

**ELCO 2018**

**MEMORIA DEL PROYECTO REALIZADO**



**GRUPO 8**

Irene Ayuso Yuste  
Alda Martín Muñoz  
Miguel García Maya  
Javier Otero Martínez  
Alejandro Gómez Molina  
Daniel Sánchez García-Monge

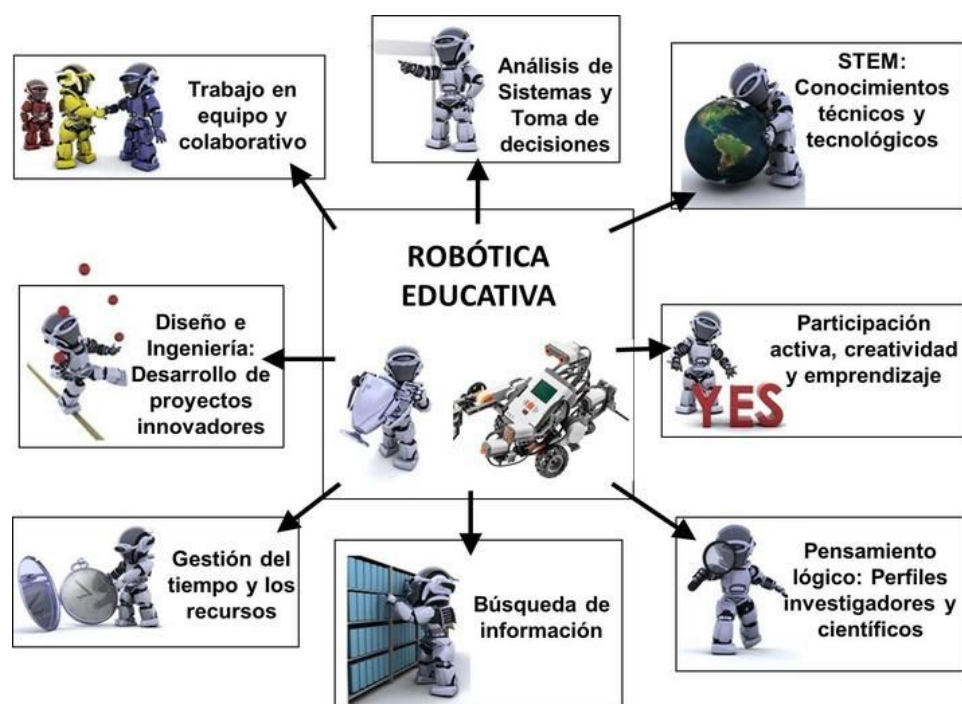
# DoiT: do it yourself, do it now!

1.	Introducción	2
2.	Mercado y análisis DAFO	3
3.	Descripción de las partes	4
3.1.	Estructura y diseño	4
3.2.	Comunicaciones	4
3.3.	Cámara y gafas VR	5
3.4.	Sensores de las gafas: acelerómetro y giroscopio	6
3.5.	Motores	6
4.	Prototipo	7
4.1.	Mecánica y hardware del prototipo	7
4.2.	Software del prototipo	9
4.3.	Pruebas de funcionamiento	10
5.	Producto final	11
5.1.	Modificaciones respecto al prototipo	11
5.2.	Proveedores	12
5.3.	Publicidad y distribución	12
5.4.	Presupuestos	13
6.	Posibles mejoras y conclusiones	14
7.	Bibliografía y referencias	14

# 1. Introducción

DoiT es un proyecto que surge de la idea de acercar la robótica a estudiantes de todas las edades de una forma amena y accesible. Su funcionamiento se basa en la implementación modular en la que el usuario dispone los módulos que desee sobre un soporte con ruedas a modo de vehículo, acompañada de una plataforma online de código abierto a la disposición de cualquiera que quiera implementar un nuevo módulo o modificar uno existente. DoiT ofrece, por tanto, la posibilidad de que sea el estudiante el que escoja el nivel de dificultad del proyecto que desea emprender: desde comprar los módulos ya hechos y cambiar su funcionamiento, hasta montarlo él mismo con una guía de ayuda.

DoiT trata de llenar un hueco que existe en un mercado en auge como es el de la realidad virtual enfocándolo de manera didáctica.



