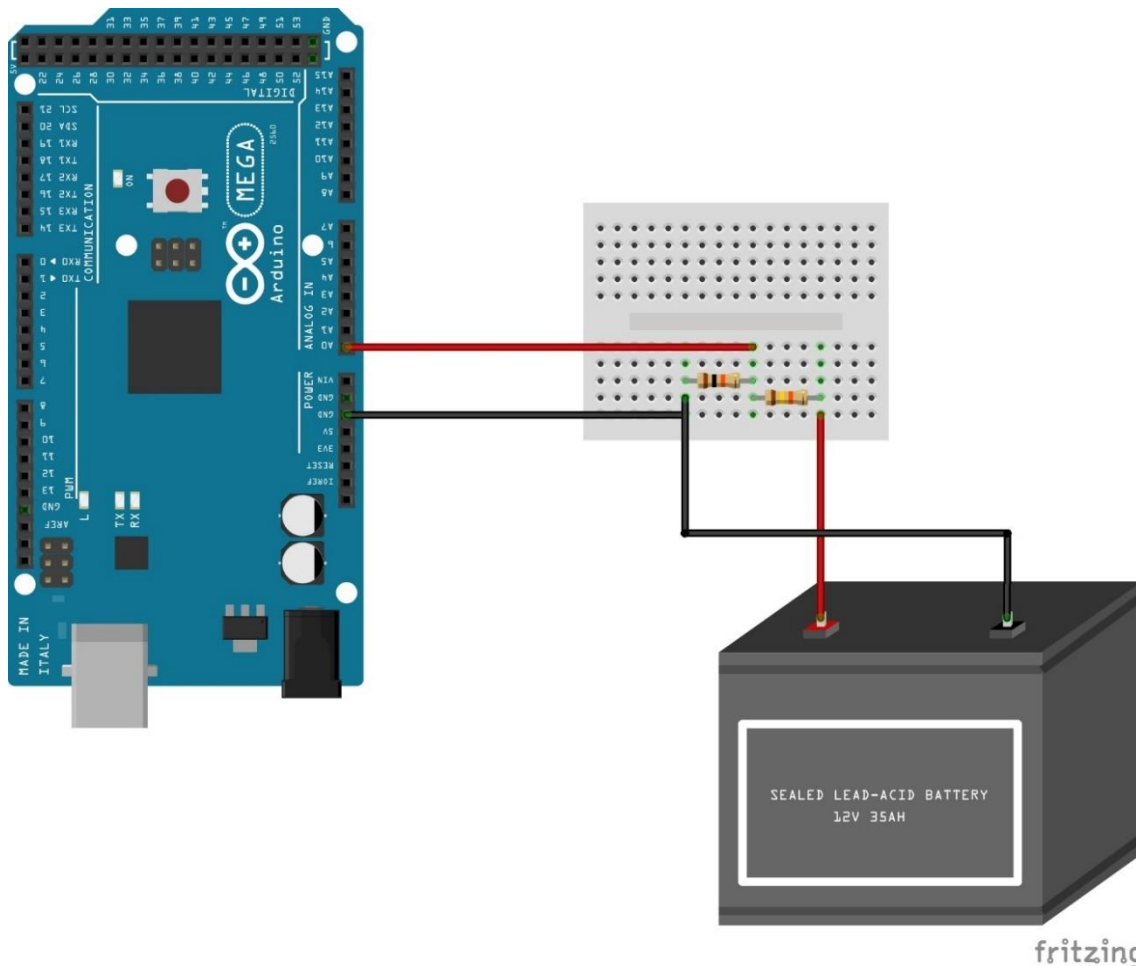
Motor 2:



*Il·lustración 76.- Conexión del motor 2 con la placa Arduino. [Fuente Propia]*

## Voltaje de batería

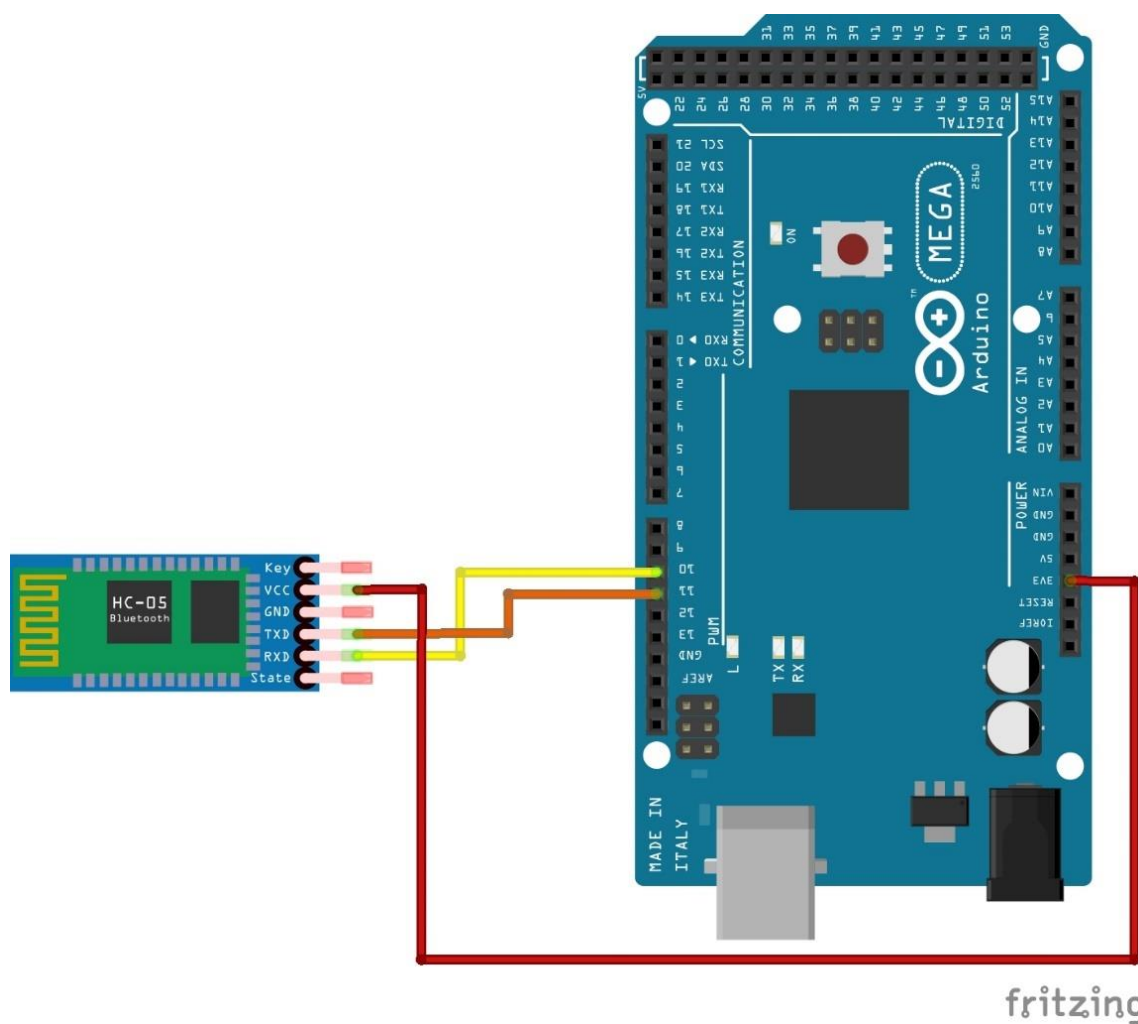
El divisor de tensión y la conexión entre Arduino y batería se muestra en la siguiente ilustración.



*Ilustración 77.- Divisor de tensión y conexión en Arduino. [Fuente Propia]*

## Comunicación bluetooth

La conexión del módulo bluetooth y la placa Mega se muestra a continuación.



