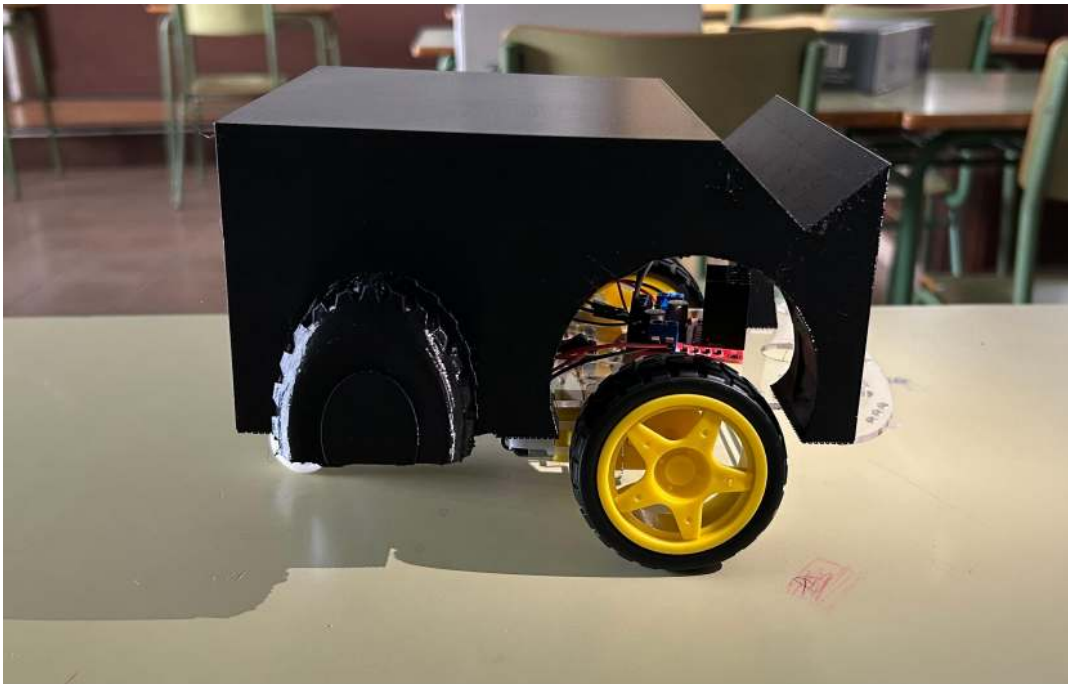




Institut Puig Castellar

Santa Coloma de Gramenet



RoboCar BT

(Projecte de desenvolupament)
CFGM Sistemes Informàtics i Xarxes

Autores: Ashar Ali , Nahim Leon
Grupo: SMX2D
Curso: CFGM SMX



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

C) Copyright

© (l'autor/a)

Todos los derechos reservados. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendido la impresión, la reprografía, el microfilm, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

Resumen del proyecto:

El proyecto RoboCar BT consiste en crear un coche robot controlado desde un móvil Android mediante Bluetooth. El objetivo es montar un vehículo con Arduino UNO R3, módulo Bluetooth HC-05, puente H L298N y una aplicación personalizada hecha con MIT App Inventor que envíe comandos A (adelante), B (atrás), C (izquierda), D (derecha) y E (parar).

El resultado es un prototipo completamente funcional que se mueve exactamente como ordena la app, demostrando control preciso por Bluetooth e integración perfecta de hardware y software básico de robótica.

Paraules clau (entre 4 y 8):

Arduino, Módulo Bluetooth HC-05, Módulo L298N, Kit coche robot, MIT App Inventor, control remoto

Índex

| | | |
|-------|--|----|
| 1 | Introducción..... | 6 |
| 1.1 | Contexto..... | 6 |
| 1.2 | Justificación..... | 7 |
| 1.3 | Objetivos..... | 7 |
| 1.3.1 | Objetivos generales..... | 7 |
| 1.3.2 | Objetivos específicos..... | 7 |
| 1.4 | Estrategia y planificación del proyecto..... | 8 |
| 1.5 | Metodología del trabajo..... | 8 |
| 1.6 | Estudio económico y presupuestario..... | 9 |
| 2 | Descripción del proyecto..... | 10 |
| 2.1 | Análisis de requisitos..... | 10 |
| 2.1.1 | Requisitos funcionales..... | 10 |
| 2.1.2 | Requisitos no funcionales..... | 11 |
| 2.2 | Previsión de tareas de investigación..... | 11 |
| 2.3 | Tecnologías..... | 12 |
| 2.3.1 | Comparativa de las tecnologías valoradas..... | 12 |
| 2.3.2 | Tecnologías escogidas..... | 12 |
| 2.4 | Estructura del proyecto..... | 13 |
| 2.5 | Descripción de los componentes..... | 13 |
| 2.5.1 | Componente 1..... | 13 |
| 2.5.2 | Componente 2..... | 13 |
| 2.5.3 | Componente 3..... | 13 |
| 2.5.4 | Componente 4..... | 14 |
| 2.5.5 | Componente 5..... | 14 |
| 2.5.6 | Componente 6..... | 15 |
| 2.5.7 | Componente 7..... | 15 |
| 2.6 | Definición de las funcionalidades..... | 15 |
| 2.6.1 | Funcionalidad 1: conexión entre el móvil y el coche..... | 15 |
| 2.6.2 | Funcionalidad 2: avance del coche..... | 15 |
| 2.6.3 | Funcionalidad 3: retroceso del coche..... | 15 |
| 2.6.4 | Funcionalidad 4: giro a la izquierda..... | 16 |
| 2.6.5 | Funcionalidad 5: giro a la derecha..... | 16 |
| 2.6.6 | Funcionalidad 6: parada del coche..... | 16 |
| 3 | Desarrollo del proyecto..... | 16 |
| 3.1 | Montaje del coche..... | 16 |
| 3.2 | Creación del chasis..... | 20 |
| 3.3 | Creación Página web..... | 21 |
| 4 | Conclusiones..... | 21 |

| | |
|---|----|
| 4.1 Conclusiones generales del proyecto..... | 21 |
| 4.2 Consecución de los objetivos..... | 21 |
| 4.3 Valoración de la metodología y planificación..... | 22 |
| 5 Visión de futuro..... | 22 |
| 6. Glossario..... | 23 |
| 7. Bibliografía..... | 25 |
| 8 Anexos..... | 26 |
| 8.1 Fotos..... | 26 |
| 7.2 Código Arduino:..... | 31 |
| 7.3 Carcasa 3D en Tinkercad: | |
| Video-3D_RoboCar BT.mp4..... | 33 |

1 Introducción

El proyecto RoboCar BT consiste en el desarrollo de un coche robot controlado remotamente desde un móvil Android mediante conexión Bluetooth. El objetivo principal es conseguir que el vehículo se mueva según las órdenes que damos desde la aplicación: adelante, atrás, giro izquierda, giro derecha y parar.

Para lograrlo se utilizan componentes electrónicos básicos: una placa Arduino UNO R3 como cerebro principal, módulo Bluetooth HC-05 para recibir los comandos del móvil, puente H L298N para controlar los dos motores del carrito y una aplicación personalizada creada con MIT App Inventor que envía las letras A, B, C, D y E correspondientes a cada movimiento.

1.1 Contexto

Se ha decidido desarrollar este proyecto debido al interés en el ámbito de la programación con Arduino y la creación de robots. Por este motivo, se ha planteado la construcción de un coche robot controlado mediante Arduino, combinando ambos campos.

Tras analizar distintas opciones, se ha observado la disponibilidad de kits de coches robot a bajo coste que incluyen la mayoría de los componentes necesarios para el montaje, como el chasis, las ruedas, la placa Arduino y el puente H. No obstante, para permitir la conexión con un dispositivo móvil, será necesario incorporar adicionalmente un módulo Bluetooth HC-05, baterías y cables jumper, que se adquirirán por separado.

El desarrollo del proyecto se divide en dos partes: por un lado, se realizará el montaje físico del vehículo y la comprobación del correcto funcionamiento de los motores mediante el módulo L298N; por otro, se elaborará el código en Arduino que permitirá interpretar las señales enviadas desde el móvil, así como el diseño de una aplicación con botones para transmitir dichas órdenes.

El objetivo es lograr un funcionamiento óptimo y sin fallos del coche robot, con el fin de presentarlo en clase y demostrar la capacidad de integrar componentes electrónicos, establecer comunicación entre ellos y resolver posibles incidencias. Este proyecto constituye una primera aproximación práctica al campo de la robótica dentro del proceso de aprendizaje.