El prototipo diseñado a modo de ejemplo de las posibilidades de DoiT consiste en un coche robot teledirigido mediante los movimientos de la cabeza que consta con una cámara para que el usuario vea en primera persona lo que ve el propio coche. Teniendo claras las funcionalidades de las que iba a disponer el prototipo, se estructuró el desarrollo del prototipo en las siguientes partes diferenciadas:

- Estructura y diseño del coche: plataforma de plástico con ruedas de dos pisos sobre la que se podrían conectar los distintos módulos.
- Protocolo de comunicaciones serie: modificación del I2C.
- Cámara FPV que envíe datos al móvil que se inserte en las gafas.
- Alimentación del prototipo.
- Control de motores.
- Control del brazo robótico.
- Sensores sobre las gafas: acelerómetro y giroscopio para dar directrices al coche.

Una vez acabado el prototipo se abordaron los detalles que asegurasen la viabilidad de cara a venderlo como producto final. Más adelante se detallará el cálculo de los costes asociados al mismo, así como la publicidad y la distribución previstas para su comercialización.

## 2. Mercado y análisis DAFO

El mercado de la robótica hoy en día tiene un enorme potencial que es extensible a muchos ámbitos. Dada la infinidad de posibilidades que hay en este terreno, es comprensible tratar de hacerse un hueco en el mercado explotándolo como algo didáctico. A ello hay que sumarle que el uso de la realidad virtual, que también está en auge hoy en día, le aporta un gran atractivo. Según las estimaciones de varias empresas como IDC y App Date, el mercado de la realidad virtual espera superar los 45 millones de euros solo este año.

En cuanto a los puntos a destacar del equipo en este proyecto, contamos con fortalezas como la sencillez y versatilidad del producto para el usuario, y el hecho de que sea un terreno un tanto desconocido y la inexperiencia del equipo en el sector como debilidades. En cuanto al análisis externo, contamos con que es un mercado en auge que no para de crecer, y por tanto la demanda va a estar asegurada. Por otro lado, los competidores que ya hay establecidos pueden resultar una amenaza.



### 3. Descripción de las partes

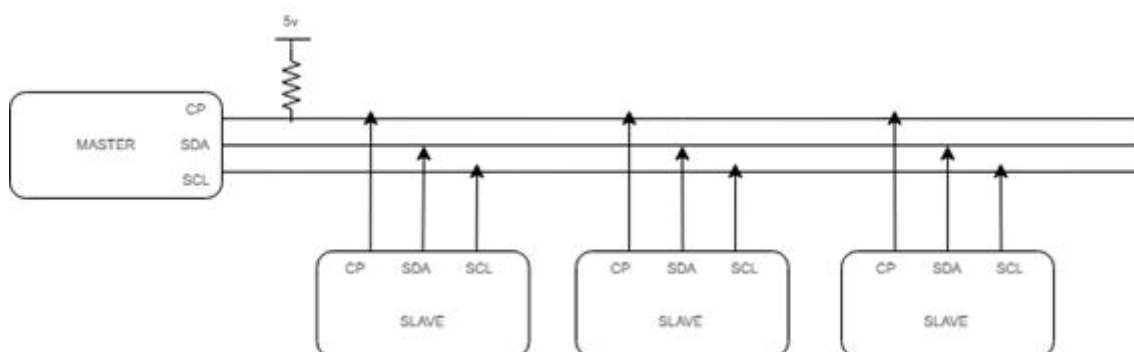
#### 3.1. Estructura y diseño

El diseño se basa en la filosofía modular del proyecto. Se trata de una base rígida en la que poder ir acoplado el resto de los módulos. Por un motivo de espacio, se acoplarán los módulos en dos pisos diferenciados, pero con unas tomas de tierra y alimentación comunes. La batería se colocará al fondo del todo para mejorar la tracción de los motores y distribuir el peso convenientemente, para evitar vuelcos. Evidentemente, el brazo se ha colocado delante para poder agarrar los objetos que queden delante, es por ello que se coloca la cámara en la parte superior, para poder grabar la frontal del vehículo sin obstáculos.

No obstante, el orden en la interconexión de los diferentes módulos es totalmente indiferente, pudiéndose modificar en función de los deseos del usuario. Hemos utilizado el esquema anterior mencionado pero hay muchas otras configuraciones igualmente válidas.