*Il·lustración 78.- Conexión del módulo bluetooth. [Fuente Propia]*



## Anexo C: Código Arduino

Aquí se recoge todo el código creado en IDE Arduino para el control de sistema.

```

/* TFG Diseño Sulfatadora Móvil
 * Alvaro Llamas
 *
 * 15/01/2022
 *
 * El objetivo de este programa es controlar el MOTOR ELECTRICO gracias a los datos que se reciben de la APP en MIT APP INVENTOR,
 * y, a su vez, enviar los datos de CARGA DE BATERIA, gracias al divisor de tensión.
 *
 */

/*Definición de los pines implicados en el GRUPO MOTOR 1*/

int R_IS_1 = 52; // El pin R_IS se encarga de asegurar que la corriente que pase en el sentido horario no supere el maximo, funciona a modo de alarma.
int R_EN_1 = 48; // El pin R_EN permite habilitar al driver el paso de corriente y, por tanto, el movimiento horario.
int R_PWM_1 = 44; // El pin R_PWM permite ajustar el valor de velocidad al que gira el motor en sentido horario.
int L_IS_1 = 50; // El pin L_IS se encarga de asegurar que la corriente que pase en el sentido horario no supere el maximo, funciona a modo de alarma.
int L_EN_1 = 46; // El pin L_EN permite habilitar al driver el paso de corriente y, por tanto, el movimiento antihorario.
int L_PWM_1 = 3; // El pin L_PWM permite ajustar el valor de velocidad al que gira el motor en sentido antihorario.

/*Definición de los pines implicados en el GRUPO MOTOR 2*/

int R_IS_2 = 49; // El pin R_IS se encarga de asegurar que la corriente que pase en el sentido horario no supere el maximo, funciona a modo de alarma.
int R_EN_2 = 41; // El pin R_EN permite habilitar al driver el paso de corriente y, por tanto, el movimiento horario.
int R_PWM_2 = 45; // El pin R_PWM permite ajustar el valor de velocidad al que gira el motor en sentido horario. // giro hacia derecha
int L_IS_2 = 51; // El pin L_IS se encarga de asegurar que la corriente que pase en el sentido horario no supere el maximo, funciona a modo de alarma.
int L_EN_2 = 47; // El pin L_EN permite habilitar al driver el paso de corriente y, por tanto, el movimiento antihorario.
int L_PWM_2 = 5; // El pin L_PWM permite ajustar el valor de velocidad al que gira el motor en sentido antihorario.

/*Definición de variables para medir el voltaje*/

float valorBAT; // Variable analógica que almacenará el valor del voltaje registrado por el divisor de tensión
float VBAT; // Variable decimal que almacenará el valor del voltaje calculado
float PBAT; // Porcentaje de batería calculado a partir del voltaje

/*Librería y asignación de pines para utilizar el modulo bluetooth*/

#include <SoftwareSerial.h> // Librería que permite el uso del módulo bluetooth.

SoftwareSerial bluetooth(10, 11); // Los pines de recepción y envío de datos del módulo se conectan a la placa Mega
// El RX del módulo debe ir al pin 11 configurado como TX en la placa, se debe alternar

/*Definición de caracteres para la lectura y envío de los datos a la APP*/

char InfoAPP; // variable que recibirá los datos de la APP para el control motor

/*APARTADO DE DEFINICIONES Y ASIGNACIÓN*/

void setup() {

/*Inicialización del puerto serie*/

Serial.begin(9600); // Se inicializa el puerto serie para poder mostrar los datos registrados y recibidos

/*Inicialización del módulo bluetooth.*/

bluetooth.flush(); // Se inicia la comunicación bluetooth
pinMode(10, INPUT); // El pin 10 se encargará de recibir datos por parte del módulo
pinMode(11, OUTPUT); // El pin 11 se encargará de enviar datos por parte del módulo
bluetooth.begin(38400); // Configuración del bluetooth a 38400 bits por segundo