1.2 Justificación

La razón por la que se ha elegido este proyecto es que llama mucho la atención el mundo de los robots y de la programación con Arduino, y se ha querido aprovechar la ocasión para profundizar de verdad en este tema. Aunque en el Grado Medio no se ha dado programación como tal, se considera que este coche controlado por Bluetooth es una buena forma de empezar, porque combina montaje de hardware, conexiones básicas y un código sencillo que se puede entender paso a paso.

Además, se ve como una manera práctica de demostrar que se es capaz de aprender cosas nuevas por cuenta propia, buscar información, adaptar ejemplos y conseguir que todo funcione. Con este proyecto se pretende ir un poco más allá de las prácticas habituales de clase y tener algo que se pueda enseñar funcionando para que se vea el esfuerzo y el interés por la robótica y por seguir aprendiendo en este campo.

1.3 Objetivos

1.3.1 Objetivos generales

El objetivo general del proyecto es diseñar y poner en funcionamiento un coche robot controlado por Bluetooth desde un teléfono móvil, utilizando una placa Arduino y componentes básicos de robótica, para demostrar que somos capaces de montar el hardware, hacer las conexiones y programar un comportamiento sencillo, aunque no hayamos dado programación en el ciclo.

1.3.2 Objetivos específicos

- Montaje del chasis del coche, junto con las ruedas, los motores y el sistema de alimentación, para que el vehículo pueda moverse de forma autónoma y tenga una base estable.
- Conexión de la placa Arduino con el puente H y el módulo Bluetooth siguiendo un esquema correcto, ya que el Arduino se encarga de dar las órdenes, el puente H controla los motores y el Bluetooth permite la comunicación con el móvil.
- Programación del Arduino para interpretar los comandos recibidos por Bluetooth y convertirlos en movimientos del coche, como avanzar, retroceder, girar o detenerse.

- Diseño de una aplicación sencilla en el móvil mediante MIT App Inventor, con botones claros para enviar las órdenes al coche de forma fácil y rápida.
- Realización de pruebas del sistema completo para comprobar que todo funciona correctamente, corregir posibles errores y dejar constancia del proceso en la memoria del proyecto.

1.4 Estrategia y planificación del proyecto

Para llevar a cabo este proyecto se ha valorado la posibilidad de comprar un coche ya montado y controlarlo directamente desde el móvil, aunque finalmente se ha optado por adaptar un kit de coche Arduino sin Bluetooth y añadirle el módulo HC-05, además de desarrollar una aplicación propia con MIT App Inventor.

Esta opción se considera la más adecuada porque el kit ya permite disponer de una base mecánica funcional, como el chasis, los motores y la alimentación, y así se puede centrar el trabajo en las conexiones, la programación y el control desde el móvil.

Además, es una alternativa viable en tiempo y presupuesto, ya que los componentes son económicos y existe bastante documentación y ejemplos similares en internet para guiar el montaje y resolver posibles problemas.

La planificación del proyecto seguirá una secuencia sencilla: primero se revisará la documentación de los componentes, después se realizará el montaje del coche, a continuación se programará el Arduino y se creará la aplicación móvil, y por último se harán pruebas, correcciones y la memoria final.

1.5 Metodología del trabajo

La metodología que se ha utilizado para organizar el proyecto ha sido el diagrama de Gantt. Este tipo de herramienta permite visualizar todas las tareas del proyecto de forma ordenada y ver cuánto tiempo dura cada una, desde que empieza hasta que termina.

En resumen, esta metodología sirve para planificarse mucho mejor.

1.6 Estudio económico y presupuestario

| Marca | Descripción | Imagen | Precio |
|-----------------------------|--|--|--------|
| ELEGOO | Cables Jumpers 120 piezas de cable DuPont, 40 pines macho-hembra, 40 macho-macho, 40 hembra-hembra. |  | 8,99€ |
| AZDelivery | Módulo transceptor RF inalámbrico Bluetooth compatible con Arduino y libro electrónico incluido. |  | 10,99€ |
| HanOaki | Kit de coche inteligente de dos ruedas con motor de seguimiento, robot inteligente, chasis con sensor de distancia y cámara. |  | 19,29€ |
| Duracell | Pilas AA |  | 1,50€ |
| Bq ZUM BT-328 Placa Arduino | Arduino con Bluetooth integrado. |  | 34,00€ |
| TOTAL: | | | 74,77€ |

2 Descripción del proyecto

2.1 Análisis de requisitos

Estos son los requisitos principales que tiene que cumplir el proyecto para darlo por terminado y bien hecho:

- El sistema tiene que tener un coche robot montado con su chasis, ruedas, motores y un sistema de alimentación con pilas o batería recargable.
- El microcontrolador debe ser una placa compatible con Arduino UNO, bien programada y capaz de controlar todos los elementos del coche.
- El coche se tiene que poder controlar a distancia por Bluetooth desde un móvil Android, permitiendo como mínimo las órdenes de avanzar, retroceder, girar y parar.
- El coche debe responder a las órdenes de forma estable, sin quedarse colgado ni hacer cosas extrañas durante las pruebas.

2.1.1 Requisitos funcionales

Estos serían los requisitos funcionales del proyecto, es decir, lo que debe ser capaz de hacer el sistema:

- El coche debe recibir órdenes por Bluetooth desde el móvil, de forma que la comunicación entre ambos sea rápida y sencilla.
- El coche debe avanzar, retroceder, girar a la izquierda, girar a la derecha y parar cuando reciba la orden correspondiente, respondiendo correctamente a cada botón o comando enviado.
- El sistema debe encenderse y apagarse mediante un interruptor general de alimentación, para poder controlar todo el montaje de manera cómoda y segura.

- El programa de Arduino debe interpretar correctamente los comandos que llegan por Bluetooth y activarlos en los pines de control del puente H, para que los motores funcionen como toca.
- La aplicación del móvil debe permitir enviar comandos al coche mediante una interfaz sencilla de botones, para que su uso sea fácil e intuitivo.

2.1.2 Requisitos no funcionales

- El sistema debe ser robusto, es decir, funcionar durante las pruebas sin reinicios, bloqueos ni fallos inesperados.
- El coche y el cableado deben ser seguros, evitando cortocircuitos, cables sueltos o partes que puedan dañarse fácilmente al moverlo.
- La aplicación del móvil y el uso del coche deben ser fáciles de usar, con botones claros y un manejo intuitivo para cualquier persona de clase.
- El conjunto debe ser mantenible, de manera que se puedan cambiar pilas, ajustar conexiones o modificar el código sin tener que desmontar todo el proyecto.

2.2 Previsión de tareas de investigación

Para llevar a cabo el proyecto se han identificado varias tareas de investigación que han sido necesarias antes de montar todo.

Primero se investigó el funcionamiento básico de la plataforma Arduino, especialmente cómo programar la placa y cómo controlar motores utilizando un puente H.

También se estudiaron los módulos Bluetooth compatibles con Arduino, comparando diferentes opciones y eligiendo el HC-05 por su facilidad de uso y compatibilidad con dispositivos Android.

Otra tarea importante fue investigar sobre cómo crear aplicaciones móviles con MIT App Inventor, ya que era necesario desarrollar una interfaz que enviará comandos al Arduino.

2.3 Tecnologías

2.3.1 Comparativa de las tecnologías valoradas

Durante el desarrollo del proyecto se han tenido en cuenta varias tecnologías para elegir cual sería la opción más adecuada. La idea era encontrar una solución que fuese sencilla de utilizar, económica y compatible con el resto de componentes.

Una de las opciones valoradas ha sido Arduino Uno con el módulo Bluetooth HC-05, ya que permite establecer una comunicación inalámbrica entre el móvil y el coche de forma bastante simple. Sus principales ventajas son el bajo coste, la facilidad de programación y la gran cantidad de tutoriales y ejemplos disponibles. Como inconveniente, depende de un módulo externo y puede dar problemas si no está bien configurado o conectado.

También tuvimos en cuenta el uso de una placa con Bluetooth y WiFi integrados. Esta opción tiene la ventaja de reducir componentes externos y ofrecer más posibilidades de conexión, pero también hace que el proyecto sea más complejo y obliga a adaptar más la programación si ya está pensado para trabajar con HC-05.