#### 3.2. Comunicaciones

El protocolo de comunicaciones serie está basado en I2C con algunas modificaciones. Todos los dispositivos están conectados entre ellos y tienen la posibilidad de comunicarse a través de un nodo maestro que actúa como punto de acceso. Para poder transmitir datos, los nodos (que generalmente están en modo esclavo) pueden cambiar su rol a maestro y mediante una línea adicional tomar el control del bus (Si la línea está ocupada, se quedan a la espera hasta que esté disponible o hasta alcanzar un TIMEOUT). Al tomar el control del bus, el nodo coordinador se pone en modo esclavo con una dirección fija (por defecto 0x00) y recibe los datos. Al terminar la transmisión el dispositivo vuelve al rol de esclavo y desactiva la línea.



El nodo coordinador es el encargado de redirigir las tramas a su destino, controlar los dispositivos que se encuentran conectado y atender las peticiones de bus de los dispositivos. Por otro lado, también hace de interfaz con el transmisor de RF, recibiendo los datos y redirigiendo la trama, con el formato adecuado, a su destino.

El protocolo, por simplicidad, sólo controla un tipo de trama que contiene 3 bytes de preámbulo y 6 bytes de datos. En la parte de preámbulos tenemos un campo de origen (1 byte), destino (1 byte) y comando (1 byte).

SRC	DEST	CMD	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6
-----	------	-----	--------	--------	--------	--------	--------	--------

Las direcciones de origen y destino se corresponden con las direcciones de 7 bits usados por el protocolo I2C. El campo de comando tiene en cuenta 3 tipos de acciones (en esta primera versión):

- Transmitir: transmite 6 bytes de datos al dispositivo especificado.
- Solicitar: Solicita datos a dispositivo especificado. Pueden usarse los 6 bytes de datos como parámetros.
- Autenticar: Notifica su existencia al nodo coordinador.

<i>CMD</i>	<i>VALOR</i>	<i>DESCRIPCION</i>
AUTH	0x55	Reporta su identificación al <u>Master</u>
DATA_IN	0x88	Inicio de trama de datos
DATA_OUT	0x25	Solicita el dato especificado por el byte 1.

### 3.3. Cámara y gafas VR

Por sencillez, el uso de la cámara instalada en el vehículo y de las gafas de realidad virtual se implementó por separado al resto de comunicaciones. Para ello, la cámara hace uso de la tecnología OTG y manda lo que ve en tiempo real a un receptor que va instalado en las gafas y conectado al móvil para que el usuario lo vea en primera persona.

### **3.4. Sensores de las gafas: acelerómetro y giroscopio**

Los sensores del casco emplean el integrado MPU6050 para Arduino, que dispone de un acelerómetro y un giroscopio. Estos se encargan de conocer los movimientos que hace el usuario con la cabeza, información que pasa por el arduino integrado en el casco para traducirlo en los movimientos que el coche debe efectuar, se envía mediante un transmisor RF al arduino del módulo central, y se de la orden que corresponda a los motores.

Según los valores leídos por los sensores se mandará a los motores si el coche debe avanzar, retroceder o quedarse parado, así como girar a la derecha, girar a la izquierda o mantenerse recto y se mandará a la cámara el grado de rotación. El arduino empleado por los sensores, también se encargará de recibir la señal de un botón situado en las gafas que acciona el movimiento del brazo.

### **3.5. Motores**

Para el movimiento del rover se van a utilizar dos motores idénticos de la serie 37D de Pololu, requiriéndose una alimentación de 12V. Los motores irán directamente conectados a los drivers, que además de proteger en corriente al conjunto, funciona como puente en H. Este montaje permite variar el sentido de giro de forma muy sencilla, y además a bajo precio. En nuestro caso, hemos optado por el modelo L298N para realizar estas funciones.