```

```

/*Definición de estado de los pines implicados en el GRUPO MOTOR 1*/
pinMode(R_IS_1, OUTPUT); // Se define el pin R_IS como salida, aunque no se va a utilizar en esta aplicación pero se conecta por seguridad.
pinMode(R_EN_1, OUTPUT); // Se define el pin R_EN como salida, dependiendo del estado que se le aplique estará activo o no.
pinMode(R_PWM_1, OUTPUT); // Se define el pin R_PWM como salida, dado que en función del valor que se envíe el motor irá más o menos rápido.
pinMode(L_IS_1, OUTPUT); // Se define el pin L_IS como salida, aunque no se va a utilizar en esta aplicación pero se conecta por seguridad.
pinMode(L_EN_1, OUTPUT); // Se define el pin L_EN como salida, dependiendo del estado que se le aplique estará activo o no.
pinMode(L_PWM_1, OUTPUT); // Se define el pin L_PWM como salida, dado que en función del valor que se envíe el motor irá más o menos rápido.

digitalWrite(R_IS_1, LOW); // Se aplica el estado LOW, y por tanto desconectado, al pin R_IS.
digitalWrite(L_IS_1, LOW); // Se aplica el estado LOW, y por tanto desconectado, al pin L_IS.
digitalWrite(R_EN_1, HIGH); // Se aplica el estado HIGH, por tanto activo, al pin R_EN para permitir el movimiento horario.
digitalWrite(L_EN_1, HIGH); // Se aplica el estado HIGH, por tanto activo, al pin L_EN para permitir el movimiento antihorario.

/*Definición de estado de los pines implicados en el GRUPO MOTOR 2*/
pinMode(R_IS_2, OUTPUT); // Se define el pin R_IS como salida, aunque no se va a utilizar en esta aplicación pero se conecta por seguridad.
pinMode(R_EN_2, OUTPUT); // Se define el pin R_EN como salida, dependiendo del estado que se le aplique estará activo o no.
pinMode(R_PWM_2, OUTPUT); // Se define el pin R_PWM como salida, dado que en función del valor que se envíe el motor irá más o menos rápido.
pinMode(L_IS_2, OUTPUT); // Se define el pin L_IS como salida, aunque no se va a utilizar en esta aplicación pero se conecta por seguridad.
pinMode(L_EN_2, OUTPUT); // Se define el pin L_EN como salida, dependiendo del estado que se le aplique estará activo o no.
pinMode(L_PWM_2, OUTPUT); // Se define el pin L_PWM como salida, dado que en función del valor que se envíe el motor irá más o menos rápido.

digitalWrite(R_IS_2, LOW); // Se aplica el estado LOW, y por tanto desconectado, al pin R_IS.
digitalWrite(L_IS_2, LOW); // Se aplica el estado LOW, y por tanto desconectado, al pin L_IS.
digitalWrite(R_EN_2, HIGH); // Se aplica el estado HIGH, por tanto activo, al pin R_EN para permitir el movimiento horario.
digitalWrite(L_EN_2, HIGH); // Se aplica el estado HIGH, por tanto activo, al pin L_EN para permitir el movimiento antihorario.

/*Definición del pin analógico para el registro de voltaje*/
pinMode(A0, INPUT); // Pin analógico para determinar carga de batería
}

/*PROGRAMA GRUPO MOTOR*/
void leer_mensaje()
{
  InfoAPP = bluetooth.read(); // Se leen los datos que esta enviado el modulo bluetooth y se almacena en la variable InfoAPP

  if ( InfoAPP == 'a' ) // Cuando se recibe la variable "a" desde la APP se avanza
  {
    analogWrite(L_PWM_1, 150); // Se envia una señal PWM de 100 al GRUPO MOTOR 1 en sentido antihorario
    analogWrite(R_PWM_2, 150); // Se envia una señal PWM de 100 al GRUPO MOTOR 2 en sentido horario
  }

  if ( InfoAPP == 'v' ) // Cuando se recibe la variable "v" desde la APP se avanza a más velocidad
  {
    analogWrite(L_PWM_1, 250); // Se envia una señal PWM de 200 al GRUPO MOTOR 1 en sentido antihorario
    analogWrite(R_PWM_2, 250); // Se envia una señal PWM de 200 al GRUPO MOTOR 2 en sentido horario
  }

  if ( InfoAPP == 'r' ) // Cuando se recibe la variable "r" desde la APP se retrocede
  {
    analogWrite(R_PWM_1, 150); // Se envia una señal PWM de 100 al GRUPO MOTOR 1 en sentido horario
    analogWrite(L_PWM_2, 150); // Se envia una señal PWM de 100 al GRUPO MOTOR 2 en sentido antihorario
  }

  if ( InfoAPP == 't' ) // Cuando se recibe la variable "t" desde la APP se retrocede a más velocidad
  {
    analogWrite(R_PWM_1, 200); // Se envia una señal PWM de 100 al GRUPO MOTOR 1 en sentido horario
    analogWrite(L_PWM_2, 200); // Se envia una señal PWM de 100 al GRUPO MOTOR 2 en sentido antihorario
  }

  if ( InfoAPP == 'd' ) // Cuando se recibe la variable "d" desde la APP se gira a derecha
  {
    analogWrite(L_PWM_1, 150); // Se envia una señal PWM de 100 al GRUPO MOTOR 1 en sentido antihorario
    analogWrite(R_PWM_2, 50); // Se envia una señal PWM de 100 al GRUPO MOTOR 2 en sentido antihorario
  }
}