Para el desarrollo de la aplicación móvil se ha valorado MIT App Inventor, ya que permite crear una app de forma visual mediante bloques. Su principal ventaja es que resulta fácil de aprender y es muy útil en proyectos educativos, aunque tiene menos flexibilidad que otros sistemas de desarrollo más avanzados.

2.3.2 Tecnologías escogidas

Las principales tecnologías que hemos escogido son:

1. Copia de la placa de Arduino UNO que me vino con el kit entero del coche.
2. El módulo Bluetooth HC-05 lo elegimos porque es fácil de conectar al Arduino y hay apps gratuitas en Android que ya funcionan con él sin tener que programar nada. No pusimos WiFi ESP32, ya que el Bluetooth es más directo para controlar un coche y no necesitamos internet.
3. Para los motores usamos el puente H L298N, que es el más típico y fiable con Arduino, porque aguanta perfectamente los motores del kit y se conecta con tornillos sin soldar.
4. Para las conexiones entre Arduino y el puente H utilizaremos los cables jumpers, ya que son fáciles de cambiar si hay fallos y no necesitan soldadura.

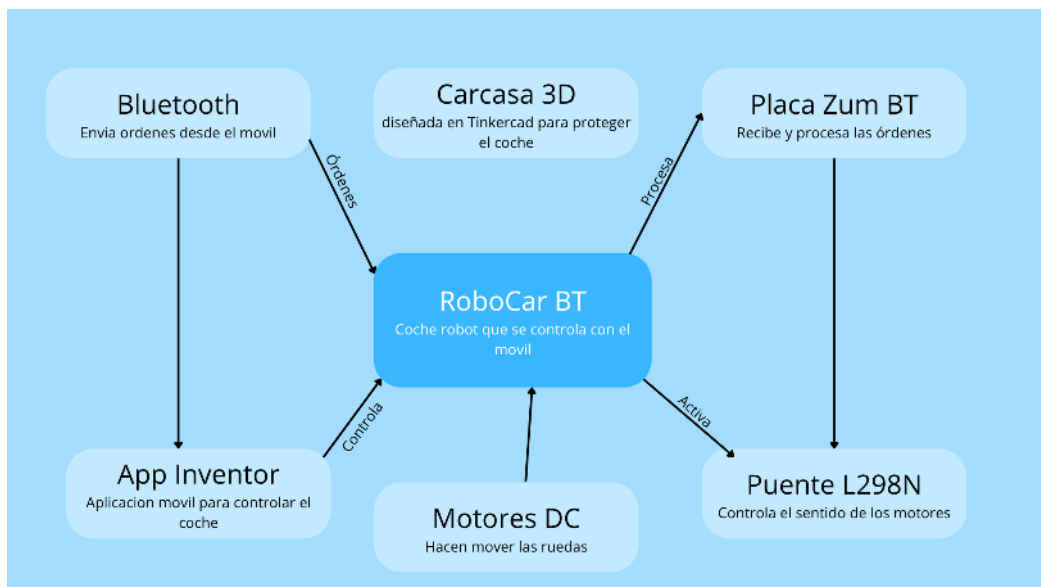
- Para controlar el coche, crearemos una app con MIT App Inventor porque es visual, sencilla para nuestro nivel y podemos personalizar a nuestro gusto, como los botones para avanzar, retroceder, girar y parar en poco tiempo.

2.4 Estructura del proyecto

Este proyecto consiste en un coche robot controlado desde el móvil por Bluetooth. Para hacerlo, utilizamos una placa Arduino, un módulo Bluetooth HC-05, un puente H L298N, motores DC, ruedas y una batería para alimentarlo.

La idea es que la app del móvil mande las órdenes, el Bluetooth las reciba y el Arduino las procese para mover el coche. Después, el puente H se encarga de hacer funcionar los motores según la orden recibida.

En resumen, la estructura del proyecto sería: móvil → Bluetooth → Arduino → puente H → motores → movimiento del coche. Con esto conseguimos un sistema sencillo, práctico y fácil de entender.



2.5 Descripción de los componentes

2.5.1 Componente 1

- Arduino UNO compatible:

Es la parte principal del proyecto. Se encarga de recibir las órdenes y mandar las señales necesarias para mover el coche.

2.5.2 Componente 2

- Módulo Bluetooth HC-05:

Sirve para conectar el móvil con el Arduino sin cables. Recibe los comandos de la app y los envía a la placa.

2.5.3 Componente 3

- Puente H L298N:

Se usa para controlar los motores, ya que el Arduino no puede moverlos directamente. Gracias a este módulo, el coche puede avanzar, retroceder o parar.

2.5.4 Componente 4

- Motores DC:

Son los que hacen que el coche se mueva. Al recibir corriente, hacen girar las ruedas.

2.5.5 Componente 5

- Chasis, ruedas y alimentación:

El chasis es la base donde va montado todo. Las ruedas permiten el movimiento y las pilas dan la energía al sistema.

2.5.6 Componente 6

- MIT App Inventor:

Se ha utilizado para hacer la aplicación del móvil. Con ella podremos crear botones sencillos para controlar el coche por Bluetooth.

2.5.7 Componente 7

- Arduino Zum BT-328

Se ha utilizado para reemplazar al modulo bluetooth y arduino uno, para tenerlo todo en uno, hará lo mismo que el modulo y que el arduino

2.6 Definición de las funcionalidades

2.6.1 Funcionalidad 1: conexión entre el móvil y el coche

Esta funcionalidad permite establecer la comunicación entre la aplicación y la placa Arduino con Bluetooth. El proceso consiste en activar el Bluetooth del móvil, abrir la aplicación y realizar la conexión con el dispositivo para que se puedan enviar órdenes al coche. Esta funcionalidad ha quedado implementada totalmente, ya que sin ella el proyecto no podría funcionar.

2.6.2 Funcionalidad 2: avance del coche

Esta funcionalidad permite mover el coche hacia delante. Cuando el usuario pulsa el botón correspondiente en la aplicación, se envía una orden al Arduino y este activa los motores para que el vehículo avance. Esta funcionalidad ha quedado implementada totalmente.

2.6.3 Funcionalidad 3: retroceso del coche

Esta funcionalidad permite que el coche se desplace hacia atrás. El funcionamiento es similar al anterior: la aplicación envía una orden y la placa activa los motores en el sentido contrario. Esta funcionalidad ha quedado implementada totalmente.

2.6.4 Funcionalidad 4: giro a la izquierda

Esta funcionalidad permite girar el coche hacia la izquierda. Para hacerlo, la aplicación manda una orden concreta y el Arduino modifica el movimiento de los motores para producir el giro. Esta funcionalidad ha quedado implementada totalmente.

2.6.5 Funcionalidad 5: giro a la derecha

Esta funcionalidad permite girar el coche hacia la derecha. El proceso es el mismo que en el giro a la izquierda, pero aplicando la orden correspondiente para cambiar el sentido del movimiento. Esta funcionalidad ha quedado implementada totalmente.

2.6.6 Funcionalidad 6: parada del coche

Esta funcionalidad permite detener el coche en cualquier momento. Cuando el usuario pulsa el botón de parar, la aplicación envía una orden al Arduino y este deja de alimentar los motores, haciendo que el vehículo se detenga. Esta funcionalidad ha quedado implementada totalmente.

3 Desarrollo del proyecto

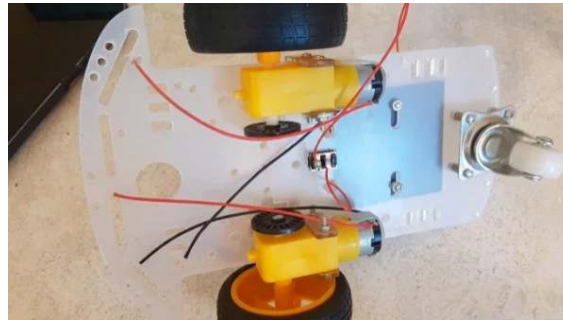
3.1 Montaje del coche

RoboCar BT es un coche robot que se controla desde una aplicación móvil mediante Bluetooth. En este proyecto se ha realizado el montaje completo del coche, la conexión de sus componentes y la programación necesaria para que responda a las órdenes enviadas desde el móvil. Para ello se han utilizado un kit del coche con el driver L298N y los motores, una placa ZUM BT con Bluetooth integrado, cuatro pilas AA, cables jumpers, un interruptor y la aplicación creada con App Inventor.