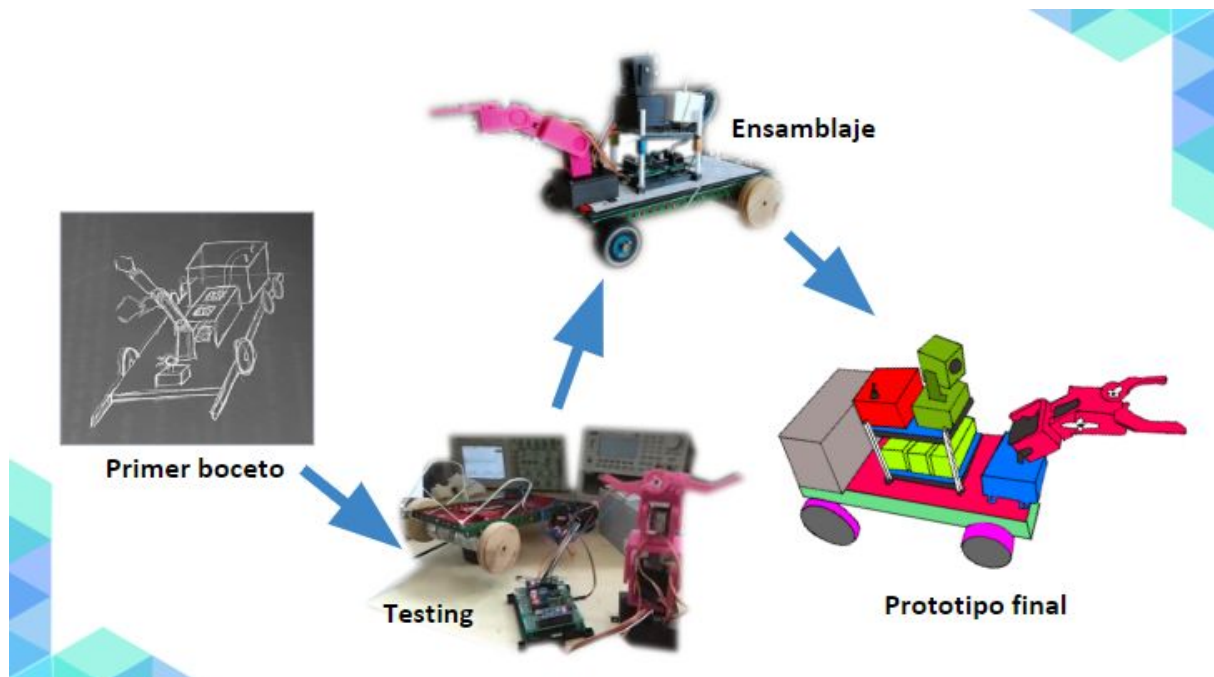
Estos motores tienen un par muy elevado, ya que montan una reductora 100:1. Esta gran relación de reducción provoca que la velocidad del vehículo no sea muy elevada, pero en este tipo de rover es mucho más conveniente que el motor tenga la fuerza necesaria para mover el vehículo que una gran velocidad, haciendo estos motores los más adecuados.

Finalmente hemos acoplado las ruedas de madera a estos motores mediante unas piezas destinadas a tal fin proporcionados por el fabricante. Además de colocar y pegar unas gomas en las ruedas para que no patinen en superficies lisas. Las ruedas delanteras van locas y no tienen ningún tipo de motorización.

## 4. Prototipo

En este apartado se detallan la mecánica, el hardware y el software que fueron necesarios para obtener el prototipo, así como las pruebas de funcionamiento que se realizaron una vez acabado.

Aunque la idea inicial era algo más ambiciosa, los resultados de las pruebas fueron satisfactorios y su funcionamiento se dió según lo esperado.



### 4.1. Mecánica y hardware del prototipo

#### Componentes del prototipo

La mecánica del prototipo consiste en una sencilla estructura de dos plataformas de plástico donde se conectan los módulos, una encima de otra, que se apoyan en cuatro ruedas.

El hardware hace uso de tres Arduinos Nano que son fundamentales para procesar la información recogida por los sensores, comunicarse, y dar órdenes a la cámara, motores y brazo robótico en función de la misma.

blablabla

Los componentes empleados para el prototipo completo quedan recogidos en la siguiente tabla:



Módulo principal	Brazo robótico	Módulo de las gafas de control
<ul style="list-style-type: none"> <li>● Batería</li> <li>● Base de plástico</li> <li>● Base de módulos</li> <li>● Ruedas (x4)</li> <li>● Regulador de tensión</li> <li>● Módulo FPV <ul style="list-style-type: none"> <li>○ Cámara</li> <li>○ Emisor</li> <li>○ Receptor</li> </ul> </li> <li>● Módulo master de comunicaciones <ul style="list-style-type: none"> <li>○ Arduino Nano</li> <li>○ Antena RF</li> </ul> </li> <li>● Módulo de motores <ul style="list-style-type: none"> <li>○ Arduino Nano</li> <li>○ Motores (x2) DC12v 100rpm</li> <li>○ Driver de motores L298N</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>● Servos (x4)</li> <li>● Piezas de plástico (x6)</li> <li>● Arduino Nano</li> </ul>	<ul style="list-style-type: none"> <li>● Gafas VR</li> <li>● Arduino Nano</li> <li>● Integrado MPU-6050 con acelerómetro y giroscopio</li> <li>● Batería de plomo</li> <li>● Antena RF</li> </ul>

### **Consumo y especificaciones técnicas**

Para realizar un cálculo adecuado de la autonomía del prototipo, tuvimos que tomar en cuenta el consumo en reposo y otra cifra en pleno funcionamiento. Con esto estimamos el consumo medio, y por tanto, la duración de la batería.

En el caso de que los motores estén apagados, solo hay que contar el consumo de los cinco Arduino Nano que a de 19mA cada uno [2]. Tendremos, por tanto, un consumo de en torno a 100mA (caso peor). Es evidente que los motores requieren de mucha mas corriente que los Arduino, pudiendo despreciar el consumo de estos últimos a plena carga, donde los motores precisan de 5A. La utilización de corriente de los drivers de los motores también se desprecia frente al de los propios motores.

Con estas cifras, y las características de la batería [3], pudimos estimar la duración de la batería en reposo en un día (a 100mA), mientras que a plena carga es de 20 min (a 10A). Teniendo en cuenta que el avance del dispositivo no se lleva a cabo a la máxima velocidad posible, y que todas estas cifras están previstas para el caso peor, pudimos estimar la autonomía máxima de la batería en tres horas. Todo este cálculo está realizado en el

supuesto de temperatura nominal: 25°C, si ésta aumenta, la autonomía se reduce y viceversa.

## 4.2. Software del prototipo

Dado el uso de los arduinos para el prototipo, el lenguaje de programación empleado es C++, haciendo uso también de las librerías propias de arduino. Los bloques a programar han sido los siguientes:

- Modificación del protocolo I2C para las comunicaciones serie, especificado en el apartado 3.2. Las funciones principales de la implementación software son:

Estados: (type L\_STATUS);

STATUS\_OK -> Acción completada satisfactoriamente.

STATUS\_ERROR -> Error en la acción.

STATUS\_TIMEOUT -> Tiempo máximo de espera finalizado.

Funciones principales:

Función	Descripción
void tras_begin(uint8_t controlPin);	Inicia el protocolo en modo master
void tras_begin(uint8_t addr, uint8_t controlPin)	Inicia el protocolo en modo slave
bool data_available()	Detecta si hay nuevos datos en el buffer
void readData(uint8_t* data)	Devuelve un puntero al buffer
L_STATUS sendData(uint8_t* data, uint8_t cmd )	Envía una trama en modo slave
L_STATUS sendData(uint8_t addr, uint8_t cmd, uint8_t* data)	Envía trama en modo master
L_STATUS reqData(uint8_t addr, uint8_t* data)	Solicita datos al un slave

- Uso del acelerómetro y giroscopio para dar las órdenes de movimiento a los motores

El software aquí desarrollado lleva a cabo las siguientes funciones, desde que los sensores leen los valores hasta que se traducen en el movimiento que el vehículo va a tener que hacer:

- 1.- Calibrado de las medidas tomadas por los sensores: esta función se ejecuta siempre cuando se resetea el integrado MPU6050. Consiste en ir reduciendo el offset de los ejes por recurrencia hasta que los valores del acelerómetro son menores que (0, 0, 16384) y los del giróscopo son (0, 0, 0), para los ejes x, y, z, respectivamente.

2.- Medición del ángulo de inclinación (con el acelerómetro) y del de rotación (con el giroscopio). Cabe destacar, que el movimiento de rotación en el eje z se realizará por lo tanto empleando sólo el giroscopio, por lo que se irá introduciendo poco a poco un error mayor. Para evitar este error habría que emplear otro integrado que además incorporase magnetómetro.

3.- Escalado de las lecturas obtenidas de aceleración y velocidad angular: para obtener los valores del acelerómetro en  $m/s^2$  (aceleración) y los valores del giroscopio en  $^\circ/s$  (velocidad angular).