```

```
if ( InfoAPP == 'e' )          // Cuando se recibe la variable "e" desde la APP se gira a derecha con más velocidad
{
  analogWrite(L_PWM_1, 200); // Se envia una señal PWM de 100 al GRUPO MOTOR 1 en sentido antihorario
  analogWrite(R_PWM_2, 100); // Se envia una señal PWM de 100 al GRUPO MOTOR 2 en sentido antihorario
}

if ( InfoAPP == 'i' )          // Cuando se recibe la variable "i" desde la APP se gira a izquierda
{
  analogWrite(L_PWM_1, 50);   // Se envia una señal PWM de 100 al GRUPO MOTOR 1 en sentido horario
  analogWrite(R_PWM_2, 150);  // Se envia una señal PWM de 100 al GRUPO MOTOR 2 en sentido horario
}

if ( InfoAPP == 'z' )          // Cuando se recibe la variable "z" desde la APP se gira a izquierda con más velocidad
{
  analogWrite(L_PWM_1, 100);  // Se envia una señal PWM de 100 al GRUPO MOTOR 1 en sentido horario
  analogWrite(R_PWM_2, 200);  // Se envia una señal PWM de 100 al GRUPO MOTOR 2 en sentido horario
}

if ( InfoAPP == 's' )          // Cuando se recibe la variable "s" desde la APP se para
{
  analogWrite(R_PWM_1, 0);    // Se envia una señal PWM de 0 al GRUPO MOTOR 1 en horario
  analogWrite(L_PWM_1, 0);    // Se envia una señal PWM de 0 al GRUPO MOTOR 1 en sentido horario

  analogWrite(R_PWM_2, 0);    // Se envia una señal PWM de 0 al GRUPO MOTOR 2 en horario
  analogWrite(L_PWM_2, 0);    // Se envia una señal PWM de 0 al GRUPO MOTOR 2 en sentido horario
}
}

/*PROGRAMA LOOP*/

void loop()
{
  // Apartado para el calculo de porcentaje de bateria restante

  valorBAT = analogRead(A0);   // Variable que registra el voltaje recibido por la entrada analógica

  VBAT = valorBAT * 13.5 / 1023; // Calculo del voltaje de la bateria, transformación de numero analógico a voltaje

  //En función del voltaje calculado se asociará un porcentaje de bateria

  if (13.5 <= VBAT < 13.3){
    PBAT = 100;
  }

  if (13.3 <= VBAT < 13.1){
    PBAT = 90;
  }

  if (13.1 <= VBAT < 12.9){
    PBAT = 75;
  }

  if (12.9 <= VBAT < 12.7){
    PBAT = 60;
  }

  if (12.7 <= VBAT < 12.5){
    PBAT = 45;
  }

  if (12.5 <= VBAT < 12.3){
    PBAT = 30;
  }

  if (12.3 <= VBAT < 12.1){
    PBAT = 15;
  }
}
```

```
if (12.1 <= VBAT < 11.9){
  PBAT = 5;
}

if (11.9 <= VBAT){
  PBAT = 0;
}

// Tratamiento de datos mediante bluetooth

bluetooth.println(PBAT);          // Se envia al bluetooth el dato de la bateria
delay(1000);                      // Demora de 1 segundo

leer_mensaje();                  // para que se puedan ir leyendo todas las variables que se envien desde la APP se incluye
// esta sentencia para que cada vez que se ejecute el void loop, tambien se ejecute el void leer_mensaje.

// Serial.println(InfoAPP);       // por ultimo en caso de que se este utilizando el puerto serial se imprime
// la variable recibida por bluetooth para comprobarla manualmente.
}
```

## Anexo D: Recopilación de cálculos en Excel

A forma de resumen se recogen los cálculos realizados en Excel, que complementan el aparatado de dimensionamiento de componentes.

TFG Diseño de una sulfatadora móvil - Hoja de cálculos	
Álvaro Llamas Voltés	
15/01/2022	
Este documento tiene como objetivo mostrar los cálculos realizados durante el desarrollo del trabajo, y completeta la información recogida en la memoria.	