Antes de empezar, hay que preparar todas las piezas necesarias: el chasis, los motores, las ruedas, la placa ZUM BT, el driver L298N, el portapilas, el interruptor y los cables. Tener todo preparado desde el principio ayuda a trabajar de forma más ordenada y a evitar errores durante el montaje. Lo primero que se hace es colocar el chasis como base del proyecto y fijar los motores en los laterales con sus soportes y tornillos.

A continuación, se fijan los motores a los laterales del chasis y se colocan los cables de conexión. En cada motor se sueldan dos cables, uno negro y uno rojo. Primero se suelda el cable negro, que queda en la parte de arriba, y después el cable rojo, que queda en la parte de abajo. Así se dejan listos para

conectarlos al driver L298N. Una vez puestos, se encajan las ruedas en los ejes de los motores y se comprueba que giren bien.

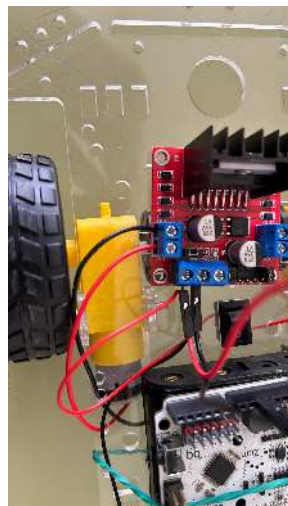


*Anexo F1

Más tarde, se colocan la placa ZUM BT y el driver de motores L298N en la parte superior del coche, dejando espacio suficiente para hacer las conexiones con comodidad. Los motores se conectan al módulo L298N: un motor va a las salidas OUT1 al cable negro y OUT2 al cable rojo y el otro a OUT3 al cable negro y OUT4 al cable rojo, para poder controlar cada lado del coche por separado. Si al probarlo un motor gira al revés, solo hay que cambiar el rojo y el negro de ese motor.

*Anexo F2

OUT1, OUT2:

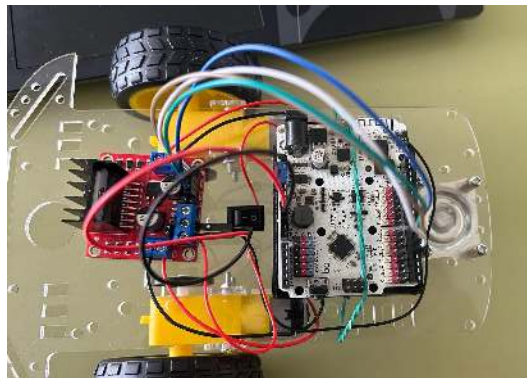


OUT3,OUT4:



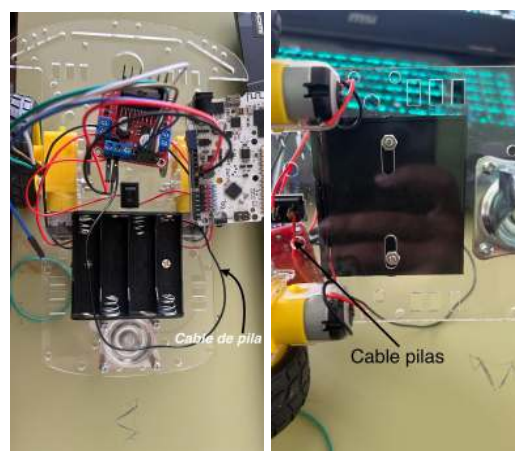
Después se conecta la placa ZUM BT al driver L298N mediante los pines de control. En el programa se usan los pines 7 (Azul), 6 (Verde), 5 (Marrón) y 4 (Blanco) de la placa para enviar las órdenes a las entradas IN1(Azul), IN2 (Verde), IN3 (Marrón) y el IN4 (Blanco) del driver. Esta conexión es la que permite controlar el sentido de los motores y hacer que el coche avance, retroceda, gire o se detenga.

*Anexo F3

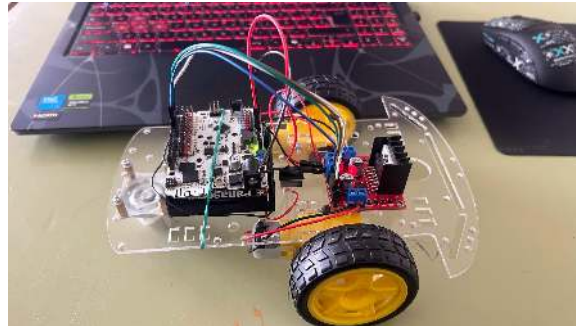


Seguidamente, se coloca el portapilas con las cuatro pilas AA y se conecta la alimentación al sistema. El cable rojo del portapilas va al interruptor y desde el interruptor sale otro cable rojo hacia la alimentación del circuito. El cable negro del portapilas va directamente a GND. El interruptor se coloca en medio del cable rojo para poder encender y apagar el coche sin tener que desconectar las pilas.

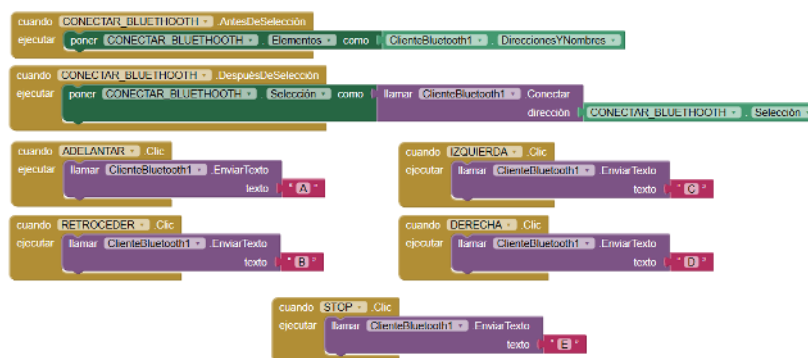
*Anexo F2



Después de eso, se unen las masas de los componentes para que todo el circuito funcione correctamente. Es importante que la placa ZUM BT, el driver L298N y la alimentación compartan la misma masa, porque si no, el coche puede fallar al recibir las órdenes. Antes de seguir, hay que revisar que todos los cables estén bien puestos y que no haya ninguna conexión suelta, si está todo bien se encenderá y parpadeará si no está conectado al bluetooth.

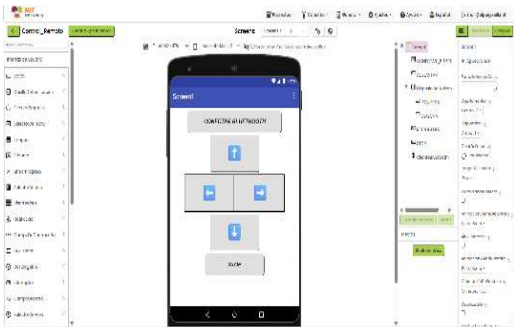


En cuanto a la programación, se hizo para que la placa recibiera comandos por Bluetooth y activará los pines del motor según la orden enviada desde la aplicación. El programa utiliza las letras A, B, C, D y E para indicar las acciones del coche: avanzar, retroceder, girar a la izquierda, girar a la derecha y detenerse. De esta forma, cada botón de la aplicación envía una letra y la placa la interpreta directamente.



La aplicación se creó con MIT App Inventor desde cero. Primero se diseñó la pantalla con un botón o lista para conectar el Bluetooth y varios botones para controlar el coche. Después se programaron los bloques para que cada botón enviara una orden distinta a la placa. Los botones principales son Adelante, Atrás, Izquierda, Derecha y Stop. Cuando el usuario pulsa uno de los botones, la app envía el comando correspondiente y el coche realiza la acción

programada, también se ha el enlace del archivo de la aplicación por si quieren saltar la parte de la aplicación .



[Control Remoto.apk](#)

Al principio se pensó en usar un Arduino Uno con un módulo HC-05 externo. Sin embargo, como surgieron varias dificultades con esa opción, se decidió cambiar a una placa ZUM BT con Bluetooth integrado. Este cambio simplificó el montaje y permitió terminar el proyecto de una forma más estable. Por eso, aunque la idea inicial fue otra, la solución final resultó más práctica para el funcionamiento del coche.

Una vez terminado el montaje, se realizaron pruebas para comprobar que el coche respondía bien a las órdenes de la aplicación. Al verificar que todo funcionaba correctamente, el proyecto quedó listo para usarse. El resultado final es un coche robot controlado desde el móvil mediante Bluetooth, con una estructura sencilla y un funcionamiento práctico, dejaré aquí una demo de como funcionara el coche.