4.- Filtro de complemento: combina los ángulos medidos para obtener el resultado final con el menor error posible.

- Diseño de un programa de control de motores para el movimiento de las ruedas

Este programa ha resultado bastante sencillo. Una vez implementado el protocolo de comunicaciones, no tiene más que quedar “escuchando” la orden que le llega. Una vez recibe una orden, la compara con las acciones definidas en distintas funciones (todo implementado mediante el software de Arduino). Finalmente, ejecuta la orden si la hubiere e imprime una pequeña traza por el puerto serie, muy útil para depurar su correcto funcionamiento.

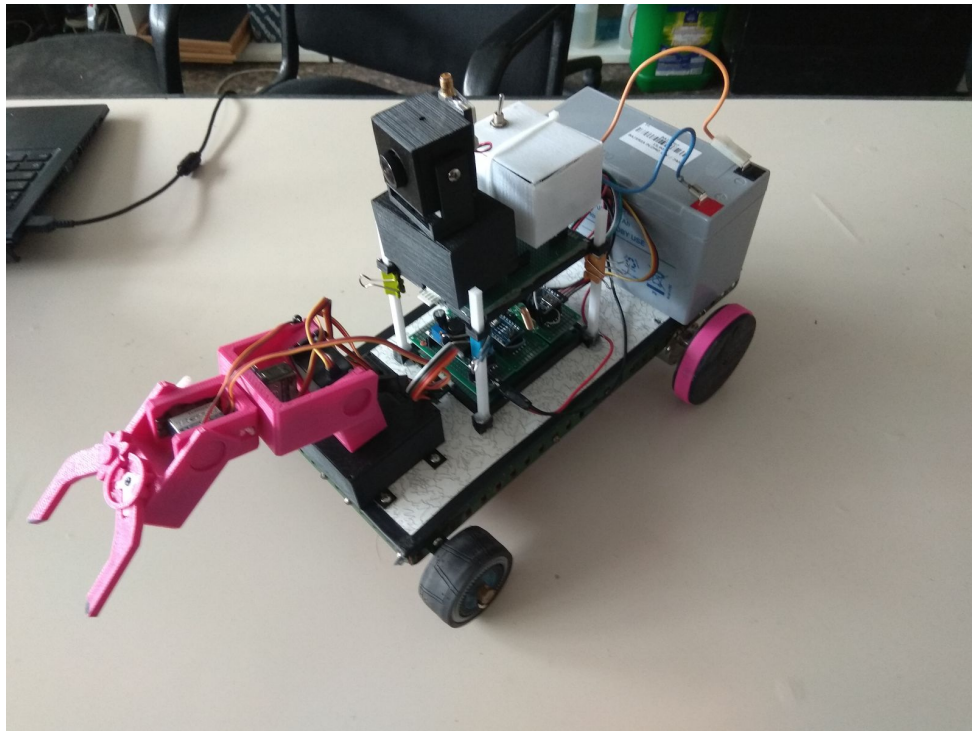
Las diferentes órdenes se clasifican en dos tipos: movimiento y giros. En primer lugar está el movimiento, que puede ser hacia adelante, hacia atrás o parado. Es importante mencionar que los motores no funcionan nunca a su velocidad máxima, ya que el rover iba demasiado rápido como para que resultara cómodo de manejar y además se incrementa la duración de la batería. Para revertir el sentido de giro, únicamente se invierten los valores enviados al puente en H, que son 2 y alternos (cuando uno vale 1 el otro vale 0 y viceversa). El movimiento de giro se logra haciendo girar la rueda externa a mayor velocidad que la interna. En nuestro caso, se ralentiza la interna también para lograr un ángulo de giro más pronunciado, ya que fuimos probando hasta encontrar las velocidades más apropiadas para el buen manejo del prototipo.