Cálculo Motor		
<b>DATOS</b>		
Masa del sistema	300 kg	
Gravedad	9,81 m/s <sup>2</sup>	
Coef estatico	1	
Diametro eje	0,16 m	
Velocidad de desplazamiento	5 m/min	
Pot bomba para sulfatación	204 W	
<b>CALCULOS</b>		
$F_F =$	2943 N	$F = F_F = \mu_e \cdot N$
$F =$	2943 N	
$M =$	235,44 Nm	$M = F \cdot d$
$n =$	9,94718394 rpm	$rpm = \frac{V}{d \cdot \pi}$
$P_{motor} =$	245,25 W	$Pot = M \cdot \frac{rpm}{60/2\pi}$
$P_{TOTAL} =$	1185 W	

Cálculo de Velocidad y Relacion de Transmision	
<b>DATOS</b>	
Diametro Motriz (Dm)	31 mm
Diametro Salida (Ds)	160 mm
RPM Motriz (Nm)	2700 rpm
RPM Salida (Ns)	523,125 rpm
Relacion de Transmision (Rt)	0,19375 -
<b>CALCULOS</b>	
$Dm \cdot Nm = Ds \cdot Ns$	$Ns = 523,125 \text{ rpm}$
$Rt = Dm / Ds = Ns / Nm$	$Rt = 0,19375 \quad 0,19375$
	Reduccion = 5,16129032
<b>SISTEMA REDUCTOR PORQUE RT &lt; 1</b>	

Cálculo Bateria		
<b>DATOS</b>		
Consumo l Motor	20 A	
Consumo Bomba Sulfatación	11 A	
Arduino Mega	0,093 A	
Tiempo Autonomia	4 h	
Cantidad Motores	4 Uds.	
Consumo reposo	10%	
<b>CALCULOS</b>		
$I_{TOTAL} =$	91,000093 A	Capacidad BAT 368 Ah
$I_{Extra} =$	92 A	
Tiempo de sulfatación =	92,08333333	

BATERIA			
Consumo (A)	Durante X tiempo (min)	Durante X tiempo (min)	Total (Ah)
8,093	1	0,016666667	0,134883333
80,093	20	0,333333333	26,69766667
11,093	92	1,533333333	17,00926667
80,093	20	0,333333333	26,69766667
8,093	1	0,016666667	0,134883333

Consumo Maximo 27 Ah

Para mantener margenes de descarga que no dañen la bateria se sobredimensiona este componente para que la capacidad sea del doble requerida como minimo

Capacidad recomendada 54 Ah



Cálculo Sección de cable

DATOS	
Long Motor DC	2 m
Intensidad Motor DC	28 A
Long Bomba	2 m
Intensidad Bomba	11 A
Long Bateria	1 m
Intensidad Mega	0,093 A
Constante cobre	56
Caida tension	2 %
Voltaje sistema	12 V
Voltaje caida	0,24 V

CALCULOS	
Motor DC - Bateria	
Sección:	8,33333333 mm <sup>2</sup>
Bomba - Bateria	
Sección:	3,27380952 mm <sup>2</sup>
Bateria - Arduino	
Sección:	0,01383929 mm <sup>2</sup>

Cálculo Monitorizar Voltaje Bateria

DATOS	
R2	10 kΩ
Vbat	12 V
Varduino	5 V

CALCULOS	
R1 =	14 kΩ      Tengo de 10k y 2k poner en serie 10+2=14

Para determinar si la bateria esta cargada o descargada se utilizará arduino. Se leerá la entrada anaogica de arduino para que dé un valor de entre 0 y 1023, se ajustará entre 0 y 100 % y se mostrará en la LCD o en la APP.  
 Para ajustar la entrada, máxima de 5 V en arduino, dado que la bateria es de 12 V se debe hacer un divisor de tensión que aqui se calcula.