3.2. Creación del chasis

Para mejorar el proyecto, se ha diseñado una carcasa en 3D utilizando Tinkercad. Esta carcasa se ha creado para proteger mejor los componentes del coche y también para darle un aspecto más cuidado y profesional. Además, ayuda a sujetar mejor algunas partes del montaje y hace que el robot tenga una estructura más ordenada.

El diseño se ha hecho pensando en las medidas del coche y en los elementos que lleva montados, para que todo encaje correctamente y no interfiera en su funcionamiento. Gracias a esto, el proyecto no solo cumple su función técnica, sino que también tiene una presentación visual mejorada. Este trabajo con Tinkercad ha sido útil porque nos ha permitido aplicar conocimientos de diseño 3D y adaptarlos a una necesidad real del proyecto. Al final en la parte de anexo, se ha añadido un enlace del vídeo y capturas para mostrar cómo ha quedado el resultado final.

3.3 Creación Página web

Finalmente, también se ha creado una página web relacionada con el proyecto. Al entrar en ella, primero aparece una introducción donde se explica de forma general en qué consiste el proyecto RoboCar BT. Después, la web incluye un apartado de comentarios pensado para que cualquier persona pueda escribir dudas o preguntas sobre el manual de instalación.

Este espacio de comentarios permite que otras personas también puedan responder a las dudas planteadas, de manera que se pueda crear una especie de ayuda entre usuarios. Así, si alguien no entiende alguna parte del manual, puede dejar su comentario y otra persona puede contestarle si sabe la respuesta. Hay otro apartado donde está el manual en sí y si quieren un manual completo con fotos, hay un pdf si lo abren ahí estará el manual completo. Además, los administradores de la página, que son los usuarios que conocen la contraseña, pueden eliminar comentarios si es necesario.

Para su desarrollo se han utilizado tecnologías como Apache, SQLite, CSS, HTML y PHP, lo que ha permitido crear una web funcional y ordenada. Las imágenes de cómo ha quedado la página web se han incluido en la parte de anexos para mostrar el resultado final.

4 Conclusiones

4.1 Conclusiones generales del proyecto

Este proyecto ha servido para aprender a trabajar de una forma más organizada y a resolver problemas que han ido apareciendo durante el desarrollo. Aunque al principio parecía un proyecto sencillo, al final han surgido varias dificultades, sobre todo con la parte del Bluetooth, y eso ha hecho que hubiera que buscar soluciones y adaptarse a los cambios.

El proyecto ha sido útil para poner en práctica conocimientos de programación, electrónica, diseño 3D, montaje de componentes y de diseño de la página web. También ha ayudado a entender mejor cómo se relacionan entre sí las distintas partes de un sistema, desde la aplicación móvil hasta el control del coche.

A nivel personal y profesional, el desarrollo del proyecto ha servido para ganar más confianza trabajando con herramientas como Arduino, Tinkercad y la impresión 3D. Además, ha permitido desarrollar más la capacidad de organizar el trabajo, corregir errores y seguir adelante cuando algo no funcionaba como se esperaba.

En general, se puede decir que ha sido un proyecto bastante completo y práctico, porque ha permitido aprender cosas nuevas y aplicar conocimientos que se han ido viendo durante el curso.

4.2 Consecución de los objetivos

Los objetivos planteados al inicio de la memoria se han cumplido en su mayoría. El objetivo principal, que era conseguir controlar el coche desde el móvil, se ha alcanzado, aunque no exactamente con la solución que se había pensado al principio. El módulo Bluetooth que teníamos previsto no funcionó correctamente, así que finalmente se decidió utilizar un Arduino con Bluetooth, lo que permitió sacar el proyecto adelante.

Además, también se ha conseguido crear una página web como parte del proyecto, lo que ha servido para complementar el trabajo y darle una presentación más completa. Por otra parte, se ha diseñado una carcasa para el coche con TinkerCad y después se ha imprimido en 3D, algo que ha mejorado bastante el aspecto final del montaje.

En general, se puede decir que los objetivos principales sí se han cumplido, aunque algunos han necesitado cambios durante el desarrollo. Esto ha pasado sobre todo por los problemas con el Bluetooth, que obligaron a adaptar la idea inicial para poder terminar el proyecto correctamente.

4.3 Valoración de la metodología y planificación

La planificación se ha seguido bastante bien, ya que el proyecto se organizó por fases: primero la planificación, después el diseño, luego el desarrollo, las pruebas, el despliegue y por último el cierre. Aun así, no todo ha salido como estaba previsto al principio, porque hemos tenido bastantes problemas con el módulo Bluetooth y eso ha hecho que algunas tareas tardaran más de lo esperado.

La metodología por fases nos ha parecido adecuada, porque nos ha permitido ir avanzando paso a paso y comprobar cada parte antes de pasar a la siguiente. Gracias a eso, hemos podido corregir fallos sobre la marcha y seguir trabajando sin desordenarnos. Además, para mejorar el proyecto se nos ocurrió añadir una página web, que nos ha servido para presentar mejor la información y organizar el contenido de forma más clara. En el manual se explica también qué hace esta página web y cómo complementa el proyecto.

En general, se puede decir que la planificación nos ha servido como guía, pero ha habido que hacer algunos ajustes para poder acabar el proyecto bien. Sobre todo, el tiempo extra que hemos necesitado con la conexión Bluetooth ha sido lo que más ha afectado al ritmo de trabajo.

Si quieres, te lo puedo dejar ahora mismo en un tono más formal de memoria o más sencill

5 Visión de futuro

De cara al futuro, este proyecto se puede seguir mejorando bastante, tanto en la parte de la aplicación como en la del propio coche. Una de las ideas principales es hacer una app más completa, con una interfaz mejor organizada, más botones y nuevas opciones de control para que sea más cómoda de usar desde el móvil. En proyectos con Arduino y App Inventor es habitual ampliar poco a poco la aplicación para añadir más funciones y mejorar la comunicación con el dispositivo.

Además, también se tiene pensado añadir nuevos componentes al coche para que no solo se mueva, sino que pueda realizar más acciones. Entre las mejoras previstas están incorporar un sensor de movimiento y una mini pantalla, de forma que el coche pueda ofrecer más información y tener más utilidades. Este tipo de ampliaciones permitirían que el proyecto evolucionara y no se quedara solo en un control básico por Bluetooth.

Otra posible mejora sería ajustar mejor la programación del Arduino para que el coche responda con más precisión a algunas órdenes, por ejemplo en los giros o en nuevas funciones que se añadan más adelante. También se podría mejorar la conexión entre la app y el coche para hacerla más estable y más completa con el paso del tiempo.

En general, la idea es que este proyecto no termine solo con la versión actual, sino que pueda servir como base para seguir aprendiendo y añadiendo nuevas funciones en el futuro. Aunque por tiempo no se han podido desarrollar todas estas mejoras, sí quedan planteadas como una continuación lógica del trabajo realizado hasta ahora.

6. Glossario

- Arduino: placa electrónica programable utilizada para controlar el funcionamiento del proyecto.
- Bluetooth: tecnología de comunicación inalámbrica que permite enviar órdenes desde el móvil al vehículo.
- HC-05: módulo Bluetooth utilizado para la comunicación inalámbrica entre un móvil y una placa electrónica.
- ZUM BT: placa compatible con Arduino que incorpora Bluetooth integrado.
- L298N: controlador de motores que permite gestionar el movimiento de los motores del robot.
- Motor DC: motor de corriente continua utilizado para hacer mover las ruedas.
- Chasis: estructura base sobre la que se montan todos los componentes del robot.
- App Inventor: entorno de programación visual utilizado para crear la aplicación móvil.
- Tinkercad: herramienta de diseño 3D utilizada para crear la carcasa del robot.
- Portapilas: soporte utilizado para colocar las pilas que alimentan el sistema.
- Microcontrolador: componente principal de la placa que ejecuta el programa.
- Código: conjunto de instrucciones que indican a la placa qué debe hacer.
- Página web: sitio creado para presentar el proyecto, explicar sus partes y mostrar la información de forma organizada.
- HTML: lenguaje utilizado para estructurar el contenido de la página web.

- CSS: lenguaje utilizado para dar estilo y mejorar el diseño visual de la página web.
- PHP: lenguaje de programación utilizado en la parte dinámica de la web, como los comentarios y el control de administración.
- Servidor web: sistema que permite mostrar la página web a través de una dirección o URL.
- Apache: programa que actúa como servidor web y permite ejecutar páginas PHP.
- URL: dirección que se utiliza para acceder a la página web desde el navegador.
- Comentarios: apartado de la web donde los usuarios pueden dejar mensajes o dudas sobre el proyecto.
- Administrador: persona con permiso para borrar comentarios o gestionar ciertas partes de la web.