### **4.3. Pruebas de funcionamiento**

Para comprobar el correcto funcionamiento del prototipo hicimos pruebas a cada uno de los módulos del robot por separado para asegurarnos que todos ellos cumplieran con su función. Luego, pasamos a la integración de todos los módulos y a probar el

funcionamiento del prototipo completo. Los problemas que surgieron se debieron fundamentalmente a:

- La comunicación de las gafas con el módulo receptor del robot. Las antenas del prototipo tenían poco alcance y alguna de las órdenes se perdían en la transmisión. De cara al producto final hemos mejorado este alcance incluyendo unas antenas con mayor radio de alcance.
- La imposibilidad de obtener valores coherentes del acelerómetro y del giroscopio. Inicialmente usamos el integrado para arduino 10DOF-GY80 del que disponíamos. Tras varios días tratando de leer las coordenadas en el entorno de desarrollo con una placa de inserción viendo la imposibilidad de hacerlo funcionar, decidimos cambiar de integrado a uno más comercial del que había más documentación y que funcionó a la perfección desde el principio.



## 5. Producto final

### 5.1. Modificaciones respecto al prototipo

Las principales diferencias entre el prototipo realizado y el producto de venta final serán el material de la estructura del robot, las gafas de realidad virtual y los circuitos electrónicos de los módulos. En el producto final, el material de la estructura del coche será plástico, ya que se encargarán piezas de inyección a una empresa externa para la fabricación de la primera tirada de producto. Las gafas de realidad virtual se comprarán fabricadas a un proveedor chino y los circuitos electrónicos de los módulos principales se presentarán en PCBs que faciliten la inserción de los componentes.

## **5.2. Proveedores**

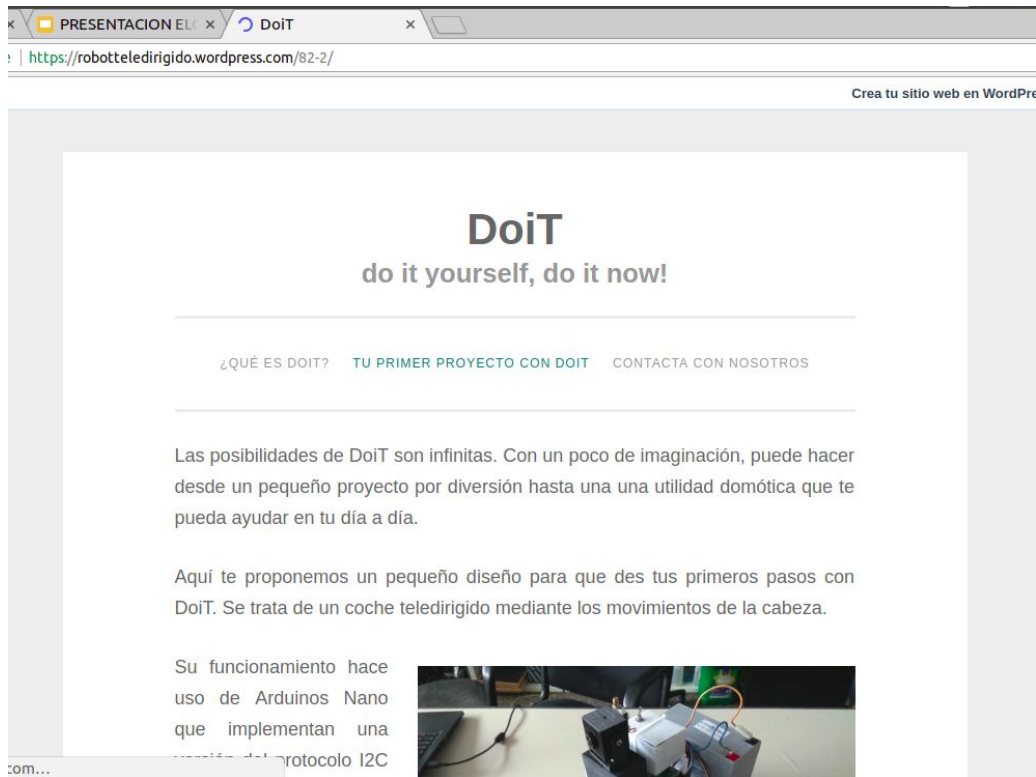
Los proveedores principales escogidos para el suministro de cada uno de los componentes son Alibaba y Bangood. En la tabla del apartado 6 adjunta más abajo se detalla el proveedor para cada componente.

## **5.3. Publicidad y distribución**

La publicidad prevista para el producto final se realizará mediante:

- Anuncios en plataformas de contenido multimedia.
- Anuncios en redes sociales.
- Google Ads.

Para reforzar la imagen corporativa de Dolt y hacer más atractivo el producto, diseñamos una página web sencilla que explicara cómo funciona la plataforma y la utilidad didáctica del producto en el campo de la robótica.