Cálculo Bomba V2.0	
<b>DATOS</b>	<b>CALCULOS</b>
<p>Rugosidad absoluta (ε) 0,0015 mm                      Viscosidad Agua 20°C (μ) 0,001 kg/m·s = Pa·s                      Diámetro tubo (D) 20 mm                      Diámetro tubo (D) 0,02 m                      Velocidad Bombeo 15 lpm                      Densidad agua (ρ) 998 kg/m<sup>3</sup>                      Longitud del sistema (L) 50 m                      Gravedad (g) 9,81 m/s<sup>2</sup>                      Rendimiento Global (η) 0,6</p>	<p>Rugosidad relativa (εr) 0,000075                      Caudal (Q) 0,00025 m<sup>3</sup>/s                      Sección tubo (S) 0,000314159 m<sup>2</sup>                      Velocidad (u) 0,795774715 m/s                      Reynolds (Re) 15883,66332 Adimensional MAYOR A 4000 = TURBULENTO                      Reynolds (Re) 1,59E+04</p> <p>Coefficiente de fricción (f) 0,027 <span style="border: 1px solid black; padding: 2px;">En el diagrama de modi para Re y εr</span></p> <p>Perdidas rozamiento (hf) 2,178637836 m                      Trabajo bomba (W0) 2,178637836 m                      Potencia bomba (N) 0,118498291 CV <span style="border: 1px solid black; padding: 2px;">W0 = hf, porque no hay cambios ni de</span>                      Potencia bomba (N) 87,15534458 W</p>
<b>MANGUERA DE PE</b>	

Régimen {  
 Laminar: Re ≤ 2000 (nº Reynolds crítico inferior)  
 Transición: 2000 < Re ≤ 4000  
 Turbulento: Re > 4000 (nº Reynolds crítico superior)

Otra forma de escribir el número de Reynolds:

$$u = \frac{Q}{S} = \frac{4 \cdot Q}{\pi \cdot D^2}$$

$$Re = \frac{4 \cdot Q \cdot D \cdot \rho}{\pi \cdot D^2 \cdot \mu} = \frac{4 \cdot Q \cdot \rho}{\pi \cdot \mu \cdot D} = \frac{4 \cdot Q}{\pi \cdot D \cdot \nu} = Re$$

$$Re = \frac{4 \cdot Q}{\pi \cdot D \cdot \nu}$$

• Si el régimen es turbulento ⇒ ecuación de Fanning:

$$h_f = f \cdot \frac{L \cdot u^2}{2 \cdot g \cdot D} \quad f = \text{coeficiente de fricción}$$

$$\Rightarrow h_1 - h_2 + \frac{u_1^2 - u_2^2}{2 \cdot g} + \frac{P_1 - P_2}{\gamma} + W_o - h_{f1-2} = 0$$

RUGOSIDAD ABSOLUTA DE MATERIALES			
Material	ε (mm)	Material	ε (mm)
Plástico (PE, PVC)	0,0015	Fundición asfaltada	0,06-0,18
Poliéster reforzado con fibra de vidrio	0,01	Fundición	0,12-0,60
Tubos estirados de acero	0,0024	Acero comercial y soldado	0,03-0,09
Tubos de latón o cobre	0,0015	Hierro forjado	0,03-0,09
Fundición revestida de cemento	0,0024	Hierro galvanizado	0,06-0,24
Fundición con revestimiento bituminoso	0,0024	Madera	0,18-0,90
Fundición centrifugada	0,003	Hormigón	0,3-3,0

$$Re = \frac{u \cdot D \cdot \rho}{\mu}$$

Cálculo Emisiones CO2	
<b>DATOS</b>	<b>CALCULOS</b>
<p>1 L de gasolina emite 2,37 kg de CO2                      Consumo motor 0,54 L/hora                      1 hora de sulfatación al año 50 horas                      Precio gasolina 1,5 €/L</p>	<p>Emisiones por hora 1,2798 kg CO2/hora                      63,99 kg CO2/año</p> <p>L consumidos anual 27 L                      Coste 40,5 €</p>

Cálculo Sistema de carga	
<b>DATOS</b>	<b>CALCULOS</b>
<p>Voltaje Batería 12 V                      Capacidad Batería 90 Ah                      Voltaje Solar 12 V                      Potencia Panel Solar 380 W                      Numero de Paneles Solares 1 Uds.</p>	<p>Energía Necesaria para cargar la batería 1080 Wh <span style="border: 1px solid black; padding: 2px;">Se necesitaran 1.080 W para cargar</span></p> <p>Potencia pico Instalación Solar 380 W</p> <p>Tiempo de carga con el panel seleccionado 2,84210526 h      170,526316 min      10231,5789 s</p> <p>Tiempo = 2 h      50,5263158 min      31,5789474 s</p> <p style="text-align: center;">Tiempo de carga con un panel de 380 W</p>