7. Bibliografía

08/04/26:

▶ Cómo armar y programar un carrito con Arduino controlado por Bluetooth ...

▶ 💡 CURSO de ARDUINO desde cero en español [2021] - Capítulo #01 📱

<https://codigo21.educacion.navarra.es/autoaprendizaje/primeros-pasos-con-ap-p-inventor-2/>



Autor: Antonio Ricoy Riego

15/04/2026:

▶ 💡 Cómo PROGRAMAR arduino UNO desde CERO[2021] - Capítulo #02 📱

22/04/2026:

▶ How to Make a Bluetooth Controlled Car Using Arduino UNO | DIY Arduino ...

30/04/2026:

▶ ✅ Configurar el módulo Bluetooth HC-05 con los comandos AT

07/05/2026:

▶ APRENDE MODELADO 3D EN UN SOLO VIDEO | GUÍA DIFINITIVA de BLEND...

▶ ¿Cómo usar Blender 3D? 😊 | Tutorial para principiantes DESDE CERO en e...

▶ Como diseñar carros en 3D! 🚗

▶ 🔥 Cómo Usar TinkerCAD 🔥 TUTORIAL de Diseño 3D para Principiantes

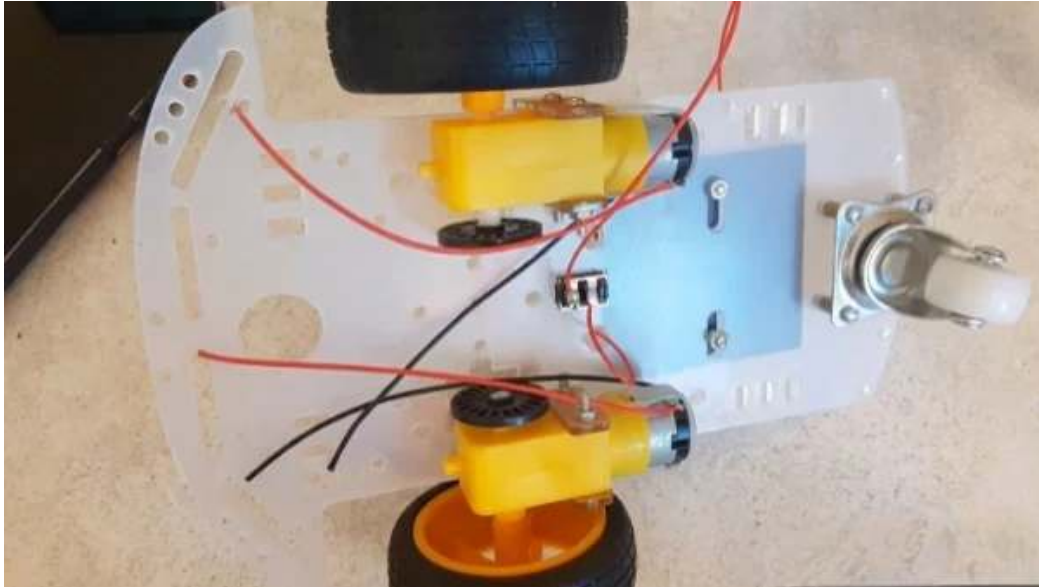
11/05/2026

<https://github.com/bq/ZUMCore>

8 Anexos

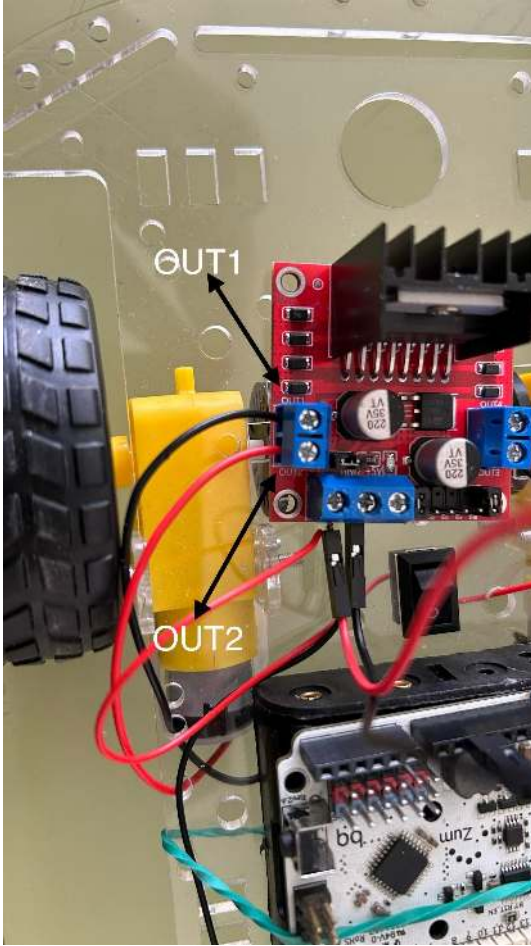
8.1 Fotos

*Anexo F1

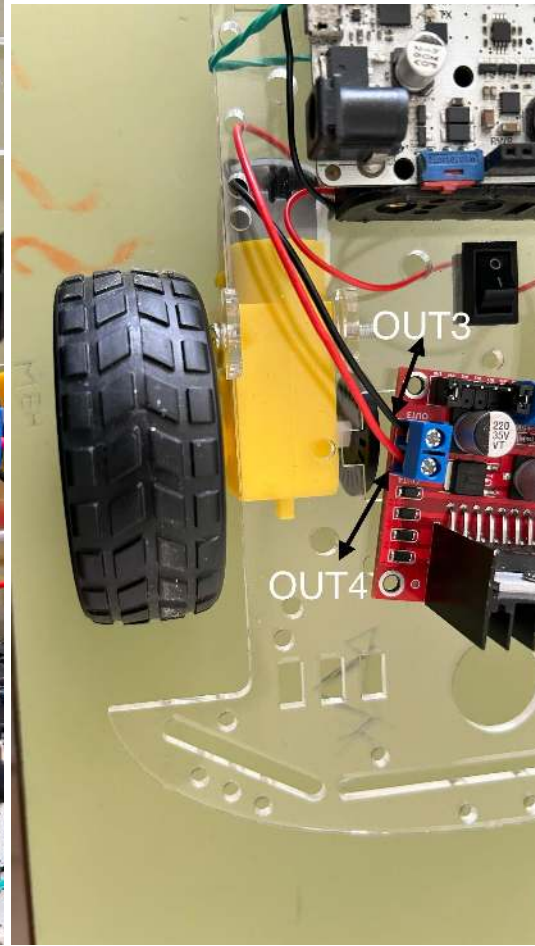


*Anexo 2

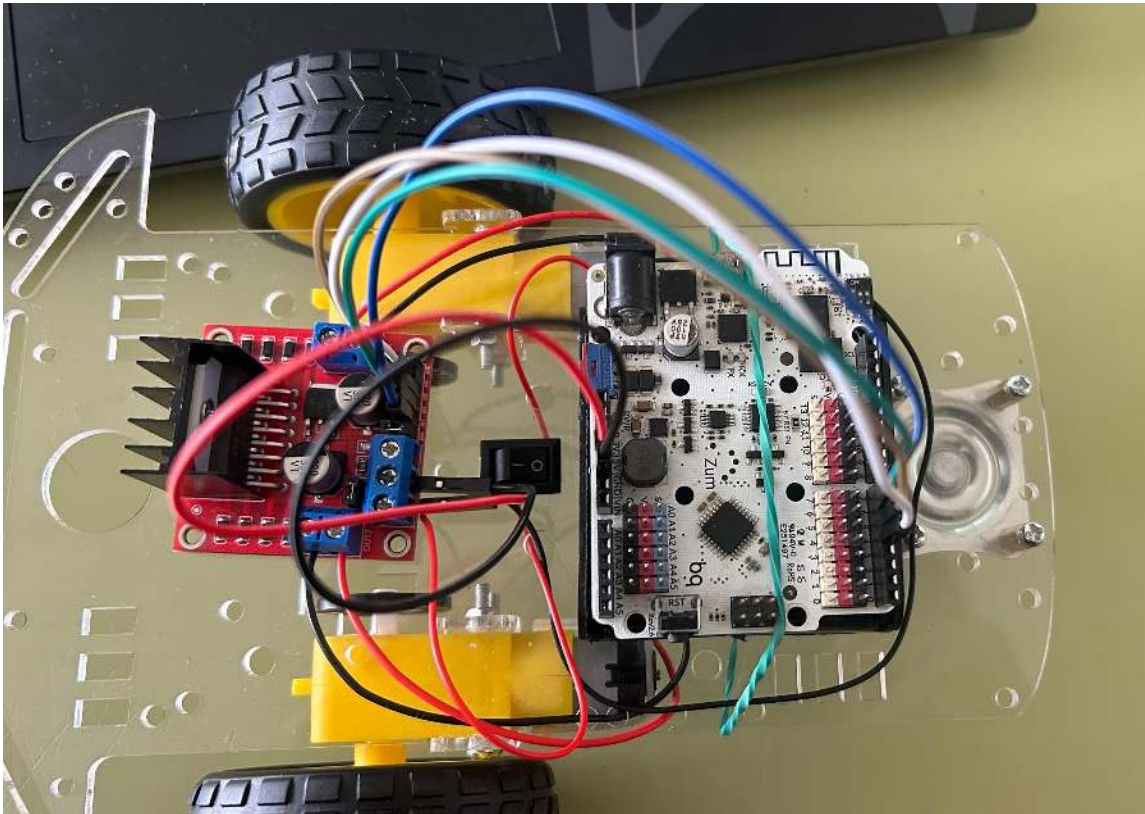
OUT1, OUT2:



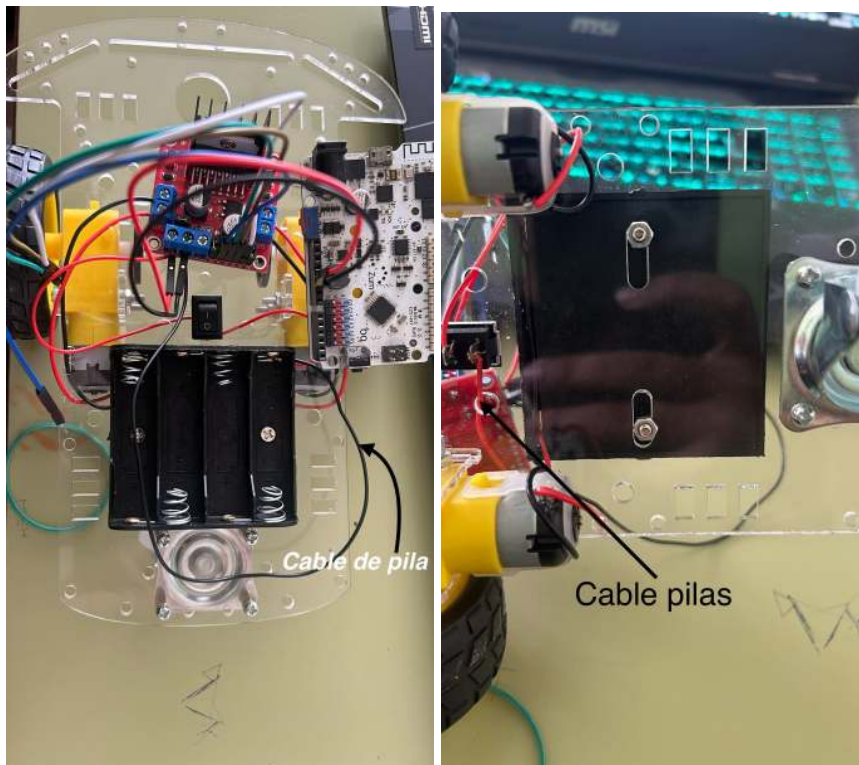
OUT3, OUT4:



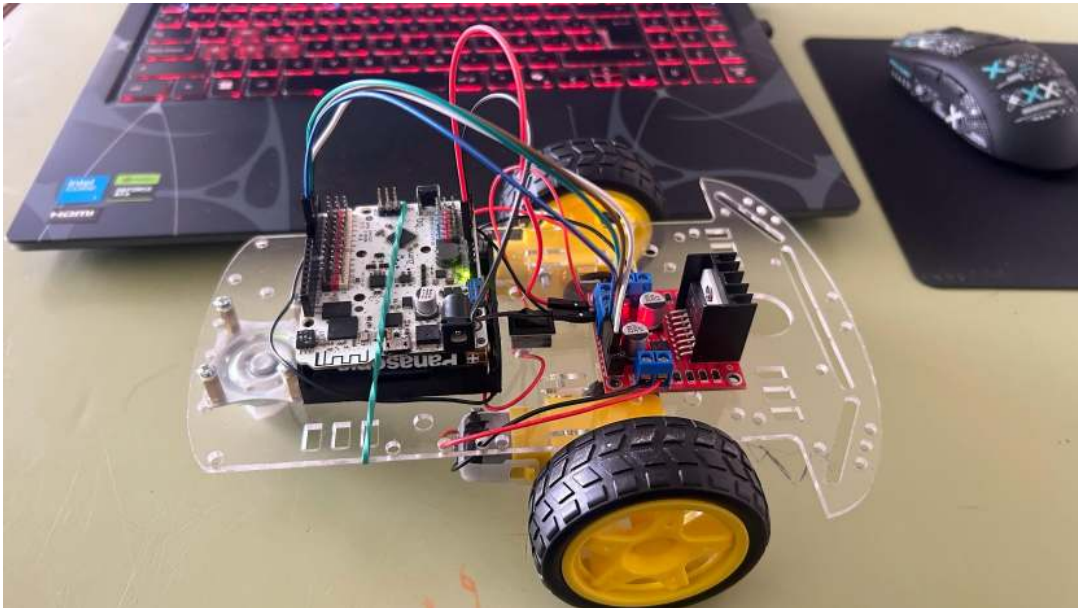
*Anexo 3



*Anexo 4



*Anexo 5



*Anexo 6

```
cuando CONECTAR_BLUETHOOTH . AntesDeSelección
ejecutar poner CONECTAR_BLUETHOOTH . Elementos como ClienteBluetooth1 . DireccionesYNombres

cuando CONECTAR_BLUETHOOTH . DespuésDeSelección
ejecutar poner CONECTAR_BLUETHOOTH . Selección como llamar ClienteBluetooth1 . Conectar
dirección CONECTAR_BLUETHOOTH . Selección

cuando ADELANTAR . Clic
ejecutar llamar ClienteBluetooth1 . EnviarTexto
texto " A "

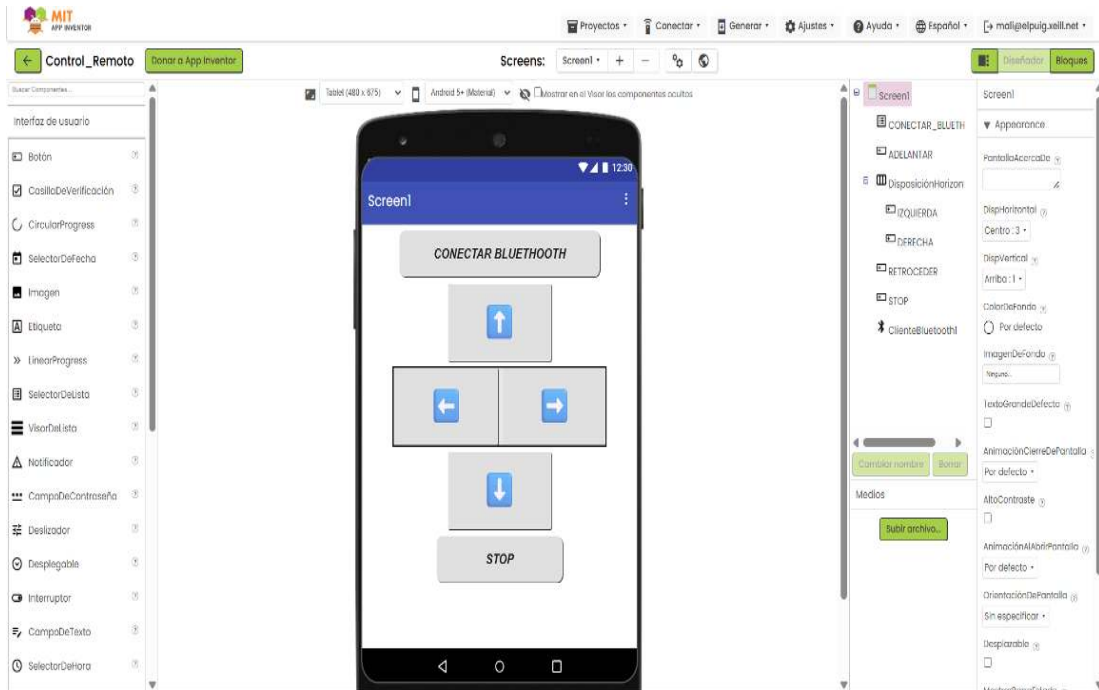
cuando IZQUIERDA . Clic
ejecutar llamar ClienteBluetooth1 . EnviarTexto
texto " C "

cuando RETROCEDER . Clic
ejecutar llamar ClienteBluetooth1 . EnviarTexto
texto " B "

cuando DERECHA . Clic
ejecutar llamar ClienteBluetooth1 . EnviarTexto
texto " D "

cuando STOP . Clic
ejecutar llamar ClienteBluetooth1 . EnviarTexto
texto " E "
```