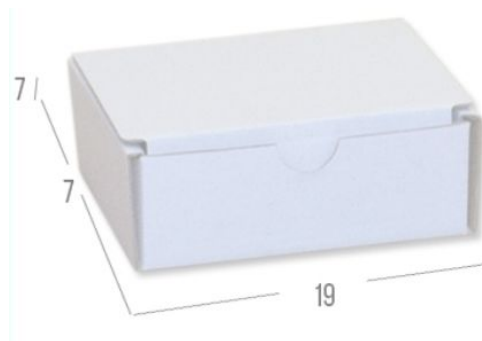


<https://robottedirigido.wordpress.com>

En cuanto a la distribución, el producto se enviará mediante empresas de paquetería. Finalizado el prototipo, medimos el tamaño del mismo y escogimos unas dimensiones adecuadas para el embalaje que dejaran cierto margen. El módulo del brazo robótico se empaqueta por separado.



Módulo principal



Módulo del brazo robótico

## 5.4. Presupuestos

La estimación realizada en este apartado se calcula suponiendo una fabricación de 10.000 unidades.

Dadas las cifras presentadas más abajo, con esta cantidad de unidades el coste unitario asciende a 92€, por lo que fijamos el precio de venta final en 200€.

Componente	Uds/robot	Precio/ud	Proveedor	Unidades necesarias	Precio total (€)
Motores	2	6,86	Bangood	20000	137200
Driver motores	1	0,08	Alibaba	10000	800
Piezas adaptación	2	0,01	Alibaba	20000	200
Cámara	1	2,5	Alibaba	10000	25000
Arduino Nano	5	2,5	Alibaba	50000	125000
Servo RC	5	0,84	Alibaba	50000	42000
Antena emisora FPV	1	24	Alibaba	10000	240000
Emisor y receptor RF	1	0,83	Bangood	10000	8300
Batería del robot	1	5,2	Alibaba	10000	52000
Transformador	1	0,43	Alibaba	10000	4300
Batería de las gafas	1	0,84	Alibaba	10000	8400
Gafas FPV	1	2,53	Alibaba	10000	25300
Receptor FPV	1	15,15	Alibaba	10000	151500
Piezas de plástico	1	10	Alibaba	10000	100000

## **6. Posibles mejoras y conclusiones**

Entre las posibles mejoras del producto final, la principal es la implementación de nuevos módulos que permitan darle más funcionalidades al robot, como por ejemplo un módulo de personalización que incluya leds junto con un sensor de luminosidad o un altavoz para añadir sonidos y hacer del vehículo algo similar a un juguete, o un módulo de sensores que permita al robot moverse de forma autónoma sin chocarse. Además, se podrían plantear diferentes movimientos para el brazo robótico con un mando en lugar de con un botón.

Como conclusión, podemos afirmar que el prototipo ha dado un buen resultado en base a la idea inicial que tuvimos de él. Quizás la parte más costosa por los problemas que ha dado ha incurrido en montar la mecánica del robot. Como ya hemos comentado, el principal contratiempo que nos encontramos a la hora de probarlo fue el poco alcance real de las antenas frente a las

especificaciones que nos esperábamos. A esto hay que sumarle que hubo componentes que nos costó conseguir, y que la programación de los sensores del casco tardó en arrancar hasta que decidimos cambiar de integrado.

Sin duda ha sido un proyecto gratificante con el que hemos podido poner en práctica conocimientos estudiados en el grado, referidos principalmente al diseño electrónico y al desarrollo software.

## **7. Bibliografía y referencias**

[1] Características de los motores, fabricante: <https://www.pololu.com/product/1106>

[2] Consumo de los Arduino, datasheet ubicado en <https://store.arduino.cc/arduino-nano>

[3] Características de la batería, datasheet:

<https://www.electronicaembajadores.com/datos/pdf1/ba/ba11/ba11704.pdf>

[4] Implementación de los sensores del casco mediante el MPU-6050:

[https://naylorlampmechatronics.com/blog/45\\_Tutorial-MPU6050-Aceler%C3%B3metro-y-Giroscopio.html](https://naylorlampmechatronics.com/blog/45_Tutorial-MPU6050-Aceler%C3%B3metro-y-Giroscopio.html)