*Anexo 7



8.2 Código Arduino:

Este es el código que se ha usado para que el Arduino funcione y mande las órdenes al coche. Para poder subirlo bien, hace falta tener instalados los drivers en el ordenador, porque si no, la placa puede no detectarse correctamente en Arduino IDE. Así, el Arduino elegido para este proyecto puede conectarse bien con el programa y hacer que el coche responda a lo que se le envía desde la aplicación.

```
#include <BitbloqSoftwareSerial.h>

BqSoftwareSerial board_bluetooth(0, 1, 19200);

char comando;

int IN1 = 7;
int IN2 = 6;
int IN3 = 5;
int IN4 = 4;

void setup() {
  board_bluetooth.begin(19200);

  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
}

void loop() {

  if (board_bluetooth.available()) {

    comando = board_bluetooth.read();

    switch (comando) {

      case 'A': // Adelante
        digitalWrite(IN1, HIGH);
        digitalWrite(IN2, LOW);
        digitalWrite(IN3, HIGH);
        digitalWrite(IN4, LOW);
        break;
```


```
case 'B': // Atrás
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    break;

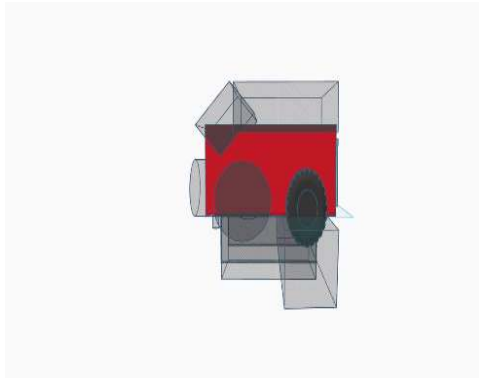
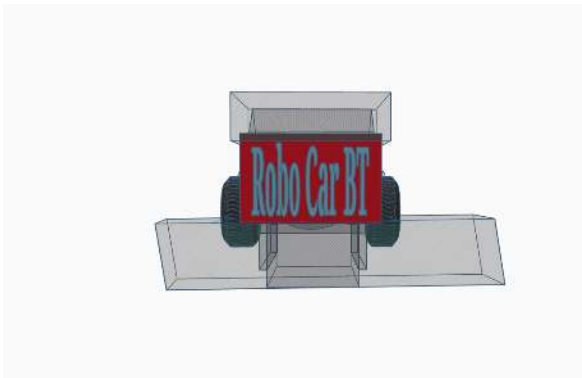
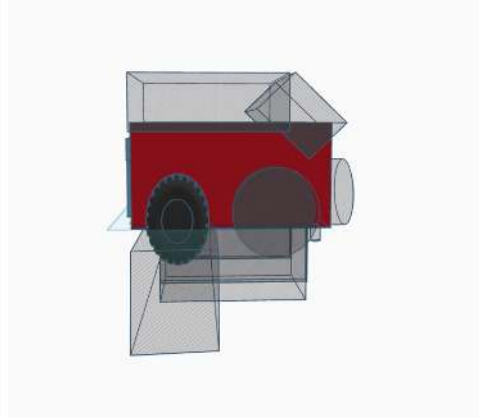
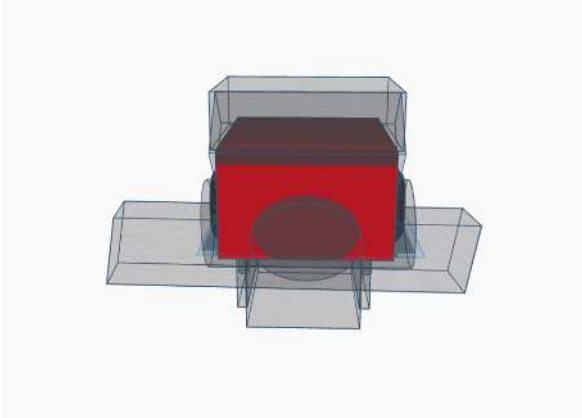
case 'C': // Izquierda
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    break;

case 'D': // Derecha
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    break;

case 'E': // Stop
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
    break;
}
}
```


8.3 Carcasa 3D en Tinkercad:

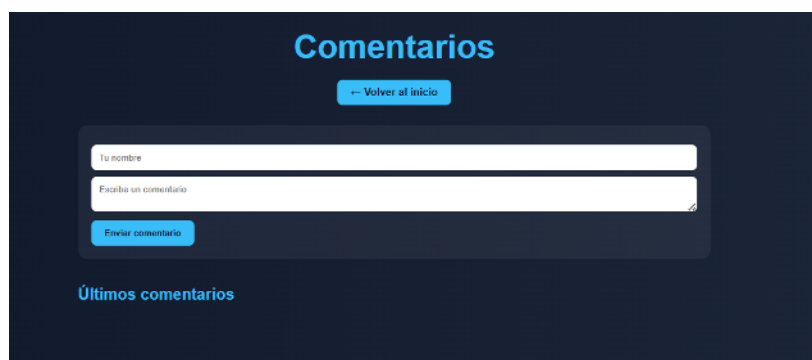
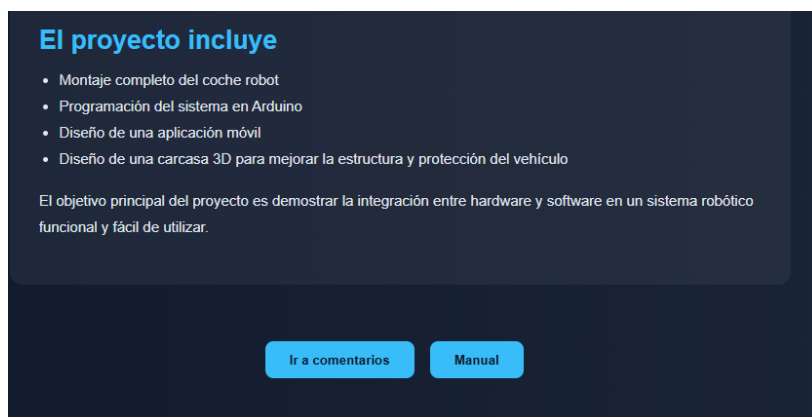
 Video-3D_RoboCar BT.mp4



8.4 Página web

Aquí están las imágenes de cómo ha quedado la página web, y también se ha añadido un vídeo para mostrar, de forma general, cómo funciona cada apartado.

 Demostracion-pagina web.mp4



Manual RoboCar BT

En este manual se explica cómo se realizó el montaje, la conexión y la programación del proyecto RoboCar BT paso a paso.

Materiales utilizados

- Placa ZUM BT con Bluetooth integrado
- Driver L298N
- Motores DC
- Chasis del coche
- 4 pilas AA
- Portapiñas
- Interruptor
- Cables jumpers
- Aplicación creada con MIT App Inventor

Montaje del coche

Primero se prepara el chasis y se colocan los motores en los laterales usando soportes y tornillos. Después se sueldan los cables rojo y negro en cada motor para conectarlos al driver L298N.

Una vez colocados los motores, se montan las ruedas y se comprueba que giren correctamente.

Resultado final

Después de realizar todas las pruebas, el coche quedó funcionando correctamente y respondiendo a las órdenes enviadas desde el móvil.

El resultado final es un coche robot funcional, sencillo y práctico, desarrollado para aprender programación, robótica y electrónica.

[← Volver al inicio](#) [Comentarios](#) [Ver manual completo](#)

RoboCar BT | Manual del proyecto